VideoAgent2: Enhancing the LLM-Based Agent System for Long-Form Video Understanding by Uncertainty-Aware CoT

Zhuo Zhi
University College London
zhuo.zhi.21@ucl.ac.uk

Qiangqiang Wu
City University of Hong Kong
qiangqwu2@cityu.edu.hk

Minghe Shen
University College London
minghe.shen.24@ucl.ac.uk

Wenbo Li
Huawei Noah's Ark Lab
fenglinglwb@gmail.com

Yinchuan Li
Huawei Noah's Ark Lab
yinchuan.li.cn@gmail.com

Kun Shao
Huawei Noah's Ark Lab
shaokun2@huawei.com

Kaiwen Zhou
Huawei Noah's Ark Lab
zhoukaiwen2@huawei.com

Abstract

Long video understanding has emerged as an increasingly important yet challenging task in computer vision. Agent-based approaches are gaining popularity for processing long videos, as they can handle extended sequences and integrate various tools to capture fine-grained information. However, existing methods still face several challenges: (1) they often rely solely on the reasoning ability of large language models (LLMs) without dedicated mechanisms to enhance reasoning in long video scenarios; and (2) they remain vulnerable to errors or noise from external tools. To address these issues, we propose a specialized chain-of-thought (CoT) process tailored for long video analysis. Our proposed CoT with plan-adjust mode enables the LLM to incrementally plan and adapt its information-gathering strategy. We further incorporate heuristic uncertainty estimation of both the LLM and external tools to guide the CoT process. This allows the LLM to assess the reliability of newly collected information, refine its collection strategy, and make more robust decisions when synthesizing final answers. Empirical experiments show that our uncertainty-aware CoT effectively mitigates noise from external tools, leading to more reliable outputs. We implement our approach in a system called VideoAgent2, which also includes additional modules such as general context acquisition and specialized tool design. Evaluation on three dedicated long video benchmarks (and their subsets) demonstrates that VideoAgent2 outperforms the previous state-of-the-art agent-based method, VideoAgent, by an average of 13.1% and achieves leading performance among all zero-shot approaches.

1 Introduction

Long video understanding has become increasingly important in computer vision due to the widespread presence of lengthy video content in domains such as entertainment, surveillance, and

39th Conference on Neural Information Processing Systems (NeurIPS 2025).

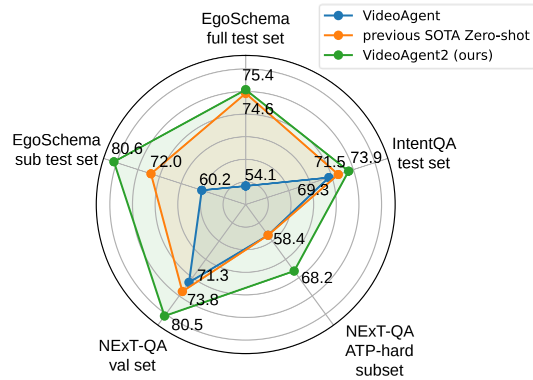

Figure 1: Performance of previous SOTA agent-based method VideoAgent [1], previous SOTA Zero-shot method [2, 3, 4, 1, 5] and our proposed VideoAgent2 on all evaluation datasets. The metric is accuracy.

autonomous driving [6]. Unlike short clips, long videos introduce challenges in modeling extended temporal dependencies while maintaining feasible computational costs.

Multimodal large language models (MLLMs) have achieved promising results on short clips [7, 8, 9, 10, 11], but still struggle with long videos. This is primarily due to (1) the computational burden of processing lengthy sequences using transformer-based architectures [12], and (2) limited granularity in spatio-temporal perception constrained by the encoder and training data. While context compression strategies have been explored [13, 14, 15], they risk losing crucial information. Furthermore, different user queries often require information at varied granularities, making generalization difficult. Efforts like fine-grained instruction tuning [16], chain-of-thought (CoT) reasoning [17], and specialized modules [10] aim to address this but face challenges in complexity and reasoning latency.

Recently, agent-based frameworks have emerged as a promising direction [18, 1, 19, 20, 21, 22]. These methods utilize LLMs to reason over content retrieved via pre-trained video/image tools, leveraging the LLM's textual reasoning capabilities while maintaining efficiency. For instance, VideoAgent [1] emulates human video viewing by first extracting coarse context, then iteratively retrieving frames based on user queries until sufficient information is gathered. Agent-based approach avoids full video processing and supports the acquisition of information at arbitrary levels of granularity by using various tools.

Despite their promise, agent-based methods face two main issues: (1) overemphasis on architectural design while underexploring ways to improve LLM reasoning in long video scenarios [19, 1], and (2) susceptibility to noise and hallucination from external tools [23].

To overcome these limitations, we propose **VideoAgent2**, a LLM-based agent system designed to improve reasoning accuracy and robustness in long video understanding. We introduce a specialized CoT process modeled on human cognition, using a plan-adjust mechanism to progressively refine retrieved information from coarse to fine detail. Specifically, we introduce a specialized CoT process for long-video agent systems that helps improve LLM's reasoning and decision-making capabilities. This process mimics human video understanding, using a plan-adjust mode to incrementally acquire information from coarse to fine-grained details. Additionally, we integrate heuristic uncertainty estimation from both the LLM and tools into the CoT pipeline. This helps filter unreliable information, guiding retrieval and decision-making without introducing extra parameters or inference overhead, outperforming previous CoT methods [24, 25] in efficiency. Finally, we propose a new pipeline for implementing VideoAgent2, incorporating important components such as general context information acquisition and specialized tool design. Our contributions are threefold:

- We design a specialized CoT process based on the plan-adjust mode to enhance LLM reasoning and decision-making in long video understanding.
- We introduce uncertainty-guided CoT reasoning, which mitigates noise and hallucination in the system while requiring no additional parameters.
- We propose the VideoAgent2 pipeline, which integrates innovative designs such as general context acquisition and specialized tools. Videoagent2 achieves superior performance on

long-form video understanding benchmarks such as Ego-Schema [26], NExT-QA [27], and IntentQA [28], as shown in Fig. 1.

The remainder of this paper is structured as follows: Section 2 reviews related work, Section 3 presents our method, Section 4 details experiments and analysis, and Section 5 concludes with limitations and future directions.

2 Related work

2.1 MLLM for long-form video understanding

Substantial progress has been made in developing multimodal large language models (MLLMs) for video understanding, focusing on two core challenges: managing the computational load of long videos and extracting fine-grained spatio-temporal information. Techniques such as LLaMA-VID and MA-LLM reduce input size by compressing video frames into compact token representations [29, 13], while methods like sliding windows [15] and token shuffling [30] further optimize efficiency. To enhance fine-grained perception, VTimeLLM introduces boundary-aware training [16], and TimeSuite applies temporal adaptive encoding [30]. Other approaches, including VideoLLaMA 2 [31] and Slowfast-LLava [32], leverage specialized spatial-temporal modules. Despite these advances, issues such as information loss and high computational cost remain. To address this, we propose an LLM-based agent framework that avoids full video processing and using diverse tools to extract multi-granular information.

2.2 LLM-based agent system for video understanding

LLM-based agent systems have emerged as a key application of large language models [33, 34, 35], and the computer vision field is exploring their use in video understanding [36, 37]. VideoINSTA [38] employs event-based temporal and content-based spatial reasoning to enhance LLMs' video reasoning. In [39], structured spatio-temporal memory is proposed for supporting video agent systems. VideoAgent [1] presents a framework for interactive reasoning and planning with long visual inputs. While these works mainly focus on agent architecture, our approach emulates human video understanding by introducing a dedicated CoT with a plan-adjust mode to strengthen LLM reasoning and decision-making in long-video scenarios.

2.3 Bootstrapping CoT in LLM reasoning

CoT has been widely applied to enhance LLM reasoning [40, 41, 42], with research often emphasizing reward process design, such as tree search with reinforcement learning [43], Q-value ranking optimization [44], advantage verifiers [25], and meta-reward steps [45]. However, these methods typically require extra parameters or reasoning steps, complicating their use in video agent systems. Instead, our approach leverages uncertainty in both LLMs and tools to guide the CoT process. This strategy is easy to implement and mitigates hallucinations frequently observed in agent systems.

3 Method

This section introduces VideoAgent2, with the overall framework illustrated in Fig. 2. Compared to the previous SOTA method, VideoAgent [1], our approach incorporates two major improvements: (1) New information retrieval method based on uncertainty-aware CoT reasoning: VideoAgent retrieves key frames by exhaustively comparing CLIP embeddings of all frames with the target, resulting in fixed granularity, inefficiency, and limited reasoning capacity. It also heavily relies on CLIP, which can introduce bias. In contrast, VideoAgent2 enhances efficiency and maximizes the LLM's reasoning capacity. It allows the LLM to autonomously identify temporal intervals and use an uncertainty-aware CoT process to enable iterative, coarse-to-fine reasoning and better adapts to diverse user needs and varying levels of granularity. (2) Redesigned pipeline: To address the limitations in VideoAgent—where uniform sampling for context retrieval may overlook critical information and cause subsequent retrieval failures—VideoAgent2 adopts a segment-based caption and summarization approach, which will be described in detail in the following section. Furthermore, we develop a variety of specialized tools tailored for VideoAgent2 to enhance its overall effectiveness.

VideoAgent2 mimics human video comprehension: given a video and a question, a person first watches the video roughly to gain a general context, then retrieves specific segments iteratively until sufficient evidence is gathered. Depending on the complexity of the question, this retrieval process may involve multiple steps, each of which is an assessment made to determine the adequacy of the

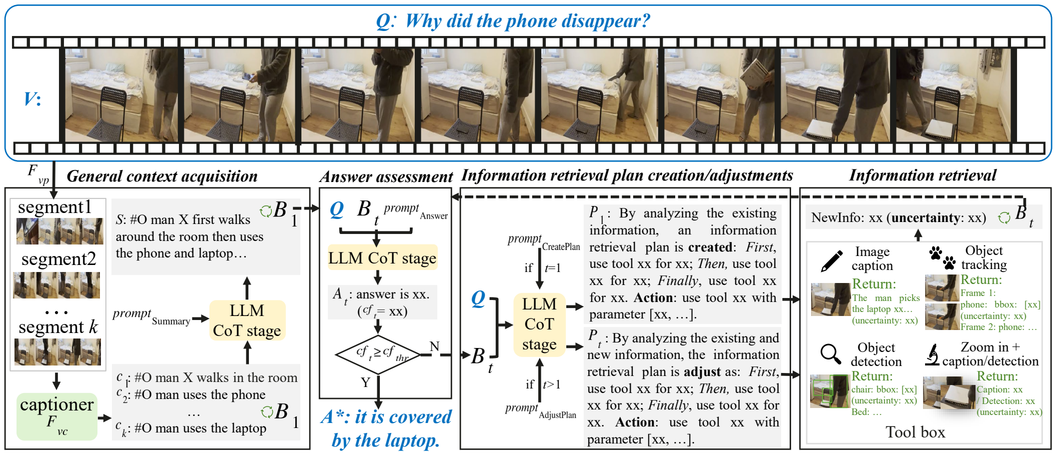

Figure 2: Overview of VideoAgent2. VideoAgent2 answers a question Q about a video V through a pipeline consisting of four phases: general context acquisition, answer assessment, information retrieval plan creation/adjustment, and information retrieval. Details of each phase are introduced in Section 3.

information obtained so far. We formalize the answering process to a query Q over video V in four stages: general context acquisition, answer assessment, retrieval plan formulation/adjustment, and targeted information retrieval. At each step t, the system state is $B_t, P_t, A_t, cf_t \mid 1 \leq t \leq T$, where B_t is the information memory bank storing all information, P_t is the plan for retrieving additional information, A_t is the answer generated based on B_t, and cf_t its confidence of the A_t.

3.1 Phase 1: General context acquisition

Inspired by how humans comprehend long videos—first grasping general context, then examining segments relevant to the question—we start by acquiring the general context of the target video. Two key issues are addressed in this phase: (1) Computational efficiency: Excessive detail increases computational load and risks focusing on irrelevant content [13]. (2) Information integrity: The general context should contain as much information as possible to avoid failing to identify key segments later. To achieve this, we proceed as follows:

- *Step 1*. The long video of length L is downsampled to frame rate fps_d and split into segments of n seconds using a preprocessor F_{vp}.
- *Step 2*. Captions $C = \{c_1, c_2, \ldots, c_k\}, k = L/n$, are generated for all segments via a lightweight video captioner F_{vc}.
- *Step 3*. C is input to an LLM to produce a summary S for the entire video.

Computational efficiency is ensured in *Step 1* by reducing frame numbers. In *Step 2*, unlike uniformly sampling frames and using an image captioner in VideoAgent, a video captioner preserves temporal context and prevents the context information loss. To avoid irrelevant or disconnected captions, and following [2], the LLM analyzes all captions and generates a summary via CoT prompting, allowing it to reflect on temporal and spatial relationships and better understand the content. By the end of Phase 1, the general context $B_1 = C, S$ is obtained. As B_1 may lack fine-grained details, we first attempt to answer the question based on B_1, then evaluate if further information retrieval is needed. Details of this process are provided in the next section.

3.2 Phase 2: Answer assessment

Humans typically evaluate whether their current information suffices to answer a question, a process termed answer assessment. Building on [1], we refine the prediction-evaluation procedure by merging it into a single step: the LLM generates both an answer A_t to question Q based on B_t and a confidence score cf_t (ranging from 0 to 5) simultaneously. The subsequent action is determined by comparing cf_t with a manually set threshold cf_{thr}:

- *Action 1*: If $cf_t \geq cf_{thr}$, the LLM is deemed to have sufficient information, confirming final answer A^* as A_t.

- *Action 2*: If $cf_t < cf_{thr}$, more information retrieval is needed, initiating Phase 3 for plan creation or adjustment.

This phase leverages LLM uncertainty (self-reflection) to judge information adequacy [46]. When information is insufficient, the LLM is guided to create or revise the retrieval plan, enabling it to iteratively plan tool calls and interpret retrieved information within a coherent chain of thought.

3.3 Phase 3: Information retrieval plan creation/adjustments

When humans find their current information insufficient to answer a question, they typically recall relevant ranges and scrutinize specific content, first devising an information retrieval plan based on the question before examining pertinent segments. For instance, to answer "What is the animal in the photo that the man is holding in his hand at the beginning of the video?", one might plan: (1) find the man in 0-10s; (2) locate his right hand; (3) closely examine the photo in the hand. As new information is acquired, the plan is adjusted—e.g., if no man is found in 0-10s, shift the search to 10-20s, or add steps if needed.

Inspired by this, we propose a plan-adjust CoT mode where the LLM generates an initial retrieval plan based on the general context and question, then iteratively adjusts the plan as new information is acquired to guide subsequent retrieval. Actions in Phase 3 are as follows:

- *Action 1*: If $t = 1$, the system is in the initial state. The LLM creates an initial information retrieval plan P_1 based on B_1 and Q.
- *Action 2*: If $t > 1$, the system is in the retrieval process. The LLM updates the previous plan P_{t-1} to produce P_t based on B_t and Q.

Unlike VideoAgent [1], which uses CLIP-based similarity for retrieval range selection—incurring high computational cost—we allow the LLM to directly decide two key parameters per tool invocation (for system stability, only one tool is invoked per step): (1) the frame range for retrieval, determined from C, and (2) tool-specific parameters. This flexible approach enables progressive localization at varying granularities, supporting more precise information retrieval.

3.4 Phase 4: Information retrieval

In this phase, the LLM invokes tools to retrieve new information based on the created or adjusted plan P_t. However, tools can introduce errors or noise; for example, object detectors may misidentify rare objects, and image captioners may inaccurately describe events. Such errors can compound during iterative retrieval, affecting the LLM's plan adjustments. To mitigate this, we introduce uncertainty to guide the CoT process: each tool returns a confidence score along with its output, as detailed in Section 4.1. This allows the LLM to consider both the content and reliability of retrieved information when refining the retrieval plan and integrating results to make the final decision. Together with cf_t, a complete uncertainty-aware CoT process is established in VideoAgent2, where both LLM and tool uncertainties inform information acquisition and analysis, thereby improving overall system reliability.

The VideoAgent2 pipeline is summarized in Algorithm 1.

4 Experiments

We first introduce the experimental settings and then present the experimental results of our methods and baselines, demonstrating the effectiveness of our method.

4.1 Experimental Setting

Datasets. We follow the strong baselines [1, 4] using three well-established datasets to evaluate the proposed method:

- Egoschema [26]. EgoSchema contains over 5,000 human-curated multiple-choice QA pairs from more than 250 hours of real video. The subtest set includes 500 questions with public labels. We compare VideoAgent2 to leading published methods on the leaderboard, following the evaluation protocol in [2].
- NExT-QA [27]. The NExT-QA dataset features 5,440 natural videos and 4,880 multiple-choice questions spanning action reasoning, temporal action reasoning, and scene comprehension. Its validation set contains 570 videos and 5,000 questions. The ATP-hard subset

Algorithm 1 VideoAgent2

Require: long video V, question Q, LLM F_{llm}, video preprocessor F_{vp}, video captioner F_{vc}, video tools $F_{vt1}, F_{vt2}, \cdots, F_{vtM}$, max number of answer assessments T, confidence threshold cf_{thr}
Ensure: information memory bank, information retrieval plan and predicted answer $\{B_t, P_t, A_t | 1 \leq t \leq T\}$

```
C = {c_1, c_2, ..., c_k} ← F_vc(F_vp(V))
S ← F_llm(C, prompt_Summary)
B_1 = {C, S}
for t = 1 to T do
    A_t, cf_t ← F_llm(B_t, Q, prompt_Answer)
    if cf_t ≥ cf_thr then
        break
    else
        if t == 1 then
            P_t ← F_llm(B_t, Q, prompt_CreatePlan)
        else
            P_t ← F_llm(B_t, Q, P_{t-1}, prompt_AdjustPlan)
        end if
        NewInfo ← F_vtm(P_t)
        B_{t+1} ← Merge(B_t, NewInfo)
    end if
end for
return A* = A_t
```

comprises the most challenging questions requiring long-term temporal reasoning. We mainly compare VideoAgent2 to top zero-shot methods on the NExT-QA validation set and ATP-hard subset, following [1, 47], and also report leading supervised results.

- IntentQA [28]. IntentQA includes 4,303 videos and 16,000 multiple-choice QA pairs focused on intent reasoning. The test set comprises 567 videos and 2,134 questions. We mainly compare VideoAgent2 to top zero-shot methods on the test set, following [4], and also report the leading supervised results.

Metric. Since the tasks in all datasets are multiple-choice, accuracy is used as the evaluation metric by following [4, 1], defined as the number of correct answers divided by the total number.

System parameters. fps_d is set to 1 (we also compare different fps_d in the ablation study), and n is set to 4 to ensure computational efficiency. cf_{thr} is set to 5 to ensure sufficient confidence in the final answer. T is set to 5 to prevent dead ends in extreme situations.

LLM and tools selection. We adopt GPT-4o [48] as the LLM F_{llm}, following [2]. For video captioning, we use the lightweight LaViLa model [49] as recommended in [1, 2]. The selected or designed tools, along with confidence score extraction methods, are as follows:

- *Image caption*: GPT-4o [48] generates frame captions, automatically assigning a confidence score (0–1) to each sentence or clause by using an in-context learning prompt (e.g., "The image shows a person sewing fabric (confidence=0.9)...").
- *Object detection*: A designed tool based on SAM2 [50] and Yolov11 [51] detects objects in specified frames, using SAM2's confidence score for each object.
- *Image zoom in + caption*: OpenCV [52] zooms into specified image areas, and GPT-4o captions the zoomed region with confidence scores as above.
- *Image zoom in + object detection*: OpenCV zooms into regions for object detection, with confidence scores from the object detection tool.
- *Object tracking*: A designed object tracking tool based on SAM2 and Yolov11 tracks objects within a given frame range, using SAM2's confidence in each frame.

Detailed parameters/descriptions of these tools are provided in the Appendix A.
Tool parameters. To ensure reproducibility and maintain consistency, we retain the default pa-

Table 1: EgoSchema dataset

Method	full test set	sub test set
LongViViT [53]	33.3	56.8
LLoVi [4]	50.3	57.6
VideoAgent [1]	54.1	60.2
GPT-4V [54]	55.6	63.5
ProViQ [55]	57.1	61.2
InternVideo2 [8]	60.2	-
Gemini1.5 Pro [56]	63.2	-
LifelongMem [3]	64.7	72.0
iLearn [2]	74.6	58.8
VideoAgent2 (ours)	**75.4**	**80.6**

Table 2: NExT-QA dataset

Method	val set	ATP hard set
Supervised		
ViLA [20]	74.4	-
VideoChat2 [7]	79.5	68.2
LLaVA-OV [57]	80.2	-
LinVT [58]	85.5	69.1
Zero-shot		
ViperGPT [59]	60.0	-
SeViLA [21]	63.6	50.8
VideoAgent [1]	71.3	58.4
LLoVi [4]	73.8	-
VideoAgent2	**80.5**	**68.2**

Table 3: IntentQA dataset

Method	test set
Supervised	
IntentQA [28]	57.6
Human [28]	78.5
VideoChat2 [7]	81.9
Zero-shot	
LLoVi [4]	67.1
VideoAgent [1]	69.3
LVNet [60]	71.1
ENTER [5]	71.5
VideoAgent2	**73.9**

rameters for each tool during our experiments. These settings are listed in Appendix A, Table 8.

Baselines. We compare VideoAgent2 with work that performed strongly on each dataset by following [1, 4].

4.2 Main results

VideoAgent2 achieves the SOTA among zero-shot methods on all datasets (including subsets), which are shown in Tables 1, 2 and 3, respectively.

As shown in Tables 1, 2, and 3, VideoAgent2 achieves the best results on the EgoSchema dataset and among zero-shot methods for NExT-QA and IntentQA. Compared to previous SOTA zero-shot results, accuracy increases by 0.8% and 8.6% on the EgoSchema full and sub test sets, 6.7% and 9.8% on the NExT-QA validation set and ATP-hard subset, and 2.4% on the IntentQA test set. Notably, while iLearn performs well on the EgoSchema full set, its accuracy drops on the subset, likely due to its reliance on video captioner outputs, making it less robust to distribution shifts. In contrast, VideoAgent2 maintains strong performance on both sets, demonstrating robustness. On NExT-QA, VideoAgent2 notably surpasses previous zero-shot methods and approaches supervised SOTA results on the ATP-hard subset, highlighting its strength in causal, temporal, and descriptive reasoning. Additionally, VideoAgent2 nearly reaches human-level performance among zero-shot methods on IntentQA.

4.3 Case study

We present a case study using the video and question from Fig. 2. Due to space constraints, we illustrate the full reasoning process in Appendix B, Fig. 5. Ellipses (. . .) indicate omitted content, with complete prompt details available in the Appendix C. From this example, we draw the following insights:

- VideoAgent2 completes the task with just three tool calls and four answer assessments, primarily focusing on three key frames—demonstrating high frame efficiency. In contrast, VideoAgent [1] processes 15 frames and still fails, as does Llava-Onevision-7B.
- The plan-adjust CoT effectively adapts the retrieval strategy over time. Between $t = 1$ and $t = 3$, rather than following a fixed retrieval plan, VideoAgent2 dynamically revises its plan based on newly retrieved data, progressing from coarse to fine-grained information.
- The uncertainty-guided CoT effectively addresses tool noise and improves multi-tool integration. For instance, while the caption for 12s–16s misdescribes the action and object detection at $t = 2$ fails to recognize key items, the LLM identifies uncertainty, revises the plan, and leverages additional tools. By $t = 4$, it successfully compares frames 19 and 20, using confidence scores to reach the correct answer. This demonstrates how uncertainty awareness helps suppress noise and hallucination, guiding more accurate and robust decision-making.

4.4 Analysis of tool call

The key feature of VideoAgent2 is its uncertainty-aware CoT process, which invokes tools through the plan-adjust mode. To better understand this process, we analyze tool calls across different datasets and question types.

Number of tool calls for different datasets. We analyze the number of tool calls per sample on the EgoSchema test set, NExT-QA validation set, and IntentQA test set, with results shown in Fig. 3a. A count of 0 indicates that VideoAgent2 answers the question using only the initial context B_1, without further retrieval. The maximum number of tool calls is $T-1$ (with $T=5$ in our setup). In all datasets, samples with zero tool calls represent the largest proportion—over 30%—we attribute this to the fact that video clip captions ensure the completeness of the context information, and the model also deepens its understanding of the spatio-temporal content of the full video by summarizing clip captions. EgoSchema shows a lower percentage of zero-call cases and more samples requiring 3–4 tool calls, likely due to its longer videos and higher reasoning complexity [26]. In contrast, NExT-QA and IntentQA peak at 2 and 3 calls, respectively, with IntentQA exhibiting higher complexity, as it includes the most challenging questions from NExT-QA.

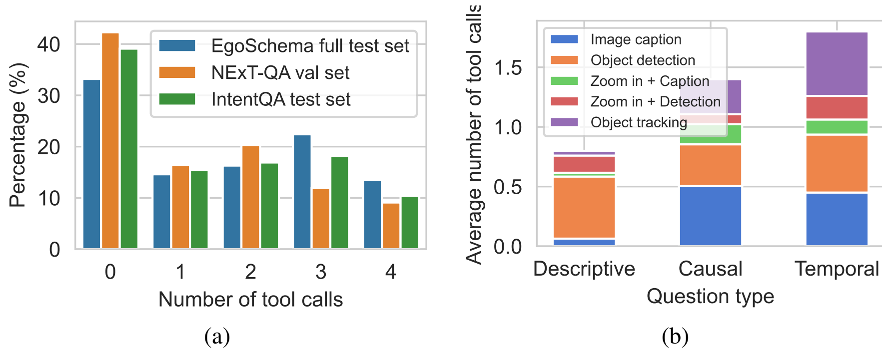

Figure 3: Analysis of tool call number statistics. (a) Proportion of samples with different numbers of tool calls in different datasets. A tool call number of 0 means that for this sample, VideoAgent2 has obtained enough information from the general context information B_1 to answer the question without the need for new information retrieval. The maximum tool call number is equal to $T-1$, T is set to 5 in our experiment. (b) The average number of tool calls for each type of question and the average number of calls for each tool in NExT-QA val set.

Number of tool calls for different question types. The NExT-QA dataset classifies questions as: (1) *Causal Questions*, which investigate observable cause-effect relations; (2) *Temporal Questions*, which assess the sequence of multi-object interactions; and (3) *Descriptive Questions*, focused on scene elements like locations, objects, and key actions. We compute the average number of tool calls per sample and per tool for each type in the NExT-QA validation set (Fig. 3b). Descriptive questions have the lowest average tool calls (0.8), mostly using object detection, as factual descriptions are generally covered in the context and require only basic retrieval. In contrast, temporal questions require the most tool calls, especially from the object tracking model, due to their complex spatio-temporal reasoning. Both causal and temporal questions also see increased use of image captioning tools, as these help visualize event details needed for reasoning, consistent with findings in [1].

4.5 Ablation study

We perform comprehensive ablation experiments to demonstrate the effectiveness of the proposed approach.

Different frame rate fps_d. It is worth noting that the modest 0.8% gain on the EgoSchema full set results from our use of the commonly adopted 1 fps frame rate [26, 1], unlike the best-performing baseline, which uses 30 fps [2]. To compare fairly, we also tested the 30 fps setting and achieved 78.6, a 4.0% improvement. This confirms that higher frame rates provide richer context but at a much higher computational cost.

Running cost. We employ GPT-4o as the LLM in VideoAgent2 and compare average cost and time per sample on the EgoSchema full test set with the original VideoAgent. Results in Table 4 show that VideoAgent2 delivers higher performance with comparable computational costs and faster

Table 4: Average cost and time consumption of each sample on egoschema full test set

Model	average cost	average time
VideoAgent	0.041 USD	2.7mins
VideoAgent2 (*ours*)	0.047 USD	2.3mins

inference. For details on performance gains, see the Method section of the original paper. The efficiency improvement is largely due to VideoAgent2 enabling the LLM to selectively analyze key temporal ranges, avoiding the exhaustive CLIP embedding and similarity computation required by VideoAgent. Typical GPU memory usage for all tools is reported in Appendix B, Table 9.

Ablation of uncertainty-aware CoT. We evaluate the impact of uncertainty-aware CoT in VideoAgent2 using four experimental settings: (1) disabling tool-generated confidence scores, (2) disabling plan adjustment so the LLM executes a fixed retrieval plan, (3) disabling the entire CoT process so the LLM calls tools and answers directly, and (4) disabling all tool calls so the LLM relies only on B_1. Each setting is incrementally built on the previous one. Experiments are conducted on the EgoSchema full test set, NExT-QA validation set, and IntentQA test set, with results in Table 5. Settings (1) and (2) show the greatest negative impact, with accuracy dropping by 4.3% and 4.0%, respectively. This underscores the importance of tool uncertainty quantification and iterative plan adjustment in the CoT process, enabling the LLM to assess reliability and adapt its retrieval strategy for better answers. The further declines in settings (3) and (4) also emphasize the necessity of tool calls and effective call planning in VideoAgent2.

Table 5: Results of VideoAgent2 under different settings

Method	EgoSchema fullset	NExT-QA val set	IntentQA test set	average
VideoAgent2	75.4	80.5	73.9	76.6
setting 1	71.3	76.0	69.5	72.3
setting 2	67.1	71.5	66.3	68.3
setting 3	65.6	69.8	65.6	67.0
setting 4	60.2	68.9	63.1	64.1

Ablation of different LLM. We evaluate the impact of different LLMs in VideoAgent2 across the EgoSchema full test set, NExT-QA validation set, and IntentQA test set (Table 6). Commercial models like GPT-4o [48] and GPT-4 [61] are compared with open-source alternatives such as Deepseek-V3 [62] and Llama3.3-70B [63]. GPT-4o achieves the best overall performance, while Deepseek-V3 performs competitively, offering a flexible open-source option.

Table 6: Results of VideoAgent2 with different LLM

LLM	EgoSchema full test set	NExT-QA val set	IntentQA test set
GPT-4o	75.4	80.5	73.9
GPT-4	74.1	78.2	72.6
Deepseek-V3	74.7	79.5	73.4
Llama3.3-70B	70.5	75.3	68.6

Failure analysis. We analyze the failure cases in VideoAgent2 and summarize two main issues (1) Omission of general context information and (2) Limitations of current tools in capturing spatiotemporal information. We show a more detailed analysis in Appendix D.

Comparison with advanced MLLMs. We compare the performance of VideoAgent2 with four leading MLLMs on the EgoSchema full test set and the Next-QA validation set, as shown in Table 7. We observe that VideoAgent2 significantly outperforms the 32B and 38B models, and even surpasses InternVL2.5-78B. Its performance is slightly below that of Qwen2.5-VL-72B, which we believe may be attributed to Qwen2.5-VL's use of advanced pre-training strategies and large-scale video data.

Table 7: Performance comparison with advanced MLLMs

Model	EgoSchema	NExT-QA
Qwen2.5-VL-32B	71.1	77.2
Qwen2.5-VL-72B	76.8	83.3
InternVL2.5-38B	68.5	75.8
InternVL2.5-78B	73.7	80.1
VideoAgent2 (*ours*)	75.4	80.5

5 Conclusion

In this paper, we introduce VideoAgent2, an enhanced LLM-based agent system designed for effective long-form video understanding through a novel uncertainty-aware Chain-of-Thought (CoT) mechanism. VideoAgent2 addresses critical challenges faced by existing video agent systems, including limited reasoning capabilities and susceptibility to errors introduced by external tools. The proposed uncertainty-aware CoT mechanism enables adaptive and robust reasoning by incrementally refining information retrieval plans, guided by the uncertainty estimation derived from both internal assessments by the LLM and external tool outputs. Extensive experiments conducted on prominent benchmarks—including EgoSchema, NExT-QA, and IntentQA—demonstrate that VideoAgent2 achieves state-of-the-art performance, significantly outperforming existing zero-shot methods. Future work will focus on further refining uncertainty estimation methods and exploring additional multi-modal integration strategies to continuously improve the generalization and efficiency of LLM-based video understanding systems.

References

[1] Xiaohan Wang, Yuhui Zhang, Orr Zohar, and Serena Yeung-Levy. Videoagent: Long-form video understanding with large language model as agent. *arXiv preprint arXiv:2403.10517*, 2024.

[2] Hcqa @ ego4d egoshema challenge 2024. 2024.

[3] Ying Wang, Yanlai Yang, and Mengye Ren. Lifelongmemory: Leveraging llms for answering queries in egocentric videos. *arXiv preprint arXiv:2312.05269*, 2023.

[4] Ce Zhang, Taixi Lu, Md Mohaiminul Islam, Ziyang Wang, Shoubin Yu, Mohit Bansal, and Gedas Bertasius. A simple llm framework for long-range video question-answering. *arXiv preprint arXiv:2312.17235*, 2023.

[5] Hammad Ayyubi, Junzhang Liu, Ali Asgarov, Zaber Ibn Abdul Hakim, Najibul Haque Sarker, Zhecan Wang, Chia-Wei Tang, Hani Alomari, Md Atabuzzaman, Xudong Lin, et al. Enter: Event based interpretable reasoning for videoqa. *arXiv preprint arXiv:2501.14194*, 2025.

[6] Xinran Dai and Jing Wang. Effect of online video infotainment on audience attention. *Humanities and Social Sciences Communications*, 10(1):1–18, 2023.

[7] Kunchang Li, Yali Wang, Yinan He, Yizhuo Li, Yi Wang, Yi Liu, Zun Wang, Jilan Xu, Guo Chen, Ping Luo, et al. Mvbench: A comprehensive multi-modal video understanding benchmark. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 22195–22206, 2024.

[8] Yi Wang, Kunchang Li, Xinhao Li, Jiashuo Yu, Yinan He, Guo Chen, Baoqi Pei, Rongkun Zheng, Jilan Xu, Zun Wang, et al. Internvideo2: Scaling video foundation models for multimodal video understanding. *arXiv preprint arXiv:2403.15377*, 2024.

[9] Yang Jin, Zhicheng Sun, Kun Xu, Liwei Chen, Hao Jiang, Quzhe Huang, Chengru Song, Yuliang Liu, Di Zhang, Yang Song, et al. Video-lavit: Unified video-language pre-training with decoupled visual-motional tokenization. *arXiv preprint arXiv:2402.03161*, 2024.

[10] Zesen Cheng, Sicong Leng, Hang Zhang, Yifei Xin, Xin Li, Guanzheng Chen, Yongxin Zhu, Wenqi Zhang, Ziyang Luo, Deli Zhao, and Lidong Bing. Videollama 2: Advancing spatial-temporal modeling and audio understanding in video-llms. *arXiv preprint arXiv:2406.07476*, 2024. URL https://arxiv.org/abs/2406.07476.

[11] Hao Fei, Shengqiong Wu, Wei Ji, Hanwang Zhang, Meishan Zhang, Mong-Li Lee, and Wynne Hsu. Video-of-thought: Step-by-step video reasoning from perception to cognition. 2024.

[12] Iz Beltagy, Matthew E Peters, and Arman Cohan. Longformer: The long-document transformer. *arXiv preprint arXiv:2004.05150*, 2020.

[13] Bo He, Hengduo Li, Young Kyun Jang, Menglin Jia, Xuefei Cao, Ashish Shah, Abhinav Shrivastava, and Ser-Nam Lim. Ma-lmm: Memory-augmented large multimodal model for long-term video understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13504–13514, 2024.

[14] Yanwei Li, Chengyao Wang, and Jiaya Jia. Llama-vid: An image is worth 2 tokens in large language models. *arXiv preprint arXiv:2311.17043*, 2023.

[15] Enxin Song, Wenhao Chai, Guanhong Wang, Yucheng Zhang, Haoyang Zhou, Feiyang Wu, Haozhe Chi, Xun Guo, Tian Ye, Yanting Zhang, et al. Moviechat: From dense token to sparse memory for long video understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18221–18232, 2024.

[16] Bin Huang, Xin Wang, Hong Chen, Zihan Song, and Wenwu Zhu. Vtimellm: Empower llm to grasp video moments. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14271–14280, 2024.

[17] Hao Fei, Shengqiong Wu, Wei Ji, Hanwang Zhang, Meishan Zhang, Mong-Li Lee, and Wynne Hsu. Video-of-thought: Step-by-step video reasoning from perception to cognition. In *Forty-first International Conference on Machine Learning*, 2024.

[18] Chaoyi Zhang, Kevin Lin, Zhengyuan Yang, Jianfeng Wang, Linjie Li, Chung-Ching Lin, Zicheng Liu, and Lijuan Wang. Mm-narrator: Narrating long-form videos with multimodal in-context learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13647–13657, 2024.

[19] Yue Fan, Xiaojian Ma, Rujie Wu, Yuntao Du, Jiaqi Li, Zhi Gao, and Qing Li. Videoagent: A memory-augmented multimodal agent for video understanding. *arXiv preprint arXiv:2403.11481*, 2024.

[20] Xijun Wang, Junbang Liang, Chun-Kai Wang, Kenan Deng, Yu Michael Lou, Ming Lin, and Shan Yang. Vila: Efficient video-language alignment for video question answering. 2024.

[21] Shoubin Yu, Jaemin Cho, Prateek Yadav, and Mohit Bansal. Self-chained image-language model for video localization and question answering. *Advances in Neural Information Processing Systems*, 36, 2024.

[22] 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. *arXiv preprint arXiv:2405.19209*, 2024.

[23] Zhuo Zhi, Chen Feng, Adam Daneshmend, Mine Orlu, Andreas Demosthenous, Lu Yin, Da Li, Ziquan Liu, and Miguel RD Rodrigues. Seeing and reasoning with confidence: Supercharging multimodal llms with an uncertainty-aware agentic framework. *arXiv preprint arXiv:2503.08308*, 2025.

[24] Dan Zhang, Sining Zhoubian, Ziniu Hu, Yisong Yue, Yuxiao Dong, and Jie Tang. Rest-mcts*: Llm self-training via process reward guided tree search, 2024a. *URL https://arxiv.org/abs/2406.03816*.

[25] Amrith Setlur, Chirag Nagpal, Adam Fisch, Xinyang Geng, Jacob Eisenstein, Rishabh Agarwal, Alekh Agarwal, Jonathan Berant, and Aviral Kumar. Rewarding progress: Scaling automated process verifiers for llm reasoning. *arXiv preprint arXiv:2410.08146*, 2024.

[26] 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.

[27] Junbin Xiao, Xindi Shang, Angela Yao, and Tat-Seng Chua. Next-qa: Next phase of question-answering to explaining temporal actions. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9777–9786, 2021.

[28] Jiapeng Li, Ping Wei, Wenjuan Han, and Lifeng Fan. Intentqa: Context-aware video intent reasoning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 11963–11974, 2023.

[29] Yanwei Li, Chengyao Wang, and Jiaya Jia. Llama-vid: An image is worth 2 tokens in large language models. In *European Conference on Computer Vision*, pages 323–340. Springer, 2025.

[30] Xiangyu Zeng, Kunchang Li, Chenting Wang, Xinhao Li, Tianxiang Jiang, Ziang Yan, Songze Li, Yansong Shi, Zhengrong Yue, Yi Wang, et al. Timesuite: Improving mllms for long video understanding via grounded tuning. *arXiv preprint arXiv:2410.19702*, 2024.

[31] 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.

[32] Mingze Xu, Mingfei Gao, Zhe Gan, Hong-You Chen, Zhengfeng Lai, Haiming Gang, Kai Kang, and Afshin Dehghan. Slowfast-llava: A strong training-free baseline for video large language models. *arXiv preprint arXiv:2407.15841*, 2024.

[33] Lei Wang, Chen Ma, Xueyang Feng, Zeyu Zhang, Hao Yang, Jingsen Zhang, Zhiyuan Chen, Jiakai Tang, Xu Chen, Yankai Lin, et al. A survey on large language model based autonomous agents. *Frontiers of Computer Science*, 18(6):186345, 2024.

[34] Weiwen Liu, Xu Huang, Xingshan Zeng, Xinlong Hao, Shuai Yu, Dexun Li, Shuai Wang, Weinan Gan, Zhengying Liu, Yuanqing Yu, et al. Toolace: Winning the points of llm function calling. *arXiv preprint arXiv:2409.00920*, 2024.

[35] Taicheng Guo, Xiuying Chen, Yaqi Wang, Ruidi Chang, Shichao Pei, Nitesh V Chawla, Olaf Wiest, and Xiangliang Zhang. Large language model based multi-agents: A survey of progress and challenges. *arXiv preprint arXiv:2402.01680*, 2024.

[36] Difei Gao, Lei Ji, Luowei Zhou, Kevin Qinghong Lin, Joya Chen, Zihan Fan, and Mike Zheng Shou. Assistgpt: A general multi-modal assistant that can plan, execute, inspect, and learn. *arXiv preprint arXiv:2306.08640*, 2023.

[37] Zongxin Yang, Guikun Chen, Xiaodi Li, Wenguan Wang, and Yi Yang. Doraemongpt: Toward understanding dynamic scenes with large language models. *arXiv preprint arXiv:2401.08392*, 2024.

[38] Ruotong Liao, Max Erler, Huiyu Wang, Guangyao Zhai, Gengyuan Zhang, Yunpu Ma, and Volker Tresp. Videoinsta: Zero-shot long video understanding via informative spatial-temporal reasoning with llms. *arXiv preprint arXiv:2409.20365*, 2024.

[39] 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, 2025.

[40] Guowei Xu, Peng Jin, Li Hao, Yibing Song, Lichao Sun, and Li Yuan. Llava-o1: Let vision language models reason step-by-step. *arXiv preprint arXiv:2411.10440*, 2024.

[41] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837, 2022.

[42] Zhuosheng Zhang, Aston Zhang, Mu Li, and Alex Smola. Automatic chain of thought prompting in large language models. *arXiv preprint arXiv:2210.03493*, 2022.

[43] Dan Zhang, Sining Zhoubian, Ziniu Hu, Yisong Yue, Yuxiao Dong, and Jie Tang. Rest-mcts*: Llm self-training via process reward guided tree search. *arXiv preprint arXiv:2406.03816*, 2024.

[44] Wendi Li and Yixuan Li. Process reward model with q-value rankings. *arXiv preprint arXiv:2410.11287*, 2024.

[45] Tianhao Wu, Weizhe Yuan, Olga Golovneva, Jing Xu, Yuandong Tian, Jiantao Jiao, Jason Weston, and Sainbayar Sukhbaatar. Meta-rewarding language models: Self-improving alignment with llm-as-a-meta-judge. *arXiv preprint arXiv:2407.19594*, 2024.

[46] Noah Shinn, Federico Cassano, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. Reflexion: Language agents with verbal reinforcement learning. *Advances in Neural Information Processing Systems*, 36, 2024.

[47] Shyamal Buch, Cristóbal Eyzaguirre, Adrien Gaidon, Jiajun Wu, Li Fei-Fei, and Juan Carlos Niebles. Revisiting the" video" in video-language understanding. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2917–2927, 2022.

[48] Aaron Hurst, Adam Lerer, Adam P Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, et al. Gpt-4o system card. *arXiv preprint arXiv:2410.21276*, 2024.

[49] Yue Zhao, Ishan Misra, Philipp Krähenbühl, and Rohit Girdhar. Learning video representations from large language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6586–6597, 2023.

[50] Nikhila Ravi, Valentin Gabeur, Yuan-Ting Hu, Ronghang Hu, Chaitanya Ryali, Tengyu Ma, Haitham Khedr, Roman Rädle, Chloe Rolland, Laura Gustafson, Eric Mintun, Junting Pan, Kalyan Vasudev Alwala, Nicolas Carion, Chao-Yuan Wu, Ross Girshick, Piotr Dollár, and Christoph Feichtenhofer. Sam 2: Segment anything in images and videos. *arXiv preprint arXiv:2408.00714*, 2024. URL `https://arxiv.org/abs/2408.00714`.

[51] Glenn Jocher and Jing Qiu. Ultralytics yolo11, 2024. URL `https://github.com/ultralytics/ultralytics`.

[52] Ivan Culjak, David Abram, Tomislav Pribanic, Hrvoje Dzapo, and Mario Cifrek. A brief introduction to opencv. In *2012 proceedings of the 35th international convention MIPRO*, pages 1725–1730. IEEE, 2012.

[53] Pinelopi Papalampidi, Skanda Koppula, Shreya Pathak, Justin Chiu, Joe Heyward, Viorica Patraucean, Jiajun Shen, Antoine Miech, Andrew Zisserman, and Aida Nematzdeh. A simple recipe for contrastively pre-training video-first encoders beyond 16 frames. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14386–14397, 2024.

[54] Ivana Balažević, Yuge Shi, Pinelopi Papalampidi, Rahma Chaabouni, Skanda Koppula, and Olivier J Hénaff. Memory consolidation enables long-context video understanding. *arXiv preprint arXiv:2402.05861*, 2024.

[55] Rohan Choudhury, Koichiro Niinuma, Kris M Kitani, and László A Jeni. Zero-shot video question answering with procedural programs. *arXiv preprint arXiv:2312.00937*, 2023.

[56] 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.

[57] Bo Li, Yuanhan Zhang, Dong Guo, Renrui Zhang, Feng Li, Hao Zhang, Kaichen Zhang, Peiyuan Zhang, Yanwei Li, Ziwei Liu, et al. Llava-onevision: Easy visual task transfer. *arXiv preprint arXiv:2408.03326*, 2024.

[58] Lishuai Gao, Yujie Zhong, Yingsen Zeng, Haoxian Tan, Dengjie Li, and Zheng Zhao. Linvt: Empower your image-level large language model to understand videos. *arXiv preprint arXiv:2412.05185*, 2024.

[59] Dídac Surís, Sachit Menon, and Carl Vondrick. Vipergpt: Visual inference via python execution for reasoning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 11888–11898, 2023.

[60] Jongwoo Park, Kanchana Ranasinghe, Kumara Kahatapitiya, Wonjeong Ryoo, Donghyun Kim, and Michael S Ryoo. Too many frames, not all useful: Efficient strategies for long-form video qa. *arXiv preprint arXiv:2406.09396*, 2024.

[61] 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.

[62] Aixin Liu, Bei Feng, Bing Xue, Bingxuan Wang, Bochao Wu, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, et al. Deepseek-v3 technical report. *arXiv preprint arXiv:2412.19437*, 2024.

[63] Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

Appendix for VideoAgent2: Enhancing the LLM-based agent system for long-form video understanding by uncertainty-aware CoT

This document provides more details of our approach, organized as follows:

- A. Details of used/designed tools in VideoAgent2
- B. More experiment results
- C. Prompts for VideoAgent2
- D. Failure analysis

A Details of used/designed tools in VideoAgent2

We show more details of used/designed tools in VideoAgent2.

A.1 Image caption

We use GPT-4o to generate captions for specified frame images. We ask GPT-4o to automatically generate a confidence score ranging from 0 to 1 after each sentence or clause in the caption by the prompt:

```
"You are an assistant that generates descriptive captions for images.
For each sentence or clause in the caption, include a confidence score in the format (confidence=0.xx) after the description.
This confidence is from 0 to 1, reflecting your confidence of the caption.
Here is an example:
'The image shows a small kitchen counter with a kettle (confidence=0.94),  a round black electronic device (confidence=0.85), a loaf of bread (confidence=0.73), and some cleaning supplies (confidence=0.95). There is a trash can on the floor (confidence=0.85) and a blue tiled backsplash (confidence=0.62).'"
```

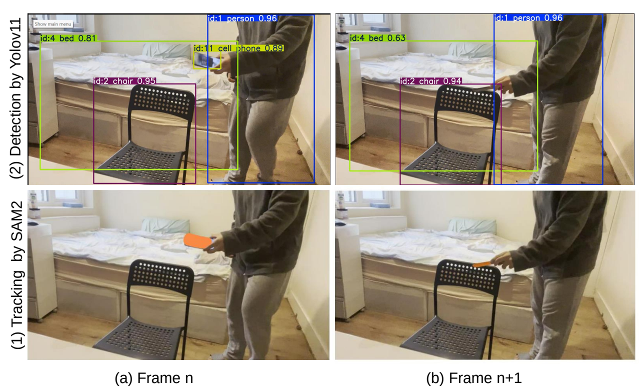

Figure 4: An example of miss detection in object detection result. In (a)(2), the target detection model correctly identifies the mobile phone with high confidence. However, in (b)(2), due to a change in the phone's orientation, the model fails to detect it. By contrast, leveraging SAM2 to track the mobile phone across these two frames effectively addresses this issue, as demonstrated in (a)(1) and (b)(1).

A.2 Object detection

Object detection models such as the Yolo series offer strong performance and speed but still encounter challenges such as missed detections, as illustrated in Fig. 4. To address this issue, we incorporate

Table 8: Parameter setting of all tools

tool name	parameter
LaViLa	max_text_length=77, top_p=0.95, num_return_sequences=4, temperature=0.7
GPT-4o	temperature=1, top_p=1,
Yolov11	iou=0.7, imgsz=640, weight=yolo11x.pt
SAM2	weight=sam2_l.pt, iou=0.45, imgsz=640

Table 9: GPU memory consumption of all tools

Tool name	GPU memory consumption
Video Clip Captioner	11.4 GB
Object detection	12.9 GB
Image caption	None (API)
Image zoom in	None (CPU operation)
Object tracking	12.9 GB
LLM	None (API)

SAM2 to construct a multi-round target detection framework that significantly reduces target loss by tracking the specified target across frames. Taking the object detection for the m-th frame as an example, the overall process is summarized as follows:

1. The frame range is extended to $[m - \alpha,\ m + \alpha]$, and all frames within this range are processed using Yolov11. To maintain computational efficiency, α is set to a small value, typically 5.
2. For each target, the frame with the highest confidence score provided by Yolov11, exceeding a predefined threshold, is selected. The bounding box from this frame is then used to initialize SAM2, which performs bi-directional tracking to retrieve the target's information in the m-th frame. We calculate the bounding box as the final detection result based on the mask provided by SAM2 in the m-th frame.

A.3 Object tracking

SAM2 demonstrates strong performance in target tracking. However, it requires manual specification of the target's initial position before starting. To enable object tracking in VideoAgent2 by simply specifying the name of the item, we use Yolov11 to automatically initialize SAM2. Taking the task of tracking the target "mobile phone" within the frame range $[m, n]$ as an example, the overall process is as follows:

1. Apply Yolov11 to detect the target "mobile phone" in frames $[m, n]$.
2. If the detection confidence exceeds a predefined threshold, use the detected bounding box to initialize SAM2. Bi-directional tracking is then performed to obtain the complete trajectory of the target across the frame range $[m, n]$.

A.4 Tool parameters.

To ensure reproducibility and maintain consistency, we retain the default parameters for each tool during our experiments. These settings are listed in Table 8.

B More experiment results

B.1 GPU memory consumption of all tools.

We record the typical GPU memory consumption of all tools involved, which is summarized in Table 9.

B.2 Case study.

We use the video and question in Fig. 2 as a case study. Fig. 5 illustrates how VideoAgent2 answers this question, providing a clear example of the proposed approach. Due to space limitations, some information is omitted using ..., and more prompt details are shown in Appendix C. Based on Fig. 5, we make the following observations:

- VideoAgent2 answers the question through three tool calls, four answer assessments and primarily focuses on three frames, demonstrating high frame efficiency. In comparison, VideoAgent [1] uses information from 15 frames and still fails to provide the correct answer. The MLLM Llava-Onevision-7B also gives the wrong answer.
- The proposed plan-adjust CoT excels in handling this complex problem. From $t = 1$ to $t = 3$, rather than following a fixed retrieval plan, VideoAgent2 adjusts the information retrieval scheme based on newly acquired data, progressing from coarse-grained to fine-grained information acquisition.
- The uncertainty-guided CoT process effectively addresses the noise introduced by the tools and enables seamless integration of different tools within the agent system to yield more reliable answers. A notable issue is that Caption 12s-16s in the general context incorrectly describes the man's action as placing the mobile phone on the bed. This mistake is avoided by implementing the retrieval plan. In NewInfo (t=2), the object detection model fails to detect the phone and the laptop, leading the LLM to incorrectly interpret that the phone is covered by the chair. However, the uncertainty in both the LLM's answer and the tool's return value successfully guides the LLM to adjust the retrieval plan and incorporate new tools. Similarly, when integrating information from different tools to make the final judgment at $t = 4$, the LLM correctly compares the content of frame19 and frame20, as well as their associated confidence scores, to give an accurate final answer. In summary, the proposed uncertainty-aware CoT effectively mitigates noise and hallucination in both the LLM and tools, allowing the LLM to refine its information retrieval strategy and make more reliable decisions when synthesizing final answers.

C Details of prompt in VideoAgent2

We show all the prompts in VideoAgent2.

- $prompt_{\text{Summary}}$

```
"A long video is segmented into consecutive 4-second clips.
Given the captions for all the clips, {C}, please summarize
    ↪ them into a coherent description of the entire video.
Please approach this task step by step, carefully
    ↪ considering the temporal and spatial relationships
    ↪ between the content in each clip during the
    ↪ summarization process.
Note for captions:
- '#C' indicates actions performed by the camera wearer (
    ↪ the person who recorded the video while wearing the
    ↪ camera).
- '#O' indicates actions performed by someone other than
    ↪ the camera wearer."
```

- $prompt_{\text{Answer}}$

```
"Now we have a memory bank {B} which stores all the
    ↪ information of a video.
In this bank:
- 'Caption' is the caption of all consecutive sub-segments
    ↪ of the video.
- 'Summary' is the summarized description of the video.
- 'Tools return value' is the retrieved information by some
    ↪ tools.
```

Video and Question: See Fig. 2.

Answer by Llava-Onevision-7B: The phone ended up on a chair. There's no evidence of loss. (×)
Answer by VideoAgent: The phone was left on the bed, probably covered by a man. (×)
Answer by VideoAgent2 (*ours*): The phone was obscured by the laptop. (√)

General context: Caption: … 8s-12s: #O man X operates the phone, 12s-16s: #O man X puts the phone on the bed, 16s-20s: #O man X picks the laptop, … **Summary**: … the man initially walks, then moves around the room and interacts with a phone, … (B is updated.)

Prompt$_{\text{Answer}}$: …, please think step by step, are you able to answer the question {Q}? Please generate a confidence score scale from 1-5.
LLM: No, I do not have enough information to answer the question. (confidence score = 0).
Prompt$_{\text{CreatePlan}}$: Here are some tools: … Please think step by step and produce an information retrieval plan by considering the relationship between the information you need and the tool function. Please output the next action in json format: {'Action': tool call command}.
LLM: The following information retrieval plan has been developed:

t=1

- Step1: Identify the precise moment and location of the phone's last visible appearance.
- Step2: Analyze the phone's movement prior to its disappearance to determine if it was removed from the view, obscured, or other ways.
- Step3: Detect and examine objects in the vicinity of the phone's final appearance to investigate potential causes for its disappearance.

Action: {'tool_name': 'Object Tracking Tool', 'object_name': 'phone', 'frame_range': '12-28'}
NewInfo: … {"frame_id": 19, "object_name": "phone", "object_bbox": [617, 878, 807, 949], "confidence": 0.80}{"frame_id": 20, "object_name": "phone", "object_bbox": [611, 913, 812, 958], "confidence": 0.62}{"frame_id": 21, "object_name": "phone", "object_bbox": null, "confidence": null}… (B is updated.)

Prompt$_{\text{Answer}}$: Please consider all existing information in $\{B\}$ and think step by step to answer the question again.
LLM: No, I do not have enough information to answer the question (confidence score = 0).
Prompt$_{\text{AdjustPlan}}$: …, please think step by step to adjust your information retrieval plan based on the new information and the uncertainty in it.
LLM: The mobile phone remained in a stable position prior to its disappearance, suggesting that it was not removed from the camera's field of view. The cause of its loss can be deduced by analyzing the final location of the phone and the surrounding objects at that moment.

t=2

- Step1: Detect objects in the vicinity of the phone's last visible frame.
- Step2: Consider zooming in to focus on the area where the phone is located for further analysis if objects are concentrated.

Action: { 'tool_name': 'Object Detection Tool ', 'frame_range': '20-21' }.
NewInfo: {"frame_id": 20, "detections": [{"object_name": "chair1", "bbox": [579, 436, 1048, 894], "confidence": 0.54}, {"object_name": "person1", "bbox": [995, 2, 1709, 1065], "confidence": 0.97}, {"object_name": "bed1", "bbox": [190, 167, 1065, 679], "confidence": 0.54}]} {"frame_id": 21, … (B is updated.)

Prompt$_{\text{Answer}}$: Please consider all existing information in $\{B\}$ and think step by step to answer the question again.
LLM: Yes, the answer is the phone is covered by the chair (confidence score = 3).
Prompt$_{\text{AdjustPlan}}$: …, please think step by step to adjust your information retrieval plan based on the new information and the uncertainty in it.
LLM: In the results from the object detection and tracking tool, the positions of the mobile phone and the chair in frame 20 overlap; however, both of them have low confidence scores. To resolve this, it is advisable to use alternative tools to gather additional information for these moments. The image caption tool can be employed to complement the data for these specific frames.

t=3

- Step1: Utilize the image captioning tool to extract relevant events occurring during this period.
- Step2: If uncertainty remains high, consider extending the frame to capture more contextual information.

Action: {'tool_name': 'Image Caption Tool', 'frame_range': '19-21' }
NewInfo: {" frame_id " : 19, " caption " : " A person is placing a laptop on a black chair that already has a smartphone on its seat (confidence score = 0.85), with a neatly made bed in the background (confidence score = 0.90)."} {" frame_id " : 20, " caption " : " A person is placing a laptop onto a black perforated chair that already has a smartphone on it (confidence score = 0.62), in a tidy bedroom with a neatly made bed in the background (confidence score = 0.95). "} {"frame_id": 21, "caption": "A person is moving a black perforated-back chair with a closed white laptop placed on it in a bedroom (confidence score = 0.95). "} (B is updated.)

Prompt$_{\text{Answer}}$: Please consider all existing information in $\{B\}$ and think step by step to answer the question again.
LLM: Although the confidence level for the computer being placed on the chair with the mobile phone on frame 20 is relatively low, the caption for frame 19 demonstrates high confidence and provides similar content. By integrating this information with data from other tools, it can be concluded that the reason is that **the phone was obscured by the laptop** (confidence score = 5).

t=4

frame19 frame20 frame21

…  …

Figure 5: Case study of VideoAgent2. The video and associated question are presented in Fig. 2. Both the popular MLLM Llava-OneVision and the SOTA agent baseline, VideoAgent, fail to provide the correct answer. In contrast, our proposed VideoAgent2 correctly answers the question through three tool calls and four answer assessments. VideoAgent2 leverages the information and uncertainty provided by the tools, enabling the LLM to continuously adjust its information retrieval plan, and make more reliable decisions when synthesizing the final answer.

```
Please think step by step. Are you able to answer the question {Q}?
If you don't think there is enough information to answer the question, please reply as 'No, I do not have enough information to answer the question. (confidence score = 0)'.
If you can answer the question, please reply as 'Yes, the answer is xx, (confidence score = xx)', note that you need to generate a confidence score for your answer, scaled from 1-5."
```

- $prompt_{\text{CreatePlan}}$

```
"To assist you in answering the question more effectively, I have provided some tools.
Below are tool descriptions, notes on using tools, and the call command format:

1. Image Caption Tool
   - Function: Generates captions for specific image frames.
```

```
  - Usage: Specify a single frame index or a range of frames.
  - Return Values: A list of dictionaries, each containing the frame_id and the caption (e.g., {'frame_id': 'xx', 'caption': 'xx'}). A confidence score is provided for each sentence or clause in the caption.

2. Object Detection Tool
  - Function: Identifies all objects within specific image frames and provides their bounding boxes.
  - Usage: Specify a single frame index or a range of frames.
  - Return Values: A list of dictionaries, each containing the frame_id and the detection results (e.g., {'frame_id': 'xx', 'det_info': {'id': 'xx', 'name': 'xx', 'bbox': '[xmin, ymin, xmax, ymax]', 'confidence': 'xx'}}).

3. Image Zoom in and Caption Tool
  - Function: First zoom in on an area of an image frame and then generate a caption.
  - Usage: Specify a single frame index and the bbox of the area you are interested in.
  - Return Values: A list of dictionaries, each containing the frame_id, the bbox of the interested area, and the caption (e.g., {'frame_id': 'xx', 'bbox': 'xx', 'caption': 'xx'}). A confidence score is provided for each sentence or clause in the caption.

4. Image Zoom in and Object Detection Tool
  - Function: First zoom in on an area of an image frame and then detect all objects in the area.
  - Usage: Specify a single frame index and the bbox of the area you are interested in.
  - Return Values: A list of dictionaries, each containing the frame_id, the bbox of the interested area, and the detection results (e.g., {'frame_id': 'xx', 'bbox': 'xx', 'det_info': {'id': 'xx', 'name': 'xx', 'bbox': '[xmin, ymin, xmax, ymax]', 'confidence': 'xx'}}).

5. Object Tracking Tool
  - Function: Provides the bounding box (bbox) of an object in each frame of a video clip.
  - Usage: Specify the object name and the frame range.
  - Return Values: A list of dictionaries, where each dictionary contains the frame id, object name, bbox and confidence (e.g., {'frame_id': 'xx', 'object_name': 'xx', 'bbox': '[xmin, ymin, xmax, ymax]', 'confidence': 'xx'}).

The call command for the Image Caption Tool is:
{'tool_name': 'Image Caption Tool', 'frame_range': 'frame_id' # or 'start frame-end frame'}.

The call command for the Object Detection Tool is:
{'tool_name': 'Object Detection Tool', 'frame_range': 'frame_id' # or 'start frame-end frame'}.
```

```
The call command for the Image Zoom in and Caption Tool is:
{'tool_name': 'Image Zoom in and Caption Tool', 'frame_range
    ↪ ': 'frame_id', 'bbox': '[xmin, ymin, xmax, ymax]'}.

The call command for the Image Zoom in and Object Detection
    ↪ Tool is:
{'tool_name': 'Image Zoom in and Object Detection Tool', '
    ↪ frame_range': 'frame_id', 'bbox': '[xmin, ymin, xmax,
    ↪ ymax]'}.

The call command for the Object Tracking Tool is:
{'tool_name': 'Object Tracking Tool', 'object_name': 'xx', '
    ↪ frame_range': 'frame_id' # or 'start frame-end frame
    ↪ '}.

You are allowed to call the tool multiple times to retrieve
    ↪ the information you need, but only one tool can be
    ↪ called at a time.
Please think step by step and first make an information
    ↪ retrieval plan  to help you gather the useful
    ↪ information.
Consider the relationship between the information you need
    ↪ and the tool function.
Then please output the first action in the following JSON
    ↪ format: {'Action': 'tool call command'}."
```

- $prompt_{\text{AdjustPlan}}$

```
"Your answer is not confident enough.
Please think step by step to adjust your information
    ↪ retrieval plan based on the new information and the
    ↪ uncertainty in it and output  the first action in the
    ↪ following JSON format:
{'Action': 'tool call command'}."
```

D Failure analysis

We have analyzed the failure cases in VideoAgent2 and summarize two main issues (1) Omission of general context information and (2) Limitations of current tools in capturing spatiotemporal information.

1. Omission of general context information: As highlighted in the paper, acquiring context information is a key component of VideoAgent2. This stage must balance performance with computational cost. There are two main challenges:
    - Computational efficiency: Over-focusing on detailed video segments can reduce efficiency, especially for long videos, and may lead to processing irrelevant content without regard to the question.
    - Information integrity: The general context must capture enough information to ensure that critical segments are not missed in subsequent steps.

    Our analysis of failure cases shows that missing important events in the context stage can hinder the LLM's ability to identify relevant intervals. This may lead to retrieval failure or an excessive number of retrieval attempts, eventually exceeding the allowed retrieval limit. To address this, we are exploring improvements such as integrating a variety of context-capturing tools (e.g., object detection-based methods) to enhance coverage.
2. Limitations of current tools in capturing spatiotemporal information: We observed that some spatiotemporal events are difficult to express in text or detect in static frames. For instance, whether someone is picking up or putting down a laptop may depend on subtle motion

details. To address this, we are trying the solution such as extracting features directly at the visual (rather than textual) level, and incorporating tools that analyze video clips for spatiotemporal features.