SAGE: Training Smart Any-Horizon Agents for Long Video Reasoning with Reinforcement Learning

Anonymous CVPR submission

Paper ID 5609

Abstract

*As humans, we are natural any-horizon reasoners, i.e., we can decide whether to iteratively skim long videos or watch short ones in full when necessary for a given task. With this in mind, one would expect video reasoning models to reason flexibly across different durations. However, SOTA models are still trained to predict answers in a single turn while processing a large number of frames, akin to watching an entire long video, requiring significant resources. This raises the question: **Is it possible to develop performant any-horizon video reasoning systems?** Inspired by human behavior, we first propose **SAGE**, an agent system that performs multi-turn reasoning on long videos while handling simpler problems in a single turn. Secondly, we introduce an easy synthetic data generation pipeline using Gemini-2.5-Flash to train the orchestrator, **SAGE-MM**, which lies at the core of SAGE. We further propose an effective RL post-training recipe essential for instilling any-horizon reasoning ability in SAGE-MM. Thirdly, we curate **SAGE-Bench** with an average duration of greater than 700 seconds for evaluating video reasoning ability in real-world entertainment use cases. Lastly, we empirically validate the effectiveness of our system, data, and RL recipe, observing notable improvements of up to **6.1%** on open-ended video reasoning tasks, as well as an impressive **8.2%** improvement on videos longer than 10 minutes. We will open-source our system code, data, and checkpoints upon publication.*

1. Introduction

In the last year, there has been a natural shift from developing models for solely image reasoning [7, 8, 12, 23, 24, 39, 40, 52] to also tackling video reasoning [2, 5, 35, 36, 43, 53] in the research community. Among the various model releases, the recent Gemini-2.5 [35] and Qwen3-VL [36] models pushed the frontier in video reasoning due to their ability to perform well on both short and long videos.

Although the aforementioned SOTA models differ in their training data, recipe, and architecture, among other things, they all function in a standard way when reasoning over videos: given a set of sampled frames, output the final answer with a single sequence prediction process, *i.e.*, single turn reasoning. We refer to this line of work as falling under the DIRECT paradigm. Orthogonal to the works mentioned above, a few methods [1, 3, 21, 25, 44, 51] take an agentic route to predicting answers through multi-turn reasoning, falling under the AGENT paradigm.

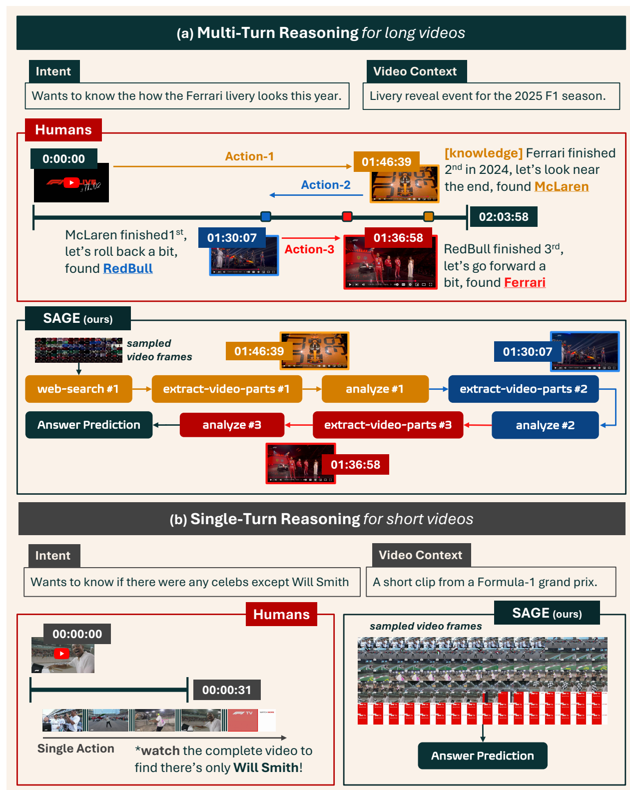


Figure 1. **Human behavior-inspired design of SAGE.** We design SAGE to resemble humans' adaptive reasoning behavior, capable of following a knowledge-driven multi-turn reasoning process using tool calls for long-horizon tasks (Tab. 1) while being able to predict an answer for short-horizon problems directly.

Humans excel at tasks that require multi-turn reasoning. For example, when viewing a 2-hour-long video, as humans, we take an iterative approach to finding the target information (Fig. 1). With the recent overwhelming success of RL post-training for training multi-turn agent systems for long-horizon tasks like software engineering [34, 37, 46], computer-use [33, 36, 47], and deep-research [14, 22, 38], it is natural to expect multi-turn agent systems to do well at long video reasoning. Despite the analogy above, most of the existing long video reasoning systems are still trained following the DIRECT paradigm, even with RL [6, 43].

Motivated by the above realization, we explore the question: **What are the technical challenges toward effectively training video reasoning models under the AGENT paradigm with Reinforcement Learning?** We outline three significant aspects for answering the above question: training data **(A1)**, efficient system design **(A2)**, and RL recipe for multi-turn reasoning **(A3)**.

(A1) The training data for an agent model capable of long video reasoning requires access to high-quality question-answer (QnA) pairs. Collecting QnA pairs for long videos poses a daunting challenge due to their lengthy duration. For example, having a human annotate a single 1-hour-long video can cost approximately $30 on the Prolific platform, making it expensive for data collection at scale. To avoid such high costs, existing works typically employ a synthetic data curation process by iteratively processing 10-30 second-long subclips using models adept at short video understanding to either generate QnA pairs directly [4] or captions followed by QnA pairs using an LLM [5, 6]. Although inexpensive compared to human annotation, the mentioned bottom-up pipeline is slow and resource-intensive — imagine processing 120 subclips for an hour-long video; even with each subclip taking only 10 seconds, it would take 20 minutes to process a single video. Therefore, to save time and money, we leverage the long-context modeling capabilities of Gemini-2.5-Flash to generate synthetic, high-quality QnA pairs with a carefully designed prompt, ensuring the generated questions span the whole video. Moreover, we manually verify over 1700 generated samples and find a low 5% error rate while achieving nearly 100× cost and 10× time savings compared to human annotation and subclip processing pipelines, respectively.

(A2) Existing multi-turn agent systems usually use an LLM/VLM to orchestrate the calls to only a temporal grounder tool [9, 25, 48] to iteratively locate an event over the entire video needed for finding an answer to a given question. However, we posit that attempting to ground an event in the whole video is not always the most effective approach due to the lack of robust temporal grounding models for long videos. For example, knowing the Formula 1 2024 season standings enables intelligent reasoning with a small temporal search space when watching the 2025 season livery reveal event video (Fig. 1a). Motivated by similar use cases, we introduce the **SAGE** (**S**mart **A**ny-horizon a**GE**nt) system for long video reasoning. Particularly, we take a more innovative approach by equipping our system with tools such as web search and speech transcription, in addition to temporal grounding, to ensure that it is adept at not only utilizing visual signals from the video but also leveraging verbal and external knowledge. At the core of our system lies an orchestrator VLM, **SAGE-MM**, responsible for deciding between multi-turn and single-turn behavior for effective any-horizon reasoning. Moreover, guided by the fact that a user typically interacts with videos for entertainment [10, 19], we focus our efforts on verifying the effectiveness of our approach on **SAGE-Bench**, curated with videos from popular YouTube channels to simulate use cases in the daily lives of users. Interestingly, we find existing agent systems to be over-engineered toward answering multiple-choice questions, often underperforming at the open-ended problems under SAGE-Bench (Tab. 4), demonstrating their ineffectiveness for real-world use-cases.

(A3) The variable duration of videos presents a unique challenge to training multi-turn agents. Specifically, during the RL post-training stage, the model should learn to function as an any-horizon agent, i.e., directly output the answer for simple problems while using multi-turn reasoning for harder problems [49]. We believe that the optimization challenge posed by the dynamic nature of videos presents a challenge for training agent models using existing RL recipes, which have been shown to work well for training DIRECT models [6, 17]. Moreover, extending the RLVR techniques [11, 32] to video reasoning presents another challenge due to the task's open-ended nature, which results in a lack of verifiable rewards. A few DIRECT approaches [6, 42] overcome the verifiable reward challenge by training only on MCQ problems and/or using some form of string-overlap metrics [17, 41], rendering them ineffective at open-ended problems (Tab. 4).

To that end, we propose a multi-reward RL recipe that utilizes strong reasoning LLMs [28] to validate the correctness of answers during the RL post-training stage. Moreover, moving away from using string-matching for evaluation, we adopt a universal LLM-as-a-judge evaluation approach to maintain uniformity across our training and evaluation setups. Our RL recipe improves the SFT model by 4.1% and surpasses the base by 5.7%, demonstrating its effectiveness. Moreover, for videos longer than 10 minutes, we observe performance improvements of up to **14.6%** along with **4.8%** for videos shorter than 10 minutes, proving SAGE's effectiveness on any-horizon video reasoning.

In summary, we make the following contributions:

- We propose **SAGE**, an any-horizon agent for long-video reasoning, equipped with a web-search tool for knowledge-driven multi-turn reasoning.
- We introduce a cost-effective synthetic QnA pipeline

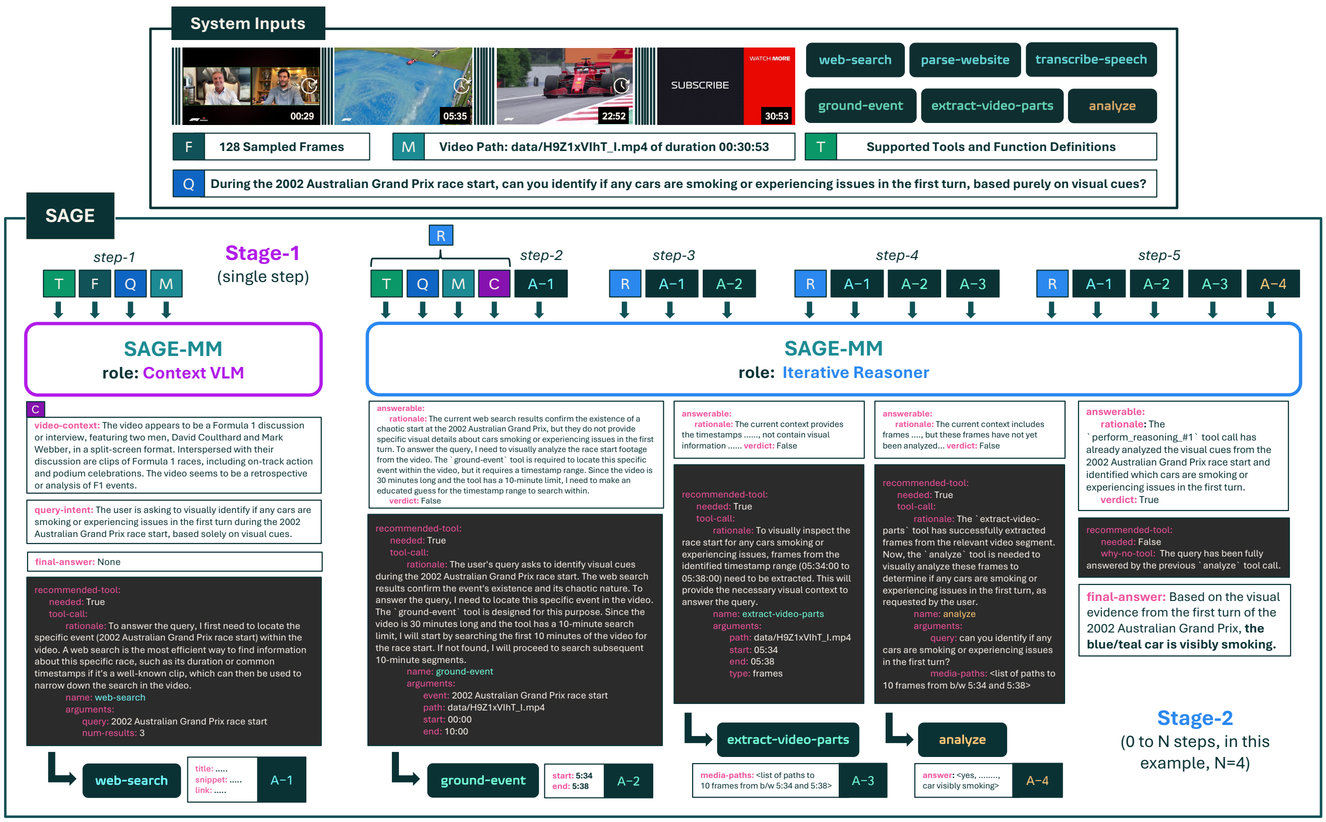


Figure 2. **SAGE Workflow.** Our system accepts four inputs (shown at the top): sampled video frames (F), metadata about the video (M), available tool definitions (T), and the user query (Q). Given these inputs, SAGE operates in two stages based on the role of SAGE-MM. In **Stage-1**, SAGE-MM is responsible for providing information about the video's context (C) along with either a final answer prediction or a tool call to be executed before the next step. At every subsequent step in **Stage-2**, SAGE-MM uses the video context (C) and the tool call results from previous steps to decide either to predict the final answer or call another tool in an iterative reasoning process.

using Gemini-2.5-Flash to train and evaluate our system on entertainment videos for real-world use.
- We train SAGE-MM with an effective RL post-training recipe to instill any-horizon reasoning, demonstrating the scalability of our system design for RL.

2. Related Work

2.1. Long Video Reasoning Agents

Existing long video reasoning agent systems are usually composed of two core components: *an orchestrator*, and *a tool set*, with a temporal grounder being a standard tool among all methods. The orchestrator is responsible for determining the actions to execute while interacting with the available tools within a multi-turn pipeline.

VideoAgent [16] creates a memory using the caption and keyframe features from the video subclips and incorporates tools to retrieve information from memory for reasoning. Similarly, VideoChat-A1 [44] employs keyframe retrieval to perform chain-of-shot reasoning. VideoMind [25] tunes LoRA adapters for the base Qwen2-VL [40] model as a verifier to verify outputs from a separate temporal grounder module before final answer prediction. VideoExplorer [48] optimizes the planner module with DPO [31] for better trajectory reasoning. LVAgent [3] leverages collaboration among multiple MLLMs with iterative reflection and key frame perception to reach the final answer.

In this work, we move beyond over-reliance on temporal grounding by incorporating tools like web search and speech transcription to enable intelligent event localization.

2.2. Reinforcement Learning for Video Reasoning

Following the success of DeepSeek-R1 [11] at using Reinforcement Learning with Verifiable Rewards (RLVR) to improve reasoning abilities in LLMs, various works have tried to leverage GRPO [11, 32] to train DIRECT video reasoning models capable of *thinking* and then answering. Video-R1 [17] follows the optimization approach of DeepSeek-R1 and introduces a contrastive temporal variant of GRPO, comparing answers between inputs with correct and incorrect frame ordering to enforce temporal dependence during reasoning. VideoRFT [41] introduces a semantic-consistency reward between the reasoning trace and video frames. Video-Thinker [42] optimizes the model

tool-name	purpose	arguments	returns
`web-search`	Perform web search using a text query.	query (*str*); num-results (*int*)	List of URL, title, and snippet for search results.
`parse-website`	Parse web data from a given URL.	website-url (*str*)	Parsed HTML content of the website.
`transcribe-speech`	Perform ASR on the video.	path (*str*), start (*str*), end (*str*)	Segment-level verbal transcript between the start and end timestamps.
`ground-event`	Identify timestamps for an event in the video.	event (*str*), path (*str*), start (*str*), end (*str*)	Timestamps for the event between the start and end timestamps.
`extract-video-parts`	Extract frames or subclips between two timestamps.	type (*str*), path (*str*), start (*str*), end (*str*)	List of paths to the saved extracted parts (either frames or a subclip).
`analyze`	Analyze a set of media based on a query.	query (*str*), media-paths (*List[str]*)	Answer to the query.

Table 1. **Supported tools in SAGE.** Our system has access to six tools, including web search (via the Serper-hosted Google Search API), for performing knowledge-driven reasoning. We implement the ground-event and analyze tools using existing MLLMs [36].

to output multiple temporal grounding instances within a single reasoning trace by carefully curating the cold-start SFT dataset. LongVILA-R1 [6] enables the use of thousands of frames during the RL post-training stage with sequence parallelism. All the above methods utilize option-matching and ROUGE metrics to compute rewards, rendering their approach suboptimal for open-ended problems.

We train SAGE-MM to learn the ability to perform any-horizon reasoning using GRPO while leveraging an LLM-as-a-Judge to handle rewards for open-ended problems.

3. Method

In the daily life of a human, entertainment is the primary purpose for interacting with videos [10, 19], from watching sports videos on YouTube to scrolling through hundreds of short reels on Instagram. Therefore, it's only natural to develop video reasoning models, keeping the user's needs in mind. Among those needs, the open-ended interaction holds a vital place. For instance, as shown in Fig. 1, a user would usually ask: *"How does the Ferrari livery look this year?"* as an open-ended question and expect the model to provide an answer in real-time. We introduce **SAGE**, a system designed to answer users' questions while they enjoy entertainment videos. In the following subsections, we present technical details about SAGE (Sec. 3.1), followed by our synthetic data generation pipeline (Sec. 3.2). Lastly, we provide information on training the orchestrator (SAGE-MM) using RL for the system (Sec. 3.3).

3.1. System Design

As shown at the top of Fig. 2, our SAGE expects four inputs: 128 sampled frames from the video (F), metadata about the video (M), available tools' definitions (T), and the user query (Q). SAGE operates in two stages, based on the role of the orchestrator (SAGE-MM) (Fig. 2 bottom):

Stage-1 (role: Context VLM): In this single-step stage, SAGE-MM accepts the system inputs ($T|F|Q|M$) and outputs a JSON action string with required fields:

- *video-context* (C): Information about the video's setting.
- *query-intent*: The intent behind the user's query.
- *recommended-tool*: Information about the next tool call if a final answer cannot be generated at the current step.
- *final-answer*: null if tool call; otherwise predicted answer.

The metadata string (M) comprises information about the video path and duration, which are necessary to predict the arguments for the tool call. We list the supported tools in SAGE in Tab. 1. Notably, unlike previous methods, which either perform temporal grounding over the complete video [25, 48], our SAGE autonomously predicts segment-level timestamps to ground events over a maximum duration of 10 minutes, as we qualitatively found that existing models struggle on longer entertainment videos.

Stage-2 (role: Iterative Reasoner): In this multi-step stage, SAGE-MM accepts the tool call and video context results from all the previous steps, along with the other textual inputs ($T|Q|M$) and decides if the user query can be answered or another tool call is needed. At every step, SAGE-MM outputs a JSON action string with three required fields:

- *answerable*: Whether the query can be answered.
- *recommended-tool*: Information about the next tool call if a final answer cannot be generated at the current step.
- *final-answer*: null if tool call; otherwise predicted answer.

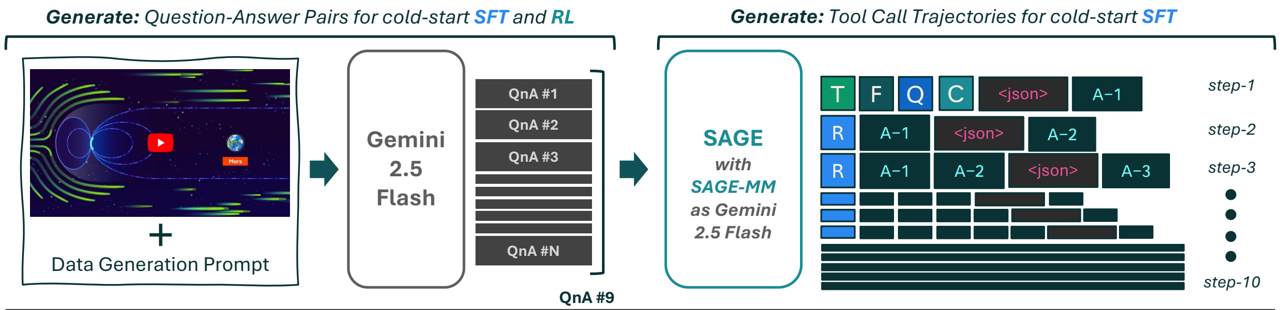
We set the maximum number of steps under stage 2 to nine to prevent indefinite execution length. We provide an example execution graph for SAGE at the bottom of Fig. 2.

3.2. Synthetic Data Generation

We collect videos and shorts from 13 popular YouTube channels across diverse genres, including sports (*Formula1*), food (*ZachChoi*), comedy (*TheDailyShow, MrBean, TheOffice, Friends, fluffyguy, trevornoah*), education (*Vox, kurzgesagt, veritasium, QuantaScienceChannel*), and travel (*WalkingAlice*). Given a video, our synthetic data generation pipeline includes two stages: (i) question-answer (QnA) pair generation using Gemini-2.5-Flash for training and evaluation, and (ii) tool call trajectory generation using SAGE with Gemini-2.5-Flash as the SAGE-MM for cold-start SFT, as shown in Fig. 3.

QnA Pairs. We leverage the long context modeling abilities of Gemini-2.5-Flash [35] to generate questions and answers for a given video in a single pass using a carefully designed prompt. We find that for videos longer than 5 minutes, having the model predict a **`percent_video_parsed`** field is critical to ensure that the generated questions temporally span the complete video, as shown at the bottom of Fig. 3. We generate 10-20 QnA pairs per video.

Tool Call Trajectories. We observe that existing open-source VLMs are not adept at functioning as SAGE-MM right off the shelf, which is a necessity for successful RL post-training. Therefore, we also generate four tool call tra-



{
"index": 9, "type": "open_ended", "difficulty": "hard", "difficulty_rationale": "Requires interpreting abstract visual representations of scientific concepts (magnetic fields) and their application.", "modality": "visual", "modality_rationale": "The magnetic field problem is illustrated visually, and the proposed 'magnetic umbrella' solution is shown without detailed verbal description of its mechanics, requiring visual interpretation.",
"answer": "The fundamental long-term problem for Mars is that its core does not produce a magnetic field, leaving the planet and its atmosphere unprotected from harmful solar radiation and cosmic rays. The proposed solution is to construct an artificial magnetic field. The video depicts this as a large, purple, superconducting ring or 'magnetic umbrella' powered by nuclear facilities. This ring would be positioned at the Mars-Sun L1 point, effectively splashing the solar wind to the sides and deflecting it away from Mars, thus protecting the planet and its newly formed atmosphere.",
"question": "Based on the visual information, what is the fundamental long-term problem for Mars and its inhabitants, and what is the proposed solution depicted to mitigate this issue? Describe how this solution would function.", "options": null, "requires_web_search": false, "why_web_search": null,
"final_timestamp": "00:11:17", "start_timestamp": "00:08:45", "end_timestamp": "00:09:30", "compute_percent_video_parsed": "(570/677)*100 = 84.19",
"percent_video_parsed": 84.19
}

Figure 3. **Synthetic Data Generation Pipeline.** We leverage Gemini-2.5-Flash to generate 10-20 QnA pairs, covering the full temporal span of the video. We find that instructing the model to predict a `percent_video_parsed` field for every QnA pair helps in enforcing proper coverage. We use a SAGE with Gemini-2.5-Flash as the orchestrator to synthesize tool call trajectories for a cold-start SFT stage.

	0–60	60–180	180–300	300–600	600–1200	1200–2400	2400+	total
#videos	1642	1770	546	606	1067	461	567	6659
#QnA	20.2k	23.0k	8.0k	9.4k	22.2k	7.1k	9.4k	99.1k
#actions	43.4k	43.9k	38.6k	49.6k	115.0k	52.2k	75.0k	417.7k

Table 2. **Training Data Statistics.** We generate over 99k questions for more than 6600 videos from popular YouTube channels.

jectories for each question and use input-action pairs from unique trajectories to create a cold-start SFT dataset to fine-tune our own SAGE-MM model before the RL post-training stage. Tab. 2 lists statistics for our training data.

3.3. RL Post Training

We use GRPO [11, 32] as the policy optimization algorithm during the RL post-training stage for trajectory-level optimization. Specifically, during the rollout generation, the ith action rollout trajectory for a given input set $S_1 = \{T, F, M, Q\}$ is represented by τ_i. Therefore, we can formulate τ_i as a sequence of state-action pairs $\forall j \in [0, N]$:

$$\begin{aligned} \tau_i &= \left[(S_1, A_1), (S_2, A_2), \ldots, (S_N, A_N)\right], \\ A_j &= \textbf{SAGE-MM}(S_j), \\ S_{j+1} &= \{T, Q, M, C, A_1 ... A_j\} \end{aligned} \tag{1}$$

During the advantage computation step in GRPO, we assign a single scalar reward R_i to every action in the trajectory τ_i with N steps. The reward consists of (i) step-level rewards s_j collected at each step, and (ii) a final accuracy reward a_N at the end of the trajectory. The resulting reward R_i is then uniformly assigned to all actions in τ_i:

$$\begin{aligned} R_i &= (s_1 + s_2 + s_3 + ... + s_N) \; + \; a_N \\ r(A_1) &= r(A_2) = ... = r(A_N) = R_i \end{aligned} \tag{2}$$

Note that we can assign final rewards to all steps because rollout generation is synchronous, *i.e.*, advantages are computed only after all trajectories are completed in a batch.

Step-Level Rewards. The reward (s_j) for a step j in a trajectory is a sum of four scores:

- *format*: Encourages producing a JSON action string with only the required fields.

$$s_{\text{format}} = \begin{cases} +0.05, & \text{if JSON contains only required fields} \\ -0.10, & \text{otherwise} \end{cases}$$

- *reasonable-tool*: Encourages the model to perform sensible multi-step tool usage. Specifically, at each step, we ask GPT-4o to judge whether the current tool call is rational, given the question and the previous tool calls.

$$s_{\text{reasonable-tool}} = \begin{cases} +0.10, & \text{if current tool call is reasonable} \\ -0.10, & \text{otherwise} \end{cases}$$

- *args-repeat*: Penalizes repetitive tool call arguments.

$$s_{\text{args-repeat}} = -0.05 \cdot \sqrt{\text{num-repetitions}}$$

- *args-valid*: Penalizes invalid tool-call arguments.

$$s_{\text{args-valid}} = \begin{cases} -0.1, & \text{if arguments are invalid} \\ 0, & \text{otherwise} \end{cases}$$

We set the values for the step rewards such that the accumulated step-level reward for a trajectory with 10 steps would be comparable to the accuracy reward.

Accuracy Reward. We compute the outcome reward for a trajectory of length N based on the final answer prediction using an LLM judge (GPT-4o [28]) to obtain a binary verdict indicating correctness at the last step.

Overall	**Count**	**Modality**	**Count**
# samples	1744	visual only	1216
– # mcq	802	verbal only	134
– # open-ended	942	visual + verbal *(both)*	394
Duration (*avg*: 727 sec.)			
Bucket (sec.)	**Count**	**Bucket (sec.)**	**Count**
0–60	261	600–1200	484
60–180	390	1200–2400	147
180–300	116	2400+	180
300–600	186		

Table 3. **SAGE-Bench Statistics.** Our evaluation set holds 1744 manually verified samples spanning diverse durations, with an emphasis on questions that require visual information to answer.

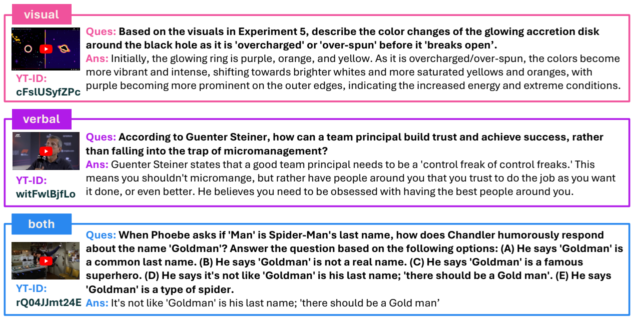


Figure 4. **Qualitative Samples from SAGE-Bench.** Our evaluation set contains questions that mirror what a user might naturally ask while or after watching the corresponding video.

$$a_N = \begin{cases} -2.0, & \text{if JSON action string is invalid} \\ -0.5, & \text{if wrong answer and } N \geq 1 \\ +1.25, & \text{if correct answer and visual tools in } \tau_i \\ +1.0, & \text{otherwise} \end{cases}$$

During training and inference, we set $N_{max} = 10$ by default. However, during the RL stage, we find that setting $N_{max} = 5$ for the first 100 steps is necessary for stable training, aligned with findings from a concurrent work for training long-horizon LLM agents [45]. Moreover, we penalize the model for predicting a wrong answer with tool calls to compensate for the positive step-level rewards while enforcing the any-horizon nature, i.e., making the model capable of predicting a direct answer. Conversely, we grant a slightly higher reward of +1.25 when the answer is correct and SAGE used visual tools (`extract-video-parts` or `ground-event`), reflecting the higher difficulty and importance of getting these tool calls right.

4. Experiments

For our experiments, we finetune three MLLMs, using both cold-start SFT (denoted by **SFT**) and RL post-training (denoted by **RL**) stages to obtain the SAGE-MM: Qwen2.5-VL-7B-Instruct [2], Qwen3-VL-4B-Instruct [36], and Qwen3-VL-8B-Instruct [36]. By default, we use the Qwen3-VL-8B-Instruct as the base SAGE-MM for all our ablations. We implement the `transcribe-speech` tool using the Whisper-large-v3 [30] model. We use the Qwen3-VL-30B-A3B-Instruct [36] model to perform temporal grounding and reasoning with the `ground-event` and `analyze` tools, respectively.

4.1. Implementation Details

Training Data. As shown in Tab. 2, we synthesize 99.1k training questions from 6659 videos, covering a wide range of durations. Additionally, we generate 417.7k state–action pairs for **SFT**. For **RL**, we construct a dataset of 7.68k samples, filtered using synthetic tool-call trajectories, where half of the samples required tool calls and the other half had single-turn responses, promoting any-horizon reasoning.

Training Recipe. During **SFT**, we train our model for one epoch with a batch size of 64 and an initial learning rate of $1e^{-5}$ with a linear decay scheduler. We sample 128 frames at 2 FPS and use a temporal pooling factor of 2, setting the maximum and minimum numbers of tokens per frame to 128 and 192, respectively. During **RL**, we use a batch size of 16 and rollout eight action trajectories per sample. We use an initial learning rate of $1e^{-6}$ with a cosine decay scheduler. We set the KL-divergence loss coefficient to 0.005. Note that we report numbers for the model trained for 480 steps during the **RL** stage. We train all our models using 16× NVIDIA H100 GPUs during both **SFT** and **RL**.

Evaluation. We evaluate all DIRECT baselines with 128 sampled frames as input, comparable to SAGE-MM's input setting. Moreover, we also pass the video transcript as extra context to the DIRECT baselines for fair comparison. For AGENT baselines, we follow their recommended setup. By default, we use LLM-as-judge (GPT-4o) for evaluating all models on both open-ended and MCQ problems. We set the temperature to 0.0 for all evaluations. However, because the action strings must follow a strict JSON schema, SAGE-MM occasionally produces malformed outputs. In such cases, we regenerate the response with a temperature of 0.7 for up to four attempts, which may lead to non-deterministic behavior during inference. We serve all supported models using vLLM [20] during evaluation.

We share more details, including the system, data generation, and evaluation prompts, in the appendix.

4.2. SAGE-Bench

Driven by the limitations of current video reasoning benchmarks due to their purely MCQ nature, we curate our own evaluation set, **SAGE-Bench**, with a focus on open-ended questions simulating the needs for real-world use-cases for entertainment videos. We begin by sampling a subset of synthetic QnA pairs that is strictly disjoint from the training

Method	Orchestrator	Video Reasoning Mode		overall	mcq	open-ended	both	verbal	visual
		train	eval	(1744)	(802)	(944)	(394)	(134)	(1216)
Gemini-2.5-Flash [35]	N/A	DIRECT	DIRECT	68.1	77.2	60.4	74.9	71.6	65.5
SAGE-Flash (ours)	**SAGE-MM:** Gemini-2.5-Flash	N/A	AGENT	**71.3**	**81.2**	**62.9**	**76.3**	**84.3**	**68.3**
GPT-4o [28]	N/A	DIRECT	DIRECT	71.6	80.9	63.6	75.1	73.9	70.1
SAGE-Flash (ours)	**SAGE-MM:** GPT-4o	N/A	AGENT	**73.4**	**81.0**	**66.9**	**78.2**	**79.9**	**71.1**
Video-Thinker-7B [42]	N/A	DIRECT	DIRECT	41.3	70.1	16.8	48.2	41.8	39.0
LongVILA-R1-7B [6]	N/A	DIRECT	DIRECT	52.6	68.8	38.7	57.6	64.9	49.6
VideoRFT-7B [41]	N/A	DIRECT	DIRECT	55.3	71.6	41.4	65.2	67.2	50.7
Video-R1-7B [17]	N/A	DIRECT	DIRECT	57.6	73.6	43.9	67.5	67.2	53.3
VideoAgent [16]	GPT-4o	N/A	AGENT	42.0	52.6	32.9	42.6	29.1	43.2
LVAgent [3]	InternVL-8/72B [7] + LLaVA-Video-72B [50]	N/A	AGENT	49.7	70.5	32.1	54.1	48.5	48.4
VideoMind-7B [25]	VideoMind-Planner	AGENT	AGENT	50.0	69.7	33.2	50.8	41.8	50.7
VideoExplorer-7B [48]	VideoExplorer-Planner	AGENT	AGENT	50.1	69.6	35.1	52.0	40.2	51.3
Qwen2.5-VL-7B-Instruct [2]	N/A	DIRECT	DIRECT	58.6	74.2	45.4	65.8	68.7	55.2
SAGE (ours)	**SAGE-MM:** Qwen2.5-VL-7B-Instruct [+SFT]	AGENT	AGENT	61.1	74.1	50.1	62.9	**69.4**	59.6
SAGE (ours)	**SAGE-MM:** Qwen2.5-VL-7B-Instruct [+SFT] [+RL]	AGENT	AGENT	**63.4**	**77.2**	**51.5**	**66.1**	65.7	**62.2**
Qwen3-VL-4B-Instruct [36]	N/A	DIRECT	DIRECT	62.7	75.8	51.6	69.3	66.4	60.2
SAGE (ours)	**SAGE-MM:** Qwen3-VL-4B-Instruct [+SFT]	AGENT	AGENT	64.6	77.3	53.7	66.2	67.2	63.7
SAGE (ours)	**SAGE-MM:** Qwen3-VL-4B-Instruct [+SFT] [+RL]	AGENT	AGENT	**68.4**	**81.3**	**57.4**	**78.4**	**80.6**	**63.8**
Qwen3-VL-8B-Instruct [36]	N/A	DIRECT	DIRECT	64.9	77.7	54.0	72.8	68.7	61.9
SAGE (ours)	**SAGE-MM:** Qwen3-VL-8B-Instruct [+SFT]	AGENT	AGENT	63.9	77.4	52.4	72.3	74.6	60.0
SAGE (ours)	**SAGE-MM:** Qwen3-VL-8B-Instruct [+SFT] [+RL]	AGENT	AGENT	**68.0**	**82.6**	**55.6**	**75.4**	**82.8**	**64.0**
SAGE-Flash (ours)	**SAGE-MM:** Qwen3-VL-8B-Instruct [+SFT] [+RL]	AGENT	AGENT	**71.8**	**82.8**	**62.4**	**75.1**	**79.1**	**69.9**

Table 4. **Comparison to Baselines.** Using closed-source Gemini-2.5-Flash [35] and GPT-4o [28] as SAGE-MM improves upon the base models, showing the effectiveness of our system design. Our trained SAGE-MM also shows consistent improvements over all the baselines. SAGE-Flash refers to the setting where we use Gemini-2.5-Flash as the backend model for the `ground-event` and `analyze` tools. Existing AGENT systems exhibit considerably worse performance on open-ended problems compared to our SAGE.

	SAGE-MM	overall	0–600s	600+s
	training	(1473)	(842)	(631)
Qwen2.5-VL-7B-Instruct [2]	N/A	32.7	37.8	25.8
VideoRFT-7B [41]	N/A	30.4	33.5	26.2
Video-R1-7B [17]	N/A	31.5	36.0	25.8
SAGE (ours)	**SFT**	28.3	30.3	24.3
SAGE (ours)	**SFT** + **RL**	32.0	34.7	**28.4**
SAGE-Flash (ours)	**SFT** + **RL**	**32.9**	35.6	**29.0**

Table 5. **Performance on MINERVA [27].** Our SAGE shows significant improvements on videos longer than 600 seconds.

train strategy	train mode	eval mode	mcq	open-ended	overall
Qwen3-VL-4B-Instruct		DIRECT	75.8	51.5	62.7
Qwen3-VL-4B-Thinking		DIRECT	75.3	48.6	60.1
SFT	DIRECT	DIRECT	**83.2**	51.1	65.8
SFT + **RL**	DIRECT	DIRECT	83.0	52.0	66.3
SFT (ours)	AGENT	AGENT	77.3	53.7	64.6
SFT + **RL** (ours)	AGENT	AGENT	81.3	**57.4**	**68.4**

Table 6. **Training Mode.** Our AGENT system performs better than the DIRECT baseline, with **RL** playing a critical role in the former's success, specifically on open-ended problems.

set (videos can be common) and manually verifying each sample for correctness. Notably, fewer than 5% of the samples required edits during verification, demonstrating that our synthetic data generation pipeline produces high-quality data at low cost. The statistics of SAGE-Bench are provided in Tab. 3. We also provide qualitative examples in Fig. 4.

4.3. Main Results

In Tab. 4, we compare our SAGE to DIRECT video reasoning methods, including models trained without RL post-training, like Qwen3-VL-4/8B-Instruct [36], and RL-tuned models, like Video-R1 [17]. We also evaluate AGENT systems like VideoMind [25] and VideoExplorer [48].

Effective System Design. We separately evaluate the performance of our system with two API-based models SAGE-MM: Gemini-2.5-Flash [35] and GPT-4o [28]. For this setting, we use Gemini-2.5-Flash as the backend model for the `ground-event` and `analyze` tools; therefore, we denote the system as **SAGE-Flash**. We observe improvements of up to **3.2%** over the base API models, validating the effectiveness of our system design.

Effective Training Recipe. As shown in Tab. 4, our SAGE with a trained SAGE-MM achieves notable improvements across different base MLLMs. Specifically, SAGE surpasses Qwen2.5-VL-7B-Instruct by **4.8%** overall, with substantial gains of **+6.1%** on open-ended and **+7.0%** on visual questions, underscoring the effectiveness of our training strategy. Interestingly, models such as Video-R1 [17], VideoRFT [41], and VideoExplorer [48], despite employing finetuned Qwen2.5-VL-7B-Instruct backbones, underperform relative to the base model, particularly on open-ended questions. Moreover, as shown in the last row of Tab. 4, SAGE-Flash further improves upon SAGE by **3.8%**, even outperforming the Gemini-2.5-Flash variant of SAGE-

Method	Model	Eval Mode	0-60	60-180	180-300	300-600	600-1200	1200-2400	2400+	overall
			(261)	(390)	(116)	(186)	(484)	(147)	(180)	(1744)
Qwen3-VL (baseline)	Qwen3-VL-8B-Instruct	DIRECT	73.9	72.3	**81.9**	71.5	55.0	59.2	47.5	64.9
SAGE (ours)	Qwen3-VL-8B-Instruct [+SFT]	AGENT	74.3	68.1	75.0	72.0	56.8	55.8	48.1	63.9
SAGE (ours)	**SAGE-MM**: Qwen3-VL-8B-Instruct [+SFT] [+RL]	AGENT	**78.5** (+4.6)	70.3 (-2.0)	77.4 (-4.5)	**72.6** (+1.1)	**63.2** (+8.2)	**61.9** (+2.7)	**53.8** (+6.3)	**68.1** (+3.2)
SAGE-Flash (ours)	**SAGE-MM**: Qwen3-VL-8B-Instruct [+SFT] [+RL]	AGENT	**77.8** (+3.9)	**73.6** (+1.3)	80.2 (-1.7)	**76.3** (+4.8)	**69.6** (+14.6)	**68.0** (+8.8)	**56.2** (+8.7)	**71.8** (+6.9)

Table 7. **Duration-wise Accuracy.** Our SAGE shows significant improvements on samples belonging to buckets with duration longer than 600 seconds, with even more improvements when using Gemini-2.5-Flash as a tool with SAGE-Flash.

system	SAGE-MM	single-turn		multi-turn		overall
	Qwen3-VL-8B-Instruct *(base)*	count	acc.	count	acc.	acc.
SAGE-Flash	Gemini-2.5-Flash (expert)	859	76.9	885	66.0	71.3
SAGE	[+SFT] (ours)	706	79.0	1038	53.7	64.6
SAGE	[+SFT] [+RL] (ours)	948	79.6	796	54.3	68.0
SAGE-Flash	[+SFT] [+RL] (ours)	940	78.8	804	63.4	71.8

Table 8. **Any-Horizon Reasoning. RL** refines the tool's overcalling behavior of the **SFT** model, resulting in a distribution closer to the expert Gemini-2.5-Flash and thus, improved performance.

MM. This indicates that our finetuned SAGE-MM not only learns to invoke tools effectively but also benefits from more accurate tool outputs.

Additionally, we report results with Qwen2.5-VL-7B-Instruct based SAGE-MM on MINERVA [27], a complex video reasoning benchmark that covers domains such as sports, short films, and cooking videos. As shown in Tab. 5, our SAGE shows an improvement of **2.6%** on long videos (duration >600 seconds) compared to the base model while outperforming other reasoning models, validating the effectiveness of our approach for long video reasoning.

4.4. Ablations

Training Mode. In Tab. 6, we finetune a Qwen3-VL-4B-Instruct model on the synthetic QnA pairs with DIRECT answering mode under the same data setting. We observe that our AGENT training recipe outperforms the direct baseline, underscoring the effectiveness of our approach. Specifically, while training the DIRECT baseline with **SFT**, we supervise the model with only the correct final answer and not the tool call actions. During **RL**, we use only the accuracy reward to train the DIRECT baseline.

Duration-wise accuracy. We report duration-wise accuracies on SAGE-Bench in Tab. 7. Notably, our SAGE exhibits substantially higher gains on longer videos compared to shorter ones, achieving a remarkable **8.2%** improvement in the 600–1200 seconds bucket. Incorporating Gemini-2.5-Flash as a tool (SAGE-Flash) further boosts this gain to **14.6%**, with more than 8% improvements in the 1200–2400 and 2400+ second buckets as well.

Any-Horizon Reasoning. A core aspect of system's design is to enable any-horizon reasoning, *i.e.*, it is adept at multi-turn reasoning and also directly outputting an answer in a single step. As shown in Tab. 8, our **SFT** model, distilled from the expert Gemini-2.5-Flash, inherits strong single-turn ability but tends to show signs of overcalling tools. Incorporating RL further refines this behavior while improving single-turn and multi-turn accuracies.

	overall	both	verbal	visual
SAGE (ours)	**68.0**	**75.4**	**82.8**	64.0
w/o ground-event	67.3	72.3	79.9	**64.3**
w/o web-search/parse-website	65.5	70.1	80.6	62.4
w/o analyze	63.4	70.6	80.6	59.1
w/o extract-video-parts	63.0	70.8	79.9	58.6
w/o transcribe-speech	62.5	66.8	46.3	62.9

Table 9. **Dropping Tools during inference.** All tools are critical to the success of SAGE as a system, with the extract-video-parts and transcribe-speech being the most important ones for answering the visual and verbal/both questions, respectively, as expected.

Importance of Supported Tools. We ablate the contribution of each tool in Tab. 9. Dropping the *transcribe-speech*, *extract-video-parts*, and *analyze* tools leads to the most significant performance decline, highlighting their fundamental role in long-video reasoning. In contrast, removing the *ground-event* tool results in only a minor drop, likely due to the tool's inherent inaccuracy. This observation underscores the need for developing better temporal grounding modules.

5. Conclusion

In this work, we introduced SAGE, an any-horizon reasoning system for long video reasoning. We also designed a cost-effective synthetic data generation pipeline for training and evaluating with the target use case of aiding users with open-ended queries while they watch entertainment videos in mind. Through extensive experiments, we validated the effectiveness of our system design and RL post-training recipe at enabling any-horizon reasoning, with considerable gains on videos longer than 10 minutes. We hope our work can serve as a vital proof-of-concept toward training practical AGENT systems for long video reasoning in the future, moving away from purely DIRECT approaches.

Future Work. Looking ahead, training on data from broader domains to strengthen the inherent single-turn reasoning ability, especially for short videos is a natural advancement. In addition, integrating more advanced agent-centric policy optimization algorithms [13, 15, 18] for **RL** presents a promising avenue. Finally, empowering the system to select the appropriate tools and synthesize new ones when necessary [26, 29] represents an exciting direction.

References

[1] Kirolos Ataallah, Xiaoqian Shen, Eslam Abdelrahman, Essam Sleiman, Mingchen Zhuge, Jian Ding, Deyao Zhu, Jürgen Schmidhuber, and Mohamed Elhoseiny. Goldfish: Vision-language understanding of arbitrarily long videos, 2024. 1

[2] Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, Humen Zhong, Yuanzhi Zhu, Mingkun Yang, Zhaohai Li, Jianqiang Wan, Pengfei Wang, Wei Ding, Zheren Fu, Yiheng Xu, Jiabo Ye, Xi Zhang, Tianbao Xie, Zesen Cheng, Hang Zhang, Zhibo Yang, Haiyang Xu, and Junyang Lin. Qwen2.5-vl technical report. *arXiv*, 2025. 1, 6, 7

[3] 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*, 2025. 1, 3, 7

[4] 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. *arXiv*, 2025. 2

[5] Yukang Chen, Fuzhao Xue, Dacheng Li, Qinghao Hu, Ligeng Zhu, Xiuyu Li, Yunhao Fang, Haotian Tang, Shang Yang, Zhijian Liu, Ethan He, Hongxu Yin, Pavlo Molchanov, Jan Kautz, Linxi Fan, Yuke Zhu, Yao Lu, and Song Han. Longvila: Scaling long-context visual language models for long videos. *arXiv*, 2024. 1, 2

[6] Yukang Chen, Wei Huang, Baifeng Shi, Qinghao Hu, Hanrong Ye, Ligeng Zhu, Zhijian Liu, Pavlo Molchanov, Jan Kautz, Xiaojuan Qi, Sifei Liu, Hongxu Yin, Yao Lu, and Song Han. Scaling rl to long videos. In *NeurIPS*, 2025. 2, 4, 7

[7] Zhe Chen, Jiannan Wu, Wenhai Wang, Weijie Su, Guo Chen, Sen Xing, Muyan Zhong, Qinglong Zhang, Xizhou Zhu, Lewei Lu, Bin Li, Ping Luo, Tong Lu, Yu Qiao, and Jifeng Dai. Internvl: Scaling up vision foundation models and aligning for generic visual-linguistic tasks. *arXiv*, 2023. 1, 7

[8] 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*, 2024. 1

[9] Jisheng Dang, Huilin Song, Junbin Xiao, Bimei Wang, Han Peng, Haoxuan Li, Xun Yang, Meng Wang, and Tat-Seng Chua. Mupa: Towards multi-path agentic reasoning for grounded video question answering. *arXiv*, 2025. 2

[10] Claire Dannenbaum. 5 facts about americans and youtube. Pew Research Center, 2025. 2, 4

[11] DeepSeek-AI. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv*, 2025. 2, 3, 5

[12] Matt Deitke, Christopher Clark, Sangho Lee, Rohun Tripathi, Yue Yang, Jae Sung Park, Mohammadreza Salehi, Niklas Muennighoff, Kyle Lo, Luca Soldaini, Jiasen Lu, Taira Anderson, Erin Bransom, Kiana Ehsani, Huong Ngo, YenSung Chen, Ajay Patel, Mark Yatskar, Chris Callison-Burch, Andrew Head, Rose Hendrix, Favyen Bastani, Eli VanderBilt, Nathan Lambert, Yvonne Chou, Arnavi Chheda, Jenna Sparks, Sam Skjonsberg, Michael Schmitz, Aaron Sarnat, Byron Bischoff, Pete Walsh, Chris Newell, Piper Wolters, Tanmay Gupta, Kuo-Hao Zeng, Jon Borchardt, Dirk Groeneveld, Jen Dumas, Crystal Nam, Sophie Lebrecht, Caitlin Wittlif, Carissa Schoenick, Oscar Michel, Ranjay Krishna, Luca Weihs, Noah A. Smith, Hannaneh Hajishirzi, Ross Girshick, Ali Farhadi, and Aniruddha Kembhavi. Molmo and pixmo: Open weights and open data for state-of-the-art multimodal models. In *CVPR*, 2025. 1

[13] Guanting Dong, Licheng Bao, Zhongyuan Wang, Kangzhi Zhao, Xiaoxi Li, Jiajie Jin, Jinghan Yang, Hangyu Mao, Fuzheng Zhang, Kun Gai, Guorui Zhou, Yutao Zhu, Ji-Rong Wen, and Zhicheng Dou. Agentic entropy-balanced policy optimization. *arXiv*, 2025. 8

[14] Guanting Dong, Yifei Chen, Xiaoxi Li, Jiajie Jin, Hongjin Qian, Yutao Zhu, Hangyu Mao, Guorui Zhou, Zhicheng Dou, and Ji-Rong Wen. Tool-star: Empowering llm-brained multi-tool reasoner via reinforcement learning. *arXiv*, 2025. 2

[15] Guanting Dong, Hangyu Mao, Kai Ma, Licheng Bao, Yifei Chen, Zhongyuan Wang, Zhongxia Chen, Jiazhen Du, Huiyang Wang, Fuzheng Zhang, Guorui Zhou, Yutao Zhu, Ji-Rong Wen, and Zhicheng Dou. Agentic reinforced policy optimization. *arXiv*, 2025. 8

[16] 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 *ECCV*, 2024. 3, 7

[17] Kaituo Feng, Kaixiong Gong, Bohao Li, Zonghao Guo, Yibing Wang, Tianshuo Peng, Benyou Wang, and Xiangyu Yue. Video-r1: Reinforcing video reasoning in mllms. In *NeurIPS*, 2025. 2, 3, 7

[18] Lang Feng, Zhenghai Xue, Tingcong Liu, and Bo An. Group-in-group policy optimization for llm agent training. *arXiv*, 2025. 8

[19] Sharon Hafuta. Video marketing statistics — the ultimate video marketing stats report. Wix Blog, 2025. 2, 4

[20] Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph E. Gonzalez, Hao Zhang, and Ion Stoica. Efficient memory management for large language model serving with pagedattention. In *Proceedings of the ACM SIGOPS 29th Symposium on Operating Systems Principles*, 2023. 6

[21] Boyi Li, Ligeng Zhu, Ran Tian, Shuhan Tan, Yuxiao Chen, Yao Lu, Yin Cui, Sushant Veer, Max Ehrlich, Jonah Philion, et al. Wolf: Dense video captioning with a world summarization framework. *Transactions on Machine Learning Research*, 2025. 1

[22] Xiaoxi Li, Wenxiang Jiao, Jiarui Jin, Guanting Dong, Jiajie Jin, Yinuo Wang, Hao Wang, Yutao Zhu, Ji-Rong Wen, Yuan Lu, and Zhicheng Dou. Deepagent: A general reasoning agent with scalable toolsets. *arXiv*, 2025. 2

[23] Ji Lin, Hongxu Yin, Wei Ping, Yao Lu, Pavlo Molchanov, Andrew Tao, Huizi Mao, Jan Kautz, Mohammad Shoeybi, and Song Han. Vila: On pre-training for visual language models. *arXiv*, 2023. 1

[24] Haotian Liu, Chunyuan Li, Yuheng Li, and Yong Jae Lee. Improved baselines with visual instruction tuning. In *CVPR*, 2024. 1

[25] Ye Liu, Kevin Qinghong Lin, Chang Wen Chen, and Mike Zheng Shou. Videomind: A chain-of-lora agent for long video reasoning. *arXiv*, 2025. 1, 2, 3, 4, 7

[26] Damiano Marsili, Rohun Agrawal, Yisong Yue, and Georgia Gkioxari. Visual agentic ai for spatial reasoning with a dynamic api. In *CVPR*, 2025. 8

[27] Arsha Nagrani, Sachit Menon, Ahmet Iscen, Shyamal Buch, Ramin Mehran, Nilpa Jha, Anja Hauth, Yukun Zhu, Carl Vondrick, Mikhail Sirotenko, Cordelia Schmid, and Tobias Weyand. Minerva: Evaluating complex video reasoning. *arXiv*, 2025. 7, 8

[28] OpenAI. Gpt-4o system card. *arXiv*, 2024. 2, 5, 7

[29] Viraj Prabhu, Yutong Dai, Matthew Fernandez, Jing Gu, Krithika Ramakrishnan, Yanqi Luo, Silvio Savarese, Caiming Xiong, Junnan Li, Zeyuan Chen, and Ran Xu. Walt: Web agents that learn tools. *arXiv*, 2025. 8

[30] Alec Radford, Jong Wook Kim, Tao Xu, Greg Brockman, Christine McLeavey, and Ilya Sutskever. Robust speech recognition via large-scale weak supervision, 2022. 6

[31] Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model. In *NeurIPS*, 2023. 3

[32] Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Y.K. Li Mingchuan Zhang, Y. Wu, and Daya Guo. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv*, 2024. 2, 3, 5

[33] ByteDance Seed Team. Seed1.5-vl technical report. *arXiv*, 2025. 2

[34] FAIR CodeGen team, Jade Copet, Quentin Carbonneaux, Gal Cohen, Jonas Gehring, Jacob Kahn, Jannik Kossen, Felix Kreuk, Emily McMilin, Michel Meyer, Yuxiang Wei, David Zhang, Kunhao Zheng, Jordi Armengol-Estapé, Pedram Bashiri, Maximilian Beck, Pierre Chambon, Abhishek Charnalia, Chris Cummins, Juliette Decugis, Zacharias V. Fisches, François Fleuret, Fabian Gloeckle, Alex Gu, Michael Hassid, Daniel Haziza, Badr Youbi Idrissi, Christian Keller, Rahul Kindi, Hugh Leather, Gallil Maimon, Aram Markosyan, Francisco Massa, Pierre-Emmanuel Mazaré, Vegard Mella, Naila Murray, Keyur Muzumdar, Peter O'Hearn, Matteo Pagliardini, Dmitrii Pedchenko, Tal Remez, Volker Seeker, Marco Selvi, Oren Sultan, Sida Wang, Luca Wehrstedt, Ori Yoran, Lingming Zhang, Taco Cohen, Yossi Adi, and Gabriel Synnaeve. Cwm: An open-weights llm for research on code generation with world models. *arXiv*, 2025. 2

[35] Gemini Team. Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities. *arXiv*, 2025. 1, 4, 7

[36] Qwen Team. Qwen3-vl: Sharper vision, deeper thought, broader action, 2025. 1, 2, 4, 6, 7

[37] Qwen Team. Qwen3 technical report. *arXiv*, 2025. 2

[38] Tongyi DeepResearch Team, Baixuan Li, Bo Zhang, Dingchu Zhang, Fei Huang, Guangyu Li, Guoxin Chen, Huifeng Yin, Jialong Wu, Jingren Zhou, et al. Tongyi deepresearch technical report. *arXiv*, 2025. 2

[39] Shengbang Tong, Ellis Brown, Penghao Wu, Sanghyun Woo, Manoj Middepogu, Sai Charitha Akula, Jihan Yang, Shusheng Yang, Adithya Iyer, Xichen Pan, Austin Wang, Rob Fergus, Yann LeCun, and Saining Xie. Cambrian-1: A fully open, vision-centric exploration of multimodal llms. In *NeurIPS*, 2024. 1

[40] 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*, 2024. 1, 3

[41] Qi Wang, Yanrui Yu, Ye Yuan, Rui Mao, and Tianfei Zhou. Videorft: Incentivizing video reasoning capa-

bility in mllms via reinforced fine-tuning. *arXiv*, 2025. 2, 3, 7

[42] Shijian Wang, Jiarui Jin, Xingjian Wang, Linxin Song, Runhao Fu, Hecheng Wang, Zongyuan Ge, Yuan Lu, and Xuelian Cheng. Video-thinker: Sparking” thinking with videos” via reinforcement learning. *arXiv*, 2025. 2, 3, 7

[43] 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*, 2025. 1, 2

[44] 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*, 2025. 1, 3

[45] Zhiheng Xi, Jixuan Huang, Chenyang Liao, Baodai Huang, Honglin Guo, Jiaqi Liu, Rui Zheng, Junjie Ye, Jiazheng Zhang, Wenxiang Chen, Wei He, Yiwen Ding, Guanyu Li, Zehui Chen, Zhengyin Du, Xuesong Yao, Yufei Xu, Jiecao Chen, Tao Gui, Zuxuan Wu, Qi Zhang, Xuanjing Huang, and Yu-Gang Jiang. Agentgym-rl: Training llm agents for long-horizon decision making through multi-turn reinforcement learning. *arXiv*, 2025. 6

[46] John Yang, Carlos E Jimenez, Alexander Wettig, Kilian Lieret, Shunyu Yao, Karthik R Narasimhan, and Ofir Press. SWE-agent: Agent-computer interfaces enable automated software engineering. In *NeurIPS*, 2024. 2

[47] Yan Yang, Dongxu Li, Yutong Dai, Yuhao Yang, Ziyang Luo, Zirui Zhao, Zhiyuan Hu, Junzhe Huang, Amrita Saha, Zeyuan Chen, Ran Xu, Liyuan Pan, Silvio Savarese, Caiming Xiong, and Junnan Li. Gta1: Gui test-time scaling agent. *arXiv*, 2025. 2

[48] Huaying Yuan, Zheng Liu, Junjie Zhou, Hongjin Qian, Yan Shu, Nicu Sebe, Ji-Rong Wen, and Zhicheng Dou. Think with videos for agentic long-video understanding. In *ICLR*, 2025. 2, 3, 4, 7

[49] Zizheng Zhan, Ken Deng, Huaixi Tang, Wen Xiang, Kun Wu, Weihao Li, Wenqiang Zhu, Jingxuan Xu, Lecheng Huang, Zongxian Feng, Shaojie Wang, Shangpeng Yan, Xuxing Chen, Jiaheng Liu, Zhongyuan Peng, Zuchen Gao, Haoyang Huang, Xiaojiang Zhang, Jinghui Wang, Zheng Lin, Mengtong Li, Huiming Wang, Ziqi Zhan, Yanan Wu, Yuanxing Zhang, Jian Yang, Guang Chen, Haotian Zhang, Bin Chen, and Bing Yu. Kat-v1: Kwai-autothink technical report. *arXiv*, 2025. 2

[50] Yuanhan Zhang, Jinming Wu, Wei Li, Bo Li, Zejun Ma, Ziwei Liu, and Chunyuan Li. Video instruction tuning with synthetic data. *arXiv*, 2024. 7

[51] Zhuo Zhi, Qiangqiang Wu, Minghe shen, Wenbo Li, Yinchuan Li, Kun Shao, and Kaiwen Zhou. Videoagent2: Enhancing the llm-based agent system for long-form video understanding by uncertainty-aware cot. *arXiv*, 2025. 1

[52] Deyao Zhu, Jun Chen, Xiaoqian Shen, Xiang Li, and Mohamed Elhoseiny. Minigpt-4: Enhancing vision-language understanding with advanced large language models. *arXiv*, 2023. 1

[53] 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*, 2025. 1