VideoExplorer: Advancing Long-Horizon Video Understanding via Hierarchical Orchestration

Huaying Yuan
Gaoling School of Artificial Intelligence, Renmin University of China
Beijing, China
hyyuan@ruc.edu.cn

Zheng Liu*
Beijing Academy of Artificial Intelligence
Beijing, China
Hong Kong Polytechnic University
Hong Kong, China

Junjie Zhou
Beijing University of Posts and Telecommunications
Beijing, China

Hongjin Qian
Peking University
Beijing, China
Beijing Academy of Artificial Intelligence
Beijing, China

Yan Shu
University of Trento
Trento, Italy

Nicu Sebe
University of Trento
Trento, Italy

Ji-Rong Wen*
Gaoling School of Artificial Intelligence, Renmin University of China
Beijing, China

Zhicheng Dou
Gaoling School of Artificial Intelligence, Renmin University of China
Beijing, China

Abstract

Current agentic frameworks for Long-Video Understanding (LVU) remain limited by two critical problems: **ineffective control**, where traditional monolithic agents struggle with high-branching, multi-granularity decision processes; and **inefficient supervision**, where sparse, outcome-based feedback fails to guide long-horizon reasoning. To resolve these challenges, we propose **VideoExplorer**, a novel agentic system designed to advance long-video reasoning on top of **structured control** and **trajectory-level optimization**. First, VideoExplorer innovates a hierarchically orchestrated framework: it employs a planning agent to focus on creating high-level reasoning strategies and specialized sub-agents (including a temporal grounder and a visual perceiver) to accomplish fine-grained reasoning executions, thereby substantially reducing the complexity of reasoning process. Second, VideoExplorer introduces a novel optimization approach, Trajectory level Direct Preference Optimization (TDPO), to mitigate inefficient supervision. Unlike standard methods that optimize single turn responses, TDPO aligns the entire planning trajectory, including evidence routing and termination decisions, with end task success, which effectively mitigates premature commitment and compounding errors. To better support the conduct of TDPO, we further create fine-grained supervision data via a teacher-guided, difficulty-adaptive sampling process. Extensive experiments on MLVU, LVBench, and MH-NIAH demonstrate that VideoExplorer consistently outperforms monolithic baselines in both accuracy and efficiency, validating the effectiveness of structured control and trajectory-level optimization in long-video reasoning. Our code is available in this repository[1].

CCS Concepts

• **Information systems → Multimedia information systems**.

Keywords

Long Video Understanding; Agentic Tool Using

ACM Reference Format:
Huaying Yuan, Zheng Liu, Junjie Zhou, Hongjin Qian, Yan Shu, Nicu Sebe, Ji-Rong Wen, and Zhicheng Dou. 2026. VideoExplorer: Advancing Long-Horizon Video Understanding via Hierarchical Orchestration. In *Proceedings of the 32nd ACM SIGKDD Conference on Knowledge Discovery and Data Mining V.2 (KDD '26), August 09–13, 2026, Jeju Island, Republic of Korea.* ACM, New York, NY, USA, 12 pages. https://doi.org/10.1145/3770855.3817705

1 Introduction

Long-video understanding (LVU) is increasingly important for data-intensive video applications, including embodied AI [31], anomaly detection [17, 32], and movie analysis [1]. Although current Vision Language Models (VLMs) [10, 26, 33, 38] perform effectively on short clips, their performance degrades with increasing video length and task complexity. To overcome the context-length bottleneck, recent works [9, 28, 29, 34] have adopted agentic LVU frameworks, where a text-based reasoner iteratively queries the video through external tools (e.g., retrieval, captioning, or perceptual modules) rather than processing the entire video stream. This agentic workflow reduces long-context cost by selectively processing only a

*Corresponding authors.

KDD '26, Jeju Island, Republic of Korea

ACM ISBN 979-8-4007-2259-2/2026/08
https://doi.org/10.1145/3770855.3817705

[1] https://github.com/RUC-NLPIR/VideoDeepResearch

small number of segments of the timeline, thereby enabling practical analysis of hour-long videos.

Despite variations in their tools and pipelines, existing agentic LVU systems share two coupled bottlenecks: a **control mismatch** and a **supervision mismatch**. On the control side, a single generalist planner must simultaneously (i) decompose the task, (ii) localize relevant temporal windows, (iii) select an appropriate level of perception granularity, (iv) formulate specific queries about candidate clips, and (v) aggregate evidence into a coherent final answer. Consequently, inference over long videos becomes *a high-branching sequential decision process* whose action space factorizes into "tool choice × query formulation × temporal span selection × clip-level questioning". This leads to diffuse causality, where failures can originate ambiguously from flawed task decomposition, imprecise temporal grounding, or insufficient perception detail. As a result, the process becomes difficult to inspect, debug, and stabilize.

On the supervision side, training signals are often solely based on outcome (e.g., final answer accuracy). This yields **sparse and delayed feedback** that is insufficient to support temporal credit assignment during the model's multi-step exploration. This weak supervision leads to frequent long-horizon failures, such as compounding errors and redundant exploration, and it also induces premature commitment, such as early stopping and hallucinated justifications. Furthermore, standard supervised fine-tuning (SFT) is unable to correct errors that arise only after the model deviates from the expert trajectory, due to exposure bias.

To resolve **control–supervision mismatches** in agentic long-video understanding, we introduce **VideoExplorer**, a hierarchically orchestrated agentic system trained with trajectory-level preference optimization to stabilize long-horizon reasoning. VideoExplorer regains control via modular orchestration with explicit interfaces: it decouples high-level orchestration (planning, evidence integration, stopping) from low-level execution (temporal retrieval, clip inspection, post-retrieval verification). By structuring the agent into a strategic-steering planner with well-defined interfaces to specialist sub-agents (temporal grounder/visual perceiver), it turns a monolithic, high-branching action space into an ordered stream of semantically grounded sub-problems, thereby sharply reducing search complexity. Crucially, low-level skills are executed by specialized sub-agents that are first warmed up with task-oriented training and then frozen. High-level control and decision-making are learned via reinforcement training of a planner that orchestrates these specialized sub-agents. This reframes the complex end-to-end policy learning problem as "learning how to schedule and decompose," thereby improving sample efficiency and training stability. By freezing execution-level behaviors, we make the sub-agent interface effectively stationary, so the planner learns over a lower-variance signal and can focus on a stable high-level strategy.

To further mitigate the supervision mismatch, we propose a **Trajectory-level Direct Preference Optimization (TDPO)**, a novel approach to directly shape long-horizon decision-making. Unlike standard DPO, which primarily optimizes single-turn responses (i.e., the final answer), TDPO extends preference optimization to the *entire trajectory*, aligning multi-step planning and exploration with end-task success. TDPO constructs trajectory-level preference pairs by contrasting a verified oracle trajectory with well-structured decision-making against a planner-generated trajectory that leads to incorrect outcomes—often due to premature commitment, early stopping, or evidence misrouting. Importantly, TDPO is applied only to train the high-level planner while keeping the sub-agents (temporal grounder and visual perceiver) frozen. This prevents optimization signals from backpropagating into low-level execution branches, concentrating updates on high-level planning, evidence routing, and termination decisions. As a result, TDPO improves the stability of the planning process and scales effectively even as low-level traces become long and heterogeneous, enabling more robust long-video reasoning.

Existing video training datasets [11, 33] are QA-only and lack fine-grained, trajectory-level supervision. We address this by augmenting the LVU training set [11] with a two-stage pipeline: a strong teacher generates expert trajectories, then a **difficulty-adaptive sampling** scheme filters and upweights hard cases. Specifically, the teacher [23] serves as the planner and temporal grounder, paired with a pretrained VLM [2] as the visual perceiver. Given each long-horizon question, we generate multi-step trajectories and retain only those with correct final answers, then upsample challenging examples based on the difficulty estimated in the first pass. These trajectories provide step-level supervision to warm up the planner and temporal grounder before preference optimization. TDPO then samples candidate trajectories from the SFT checkpoint and constructs preference pairs using answer correctness and trajectory quality, enabling robust long-horizon reasoning without costly human annotation.

Collectively, the hierarchical architecture optimized via trajectory-level objectives enables **VideoExplorer** to scale long-video reasoning through *structured control* rather than brute-force context expansion. We conduct extensive experiments on MLVU [35], LVBench [27], and MH-NIAH [8], and show that VideoExplorer consistently outperforms monolithic baselines in both accuracy and efficiency. In summary, our contributions are:

(1) We propose **VideoExplorer**, a hierarchical Orchestration architecture that stabilizes long-horizon LVU by decoupling orchestration from execution.

(2) We introduce **TDPO** for LVU, a trajectory-level preference optimization paradigm that improves reasoning stability and evidence verification, addressing the sparse-reward bottleneck.

(3) Extensive experiments on widely-used benchmarks show that VideoExplorer consistently outperforms monolithic baselines in both accuracy and efficiency.

2 Related Work

2.1 Visual Language Models

The evolution of Visual Language Models (VLMs) has progressed from image understanding to video understanding. Current VLMs, such as the LLaVA-Video series [10, 12, 33], Qwen2/2.5-VL [2, 26], and InternVL3 [38], leverage powerful visual encoders and large language backbones to achieve impressive performance on minute-level clips. However, their reliance on dense frame sampling prevents scalability to hour-long videos due to quadratic token complexity. With the rise of reasoning models [23], some works [5, 7] have attempted to incorporate internal chains of thought into VLMs to enhance their understanding of difficult problems. However, the input context windows of these approaches remain short (typically

within 64 frames), making them ill-suited for long-form video understanding. Long-Context VLMs [4, 8, 13, 18, 21, 22] attempt to mitigate this by extending context windows via mechanisms like token compression or sparse attention, theoretically supporting thousands of frames. Yet, these works largely treat all visual tokens uniformly, even though in long-video understanding, the answer-relevant segments typically comprise only a small fraction of the full content. As video duration increases, this uniform allocation leads to substantial wasted computation and memory. Unlike these methods that passively ingest massive token sequences—leading to high redundancy and attention degradation, **VideoExplorer** extends context-fixed reasoning with active information acquisition, enabling it to scale more effectively to long videos under a constrained token budget.

2.2 Agentic Long Video Understanding

To bypass context limits, recent work adopts agentic frameworks where an LLM controller interacts with videos via external tools. Early systems, such as VideoAgent [28], VideoTree [29], and others [6, 9, 15], often convert videos into static textual memories (e.g., dense captions) for querying, which is efficient but introduces an *information bottleneck* by discarding uncaptured visual details. More recent approaches [3, 20, 34] enable iterative segment querying, but typically rely on a **monolithic planner**, yielding a high-dimensional action space and trajectory drift, and are trained mostly with outcome-only SFT or prompt engineering. Multi-agent variants [6, 30, 36] further decompose the workflow: VideoMultiAgents [6] aggregates reports from modality-specialized agents; and ReAgent-V [36] introduces a critic agent to refine a target agent for higher accuracy. However, they largely focus on modular perception or post-hoc correction, leaving the core long-horizon control and temporal credit assignment under-specified; as a result, the planner still operates in a high-branching space and remains prone to trajectory drift. In contrast, **VideoExplorer** targets the *control–supervision mismatches* in long-video reasoning by (i) **hierarchical orchestration** that decouples high-level planning/evidence routing/stopping from low-level grounding and clip inspection, and (ii) **TDPO**, which trains only the planner for high-level decision making and planning with trajectory-level preferences, while freezing specialized sub-agents to stabilize long-horizon behavior.

3 Methodology

3.1 Overview

As discussed in the introduction, monolithic agentic LVU suffers from coupled **control–supervision mismatches**: a monolithic planner faces a high-branching spatiotemporal action space, yet receives sparse and delayed outcome supervision, leading to *diffuse causality* and unstable long-horizon reasoning. To address this, we introduce **VideoExplorer**, a hierarchically orchestrated framework that stabilizes long-video understanding through *structured control* and *trajectory-level supervision*. Concretely, VideoExplorer (1) formulates long-video reasoning as a *hierarchically controlled Partially Observable Markov Decision Process*, (2) *decoupling orchestration from execution* with well-defined interfaces, where a central planner performs high-level strategic reasoning, while specialized sub-agents handle temporal grounding and visual perception; and (3) trains the high-level planner with **Trajectory-level Direct Preference Optimization (TDPO)** to optimize preferences over entire decision trajectories rather than only final answers. The overall architecture is illustrated in Figure 1.

3.2 Problem Formulation

We formalize Long-Video Understanding (LVU) as a sequential decision-making problem under partial observability, modeled as a tuple $(\mathcal{S}, \mathcal{A}, \mathcal{O}, \mathcal{T}, \mathcal{R})$.

- **State** $\mathcal{S}$: The full latent state includes the raw pixel stream of the long video $V \in \mathbb{R}^{T\times H\times W\times 3}$ and the user query q. Due to context constraints, V cannot be fully observed in a single pass.
- **Action** $\mathcal{A}$: The agent selects actions a_t based on a pre-defined action space, including *planning thought*, *tool invocation*(e.g. retriever), and *answer generation*.
- **Observation** $\mathcal{O}$: Upon executing a_t (e.g., retrieving a clip), the agent receives an observation o_t (e.g., relevant time windows, visual descriptions, or question-conditioned visual perceptions), updating its internal history $h_t = \{q, a_1, o_1, \ldots, a_t\}$.
- **Policy** π_θ**:** The objective is to learn a policy $\pi_\theta(a_t \mid h_{t-1})$ that maximizes the expected reward:

$$J(\theta) = \mathbb{E}_{\tau\sim\pi_\theta}\left[\sum R(s_t, a_t)\right],$$

  where the reward is typically sparse (non-zero only at the terminal step based on answer correctness).

The Structural Challenge: As discussed in the introduction, monolithic agents suffer from a *high-branching factor*. **The variance of the gradient estimator $\nabla_\theta J$ grows exponentially with the trajectory length $|\tau|$.** To address this, **VideoExplorer** factorizes the monolithic policy into a hierarchical structure: a central *Planner* π_{plan} that operates on high-level strategic reasoning, and specialized *Sub-agents* (Temporal Grounder π_{g} and Visual Perceiver π_{p}) that handle specific task execution.

3.3 The VideoExplorer Architecture

To address the structural misalignment between control complexity and supervision density, VideoExplorer factorizes the monolithic policy into an **Planner** for strategic reasoning with two specialized sub-agents: a **Temporal Grounder** for task-oriented temporal localization, and a **Visual Perceiver** for temporal-scalable visual inspection, enabling precise functional specialization and stable trajectory learning.

3.3.1 The Strategic-Steering Planner. The Planner, parameterized by an LLM π_{plan}, is the *high-level orchestrator* of **VideoExplorer**. It operates in a semantic action space and issues structured commands to specialized sub-agents, decoupling orchestration from long, noisy retrieval and visual-inspection traces. This abstraction alleviates the control–supervision mismatch by narrowing what the planner must represent and optimize.

At each step t, the planner conditions on the latest observation o_{t-1} and the accumulated history h_{t-1}, and outputs a *Thought–Action* pair:

$$z_t, a_t \sim \pi_{\text{plan}}(\cdot \mid h_{t-1}, o_{t-1}). \tag{1}$$

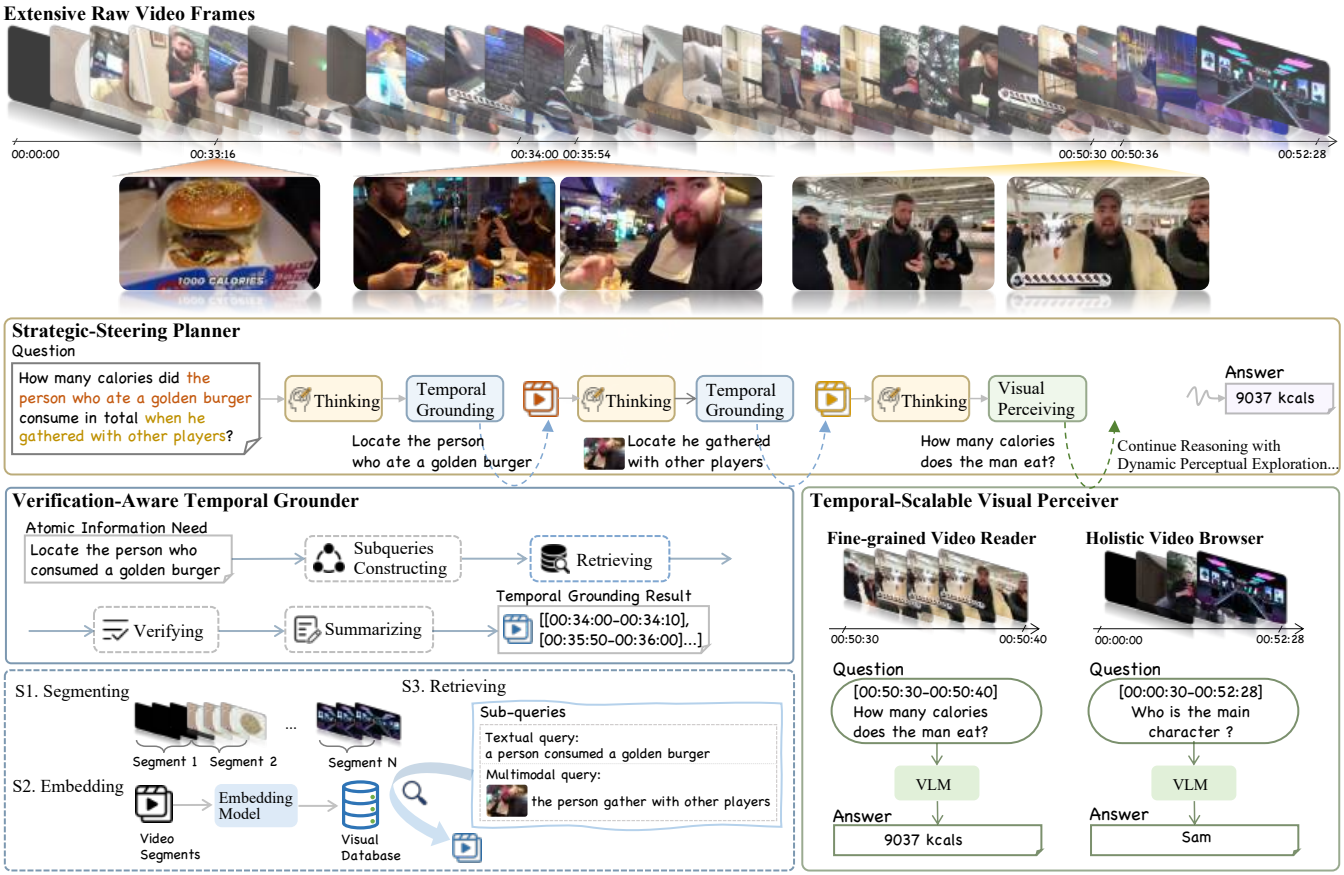

Figure 1: Illustration of VideoExplorer. Rather than forcing a monolithic agent to grapple with the high-branching complexity of simultaneous planning and perception, VideoExplorer employs hierarchical orchestration to explicitly decouple high-level orchestration from low-level execution: A central Planner directs the reasoning strategy and evidence integration, while specialized sub-agents execute specific perception tasks: a Temporal Grounder retrieves and validates relevant time windows, and a Visual Perceiver extracts fine-grained or global visual evidence conditioned on the input time windows. This structural separation narrows the decision surface and isolates optimization variance, enabling TDPO to effectively stabilize long-horizon planning and align the entire reasoning trajectory with task success.

Here, z_t is the rationale that specifies the missing information and its relevance, and a_t is a structured command that delegates execution to downstream sub-agents.

The planner defines the policy $\pi_\theta(a_t \mid h_{t-1})$ and performs four strategic functions:

- **Strategic State Decomposition:** From h_{t-1}, it identifies the gap between the current belief state and the target solution, and decomposes the query q into sequential sub-goals that maximize expected information gain—explicitly deciding *what* is missing (e.g., temporal localization vs. visual attributes).
- **Hierarchical Action Dispatching:** It maps each sub-goal to an action a_t in the hierarchical toolset and selects perception granularity by routing to the appropriate specialist (e.g., invoking the **Temporal Grounder** for span proposals or the **Visual Perceiver** for attribute verification), turning pixel-level processing into semantic API calls.
- **Belief State Update & Evidence Integration:** Given o_t, it updates its working memory h_t in a Bayesian-like manner, integrating new evidence with prior findings, filtering noise, and resolving conflicts across retrieved spans to maintain a coherent reasoning chain.
- **Confidence-Aware Termination with Reward Maximization:** It assesses whether h_t sufficiently answers q and triggers final answer generation only when confidence is adequate, balancing the cost of further exploration against the risk of incomplete grounding (premature commitment).

Critically, h_{t-1} contains only *high-level, inspectable interface signals*: prior (z_i, a_i), the Temporal Grounder's verified span outputs (e.g., timestamps, short evidence summaries, verification flags), and the Visual Perceiver's compact observations for those spans. It excludes lengthy retrieval traces, dense frame tokens, and low-level executor trajectories. Thus, learning focuses on the *reasoning chain* and discrete interface-level control decisions rather than

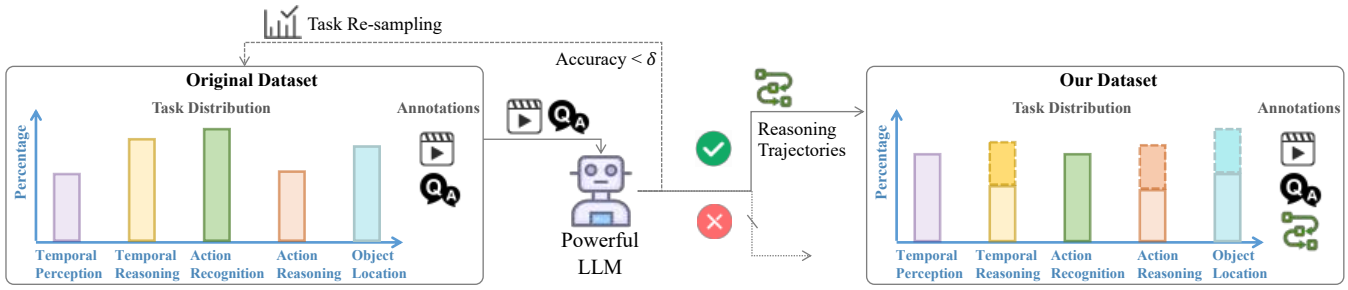

Figure 2: Illustration of difficulty-adaptive trajectory synthesis.

execution artifacts that can blur supervision. This shortens the effective horizon, reduces gradient variance, and improves training stability and scalability as video length and tool branching increase.

3.3.2 Executor I: Verification-Aware Temporal Grounder. A central failure mode in agentic LVU is *temporal drift*: as the agent iteratively retrieves and reasons, irrelevant segments accumulate in the context, polluting subsequent decisions and encouraging premature commitment. This is exacerbated by standard top-k similarity retrieval, which is optimized for recall but often returns semantically adjacent yet *decision-irrelevant* windows. Such noise directly conflicts with our motivation: when supervision is outcome-only, the planner receives no clear signal about which retrieved evidence was trustworthy, making long-horizon control variance-dominated.

To address this, we design a **Verify-then-Return** Temporal Grounder π_{ground} that turns retrieval into a *verification-aware* executor. Given a sub-query q_{sub} from the Planner, the Temporal Grounder performs a two-stage procedure:

(1) Offline Indexing. We partition each video into fixed, non-overlapping temporal windows that cover the full timeline without gaps. For each window, we compute an embedding using a visual embedding model and build an approximate nearest-neighbor index over all window representations.

(2) Retrieval and Verification. Given an atomic information need, the Temporal Grounder executes a specialized retrieval routine: it constructs retrieval-oriented subqueries, embeds them, and fetches candidate segments from the visual database. Crucially, it then calling VLMs to verify each candidate before returning it—discarding segments that were mistakenly retrieved by the embedder and keeping only those that satisfy the sub-query constraints. The Temporal Grounder finally aggregates the verified results into a compact evidence package, returning (i) timestamp intervals of relevant segments and (ii) short, inspectable summaries with lightweight verification signals to the Planner.

The Temporal Grounder is a specialized sub-agent that absorbs the long retrieval context and noisy intermediate traces, then exposes only verified, compact outputs through a clean interface. This reduces context drift in the planner's history, keeps the main reasoning chain focused on high-level control decisions, and makes responsibility boundaries sharper (retrieval/verification vs. planning/integration). In addition, the Temporal Grounder can run in parallel with other executors, enabling efficient exploration without inflating the planner's horizon or context length.

3.3.3 Executor II: Temporal-scalable Visual Perceiver. The Visual Perceiver π_{perc} is a lightweight executor that performs *on-demand* visual question answering over a small set of frames. It is intentionally *not* a controller: it does not decide where to search or how much to sample. Instead, it executes the Planner's decision, which keeps low-level visual computation out of the Planner's optimization path and helps mitigate the control–supervision mismatch in long-horizon LVU.

When Planner issues a $\texttt{Watch}(t_{\text{start}}, t_{\text{end}}, q_{\text{focus}}, S_{\text{type}})$ command, it specifies both the temporal interval and the sampling mode S_{type}:

- **Sparse Scanning (S_{global}):** uniformly sample up to a fixed frame budget across $[t_{\text{start}}, t_{\text{end}}]$ for holistic subgoals.
- **Dense Inspection (S_{local}):** densely sample within a short window (e.g., [00:50:30–00:50:40]) for fine-grained subgoals that hinge on micro-evidence.

Given the sampled frames and the focused query q_{focus}, the Visual Perceiver feeds them into a VLM and returns a *compact, inspectable* answer to the Planner. By restricting the Visual Perceiver to execution-only behavior and returning only summarized evidence (rather than dense frame traces), the system preserves a clean interface for training: the Planner is optimized on reasoning chains and interface-level decisions, while visual execution remains modular and stable as video length and tool branching grow.

3.4 Difficulty-Adaptive Trajectory Synthesis

Training hierarchical agents requires supervision on the process, not just the outcome. In long-horizon LVU, outcome-only labels make orchestration underdetermined: multiple control strategies can reach the same answer, while failures provide little signal about which decision to correct. Since trajectory-level supervision is not available in existing LVU datasets, we introduce a scalable synthesis pipeline that produces *high-quality reasoning trajectories* aligned with our hierarchical architecture.

Step 1: Oracle Trace Generation. We build on VideoMarathon [11], which contains long videos up to 1 hour with associated QA pairs. We select QA instances whose relevant video span exceeds 3 minutes as our long-horizon split. Since only final-answer labels are provided, we prompt a strong teacher model [23] to act as the Planner and Temporal Grounder, producing step-by-step expert trajectories τ_{oracle} and tuples $(q, y_{gt}, \tau_{\text{oracle}})$. Each τ_{oracle} includes subgoal decomposition, span proposals, verification outcomes, and

evidence summaries. We apply a strict correctness filter, keeping only trajectories whose final answer matches y_{gt}.

Step 2: Difficulty Estimation. To avoid shortcut instances that don't stress long-horizon control, we estimate difficulty from the success rate in Step 1. For each instance i, we define $D_i = 1 - \mathrm{Acc}_i$, where Acc_i is the teacher's answer accuracy under the same synthesis protocol; higher D_i indicates harder temporal reasoning and greater susceptibility to control errors.

Step 3: Adaptive Resampling. We form $\mathcal{D}_{\text{train}}$ by *upweighting task categories* with high difficulty. Rather than generating multiple trajectories per instance, we resample new QA instances from the difficult categories determined by Step 2, and repeat Step 1 to obtain their oracle traces. more challenging dataset. Only trajectories with correct answers are retained to ensure validity. In total, we curate 11.1k planner trajectories and 10.8k temporal grounder trajectories. This difficulty-aware resampling shifts training toward regimes where long-horizon control is most error-prone (sustained search, precise grounding, verification), allocating supervision to behaviors most critical for stable hierarchical orchestration under weak, outcome-only labels.

3.5 Trajectory Direct Preference Optimization

We first warm up the planner and the temporal grounder with Supervised Fine-Tuning (SFT) on $\mathcal{D}_{\text{train}}$, using synthesized expert trajectories as demonstrations. SFT yields a strong initialization for step-wise decisions, yet it remains insufficient for long-horizon LVU: trained only on expert traces, the planner is rarely exposed to its own deviations, and therefore lacks an effective mechanism to correct typical control failures such as over-exploration, under-exploration, and premature commitment. To more directly shape long-horizon behavior, we further optimize the planner with **Trajectory-level Direct Preference Optimization (TDPO)**, which aligns multi-step planning and evidence routing with end-task success at the trajectory level.

Preference Pair Construction. Since low-level tool execution is long and heterogeneous, jointly optimizing all components can introduce substantial variance and confound credit assignment. We therefore freeze the pretrained specialist sub-agents and apply TDPO only to the high-level planner to focus updates on strategic decisions while keeping execution behavior stable. The optimization unit is a complete planner interaction trajectory τ. Starting from the SFT-initialized planner, we sample multiple trajectories from the current policy π_θ for each input. We then construct a preference pair (τ_w, τ_l), where τ_w is the synthesized expert trajectory with verified evidence, and τ_l is a policy-sampled trajectory. We primarily label failures by outcome: trajectories that end with an incorrect final answer are treated as negative. This design concentrates learning signals on high-level decisions (planning, evidence routing, and termination), while avoiding optimization variance from low-level execution.

The TDPO Objective. The standard DPO optimizes preferences over a single response. We lift it to the trajectory-level by scoring the *decision path*:

$$\mathcal{L}_{\text{TDPO}}(\pi_\theta; \pi_{\text{ref}}) = -\mathbb{E}_{\mathcal{D}}\left[\log \sigma\left(\beta \log \frac{\pi_\theta(\tau_w)}{\pi_{\text{ref}}(\tau_w)} - \beta \log \frac{\pi_\theta(\tau_l)}{\pi_{\text{ref}}(\tau_l)}\right)\right],$$

where $\pi(\tau) = \prod_{(a_t, h_t)\in\tau} \pi(a_t \mid h_t)$ and $(\tau_w, \tau_l) \sim \mathcal{D}$.

In practice, we compute $\pi(\tau)$ over the *planner-side tokens only*: all tool outputs and executor returns are masked out, and the trajectory likelihood is evaluated on the remaining tokens that correspond to the planner's reasoning and action-generation (i.e., the high-level reasoning chain and structured commands). This focuses TDPO on optimizing hierarchical control decisions while avoiding backpropagation through low-level execution traces, improving stability and scalability as tool traces grow long and heterogeneous.

Table 1: Main results on LVU benchmarks. For fair framework comparison, all baselines share the same visual encoder (Qwen2.5-VL-7B/32B). We report fine-tuned LLM results when available; otherwise, zero-shot Qwen2.5-32B.

Model	LLM	VLM	LVBench	MLVU	MH-NIAH	Avg
Proprietary VLMs						
GPT-4o	-	-	48.9	54.9	-	-
Gemini-1.5-pro	-	-	33.1	53.8	-	-
Reasoning VLMs						
Video-R1	-	7B	38.3	46.2	42.0	42.2
Long VLMs						
LongVA	-	7B	33.8	41.1	40.7	38.5
LongVila	-	7B	41.2	<u>49.0</u>	<u>43.1</u>	<u>44.4</u>
Video-XL	-	7B	39.9	45.5	42.3	42.6
Agentic Frameworks (Qwen2.5VL-7B)						
Vanilla	-	7B	34.5	36.1	39.0	36.5
VideoAgent	32B	7B	34.9	38.6	43.0	38.8
VideoTree	32B	7B	30.3	38.9	41.1	36.8
VideoRAG	-	7B	36.2	37.8	39.4	37.8
Ego-R1	3B	7B	39.6	40.9	42.3	41.0
LVAgent	-	7B	<u>47.7</u>	40.7	41.6	43.3
VideoExplorer	7B	7B	**50.6**	**55.4**	**49.1**	**51.7**
Agentic Frameworks (Qwen2.5VL-32B)						
Vanilla	-	32B	34.9	41.8	39.5	38.7
VideoAgent	32B	32B	35.8	42.7	<u>43.7</u>	40.7
VideoTree	32B	32B	32.4	39.7	41.4	37.8
VideoRAG	-	32B	37.1	42.7	39.8	39.9
Ego-R1	3B	32B	<u>41.6</u>	<u>43.0</u>	42.6	<u>42.4</u>
VideoExplorer	7B	32B	**51.4**	**58.6**	**53.4**	**54.5**

4 Experiments

4.1 Experimental Settings

Baselines. We compare VideoExplorer with (i) state-of-the-art VLMs and (ii) agentic retrieval-augmented frameworks. For VLM baselines, we consider proprietary models **GPT-4o** [16] and **Gemini-1.5-pro** [24], the reasoning VLM **Video-R1** [5], and long-video VLMs **LongVila** [4] and **Video-XL** [21]. For agentic frameworks, we compare: (1) **Vanilla**, which feeds frames sampled at 1 fps to a VLM for direct answering; (2) **VideoAgent** [28], which iteratively identifies informative segments using pre-computed captions; (3) **VideoTree** [29], which builds a query-adaptive hierarchical video

Table 2: Ablation study on the effectiveness of model framework, training strategy, and dataset construction strategy.

Settings	MLVU	MH-NIAH
VideoExplorer	**55.4**	**49.1**
w/o Hierarchy	53.0	47.3
w/o TDPO	52.1	47.0
w/o Difficulty Sampling	53.3	47.6

Table 3: Ablation study of retrieval model.

Retriever	MLVU	MH-NIAH
CLIP [19]	54.1	48.7
LanguageBind [37]	**55.4**	**49.1**

representation for LLM reasoning; (4) **VideoRAG** [14], which leverages visually aligned auxiliary text to enhance cross-modal alignment and provide additional information beyond visual content; (5) **LVAgent** [3], which employs multiple agents to vote at each step; and (6) **Ego-R1** [25], an RL-trained agent that navigates a hierarchical video index to execute a structured chain-of-thought-for-tool process. We evaluate all models under two visual-module settings: Qwen2.5VL-7B/32B, using official weights when available. For baselines without public weights, we substitute Qwen2.5-32B and evaluate in a zero-shot setting. Unless otherwise specified, all analyses use the 7B versions of both the LLM and visual module. Full implementation details are provided in Appendix B.

Dataset. We evaluate our method on two categories of benchmarks: (1) *Long Video Understanding* on LVBench [27] and MLVU (test set) [35], (2) *Multi-hop Reasoning* on MH-NIAH [8] (max frames is set to 10,000). We report accuracy for all benchmarks. Besides, we assessed temporal grounding effectiveness using the IoU@0.5 metric, which compares the framework-grounded temporal segments to the golden temporal intervals provided by LVBench.

4.2 Main Experiments

We present the main experimental results in Table 1, from which we have several key findings:

(1) Clear wins over prior agentic frameworks. In Table 1, VideoExplorer achieves the best results across LVBench, MLVU, and MH-NIAH under both 7B and 32B backbones. Compared with representative agentic baselines (VideoAgent/VideoTree/VideoRAG/LVAgent), it delivers consistent and sizable gains, indicating that agentic formulations alone do not sufficiently address LVU bottlenecks without principled control and optimization mechanisms.

(2) Structured control matters for high-branching LVU. Baselines with flatter or loosely coordinated agent designs plateau, especially on LVBench and MLVU where multi-granularity decisions and frequent evidence routing are required. VideoExplorer's hierarchical orchestration (planner plus specialized sub-agents) better manages branching and reduces error propagation, which aligns with the introduction's motivation on ineffective control.

(3) Advantages over long-context VLMs at extreme video lengths. Long VLMs such as LongVA, LongVila are designed to process very long inputs, yet their effectiveness degrades as the number of frames keeps increasing. For example, models that can nominally handle on the order of 1,000 frames may fail to scale to the 10,000-frame regime in MH-NIAH. In contrast, VideoExplorer performs active exploration and only reads frames that are relevant to the task, so it is not constrained by a fixed context length budget and can naturally scale to substantially longer videos.

(4) Better scaling with stronger backbones. When upgrading visual module from 7B to 32B, VideoExplorer improves more than most baselines. This suggests that performance is less constrained by coordination and control issues, allowing stronger visual modules to translate into larger improvements.

Table 4: Ablation study on candidate clip interval.

Clip Interval (s)	MLVU	LVBench
5	55.2	50.4
10	**55.4**	**50.6**
15	55.1	50.3
20	55.0	50.0

Table 5: Evaluating temporal grounding accuracy and question answering accuracy on the LVBench dataset.

Model	VLM	IoU@0.5	Accuracy
VideoAgent	7B	4.2	35.9
VideoRAG	7B	3.1	36.2
Ego-R1	7B	4.9	39.6
VideoExplorer	7B	**12.8**	**50.6**

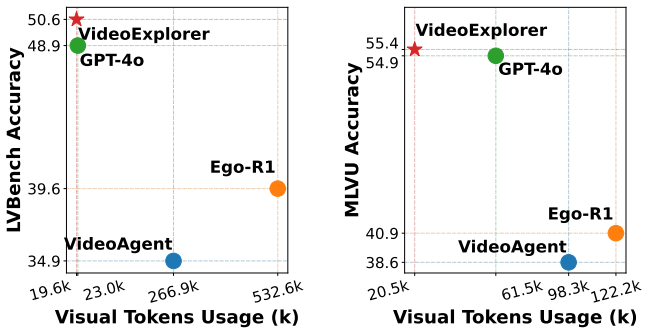

Figure 3: Visual token usage comparison. VideoExplorer uses fewer visual tokens while improving accuracy, suggesting less over-exploration during clip inspection.

4.3 Ablation Study

We evaluate the impact of VideoExplorer's components through ablation experiments (Table 2, Table 3 and Table 4). (1) **Training Strategies:** Removing TDPO causes a notable drop of 3.3% on MLVU and 2.1% on MH-NIAH, showing that it is crucial for stable optimization. By enlarging the reward gap between preferred and rejected outputs based on environment feedback, TDPO encourages the planner to avoid shallow reasoning and instead pursue adaptive, multi-step exploration, ultimately improving the reliability of generated trajectories. (2) **Framework Design:** Decoupling

Query: Which company does the presenter mentions before NetApp? A. Snowflake B. Servicenow C. Cohesity D. Dell Technologies
Ground-truth target: A. Snowflake **Gemini 1.5 Pro**: C. Cohesity **VideoRAG**: D. Dell Technologies **VideoAgent:** D. Dell Technologies

<think>The user is asking to identify a company mentioned by the presenter before "NetApp". I need to find the specific segment where "NetApp" is mentioned and then check the preceding company mentioned. I will use the `temporal_grounding_agent` to locate the "NetApp" mention first, and then investigate the preceding segment using `video_reader` to identify the company.</think>
<temporal_grounding_agent>The question asks about the company mentioned by the presenter before NetApp. I need to find the segment where "NetApp" is mentioned.</temporal_grounding_agent> [Pause]"

There are 1 related segments in the video: [(5210, 5230, "The company NetApp is mentioned and its logo is displayed alongside NVIDIA's, with a slide titled 'NetApp Unlocks Exabytes of Data for Secure, Private Gen AI' and another titled 'Dell AI Factory with NVIDIA'. ")]

<think>The `temporal_grounding_agent` successfully identified the segment where "NetApp" is mentioned (5210-5230). Now I need to investigate the video content just before this segment to find the company mentioned then. I will use `video_reader` to check the segment leading up to 5210, specifically from 5200 to 5210. I will ask which company is mentioned or shown in that segment with the given options.</think> <video_reader>5200:5210</video_reader><video_reader_question>Which company is mentioned or shown in this segment? A. Snowflake B. Servicenow C. Cohesity D. Dell Technologies</video_reader_question>

The tool result for <video_reader>5200:5210</video_reader> is The description of the video is: The video features a speaker on a stage with a dark background. The logos of "Snowflake" and "NVIDIA" are prominently displayed on the screen behind the speaker. The speaker is dressed in a casual outfit and appears to be presenting. The date "MARCH 18, 2024" is shown in the top right corner, and the "y! highlights" logo is visible in the bottom right corner. The speaker is gesturing with their hands, suggesting they are explaining or discussing a topic related to the displayed logos. The answer is: A. Snowflake

<think>The `video_reader` for the segment 5200-5210 clearly states that "Snowflake" logo is displayed alongside "NVIDIA" and the speaker is presenting. This means "Snowflake" is the company mentioned before NetApp. This directly answers the question.</think>
<answer>A</answer> ← **VideoExplorer**

Figure 4: Case study of VideoExplorer on LVBench. "5210:5230" denotes the time span in seconds. The planner adopts an anchor-and-verify strategy: it delegates temporal grounding to the temporal grounder to localize the NetApp mention, then queries the visual perceiver on the immediately preceding clip for fine-grained recognition, yielding the correct answer. In contrast, baselines fail to reliably support this multi-hop temporal tracing, leading to missed evidence or premature commitment.

temporal grounding from reasoning reduces the planner's burden and prevents premature answering. Without the grounding agent, performance decreases by 2.4% and 1.8%, as the planner struggles to gather sufficient evidence before producing an answer. This highlights that explicit grounding enables the planner to focus on higher-level reasoning while relying on the temporal grounder to supply relevant temporal spans. (3) **Dataset Effectiveness:** Uniform sampling causes 2.1/1.5-point drops, highlighting the benefit of difficulty-adaptive sampling for improving generalization. Overall, the ablations show that training strategy, framework design, dataset construction, and scaling each contribute to performance, and their combination is critical to VideoExplorer's effectiveness. (4) **Candidate Clip Intervals:** We examine the effect of candidate clip interval (Table 4). A 10s interval performs best on both LVBench and MLVU, balancing temporal coverage and semantic completeness. Shorter clips (5s) overly fragment content and increase redundancy, while longer clips (15–20s) blur event boundaries and weaken temporal precision, leading to degraded performance. (5) **Retrieval Models:** We compare retrievers in Table 2. Replacing CLIP with LanguageBind yields consistent gains on MLVU and MH-NIAH, suggesting that better video retrieval quality directly benefits downstream temporal grounding and reasoning, improving overall performance.

4.4 Temporal Grounding Analysis

Table 5 shows that VideoExplorer markedly improves both temporal grounding and QA on LVBench, reaching 12.8 IoU@0.5 (vs. 4.9 for Ego-R1 and 4.2 for VideoAgent) and 50.6 accuracy. This gain comes from the hierarchical design with a verification-aware temporal grounder: the planner formulates clear grounding requests, and the specialized sub-agent performs grounding on concise, task-relevant cues rather than within long, weakly related reasoning context, yielding more precise temporal localization. The concurrent rise in IoU and QA accuracy highlights the importance of accurate modular temporal reasoning for LVU.

4.5 Token Efficiency

We compare total visual token usage of different methods in Figure 3. Unlike methods such as VideoAgent and Ego-R1, which indiscriminately process all video segments and thus incur heavy token consumption, VideoExplorer performs dynamic reasoning by selectively focusing only on on-demand task-relevant segments. Despite

using far fewer visual tokens, VideoExplorer achieves significantly better performance. This demonstrates its ability to progressively reason and precisely localize meaningful video content, leading to both higher efficiency and stronger effectiveness.

4.6 Case Study

We illustrate VideoExplorer's reasoning on an LVBench example in Figure 4. The question, "Which company does the presenter mention before NetApp?", requires multi-hop temporal reasoning by first grounding the NetApp mention and then verifying the immediately preceding segment. Gemini-1.5-pro fails under aggressive downsampling, while one-pass RAG-style retrieval often misses the required evidence because it cannot reliably support backward tracing. VideoAgent exhibits premature termination and stops without ever retrieving the correct clip. In contrast, VideoExplorer's planner follows an anchor-and-verify strategy that reduces redundant exploration and avoids early stopping. The system enforces a clear division of labor: the temporal grounder first retrieves and localizes the NetApp occurrence (5210:5230), the planner then routes exploration to the preceding moment, and the visual perceiver identifies the mentioned company to produce the correct answer. This workflow yields a concise and inspectable reasoning chain. Additional cases and failure analyses are provided in Appendix C.

5 Conclusion

In this work, we presented VideoExplorer, a hierarchical agentic system for long-video understanding that tackles two core bottlenecks in existing frameworks: weak controllability in high-branching, multi-granularity reasoning and limited learning signals from sparse outcome-only supervision. By combining hierarchical structured control, in which a planning agent coordinates specialized sub-agents, with trajectory-level optimization via TDPO, VideoExplorer better aligns long-horizon planning, evidence routing, and termination with task success, reducing premature commitment and cascading errors. Experiments show that VideoExplorer consistently improves both accuracy and efficiency on MLVU, LVBench, and MH-NIAH, underscoring the value of structured orchestration and trajectory-aware learning for reliable long video reasoning.

6 Acknowledgments

This work was supported by National Natural Science Foundation of China No. 62502049 and partially supported by Beijing Postdoctoral Research Foundation.

References

[1] Dawit Mureja Argaw, Joon-Young Lee, Markus Woodson, In So Kweon, and Fabian Caba Heilbron. 2023. Long-range Multimodal Pretraining for Movie Understanding. In *IEEE/CVF International Conference on Computer Vision, ICCV 2023, Paris, France, October 1-6, 2023*. IEEE, 13346–13357. doi:10.1109/ICCV51070.2023.01232

[2] Shuai Bai, Keqin Chen, Xuejing 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. 2025. Qwen2.5-VL Technical Report. arXiv:2502.13923 [cs.CV] https://arxiv.org/abs/2502.13923

[3] Boyu Chen, Zhengrong Yue, Siran Chen, Zikang Wang, Yang Liu, Peng Li, and Yali Wang. 2025. LVAgent: Long Video Understanding by Multi-Round Dynamical Collaboration of MLLM Agents. In *IEEE/CVF International Conference on Computer Vision, ICCV 2025, Honolulu, HI, USA, October 19-25, 2025*. IEEE, 20237–20246. doi:10.1109/ICCV51701.2025.01882

[4] Yukang Chen, Fuzhao Xue, Dacheng Li, Qinghao Hu, Ligeng Zhu, Xiuyu Li, Yunhao Fang, Haotian Tang, Shang Yang, Zhijian Liu, Yihui He, Hongxu Yin, Pavlo Molchanov, Jan Kautz, Linxi Fan, Yuke Zhu, Yao Lu, and Song Han. 2025. LongVILA: Scaling Long-Context Visual Language Models for Long Videos. In *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*. OpenReview.net. https://openreview.net/forum?id=wCXAlfvCy6

[5] Kaituo Feng, Kaixiong Gong, Bohao Li, Zonghao Guo, Yibing Wang, Tianshuo Peng, Junfei Wu, Xiaoying Zhang, Benyou Wang, and Xiangyu Yue. 2025. Video-R1: Reinforcing Video Reasoning in MLLMs. arXiv:2503.21776 [cs.CV] https://arxiv.org/abs/2503.21776

[6] Noriyuki Kugo, Xiang Li, Zixin Li, Ashish Gupta, Arpandeep Khatua, Nidhish Jain, Chaitanya Patel, Yuta Kyuragi, Yasunori Ishii, Masamoto Tanabiki, Kazuki Kozuka, and Ehsan Adeli. 2025. VideoMultiAgents: A Multi-Agent Framework for Video Question Answering. arXiv:2504.20091 [cs.CV] https://arxiv.org/abs/2504.20091

[7] Rui Li, Xiaohan Wang, Yuhui Zhang, Orr Zohar, Zeyu Wang, and Serena Yeung-Levy. 2025. Temporal Preference Optimization for Long-Form Video Understanding. arXiv:2501.13919 [cs.CV] https://arxiv.org/abs/2501.13919

[8] Xinhao Li, Yi Wang, Jiashuo Yu, Xiangyu Zeng, Yuhan Zhu, Haian Huang, Jianfei Gao, Kunchang Li, Yinan He, Chenting Wang, Yu Qiao, Yali Wang, and Limin Wang. 2025. VideoChat-Flash: Hierarchical Compression for Long-Context Video Modeling. arXiv:2501.00574 [cs.CV] https://arxiv.org/abs/2501.00574

[9] Ruotong Liao, Max Erler, Huiyu Wang, Guangyao Zhai, Gengyuan Zhang, Yunpu Ma, and Volker Tresp. 2024. VideoINSTA: Zero-shot Long Video Understanding via Informative Spatial-Temporal Reasoning with LLMs. arXiv:2409.20365 [cs.CV] https://arxiv.org/abs/2409.20365

[10] Bin Lin, Yang Ye, Bin Zhu, Jiaxi Cui, Munan Ning, Peng Jin, and Li Yuan. 2024. Video-LLaVA: Learning United Visual Representation by Alignment Before Projection. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing, EMNLP 2024, Miami, FL, USA, November 12-16, 2024*, Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen (Eds.). Association for Computational Linguistics, 5971–5984. doi:10.18653/V1/2024.EMNLP-MAIN.342

[11] Jingyang Lin, Jialian Wu, Ximeng Sun, Ze Wang, Jiang Liu, Yusheng Su, Xiaodong Yu, Hao Chen, Jiebo Luo, Zicheng Liu, and Emad Barsoum. 2025. Unleashing Hour-Scale Video Training for Long Video-Language Understanding. arXiv:2506.05332 [cs.CV] https://arxiv.org/abs/2506.05332

[12] Haotian Liu, Chunyuan Li, Yuheng Li, Bo Li, Yuanhan Zhang, Sheng Shen, and Yong Jae Lee. 2024. LLaVA-NeXT: Improved reasoning, OCR, and world knowledge. https://llava-vl.github.io/blog/2024-01-30-llava-next/

[13] Zhijian Liu, Ligeng Zhu, Baifeng Shi, Zhuoyang Zhang, Yuming Lou, Shang Yang, Haocheng Xi, Shiyi Cao, Yuxian Gu, Dacheng Li, et al. 2025. Nvila: Efficient frontier visual language models. In *Proceedings of the Computer Vision and Pattern Recognition Conference*. 4122–4134.

[14] Yongdong Luo, Xiawu Zheng, Xiao Yang, Guilin Li, Haojia Lin, Jinfa Huang, Jiayi Ji, Fei Chao, Jiebo Luo, and Rongrong Ji. 2024. Video-RAG: Visually-aligned Retrieval-Augmented Long Video Comprehension. arXiv:2411.13093 [cs.CV] https://arxiv.org/abs/2411.13093

[15] Ziyu Ma, Chenhui Gou, Hengcan Shi, Bin Sun, Shutao Li, Hamid Rezatofighi, and Jianfei Cai. 2025. DrVideo: Document Retrieval Based Long Video Understanding. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2025, Nashville, TN, USA, June 11-15, 2025*. Computer Vision Foundation / IEEE, 18936–18946. doi:10.1109/CVPR52734.2025.01764

[16] OpenAI. 2024. GPT-4o. https://openai.com/index/hello-gpt-4o/.

[17] Guansong Pang, Chunhua Shen, Longbing Cao, and Anton Van Den Hengel. 2021. Deep Learning for Anomaly Detection: A Review. *Comput. Surveys* 54, 2 (March 2021), 1–38. doi:10.1145/3439950

[18] Minghao Qin, Xiangrui Liu, Zhengyang Liang, Yan Shu, Huaying Yuan, Juenjie Zhou, Shitao Xiao, Bo Zhao, and Zheng Liu. 2025. Video-XL-2: Towards Very Long-Video Understanding Through Task-Aware KV Sparsification. arXiv:2506.19225 [cs.CV] https://arxiv.org/abs/2506.19225

[19] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. 2021. Learning transferable visual models from natural language supervision. In *International conference on machine learning*. PMLR, 8748–8763.

[20] Chuyi Shang, Amos You, Sanjay Subramanian, Trevor Darrell, and Roei Herzig. 2024. TraveLER: A Modular Multi-LMM Agent Framework for Video Question-Answering. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing, EMNLP 2024, Miami, FL, USA, November 12-16, 2024*, Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen (Eds.). Association for Computational Linguistics, 9740–9766. doi:10.18653/V1/2024.EMNLP-MAIN.544

[21] Yan Shu, Zheng Liu, Peitian Zhang, Minghao Qin, Junjie Zhou, Zhengyang Liang, Tiejun Huang, and Bo Zhao. 2025. Video-XL: Extra-Long Vision Language Model for Hour-Scale Video Understanding. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2025, Nashville, TN, USA, June 11-15, 2025*. Computer Vision Foundation / IEEE, 26160–26169. doi:10.1109/CVPR52734.2025.02436

[22] Enxin Song, Wenhao Chai, Guanhong Wang, Yucheng Zhang, Haoyang Zhou, Feiyang Wu, Haozhe Chi, Xun Guo, Tian Ye, Yanting Zhang, Yan Lu, Jenq-Neng Hwang, and Gaoang Wang. 2024. MovieChat: From Dense Token to Sparse Memory for Long Video Understanding. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2024, Seattle, WA, USA, June 16-22, 2024*. IEEE, 18221–18232. doi:10.1109/CVPR52733.2024.01725

[23] DeepSeek-AI Team. 2025. DeepSeek-R1: Incentivizing Reasoning Capability in LLMs via Reinforcement Learning. arXiv:2501.12948 [cs.CL] https://arxiv.org/abs/2501.12948

[24] Gemini Team and etc. 2024. Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context. arXiv:2403.05530 [cs.CL] https://arxiv.org/abs/2403.05530

[25] Shulin Tian, Ruiqi Wang, Hongming Guo, Penghao Wu, Yuhao Dong, Xiuying Wang, Jingkang Yang, Hao Zhang, Hongyuan Zhu, and Ziwei Liu. 2025. Ego-R1: Chain-of-Tool-Thought for Ultra-Long Egocentric Video Reasoning. arXiv:2506.13654 [cs.CV] https://arxiv.org/abs/2506.13654

[26] Peng Wang, Shuai Bai, Sinan Tan, Shijie Wang, Zhihao Fan, Jinze Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, et al. 2024. Qwen2-vl: Enhancing vision-language model's perception of the world at any resolution. *arXiv preprint arXiv:2409.12191* (2024).

[27] Weihan Wang, Zehai He, Wenyi Hong, Yean Cheng, Xiaohan Zhang, Ji Qi, Ming Ding, Xiaotao Gu, Shiyu Huang, Bin Xu, Yuxiao Dong, and Jie Tang. 2025. LVBench: An Extreme Long Video Understanding Benchmark. In *IEEE/CVF International Conference on Computer Vision, ICCV 2025, Honolulu, HI, USA, October 19-25, 2025*. IEEE, 22958–22967. doi:10.1109/ICCV51701.2025.02131

[28] Xiaohan Wang, Yuhui Zhang, Orr Zohar, and Serena Yeung-Levy. 2024. VideoAgent: Long-Form Video Understanding with Large Language Model as Agent. In *Computer Vision – ECCV 2024: 18th European Conference, Milan, Italy, September 29–October 4, 2024, Proceedings, Part LXXX* (Milan, Italy). Springer-Verlag, Berlin, Heidelberg, 58–76. doi:10.1007/978-3-031-72989-8_4

[29] Ziyang Wang, Shoubin Yu, Elias Stengel-Eskin, Jaehong Yoon, Feng Cheng, Gedas Bertasius, and Mohit Bansal. 2025. VideoTree: Adaptive Tree-based Video Representation for LLM Reasoning on Long Videos. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2025, Nashville, TN, USA, June 11-15, 2025*. Computer Vision Foundation / IEEE, 3272–3283. doi:10.1109/CVPR52734.2025.00311

[30] Kangda Wei, Zhengyu Zhou, Bingqing Wang, Jun Araki, Lukas Lange, Ruihong Huang, and Zhe Feng. 2025. PreMind: Multi-Agent Video Understanding for Advanced Indexing of Presentation-style Videos. arXiv:2503.00162 [cs.CV] https://arxiv.org/abs/2503.00162

[31] Zhiyuan Xu, Kun Wu, Junjie Wen, Jinming Li, Ning Liu, Zhengping Che, and Jian Tang. 2024. A survey on robotics with foundation models: toward embodied ai. *arXiv preprint arXiv:2402.02385* (2024).

[32] Huaxin Zhang, Xiaohao Xu, Xiang Wang, Jialong Zuo, Xiaonan Huang, Changxin Gao, Shanjun Zhang, Li Yu, and Nong Sang. 2025. Holmes-VAU: Towards Long-term Video Anomaly Understanding at Any Granularity. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2025, Nashville, TN, USA, June 11-15, 2025*. Computer Vision Foundation / IEEE, 13843–13853. doi:10.1109/CVPR52734.2025.01292

[33] Yuanhan Zhang, Jinming Wu, Wei Li, Bo Li, Zejun Ma, Ziwei Liu, and Chunyuan Li. 2025. LLaVA-Video: Video Instruction Tuning With Synthetic Data. *Trans. Mach. Learn. Res.* 2025 (2025). https://openreview.net/forum?id=EElFGvt39K

[34] Zhuo Zhi, Qiangqiang Wu, Minghe shen, Wenbo Li, Yinchuan Li, Kun Shao, and Kaiwen Zhou. 2025. VideoAgent2: Enhancing the LLM-Based Agent System for Long-Form Video Understanding by Uncertainty-Aware CoT. arXiv:2504.04471 [cs.CV] https://arxiv.org/abs/2504.04471

[35] Junjie Zhou, Yan Shu, Bo Zhao, Boya Wu, Shitao Xiao, Xi Yang, Yongping Xiong, Bo Zhang, Tiejun Huang, and Zheng Liu. 2024. MLVU: A Comprehensive Benchmark for Multi-Task Long Video Understanding. *arXiv preprint arXiv:2406.04264* (2024).

[36] Yiyang Zhou, Yangfan He, Yaofeng Su, Siwei Han, Joel Jang, Gedas Bertasius, Mohit Bansal, and Huaxiu Yao. 2025. ReAgent-V: A Reward-Driven Multi-Agent Framework for Video Understanding. arXiv:2506.01300 [cs.CV] https://arxiv.org/abs/2506.01300

[37] Bin Zhu, Bin Lin, Munan Ning, Yang Yan, Jiaxi Cui, Hongfa Wang, Yatian Pang, Wenhao Jiang, Junwu Zhang, Zongwei Li, Caiwan Zhang, Zhifeng Li, Wei Liu, and Li Yuan. 2024. LanguageBind: Extending Video-Language Pretraining to N-modality by Language-based Semantic Alignment. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net. https://openreview.net/forum?id=QmZKc7UZCy

[38] Jinguo Zhu and etc. 2025. InternVL3: Exploring Advanced Training and Test-Time Recipes for Open-Source Multimodal Models. arXiv:2504.10479 [cs.CV] https://arxiv.org/abs/2504.10479

A Use of LLMs

The entire research project, including the formulation of the core idea and all experimental work, was carried out without LLM assistance.

B Implementation Details

Our experiments are based on the open-source implementations of most baselines. As VideoAgent's code is not publicly available, we replicated the method based on its paper. To ensure the fairness and rationality of our evaluation, we implement the following controls: (1) The maximum input length for the proprietary VLMs and reasoning VLMs is set to 32 frames. The maximum input length for long VLMs is determined according to the experimental settings reported in their paper. For Vanilla and VideoRAG, each video is uniformly downsampled to 32 frames as input to the VLM; for Ego-R1 and VideoExplorer, each VLM call to the VLM does not exceed 32 frames in length. (2) For all benchmarks, all baselines rely solely on visual information to answer questions; audio is not provided to prevent unfairness due to information asymmetry.

For VideoAgent and VideoTree, raw videos are segmented into 4s video segments and dense captioned by the VLM. For the Ego-R1 baseline, we adopted the implementation from its official code repository, which builds a hierarchical index using both 10-second and 600-second segments. Both Qwen2.5-VL-7B and Qwen2.5-VL-32B are deployed using the vLLM inference framework to accelerate the inference process. All experiments are conducted on 1 node of an 880G H100 GPU.

For VideoExplorer, we initialize both planner and temporal grounder with Qwen2.5-7B []. We use LanguageBind [37] as the video clip embedder. For textual query, we use LanguageBind [37] to calculate the text embedding, and calculate similarity with the video clip embeddings. For multimodal query, we average the textual embedding and query image embedding to get the multimodal embedding, and calculate similarity with the video clip embeddings. We set top-k to 10 to balance the accuracy and efficiency. The maximum turn of the planner is 20.

C Supplementary Case Studies and Failure Analysis

Query: In what setting does the entire video take place? A. Beach B. Library C. Desert D. City E. Forest F. Campus
Ground-truth target: D **VideoAgent**: F **Gemini 1.5 Pro**: D

<think>The question asks about the overall setting of the entire video. This is a holistic question that the `video_browser` agent is best suited to answer. I will use `video_browser` to get a general idea of the video's setting, and then use `temporal_grounding_agent` and `video_reader` to confirm it if necessary.</think><video_browser>In what setting does the entire video take place? A. Beach B. Library C. Desert D. City E. Forest F. Campus</video_browser>

The tool results for <video_browser>In what setting does the entire video take place? A. Beach B. Library C. Desert D. City E. Forest F. Campus</video_browser> is:The video features scenes that include a car interior, a street with a yellow bus, and a lively street market, which are all typical of a city environment. Therefore, the correct answer is D. City.

<think>The `video_browser` agent has provided a clear answer, stating that the video's setting is a city based on the scenes of a car interior, a street with a yellow bus, and a lively street market. This seems to be a sufficient answer for the holistic question. I will proceed with this answer.</think><answer>D</answer>" ← **VideoExplorer**

Figure 5: Case Study of Holistic Task on MLVU.

Query: Which of the following options correctly matches the sequence of actions as they actually appear in the video?
A. jetskiing --> shredding paper --> paragliding --> abseiling
B. abseiling --> jetskiing --> paragliding --> shredding paper
C. paragliding --> shredding paper --> abseiling --> jetskiing
D. abseiling --> shredding paper --> jetskiing --> paragliding
E. jetskiing --> paragliding --> abseiling --> shredding paper
F. shredding paper --> jetskiing --> abseiling --> paragliding
Ground-truth target: F **VideoAgent**: F. **Gemini 1.5 Pro**: C.

<think> The user is asking to identify the correct sequence of four actions as they appear in the video. I need to find the temporal order of "jetskiing", "shredding paper", "paragliding", and "abseiling". I will use the temporal grounding agent to locate each of these actions and then determine their order.</think> <temporal_grounding_agent>Locate the segments for "jetskiing", "shredding paper", "paragliding", and "abseiling". </temporal_grounding_agent>

There are 4 related segments in the video: [(200, 220, 'Jetskiing'), (40, 50, 'Shredding paper'), (600, 610, 'Abseiling'), (630, 640, 'Paragliding')]

<think>I have received the timestamps for all four actions: - Jetskiing: 200-220 - Shredding paper: 40-50 - Abseiling: 600-610 - Paragliding: 630-640 Now I need to arrange them in chronological order based on their start times. 1. Shredding paper (40-50) 2. Jetskiing (200-220) 3. Abseiling (600-610) 4. Paragliding (630-640) Let's check the options. A. jetskiing --> shredding paper --> paragliding --> shredding paper (Incorrect) B. abseiling --> jetskiing --> paragliding --> shredding paper (Incorrect) C. paragliding --> shredding paper --> abseiling --> jetskiing (Incorrect) D. abseiling --> shredding paper --> jetskiing --> paragliding (Incorrect) E. jetskiing --> paragliding --> abseiling --> shredding paper (Incorrect) F. shredding paper --> jetskiing --> abseiling --> paragliding (Correct based on 40-50, 200-220, 600-610, 630-640) The sequence derived from the timestamps matches option F.</think> <answer>F</answer> ← **VideoExplorer**

Figure 6: Case Study of Multi-detail Task on MLVU.

Query: What did the yellow sponge use to fly into the sky? A. Wings B. Spoon C. Boat D. Pot Lid E. Bamboo Dragonfly F. Shovel
Ground-truth target: F **VideoAgent**: A **Gemini 1.5 Pro**: A

<think>The user is asking a question about what the yellow sponge used to fly. I need to first locate the segment where the yellow sponge flies into the sky. Then, I will examine that segment to determine what it used for lift-off. I will use the temporal grounding agent to find the relevant segment.</think><temporal_grounding_agent>The yellow sponge uses something to fly into the sky. What did it use? A. Wings B. Spoon C. Boat D. Pot Lid E. Bamboo Dragonfly F. Shovel</temporal_grounding_agent>

There are 4 related segments in the video: [(190, 200, 'The yellow sponge flies using a spatula and megaphone. '), (210, 220, 'The yellow sponge flies using a spatula as a propeller. '), (390, 400, 'The yellow sponge flies using a large, white, rectangular device with a red propeller at the back. '), (420, 430, 'The yellow sponge flies using a modified boat and a mechanical device attached to it. ')]

<think>The temporal grounding agent found four segments where the yellow sponge is flying and being propelled by different means. The question asks what it used to "fly into the sky", implying the initial means of ascent. I need to check each segment to confirm if it's the "initial" use. I will start by examining the earliest segment provided by the temporal grounding agent that mentions flight. The earliest relevant segment is (190, 200). I will ask a specific question about what is used for flying in that segment, including all the options to help the video reader. I will then proceed to other segments if needed.</think> <video_reader>190:200</video_reader><video_reader_question>What did the yellow sponge use to fly or ascend in this segment? A. Wings B. Spoon C. Boat D. Pot Lid E. Bamboo Dragonfly F. Shovel</video_reader_question>

The tool result for <video_reader>190:200</video_reader> is The description of the video is: In this segment from an animated show, a yellow sponge character is seen holding a spoon and a pot lid. The sponge character is standing on a wooden surface, and there are other characters in the background. The sponge character then uses the pot lid to fly or ascend into the air, while the other characters watch. The environment appears to be indoors, possibly on a ship or a boat, given the wooden planks and the presence of other characters in the background. The sponge character's expression is one of excitement and determination as it uses the pot lid to fly. The answer is: D. Pot Lid

<think>The video reader for the segment 190:200 explicitly states that "The sponge character then uses the pot lid to fly or ascend into the air". This directly answers the question "What did the yellow sponge use to fly into the sky?". Option D, "Pot Lid", matches this description. This seems to be the correct answer. I should confirm if this is the *initial* use if the question implies that. However, since the question is about "What did the yellow sponge use to fly into the sky?", and the video reader clearly states the pot lid is used for flight, this seems to be the correct answer.</think>
<answer>D</answer> ← **VideoExplorer**

Figure 7: Case study of a failure on MLVU. The planner correctly identifies the temporal interval, but the VLM fails due to semantic hallucination, leading to the wrong answer.