arXiv:2511.19524v2 [cs.CV] 4 Mar 2026

VideoChat-M1: Collaborative Policy Planning for Video Understanding via Multi-Agent Reinforcement Learning

Boyu Chen[1,2,3 *], Zikang Wang[4,6 *], Zhengrong Yue[6 *], Kainan Yan[1,2 *], Chenyun Yu[5 †], Yi Huang[3 †], Zijun Liu[8], Yafei Wen[3], Xiaoxin Chen[3], Yang Liu[4,7,8], Peng Li[7,8 †], Yali Wang[1,4 †]
[1]Shenzhen Key Lab of Computer Vision and Pattern Recognition, Shenzhen Institutes of Advanced Technology, Chinese Academy of Sciences
[2]School of Artificial Intelligence, University of Chinese Academy of Sciences
[3]VIVO AI Lab, [4]Shanghai Artificial Intelligence Laboratory,
[5]Shenzhen Campus of Sun Yat-sen University, [6]Shanghai Jiao Tong University,
[7]Institute for AI Industry Research (AIR), Tsinghua University,
[8]Dept. of Comp. Sci. & Tech., Institute for AI, Tsinghua University

Abstract

*Most of the multi-agent video understanding frameworks adopt static and non-learnable tool invocation mechanisms, which limit the discovery of diverse clues essential for robust perception and reasoning regarding temporally or spatially complex videos. To address this challenge, we propose a novel **M**ulti-agent system for video understanding, namely **VideoChat-M1**. Instead of using a single or fixed policy, we adopt a distinct Collaborative Policy Planning (CPP) paradigm with multiple policy agents, which comprises three key processes. (1) Policy Generation: Each agent generates its unique tool invocation policy tailored to the user's query; (2) Policy Execution: Each agent sequentially invokes relevant tools to execute its policy and explore the video content; (3) Policy Communication: During the intermediate stages of policy execution, agents interact with one another to update their respective policies. Through this collaborative framework, all agents work in tandem, dynamically refining their preferred policies based on contextual insights from peers. Moreover, we equip our CPP paradigm with Multi-Agent Reinforcement Learning (MARL). Consequently, policy agents can be jointly optimized to enhance the performance, guided by both the final answer reward and intermediate collaborative process feedback. Extensive experiments demonstrate that VideoChat-M1 achieves SOTA performance across eight benchmarks on four tasks. Notably, on LongVideoBench, our method outperforms Gemini 2.5 pro by 3.6% and GPT-4o by 15.6%.*

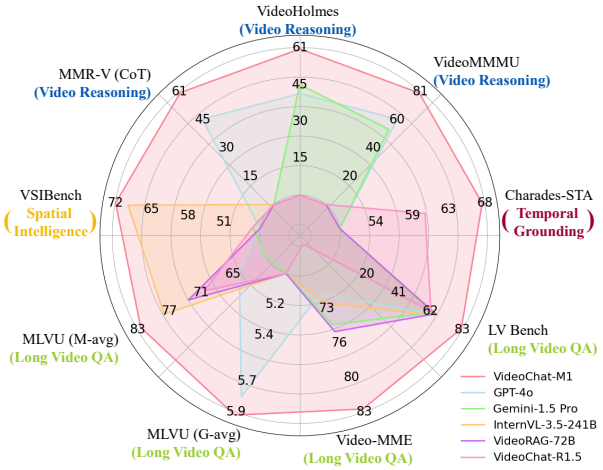


Figure 1. **Comparison with SOTA.** VideoChat-M1 outperforms closed-source models (including GPT-4o) and open-source models (including InternVL-3.5-241B) in mainstream video tasks.

1. Introduction

Video understanding is a critical topic in computer vision [27, 47, 50]. Recent advancements in this field have been predominantly driven by Multimodal Large Language Models (MLLMs) [1, 7, 16, 24, 25, 51, 59, 68]. However, most MLLMs excel at processing short video clips while struggling to understand videos with long temporal contexts and/or complex spatial structures [1, 16, 24, 25, 51, 59]. Recently, agent-based frameworks have shown significant potential to overcome these limitations in video understanding [6, 19, 32, 61, 64, 65, 75]. Instead of directly feeding massive video frames into MLLMs, these frameworks enhance video understanding by invoking various tools to extract key video clues, either in an off-the-shelf [19, 64] or iterative [61, 65, 72, 75, 79, 85] manner. Nevertheless, the

*Equal contribution.
†Equal corresponding author.

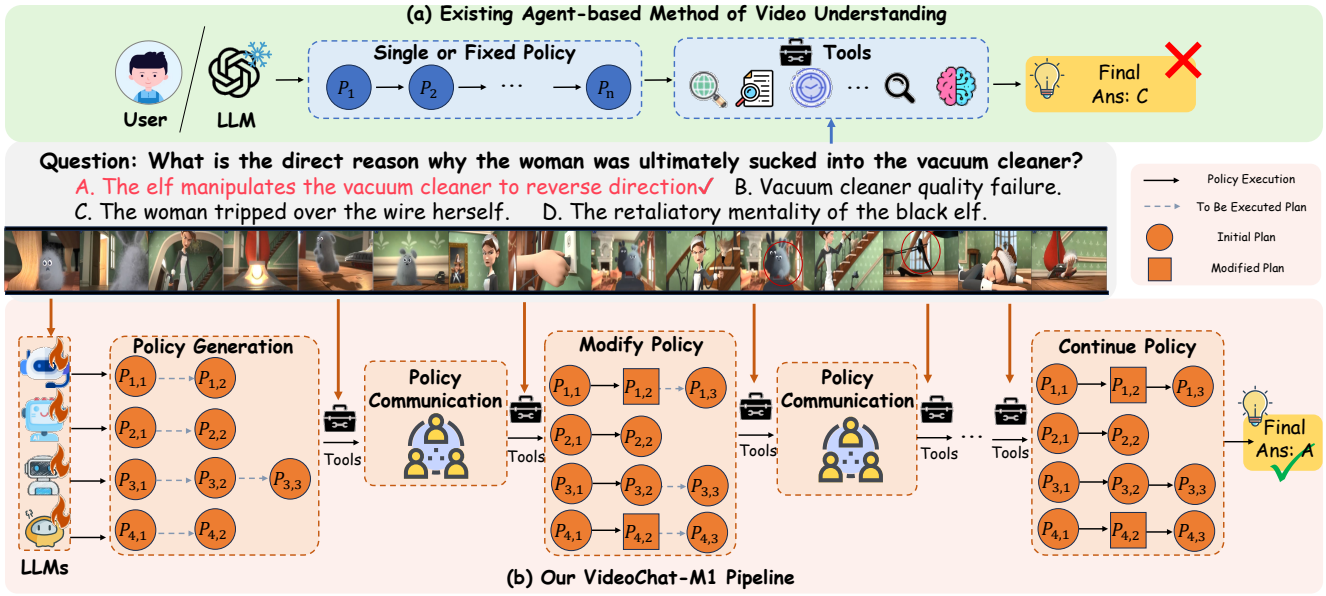


Figure 2. Architecture and Working Mode Comparison (Existing Agent-based Method vs. Our VideoChat-M1). While prior methods rely on a fixed policy, VideoChat-M1 introduces a collaborative multi-agent policy planning pipeline that generates, executes, communicates and refines plans iteratively, enabling more adaptive and accurate long-video reasoning.

tool invocation policy in existing agent-based frameworks is straightforward and fixed as shown in Fig 2 (a), i.e., they adhere to pre-defined rules for tool selection and invocation during video understanding, without adaptive learning. Such ad-hoc policies inherently prevent them from identifying, tracking, and summarizing rich clues on diverse temporal scales, leading to suboptimal perception and reasoning capabilities for complex videos [1, 16, 24, 25, 51, 59].

To address this core challenge, we introduce **VideoChat-M1**, a novel **M**ulti-agent system for multiple video understanding tasks. Unlike frameworks with predefined tool policies, VideoChat-M1 introduces a Collaborative Policy Planning (CPP) paradigm where agents autonomously generate and adaptively update their policies for better video understanding. As shown in Fig 2(b), CPP involves three core stages: policy generation (where agents formulate tool invocation strategies), policy execution, and policy communication. Subsequently, during policy execution process, each agent implements its policy progressively by using relevant tools to discover video clues. To boost policy effectiveness and robustness, we further integrate policy communication into the execution process: after each step of execution, each agent receives video clues from peers. Through this concise communication, agents synthesize contextual information from others and decide whether to refine their original policies into more optimal ones. Via the CPP paradigm, all agents in our VideoChat-M1 framework collaborate to generate diversified tool-invocation policies, enabling the extraction of richer video clues to deeply understand complex queries or questions in videos.

To ensure the robustness and effectiveness of VideoChat-M1, we equip the CPP paradigm with a concise Multi-Agent Reinforcement Learning (MARL) method. To our best knowledge, this is the first policy learning framework that supports joint RL training among multiple agents for video understanding. Specifically, we treat each agent as a policy model and design three types of rewards for their joint optimization. First, we use each agent's query answer and format constraints as rewards to encourage correct responses and penalize incorrect ones. More critically, we employ an LLM as the reward model to evaluate the intermediate collaboration process, i.e., rewarding agents that generate superior policies via the CPP paradigm and penalizing those with inferior ones. Guided by these rewards, we adapt Group Relative Policy Optimization (GRPO) to optimize the entire VideoChat-M1 agent group, creating a more effective policy planning to improve video understanding.

Finally, we evaluated VideoChat-M1 on 8 challenging benchmarks spanning long video QA, video reasoning, spatial intelligence, and temporal grounding. Extensive experiments demonstrate that our VideoChat-M1 method achieves exceptional performance across all tasks, outperforming both closed-source and open-source baselines. Notably, For the long video question answering task on LongVideoBench [67], we outperform GPT-4o [51] by 15.6%. For video reasoning on VideoMMMU [31], our 37B agent group delivers results comparable to Qwen3-VL-235B [1] while using only 15% of the model parameters. In the spatial intelligence task on VSIBench [71], our model exceeds Gemini 1.5 Pro [55] by 26.5%. For the temporal grounding task on Charades-STA [23], we achieve a 3.0% improvement over Seed 1.5VL [24]. Our key contributions are summarized as follows:

- We propose VideoChat-M1, the first multi-agent frame-

work for video understanding that replaces the conventional single, fixed policy with a Collaborative Policy Planning (CPP) paradigm, enabling agents to dynamically generate and adapt tool-use strategies through multi-agent policy communication.
- We introduce a pioneering Multi-Agent Reinforcement Learning (MARL) method to optimize the collaborative process. It uniquely employs a hybrid reward system that evaluates both the final answer accuracy and the intermediate quality of multi-agent collaboration.
- Extensive experiments show that VideoChat-M1 achieves SOTA performance on eight challenging benchmarks. It significantly outperforms leading models such as GPT-4o and Gemini 1.5 Pro, while exhibiting superior parameter efficiency compared to substantially larger models.

2. Related Work

Video Understanding. Understanding videos is a major challenge for Vision-Language Models (VLMs). Early efforts improved single-model performance through architecture scaling [2, 13, 37, 82], context window extension [43, 56, 60], token compression [41, 54, 78], or reinforcement learning [12, 40]. However, these approaches struggle with precise retrieval because a single agent cannot easily manage perception, retrieval, and synthesis simultaneously To address this issue, subsequent work has equipped a single agent with retrieval [83], memory [19, 26], and search tools [34, 64, 74], but their general-purpose design limits effective integration and reasoning. Unlike systems such as LVAgent [6] that rely on static, untrained collaboration, we introduce a trainable multi-agent framework for dynamic adaptability across diverse video tasks.

Multi-Agent Reinforcement Learning. Recent advancements in LLMs have driven the development of multi-agent systems, which follow two paradigms. First, training-free systems (e.g., CAMEL [36] and MetaGPT [29]) rely on engineered logic and fixed roles. However, their static text-centric design often fail to handle dynamic multi-modal tasks like long-form video understanding. Second, RL-based paradigm train collaborative policies by optimizing agent behaviors [3–5, 10, 11, 42, 52, 63, 65, 73, 88] or interaction architectures [49, 76, 77]. Despite growing sophistication [18, 22, 48, 57, 66], these methods remain confined to unimodal text domains, overlooking temporal and perceptual challenges specific to video. Existing RL methods struggle to co-train heterogeneous agents, limiting their synergy. To address this, we draw inspiration from successful multi-agent GUI pipelines [28] to introduce VideoChat-M1, a novel framework that enables joint training of diverse agents for complex multi-modal tasks.

3. Methodology

This section details the VideoChat-M1 framework, introducing its Collaborative Policy Planning (CPP) paradigm, followed by the Multi-Agent Reinforcement Learning (MARL) method that enhances its effectiveness.

3.1. Collaborative Policy Planning Pipeline (CPP)

As noted earlier, existing agent-based frameworks primarily adopt a single and fixed tool invocation policy, resulting in suboptimal performance on long-form and complex videos. Therefore, we propose a distinct CPP paradigm for enhancing video understanding, where multiple agents collaborate to dynamically refine tool-invocation policies. For clarity, let $\mathcal{Q}$ denote a user query for a video $\mathcal{V}$ (a question about specific video content). Our CPP paradigm comprises a set of policy agents $\mathcal{G} = \{\mathcal{G}_i\}$, a set of video perception tools $\mathcal{T} = \{\mathcal{T}_j\}$, and a shared memory buffer $\mathcal{M} = \{\mathcal{M}_i\}$. This buffer records key historical information from all agents (e.g., initial policy plan, intermediate answer) and supports policy updates through agent communication. Further details are provided in Appendix A.2. Fig 3 illustrates the CPP workflow, an iterative collaborative framework where each policy agent autonomously executes three core phases: policy generation, policy execution, and policy communication. During each execution step, all agents share the group's state, key video clues, and decision-making information via the shared memory buffer, dynamically optimizing tool-invocation decisions for the next step. Through the CPP pipeline, agents collaborate to generate diversified tool-invocation plans, extracting richer video clues to deeply understand complex video queries.

3.1.1. Policy Generation

Our CPP pipeline begins with a policy generation stage, where the core video understanding task is sequentially decomposed into smaller and manageable sub-tasks. Specifically, each agent $\mathcal{G}_i$ generates an initial policy, which specifies an explicit solution to address the user query $\mathcal{Q}$ by invoking a sequence of tools from toolkit $\mathcal{T}$. This process is formally defined as:

$$\mathcal{P}_i = \mathcal{G}_i(\mathcal{Q},\ \mathcal{T}), \tag{1}$$

where $\mathcal{P}_i = \{\mathcal{P}_{i,1} \rightarrow \mathcal{P}_{i,2} \rightarrow ... \rightarrow \mathcal{P}_{i,N}\}$ denotes the initial policy of agent $\mathcal{G}_i$, and $\mathcal{P}_{i,N}$ is the N-th policy step that specifies a certain tool in $\mathcal{T}$ for analyzing the input video.

3.1.2. Policy Execution

After generating the initial policy $\mathcal{P}_i$, agent $\mathcal{G}_i$ proceeds with execution. Specifically, at the n-th policy step, the agent utilizes $\mathcal{P}_{i,n}$ to retrieve the corresponding tool from the toolkit $\mathcal{T}$. It then employs this tool to analyze the input video $\mathcal{V}$ to obtain the intermediate answer at the step n, based on the $(n-1)$-th step answer:

$$\mathcal{A}_{i,n} = \mathcal{P}_{i,n}(\mathcal{V},\ \mathcal{T},\ \mathcal{A}_{i,n-1}), \tag{2}$$

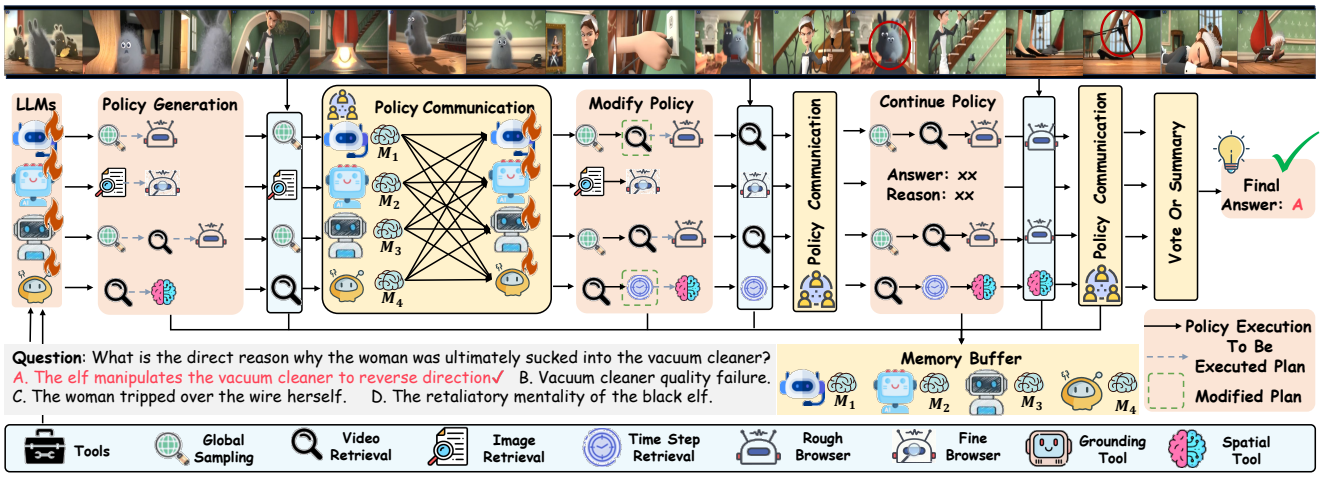


Figure 3. The workflow of Collaborative Policy Planning (CPP) in the Reasoning Phase. Multiple agents independently generate initial plans, communicate to exchange reasoning states, and iteratively refine their policies using different tools. Through repeated rounds of communication and plan updates, the agents collectively vote or summarize to produce a reliable final answer.

where $\mathcal{A}_{i,n}$ and $\mathcal{A}_{i,n-1}$ denote the n-th and $(n-1)$-th step answers for agent $\mathcal{G}_i$, respectively. This process iterates until obtaining the final answer of user's query, i.e., $\mathcal{A}_i = \{\mathcal{A}_{i,1} \rightarrow \mathcal{A}_{i,2} \rightarrow ... \rightarrow \mathcal{A}_{i,N}\}$.

However, the initial policies of agents may lack reliability, executing such suboptimal policies may fail to generate satisfactory results. Hence, we propose a policy communication stage integrated with policy execution, enabling dynamic updates to tool-invocation policies as needed.

3.1.3. Policy Communication

To enhance the robustness of tool-invocation policies for answering the user query, we introduce policy communication with agent group collaboration. After executing each step of their initial policies, the agent team $\mathcal{G}$ generates intermediate results $\mathcal{A}$, and stores them in a shared memory $\mathcal{M}$. Subsequently, each agent references its initial policy and the team's intermediate memory to determine whether to update its policies for subsequent steps, formulated as:

$$\mathcal{P}_i' = \mathcal{G}_i(\mathcal{Q}, \mathcal{T}, \mathcal{M}, \mathcal{P}_i), \tag{3}$$

where $\mathcal{P}_i'$ denotes the updated policy of agent $\mathcal{G}_i$. If the current policy $\mathcal{P}_i$ remains optimal, $\mathcal{P}_i'$ stays unchanged, and the agent continues tool invocation based on the next step of $\mathcal{P}_i$. Otherwise, the agent revises $\mathcal{P}_i$ as $\mathcal{P}_i'$, and executes the next steps of the updated policies.

Moreover, policy communication and execution are performed iteratively. This allows each agent to effectively leverage the team's intermediate results as historical experience and refine its own policy through multi-round communication during policy execution. Once all tools are executed, each agent summarizes its prior results to generate an answer. The final answer is determined by the query type: multiple-choice questions are resolved by majority voting, while open-ended and temporal grounding queries are consolidated by a designated agent, which is the best-performing model in the group (Qwen3-8B [70]).

3.2. Multi-Agent Reinforcement Learning (MARL)

Unlike previous multi-agent video understanding frameworks (no training), we propose MARL to guide the team of policy agents, enhancing VideoChat-M1's adaptability and collaborative capabilities. To the best of our knowledge, this is the first multi-agent policy learning framework designed to tackle complex video understanding tasks. As a warm-up, the supervised fine-tuning (SFT) stage equips each agent with basic abilities to produce a high-quality initial policy plan. Subsequently, the MARL pipeline trains the agent team to achieve effective collaboration.

3.2.1. Policy SFT

We first construct a high-quality policy set from open-source video datasets (see Appendix A.1) for policy SFT. The input is the user query for the training video, and the output is the policy plan for answering the query. Notably, since these open-source datasets lack pre-existing policy plans, we leverage our CPP paradigm with a high-performance team (i.e., GPT-4o and DeepSeek-R1) to automatically annotate policy plans for each training video. To guarantee annotation quality, we curate the policy data based on two core criteria: first, selecting annotated policy plans that yield the correct answer to the user's query for the training video; second, choosing plans that can be executed successfully to obtain the final answer without any policy modification. These criteria ensure the supervision signal comprises effective and efficient policy plans. Using this policy plan dataset, we fine-tune each agent in the team using the cross-entropy loss by maximizing the likelihood of generating the ground-truth plan. This SFT phase enables each agent to master the basic capacity to generate a preferred policy plan in a structured output format, laying

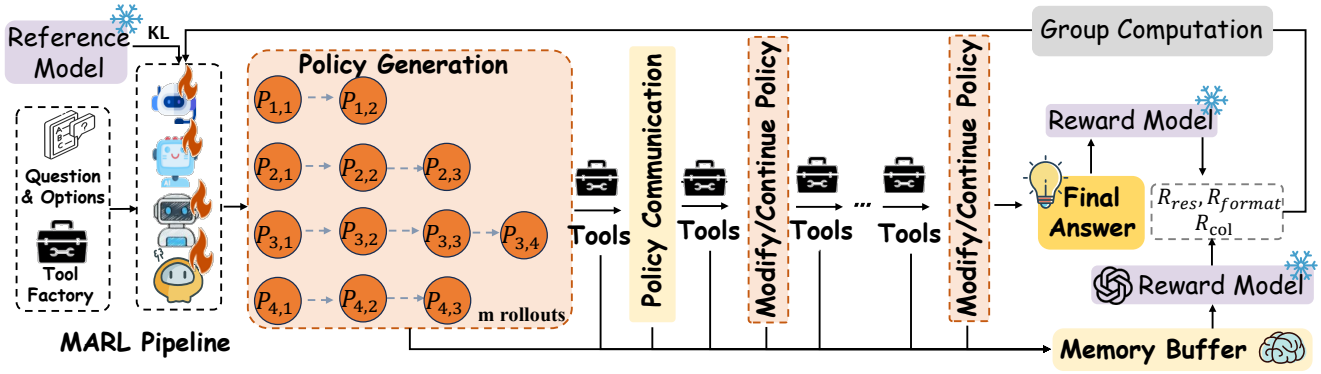


Figure 4. Training the Agent Group using Our Multi-Agent Reinforcement Learning (MARL) Method. Agents generate policies, communicate, and iteratively refine them with tool feedback, while reward and reference models guide stable joint optimization.

the foundation for collaborative learning in MARL.

3.2.2. MARL

To foster effective team collaboration, we introduce the MARL framework as shown in Fig 4 to further optimize the agent group. Specifically, we leverage three types of reward for this purpose, i.e., $\mathcal{R} = \mathcal{R}_{res} + \mathcal{R}_{format} + \mathcal{R}_{col}$.

Result Reward R_{res}: After completing our CPP pipeline, all agents generate their respective answers. We assign a positive reward for correct final answers and a negative penalty for incorrect ones.

Format Reward R_{format}: To ensure procedural reliability and system compatibility, R_{format} incentivizes syntactically correct actions. It grants rewards for well-formed, executable outputs (e.g., parsable plans, valid tool calls) and imposes penalties for format-related errors.

Collaboration Reward R_{col}: To encourage effective agent collaboration beyond the final outcome, we evaluate the intermediate planning process recorded in each agent's memory buffer. We leverage GPT-4o as an external evaluator to assess the holistic quality of the trajectory, including plan feasibility, tool call appropriateness, and step management soundness. To mitigate the inherent stochasticity of LLM-based scoring and ensure a stable learning signal, we constrain the evaluator's output to a binary reward: 1 for coherent trajectories and 0 otherwise (see Appendix A.3 for the prompt). Furthermore, to explicitly promote concise strategies and prevent reward hacking through lengthy planning, we apply a strong penalty to trajectories exceeding five tool calls. Since each agent's memory is influenced by team communication, this reward mechanism incentivizes the entire group to cooperate on developing coherent and efficient policy plans.

After specifying the reward formulation, we train the agent team using Group Relative Policy Optimization (GRPO) [25]. Specifically, each agent generates K policy plans, producing K candidate final answer, i.e., $o = \{o_1, o_2, ..., o_K\}$. The advantage $A_R(o_k)$ of each output $o_k \in o$ is then computed by standardizing its reward against the statistics of all outputs in the group:

$$A_R^{(k)} = \frac{R(o_k) - \text{mean}(\{R(o_1), ..., R(o_K)\})}{\text{std}(\{R(o_1), ..., R(o_K)\})} \quad (4)$$

Finally, we optimize the model parameters of each agent by maximizing the GRPO objective function:

$$\max_{\pi_\theta} \mathbb{E}_{o\sim\pi_{\theta_{\text{old}}}} \Big[\sum_{k=1}^{K} \frac{\pi_\theta(o_k)}{\pi_{\theta_{\text{old}}}(o_k)} \cdot A_R^{(k)} - \beta\, \mathrm{D}_{\text{KL}}\Big(\pi_\theta \,\|\, \pi_{\text{ref}}\Big) \Big]$$

The GRPO objective function balances two components: a reward-seeking term that encourages high-scoring responses, and a KL-divergence penalty that regularizes the policy. This penalty, weighted by the coefficient β, constrains the optimized policy π_θ of agent $\mathcal{G}_i$ to remain close to the reference policy π_{ref}, ensuring training stability. This MARL encourages each agent to refine its policies and collaborate flexibly to answer user queries about the video.

4. Experiments

Datasets. We conducted evaluations on 8 video understanding benchmarks described as follows: MLVU-Dev [84] includes 2,174 multiple-choice questions and 417 open-ended questions with videos averaging 930 seconds. LongVideoBench [67] provides 1,337 multi-domain QA pairs and videos averaging 473 seconds. VSI-Bench [71] focuses on spatial-temporal reasoning with 2,500 QA pairs that require fine-grained inference of object interactions and temporal causality. VideoMME [21] offers 900 videos (11s-1h) with 2,700 QA pairs, and MMR-V [87] consists of 1257 QA pairs in test set, emphasizing cross-modal and multi-step reasoning. VideoMMMU [30] provides 900 QA pairs for video reasoning, while Video-Holmes [14] comprises 270 suspense films and 1,837 QA pairs to evaluate complex reasoning via cross-temporal visual clue integration. Charades-STA [23] is a large-scale dataset for evaluating temporal grounding tasks with 4233 QA pairs.

Metrics. In Tab 1, accuracy is adopted as the primary evaluation metric across all benchmarks. For MMR-V, the CoT column reflects multi-step and compositional reasoning ability by measuring performance under Chain-of-

Model	Long Video QA						Video Reasoning			Spatial Intelligence				Temporal Grounding
	LongVideo Bench	Video-MME			MLVU		Video Holmes	Video MMMU	MMR-V	VSIBench				Charades
		M	L	Avg	M-avg	G-avg			CoT	Dist	Dir	Order	Avg	m-IOU
Closed-source Large-sized MLLM														
Gemini 2.5 Pro [16]	78.7	-	-	84.3	-	-	-	83.6	-	-	-	-	-	-
Gemini 1.5 Pro [55]	64.0	74.3	67.4	75.0	-	-	45.7	53.9	-	51.3	46.3	34.6	45.4	-
GPT-5-thinking	-	-	-	-	-	-	-	84.6	-	-	-	-	-	-
GPT-4o [51]	66.7	70.3	65.3	71.9	64.6	5.80	42.0	61.2	46.1	37.0	41.3	28.5	34.0	-
OpenAI O3	-	-	-	-	-	-	-	83.3	-	-	-	-	-	-
Seed 1.5VL [24]	74.0	-	-	77.9	82.1	-	-	81.4	-	-	-	-	-	64.7
Open-source Large-sized MLLM														
InternVL-3.5-241B [59]	67.1	-	-	72.9	78.2	-	-	-	-	-	-	-	69.5	-
Qwen3-VL-235B-Instruct [1]	-	-	-	79.2	84.3	-	-	74.7	-	-	-	-	62.6	64.8
Qwen3-VL-235B-Thinking [1]	-	-	-	79.0	83.8	-	-	80.0	-	-	-	-	-	63.5
Qwen2.5-VL-72B [1]	60.7	-	-	73.3	74.6	-	-	-	40.4	-	-	-	-	50.9
Qwen2-VL-72B [58]	-	71.3	62.2	71.2	-	-	-	-	40.4	-	-	-	36.1	-
LLAVA-Video-72B [82]	64.9	68.9	61.5	70.6	-	-	-	49.7	-	42.4	36.7	48.6	40.9	-
LLaVA-ov-72B [33]	61.3	62.2	60.0	66.3	68.0	-	-	48.3	-	42.5	39.9	44.6	40.2	-
VideoLLaMA2-72B [15]	-	59.9	57.6	62.4	45.6	3.78	-	-	-	-	-	-	-	-
InternVL-2.5-78B [13]	63.6	70.9	62.6	72.1	75.7	-	-	-	-	-	-	-	-	-
InternVL-3-78B [86]	65.7	-	-	72.7	79.5	-	-	-	-	55.9	39.5	54.5	48.4	-
InternVL-3.5-78B [59]	65.7	-	-	70.9	77.0	-	-	-	-	-	-	-	66.3	-
Aria-28B [35]	64.2	67.0	58.8	67.6	72.3	5.02	-	50.8	-	-	-	-	-	-
Oryx-34B [44]	-	65.3	59.3	67.3	70.6	-	-	-	-	-	-	-	-	-
Open-source Medium-sized MLLM														
VideoXL2-8B [53]	61.0	-	-	66.6	74.8	-	-	-	-	-	-	-	-	54.2
Video-R1-7B [20]	-	-	-	61.4	-	-	36.5	-	-	-	-	-	37.1	-
VideoChat-Flash-7B [39]	64.7	-	55.4	65.3	74.7	-	-	-	-	-	-	-	-	48.0
VideoChat2-7B [38]	39.3	37.0	33.2	39.5	47.9	3.81	-	-	-	-	-	-	-	-
VideoChat-R1-7B [40]	-	-	-	-	-	-	33.0	-	-	-	-	-	-	-
VideoChat-R1.5-7B [69]	62.6	-	-	67.1	70.9	-	-	51.4	-	-	-	-	-	60.6
InternVideo2.5-7B [62]	60.6	-	-	65.1	72.8	-	-	-	-	-	-	-	-	-
InternVL-2.5-8B [13]	60.0	-	-	64.2	68.9	-	23.6	-	-	-	-	-	-	-
InternVL-3-8B [86]	58.8	-	-	66.3	71.4	-	32.3	-	32.9	48.3	36.4	35.4	42.1	-
InternVL-3.5-8B [59]	62.1	-	-	66.0	70.2	-	-	-	-	-	-	-	56.3	-
Qwen2-VL-7B [58]	55.6	-	-	63.3	-	-	27.8	-	32.4	-	-	-	-	-
Qwen2.5-VL-7B [1]	56.0	-	-	65.1	70.2	-	-	-	-	-	-	-	35.9	43.6
Qwen3-VL-8B-Instruct [1]	-	-	-	71.4	78.1	-	-	65.3	-	-	-	-	59.4	55.5
LongVA-7B [80]	51.3	50.4	46.2	52.6	-	4.33	-	24.0	-	33.1	43.3	15.7	29.2	-
LongVILA-7B [68]	57.1	58.3	53.0	60.1	-	-	-	-	-	-	-	-	-	-
LongRL-7B [12]	58.1	63.2	55.2	65.1	-	-	-	-	-	-	-	-	-	-
LLaVA-Video-7B [82]	58.2	-	-	63.3	70.8	3.84	-	36.1	17.6	43.5	42.4	30.6	35.6	-
Eagle-2.5-8B [7]	66.4	-	-	72.4	77.6	-	-	-	-	-	-	-	-	65.9
ShareGPT4Video-8B [8]	39.7	36.3	35.0	39.9	46.4	3.77	-	-	-	-	-	-	-	-
Agent-based Methods														
DeepVideoDiscovery [81]	71.6	-	67.3	-	-	-	-	-	-	-	-	-	-	-
DrVideo [46]	-	-	-	51.7	-	-	-	-	-	-	-	-	-	-
ReAgent-V-72B [85]	-	72.3	72.9	75.1	74.2	-	-	-	-	-	-	-	-	-
VCA [75]	41.3	-	-	-	-	-	-	-	-	-	-	-	-	-
VITAL-7B [79]	-	-	54.0	64.1	-	-	-	-	-	-	-	-	-	59.9
VideoChat-A1 [65]	65.4	72.8	65.0	72.9	76.2	-	-	-	-	-	-	-	-	-
VideoExplorer-14B [72]	-	-	-	-	55.4	-	-	-	-	-	-	-	-	-
VideoExplorer-39B [72]	-	-	-	-	58.6	-	-	-	-	-	-	-	-	-
VideoRAG-72B [32]	65.4	72.9	73.1	75.7	73.8	-	-	-	-	-	-	-	-	-
VideoChat-M1 (37B)	**82.3**	**84.2**	**76.7**	**83.2**	**83.4**	**5.92**	**60.5**	**80.0**	**60.4**	**88.3**	**70.8**	**66.7**	**71.9**	**67.7**

Table 1. Algorithm Comparison. Our VideoChat-M1 results are bolded, and the best results of each group of methods are marked in blue.

Thought reasoning. In MLVU, M-avg and G-avg stand for the arithmetic and geometric mean accuracy across multiple sub-tasks, respectively. For Video-MME, S/M/L correspond to short-, medium-, and long-duration videos. In VSI-Bench, Dist, Dir, and Order denote reasoning categories for spatial distance, direction, route and temporal order, with Avg representing the overall accuracy. For Charades, we report the mean Intersection-over-Union between predicted and ground-truth temporal segments.

Implementation Details. For our training and testing, we utilized a setup of eight A100 80G GPUs. The learning rates for SFT and MARL was set to 1e-6 and 1e-7. We performed one epoch of SFT with batch size 32 for each agent on our collected dataset. The best performance was achieved with 200 steps of Multi-Agent Reinforcement Learning (MARL) with 4 rollouts and 8 batch size. To enhance the gener-

Model	Frames	Inference Time	LongVideo Bench	Video MME
Qwen2-VL-72B[58]	568	90.5s	55.6	71.2
GPT-4o[51]	384	153.6s	66.7	71.9
Gemini-1.5-Pro[55]	568	227.2s	64.0	75.0
VideoChat-M1	**69.9**	**19.8s**	**82.3**	**83.2**

Table 2. Average Frame Number and Inference Latency.

Num Agents	Qwen 2.5-3B	Qwen 2.5-7B	Qwen 3-4B	Qwen 3-8B	Video Holmes	LongVideo Bench
1	✓				27.8	59.2
		✓			29.9	61.1
			✓		28.9	60.4
				✓	31.2	61.9
2	✓	✓			41.8	66.8
	✓		✓		41.4	65.9
	✓			✓	42.3	67.2
		✓	✓		42.4	67.1
		✓		✓	43.5	67.9
			✓	✓	42.9	67.2
3	✓	✓	✓		54.8	77.2
	✓	✓		✓	55.3	78.2
	✓		✓	✓	55.1	77.8
		✓	✓	✓	55.9	78.9
4	✓	✓	✓	✓	**60.5**	**82.3**

Table 3. Effects of Agent Group Composition and Scale.

alization of collaboration and avoid co-adaptation, we apply agent dropout. At each training step, a random DAG is sampled from the fully connected agent graph to define the communication topology. This dynamic structure encourages agents to develop robust and flexible communication strategies. Agent Teams for each task, visualizations and further details are provided in Appendix A.6.

4.1. Comparison with SOTA

Performance Comparison. As shown in Tab 1, on models under the 80B scale, we achieved SOTA on 8 datasets. Our VideoChat-M1 approach achieves SOTA on LongVideoBench, outperforming Gemini 2.5 Pro and GPT-4o by 3.6% and 15.6%, respectively. It also achieves SOTA performance on the Video-Holmes and MMR-V benchmarks with gains of 14.8% and 14.3%. In specialized tasks, our model also achieves the best performance, with a 2.4% improvement on the VSIBench spatial intelligence task and a 1.8% lead on the Charades Temporal Grounding task. Notably, our efficient 37B model delivers performance comparable to much larger models (such as the Qwen3-VL-235B, Gemini 2.5 pro) on the Video-MME, MLVU, and VideoMMMU benchmarks. Our method uses a CPP mechanism for task decomposition and Multi-Agent Reinforcement Learning (MARL) to enhance cooperation and communication, boosting the group's collective effectiveness.

Efficiency Comparison. From Tab 2, we observe that VideoChat-M1 uses only 69.9 frames per video, accounting for 12.3%∼18.2% of other models. Meanwhile, its average inference time is 19.8s, which is merely 8.7%∼21.9% of the baselines. Notably, despite the significantly reduced computational cost, VideoChat-M1 still achieves top scores on LongVideoBench (82.3%) and VideoMME (83.2%), highlighting its superior efficiency–performance trade-off.

Num Agents	Qwen 2.5-3B	Qwen 2.5-7B	Qwen 3-4B	Qwen 3-8B	Video Holmes	LongVideo Bench
4	× 2		× 2		55.9	79.3
		× 2		× 2	**58.8**	**80.9**
	× 2	× 2			56.0	79.4
			× 2	× 2	56.2	79.7
		× 1		× 3	57.4	80.5
			× 1	× 3	57.2	80.1
				× 4	55.8	79.2

Table 4. Impact of Architectural Diversity in the 4-Agent Group.

Agent Group	Video Holmes	LongVideo Bench
1× GPT-4o [51] + 1× DeepseekR1 [25]	51.6	71.8
2× GPT-4o [51] + 2× DeepseekR1 [25]	56.2	75.9
4× Deepseek-R1 [25]	51.8	71.4
4× GPT-4o [51]	52.7	72.9
VideoChat-M1	**60.5**	**82.3**

Table 5. Comparison with Foundation LLM Agent Groups.

R_{format}	R_{col}	R_{res}	Agent Dropout	Video Holmes	LongVideo Bench
✓	✓	✗	✓	32.4	63.8
✓	✗	✓	✓	59.4	81.1
✗	✓	✓	✓	60.2	82.0
✓	✓	✓	✗	58.5	79.9
✓	✓	✓	✓	**60.5**	**82.3**

Table 6. Ablation on Components of MARL.

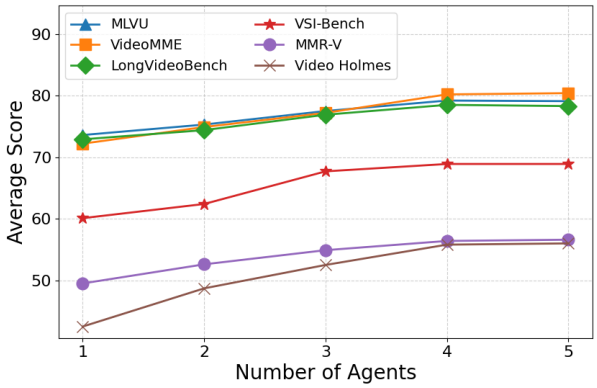


Figure 5. Effects of the Number of Homogeneous Agents.

4.2. Ablation Studies

Effects of the Number of Homogeneous Agents. To investigate how the number of agents affects performance, we conducted experiments using the best-performing Qwen3-8B architecture. As shown in Fig 5, performance improves steadily as the number of agents increases from one to four. However, further increasing the number beyond four leads to performance saturation, with negligible additional gains.

SFT	MARL	Video Holmes	LongVideo Bench
✗	✗	52.1	69.3
✓	✗	55.2	75.9
✗	✓	57.9	80.2
✓	✓	**60.5**	**82.3**

Table 7. Ablation on SFT and RFT.

Full Finetune	LoRA	Video Holmes	LongVideo Bench
✗	✗	55.2	75.9
✗	✓	59.4	81.2
✓	✗	**60.5**	**82.3**

Table 8. Different tuning methods.

Decision Methods	Video Holmes	LongVideo Bench
Best Score	59.9	81.2
Decide by Agent	60.2	81.6
Vote	**60.5**	**82.3**

Table 9. Different discussion mechanisms.

Effects of Agent Group Composition and Scale. To investigate the influence of agent group composition and scale, we conducted experiments with diverse configurations (Tab 3). Our findings reveal two key trends: first, performance consistently improves as the total number of agents increases. Second, for groups of the same size, those with a larger parameter capacity achieve superior results.

Impact of Architectural Diversity within Agent Groups. As shown in Tab 4, we further explore the impact of architectural diversity within a 4-agent setup. We consider configurations where some agents share identical architectures against groups composed of entirely distinct agents. Experimental results indicate that structural redundancy among agents reduces discussion diversity, leading to diminished collaborative gains compared to fully heterogeneous groups. Thus, even with a smaller overall parameter count, our CPP paradigm, when applied to diverse agent groups, enables greater performance improvements than cooperation among homogeneous agents.

Comparison with Foundation LLM Agent Groups. Tab 5 reports the performance of untrained close-sourced foundation LLM teams following the same CPP protocol. Two GPT-4o and two DeepSeek-R1 agents achieve 56.2 and 75.9, respectively, while homogeneous teams of four GPT-4o or four DeepSeek-R1 remain below 53 and 73. VideoChat-M1, trained with MARL, outperforms them by at least 4.3 and 6.4 points. The gap verifies that collaborative fine-tuning injects task-specific coordination patterns that even stronger proprietary models fail to discover via zero-shot reasoning alone.

Ablation Study on Key Components of MARL. Tab 6 evaluates the individual contributions of the reward components and agent dropout. Removing the process reward drops performance by one point on both benchmarks, while omitting the format reward causes a similar degradation. Disabling agent dropout incurs a larger penalty of two points, indicating that dynamic topology is the most critical regularizer. The full configuration yields the highest scores, confirming that dense process feedback and stochastic communication are both necessary for optimal collaborative policy learning.

Ablation on SFT and RFT. Tab 7 disentangles the contributions of supervised fine-tuning (SFT) and collaborative reinforcement learning (RFT). The foundation model without either stage achieves 52.1 and 69.3. Applying SFT alone boosts scores to 55.2 and 75.9, while RFT alone reaches 57.9 and 80.2. The full pipeline, which first establishes reliable planning priors through SFT and then refines inter-agent coordination via RFT, attains peak performance of 60.5 and 82.3. These additive gains confirm that principled initialization (from SFT) and emergent collaboration (from RFT) are both indispensable for maximum performance.

Ablation on Finetuning Strategies. To validate the necessity of full-parameter finetuning, we compare its performance with LoRA that only updates about 2% of the training parameters. As shown in Tab 8, while full-parameter finetuning yields slightly superior performance, the marginal gap confirms that our collaborative policy can be successfully implanted by tuning just this small subset of parameters. This highlights LoRA as a lightweight deployment option without significant accuracy loss.

Ablation on Discussion Mechanisms. Tab 9 evaluates three different discussion mechanisms for aggregating individual agent conclusions. The first ("Best Score") involves each agent scoring its own and others' responses, with the highest-scoring result selected (59.9/81.2). The second ("Decide by Agent") directly adopts the output of Qwen3-8B (chosen for its strongest performance), yielding 60.2/81.6. The third ("Vote") selects the majority-endorsed answer, which further elevates performance to 60.5/82.3. This confirms that the diversity generated by independent agent planning is best leveraged through lightweight majority consensus, outperforming score-based or authority-based selection strategies.

5. Conclusion

We introduce VideoChat-M1, a novel multi-agent framework for adaptive tool invocation in video understanding. Built on a Collaborative Policy Planning (CPP) paradigm and trained with a streamlined Multi-Agent Reinforcement Learning (MARL) approach, the framework dynamically discovers critical clues to achieve robust video reasoning. VideoChat-M1 achieves SOTA performance across eight benchmarks on four mainstream video tasks: long-form video QA, video reasoning, spatial intelligence, and temporal grounding. To the best of our knowledge, this is the first multi-agent policy learning framework for tackling complex video understanding tasks, contributing to the development of more adaptive and intelligent video understanding.

6. Acknowledgements

This work was supported by Guangdong Science and Technology Program (Grant No. 2024TQ08X365)

References

[1] Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, Humen Zhong, Zhu, et al. Qwen2.5-vl technical report. *arXiv preprint arXiv:2502.13923*, 2025. 1, 2, 6

[2] Boyu Chen, Yu Qiao, and Yali Wang. Low-resolution action recognition for tiny actions challenge. *arXiv preprint arXiv:2209.14711*, 2022. 3

[3] Boyu Chen, Siran Chen, Kunchang Li, Qinglin Xu, Yu Qiao, and Yali Wang. Super encoding network: Recursive association of multi-modal encoders for video understanding. *arXiv preprint arXiv:2506.07576*, 2025. 3

[4] Boyu Chen, Siran Chen, Kunchang Li, Qinglin Xu, Yu Qiao, and Yali Wang. Percept, chat, adapt: Knowledge transfer of foundation models for open-world video recognition. *Pattern Recognition*, 160:111189, 2025.

[5] Boyu Chen, Siran Chen, Zhengrong Yue, Kainan Yan, Chenyun Yu, Beibei Kong, Cheng Lei, Chengxiang Zhuo, Zang Li, and Yali Wang. G-ubs: Towards robust understanding of implicit feedback via group-aware user behavior simulation. *arXiv preprint arXiv:2508.05709*, 2025. 3

[6] Boyu Chen, Zhengrong Yue, Siran Chen, Zikang Wang, Yang Liu, Peng Li, and Yali Wang. Lvagent: Long video understanding by multi-round dynamical collaboration of mllm agents. *arXiv preprint arXiv:2503.10200*, 2025. 1, 3

[7] Guo Chen, Zhiqi Li, Shihao Wang, Jindong Jiang, Yicheng Liu, Lidong Lu, De-An Huang, Wonmin Byeon, Matthieu Le, Tuomas Rintamaki, Tyler Poon, Max Ehrlich, Tuomas Rintamaki, Tyler Poon, Tong Lu, Limin Wang, Bryan Catanzaro, Jan Kautz, Andrew Tao, Zhiding Yu, and Guilin Liu. Eagle 2.5: Boosting long-context post-training for frontier vision-language models, 2025. 1, 6, 2

[8] Lin Chen, Xilin Wei, Jinsong Li, Xiaoyi Dong, Pan Zhang, Yuhang Zang, Zehui Chen, Haodong Duan, Zhenyu Tang, Li Yuan, et al. ShareGpt4Video: Improving video understanding and generation with better captions. *Advances in Neural Information Processing Systems*, 37:19472–19495, 2024. 6

[9] Siran Chen, Qinglin Xu, Yue Ma, Yu Qiao, and Yali Wang. Attentive snippet prompting for video retrieval. *IEEE Transactions on Multimedia*, 26:4348–4359, 2023. 1

[10] Siran Chen, Boyu Chen, Chenyun Yu, Yuxiao Luo, Ouyang Yi, Lei Cheng, Chengxiang Zhuo, Zang Li, and Yali Wang. Vragent-r1: Boosting video recommendation with mllm-based agents via reinforcement learning. *arXiv preprint arXiv:2507.02626*, 2025. 3

[11] Siran Chen, Boyu Chen, Chenyun Yu, Yi Ouyang, Cheng Lei, Chengxiang Zhuo, Zang Li, and Yali Wang. When top-ranked recommendations fail: Modeling multi-granular negative feedback for explainable and robust video recommendation. *arXiv preprint arXiv:2511.18700*, 2025. 3

[12] Yukang Chen, Wei Huang, Baifeng Shi, Qinghao Hu, Hanrong Ye, Ligeng Zhu, Zhijian Liu, Pavlo Molchanov, Jan Kautz, Xiaojuan Qi, et al. Scaling rl to long videos. *arXiv preprint arXiv:2507.07966*, 2025. 3, 6

[13] Zhe Chen, Weiyun Wang, Yue Cao, Yangzhou Liu, Zhangwei Gao, Erfei Cui, Jinguo Zhu, Shenglong Ye, Hao Tian, Zhaoyang Liu, et al. Expanding Performance Boundaries of Open-Source Multimodal Models with Model, Data, and Test-Time Scaling. *arXiv preprint arXiv:2412.05271*, 2024. 3, 6

[14] Junhao Cheng, Yuying Ge, Teng Wang, Yixiao Ge, Jing Liao, and Ying Shan. Video-holmes: Can mllm think like holmes for complex video reasoning? *arXiv preprint arXiv:2505.21374*, 2025. 5

[15] 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. 6

[16] Gheorghe Comanici, Eric Bieber, Mike Schaekermann, Ice Pasupat, Noveen Sachdeva, Inderjit Dhillon, Marcel Blistein, Ori Ram, Dan Zhang, Evan Rosen, et al. Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities. *arXiv preprint arXiv:2507.06261*, 2025. 1, 2, 6

[17] Tri Dao. FlashAttention-2: Faster attention with better parallelism and work partitioning. In *International Conference on Learning Representations (ICLR)*, 2024. 8

[18] Andrew Estornell, Jean-Francois Ton, Muhammad Faaiz Taufiq, and Hang Li. How to train a leader: Hierarchical reasoning in multi-agent llms. *arXiv preprint arXiv:2507.08960*, 2025. 3

[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, 2024. 1, 3

[20] Kaituo Feng, Kaixiong Gong, Bohao Li, Zonghao Guo, Yibing Wang, Tianshuo Peng, Junfei Wu, Xiaoying Zhang, Benyou Wang, and Xiangyu Yue. Video-r1: Reinforcing video reasoning in mllms. *arXiv preprint arXiv:2503.21776*, 2025. 6

[21] Chaoyou Fu, Yuhan Dai, Yongdong Luo, Lei Li, Shuhuai Ren, Renrui Zhang, Zihan Wang, Chenyu Zhou, Yunhang Shen, Mengdan Zhang, Peixian Chen, Yanwei Li, Shaohui Lin, Sirui Zhao, Ke Li, Tong Xu, Xiawu Zheng, Enhong Chen, Rongrong Ji, and Xing Sun. Video-MME: The First-Ever Comprehensive Evaluation Benchmark of Multi-modal LLMs in Video Analysis, 2024. 5

[22] Hongcheng Gao, Yue Liu, Yufei He, Longxu Dou, Chao Du, Zhijie Deng, Bryan Hooi, Min Lin, and Tianyu Pang. Flowreasoner: Reinforcing query-level meta-agents. *arXiv preprint arXiv:2504.15257*, 2025. 3

[23] Jiyang Gao, Chen Sun, Zhenheng Yang, and Ram Nevatia. Tall: Temporal activity localization via language query. In *Proceedings of the IEEE international conference on computer vision*, pages 5267–5275, 2017. 2, 5

[24] Dong Guo, Faming Wu, Feida Zhu, Fuxing Leng, Guang Shi, Haobin Chen, Haoqi Fan, Jian Wang, Jianyu Jiang, Jiawei Wang, et al. Seed1. 5-vl technical report. *arXiv preprint arXiv:2505.07062*, 2025. 1, 2, 6

[25] Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, et al. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*, 2025. 1, 2, 5, 7

[26] 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, 2024. 3

[27] Zhuohong He, Ali Mottaghi, Aidean Sharghi, Muhammad Abdullah Jamal, and Omid Mohareri. An Empirical Study on Activity Recognition in Long Surgical Videos. In *Proceedings of the 2nd Machine Learning for Health symposium*, pages 356–372. PMLR, 2022. 1

[28] Zhitao He, Zijun Liu, Peng Li, May Fung, Ming Yan, Ji Zhang, Fei Huang, and Yang Liu. Enhancing language multi-agent learning with multi-agent credit re-assignment for interactive environment generalization. *arXiv preprint arXiv:2502.14496*, 2025. 3

[29] Sirui Hong, Mingchen Zhuge, Jonathan Chen, Xiawu Zheng, Yuheng Cheng, Jinlin Wang, Ceyao Zhang, Zili Wang, Steven Ka Shing Yau, Zijuan Lin, et al. Metagpt: Meta programming for a multi-agent collaborative framework. In *The Twelfth International Conference on Learning Representations*, 2023. 3

[30] Kairui Hu, Penghao Wu, Fanyi Pu, Wang Xiao, Yuanhan Zhang, Xiang Yue, Bo Li, and Ziwei Liu. Video-mmmu: Evaluating knowledge acquisition from multi-discipline professional videos. *arXiv preprint arXiv:2501.13826*, 2025. 5

[31] Kairui Hu, Penghao Wu, Fanyi Pu, Wang Xiao, Yuanhan Zhang, Xiang Yue, Bo Li, and Ziwei Liu. Video-mmmu: Evaluating knowledge acquisition from multi-discipline professional videos. *arXiv preprint arXiv:2501.13826*, 2025. 2

[32] Soyeong Jeong, Kangsan Kim, Jinheon Baek, and Sung Ju Hwang. VideoRAG: Retrieval-Augmented Generation over Video Corpus, 2025. 1, 6

[33] Bo Li, Yuanhan Zhang, Dong Guo, Renrui Zhang, Feng Li, Hao Zhang, Kaichen Zhang, Yanwei Li, Ziwei Liu, and Chunyuan Li. LLaVA-OneVision: Easy Visual Task Transfer. *arXiv preprint arXiv:2408.03326*, 2024. 6

[34] Chuanhao Li, Zhen Li, Chenchen Jing, Shuo Liu, Wenqi Shao, Yuwei Wu, Ping Luo, Yu Qiao, and Kaipeng Zhang. SearchLVLMs: A Plug-and-Play Framework for Augmenting Large Vision-Language Models by Searching Up-to-Date Internet Knowledge, 2024. 3

[35] Dongxu Li, Yudong Liu, Haoning Wu, Yue Wang, Zhiqi Shen, Bowen Qu, Xinyao Niu, Guoyin Wang, Bei Chen, and Junnan Li. Aria: An Open Multimodal Native Mixture-of-Experts Model. *arXiv preprint arXiv:2410.05993*, 2024. 6

[36] Guohao Li, Hasan Abed Al Kader Hammoud, Hani Itani, Dmitrii Khizbullin, and Bernard Ghanem. CAMEL: Communicative agents for "mind" exploration of large language model society. In *Proceedings of Thirty-seventh Conference on Neural Information Processing Systems*, 2023. 3

[37] KunChang Li, Yinan He, Yi Wang, Yizhuo Li, Wenhai Wang, Ping Luo, Yali Wang, Limin Wang, and Yu Qiao. VideoChat: Chat-Centric Video Understanding. *arXiv preprint arXiv:2305.06355*, 2023. 3

[38] 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. 6

[39] Xinhao Li, Yi Wang, Jiashuo Yu, Xiangyu Zeng, Yuhan Zhu, Haian Huang, Jianfei Gao, Kunchang Li, Yinan He, Chenting Wang, et al. Videochat-flash: Hierarchical compression for long-context video modeling. *arXiv preprint arXiv:2501.00574*, 2024. 6

[40] Xinhao Li, Ziang Yan, Desen Meng, Lu Dong, Xiangyu Zeng, Yinan He, Yali Wang, Yu Qiao, Yi Wang, and Limin Wang. Videochat-r1: Enhancing spatio-temporal perception via reinforcement fine-tuning. *arXiv preprint arXiv:2504.06958*, 2025. 3, 6

[41] 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, 2024. 3

[42] Junwei Liao, Muning Wen, Jun Wang, and Weinan Zhang. Marft: Multi-agent reinforcement fine-tuning. *arXiv preprint arXiv:2504.16129*, 2025. 3

[43] Hao Liu, Wilson Yan, Matei Zaharia, and Pieter Abbeel. World Model on Million-Length Video and Language with RingAttention. *arXiv preprint*, 2024. 3

[44] Zuyan Liu, Yuhao Dong, Ziwei Liu, Winston Hu, Jiwen Lu, and Yongming Rao. Oryx MLLM: On-Demand Spatial-Temporal Understanding at Arbitrary Resolution. *arXiv preprint arXiv:2409.12961*, 2024. 6

[45] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017. 7

[46] Ziyu Ma, Chenhui Gou, Hengcan Shi, Bin Sun, Shutao Li, Hamid Rezatofighi, and Jianfei Cai. DrVideo: Document Retrieval Based Long Video Understanding, 2024. 6

[47] Léa Martinez, Manuel Gimenes, and Eric Lambert. Entertainment Video Games for Academic Learning: A Systematic Review. *Journal of Educational Computing Research*, 60(5):1083–1109, 2022. 1

[48] Zhanfeng Mo, Xingxuan Li, Yuntao Chen, and Lidong Bing. Multi-agent tool-integrated policy optimization, 2025. 3

[49] Sumeet Ramesh Motwani, Chandler Smith, Rocktim Jyoti Das, Rafael Rafailov, Ivan Laptev, Philip HS Torr, Fabio Pizzati, Ronald Clark, and Christian Schroeder de Witt. Malt: Improving reasoning with multi-agent llm training. *arXiv preprint arXiv:2412.01928*, 2024. 3

[50] Michael Noetel, Shantell Griffith, Oscar Delaney, Taren Sanders, Philip Parker, Borja del Pozo Cruz, and Chris Lonsdale. Video Improves Learning in Higher Education: A Systematic Review. *Review of Educational Research*, 91(2): 204–236, 2021. 1

[51] OpenAI. GPT-4o System Card, 2024. 1, 2, 6, 7

[52] Chanwoo Park, Seungju Han, Xingzhi Guo, Asuman Ozdaglar, Kaiqing Zhang, and Joo-Kyung Kim. Maporl: Multi-agent post-co-training for collaborative large language models with reinforcement learning. *arXiv preprint arXiv:2502.18439*, 2025. 3

[53] Minghao Qin, Xiangrui Liu, Zhengyang Liang, Yan Shu, Huaying Yuan, Juenjie Zhou, Shitao Xiao, Bo Zhao, and Zheng Liu. Video-xl-2: Towards very long-video understanding through task-aware kv sparsification, 2025. 6

[54] 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. 3

[55] Gemini Team. Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context, 2024. 2, 6, 7

[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. 3

[57] Ziyu Wan, Yunxiang Li, Xiaoyu Wen, Yan Song, Hanjing Wang, Linyi Yang, Mark Schmidt, Jun Wang, Weinan Zhang, Shuyue Hu, et al. Rema: Learning to meta-think for llms with multi-agent reinforcement learning. *arXiv preprint arXiv:2503.09501*, 2025. 3

[58] Peng Wang, Shuai Bai, Sinan Tan, Shijie Wang, Zhihao Fan, Jinze Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Yang Fan, Kai Dang, Mengfei Du, Xuancheng Ren, Rui Men, Dayiheng Liu, Chang Zhou, Jingren Zhou, and Junyang Lin. Qwen2-VL: Enhancing Vision-Language Model's Perception of the World at Any Resolution. *arXiv preprint arXiv:2409.12191*, 2024. 6, 7

[59] Weiyun Wang, Zhangwei Gao, Lixin Gu, Hengjun Pu, Long Cui, Xingguang Wei, Zhaoyang Liu, Linglin Jing, Shenglong Ye, Jie Shao, et al. Internvl3. 5: Advancing open-source multimodal models in versatility, reasoning, and efficiency. *arXiv preprint arXiv:2508.18265*, 2025. 1, 2, 6

[60] Xidong Wang, Dingjie Song, Shunian Chen, Chen Zhang, and Benyou Wang. LongLLaVA: Scaling Multi-modal LLMs to 1000 Images Efficiently via a Hybrid Architecture. *arXiv preprint arXiv:2409.02889*, 2024. 3

[61] Xiaohan Wang, Yuhui Zhang, Orr Zohar, and Serena Yeung-Levy. VideoAgent: Long-form Video Understanding with Large Language Model as Agent, 2024. 1

[62] Yi Wang, Xinhao Li, Ziang Yan, Yinan He, Jiashuo Yu, Xiangyu Zeng, Chenting Wang, Changlian Ma, Haian Huang, Jianfei Gao, et al. Internvideo2. 5: Empowering video mllms with long and rich context modeling. *arXiv preprint arXiv:2501.12386*, 2025. 6

[63] Yinjie Wang, Ling Yang, Ye Tian, Ke Shen, and Mengdi Wang. Co-evolving llm coder and unit tester via reinforcement learning, 2025. 3

[64] 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, 2024. 1, 3

[65] Zikang Wang, Boyu Chen, Zhengrong Yue, Yi Wang, Yu Qiao, Limin Wang, and Yali Wang. Videochat-a1: Thinking with long videos by chain-of-shot reasoning. *arXiv preprint arXiv:2506.06097*, 2025. 1, 3, 6

[66] Yuan Wei, Xiaohan Shan, Ran Miao, and Jianmin Li. Lero: Llm-driven evolutionary framework with hybrid rewards and enhanced observation for multi-agent reinforcement learning. In *International Conference on Intelligent Computing*, pages 15–26. Springer, 2025. 3

[67] Haoning Wu, Dongxu Li, Bei Chen, and Junnan Li. LongVideoBench: A Benchmark for Long-context Interleaved Video-Language Understanding, 2024. 2, 5

[68] Fuzhao Xue, Yukang Chen, Dacheng Li, Qinghao Hu, Ligeng Zhu, Xiuyu Li, Yunhao Fang, Haotian Tang, Shang Yang, Zhijian Liu, et al. Longvila: Scaling long-context visual language models for long videos. *arXiv preprint arXiv:2408.10188*, 2024. 1, 6

[69] Ziang Yan, Xinhao Li, Yinan He, Zhengrong Yue, Xiangyu Zeng, Yali Wang, Yu Qiao, Limin Wang, and Yi Wang. Videochat-r1.5: Visual test-time scaling to reinforce multimodal reasoning by iterative perception, 2025. 6

[70] An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025. 4

[71] Jihan Yang, Shusheng Yang, Anjali W Gupta, Rilyn Han, Li Fei-Fei, and Saining Xie. Thinking in space: How multimodal large language models see, remember, and recall spaces. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 10632–10643, 2025. 2, 5

[72] Huaying Yuan, Zheng Liu, Junjie Zhou, Hongjin Qian, Ji-Rong Wen, and Zhicheng Dou. Videodeepresearch: Long video understanding with agentic tool using. *arXiv preprint arXiv:2506.10821*, 2025. 1, 6

[73] Zhengrong Yue, Haiyu Zhang, Xiangyu Zeng, Boyu Chen, Chenting Wang, Shaobin Zhuang, Lu Dong, KunPeng Du, Yi Wang, Limin Wang, et al. Uniflow: A unified pixel flow tokenizer for visual understanding and generation. *arXiv preprint arXiv:2510.10575*, 2025. 3

[74] 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. 3

[75] Zeyuan Yang and Delin Chen and Xueyang Yu and Maohao Shen and Chuang Gan. Vca: Video curious agent for long video understanding, 2024. 1, 6

[76] Guibin Zhang, Yanwei Yue, Xiangguo Sun, Guancheng Wan, Miao Yu, Junfeng Fang, Kun Wang, Tianlong Chen, and Dawei Cheng. G-designer: Architecting multi-agent communication topologies via graph neural networks. *arXiv preprint arXiv:2410.11782*, 2024. 3

[77] Guibin Zhang, Luyang Niu, Junfeng Fang, Kun Wang, Lei Bai, and Xiang Wang. Multi-agent architecture search via agentic supernet. *arXiv preprint arXiv:2502.04180*, 2025. 3

[78] Hang Zhang, Xin Li, and Lidong Bing. Video-llama: An instruction-tuned audio-visual language model for video understanding. *arXiv preprint arXiv:2306.02858*, 2023. 3

[79] Haoji Zhang, Xin Gu, Jiawen Li, Chixiang Ma, Sule Bai, Chubin Zhang, Bowen Zhang, Zhichao Zhou, Dongliang He, and Yansong Tang. Thinking with videos: Multimodal tool-

augmented reinforcement learning for long video reasoning. *arXiv preprint arXiv:2508.04416*, 2025. 1, 6

[80] Peiyuan Zhang, Kaichen Zhang, Bo Li, Guangtao Zeng, Jingkang Yang, Yuanhan Zhang, Ziyue Wang, Haoran Tan, Chunyuan Li, and Ziwei Liu. Long context transfer from language to vision. *arXiv preprint arXiv:2406.16852*, 2024. 6

[81] Xiaoyi Zhang, Zhaoyang Jia, Zongyu Guo, Jiahao Li, Bin Li, Houqiang Li, and Yan Lu. Deep video discovery: Agentic search with tool use for long-form video understanding. *arXiv preprint arXiv:2505.18079*, 2025. 6

[82] Yuanhan Zhang, Jinming Wu, Wei Li, Bo Li, Zejun Ma, Ziwei Liu, and Chunyuan Li. Video Instruction Tuning With Synthetic Data, 2024. 3, 6

[83] Jun Zhao, Can Zu, Hao Xu, Yi Lu, Wei He, Yiwen Ding, Tao Gui, Qi Zhang, and Xuanjing Huang. LongAgent: Scaling Language Models to 128k Context through Multi-Agent Collaboration. *arXiv preprint arXiv:2402.11550*, 2024. 3

[84] Junjie Zhou, Yan Shu, Bo Zhao, Boya Wu, Shitao Xiao, Xi Yang, Yongping Xiong, Bo Zhang, Tiejun Huang, and Zheng Liu. MLVU: A Comprehensive Benchmark for Multi-Task Long Video Understanding. *arXiv preprint arXiv:2406.04264*, 2024. 5

[85] Yiyang Zhou, Yangfan He, Yaofeng Su, Siwei Han, Joel Jang, Gedas Bertasius, Mohit Bansal, and Huaxiu Yao. Reagent-v: A reward-driven multi-agent framework for video understanding, 2025. 1, 6

[86] Jinguo Zhu, Weiyun Wang, Zhe Chen, Zhaoyang Liu, Shenglong Ye, Lixin Gu, Hao Tian, Yuchen Duan, Weijie Su, Jie Shao, et al. Internvl3: Exploring advanced training and test-time recipes for open-source multimodal models. *arXiv preprint arXiv:2504.10479*, 2025. 6

[87] Kejian Zhu, Zhuoran Jin, Hongbang Yuan, Jiachun Li, Shangqing Tu, Pengfei Cao, Yubo Chen, Kang Liu, and Jun Zhao. Mmr-v: What's left unsaid? a benchmark for multimodal deep reasoning in videos. *arXiv preprint arXiv:2506.04141*, 2025. 5

[88] Mingchen Zhuge, Wenyi Wang, Louis Kirsch, Francesco Faccio, Dmitrii Khizbullin, and Jürgen Schmidhuber. Gptswarm: Language agents as optimizable graphs. In *Forty-first International Conference on Machine Learning*, 2024. 3

VideoChat-M1: Collaborative Policy Planning for Video Understanding via Multi-Agent Reinforcement Learning

Supplementary Material

Type	Dataset	Instance Num	Avg Video Length (s)
Temporal Grounding	FineAction	5067	43.64
	QVHighlights	13790	28.36
	HiREST	3617	282.45
Long Video QA	ActivityNet-QA	16642	621.45
	LongViTU	16453	268.46
	MMBench	1673	97.51
	MovieChat	808	457.65
	Neptune	5281	149.25
Spatial Intelligence	HoursVideo	831	568.16
	SpaceR	12643	10.65
Video Reasoning	Video-R1	15123	68.56
	VideoEspresso	9432	56.12
	Video Holmes (Training Set)	1551	91.16
Total	–	**102911**	**194.6**

Table 10. Instance numbers of different datasets for VideoChat-M1 training.

A.1 Collected Dataset

To equip VideoChat-M1 with strong generalization across diverse video understanding scenarios, we assemble a comprehensive collection of datasets spanning multiple task types, including temporal grounding, long-video question answering, Spatial Intelligence analysis, and video reasoning. These datasets originate from widely used benchmarks and cover a broad spectrum of video durations, scenes, and annotation forms. The diversity of tasks and data sources empowers VideoChat-M1 to learn from heterogeneous supervision signals, enhancing its capabilities in perceiving, retrieving, and reasoning over long and complex videos. Table 10 summarizes the instance numbers and average video durations of all datasets used in our training pipeline. In total, the dataset collection comprises 102,911 instances with an overall average video duration of 194.6 seconds, laying a solid data foundation for training VideoChat-M1 on four mainstream video tasks.

A.2 Memory Buffer and Tool Use

We implement the memory buffer as a key-value pair structure, in which keys denote the agents' names and values store the structured information illustrated in Fig 6. We take the memory buffer of Qwen3-8B as an example.

Memory Buffer

Agent Name: Qwen3-8B
The initial plan is: `<Global Sampling>, <Video Retrieval>, <Rough Browser>`
Key info: The priority is to pinpoint the exact moment the elf interacts with the vacuum and confirm the instantaneous change in its operation that leads to the woman being sucked in.
Turn 1:
Executing Tool: `Global Sampling`
Tool Output: Uniform Sampling 16 frames of the video. The frame index list is [xxx]
Communication Decision: Continue with the plan, execute `Video Retrieval`
Turn 2:
Executing Tool: `Video Retrieval`
Tool Output: Sample 16 frames on clip 6 of the video based on key info. The frame index list is [xxx].
Communication Decision: Continue with the plan, execute `Rough Browser`
Turn 3:
Executing Tool: `Rough Browser`
Answer: A
Summary: The critical action occurs when these creatures turn on the vacuum cleaner. The woman then trips over the machine's power cord, which is a consequence of the creatures' actions. After she falls, they manipulate the vacuum's direction, leading directly to her being sucked in by its suction force.
Plan Finished

To enable our multi-agent framework to tackle a diverse array of video understanding tasks, we provide each agent with access to a comprehensive and specialized toolkit $\mathcal{T}$. These tools facilitate efficient information extraction, spanning coarse-grained retrieval to fine-grained perceptual analysis. The tools available are as follows:

- **Global Sampling:** For queries requiring a holistic understanding, this tool uniformly samples frames across the entire video duration.
- **Video Retrieval:** This tool first divides the video into six equal-length clips. It then employs the ASP-CLIP [9] model to score the semantic similarity between each clip and the user query, returning the highest-scoring clip for

further analysis.

- **Time Stamp Retrieval:** When a precise moment is referenced, this tool extracts a one-minute video segment centered at the specified timestamp.
- **Image Retrieval:** For the image retrieval stage, we uniformly sample frames from the source video at a rate of 2 frames per second (fps). We then employ the pre-trained CLIP model to compute the similarity score between the textual prompt and each sampled frame, ultimately selecting the top 16 (or 32) frames with the highest similarity.
- **Rough Browser:** This tool provides a rapid overview by processing a sparse set of 16 selected frames with a Multimodal Large Language Model (MLLM), such as Qwen2.5-VL-7B [1].
- **Fine Browser:** For detailed analysis where a deeper look is necessary, this tool leverages the same MLLM to process a denser sequence of 32 frames extracted from a targeted video clip.
- **Spatial Tool:** To address spatial reasoning queries, this tool employs the InternVL-3.5-8B [59] model to analyze 16 frames, which are either uniformly sampled or sourced from a retrieved clip.
- **Grounding Tool:** This specialized tool is designed for temporal grounding tasks and utilizes the Eagle2.5-7B [7] model to process the video and identify relevant time segments.

A.3 Prompt

In this section, we detail the prompts employed in each step of our proposed method.

Prompt for Policy Generation

You are an intelligent video understanding agent. Your task is to analyze a video question and select the optimal combination of tools to answer it accurately.

1. Tool Definitions

Group A: Frame Selection Tools (Retrieval Phase)

- **Uniform Sampling:** A general strategy. Use this only when the question is broad or covers the whole video. It summarizes the overall content without focusing on specific details.
- **Video Retrieval:** The standard semantic search method. Use this to locate the most relevant video clips containing the action, event, or object described in the text query.
- **Time Stamp Retrieval:** Deterministic retrieval. Use this strictly when the question mentions a specific time (e.g., "at 01:30").
- **Image Retrieval:** Fine-grained visual matching. Use this to identify specific static scenes, small objects, or person attributes by matching text descriptions to individual frames (top-k selection).

Group B: Video Browsing Tools (Reasoning Phase)

- **Rough Browser:** Provides a comprehensive yet efficient overview of the selected frames. Sufficient for answering the majority of general video understanding questions.
- **Fine Browser:** High-computation analysis. Use this *only* for cases of extreme ambiguity or when deciphering subtle details (e.g., small text, rapid motions) is critical.
- **Spatial Tool:** Specialized for spatial reasoning benchmarks (e.g., VSIBench). Use this when the question explicitly asks about relative positions, geometry, or spatial arrangements of objects.
- **Grounding Tool:** Specialized for temporal localization (e.g., Charades-STA). Use this strictly for simple, single-scene grounding tasks where the goal is to identify start/end timestamps rather than complex reasoning.

2. Recommended Workflow

You **MUST** adhere to the following selection rules:

1. **Selection Phase:** You must select **ONE** or more tools from Group A (Frame Selection).
2. **Browsing Phase:** You must select **ONE** or more tools from Group B (Video Browsing).
3. "Analyze the question and candidate options to determine the key information necessary for the reasoning process. This becomes your **Key info**.

Current Task: {task}
Question: {question}
3. Output Format & Examples
Example 1 (General Reasoning):
Question: What does the object being chased by the people refer to?
Options: A: Difficulties in life, B: His fully automatic house...

Format:
##key info: the object being chased by the people in the video.
##tool use: `<Video Retrieval>`, `<Rough Browser>`

Example 2 (Spatial Intelligence):
Question: If I am standing by the table and facing the bathtub, is the bed to my left, right, or back?

Options: A: left, B: right, C: Back

Format:
##key info: spatial relation between bathtub and bed.
##tool use: <Video Retrieval>, <Spatial Tool>

Prompt for Policy Communication

You are a strategic planning assistant. Your sole responsibility is to evaluate the current execution state and determine the immediate next step.
1. CURRENT CONTEXT
Review the following execution state carefully:
- **Original Question:** {question}
- **Memory buffer:** {memory}
- **Other Agents' Output:** {other agents output}
- **Remaining Plan:** {plan}

2. DECISION PROTOCOL
You must choose exactly **ONE** action from the list below based on the logic provided:

Option A: The Standard Path
- `continue()`: Use this to proceed with the {next tool}. **Rule:** Apply this when peer agents offer no constructive alternatives and the current internal plan remains valid and error-free.

Option B: Exception Handling
- `add tool(tool name='<name>')`: Use this **ONLY** if the current plan is logically flawed and requires a new tool (e.g., 'Video Retrieval') to proceed. Analyze the question, candidate options, and the memory of all agents to determine the key information necessary for the reasoning process. This becomes the **Key info**.

Output your response strictly in the format below.

Format:
Scenario 1: Continuing (Default)
##tool call: continue()
Scenario 2: Adding a Tool (Correction)
##tool call: add tool <Video Retrieval>
##key info: xx

Prompt for Answering the Question

(Use this when the agent's plan is fully executed, but the answer remains unresolved)

You are an intelligent agent responsible for synthesizing a final answer based strictly on the provided internal logs, referred to as {Agent Memory}. You must adhere to the following format constraints based on the presence of options.

Input Context:
- **Question**: {Question}
- **Option**: {Option} *(Note: If this field is empty, treat as an open-ended task or temporal grounding task.)*
- **Task**: {Task}
- **Agent Memory**: {Agent Memory}

Directives:
1. **Source of Truth:** Your response must be derived solely from the information contained within {Agent Memory}. Do not hallucinate or use external knowledge.
2. **Multiple Choice Logic:** If {Option} is provided (e.g., A, B, C, D), your final output must be **the single uppercase letter** corresponding to the correct choice and the reason for your answer.
3. **Open-Ended Logic:** If {Option} is not provided (e.g., Temporal Grounding or open-ended QA), your final output must be a paragraph explaining the reasoning for the answer.

Format:
##Answer: xx
##Reason: xx

Prompt for Reason Summary

Input Context:
You have been provided with the reasoning from four distinct AI agents:
- **{Agent0 name}**: {agent 0 reason}
- **{Agent1 name}**: {agent 1 reason}
- **{Agent2 name}**: {agent 2 reason}
- **{Agent3 name}**: {agent 3 reason}

Your Task:
Synthesize and summarize the reasons of each agent into a single, cohesive paragraph.
Critically, you must adhere to the following synthesis logic based on the question type:
1. **For Multiple Choice Questions (Options provided):** Identify the final consensus option (or

the selected answer). You must ONLY summarize the results and reasoning of the agents that agreed with this final option. Ignore the reasoning of dissenting agents unless it provides critical context for the correct answer.

2. **For Open-Ended Questions (No options provided):** Synthesize and summarize the reasoning from **ALL** agents to provide a comprehensive answer. In particular, the summarization should prioritize the consensus among agents, placing greater emphasis on convergent reasoning paths found in similar responses.

The final summary must be concise but accurately reflect the sequence of events and core logic.

Format:
##Final Answer: xx
##Reason Summary: xx

Prompt for Rough Browser

Input Components:
You will be provided with the following:
- A sequence of key frames extracted from a video.
- Question:{Question} and Options {if have Options or None}
- A key info text that specifies the central theme for the summary. {Key info}

Your Task:
Write a brief summary of the video's content. The summary **must be centered around** the event, object, or action described in the **Key info**. The entire summary must be **no more than 128 tokens**. If you can provide the answer to the question, you can also give the answer.

Output Format:
##Answer: xx
##Summary: A single, concise paragraph containing the summary.

Prompt for Fine Browser

Input Components:

- A video clip requiring detailed examination.
- Question:{Question} and Options {if have Options or None}
- A key info text that directs the model's focus to the most critical aspect of the video for solving the problem. {`key info`}

The model's core task is to generate a detailed summary by analyzing 32 uniformly sampled frames from the video clip. This summary must be thematically centered on the event, object, or action specified in the **Key info**. This fine-grained analysis is specifically designed to resolve high ambiguity and decipher subtle details (e.g., small text, rapid motions) that are critical for a correct interpretation. If you can provide the answer to the question, you can also give the answer.

Output Format:
##Answer: xx
##Summary: The final summary must be concise yet descriptive, and it **must not exceed 256 tokens**.

Prompt for Spatial Tool

Input Components:

- A video clip requiring detailed examination.
- Question:{Question} and Options {if have Options or None}
- A key info text that directs the model's focus to the specific spatial question that needs to be answered. {`key info`}

This tool is specifically invoked for queries concerning the relative positions, geometry, or spatial arrangements of objects, as is common in spatial reasoning benchmarks (e.g., VSIBench). To address these queries, the model's core task is to analyze 32 uniformly sampled frames to build a comprehensive understanding of the scene's spatial layout. It must then generate a descriptive summary that explicitly answers the spatial question posed in the **Key info** by identifying key objects and precisely describing their positions relative to each other. If you can provide the answer to the question, you can also give the answer.

Output Format:
##Answer: xx
##Summary: The final summary must be concise yet descriptive, and it **must not exceed 256 tokens**.

Prompt for Grounding Tool

Given a user-provided textual key info prompt and a video, the model must retrieve the precise time seg-

ment in the video that directly corresponds to the prompt. Furthermore, the model must generate a concise, natural language justification for its selection.
The textual prompt is: {**Key info**}
Output Format:
##Timestamp: [xxs - xxs]
##Reason: xxx

A.4 Visualization

To obtain qualitative insights into our method's mechanics and efficacy, this section presents a two-part visual analysis of VideoChat-M1. First, Fig 6 and Fig 7 provide a fine-grained visualization of the Collaborative Policy Planning (CPP) process, tracing policy evolution and intermediate reasoning steps to enhance the interpretability of our multi-agent framework. Second, Fig 8 reports a comparative qualitative evaluation, benchmarking the visual outputs of VideoChat-M1 against those of state-of-the-art models across four canonical video understanding tasks. This is intended to empirically validate the performance improvements achieved by our method.

Fig 6 details each step of our CPP process and its corresponding output. It demonstrates that our framework can autonomously refine its plans during execution and exhibits a high degree of fault tolerance, enabling the agent group to recover from errors made by individual agents. The final summary is generated by synthesizing the rationales from all agents that voted for the correct answer ('A'), a task facilitated by the Qwen3-8B model.

In Fig 7, we present a step-by-step visualization of our CPP framework applied to the open-ended temporal grounding task. Initially, a video retrieval tool is employed as a coarse-grained filter, significantly constricting the temporal search space to a relevant video clip. Subsequently, our CPP method operates within this narrowed window to perform fine-grained boundary refinement. As demonstrated by the query 'a woman shot the man and escaped,' the retrieval module effectively eliminated irrelevant footage, enabling our model to focus on the semantic context. Consequently, the method precisely localized the target interval, aligning perfectly with the ground truth, although the Qwen2.5-3B agent failed to find the result.

Fig 8 compares our method with recent Multimodal Large Language Models (MLLMs) across four mainstream tasks. This comparison reveals that existing models frequently rely on superficial cues, miss critical shots, or fail to maintain long-range temporal and spatial consistency, leading to incorrect reasoning. In contrast, VideoChat-M1 reliably identifies causal relations, tracks events over long video durations, infers accurate spatial layouts, and precisely localizes actions in time. These results show that our collaborative, multi-step reasoning framework delivers more accurate, stable, and interpretable video understanding compared to prior approaches.

A.5 Reinforcement Learning Analysis

While a formal convergence proof for complex Multi-Agent Reinforcement Learning (MARL) systems often remains intractable, we establish a robust rationale for the stability and convergence of our proposed training framework. Its design systematically integrates a series of principles, each targeting a known failure mode in MARL, with convergence anchored in four pillars:

1. **Policy Initialization via Supervised Fine-Tuning (SFT).** A primary challenge in RL lies in the vast and unstructured exploration space, which causes inefficient or divergent training. Our framework addresses this with a curriculum-driven SFT phase. Specifically, it provides a crucial "warm-start" by pre-training each agent on a corpus of high-quality expert policies. Consequently, the MARL optimization process is initialized in a highly structured and effective region of the joint policy space. This approach circumvents the instabilities of *tabula rasa* learning and substantially improves the tractability of subsequent exploration, or as stated in the work, it is essential for "laying the foundation for collaborative learning in MARL".
2. **Stable Policy Updates via Group Relative Policy Optimization (GRPO).** The non-stationarity inherent in multi-agent learning where each agent's optimal policy shifts as others learn can destabilize policy updates. The GRPO algorithm directly mitigates this by incorporating a KL-divergence penalty, a core principle of robust policy gradient methods like TRPO and PPO. As shown in Eq. 5, its objective function enforces a trust region for policy updates:

$$\max_{\theta} \mathbb{E}_{\mathbf{o}\sim\pi_{\text{old}}} \left[\frac{\pi_\theta(\mathbf{o}_k)}{\pi_{\text{old}}(\mathbf{o}_k)} A_R^{(k)} - \beta D_{\text{KL}}(\pi_\theta \| \pi_{\text{ref}}) \right] \quad (5)$$

   This constraint regularizes learning dynamics by limiting excessive deviations from a trusted reference policy (π_{ref}), guaranteeing a monotonic improvement trajectory and fostering training stability.
3. **Dense and Structured Reward Shaping.** MARL systems often face sparse rewards and credit assignment issues, which cause ill-defined optimization landscapes with suboptimal local equilibria. Our framework mitigates this via a dense and multi-faceted reward signal composed of three components: task success (R_{res}), procedural validity (R_{format}), and collaboration quality (R_{col}). This hybrid structure provides a continuous and informative gradient signal, guiding agents toward cor-

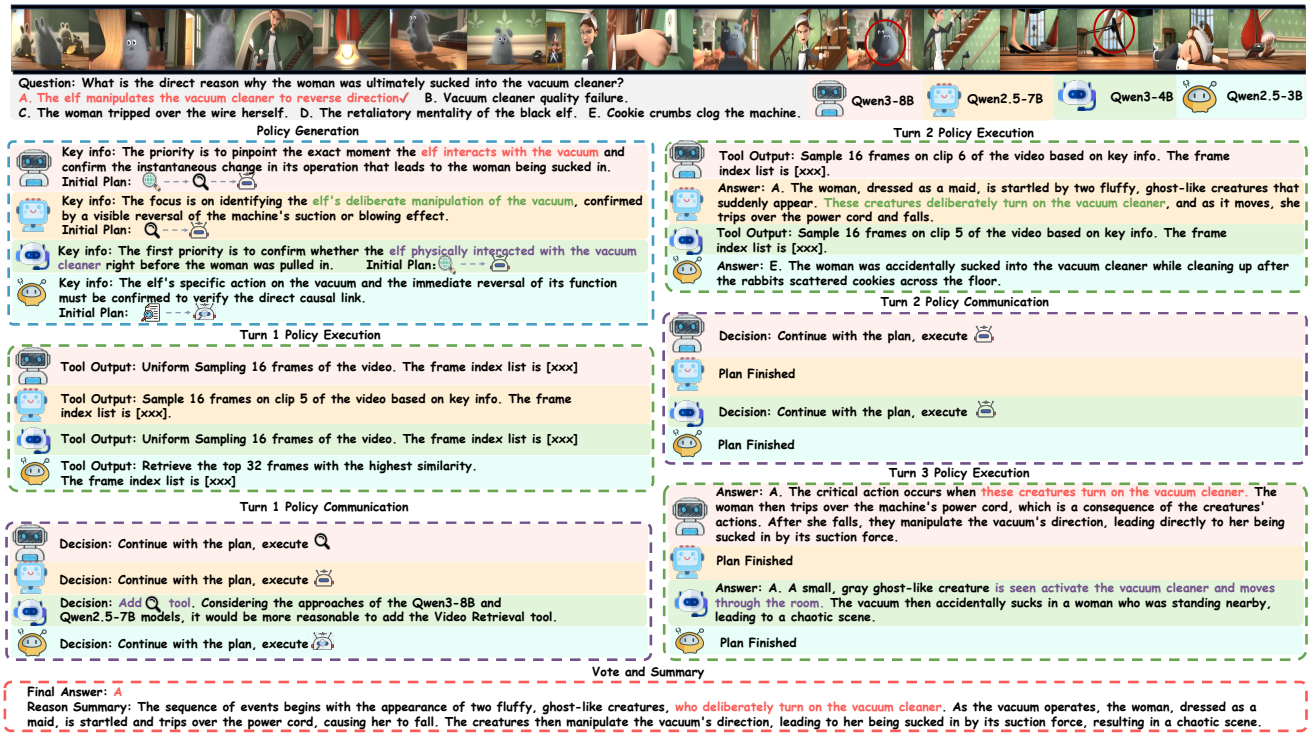


Figure 6. Visualization of VideoChat-M1 at each step of the CPP process.

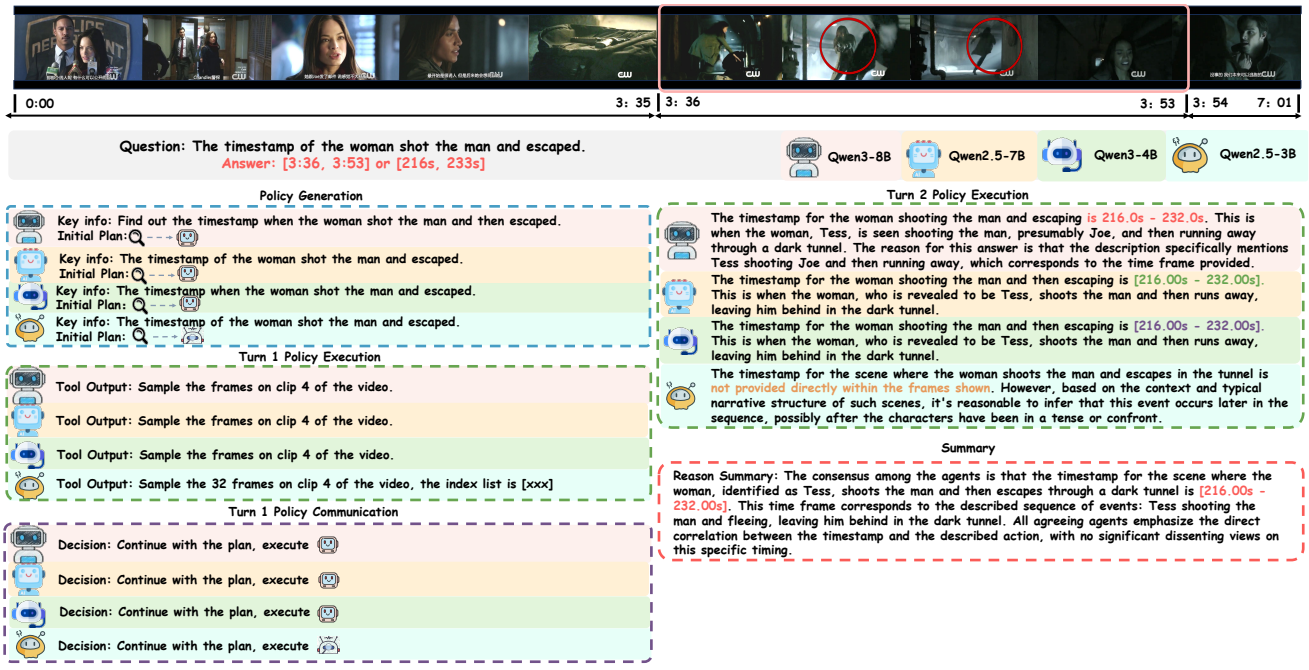


Figure 7. Visualization of VideoChat-M1 on each step of the CPP process on the temporal grounding task.

rect outcomes, valid behaviors and effective cooperation, smoothing the optimization landscape to facilitate gradient-based convergence.

4. **Robustness via Agent Dropout Regularization.** A common failure mode in MARL is inter-agent co-adaptation, where agents develop brittle strategies that are over-specialized to their teammates' specific policies. To address this, we employ agent dropout as a form of structural regularization. By dynamically and stochastically adjusting the communication topology during training,



Figure 8. Visualization of VideoChat-M1 on four mainstream tasks.

this technique discourages dependencies on any single agent and compels the development of more generalized and robust policies. This enhances the stability of the learned multi-agent equilibrium and ensures convergence to solutions resilient to minor policy perturbations, a fact supported by ablation studies identifying it as the "most critical regularizer".

In summary, VideoChat-M1's training convergence is not heuristic but derives from a principled framework design. By systematically addressing initialization (via SFT), update stability (via GRPO), reward-landscape tractability (via dense rewards) and robust generalization (via agent dropout), the framework holistically mitigates common MARL instabilities, guiding the agent system toward a stable and high-performance collaborative pipeline.

A.6 More Implementation Details

Training Setup. We employ the AdamW [45] optimizer with a learning rate of 1e-7 and a global batch size of 8. The training process is distributed across 8 NVIDIA A100

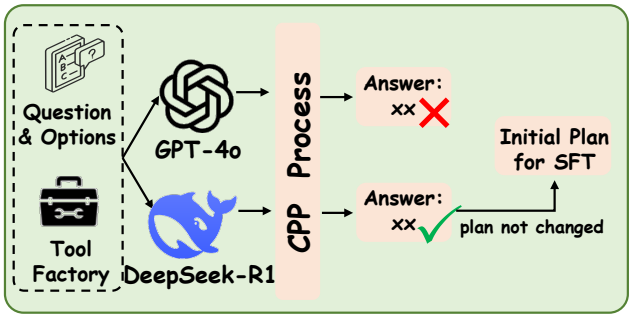


Figure 9. The process of generating the SFT data.

80G GPUs, utilizing the DeepSpeed stage 2 combined with **Flash Attention [17] and bfloat16 precision** to accelerate multi-GPU training and optimize memory efficiency. The gradient accumulation step is set to 2. Our agent team consists of four backbone models: Qwen3-8B, Qwen3-4B, Qwen2.5-7B, and Qwen2.5-3B. During training, the temperature is set to 1 for each agent to facilitate exploration, while the KL penalty coefficient β is set to 1e-5. We set the maximum prompt length to 1024 tokens and the maximum generation length to 1024 tokens. The multi-agent interaction is limited to a maximum of 5 turns. Specific prompts are provided in Appendix A.2. Additionally, we apply agent dropout to enhance the model's robustness.

LoRA Setting. As reported in Tab 8 of the submitted manuscript, we implement Low-Rank Adaptation (LoRA) using the Hugging Face `peft` library. The LoRA adapters are configured with a rank $r = 8$, a scaling factor $\alpha = 16$, and a dropout rate of 0.05. We adjust the learning rate specifically for LoRA training to 2e-6. Except for these specific adjustments, all other hyperparameters remain consistent with the full fine-tuning configuration described above.

Optimization Strategy. We adopt Group Relative Policy Optimization (GRPO) as our reinforcement learning algorithm. GRPO is selected for its suitability in scenarios involving optimization from a group of candidate outputs. By normalizing rewards against the team's average performance, GRPO provides a stable learning signal for each individual agent, aligning naturally with our multi-agent collaborative generation paradigm.

SFT Data Construction. To enable efficient Supervised Fine-Tuning (SFT), we construct a filtered dataset derived from successful interaction trajectories. As illustrated in our pipeline (see Figure 9), tools, questions, and options are input into the Agent Team. Following the Collaborative Policy Planning Process (CPP), a final answer is generated. We retain a trajectory only if: (1) at least one agent provides the correct answer, and (2) the initial plan remains unchanged throughout the process. We collect 2,000 such initial plans per task. This filtering strategy reduces unnecessary self-correction steps and significantly improves computational efficiency.

Evaluation Setup. For evaluation, all LLMs use a temperature of 0 to ensure deterministic outputs. The agent group composition remains consistent with the training phase (Qwen3-8B, Qwen3-4B, Qwen2.5-7B, and Qwen2.5-3B), totaling approximately 22B parameters. Our reported inference latency (19.8s) is achieved with 4 A100 80G GPUs via parallel processing and bfloat16 precision: (1) we implement parallel processing across the Policy Generation, Execution, and Communication stages, enabling concurrent reasoning and tool invocation across agents (instead of sequential processing); (2) we enforce strict token constraints during reasoning to prompt concise rationales, significantly reducing the decoding overhead. This evaluation can be run on only one A100 80G GPU for each task with partial parallel processing, with about 38.9s per video with 67G VRAM. However, a single A100 80G GPU is insufficient for inference on an MLLM of 72B+ parameters to handle long videos with 100+ sampled frames. When all invoked tools are exhausted or the maximum number of iterations is reached without QA consensus, we directly use Qwen3-8B to generate a summarized answer using the memory of all agents.

Tool Configurations. We tailor the tool set and underlying models for specific benchmarks to maximize performance. The standard tool library includes: *Global Sampling, Video Retrieval, Time Stamp Retrieval, Rough Browser, Fine Browser,* and *Grounding Tool*. For image retrieval, we use ViT-CLIP-B/16 (86M). For video retrieval, we use ASP-CLIP (95M).

- **General Video QA (LongVideoBench, Video-MME, MLVU, Video Holmes, MMR-V):** We utilize the standard tool library. The *Browser* model is instantiated with Qwen2.5-VL-7B, and *Grounding Tool* employs Eagle2.5-8B. The grounding tool is invoked with relatively low frequency. The total parameter count for the toolset is approximately 37B.
- **Video MMMU:** The configuration largely follows that of the General Video QA setup, except that the *Browser* model is upgraded to Qwen3VL-8B-Instruct to handle higher domain-specific demands. The total parameter count is approximately 37B.
- **Video VSIBench (Spatial Tasks):** We introduce a specialized *Spatial Tool* powered by InternVL3.5-8B. For spatial queries, the model autonomously selects between the *Browser* (Qwen2.5-VL-7B) or the *Spatial Tool* for answer generation. The total parameter count is approximately 37B.

Spatial Tool	Baseline	VideoChat-M1
InternVL3.5-8B	56.3	71.9
Qwen2.5VL-7B	35.9	70.1

Table 11. Tool Reliance Ablation on VSIBench

- **Charades-STA (Temporal Grounding):** The model dynamically chooses between the *Browser* and the *Grounding Tool*. Input videos are processed at 2 FPS. If the *Video Retrieval* tool is invoked, the retrieved video clip is subsequently fed into the model for fine-grained grounding. The total number of parameters is approximately 37B. For this dataset, we select up to three consecutive video clips. We first select the clip with the highest similarity; if the similarity of an adjacent clip with the key info exceeds 0.35, we include it as well. This prevents the situation where the answer's grounding time exceeds the duration of the retrieved clip. Additionally, it narrows the retrieval interval and eliminates redundant information, thereby improving performance

Tool Reliance Ablation: To demonstrate that the effectiveness of VideoChat-M1 originates from our Collaborative Policy Planning (CPP) framework rather than reliance on specific SOTA tools, we conducted an additional ablation study on VSIBench (see Tab 11). Specifically, we replaced the specialized 'Spatial Tool' (InternVL) with the general-purpose Qwen2.5-VL-7B. Remarkably, even with this generic backbone, our method retains SOTA performance. It continues to outperform the massive InternVL-3.5-241B, achieving a 34.2% improvement over the baseline. This confirms that our MARL-driven planning paradigm delivers substantial gains, independent of the specific tools employed.