VTimeCoT: Thinking by Drawing for Video Temporal Grounding and Reasoning

Jinglei Zhang[1,*] Yuanfan Guo[2,*] Rolandos Alexandros Potamias[3]
Jiankang Deng[3] Hang Xu[2] Chao Ma[1,†]
[1] Shanghai Jiao Tong University [2] Noah's Ark Lab [3] Imperial College London
{zhangjinglei168, chaoma}@sjtu.edu.cn
{guoyuanfan1, xu.hang}@huawei.com, {j.deng16, r.potamias}@imperial.ac.uk

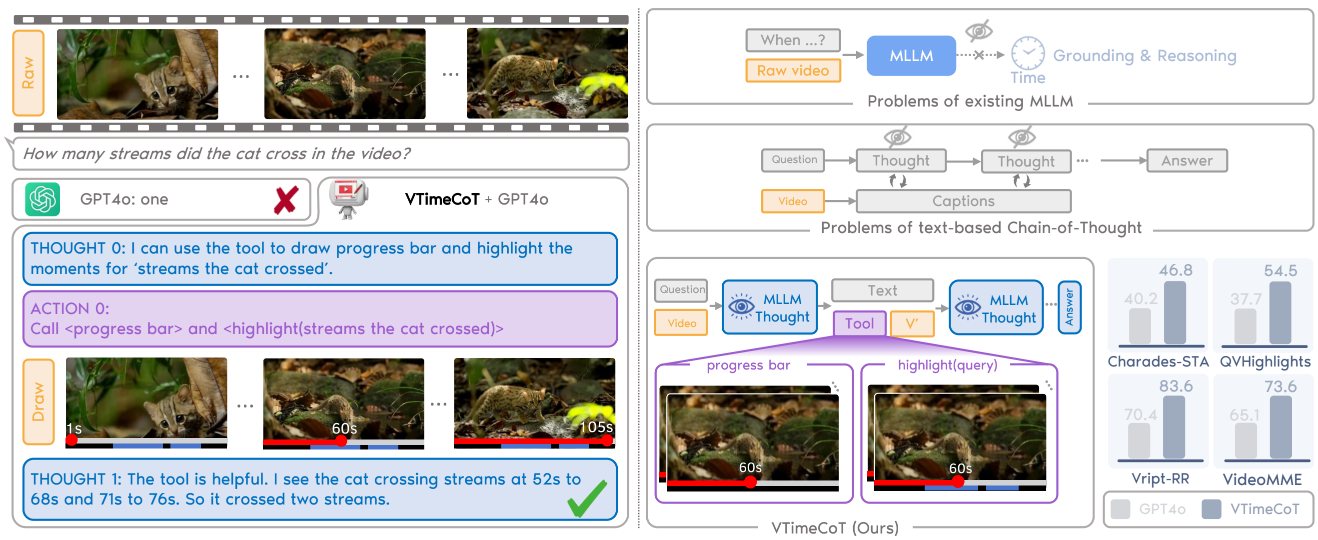

Figure 1. We propose **VTimeCoT**, a **V**isual **Time** **C**hain-**o**f-**T**hought framework for video temporal grounding and reasoning. VTimeCoT constructs cross-modality reasoning across both video and text, which enables the MLLM to utilize progress bar tools to annotate the time progression and highlight the key relevant segments to answer complex temporal questions.

Abstract

In recent years, video question answering based on multi-modal large language models (MLLM) has garnered considerable attention, due to the benefits from the substantial advancements in LLMs. However, these models have a notable deficiency in the domains of video temporal grounding and reasoning, posing challenges to the development of effective real-world video understanding systems. Inspired by how humans use video players to interact with the progress bar for video comprehension, we introduce VTimeCoT, a simple yet effective training-free framework, designed for high-performance video grounding and reasoning. The proposed framework incorporates two novel visual tools of the progress bar: a plug-and-play progress bar integration tool and a high-efficiency highlighting tool. In addition, to address the limitations of conventional text-based chain-of-thought (CoT) approaches, we introduce a visuotemporal CoT process that integrates cross-modality reasoning across both video and text. Our approach demonstrates significant performance improvements on both Qwen2VL-7B and GPT4o baselines in tasks of video temporal grounding and reasoning-based question answering. Finally, we showcase that the proposed framework achieves a compositional and interpretable reasoning process. Project page: https://vtimecot.github.io.

*Equal contribution. †Corresponding author.

1. Introduction

Video understanding is a longstanding problem in computer vision and has attracted more attention with the emergence of large language models (LLM) [5, 17, 40, 43]. Video QA is a representative task reflecting the video reasoning ability. Given that videos typically contain numerous events occurring at different time points, accurately answering questions is a highly challenging task. Recently, some end-to-end video understanding models based on multimodal large language models (MLLM) [20, 23, 26, 27, 56, 58] have been proposed to address video question-answering tasks, exhibiting remarkable capabilities. However, despite their ability to generate seemingly plausible results, these methods have been shown to exhibit a notable deficiency in temporal grounding [15] and fail to provide a temporal-grounded reasoning process.

With the advent of tool usage capabilities in LLMs, several agent-based video question-answering methods have been proposed [9, 33, 37, 44, 53], enabling step-by-step video understanding. These methods prompt the LLMs to sequentially reason and invoke external tool models (*e.g.*, object detection and captioning models) to collect textual clues, ultimately inferring the final answer. This, so called, chain-of-thought approach offers significant potential, as it is compositional and training-free, demonstrating impressive zero-shot performance. However, such methods predominantly rely on language and captions as intermediaries, limiting their ability to directly capture the visuotemporal dynamics inherent in videos, especially in the case of long videos. Inspired by the time-progress bar commonly used in video players during human interactions, we identify it as an intuitive tool for grounding time and enhancing video comprehension. In contrast to written language, the progress bar, along with the concurrent visual content, directly conveys the concept of temporal progression. The video progress bar can represent temporal relationships through sequential positioning or more abstract temporal dependencies.

In this work, we propose a simple yet effective framework that empowers multimodal LLMs to utilize a video progress bar to construct a **V**isual **Time** **C**hain-**o**f-**T**hought (VTimeCoT), facilitating long video temporal grounding and reasoning. Inspired by LLMs coupled with visual programming [12, 39], we propose enabling multimodal LLMs to generate code that overlays a progress bar on the video. First, we construct a frame-sync progress bar integration tool that can be invoked by the MLLM, which generates a progress bar displayed at the bottom of the video, marking the progress and annotating the timestamps on each frame. This is naturally adapted to long video and enables the MLLM to perceive the speed of temporal progress directly from the annotated video frames, adapting seamlessly to any frame-per-second (FPS) sampling rate. Second, to guide the model's attention on key relevant temporal segments, we propose a training-free and long-video-adaptive highlighting tool for the progress bar based on video-text similarity. Specifically, we leverage a robust video-text foundation model to identify the time intervals with the top-k highest similarity and highlight them on the visual progress bar. Finally, we prompt the multimodal LLM to invoke the progress bar tools within a visuotemporal chain-of-thought process that integrates text, program, and video annotated with the progress bar to facilitate long video reasoning. At each thought step, the model performs cross-modality reasoning and determines whether to dynamically update the video memory while progressively inferring the answer. As shown in Fig. 1, to determine how many times a cat crosses a stream in the input video, the model first invokes tools to generate a progress bar, identifying the key timestamps corresponding to 'streams the cat crossed'. By analyzing the progress bar annotated frames, the model infers the exact timing of each crossing event and accurately determines the number of crossings. This approach enhances the model's temporal reasoning capabilities by seamlessly integrating temporal-grounded visual evidence.

We demonstrate the effectiveness of the proposed framework across a wide range of video temporal grounding and video-QA tasks. The proposed method consistently outperforms the baseline models, Qwen2VL and GPT-4o in temporal grounding tasks, achieving significant performance gains with an average IoU improvement of 6.58% and 16.83% over GPT-4o across Charades-STA [11] and QVHighlights [19] benchmarks. Similarly, VTimeCoT significantly improves the accuracy of questions related to temporal retrieval, event counting, and event ordering in reasoning-based question-answering tasks. Specifically, the proposed method consistently surpasses state-of-the-art methods across Vript-RR [50] and VideoMME [10] benchmarks.

To sum up, we present VTimeCoT, a training-free model for temporal reasoning based on the visual progress bar. Specifically:

- We propose the first, to the best of our knowledge, visual time chain-of-thought framework for video temporal grounding and reasoning. In contrast to previous methods that rely solely on textual reasoning, we leverage the visual progress bar as a medium, making a significant step towards real-world video understanding systems.
- The proposed reasoning framework leverages two novel tools from video progress bar integration and highlighting, enabling the MLLM to accurately perceive the timestamp of each frame and identify key temporal segments.
- The proposed method enhances the performance of MLLMs by a large margin in video temporal grounding and reasoning tasks through a training-free approach, while also demonstrating strong reasoning capabilities.

2. Related Work

Multimodal LLM for Video. In contrast with traditional video understanding such as obeject detection, segmentation and pose estimation [34, 36, 41, 57], recent multimodal LLMs enable understanding through natural language. End-to-end video multimodal LLMs are trained on large-scale datasets, to align visual inputs with the language modality and integrate with large language models. Video-ChatGPT [31] leveraged CLIP [35] to extract spatial and temporal features from videos and integrate them with Vicuna [7]. Video-LLaMA [56] proposed a video Q-Former designed to enhance the modeling of temporal variations. Subsequently, a series of video multimodal LLM variants [6, 20, 22, 24–26, 48] have introduced advancements in enhancing video representation, expanding data scale, and refining training methods, leading to improved video question-answering performance. However, these end-to-end models operate as black boxes, directly generating answers without an explicit video reasoning process. In contrast, we propose a video understanding framework that incorporates a structured reasoning process.

Programming and Tool Usage of LLM. Due to the strong code-generation capabilities of LLMs, several works [8, 12, 14, 29, 39, 52] attempted to prompt LLMs to call compositional visual tools to address visual question-answering tasks. VisProg [12] and ViperGPT [39] defined a variety of image foundation functions (e.g., object detection) and prompted the LLMs to generate code that invokes these functions for answering questions. MM-REACT [52] tackled visual tasks through multi-step reasoning guided by text prompts to the LLM, where each step involves thought and calling visual tools. More recently, several studies [9, 33, 37, 44, 53] proposed tool-usage frameworks tailored for video understanding. By designing specialized video tools and calling steps, these approaches extract textual cues from videos to facilitate question answering. In contrast to the aforementioned works that solely rely on text as the reasoning medium, we introduce a visual progress bar, enabling more effective integration of visual and temporal cues.

Visual Prompt for Multimodal LLM. LLMs have been shown to enhance their understanding capabilities by incorporating overlaid visual prompts on images (e.g., circles, keypoints) [3, 4, 13, 21, 32, 38, 46, 47, 51]. SoM [51] proposed to utilize SAM [18] segmentation model to overlay mask colors and labels on images as input of GPT-4V [2], demonstrating strong zero-shot performance on vision-grounding tasks. Hu *et al*. [13] introduced a chain-of-thought framework combined with visual prompts to draw markers on images during reasoning, effectively enhancing the image understanding performance of MLLMs. However, all previous methods are limited to spatial prompts, limiting their understanding capabilities. In this work, we propose to design the progress bar as visual-temporal prompts to facilitate temporal reasoning.

3. Method

We propose VTimeCoT, a general framework that empowers multimodal large language models using a visual progress bar as an intermediate reasoning step, facilitating temporal grounding and reasoning. Fig. 2 illustrates how our approach works. Given a raw video and a question, our method generates a chain-of-thought and invokes a set of tools for progress bar integration. By utilizing the visual cues from the progress bar as a medium, the model progressively infers the final answer.

3.1. Visuotemporal Chain-of-Thought

In video understanding, particularly for long videos, existing MLLM methods can only provide static answers, which limits the model's ability to focus on critical scenes and results in significant shortcomings in real-world settings. Since the MLLMs fail to identify key scenes within a single forward pass, it is necessary to interact with the video to progressively infer the answer. Therefore, we propose a framework based on interactive mechanisms and dynamic video memory.

Our framework tackles video grounding and reasoning tasks through an iterative interaction with the MLLM, built upon a tool set leveraging the progress bar. Given a textual question and a video as input, VTimeCoT generates a sequence of thoughts and actions to dynamically update the video memory. By manipulating the video, it acquires the necessary information to answer the question. In this process, the MLLM reasons by plotting and analyzing the progress bar, seamlessly integrating both textual and visual reasoning into a *Visuotemporal Chain-of-Thought*. As illustrated in Fig. 2, VTimeCoT first provides the MLLM with a formatted prompt for initialization and executes actions at each time step t. The tool set includes the <progress bar>, <highlight>, <cut>, etc. The <cut>tool is utilized to trim specific segments when the MLLM determines that the video is too long to find the required information. The pseudo-code is presented in Algorithm 1, where $\mathbb{V}$ and $\mathbb{L}$ represent the modalities of video and language, respectively. In Algorithm 1, p serves as a parser to retrieve specific sub-strings by keyword, and c is a code builder that integrates the toolset library and generated code.

Initialization Prompt. To enable the MLLM to perform step-by-step reasoning and a structured output, we follow [13] and construct a set of initialization prompts. These prompts define the specific output structure (formatted by keywords of 'THOUGHT', 'ACTION', and 'TERMINATE') for each step and declare the Python tools that the model should invoke. The initialization prompts, along with the question and video frames, are then fed into the MLLM, triggering the iterative loop.

Thought. During this step, the model analyzes the historical context and video memory to generate its reasoning

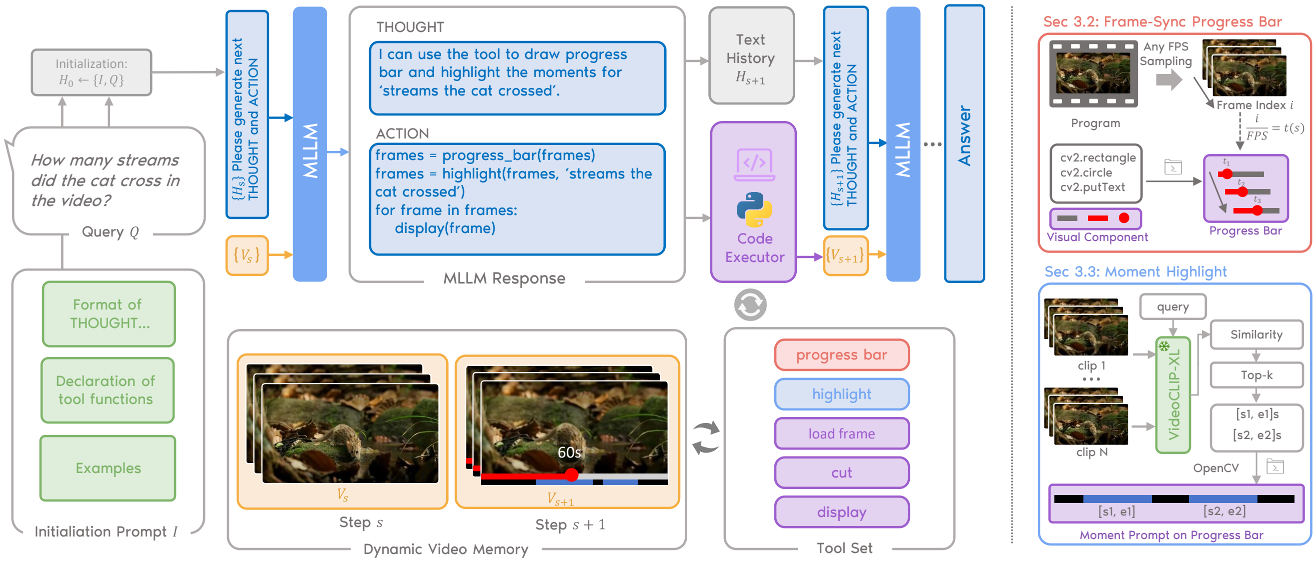

Figure 2. **Overview of our method.** On the left, we demonstrate how the framework iteratively generates thoughts and actions, which dynamically updates the video memory with an overlaid progress bar for reasoning. On the right, we illustrate two novel tools that integrate the frame-sync visual progress bar and highlight key moments.

for the current step. It determines whether a direct answer can be provided directly in this step or if tool assistance is needed, and if so, which tools should be invoked.

Action. Following the thought step, the model generates an executable Python script utilizing the given tool functions. The framework then builds the code with the toolset library and executes the script to manipulate the video frames.

Dynamic Video Memory. Using the manipulated video frames we subsequently update the frames in the video memory. The updated context with the manipulated video frames is then fed into the MLLM to proceed with the next step of thought.

To terminate the loop, the model determines at each step whether to cease reasoning and generate the 'TERMINATE' keyword to end. To prevent excessively prolonged reasoning, we impose a maximum step limit, enforcing termination at step T.

Algorithm 1 Visuotemporal Chain-of-Thought

Input: $\{V, Q \| V \in \mathbb{V}, Q \in \mathbb{L}\}; I \in \mathbb{L}$;
$\texttt{MLLM} : \mathbb{V}, \mathbb{L} \to \mathbb{L}$; Toolset : $\{T_1, \ldots, T_n\} \in \mathbb{L}$

```
s ← 0; V_0 ← V
H_0 ← {I, Q}                              ▷ Initialization prompt and query
while true do
    y_s ← MLLM (V_s, H_s)
    THOUGHT_s, ACTION_s, TERMINATE_s ← p(y_s)
    if not TERMINATE_s then
        C_s ← c(ACTION_s, Toolset)        ▷ Build code
        V_{s+1} ← C_s(V_s)                ▷ Update video
        H_{s+1} ← {H_s, y_s}              ▷ Update history
    else
        return y_s                        ▷ Respond final answer
    end if
end while
```

3.2. Frame-Sync Visual Progress Bar Integration

Enabling MLLM to perceive the precise timestamp of each frame is a significant challenge and an unresolved problem, due to the diverse temporal sampling rates during MLLM training and inference without time. Although recent approaches [15, 16] have attempted to address this issue by incorporating temporal positional embeddings before the video encoder and introducing temporal-grounding training tasks, their flexibility is limited due to the scarcity of specific data and the need for fine-tuning. This issue is particularly pronounced in long videos, where the token limitations of MLLMs prevent them from accurately perceiving the timestamp of each frame.

Inspired by visual programming paradigms [12, 39], we propose a simple yet effective method to empower MLLMs to perceive the precise timestamp of each frame, by generating the video progress bar. Leveraging the code generation capabilities of MLLMs, our approach prompts the MLLM to invoke Python plotting tools and use their universal OCR and shape understanding capabilities to perceive time, without any additional training requirements. Specifically, given a raw video $V \in \mathbb{R}^{T \times H \times W \times 3}$ as input, this tool can generate the progress bar at the bottom of the original frames and return the annotated frames. To enable adaptation to arbitrary FPS sampling, we perform a frame-synchronization step to convert frame indexes to the real second time, as shown in Fig. 2. To enable MLLM to perceive seconds directly without any additional hour-minute-second conversions, we design the timestamps using a straightforward seconds format. To construct a robust visual prompt that can be easily interpreted from the MLLM, we employ compositional components using elongated rectangles, circles, and timestamps to plot the progress bar. Finally, the progress bar is placed at the bottom of raw frames, which can be expressed as:

$$V_t' = V_t \oplus T_t, \tag{1}$$

where $\oplus$ represents the vertical concatenation operation, T_t denotes the image of the progress bar generated by OpenCV at time t and V_t' is the resulting frame. We wrap this integration process into a Python tool function, allowing MLLM to invoke it for generating the annotated video.

3.3. Moment Highlight by Video-Text Similarity

Although multimodal LLMs have made significant progress in video content captioning, recent studies [15, 16] have revealed substantial deficiencies in their temporal grounding capabilities. Existing works [15, 16] rely on constructing temporal grounding instructional data and require finetuning to emerge such capability. Besides, due to token limitations in MLLMs, they struggle to perform temporal grounding for long videos. In contrast, we propose a training-free and long-video-adaptive method to enhance the temporal grounding capabilities of MLLMs, leveraging the similarities from the robust video-text foundation models.

Given a query and a video $V \in \mathbb{R}^{T\times H\times W\times 3}$ containing multiple events, our proposed tool estimates the query-relevant temporal segments, including their start and end times, and generates a highlighted visual prompt.

Moment Retrieval by Video-Text Similarity. To construct a robust zero-shot moment retrieval module, capable of generalizing across a wide range of videos, we employ the VideoCLIP-XL [42] foundation model as our embedding extractor. The video is initially processed at a high temporal resolution of r FPS. We group every 8 frames into a video clip, resulting in N clips. It should be noted that since the temporal retrieval module operates as an external component, it can efficiently handle high frame rates without being constrained by the token count of the MLLM. This makes our method adaptive to long-form video. These clips are then processed by the VideoCLIP-XL visual encoder, producing N video embeddings. We use VideoCLIP-XL text encoder to extract the text embedding for the text query and we compute the cosine similarity between the query embedding and the video embeddings of each clip:

$$\text{sim}(i) = \frac{e_i^v \cdot e^q}{|e_i^v||e^q|}, \tag{2}$$

where e_i^v denotes the embedding of the i-th clip and e^q is the query embedding. Using top-k selection, we obtain the k clips with the highest similarity and extract the start and end timestamps of each contiguous segment.

Moment Highlight on the Progress Bar. Motivated by a series of works that demonstrated the effectiveness of visual cues as a medium for reasoning beyond text [3, 13, 21, 32, 46], we propose using moment-highlighting cues for video temporal reasoning. As shown in Fig. 2, we construct the highlight tool using OpenCV library to plot colorful highlight masks on specific intervals of the progress bar. This tool takes the video frames and time intervals as input and annotates the highlighted progress bar under the frames. To ensure that the MLLM can easily interpret the highlights, we utilize different colors between the mask and the progress bar. We wrap the aforementioned retrieval and highlighting process as a Python tool function, enabling the MLLM to invoke it to manipulate the input video frames.

4. Experiments

In this section, we first present the implementation details, datasets and evaluation metrics. Subsequently, we assess the performance of the proposed framework on the video temporal grounding benchmarks (Sec. 4.1). To further evaluate the reasoning capabilities, we conducted extensive quantitative and qualitative analyses on the "reasoning based on retrieving" benchmark of Vript-RR (Sec. 4.2). In addition, we demonstrate the significant role our reasoning framework plays in enhancing long video question-answering performance (Sec. 4.3). Finally, we give the ablations of key modules (Sec. 4.4).

Implementation Details. In the framework, we conducted experiments on two core MLLMs, Qwen2VL-7B [43] and GPT4o-20240513 [17], as they represent the state-of-the-art open-source and closed-source MLLM models, respectively. We set the decoding temperature of the MLLM to 0. By default, the input for the MLLM consists of 32 frames uniformly sampled from the video. For video frames, H and W are resized to make the longer side 480 pixels. Our agent implementation is based on AutoGen [45]. The implementations of tool functions are adapted from Gradio [1]. For GPT4o, we utilize the OpenAI API service. The details of the MLLM prompts are provided in the supplementary materials. For VideoCLIP-XL we use FPS $r = 1$ to sample video frames and group every 8 frames as a clip to extract embedding. Regarding the top-k selection of clips, k is empirically set to 8. The maximum number of reasoning steps T is set to 3.

Datasets and Evaluation Metrics. To evaluate the proposed VTimeCoT method, we utilized four benchmark datasets, spanning a wide range of video temporal grounding, reasoning, and long-video question-answering tasks.

Charades-STA [11] is a benchmark dataset for temporal

Method	R1@0.3	R1@0.5	R1@0.7	mIoU
VideoChat-7B [23]	9.00	3.30	1.30	6.50
VideoLLaMA-7B [56]	10.40	3.80	0.90	7.10
VideoChatGPT-7B [31]	20.00	7.70	1.70	13.70
LLAVA-Onevision-7B [20]	33.04	11.05	4.11	20.98
VTimeLLM-7B [15]	51.00	27.50	11.40	31.20
VTimeLLM-13B [15]	55.30	34.30	14.70	34.60
Qwen2VL-7B [43]	37.31	12.85	4.11	24.34
VTimeCoT$_{\text{Qwen2VL-7B}}$	**66.96**	**38.79**	**20.83**	**43.41**
GPT4o [17]	63.76	37.12	14.65	40.20
VTimeCoT$_{\text{GPT4o}}$	**74.06**	**51.02**	**22.45**	**46.78**

Table 1. **Quantitative comparison on the Charades-STA dataset for temporal grounding.** We report the mIoU and the recall performance at different IoU thresholds.

Method	R1@0.3	R1@0.5	R1@0.7	mIoU
LLAVA-Onevision-7B [20]	34.91	17.91	9.7	25.53
VTimeLLM-7B [15]	44.58	25.03	9.29	28.99
Qwen2VL-7B [43]	31.87	14.65	7.35	22.77
VTimeCoT$_{\text{Qwen2VL-7B}}$	**67.50**	**45.79**	**25.11**	**46.21**
GPT4o [17]	55.61	35.68	19.29	37.66
VTimeCoT$_{\text{GPT4o}}$	**79.35**	**59.74**	**33.81**	**54.49**

Table 2. **Quantitative comparison on the QVHighlights dataset for temporal grounding of discontinuous segments.** We report the mIoU and the recall performance at different IoU thresholds.

grounding, composed of 1334 videos along with the corresponding start-end frame annotations of 3720 queries. The average video length is 30 seconds. We follow [15] and report mean IoU (mIoU) and recall@1, IoU $\geq m$(R@m) metrics, where m includes 0.3, 0.5 and 0.7. IoU denotes the intersection over the union between the predicted and ground truth time segments.

QVHighlights [19] is a benchmark dataset for semantic relevance and temporal grounding from discontinuous time segments. It contains 1519 videos, with an average length of 150 seconds, and 1550 queries with time annotations, where each query spans multiple discontinuous time segments within a video. For evaluation, we report the commonly used mIoU and R@m metrics.

Vript-RR [50] is a challenging benchmark dataset for scene retrieval and multi-hop reasoning. The average length of the videos is 622 seconds. It contains 152 questions, each accompanied by a hint to locate the scene that the question refers to. The benchmark includes both multiple-choice and open-ended question-answering settings. We follow [50] and evaluate the open-ended accuracy using GPT4.

VideoMME [10] is a video question-answering benchmark tailored for MLLMs, featuring diverse videos ranging from 11 seconds to 1 hour in length. It contains 900 videos and 2700 question-answer pairs. We follow [10] to report accuracy metrics under two evaluation settings with subtitles and without subtitles.

Methods	Multi-Choice	Open
VideoChatGPT [31]	29.60	17.80
Video-LLaMA [56]	28.30	14.50
VideoChat [23]	22.40	15.10
VideoChat2 [24]	42.10	22.40
ST-LLM [28]	33.60	26.30
PLLaVA 7B [48]	55.30	36.20
VILA-1.5 8B [27]	55.30	32.30
Qwen2VL-7B [43]	59.87	35.95
VTimeCoT$_{\text{Qwen2VL-7B}}$	**62.50**	**41.45**
GPT4o [17]	70.39	61.18
VTimeCoT$_{\text{GPT4o}}$	**83.55**	**68.42**

Table 3. **Quantitative comparison on the Vript-RR dataset.** We report the response accuracy on both multiple-choice and open-ended question settings.

4.1. Temporal Grounding

To ensure accurate temporal reasoning, it is essential to achieve robust and high-performance temporal grounding. In Tab. 1, we compare VTimeCoT with two state-of-the-art baselines, Qwen2VL-7B and GPT4o, to identify the start and end timestamps of the query event. Specifically, using both Qwen2VL-7B and GPT4o models as the core MLLM of our framework, we evaluate the performance of VTimeCoT to localize the query event. As can be easily observed, although these advanced MLLMs achieve state-of-the-art performance in video content recognition, they fall short in accurately grounding temporal boundaries for events in videos, facing difficulties in perceiving the precise timestamp and the boundaries of events. In contrast, the proposed method effectively addresses these challenges, providing accurate visual cues of the progress bar through the reasoning process that facilitates the grounding performance in videos. Note that VTimeCoT demonstrates superior performance without any additional training cost, even compared to the VTimeLLM model, which has been meticulously fine-tuned on extensive temporal grounding data.

To further evaluate the temporal grounding performance of the proposed and the baseline methods in the presence of multiple discontinuous time segments for each query, we utilized the QVHighlights benchmark dataset (Tab. 2). Given a query and a video, we request from each model to identify start-end timestamps of all temporal segments in the video that fit the query. As can be observed, VTimeCoT consistently enhances the performance compared to different MLLMs. On the contrary, baseline MLLM meth-

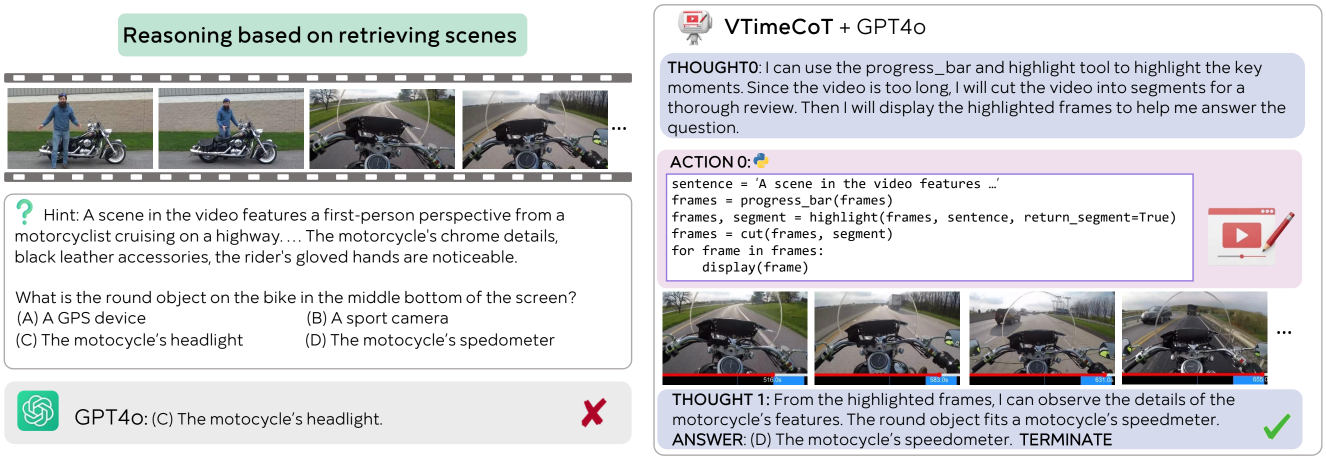

Figure 3. **Qualitative comparison on Vript-RR benchmark.** In contrast to GPT-4o which is misled by the irrelevant scenes, VTimeCoT reasons using the progress bar and highlights the key moments, leading to accurate answers.

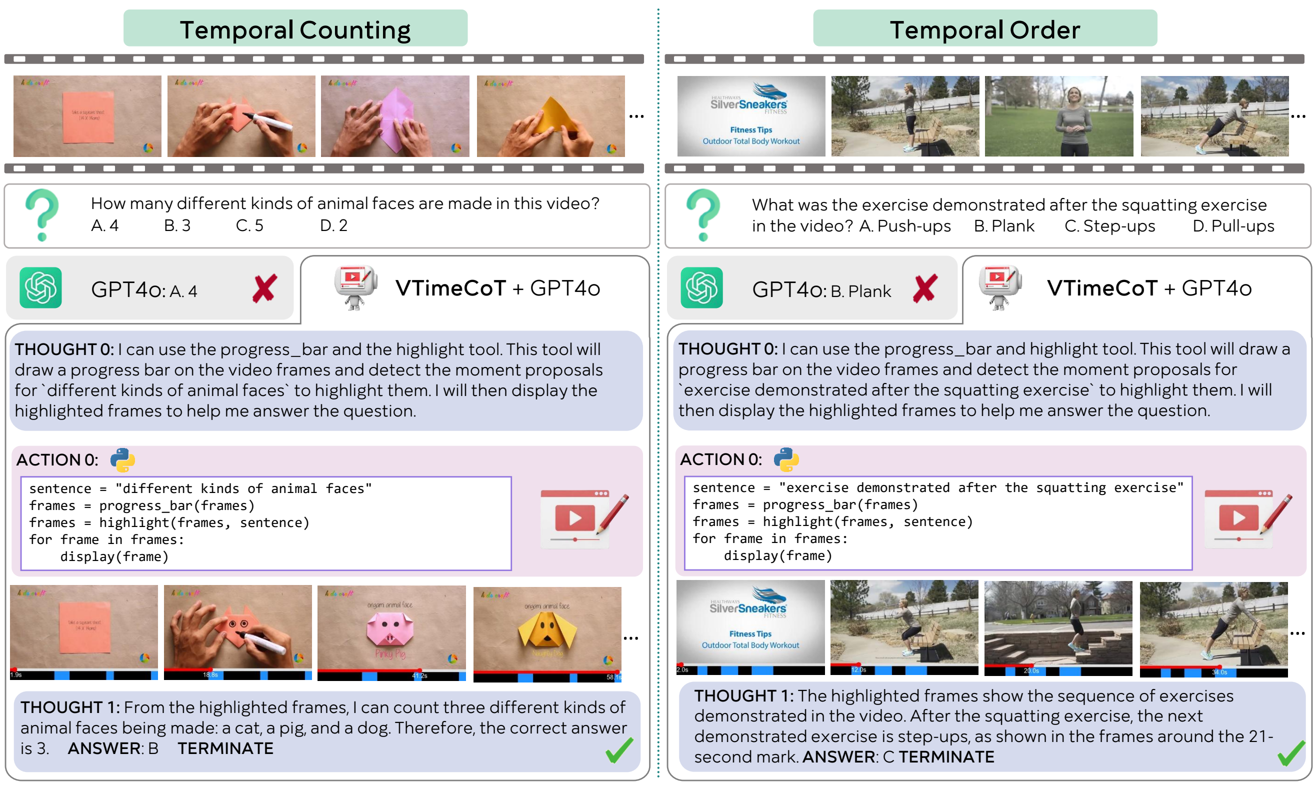

Figure 4. **Qualitative comparison on VideoMME benchmark.** Through reasoning that integrates the progress bar and highlights, VTimeCoT accurately answers questions related to temporal counting and order, which GPT-4o falls short.

ods struggle with discontinuous time segments, failing to identify multiple temporal spans and leading to degraded grounding performance.

4.2. Long-Video Reasoning Based on Retrieving

In Tab. 3, we evaluate the question-answering performance of the proposed and the baseline methods on the Vript-RR benchmark, which contains questions that require scene retrieval. Given a scene hint and a question, the model must identify the specific scene to deliver an accurate response. We compared the proposed method with the state-of-the-art baseline methods including Qwen2VL-7B and GPT4o. As can be observed VTimeCoT achieves a significant per-

Methods	Frames	Accuracy w/o subs	Accuracy w. subs
Gemini-1.5-Flash [40]	1 fps	70.3	75.0
GPT4o [17]	384	71.9	77.2
Gemini-1.5-Pro [40]	1 fps	75.0	81.3
Video-LLaVA-7B [26]	8	39.9	41.6
VideoLLaMA2-7B [6]	16	54.9	56.4
LLAVA-Onevision-7B [20]	32	58.2	61.5
LongVILA-7B [49]	256	60.1	65.1
InternVL2.5-8B [5]	64	64.2	66.9
mPLUG-Owl3-7B [54]	128	59.3	68.1
LLaVA-Video-7B [58]	64	63.3	69.7
NVILA-8B [30]	256	64.2	70.0
VideoLLaMA3-7B [55]	180	66.2	70.3
GPT4o [17]	32	61.6	65.1
VTimeCoT$_{\text{GPT4o}}$	32	64.2	73.6

Table 4. **Quantitative comparison on the VideoMME dataset.** We report the response accuracy under two settings: with subtitles and without subtitles.

formance improvement, surpassing the baseline methods in both multiple-choice and open-ended settings.

To further illustrate the superiority of the proposed method in the task of scene retrieval and reasoning, we qualitatively compare GPT4o and VTimeCoT on the Vript-RR dataset in Fig. 3. Through multi-step reasoning that leverages the progress bar and highlights the key time interval, VTimeCoT accurately identifies the correct answer, while GPT-4o falls short.

4.3. Long-Video Question Answering

A pivotal challenge in long-form video question answering lies in accurately inferring the frequency and sequential relationships of temporal events. Currently, end-to-end video MLLMs struggle to accurately localize events and capture their temporal dependencies across multiple frames, limiting their ability to generalize to real-world temporal understanding. In Tab. 4, we evaluate the performance of the proposed VTimeCoT model in question answering on long videos of the VideoMME benchmark dataset. Due to the prohibitive evaluation costs and to ensure a fair comparison, we re-evaluate GPT-4o using 32 frames. Utilizing robust temporal grounding and reasoning, VTimeCoT significantly outperforms the baseline methods in long-video question answering.

The effectiveness of the proposed method can be further validated in Fig. 4, where we compared the responses of the state-of-the-art GPT4o and VTimeCoT. In contrast to GPT4o, VTimeCoT can accurately identify the correct answers through its systematic, step-by-step reasoning process, even in challenging scenarios involving temporal counting and order discernment. It is also worth mentioning that VTimeCoT, apart from its superior performance in question-answering accuracy, demonstrates logical and interpretable reasoning steps.

MLLM	CoT	Progress Bar	Highlight	QVHighlights (mIoU)	Vript-RR (M-Acc)
GPT4o	×	×	×	37.66	70.39
GPT4o	✓	×	×	41.85	73.68
GPT4o	✓	✓	×	49.40	76.32
GPT4o	✓	✓	✓	54.49	83.55

Table 5. **Ablations on the key modules.** M-Acc is the multi-choice accuracy on Vript-RR.

4.4. Ablations

To further investigate the contribution of each component in the proposed method, we conducted an ablation study. In Tab. 5, we evaluate the importance of the Chain-of-Thought (CoT), progress bar, and highlighting modules. As can be observed the GPT4o employing standard CoT without the use of the progress bar (*i.e.*, solely on text-based step-by-step reasoning), results in significant performance degradation compared to the proposed method. To assess the effect of the proposed modules of the progress bar, we developed a configuration that utilizes only the progress bar tool without using the highlighting module. For completion, we also report the full VTimeCoT results, utilizing both the progress bar and highlighting tools. Both tools contribute to the performance improvement of the proposed method, which highlights the effectiveness of explicitly visualizing the precise timestamp and highlighting the relevant moments.

5. Conclusion

In this work, we propose the first visual time framework, designed to formulate a visuotemporal chain-of-thought for video temporal grounding and reasoning. We introduce a plug-and-play tool of the progress bar to generate visual temporal cues from any video, that can enable multi-modal large language models (MLLMs) to leverage tool-usage capabilities for accurately perceiving the speed of temporal progression. In addition, we propose a zero-shot temporal retrieval tool, built on top of a strong video-text foundational model, augmenting it with temporal grounding capabilities without any further training requirements. We integrate these two tools into a visuotemporal chain-of-thought framework to facilitate cross-modal reasoning between video and text. Through extensive experiments, we demonstrate that the proposed method surpasses previous state-of-the-art baselines in temporal grounding and reasoning-based question-answering benchmarks, demonstrating logical and interpretable reasoning steps.

Acknowledgments. This work was supported by NSFC (62322113, 62376156), Shanghai Municipal Science and Technology Major Project (2021SHZDZX0102), and the Fundamental Research Funds for the Central Universities.

References

[1] Abubakar Abid, Ali Abdalla, Ali Abid, Dawood Khan, Abdulrahman Alfozan, and James Zou. Gradio: Hassle-free sharing and testing of ml models in the wild. *arXiv preprint arXiv:1906.02569*, 2019. 5

[2] Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023. 3

[3] Syeda Nahida Akter, Aman Madaan, Sangwu Lee, Yiming Yang, and Eric Nyberg. Self-imagine: Effective unimodal reasoning with multimodal models using self-imagination. *arXiv preprint arXiv:2401.08025*, 2024. 3, 5

[4] Mu Cai, Haotian Liu, Siva Karthik Mustikovela, Gregory P Meyer, Yuning Chai, Dennis Park, and Yong Jae Lee. Vip-llava: Making large multimodal models understand arbitrary visual prompts. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12914–12923, 2024. 3

[5] Zhe Chen, Weiyun Wang, Yue Cao, Yangzhou Liu, Zhangwei Gao, Erfei Cui, Jinguo Zhu, Shenglong Ye, Hao Tian, Zhaoyang Liu, et al. Expanding performance boundaries of open-source multimodal models with model, data, and test-time scaling. *arXiv preprint arXiv:2412.05271*, 2024. 1, 8

[6] Zesen Cheng, Sicong Leng, Hang Zhang, Yifei Xin, Xin Li, Guanzheng Chen, Yongxin Zhu, Wenqi Zhang, Ziyang Luo, Deli Zhao, et al. Videollama 2: Advancing spatial-temporal modeling and audio understanding in video-llms. *arXiv preprint arXiv:2406.07476*, 2024. 3, 8

[7] Wei-Lin Chiang, Zhuohan Li, Zi Lin, Ying Sheng, Zhanghao Wu, Hao Zhang, Lianmin Zheng, Siyuan Zhuang, Yonghao Zhuang, Joseph E. Gonzalez, Ion Stoica, and Eric P. Xing. Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality, 2023. 3

[8] Rohan Choudhury, Koichiro Niinuma, Kris M Kitani, and László A Jeni. Zero-shot video question answering with procedural programs. *arXiv preprint arXiv:2312.00937*, 2023. 3

[9] Yue Fan, Xiaojian Ma, Rujie Wu, Yuntao Du, Jiaqi Li, Zhi Gao, and Qing Li. Videoagent: A memory-augmented multimodal agent for video understanding. In *European Conference on Computer Vision*, pages 75–92. Springer, 2024. 2, 3

[10] Chaoyou Fu, Yuhan Dai, Yongdong Luo, Lei Li, Shuhuai Ren, Renrui Zhang, Zihan Wang, Chenyu Zhou, Yunhang Shen, Mengdan Zhang, et al. Video-mme: The first-ever comprehensive evaluation benchmark of multi-modal llms in video analysis. *arXiv preprint arXiv:2405.21075*, 2024. 2, 6

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

[12] Tanmay Gupta and Aniruddha Kembhavi. Visual programming: Compositional visual reasoning without training. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14953–14962, 2023. 2, 3, 4

[13] Yushi Hu, Weijia Shi, Xingyu Fu, Dan Roth, Mari Ostendorf, Luke Zettlemoyer, Noah A Smith, and Ranjay Krishna. Visual sketchpad: Sketching as a visual chain of thought for multimodal language models. *arXiv preprint arXiv:2406.09403*, 2024. 3, 5

[14] Yushi Hu, Otilia Stretcu, Chun-Ta Lu, Krishnamurthy Viswanathan, Kenji Hata, Enming Luo, Ranjay Krishna, and Ariel Fuxman. Visual program distillation: Distilling tools and programmatic reasoning into vision-language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9590–9601, 2024. 3

[15] Bin Huang, Xin Wang, Hong Chen, Zihan Song, and Wenwu Zhu. Vtimellm: Empower llm to grasp video moments. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14271–14280, 2024. 2, 4, 5, 6

[16] De-An Huang, Shijia Liao, Subhashree Radhakrishnan, Hongxu Yin, Pavlo Molchanov, Zhiding Yu, and Jan Kautz. Lita: Language instructed temporal-localization assistant. In *European Conference on Computer Vision*, pages 202–218. Springer, 2024. 4, 5

[17] Aaron Hurst, Adam Lerer, Adam P Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, et al. Gpt-4o system card. *arXiv preprint arXiv:2410.21276*, 2024. 1, 5, 6, 8

[18] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C Berg, Wan-Yen Lo, et al. Segment anything. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4015–4026, 2023. 3

[19] Jie Lei, Tamara L Berg, and Mohit Bansal. Detecting moments and highlights in videos via natural language queries. *Advances in Neural Information Processing Systems*, 34: 11846–11858, 2021. 2, 6

[20] Bo Li, Yuanhan Zhang, Dong Guo, Renrui Zhang, Feng Li, Hao Zhang, Kaichen Zhang, Peiyuan Zhang, Yanwei Li, Ziwei Liu, et al. Llava-onevision: Easy visual task transfer. *arXiv preprint arXiv:2408.03326*, 2024. 2, 3, 6, 8

[21] Chengzu Li, Wenshan Wu, Huanyu Zhang, Yan Xia, Shaoguang Mao, Li Dong, Ivan Vulić, and Furu Wei. Imagine while reasoning in space: Multimodal visualization-of-thought. *arXiv preprint arXiv:2501.07542*, 2025. 3, 5

[22] Feng Li, Renrui Zhang, Hao Zhang, Yuanhan Zhang, Bo Li, Wei Li, Zejun Ma, and Chunyuan Li. Llava-next-interleave: Tackling multi-image, video, and 3d in large multimodal models. *arXiv preprint arXiv:2407.07895*, 2024. 3

[23] KunChang Li, Yinan He, Yi Wang, Yizhuo Li, Wenhai Wang, Ping Luo, Yali Wang, Limin Wang, and Yu Qiao. Videochat: Chat-centric video understanding. *arXiv preprint arXiv:2305.06355*, 2023. 2, 6

[24] Kunchang Li, Yali Wang, Yinan He, Yizhuo Li, Yi Wang, Yi Liu, Zun Wang, Jilan Xu, Guo Chen, Ping Luo, et al. Mvbench: A comprehensive multi-modal video understanding benchmark. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 22195–22206, 2024. 3, 6

[25] Yanwei Li, Chengyao Wang, and Jiaya Jia. Llama-vid: An image is worth 2 tokens in large language models. In *European Conference on Computer Vision*, pages 323–340. Springer, 2024.

[26] Bin Lin, Yang Ye, Bin Zhu, Jiaxi Cui, Munan Ning, Peng Jin, and Li Yuan. Video-llava: Learning united visual representation by alignment before projection. *arXiv preprint arXiv:2311.10122*, 2023. 2, 3, 8

[27] Ji Lin, Hongxu Yin, Wei Ping, Pavlo Molchanov, Mohammad Shoeybi, and Song Han. Vila: On pre-training for visual language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 26689–26699, 2024. 2, 6

[28] Ruyang Liu, Chen Li, Haoran Tang, Yixiao Ge, Ying Shan, and Ge Li. St-llm: Large language models are effective temporal learners. In *European Conference on Computer Vision*, pages 1–18. Springer, 2024. 6

[29] Shilong Liu, Hao Cheng, Haotian Liu, Hao Zhang, Feng Li, Tianhe Ren, Xueyan Zou, Jianwei Yang, Hang Su, Jun Zhu, et al. Llava-plus: Learning to use tools for creating multimodal agents. In *European Conference on Computer Vision*, pages 126–142. Springer, 2024. 3

[30] Zhijian Liu, Ligeng Zhu, Baifeng Shi, Zhuoyang Zhang, Yuming Lou, Shang Yang, Haocheng Xi, Shiyi Cao, Yuxian Gu, Dacheng Li, et al. Nvila: Efficient frontier visual language models. *arXiv preprint arXiv:2412.04468*, 2024. 8

[31] Muhammad Maaz, Hanoona Rasheed, Salman Khan, and Fahad Shahbaz Khan. Video-chatgpt: Towards detailed video understanding via large vision and language models. *arXiv preprint arXiv:2306.05424*, 2023. 3, 6

[32] Sachit Menon, Richard Zemel, and Carl Vondrick. Whiteboard-of-thought: Thinking step-by-step across modalities. *arXiv preprint arXiv:2406.14562*, 2024. 3, 5

[33] Juhong Min, Shyamal Buch, Arsha Nagrani, Minsu Cho, and Cordelia Schmid. Morevqa: Exploring modular reasoning models for video question answering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13235–13245, 2024. 2, 3

[34] Rolandos Alexandros Potamias, Jinglei Zhang, Jiankang Deng, and Stefanos Zafeiriou. Wilor: End-to-end 3d hand localization and reconstruction in-the-wild. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 12242–12254, 2025. 2

[35] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR, 2021. 3

[36] Nikhila Ravi, Valentin Gabeur, Yuan-Ting Hu, Ronghang Hu, Chaitanya Ryali, Tengyu Ma, Haitham Khedr, Roman Rädle, Chloe Rolland, Laura Gustafson, et al. Sam 2: Segment anything in images and videos. *arXiv preprint arXiv:2408.00714*, 2024. 2

[37] Chuyi Shang, Amos You, Sanjay Subramanian, Trevor Darrell, and Roei Herzig. Traveler: A modular multi-lmm agent framework for video question-answering. *arXiv preprint arXiv:2404.01476*, 2024. 2, 3

[38] Aleksandar Shtedritski, Christian Rupprecht, and Andrea Vedaldi. What does clip know about a red circle? visual prompt engineering for vlms. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 11987–11997, 2023. 3

[39] Dídac Surís, Sachit Menon, and Carl Vondrick. Vipergpt: Visual inference via python execution for reasoning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 11888–11898, 2023. 2, 3, 4

[40] Gemini Team, Petko Georgiev, Ving Ian Lei, Ryan Burnell, Libin Bai, Anmol Gulati, Garrett Tanzer, Damien Vincent, Zhufeng Pan, Shibo Wang, et al. Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context. *arXiv preprint arXiv:2403.05530*, 2024. 1, 8

[41] Ao Wang, Hui Chen, Lihao Liu, Kai Chen, Zijia Lin, Jungong Han, et al. Yolov10: Real-time end-to-end object detection. *Advances in Neural Information Processing Systems*, 37:107984–108011, 2024. 2

[42] Jiapeng Wang, Chengyu Wang, Kunzhe Huang, Jun Huang, and Lianwen Jin. Videoclip-xl: Advancing long description understanding for video clip models. *arXiv preprint arXiv:2410.00741*, 2024. 5

[43] Peng Wang, Shuai Bai, Sinan Tan, Shijie Wang, Zhihao Fan, Jinze Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, et al. Qwen2-vl: Enhancing vision-language model's perception of the world at any resolution. *arXiv preprint arXiv:2409.12191*, 2024. 1, 5, 6

[44] Xiaohan Wang, Yuhui Zhang, Orr Zohar, and Serena Yeung-Levy. Videoagent: Long-form video understanding with large language model as agent. In *European Conference on Computer Vision*, pages 58–76. Springer, 2024. 2, 3

[45] Qingyun Wu, Gagan Bansal, Jieyu Zhang, Yiran Wu, Shaokun Zhang, Erkang Zhu, Beibin Li, Li Jiang, Xiaoyun Zhang, and Chi Wang. Autogen: Enabling next-gen llm applications via multi-agent conversation framework. *arXiv preprint arXiv:2308.08155*, 2023. 5

[46] Wenshan Wu, Shaoguang Mao, Yadong Zhang, Yan Xia, Li Dong, Lei Cui, and Furu Wei. Mind's eye of llms: visualization-of-thought elicits spatial reasoning in large language models. *Advances in Neural Information Processing Systems*, 37:90277–90317, 2025. 3, 5

[47] Yixuan Wu, Yizhou Wang, Shixiang Tang, Wenhao Wu, Tong He, Wanli Ouyang, Philip Torr, and Jian Wu. Dettoolchain: A new prompting paradigm to unleash detection ability of mllm. In *European Conference on Computer Vision*, pages 164–182. Springer, 2024. 3

[48] Lin Xu, Yilin Zhao, Daquan Zhou, Zhijie Lin, See Kiong Ng, and Jiashi Feng. Pllava: Parameter-free llava extension from images to videos for video dense captioning. *arXiv preprint arXiv:2404.16994*, 2024. 3, 6

[49] Fuzhao Xue, Yukang Chen, Dacheng Li, Qinghao Hu, Ligeng Zhu, Xiuyu Li, Yunhao Fang, Haotian Tang, Shang Yang, Zhijian Liu, et al. Longvila: Scaling long-context visual language models for long videos. *arXiv preprint arXiv:2408.10188*, 2024. 8

[50] Dongjie Yang, Suyuan Huang, Chengqiang Lu, Xiaodong Han, Haoxin Zhang, Yan Gao, Yao Hu, and Hai Zhao. Vript: A video is worth thousands of words. *arXiv preprint arXiv:2406.06040*, 2024. 2, 6

[51] Jianwei Yang, Hao Zhang, Feng Li, Xueyan Zou, Chunyuan Li, and Jianfeng Gao. Set-of-mark prompting unleashes extraordinary visual grounding in gpt-4v. *arXiv preprint arXiv:2310.11441*, 2023. 3

[52] Zhengyuan Yang, Linjie Li, Jianfeng Wang, Kevin Lin, Ehsan Azarnasab, Faisal Ahmed, Zicheng Liu, Ce Liu, Michael Zeng, and Lijuan Wang. Mm-react: Prompting chatgpt for multimodal reasoning and action. *arXiv preprint arXiv:2303.11381*, 2023. 3

[53] Zongxin Yang, Guikun Chen, Xiaodi Li, Wenguan Wang, and Yi Yang. Doraemongpt: Toward understanding dynamic scenes with large language models (exemplified as a video agent). In *Forty-first International Conference on Machine Learning*, 2024. 2, 3

[54] Jiabo Ye, Haiyang Xu, Haowei Liu, Anwen Hu, Ming Yan, Qi Qian, Ji Zhang, Fei Huang, and Jingren Zhou. mplug-owl3: Towards long image-sequence understanding in multi-modal large language models. In *The Thirteenth International Conference on Learning Representations*, 2024. 8

[55] Boqiang Zhang, Kehan Li, Zesen Cheng, Zhiqiang Hu, Yuqian Yuan, Guanzheng Chen, Sicong Leng, Yuming Jiang, Hang Zhang, Xin Li, et al. Videollama 3: Frontier multi-modal foundation models for image and video understanding. *arXiv preprint arXiv:2501.13106*, 2025. 8

[56] Hang Zhang, Xin Li, and Lidong Bing. Video-llama: An instruction-tuned audio-visual language model for video understanding. *arXiv preprint arXiv:2306.02858*, 2023. 2, 3, 6

[57] Jinglei Zhang, Jiankang Deng, Chao Ma, and Rolandos Alexandros Potamias. Hawor: World-space hand motion reconstruction from egocentric videos. In *Proceedings of the Computer Vision and Pattern Recognition Conference (CVPR)*, pages 1805–1815, 2025. 2

[58] Yuanhan Zhang, Jinming Wu, Wei Li, Bo Li, Zejun Ma, Ziwei Liu, and Chunyuan Li. Video instruction tuning with synthetic data. *arXiv preprint arXiv:2410.02713*, 2024. 2, 8