VideoTIR: Accurate Understanding for Long Videos with Efficient Tool-Integrated Reasoning

Zhe Gao[1*], Shiyu Shen[2*], Taifeng Chai[3], Weinong Wang[4†], Haotian Xu[5], Xing Wu[4], Wenbin Li[1], Qi Fan[1✉], Yang Gao[1], and Dacheng Tao[6]

[1] Nanjing University, Nanjing, China
fanqi@nju.edu.cn
[2] Nankai University, Tianjin, China
[3] East China Normal University, Shanghai, China
[4] Tencent Hunyuan, China
[5] MiroMind, USA
[6] Nanyang Technological University, Singapore

Abstract. Existing Multimodal Large Language Models (MLLMs) often suffer from hallucinations in long video understanding (LVU), primarily due to the imbalance between textual and visual tokens. Observing that MLLMs handle shorter but more accurate visual inputs well, recent LVU works alleviate hallucinations by automatically parsing the vast visual data into manageable segments that can be effectively processed by MLLMs. SFT-based tool-calling methods can serve this purpose, but they typically require vast amounts of fine-grained, high-quality data and suffer from constrained tool-calling trajectories. We propose a novel VideoTIR that leverages Reinforcement Learning (RL) to encourage proper usage of comprehensive multi-level toolkits for efficient long video understanding. VideoTIR explores both Zero-RL and SFT cold-starting to enable MLLMs to retrieve and focus on meaningful video segments/images/regions, enhancing long video understanding both accurately and efficiently. To reduce redundant tool-calling in the early RL-stage and accelerate convergence, we propose Toolkit Action Grouped Policy Optimization (TAGPO), which enhances the efficiency of the calling process through the finer stepwise reward assignment. Additionally, we develop a sandbox-based trajectory synthesis framework to generate high-quality trajectory data. Extensive experiments on three long-video QA benchmarks demonstrate the effectiveness and efficiency of our method.

Keywords: Long Video Understanding · Vision-Language Models · Agentic RL · Tool-Integrated Reasoning

1 Introduction

Long video understanding (LVU) [27,33,34] aims to answer user questions based on long videos, whose durations typically span several minutes to multiple hours.

*Equal contribution. ✉Corresponding author. †Project leader.

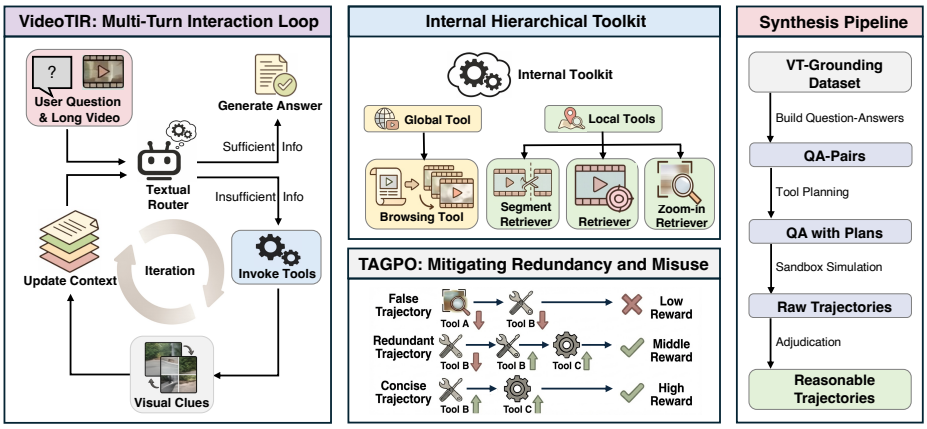


Fig. 1: We propose VideoTIR , a tool-integrated reasoning framework that flexibly and hierarchically retrieves relevant video segments through endogenous tool invocation to support long-video understanding. Furthermore, to enable SFT cold start, we introduce a sandbox-based trajectory synthesis framework. We also present TAGPO to address the inefficiency in early-stage RL exploration caused by tool misuse and overuse.

To tackle this challenge, this field has increasingly turned to Multimodal Large Language Models (MLLMs [1, 4, 9, 26, 31]), leveraging their powerful vision-language reasoning capabilities to interpret complex, long-form video content.

Recently, long video understanding methods can be broadly categorized into non-TIR frame selection strategies and tool-integrated reasoning (TIR) approaches. Non-TIR methods, such as frame selection strategies [11], select relevant frames before MLLM final reasoning. While effective in reducing redundant visual tokens, these approaches operate outside the reasoning loop and lack adaptive refinement during inference. In contrast, TIR methods integrate retrieval tools into the reasoning process, enabling iterative evidence acquisition. Existing TIR designs generally fall into two categories. The first category relies on heavy external retrieval toolkits [6,18,20,28], which often involve complex pipelines and fixed action workflows. Such rule-based or externally dependent systems suffer from limited generalization and high interaction overhead. The second category designs lightweight internal retrieval mechanisms by prompting MLLMs to output textual timestamps that guide segment extraction [37,41]. However, as illustrated in [5], current MLLMs are not sufficiently pre-trained on video datasets with fine-grained spatiotemporal annotations. Consequently, both zero-RL strategies [37] and RL-after-SFT cold-start designs [41] struggle to substantially improve localization precision due to the limited capabilities of the base model. As a result, retrieval often becomes redundant or inefficient, with repeated tool invocation over similar segments. Moreover, these methods rely on a single-tool toolkit, which limits their ability to flexibly handle video question answering tasks.

Therefore, we aim to design a diverse and efficient internal toolkit that leverages the MLLM's own encoders. In addition, we propose an RL framework to

effectively coordinate these tools, enabling flexible and question-aware tool invocation. Detailedly, we propose VideoTIR , a novel approach for long video understanding using Tool-Integrated Reinforcement Learning. VideoTIR comprises two key components: a comprehensive toolkit containing multiple specialized tools for diverse sub-tasks and a textual router that dynamically selects a subset of candidate tools from the toolkit. The toolkit includes diverse specialized tools for both global and local video processing tasks. For global tasks, we introduce a browsing tool that progressively increases video spatial resolution and sampling rate to acquire coarse-to-fine visual evidence accordingly. For localized queries, we introduce a fine-grained retrieval tool chain comprising clip-segment, frame-pick, and zoom-in tools, enabling multi-granularity and flexible retrieval to support temporally and spatially specific tool-calling requirements. This multi-level toolkit provides flexible support for adaptive decision-making across heterogeneous questions in long videos. Leveraging the extensive toolkit, the textual router utilizes the reasoning capabilities of MLLMs to understand long videos by pre-scanning them, interpreting questions, and formulating subsequent action.

However, we observe that single-tool Tool-Integrated RL methods [37,45,47], which typically adopt GRPO as the policy update strategy and design rewards primarily around final answer accuracy and tool invocation, often lead to both **misuse** and **overuse** of tools within the multi-tool settings. Misuse refers to invoking multiple tools while still producing incorrect answers. This behavior reflects reward hacking and increases the trajectory exploration burden during the early stages of RL training, resulting in excessive tool invocation. In contrast, overuse in our context refers to the tendency of MLLMs to retrieve the finest-grained frames within the retrieval tool chain, even when coarser-grained retrieval would be sufficient to answer the query. To address these issues, we propose Toolkit Action Grouped Policy Optimization (TAGPO), a novel RL optimization algorithm that improves the efficiency of tool utilization through stepwise reward assignment. Specifically, TAGPO groups invocations of the same retrieval sub-tool and designs an advantage estimation scheme that assigns higher importance to sub-tool invocations closer to the final answer, thereby mitigating tool overuse. Moreover, this advantage formulation assigns lower advantage to trajectories that reuse prior experience—i.e., invocation chains that led to correct answers in other rollouts—but still produce incorrect outputs, compared to trajectories that explore new tools, thereby reducing tool misuse.

Moreover, multi-tool invocation introduces new challenges for accurate instruction following in multi-tool agents. Multi-tool reasoning typically relies on heterogeneous interactions across various tools, where tool prompts may be invoked far from the initial system prompt, increasing the risk of instruction drift or forgetting. Although supervised fine-tuning (SFT) may alleviate this issue, it typically requires extensive high-quality multi-tool video datasets, which are often difficult to obtain. To address this problem, we propose a data synthesis framework to bootstrap multi-instruction-following capabilities for RL agents. We leverage an external MLLM (*e.g.*, GLM-4.5V [9]) to generate question–answer

pairs and explore possible tool invocation orders based on a video–text grounding dataset. We then construct a sandbox to enable MLLMs to generate additional synthetic training data, including intermediate prompts and synthesized trajectories. The synthesized data can be used to pretrain instruction-following capabilities and tool-use policies for agents.

Our main contributions are summarized as follows (Fig. 1):

- **Multi-turn multi-internal-tool agent for LVU.** We propose a multi-turn, multi-tool agent framework to effectively and efficiently understand long videos by leveraging internal tools.
- **Invocation-aware reinforcement learning.** We design a novel toolkit action grouped policy optimization to explicitly encourage concise and effective tool calls, balancing video exploration efficiency and reasoning accuracy.
- **Multi-tool trajectory synthesis.** We develop and open-source a sandbox-based trajectory synthesis framework that equips LVU models with instruction-following and tool-calling abilities before RL fine-tuning.

2 Related Works

Multimodal Models in Video Understanding. Video understanding encompasses tasks such as action recognition, temporal localization and retrieval, captioning, and summarization. Early deep learning approaches typically targeted single tasks, with architectures designed to model temporal dependencies and extract spatiotemporal features [25, 34]. With the advent of MLLMs, the field has shifted toward unified, task-agnostic frameworks. These systems commonly employ an LLM as the reasoning core while aligning video representations through a visual encoder and a multimodal connector [2]. Representative examples include Video-LLaMA [44], which integrates image/video and audio branches with an LLM; and Video-ChatGPT [21] and Video-LLaVA [16], which build instruction-following conversational capabilities from curated video QA and caption datasets. More recent unified models [26] process images and videos using shared tokenization and temporal positional encodings, achieving competitive zero- and few-shot performance across diverse benchmarks. Comprehensive evaluations [8, 15] show that MLLMs perform strongly on short- to medium-length clips for perception and reasoning but degrade as clip duration increases, indicating a need for more robust temporal and multimodal integration.

Despite this progress, applying MLLMs directly to LVU exposes key limitations. First, constrained context windows necessitate aggressive subsampling, which obscures cross-scene dependencies and amplifies duration-induced errors [22]. Second, temporal reasoning and precise localization are fragile in the absence of explicit, retrievable memory, leading to ordering and causal mistakes [35]. Third, long-horizon datasets and annotations remain scarce and expensive, biasing instruction corpora toward short clips [10].

Tool-Based Models in Video Understanding. Recent progress in long video understanding can be organized along the spectrum from frame-selection strategies to fully tool-integrated reasoning (TIR) agents. *Frame-selection methods* se-

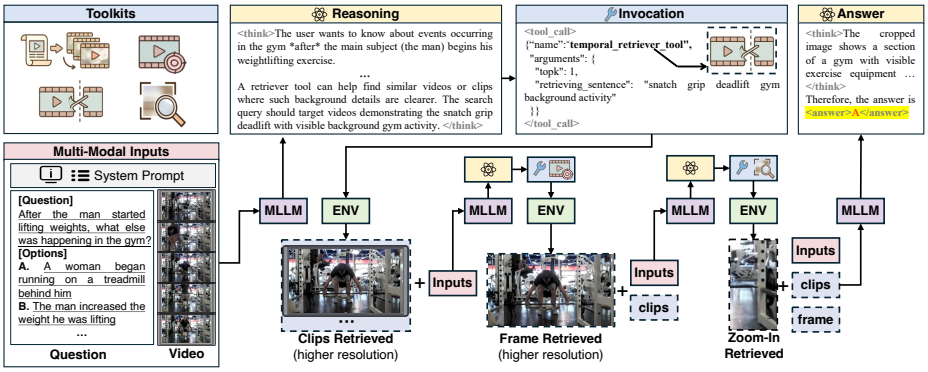


Fig. 2: Framework of our methods. VideoTIR adopts a multi-turn manner to deal with the users' input videos and questions. When the model fails to conclude an answer based on current visual information, it calls tools to perceive the absent vision clues, which is combined with the former context as the input for the next-turn reasoning.

lect relevant frames or segments prior to final reasoning, operating largely outside the reasoning loop. Such approaches reduce redundant visual tokens and improve efficiency, but lack iterative refinement and adaptive evidence acquisition. *Tool-integrated reasoning agents*, inspired by ReAct-style paradigms [23, 42], instead interleave planning, tool invocation, and reasoning. Within TIR approaches, existing systems can be broadly categorized into external-tool pipelines and lightweight internal retrieval designs. External-tool agents couple MLLMs with structured retrieval modules such as ASR, OCR, segment localizers, or memory systems [6, 12, 20]. These systems often rely on pre-defined workflows, multi-agent coordination [3, 13], or structured memory decomposition [30, 38]. While effective for task-specific benchmarks, their heavy external dependencies and fixed tool repertoires may limit generalization across diverse long-video scenarios. More recent works explore lightweight internal retrieval mechanisms, where MLLMs output textual timestamps or structured queries to guide segment extraction [37, 40, 41].

Although more tightly integrated, these methods depend heavily on the temporal grounding capability of the underlying MLLMs. As illustrated in Fig. 3, timestamp-based internal retrieval often entangles localization prediction with reasoning, making performance sensitive to the intrinsic temporal limitations of base models.

3 Methodology

In this section, we first present an overview of VideoTIR in Section 3.1. Then we introduce the design of the multi-modal hierarchical toolkits in Section 3.2. Section 3.3 focuses on the Zero-RL procedure with respect to the advantage esti-

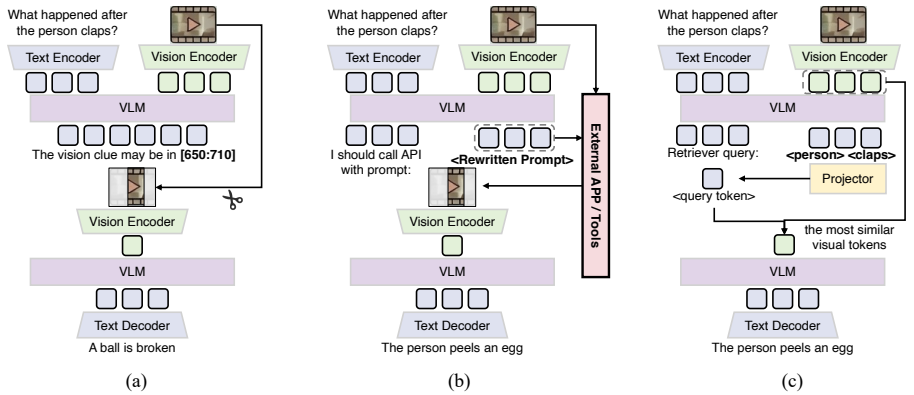


Fig. 3: Comparison of tool-integrated reasoning (TIR) designs for video understanding. (a) Methods such as [37, 41] adopt a paradigm in which the VLM outputs timestamps in text form for subsequent video clipping. (b) Alternatively, some methods rely on heavyweight external tools, incurring substantial interaction costs. In contrast, VideoTIR leverages the intrinsic encoding structure of the VLM to design internal retrieval tools, selecting visual cues from the video based on feature similarity.

mator and rewards design. Section 3.4 present a general framework to synthesize high-quality trajectories from normal video QA-data or even pure video data.

3.1 Overview

Different from MLLMs that process video-QA pairs within a single turn to provide the answer, human usually adapt a coarse-to-fine manner to divide the task in several sequential steps and then conquer by hierarchically adding fine-grained visual clues to decide whether or not to give the accurate answer. Therefore, as illustrated in Fig. 2, VideoTIR adopts a multi-turn manner to deal with the users' input videos and question. In considering of efficiency, videos are parsed with low resolution and fps initially and VideoTIR calls the textual router tool to decide whether the current visual information is enough to conduct an answer, or the router of using what tools to fetch the missing visual information corresponding to the question. In the latter cases, the fine-grained visual information generated by tools is combined with the former contexts and again parsed to the textual router tool. Such loops stop if the model conducts an answer or reaches the predefined number of max turns.

3.2 Multi-Modal Hierarchical Retrieval Toolkits

Textual Router. Our multi-modal hierarchical toolkit is designed to mimic human efficient retrieval behaviors. Humans usually first read the question and grasp the intent, and then roughly browse the videos with such intents. Therefore, we design a textual router tool as illustrated in Fig. 4 to pay attention to

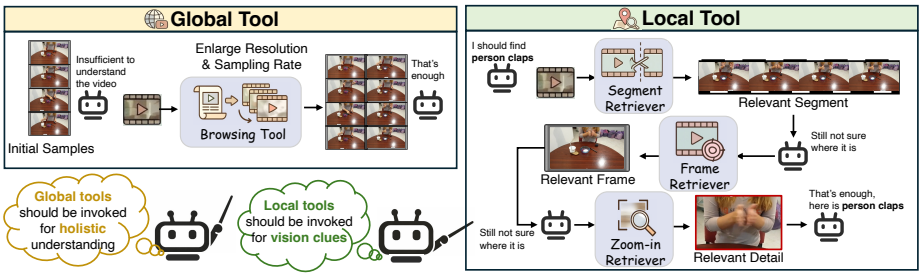


Fig. 4: Hierarchical Visual Toolkits containing both Global and Local Tools. When there's a need for more information, the textual router calls global-level browsing tools for the general questions and detail-level tools for the questions targeting at finer perception of the videos.

the textual information and decide the potential corresponding objects or scenarios to answer the question. Combined with the current visual information, the router takes different behavior in different situations:

- **Provide answers** when the corresponding visual information is enough.
- Otherwise, **Call browsing tools** when the questions' intents are at a global understanding of the videos and **temporal-spatial grounding tools** when the intents target at specific visual clues.

Browsing Tool. When the question aims at an entire understanding (*e.g.*, what is the whole video about?) and the current visual information is not able to provide an answer, humans usually tend to gather more temporal and spatial information simultaneously by slower browsing and more frame-level attention. Therefore, our browsing tool simulates such a manner by enlarging both the resolution and the fps rate compared with before.

Temporal-Spatial Grounding Tool. When the questions need fine-grained visual clues (*e.g.*, what happens between A and B?), humans usually need to locate the related parts of the videos and tend to ignore the other parts. Specifically, such locating behavior always performs a coarse-to-fine manner that first roughly locates temporally and then dives into the potential video segments for detailed spatial clues. Therefore, our temporal-spatial grounding tool is designed as a flexible hierarchical chains of retrievers with early stops:

- **Segment Retriever.** Based on the text search results planned by the textual router tool, internal tools are called to locate relevant segments in the original video by selecting the video tokens with cosine similarity comparing to textual query tokens generated by MLLMs.
- **Frame Retriever.** Input a video segment, and this tool will retrieve the keyframes in that segment that are most relevant to the question and the retrieving sentence.
- **Zoom-in Retriever.** Input the image modality, and the tool will crop out the most relevant area to answer.

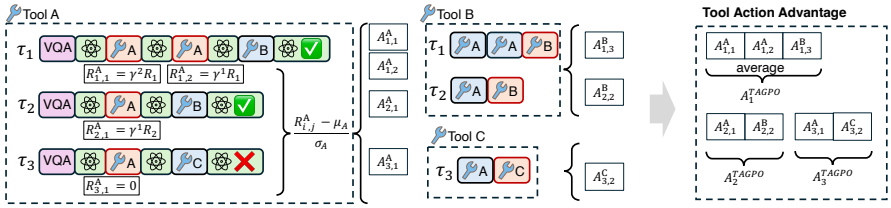


Fig. 5: Visualization of Tool Action Advantage. We define rewards for each tool-calling action to penalize redundancy. The toolkit action advantage is the average of the tool advantages.

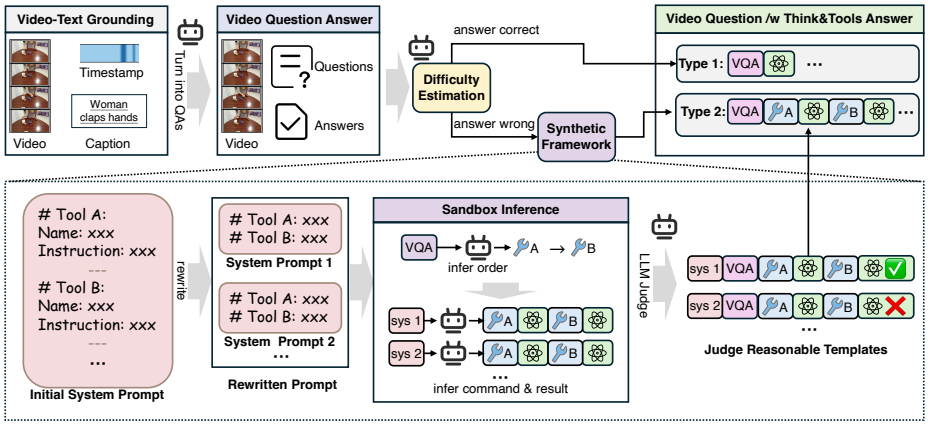


Fig. 6: Framework of Data Synthesis. For video-text grounding datasets, we first convert them into QA datasets. Then, for the easy question, we synthesize trajectories that answer directly with no tool calls. For hard questions that model answers wrong, they are processed through a sandbox to generate tool calling trajectories. Finally, a large LLM is used to judge the rationality of the trajectories and we only keep the reasonable ones.

Details of the toolkit are in the Supplementary Materials.

3.3 RL Algorithm Design

We denote the model-generated sequence as τ_i, $i \in \{1, 2, ..., N\}$, N is the rollout time. The episode-level (global) reward comprises three components: an accuracy reward $R_{\text{acc}}(\tau_i) = 1$ if the final answer is correct; a format reward $R_{\text{fmt}}(\tau_i) = 1$ if the tool-call string is parsable; and a tool-use bonus $\mathbb{I}_{R_{\text{acc}}(\tau_i)>0} \cdot R_{\text{tool}}(\tau_i)$ to encourage appropriate tool invocation [47]. To be specific:

$$R_i := R_{\text{acc}}(\tau_i) + R_{\text{fmt}}(\tau_i) + \mathbb{I}_{R_{\text{acc}}(\tau_i)>0}\, R_{\text{tool}}(\tau_i), \quad A_i^{\text{GRPO}} := \frac{R_i - \mu(R)}{\sigma(R)}. \quad (1)$$

where R_i denotes the episode reward of τ_i, A_i^{GRPO} is the batch normalized episode advantage, and $\mu(R_{(\cdot)}), \sigma(R_{(\cdot)})$ are the mean and standard deviation of rewards computed over the N rollouts.

Toolkit Action Grouped Policy Optimization. To assess the conciseness and precision of tool use, we move beyond episode-level credit assignment and evaluate each tool invocation. For every call, we define a local reward guided by a simple principle: penalize redundancy in successful episodes while permitting exploration in failed episodes. We then compute a per-call advantage, such that redundant invocations receive negative advantage whereas exploratory calls in failed rollouts receive null advantage. The trajectory level toolkit action advantage is obtained by averaging these per-call advantages across all invocations in the trajectory. The resulting advantage allocation is illustrated in Fig. 5.

Assume that $\tau_i = (t_i^1, t_i^2, ..., t_i^{L_i})$, where $t_i^j \in \{\text{tool}_1, \text{tool}_2, ...\text{tool}_K\}$ denotes a tool invocation, K is the number of available tools, L_i s the number of tool calls in τ_i. We define the per-call reward for invoking tool_k at the j_{th} step in trajectory τ_i as :

$$R_{i,j}^k := \mathbb{I}_{R_{\text{acc}}(\tau_i)>0} \cdot \gamma^{L_i - j} R_i \tag{2}$$

where i,j can only take the indices at which tool_k is called. $\gamma \in (0, 1]$ is a predefined decay coefficient. This yields intuitive behavior in comparative cases:

- If two successful trajectories (A, A, B) and (A, B) receive the same episode reward, the second A in (A, A, B) obtains the same per-call reward as the single A in (A, B); the earlier, redundant A in (A, A, B) receives a smaller reward due to the additional decay, and therefore a negative advantage under intra-tool normalization. This punishes the **overuse**.
- If (A, A, B) succeeds but (A, B) fails, all calls in (A, A, B) receive positive credit while all calls in (A, B) receive zero, reinforcing the necessity of the repeated A.
- If both (A, A, B) and (A, B) fail, no per-call credit is assigned, yielding no relative advantage and thereby encouraging broader exploration of other tool combinations. This punishes the **misuse**.

The overall toolkit action advantage of τ_i is calculated as follows:

$$A_{i,j}^k := \frac{R_{i,j}^k - \mu(R_{(\cdot,\cdot)}^k)}{\sigma(R_{(\cdot,\cdot)}^k)} \quad A_i^{\text{TAGPO}} := \mu\Big(A_{i,(\cdot)}^{(\cdot)}\Big) \tag{3}$$

where $A_{i,j}^k$ denotes the advantage of the invocation of tool_k at j_{th} step in τ_i, $\mu(R_{(\cdot,\cdot)}^k), \sigma(R_{(\cdot,\cdot)}^k)$ is the mean and standard deviation of all the rewards of calling tool_k, A_i^{TAGPO} is the mean of the per-call advantages for the trajectory.

We update the policy using a composite advantage:

$$\nabla_\theta \mathcal{L} = -\sum_i (A_i^{GRPO} + A_i^{TAGPO}) \nabla_\theta \log \pi_\theta(a_i|s_i) \tag{4}$$

where a_i denotes the action (output tokens) of the base model, s_i denotes the environment state.

3.4 Pre-Rollout and Trajectory Synthesis

Most video-QA datasets provide only question–answer pairs and ground-truth responses, lacking fine-grained annotations about tool usage or intermediate reasoning. Consequently, during RL training the reward signal is largely restricted to answer accuracy and call-string parseability, offering little supervision on the appropriateness of tool usage. To address this issue, as illustrated in Fig. 6, we introduce a multi-stage trajectory synthesis framework.

We leverage an external MLLM (e.g., GLM-4.5V) to synthesize both tool invocation sequences and pseudo environment feedback:

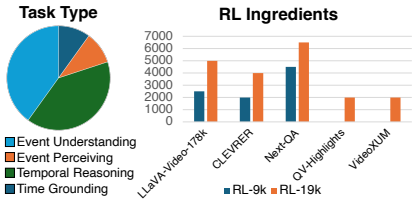
1. **Tool necessity filtering.** Using available video–text grounding, we generate multiple QA instances and prompt the MLLM to answer directly. Items solvable without tools are put aside.
2. **Tool order prediction.** For remaining items, we supply the VQA and the initial system prompt with toolkit template, and ask the MLLM to predict a plausible sequence of tool invocations.
3. **System prompt rewriting.** We rewrite the system prompt —especially the toolkit template —to diversify calling syntax and reasoning behaviors.
4. **Trajectory generation.** Conditioned on the rewritten prompt and the predicted tool order, the MLLM produces stepwise reasoning traces, explicit invocation commands, and expected environment feedback at each step within a sandboxed simulator.
5. **Trajectory adjudication.** The MLLM then ranks candidate trajectories by conciseness and precision, enabling selection of high-quality exemplars.

4 Experiments

4.1 Setups

Benchmarks. We evaluated our method on three video understanding benchmarks, encompassing videos of varying lengths and capability systems.

- **MVBench.** This benchmark is built from 11 short or medium duration video datasets containing STAR [32], PAXION [29], CLEVRER [43], and *etc* and evaluate the spatial-temporal reasoning ability by 20 diverse designed tasks. We adopt the full dataset of 4500 samples.
- **Video-MME.** Video-MME is a comprehensive benchmark for evaluating multimodal large models for video understanding. It covers 6 major visual domains and 30 subcategories, with video lengths ranging from 11 seconds to 1 hour. In addition to frame information, it also includes subtitles and audio to comprehensively examine the model's cross-modal and long-term temporal understanding capabilities. We adopt the whole 2700 samples for evaluation.
- **LongVideoBench.** This benchmark focuses on evaluating models' ability of event perception and relation understanding on medium or long videos. We adopt the validation set of this benchmark that contains 1377 samples.



Dataset	Task Range	Durations	Type
LLaVA-Video-178k	Temporal, Spatial, Causual, Speed understanding	Short(<3min)	QA
CLEVRER	counterfactual, descriptive, explanatory, predictive	Short(<1min)	QA
Next-QA	Causal Questions, Temperal Questions Descriptive Questions	Short(<2min)	QA
QV-Highlights	Part Captioning	Long(5-30min)	VTG
VideoXUM	Part Captioning	Medium(0.2-12min)	VTG

Fig. 7: Distribution and Task Range of Curated Datasets (VideoSIAM is not included in). We selected 4 general tasks that potentially need tools for finer perception. We also synthesize high-quality trajectories as the SFT dataset for 3B model cold starting.

Baseline Models. We adopt **Qwen2.5-VL** [1] as our base models for post-training. Specifically, we conduct experiments on two model scales, Qwen2.5-VL-3B and Qwen2.5-VL-7B, to evaluate the effectiveness and scalability of our method across different parameter regimes. Qwen2.5-VL is a strong open-source vision-language model that supports both image and video inputs and demonstrates competitive performance on a wide range of multimodal understanding benchmarks. Therefore, we focus on Qwen2.5-VL rather than earlier Qwen-VL-2 or more recent Qwen-VL-3 variants to ensure architectural consistency and controlled comparisons with other relative methods.

Details of Training and Inference. We use *LLaMaFactory* [46] as the training framework to Normal-SFT our 3B model with 8 NVIDIA 4090s and Random-Noising-SFT the 3B model with 2 AMD MI308Xs. In the agent-RL training stage, we use 8 NVIDIA H20s and *Volcano Engine Reinforcement Learning* (VeRL) as the training framework to train the 7B Zero-RL model and use 4 AMD MI308Xs to train the 3B Zero-RL and Cold-Started models.

We set the number of max turns as 4 and rollout per sample as 8. Also, we following the default max frames as 32 per video and max pixels of the initial turn per frame as $14*14*28*28$ (196 tokens), and total pixels as $10*14*14*28*28$ (1960 tokens). We adopt different coefficient of the rewards between zero-RL settings and SFT cold-start settings. This is motivated by observation that the ability of following formality instruction after SFT is at a high initial range and therefore the model should learn more from other rewards. We also adopt online rollout filtering [39] to ensure training stability.

While evaluating, the frame resolution settings "Low" in Tab. 1 are the same as training (1960 video tokens) and "high" refers to 1 fps sampling (at max 32 frames and 16384 video tokens). All the evaluations are 8 NVIDIA 4090s or 4 AMD MI308Xs.

RL Dataset Preparation and Curation. As shown in Fig. 7, we curated our dataset from the LLaVa-Video-155k, CLEVRER, PerceptionTest, and VideoSIAM. The first 3 datasets are curated by difficulty estimation and ground-truth verification, that do 10 times of examination and select the medium difficulty samples

Table 1: Main Results of the comparison and our method. Our method surpasses the base Qwen2.5-VL-7B model and many comparison methods that require more frames and higher resolution (Res.) as inputs.
† Following [48], we observe that enabling thinking mode in small MLLMs (<7B) degrades LVU performance. For fairness, we also report thinking-mode results.
‡ These methods are SFTed on approximately 100k TIR and non-TIR trajectories.

Model	Size	Input	Res.	Thinking	V-MME$_{\text{w/o sub}}$				MVBench	LVB$_{\text{val}}$
					Short	Medium	Long	Mean		
InternVL3 [4]	2B	16-frame	High	×†	67.3	54.2	46.2	55.9	63.9	-
Qwen2-VL [26]	2B	16-frame	High	×	58.1	46.0	41.3	48.5	57.3	46.3
Qwen2.5-VL [1]	3B	16-frame	High	×	63.4	50.8	47.7	54.0	60.9	51.8
Video-LLaVa [44]	7B	8-frame	High	✓†	-	-	-	39.9	34.1	39.1
VideoChat2 [30]	7B	16-frame	High	✓	-	-	-	39.2	51.1	39.3
Video-R1 [7]	7B	16-frame	High	✓	-	-	-	57.4	62.7	-
Video-XL [24]	7B	128-frame	High	✓	-	-	-	55.5	55.3	50.7
Video-MTR [37]	7B	32-frame	High	✓	-	-	-	59.0	-	-
LongVT-RL ‡ [41]	7B	64-frame	High	✓	-	-	-	**66.1**	-	-
Qwen2.5-VL	7B	16-frame	High	×	63.1	50.9	47.3	53.8	60.6	51.8
Qwen2.5-VL	7B	16-frame	High	✓	61.9	46.7	39.8	49.5	55.2	47.5
Qwen2.5-VL	7B	8-frame	Low	✓	-	-	-	35.0	40.3	28.1
w/ Tool (Zero-Shot)	7B	8-frame	Low	✓	52.2	46.4	30.1	42.9	47.8	38.6
w/ Tool + GRPO	7B	8-frame	Low	✓	64.3	49.7	48.3	54.1	56.9	50.9
w/ Tool + TAGPO	7B	8-frame	Low	✓	65.1	50.4	47.1	54.2	56.0	51.3
w/ Tool + TAGPO	7B	16-frame	High	✓	67.2	55.8	50.7	57.9	**64.3**	**53.1**

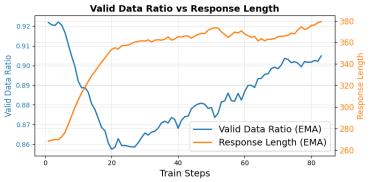
that answer correctly in range (3,7). For 3B model, we select 4.5k RL training samples and for 7B model 9k samples are selected.

SFT Trajectory Synthesis and Curation. We synthesize high-quality trajectories from video-QA datasets NextQA [35] and NextGQA [36], Perceptiontest and VTG datasets VideoXUM [17], QV-Highlights [14]. We adopt 4.5k of the tool-invoking trajectories combine with 3 times more textual CoT data from VideoR1 [7] as the cold start SFT dataset (17,384 in total). We do not apply extra difficulty estimation since this procedure has already done in the synthetic framework. The detailed ingredients are shown in Fig. 7 more details of the synthetic framework are in the supplementary materials.

4.2 Main Results

Short- and Medium-form Video Understanding. On short or medium duration videos, our methods surpass the base model Qwen2.5-VL-7B. Compared with VideoMTR, the increasing performance on MVBench and LongVideoBench illustrates that external efficient tools can achieve better visual information acquisition.

Long-form Video Understanding. Surprisingly, our model shows more increase on these long form videos. Also, the performance of VideoMTR is constrained since such constrained sampling rates lost too much temporal information, leading to difficulties retrieving clues between the sparse sampling frames by only MLLMs themselves. By contrast, our model uses efficient lightweight



(a) Valid data ratio vs. response length.

(b) TAGPO vs. episode-level GRPO.

Fig. 8: Training dynamics analysis. (a) At early stages, response length increases while format quality drops, indicating the model prioritizes rational tool exploration. Once response length stabilizes, format quality gradually improves, suggesting a balance between rationality and formality. (b) TAGPO accelerates valid tool learning. The valid tool reward rises significantly faster than episode-level GRPO, reducing the steps required to exceed 0.45 by approximately 50%.

Table 2: Formatting quality under different prompts and training strategies. The 3B base model produces near-zero valid formatting, while format-oriented SFT restores compliance and enables RL.

Model	Sys. Pr.	Fmt. Qual.
Qwen2.5-VL-7B	Short	≤1.0
Qwen2.5-VL-7B	Long	81.7
Qwen2.5-VL-3B	Short	≤1.0
w/ SFT	Short	91.7
Qwen2.5-VL-3B	Long	≤1.0
w/ SFT	Long	90.1

Table 3: Efficiency comparison of tool-integrated reasoning designs. VideoTIR's internal neural-network toolkit avoids external API or specialist-model calls, yielding the fewest average tool invocations (1.42) and the lowest latency (4.6s). Latency is measured end-to-end excluding network overhead; † values are manually set by the original papers.

Metrics	VideoMA [13]	COLT [19]	VideoMTR [37]	VideoTIR
Tools Type	Ext. (API)	Ext. (tools)	Internal	Internal
Avg. Tool Invocations	3†	3†	2†	**1.42**
Latency (↓)	27.4s	68.3s	5.9s	**4.6s**

VideoMA issues multiple GPT-4o/Gemini API calls (API-bound); COLT uses 8 external specialist models.

internal tools and is able to capture critical clues even under a sparse sampling setting.

Cases Analysis. We show several whole trajectories τ_{ori}, including both the textual and visual modality from the evaluation benchmarks. We find that the formality ability gained from the SFT stage is robust enough that even lower format reward in the sequential RL stage can be kept (also see the format quality in Fig. 8a). Moreover, as the training step increases, the tool calling is more proper. For example, at the early stage, the model is likely to generate retrieving sentences that are too short for the tools. After RL, the generated retrieving sentences are shorter and proper for effectively calling the toolkit. The increasing "valid tool call" curve (Fig. 5) and the curve of response length (Fig. 8a) of the training stage also indicate this fact.

Table 4: Normal-SFT (N.SFT) vs. Random-SFT (R.SFT) on the 3B model after 300 steps (batch size = 32). Both achieve near-perfect formatting, while Random-SFT yields higher validation accuracy.

	MvBench*	VideoMME*	LvBench*	Format Quality
N.SFT	36.3	35.0	34.0	~ 100
R.SFT †	40.7	38.5	39.0	~ 100

$*$ Randomly sampled 10%.

4.3 Efficiency Analysis

Beyond accuracy, toolkit design directly affects inference efficiency. Existing methods depend on costly external APIs or specialist models [13, 19], and even lightweight internal retrieval in VideoMTR [37] requires repeated calls. As shown in Tab. 3, these methods need 2–3 invocations and 5.9–68.3s latency. VideoTIR instead uses a lightweight internal toolkit built on the backbone encoders, avoiding inter-process overhead and enabling GPU parallelization.

4.4 Ablation Studies

Synthetic Trajectories in SFT. Besides the quantitative results in Tab. 1 that indicates the performance gains after the SFT procedure, we do detailed experiments as shown in Tab. 2.

We attempt to implement Zero-RL on 3B but fails because the complex format instruction is not suitable for the 3B model, as the valid tool call strings are extremely rare and thus the advantage estimator fails since the in-group std of the advantages is near 0. Moreover, we analyze the failure string patterns of the 3B model and design a series of corrective strategies to address common erroneous JSON formats (see Supp.).

We further compare Normal SFT and Random Noised SFT on MVBench. Surprisingly (Tab. 4), on our dataset of approximately 4.5k samples, Random-Noised SFT, where the vision modality is replaced with placeholders and masked at the attention level (see supplementary materials), outperforms Normal-SFT while maintaining high Format Quality (Detailed discussions can be found in the Supps.).

Property of Global Tool and Retrieval Tools. We select 2 specific tasks "Information Synopsis" and "Percept & Recognize" in the VideoMME benchmark and analysis the tool callings on these samples. Information Synopsis needs a global understanding of the video and is suitable for "Browsing Tool" while the Percept & Recognize task needs to accurately recall the video segments and is suitable for the temporal-spatial grounding tools. The results in Tab. 5 shows that our textual router tool is able to make the proper decisions.

TAGPO. We evaluate the quality of tool calling from the perspectives of both misuse and overuse. To measure misuse, we adopt the valid tool reward, defined as the ratio of tool invocations that lead to correct answers. In addition, we

Table 5: Reasonability analysis of the textual router on Video-MME. For Information Synopsis (I.S.) tasks, which require global video understanding, the router predominantly selects browsing-based tool chains. In contrast, for Object Recognition (O.R.) tasks that focus on specific local regions, the router favors grounding-oriented tool chains.

Tool Chain Type	I.S.	O.R.
Browser	309	3
Retriever & followings	1	335
Direct Answer	13	16

Table 6: Ablation study of TAGPO on the 7B model. Validation accuracy is reported at 30 training steps with a batch size of 32 for both GRPO and TAGPO. TAGPO achieves consistently higher performance, demonstrating improved optimization efficiency in early-stage training.

Model	Val Acc.
7B w/ GRPO	21.1
7B w/ TAGPO	24.6

compute the average number of tool calls per trajectory at each training step. As shown in Fig. 8b, TAGPO consistently surpasses the baseline GRPO in terms of valid tool reward, while maintaining a comparable level of tool invocation (i.e., tool reward) during the early training stage. From the overall growth trend of the final accuracy reward, TAGPO accelerates early policy optimization by reducing unnecessary exploration time.

For the accuracy metric, as shown in Tab. 6 before, we also evaluate the 30-step accuracy of GRPO and TAGPO and find that the latter method reaches a higher accuracy on validation settings.

5 Conclusion

We present VideoTIR, a multi-turn, multi-tool reasoning framework designed to accurately and efficiently locate the information required to solve long video understanding. For 7B models, we employ end-to-end RL to learn effective tool usage strategies. We further propose a Tool Action Group Policy Optimization (TAGPO) method, which computes advantages across tool calls within the same category. This approach provides finer-grained rewards for rational tool usage. To address this problem, we develop a sandbox-based trajectory synthesis framework to construct high-quality datasets for both SFT and RL training.

Acknowledgements

This work was supported in part by the National Natural Science Foundation of China, under Grant Nos. 62192783, 62276128, and 62406140; the Young Elite Scientists Sponsorship Program by China Association for Science and Technology, under Grant No. 2023QNRC001; the Key Research and Development Program of Jiangsu Province, under Grant No. BE2023019; and the Jiangsu Natural Science Foundation, under Grant Nos. BK20221441 and BK20241200. The authors would like to thank the support of Huawei Ascend Cloud Ecological Development Project.

References

1. Ahmed, I., Islam, S., Datta, P.P., Kabir, I., Chowdhury, N.U.R., Haque, A.: Qwen 2.5: A comprehensive review of the leading resource-efficient llm with potential to surpass all competitors. Authorea Preprints (2025)
2. Alayrac, J.B., Donahue, J., Luc, P., Miech, A., Barr, I., Hasson, Y., Lenc, K., Mensch, A., Millican, K., Reynolds, M., et al.: Flamingo: a visual language model for few-shot learning. NeurIPS (2022)
3. Chen, B., Yue, Z., Chen, S., Wang, Z., Liu, Y., Li, P., Wang, Y.: Lvagent: Long video understanding by multi-round dynamical collaboration of mllm agents. arXiv (2025)
4. Chen, Z., Wu, J., Wang, W., Su, W., Chen, G., Xing, S., Zhong, M., Zhang, Q., Zhu, X., Lu, L., et al.: Internvl: Scaling up vision foundation models and aligning for generic visual-linguistic tasks. In: CVPR (2024)
5. Clark, C., Zhang, J., Ma, Z., Park, J.S., Salehi, M., Tripathi, R., Lee, S., Ren, Z., Kim, C.D., Yang, Y., Shao, V., Yang, Y., Huang, W., Gao, Z., Anderson, T., Zhang, J., Jain, J., Stoica, G., Han, W., Farhadi, A., Krishna, R.: Molmo2: Open weights and data for vision-language models with video understanding and grounding (2026)
6. Fan, Y., Ma, X., Wu, R., Du, Y., Li, J., Gao, Z., Li, Q.: Videoagent: A memory-augmented multimodal agent for video understanding. In: ECCV (2024)
7. Feng, K., Gong, K., Li, B., Guo, Z., Wang, Y., Peng, T., Wu, J., Zhang, X., Wang, B., Yue, X.: Video-r1: Reinforcing video reasoning in mllms. arXiv (2025)
8. Fu, C., Dai, Y., Luo, Y., Li, L., Ren, S., Zhang, R., Wang, Z., Zhou, C., Shen, Y., Zhang, M., et al.: Video-mme: The first-ever comprehensive evaluation benchmark of multi-modal llms in video analysis. In: CVPR (2025)
9. GLM, T., Zeng, A., Xu, B., Wang, B., Zhang, C., Yin, D., Zhang, D., Rojas, D., Feng, G., Zhao, H., et al.: Chatglm: A family of large language models from glm-130b to glm-4 all tools. arXiv (2024)
10. Grauman, K., Westbury, A., Byrne, E., Chavis, Z., Furnari, A., Girdhar, R., Hamburger, J., Jiang, H., Liu, M., Liu, X., et al.: Ego4d: Around the world in 3,000 hours of egocentric video. In: CVPR (2022)
11. Hu, K., Gao, F., Nie, X., Zhou, P., Tran, S., Neiman, T., Wang, L., Shah, M., Hamid, R., Yin, B., et al.: M-llm based video frame selection for efficient video understanding. In: CVPR (2025)
12. Jeoung, S., Huybrechts, G., Ganesh, B., Galstyan, A., Bodapati, S.: Adaptive video understanding agent: Enhancing efficiency with dynamic frame sampling and feedback-driven reasoning. arXiv (2024)
13. Kugo, N., Li, X., Li, Z., Gupta, A., Khatua, A., Jain, N., Patel, C., Kyuragi, Y., Ishii, Y., Tanabiki, M., et al.: Videomultiagents: A multi-agent framework for video question answering. arXiv (2025)
14. Lei, J., Berg, T.L., Bansal, M.: Detecting moments and highlights in videos via natural language queries. NeurIPS (2021)
15. Li, K., Wang, Y., He, Y., Li, Y., Wang, Y., Liu, Y., Wang, Z., Xu, J., Chen, G., Luo, P., et al.: Mvbench: A comprehensive multi-modal video understanding benchmark. In: CVPR (2024)
16. Lin, B., Ye, Y., Zhu, B., Cui, J., Ning, M., Jin, P., Yuan, L.: Video-llava: Learning united visual representation by alignment before projection. In: EMNLP (2024)
17. Lin, J., Hua, H., Chen, M., Li, Y., Hsiao, J., Ho, C., Luo, J.: Videoxum: Cross-modal visual and textural summarization of videos. IEEE Trans. Multimed. (2023)

18. Liu, Y., Lin, K.Q., Chen, C.W., Shou, M.Z.: Videomind: A chain-of-lora agent for long video reasoning. arXiv (2025)
19. Liu, Y., Cao, M., Shi, X., Liang, X.: Colt: Enhancing video large language models with continual tool usage. arXiv (2026)
20. Ma, Z., Gou, C., Shi, H., Sun, B., Li, S., Rezatofighi, H., Cai, J.: Drvideo: Document retrieval based long video understanding. In: CVPR (2025)
21. Maaz, M., Rasheed, H., Khan, S., Khan, F.: Video-chatgpt: Towards detailed video understanding via large vision and language models. In: ACL (2024)
22. Mangalam, K., Akshulakov, R., Malik, J.: Egoschema: A diagnostic benchmark for very long-form video language understanding. NeurIPS (2023)
23. Shinn, N., Cassano, F., Gopinath, A., Narasimhan, K., Yao, S.: Reflexion: Language agents with verbal reinforcement learning. NeurIPS (2023)
24. Shu, Y., Liu, Z., Zhang, P., Qin, M., Zhou, J., Liang, Z., Huang, T., Zhao, B.: Video-xl: Extra-long vision language model for hour-scale video understanding. In: CVPR (2025)
25. Tong, Z., Song, Y., Wang, J., Wang, L.: Videomae: Masked autoencoders are data-efficient learners for self-supervised video pre-training. NeurIPS (2022)
26. Wang, P., Bai, S., Tan, S., Wang, S., Fan, Z., Bai, J., Chen, K., Liu, X., Wang, J., Ge, W., et al.: Qwen2-vl: Enhancing vision-language model's perception of the world at any resolution. arXiv (2024)
27. Wang, W., He, Z., Hong, W., Cheng, Y., Zhang, X., Qi, J., Ding, M., Gu, X., Huang, S., Xu, B., et al.: Lvbench: An extreme long video understanding benchmark. In: ICCV (2025)
28. Wang, X., Zhang, Y., Zohar, O., Yeung-Levy, S.: Videoagent: Long-form video understanding with large language model as agent. In: ECCV (2024)
29. Wang, Z., Blume, A., Li, S., Liu, G., Cho, J., Tang, Z., Bansal, M., Ji, H.: Paxion: Patching action knowledge in video-language foundation models. NeurIPS (2023)
30. Wang, Z., Chen, B., Yue, Z., Wang, Y., Qiao, Y., Wang, L., Wang, Y.: Videochat-a1: Thinking with long videos by chain-of-shot reasoning. arXiv (2025)
31. Weng, Y., Han, M., He, H., Chang, X., Zhuang, B.: Longvlm: Efficient long video understanding via large language models. In: ECCV (2024)
32. Wu, B., Yu, S., Chen, Z., Tenenbaum, J.B., Gan, C.: Star: A benchmark for situated reasoning in real-world videos. arXiv (2024)
33. Wu, C.Y., Feichtenhofer, C., Fan, H., He, K., Krahenbuhl, P., Girshick, R.: Long-term feature banks for detailed video understanding. In: CVPR (2019)
34. Wu, C.Y., Krahenbuhl, P.: Towards long-form video understanding. In: CVPR (2021)
35. Xiao, J., Shang, X., Yao, A., Chua, T.S.: Next-qa: Next phase of question-answering to explaining temporal actions. In: CVPR (2021)
36. Xiao, J., Yao, A., Li, Y., Chua, T.S.: Can i trust your answer? visually grounded video question answering. In: CVPR (2024)
37. Xie, Y., Chen, T., Ge, Z., Ni, L.: Video-mtr: Reinforced multi-turn reasoning for long video understanding. arXiv (2025)
38. Xu, Z., Zhang, J., Wang, Q., Liu, Y.: E-vrag: Enhancing long video understanding with resource-efficient retrieval augmented generation. arXiv (2025)
39. Xue, Z., Zheng, L., Liu, Q., Li, Y., Zheng, X., Ma, Z., An, B.: Simpletir: End-to-end reinforcement learning for multi-turn tool-integrated reasoning. arXiv (2025)
40. Xue, Z., Zhang, J., Xie, X., Cai, Y., Liu, Y., Li, X., Tao, D.: Omni-adavideorag: Omni-contextual adaptive retrieval-augmented for efficient long video understanding. arXiv (2025)

41. Yang, Z., Wang, S., Zhang, K., Wu, K., Leng, S., Zhang, Y., Li, B., Qin, C., Lu, S., Li, X., Bing, L.: Longvt: Incentivizing "thinking with long videos" via native tool calling (2025)
42. Yao, S., Zhao, J., Yu, D., Du, N., Shafran, I., Narasimhan, K.R., Cao, Y.: React: Synergizing reasoning and acting in language models. In: ICLR (2022)
43. Yi, K., Gan, C., Li, Y., Kohli, P., Wu, J., Torralba, A., Tenenbaum, J.B.: Clevrer: Collision events for video representation and reasoning. arXiv (2019)
44. Zhang, H., Li, X., Bing, L.: Video-llama: An instruction-tuned audio-visual language model for video understanding. arXiv (2023)
45. Zhang, Y.F., Lu, X., Yin, S., Fu, C., Chen, W., Hu, X., Wen, B., Jiang, K., Liu, C., Zhang, T., et al.: Thyme: Think beyond images. arXiv (2025)
46. Zheng, Y., Zhang, R., Zhang, J., Ye, Y., Luo, Z., Feng, Z., Ma, Y.: Llamafactory: Unified efficient fine-tuning of 100+ language models. arXiv (2024)
47. Zheng, Z., Yang, M., Hong, J., Zhao, C., Xu, G., Yang, L., Shen, C., Yu, X.: Deepeyes: Incentivizing" thinking with images" via reinforcement learning. arXiv (2025)
48. Zhong, Y., Hu, Z.Y., Li, Y., Wang, L.: Rethinking chain-of-thought reasoning for videos. Arxiv (2025)