








# *VideoAgent*: A Memory-augmented Multimodal Agent for Video Understanding

Yue Fan<sup>\*1</sup>, Xiaojian Ma<sup>\*†1</sup>, Rujie Wu<sup>1,2</sup>, Yuntao Du<sup>1</sup>, Jiaqi Li<sup>1</sup>, Zhi Gao<sup>1,3</sup>, and Qing Li<sup>†1</sup>

<sup>1</sup> State Key Laboratory of General Artificial Intelligence, BIGAI, Beijing, China

<sup>2</sup> School of Computer Science, Peking University, Beijing, China

<sup>3</sup> School of Intelligence Science and Technology, Peking University, Beijing, China

{maxiaojian, liqing}@bigai.ai

<https://videoagent.github.io>

**Abstract.** We explore how reconciling several foundation models (large language models and vision-language models) with a novel unified memory mechanism could tackle the challenging video understanding problem, especially capturing the long-term temporal relations in lengthy videos. In particular, the proposed multimodal agent *VideoAgent*: 1) constructs a structured memory to store both the generic temporal event descriptions and object-centric tracking states of the video; 2) given an input task query, it employs tools including video segment localization and object memory querying along with other visual foundation models to interactively solve the task, utilizing the zero-shot tool-use ability of LLMs. *VideoAgent* demonstrates impressive performances on several long-horizon video understanding benchmarks, an average increase of 6.6% on NExT-QA and 26.0% on EgoSchema over baselines, closing the gap between open-sourced models and private counterparts including Gemini 1.5 Pro. The code and demo can be found at <https://videoagent.github.io>.

**Keywords:** video understanding · LLMs · tool-use · multimodal agents

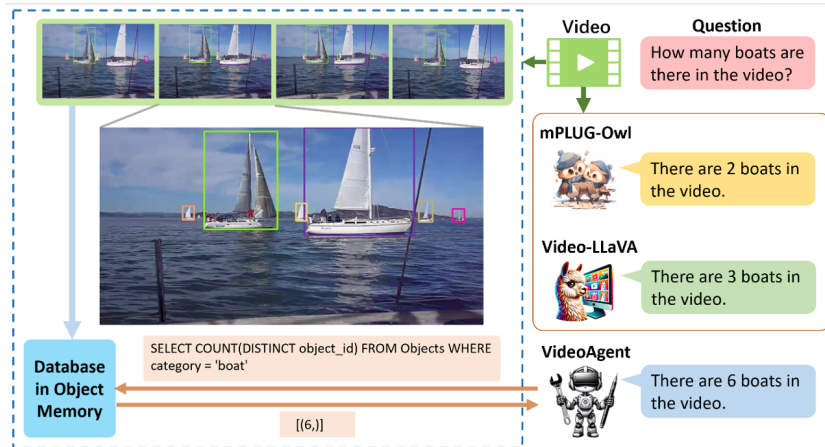
## 1 Introduction

Understanding videos and answering free-form queries (question answering, content retrieval, *etc.*) remains a major challenge in computer vision and AI [1, 8, 9, 13, 15, 22, 23, 26, 31, 47]. Notably, much of the recent progress has achieved by the end-to-end pretrained large transformer models, especially those are developed upon the powerful large language models (LLMs) [3, 12, 22, 31], *i.e.* multimodal LLMs. However, there have been increasing concerns about their capabilities to handle long-form videos with rich events and complex spatial-temporal dependencies [6, 8–10, 16, 24, 33]. Specifically, the computation, especially memory cost could grow significantly and even become prohibitively expensive

---

\*Equal contribution.

†Corresponding authors.



**Fig. 1:** A comparison between *VideoAgent* and end-to-end multimodal LLMs on video question answering. Without a unified memory as a structured representation for videos, end-to-end models could struggle with capturing basic spatial-temporal details, especially when asked about objects on lengthy videos. While *VideoAgent* can utilize a curated set of tools to perform sophisticated queries about the *temporal memory* (not shown) and *object memory*, and respond with the correct answer.

when processing lengthy videos [26, 32]. Also, the self-attention mechanism could sometimes struggle to capture the long-range relations [25]. These issues have hindered further advancement in applying sophisticated foundation models to video understanding.

More recently, thanks to the tool-use capabilities of LLMs [2, 20], there has been rapid development of a new class of multimodal understanding approaches: *multimodal agents* [5, 13, 23, 34]. The key idea is prompting LLMs into solving the multimodal tasks by invoking several **tool** foundation models (object detection, visual question answering, *etc.*) interactively. These methods have great potential as they are mostly training-free and flexible with tool sets. However, extending them to video understanding, especially on long-form videos is **non-trivial**. Simply adding video foundation models as tools could still suffer from the computation cost and attention limitation issues [12, 22]. Other research has explored more sophisticated prompting strategies with better tools [14, 30, 37], but they usually lead to complicated pipelines and the performances of these methods still fail to match their end-to-end counterparts possibly due to a lack of video-specific agent design.

In this paper, we introduce a simple yet effective LLM-based multimodal tool-use agent *VideoAgent* for video understanding tasks. Our **key insight** is to represent the video as a structured unified memory, therefore facilitating strong spatial-temporal reasoning and tool use of the LLM, and matching/outperforming end-to-end models, as shown in Fig. 1. Our memory design is **motivated** by the principle of being minimal but sufficient: we’ve found that the overall event context descriptions and temporally consistent details about objects could cover

the most frequent queries about videos. As a result, we design two memory components: 1) *temporal memory*, which stores text descriptions of each short (2 seconds) video segment sliced from the complete video; 2) *object memory*, where we track and store the occurrences of objects and persons in the video. To answer a query, the LLM will decompose it into several subtasks and invoke the tool models. The unified memory is centered around by the following tools: 🍷 *caption retrieval*, which will return all the event descriptions between two query time steps; 🍷 *segment localization*, which retrieves a short video segment of a given textual query by comparing it against the event descriptions within the temporal memory; 🍷 *visual question answering*, which answers a question given a retrieved video segment; 🍷 *object memory querying*, which allows sophisticated object state retrieval from the object memory using SQL queries. Finally, the LLM will aggregate the response of the interactive tool use and produce an answer to the input query.

We conduct extensive evaluations of *VideoAgent* on several video understanding tasks, including free-form query localization with Ego4D NLQ [4], generic video question answering with WorldQA [46] and NExT-QA [35], and egocentric question answering with EgoSchema [15], a recent benchmark focusing on complex questions about long-form videos. We compare *VideoAgent* against both the canonical end-to-end multimodal LLMs and other multimodal agents. Results demonstrate the advantages of *VideoAgent*: on averaged increasing 6.6% on NExT-QA and 26.0% on EgoSchema over baselines. Our further investigation has examined the role played by the unified memory and tool selection.

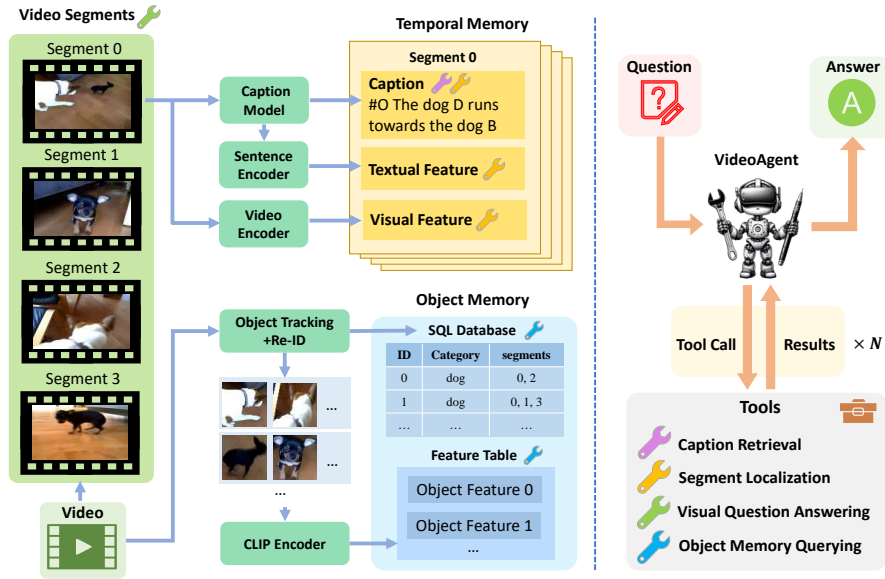
To summarize, our contributions are as follows:

- We propose a unified memory mechanism to build structured representations for long-form videos, including a *temporal memory* that stores segment-level descriptions and an *object memory* that tracks the state of objects in the video.
- Based on the unified memory, we design *VideoAgent*, an LLM-powered multimodal agent for video understanding. It decomposes the input task queries and interactively invokes tools to retrieve information from the memory until reaches the final response.
- We perform thorough evaluations of *VideoAgent* on multiple video understanding benchmarks against both end-to-end multimodal LLMs and multimodal agent baselines, demonstrating the effectiveness of *VideoAgent*. The additional ablation analysis further confirms the crucial design choices we’ve made.

## 2 VideoAgent

### 2.1 Overview

We illustrate the proposed *VideoAgent* in Fig. 2. It begins with converting the input video into a unified representation: *temporal memory* (Sec. 2.2) and *object memory* (Sec. 2.3). For any incoming task, it interactively invokes tools to collect information from the memory and the raw video segments, and ultimately produces a response (Sec. 2.4). The memory construction and task-solving (inference) procedures are summarized in Algorithm 1 and Algorithm 2, respectively.



**Fig. 2:** An overview of *VideoAgent*. Left: We first translate an input video into structured representations: a temporal memory and an object memory; Right: the LLM within *VideoAgent* will be prompted to solve the given task by interactively invoking tools (🔧🔧🔧). Our considered tools primarily work with the memory (e.g. 🏷️ interacts with the caption part of the temporal memory while 🔍 looks up the object memory).

## 2.2 Temporal Memory $\mathcal{M}_T$

The temporal memory is designed to store overall event context descriptions and features of videos. Given  $n$  video segments  $[v_1, \dots, v_n]$  sliced from a video  $V$ , we extract video segment caption  $s_{\text{caption}}$ , video segment feature  $e_{\text{video}}$  and the caption text embedding  $e_{\text{caption}}$ :

**Video segment caption.** We use a pretrained video captioning model called LaViLa [49] to produce captions for each video segment. Specifically, it takes 4 frames from a 2-second segment to produce a short caption sentence. Typical LaViLa captions can be "#C C cuts a wood with a wood cutter" and "#O The man Y pushes a stroller on the road with his left hand", where "#C" and "#O" is used to denote whether the caption sentence is about the camera wearer or someone other than the camera wearer, therefore making LaViLa captions effective in both egocentric and generic videos.

**Video segment feature and caption feature.** To obtain the video segment feature, we adopt the video encoder of ViCLIP [28] to encode video segments. We uniformly sampled 10 frames from each video segment as the input to ViCLIP, and save the generated feature of the segment. For the caption feature, we choose



**Fig. 3:** A visualization of object tracking and re-ID. 6 frames from a video are displayed in order. The cup (light green box) and the milk bottle (pink box) are successfully re-identified in different postures.

`text-embedding-3-large`<sup>1</sup> offered by OpenAI to compute the embedding of the video segment caption we obtained from LaViLa.

### 2.3 Object Memory $\mathcal{M}_O$

In addition to the general video event context stored in the temporal memory, it is also crucial to explicitly capture the temporally consistent details: *e.g.* the presence of people, objects, and the surroundings, *etc.* The intuition is that most queries about videos are object(person)-related; therefore, the occurrences of objects (and people) are tracked and stored in the *object memory*. Specifically, object memory constitutes: 1) a feature table that connects object visual features with unique object identifiers; 2) a SQL database that stores the object(person) occurrence information across the video. Details on the construction can be found below:

**Tracking and re-identification.** At the heart of our object memory construction pipeline is tracking all the objects across the video, and re-identifying (re-ID) previously appeared objects to eliminate object duplication. We pipeline an object detection model RT-DETR [48] with a multi-object tracker ByteTrack [45] for the object discovery and tracking part. This combination produces tracking IDs, categories, and bounding boxes of the tracked object occurrences in the video frames. In this phase, an object may have multiple tracking IDs due to its multiple occurrences in the video. For the re-ID part, the key idea is to first compute the features of all the object occurrences that have been discovered and tracked, then group them into object IDs based on their feature similarities. More specifically, the feature of an object occurrence (a tracking ID) is generated on

<sup>1</sup> <https://platform.openai.com/docs/guides/embeddings>

object images cropped from 10 randomly sampled frames of the tracking ID; we also follow a recent study [27] to use an ensemble of CLIP [19] and DINOv2 [18] feature similarity to group tracking IDs into object IDs:


$$\begin{aligned} \text{CLIP}(i, j) &= \frac{1}{1 + \exp[-20 * (\text{cosine}(e_i^{\text{CLIP}}, e_j^{\text{CLIP}}) - 0.925)]}, \\ \text{DINOv2}(i, j) &= \frac{1}{1 + \exp[-4.1 * (\text{cosine}(e_i^{\text{DINOv2}}, e_j^{\text{DINOv2}}) - 0.5)]}, \\ \text{sim}(i, j) &= 0.15 * \text{CLIP}(i, j) + 0.85 * \text{DINOv2}(i, j), \end{aligned}$$


where  $\text{cosine}(\cdot, \cdot)$  denotes cosine similarity,  $e_i^{\text{CLIP}}, e_j^{\text{CLIP}}$  and  $e_i^{\text{DINOv2}}, e_j^{\text{DINOv2}}$  are the CLIP and DINOv2 features of the tracking ID  $i$  and  $j$ , respectively. The hyperparameters above (coefficients and biases) are tuned with a simple grid search on EgoObjects [50]. More details about re-ID can be found in *Appendix*. An example of how our tracking and re-ID pipeline manages to handle the temporally discontinuous object presence in a kitchen can be found in Fig. 3.


**Feature table.** Assuming we’ve identified all objects (Object IDs) from the video and their object occurrences (tracking IDs) have been confirmed as well. We compute the CLIP feature  $f_{\text{object}}^{s_{\text{id}}}$  of object ID  $s_{\text{id}}$  by averaging the CLIP features of its tracking IDs, and store both the CLIP feature and the object ID in a table. This allows us to use free-form language queries (*e.g.* “red cup”) to search for objects in the video.

**SQL database.** Further, we build a relational database with three fields: object ID  $s_{\text{id}}$ , object category  $s_{\text{category}}$ , and indices of video segments  $\{I_1, \dots, I_t\}$  where the object has appeared. Later, this database can be queried using SQL code and support sophisticated querying logic.

## 2.4 Tools and Inference

Compared to counterparts that offer a large collection of tools and usually result in ambiguity in tool calling and complex tool-use pipeline, in *VideoAgent*, our design principle is to provide a minimal but sufficient tool set with a focus on querying the memory. We find this simplifies the inference procedures as well as leads to better performances. We consider the following tools ():

 **Caption retrieval.** The goal is to extract the captions from specified video segments. Concretely, given the temporal memory  $\mathcal{M}_T$ , a start and an end time step  $t_{\text{start}}$  and  $t_{\text{end}}$  as arguments, the tool `caption_retrieval()` simply retrieves these captions from the temporal memory directly. Due to the context limit, the longest time window allowed is 15 segments, *i.e.*  $t_{\text{end}} < t_{\text{start}} + 15$ .

 **Segment localization.** The goal is to localize a video segment given a text query  $s_{\text{query}}$ . The tool `segment_localization()` will compare the text feature of  $s_{\text{query}}$  against the video features in the temporal memory  $\mathcal{M}_T$ . Specifically, we consider an ensemble of the query–video similarity (made possible by ViCLIP [28], a pretrained video-text CLIP model) and the query–caption similarity (both text features are computed by `text-embedding-3-large` offered by OpenAI). Top-5 video segments will be returned by this tool.

---

**Algorithm 1:** Memory construction of *VideoAgent*.
 


---


**Input:** video  $V$ , video captioning model `video_cap( $\cdot$ )`, video embedding model `video_emb( $\cdot$ )`, text embedding model `text_emb( $\cdot$ )`, video object tracker with re-identification `object_track_reid( $\cdot$ )`

**Output:** temporal memory  $\mathcal{M}_T$ , object memory  $\mathcal{M}_O$

- 1 Initialize  $\mathcal{M}_T = \emptyset$ ,  $\mathcal{M}_O = \emptyset$ ;
- 2 Slicing video into  $n$  short segments  $V = [v_1, v_2, \dots, v_n]$  (each segment spans approximately 2 seconds);
- 3 **for**  $v_i$  *in*  $[v_1, v_2, \dots, v_n]$  **do**
- 4      $s_{\text{caption}} \leftarrow \text{video\_cap}(v_i)$ ;
- 5      $e_{\text{video}} \leftarrow \text{video\_emb}(v_i)$ ;
- 6      $e_{\text{text}} \leftarrow \text{text\_emb}(s_{\text{caption}})$ ;
- 7      $\mathcal{M}_T = \mathcal{M}_T + (s_{\text{caption}}, e_{\text{video}}, e_{\text{text}})$
- 8  $\text{results} \leftarrow \text{object\_track\_reid}(V)$ ;
- 9 **for**  $S$  *in*  $\text{results}$  **do**
- 10      $s_{\text{id}}, s_{\text{category}}, \{I_1, \dots, I_k\}, f_{\text{object}}^{\text{id}} \leftarrow S$  //See Sec. 2.3;
- 11      $\mathcal{M}_O = \mathcal{M}_O + (s_{\text{id}}, s_{\text{category}}, \{I_1, \dots, I_k\}, f_{\text{object}}^{\text{id}})$ ;
- 12 **return**  $\mathcal{M}_T, \mathcal{M}_O$ ;

---

 **Visual question answering.** The goal is to answer a given question  $s_{\text{question}}$  about a short video segment at time  $t_{\text{target}}$ , allowing to gather extra information that is not covered by the captions in temporal memory or states in object memory. Concretely, we run Video-LLaVA [12] when the tool `visual_question_answering( $\cdot$ )` is called.

 **Object memory querying.** The goal is to perform sophisticated information retrieval about objects that appeared in the video from the object memory  $\mathcal{M}_O$ . Specifically, when calling the tool `object_memory_querying( $\cdot$ )` with a text query  $s_{\text{query}}$  (e.g. “How many red cups did I take out from the fridge?”), relevant object descriptions will first be extracted from the query (e.g. “red cup”); next, we compare the text feature of the descriptions (obtained from CLIP [19]) against the object features from the feature table in  $\mathcal{M}_O$  to obtain the object IDs likely correspond to the descriptions; finally, the LLM will write SQL code based on both  $s_{\text{query}}$  and the retrieved object IDs to query the database in  $\mathcal{M}_O$  and obtain the needed information (segments that the objects appeared, *etc.*). After being further processed by the LLM, a response to  $s_{\text{query}}$  will be returned.

The inference procedure of *VideoAgent* is rather straightforward. Starting with a history buffer  $h$  initialized with the input query  $q$ , *VideoAgent* decides which tool to use, calls the tool with the produced arguments, appends the results to the buffer, and repeats until it decides to stop or a maximum number of steps is reached. Finally, a response will be made based on the content in the

---

**Algorithm 2:** Inference of *VideoAgent*.

---

**Input:** task instruction  $q$ , temporal memory  $\mathcal{M}_T$ , object memory  $\mathcal{M}_O$ , LLM  $\text{LLM}(\cdot)$ , a set of tools (see Sec. 2.4)  
**Output:** response  $a$

- 1 Initialize history  $h = [q]$ ;
- 2 Initialize inference step count  $c = 0$ ;
- 3 **while**  $c < \text{MAX\_STEP}$  **do**
- 4     action, input =  $\text{LLM}(h)$ ;
- 5     **if**  $\text{action} == \text{"caption\_retrieval"}$  **then**
- 6          $t_{\text{start}}, t_{\text{end}} \leftarrow \text{input}$ ;
- 7         results  $\leftarrow$  🗑️  $\text{caption\_retrieval}(t_{\text{start}}, t_{\text{end}}, \mathcal{M}_T)$ ;
- 8     **else if**  $\text{action} == \text{"segment\_localization"}$  **then**
- 9          $s_{\text{query}} \leftarrow \text{input}$ ;
- 10         results  $\leftarrow$  🗑️  $\text{segment\_localization}(s_{\text{query}}, \mathcal{M}_T)$ ;
- 11     **else if**  $\text{action} == \text{"visual\_question\_answering"}$  **then**
- 12          $s_{\text{question}}, t_{\text{target}} \leftarrow \text{input}$ ;
- 13         results  $\leftarrow$  🗑️  $\text{visual\_question\_answering}(s_{\text{question}}, t_{\text{target}})$ ;
- 14     **else if**  $\text{action} == \text{"object\_memory\_querying"}$  **then**
- 15          $s_{\text{query}} \leftarrow \text{input}$ ;
- 16         results  $\leftarrow$  🗑️  $\text{object\_memory\_querying}(s_{\text{query}}, \mathcal{M}_O)$ ;
- 17     **else if**  $\text{action} == \text{"stop"}$  **then**
- 18         **break**;
- 19      $h = h + [(\text{action}, \text{input}, \text{results})]$ ;
- 20      $c = c + 1$ ;
- 21 **return**  $a = \text{LLM}(h)$ ;

---

history buffer. We provide an example of this procedure in Fig. 4. *VideoAgent* is implemented using LangChain<sup>2</sup> with GPT-4 as the main LLM.

### 3 Capabilities and Analysis

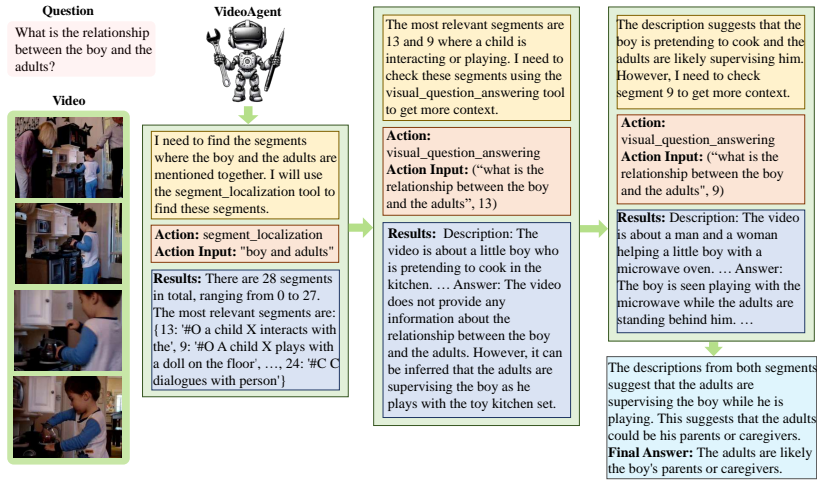
We evaluate *VideoAgent* on various long-form video understanding benchmarks including EgoSchema (Sec. 3.1), Ego4D Natural Language Queries (Sec. 3.2), WorldQA (Sec. 3.3) and NExT-QA (Sec. 3.4), and the performances are compared against state-of-the-art end-to-end multimodal LLMs and multimodal agents.

#### 3.1 EgoSchema

**Overview.** EgoSchema [15] is a benchmark that contains around 5000 questions about long-form videos. The questions typically involve challenging video-level

<sup>2</sup> <https://www.langchain.com/>





**Fig. 4:** An examples of the *VideoAgent* inference. Given a question, *VideoAgent* executes multiple tool-use steps until it reaches the answer. The yellow, red, and blue blocks in each step denote the chain of thought, action to be taken, and results of tool use.

**Table 1:** Accuracy results on the EgoSchema dataset. Top row: results on the full EgoSchema test set; Bottom row: results on the EgoSchema 500 subset.

EgoSchema (full set)					
FrozenBiLM	InternVideo	mPLUG-Owl	LLoVi	Gemini 1.5 Pro	<i>VideoAgent</i>
26.9	32.0	30.2	50.3	<b>63.2</b>	60.2
EgoSchema (subset, 500 questions)					
SeViLA	Video-LLaVA	mPLUG-Owl	LLoVi	ViperGPT	<i>VideoAgent</i>
25.8	36.8	33.8	51.8	15.8	<b>62.8</b>

reasoning such as “describe the general activity in the room and how the different characters and their actions contribute to this environment”. *VideoAgent* is both tested on the full 5031-question test set and the official 500-question subset. The comparative methods include SeViLA [39], Video-LLaVA [23], mPLUG-Owl [38], ViperGPT [40], LLoVi [41], FrozenBiLM [36], InternVideo [29] and Gemini 1.5 Pro<sup>3</sup>.

**Main results.** In Tab. 1, *VideoAgent* significantly outperforms other state-of-the-art video understanding models such as SeViLA and Video-LLaVA to nearly 30 percent, achieving an accuracy of 62.8 on the 500 questions. Besides, *VideoAgent* achieves 60.2 on the full test set, closing to the performance of Gemini 1.5 Pro. The strong performance of *VideoAgent* on EgoSchema proves that *VideoAgent* can solve complex video tasks on long-form videos better than multimodal LLMs and agent counterparts.

<sup>3</sup> [https://storage.googleapis.com/deepmind-media/gemini/gemini\\_v1\\_5\\_report.pdf](https://storage.googleapis.com/deepmind-media/gemini/gemini_v1_5_report.pdf)

**Table 2:** Comparison between supervised baselines and *VideoAgent* with different tool implementation variants on Ego4D NLQ validation set.

EGO4D NLQ Val.				
Method	<i>R1@0.3</i>	<i>R1@0.5</i>	<i>R5@0.3</i>	<i>R5@0.5</i>
<b>Supervised</b>				
2D-TAN	5.04	2.02	12.89	5.88
VSLNet	5.45	3.12	10.74	6.63
GroundNLQ	<b>27.20</b>	<b>18.91</b>	<b>54.42</b>	<b>39.98</b>
<b>Zero-Shot (<i>VideoAgent</i> with 🗡️ <code>segment_localization</code> variants)</b>				
ViCLIP	8.40	3.97	17.36	8.50
LaViLa	10.07	4.19	22.53	10.58
Ego4D	16.41	6.96	31.96	15.01
LaViLa+ViCLIP	11.13	4.76	25.31	12.08
Ego4D+ViCLIP	<b>17.39</b>	<b>7.47</b>	<b>33.05</b>	<b>15.73</b>

**Unified memory facilitates stronger reasoning.** The questions in EgoSchema are rather complex in terms of the underlying reasoning about the lengthy videos. Therefore, strong spatial-temporal reasoning is essential. What canonical approaches like multimodal LLMs (Video-LLaVA, *etc.*) or counterpart multimodal agents (ViperGPT) have in common is the lack of a unified memory as a structured representation for the videos. Without such representation, the reasoning has to be either implicit (as in end-to-end models) or quite limited by the available tools (as in ViperGPT), results in worse performances than ours.

**Holistic video understanding with flexible tool-use.** Given a typical question such as "how did c's behavior evolve throughout the video, and what stages of engagement with the tasks can you identify?", it is hard to derive a descriptive text from the question and use it for video grounding, which is a common way for multimodal LLMs (SeViLA, *etc.*) to select limited key frames for the visual input. However, apart from the 🗡️ `segment_localization`, *VideoAgent* can also use 🗡️ `caption_retrieval` to grab the main context of the video and decide which segments are critical, therefore tackling this obstacle.

### 3.2 Ego4D Natural Language Queries

**Overview.** The task of Ego4D Natural Language Queries [4] is to locate a temporal window (9 seconds on average) in the video (9 minutes on average) that can best answer a query. *VideoAgent* is evaluated zero-shot with different variants of the 🗡️ `segment_localization` tool using 1) ViCLIP visual features only; 2) textual features based on LaViLa captions or Ego4D ground-truth narrations; 3) a combination of both textual features and visual features (LaViLa+ViCLIP and Ego4D+ViCLIP, the ensemble weights can be found in *Appendix*). The methods for comparison include 2D-TAN [44], VSLNet [43], and GroundNLQ [7], which ranked first in Ego4D NLQ challenge 2023.

**Main results.** Tab. 2 presents the results on the validation set of Ego4D NLQ. A combination of both textual features and visual features in *VideoAgent* results in

**Table 3:** Comparison between two zero-shot approach: *VideoAgent* and LifeLongMemory [30] on Ego4D NLQ. \*The performances of LifelongMemory on ***R1@0.3*** and ***R5@0.3***, although not reported, must be less or equal than ***R@0.3***.

Method	<i>R1@0.3</i>	<i>R5@0.3</i>	<i>R@0.3</i>
LifeLongMemory(Ego4D)	*	*	15.99
LifeLongMemory(LaViLa)	*	*	9.74
<i>VideoAgent</i> (Ego4D)	<b>16.41</b>	<b>31.96</b>	-
<i>VideoAgent</i> (LaViLa)	10.07	22.53	-

**Table 4:** Results on WorldQA.

WorldQA					
Method	Video-LLaMA	Video-ChatGPT	Video-LLaVA	GPT-4V	VideoAgent
<b>Open-Ended</b>	26.80	28.51	30.15	<b>35.37</b>	32.53
<b>Multi-Choice</b>	4.81	13.25	35.25	32.83	<b>39.28</b>

better video grounding. Although having a performance gap with the supervised GroundNLQ, *VideoAgent* outperforms 2D-TAN and VSLNet and achieves good performance considering its simple architecture and zero-shot characteristics.

**Caption features vs. visual features.** It can be inferred from the comparison among ViCLIP, LaViLa and Ego4D that it is more effective to use the caption-query similarities for video grounding than using video-query similarities. Higher quality captions (LaViLa→Ego4D) will also lead to better performance.

**Similarity-based vs. LLM-based localization.** Tab. 3 presents a comparison between *VideoAgent* and LifeLongMemory [30]. Given a query, LifeLongMemory uses GPT-4 to digest and refine the captions of the video segments, and outputs a list of candidate windows to the query based on the captions selected by the LLM. LifeLongMemory adopts a customized *R@0.3* metric to calculate the proportion of the predictions where at least one out of all the LLM-generated candidate windows achieves an *IoU* greater than 0.3 with the ground-truth window. It can be inferred from Tab. 3 that given the same caption type (Ego4D or LaViLa), the performance of *VideoAgent* on *R1@0.3* where only 1 candidate is allowed for a query, has already surpassed the performance of LifeLongMemory on *R@0.3*. By providing 5 candidates for a query, the performance of *VideoAgent* will exceed LifeLongMemory by more than two-fold. This indicates that similarity-based segment localization is more effective than the LLM-based segment localization.

### 3.3 WorldQA

**Overview.** WorldQA [46] is a challenging video understanding benchmark that focuses on using world knowledge and long-chain reasoning to understand a long-form video (typically a 5-minute movie). We compared VideoAgent with Video-LLaMA [42], Video-ChatGPT [14], Video-LLaVA [12] and GPT-4V [17] on both generation-based Open-Ended QA and Multi-Choice QA.

**Table 5:** Results on NExT-QA. We compare baselines on both the original full set as reference and the subset (600 questions) due to the evaluation cost.

NExT-QA				
Method	Temporal	Causal	Descriptive	Average
Val. Set				
InternVideo	43.4	48.0	65.1	49.1
SeViLA(zero-shot)	<b>61.3</b>	<b>61.5</b>	<b>75.6</b>	63.6
TCR(pre-training)	-	-	-	<b>66.1</b>
Val. Subset (600)				
ViperGPT	17.0	19.0	26.5	20.8
mPLUG-Owl	36.0	41.0	52.5	43.2
Video-LLaVA	42.0	53.5	65.0	53.5
SeViLA(zero-shot)	56.0	66.5	70.0	64.2
<i>VideoAgent</i>	<b>60.0</b>	<b>76.0</b>	<b>76.5</b>	<b>70.8</b>

**Main results.** Tab. 4 shows that VideoAgent surpasses existing open-source multimodal LLMs by a significant margin on both Open-Ended QA and Multi-Choice QA. This can be mainly contributed to the rich world knowledge and the intrinsic reasoning ability of the LLM agent. Moreover, the better accuracy of VideoAgent compared to that of GPT-4V on Multi-Choice QA demonstrates the effectiveness of the structured memory in understanding long-form videos. On the open-ended QA, GPT-4V achieves better results than VideoAgent, mainly because it has video frames as visual conditions for generating better responses.

### 3.4 NExT-QA

**Overview.** NExT-QA [35] is a benchmark containing temporal, causal and descriptive multi-choice questions about videos. The accuracy *acc* is computed for each type of the questions. For the reason of cost, we randomly sampled 200 questions for each type and obtained a subset of 600 questions in total to test the performance of *VideoAgent*. Methods directly compared with *VideoAgent* on this subset include ViperGPT [23], mPLUG-Owl [38], Video-LLaVA [12] and SeViLA [39]. The results of three representative methods InternVideo [29], SeViLA [39] and TCR [10] on the full validation set are also provided.

**Main results.** Tab. 5 shows the main results on NExT-QA. In all, *VideoAgent* achieves the strongest performances among all comparative methods. Particularly, on the challenging causal questions that require strong temporal understanding and reasoning, *VideoAgent* outperforms SeViLA, one of the state-of-the-art models on NExT-QA, for nearly 10 percent. Besides, the comparison between *VideoAgent* and Video-LLaVA, which is used by the 🦜 `video_question_answering` tool, indicates that our *VideoAgent* allows such multimodal LLM to work better as part of the multimodal tool-use agent than being used alone.

**Settings for ablation studies.** We extract 50 questions for each question type from the 600-question subset, resulting in a subset of 150 questions in total, to

**Table 6:** The effectiveness of different components of *VideoAgent* on NExT-QA subset. ✓ and ✗ indicates whether or not the tool is included. "w/ re-ID" uses an object memory constructed with re-ID, while "w/o re-ID" uses an object memory that might include duplicated objects.

Type	VQA	Grounding	Captions	Database	Tem.	Cau.	Des.	Avg.
1	GPT-4V	✓	✓	w/ re-ID	64.0	78.0	82.0	74.7
2	Video-LLaVA	✓	✓	w/ re-ID	60.0	74.0	80.0	71.3
3	Video-LLaVA	✓	✓	✗	46.0	64.0	78.0	62.7
4	✗	✗	✓	w/ re-ID	48.0	52.0	68.0	56.0
5	✗	✗	✓	w/o re-ID	46.0	46.0	54.0	48.7
6	✗	✗	✓	✗	34.0	46.0	42.0	40.7

evaluate the contributions of different components in *VideoAgent* as ablation studies. Tab. 6 shows the performances of 6 ablations of *VideoAgent*, with each equipped with a unique set of tools among 🍷 *visual question answering*, 🍷 *segment localization*, 🍷 *caption retrieval* and 🍷 *object memory querying*, denoted as ‘VQA’, ‘Grounding’, ‘Captions’ and ‘Database’ in Tab. 6.

**The necessity of caption retrieval.** The 🍷 *caption retrieval* tool lays the foundation for *VideoAgent* since it provides the basic information about the main context of the video. With 🍷 *caption retrieval* only, *VideoAgent* of type 6 achieves an average result of 40.7 already, which is comparable to the performance 43.2 of the end-to-end video-language model mPLUG-Owl on the 600-question subset.

**Object memory boosts all question types.** The comparison between type 2 and 3 indicates that a reliable object memory can substantially help with temporal and causal questions since it offers crucial temporally consistent object information across video segments, facilitates object-related temporal localization, and enhances the agent’s understanding of the video. The performance gap between type 4 and type 5 suggests that with the object re-ID algorithm, the performance on descriptive questions (mostly about quantity) will be significantly improved, validating the effectiveness of object re-ID.

**VQA and segment localization offer the most bonus.** By comparing type 3 and 6, it can be seen that simultaneously adding 🍷 *visual question answering* and 🍷 *segment localization* boost the caption-only *VideoAgent* by 22 percent on the average performance, compared to 15.3 percent boost by adding the object memory (inferred from type 4 and 6). Moreover, by switching from Video-LLaVA to GPT-4V in 🍷 *visual question answering* (type 1 and 2), the performance will be raised by 3.4 percent, indicating that accurate visual details identified by the powerful VQA model will aid in better question answering performance.

## 4 Related Work

### 4.1 Multimodal LLMs for video understanding

Since LLMs have demonstrated an excellent ability to process and understand natural language [3, 17], several recent works have explored extending them

to multimodal setting, especially for images and videos [1, 11, 12, 21, 26, 47]. LaViLa [49] manages to create a massive and diverse set of text as automatic video narrators for video-text contrastive representation pretraining. VideoLLaMA [42] enables video comprehension by capturing the temporal changes in visual scenes and integrating audio-visual signals for better cross-modal training. As we discussed in Sec. 1, many of these multimodal foundation models could struggle with long-form video understanding. To remedy this, LSTP [31] utilize spatial and temporal sampler modules to extract optical flow based temporal features and aligned spatial relations from the video to achieve long-form video understanding; Gemini [26] scales the multimodal models to longer videos with tens of thousands of TPUs and massive private video-text datasets. Albeit the prompt progress made by these end-to-end models, prohibitive computation costs and the inherent limitation of the transformer on long-form videos remain significant in applying these end-to-end learned multimodal foundation models to video understanding.

#### 4.2 Multimodal tool-use agents for video understanding

Another line of research focuses on augmenting LLMs with a set of **tools** to solve multimodal tasks without costly training. In particular, LLMs within these **multimodal agents** are prompted to produce a step-by-step plan to address the original task, and interactively invoke several multimodal foundation models (“tools”), *e.g.* captioning, VQA, *etc.* VisProg [5] pilots this direction by equipping the GPT-3 planner with a large collection of visual tools, solving complex real-world visual reasoning problems. Applying these agents to video understanding requires careful design as many of the tool models do not guarantee generalization to videos. LifeLongMemory [30] employs natural language video narrations to create a text-based episodic memory and prompt LLMs to reason and retrieve required information for the downstream task. DoraemonGPT [37] introduces a sophisticated prompting strategy with Monte Carlo Tree Search (MCTS) to invoke both tools and a structured memory to solve video understanding tasks. These multimodal agents have great potential but so far they mostly struggle with attaining on-par performances to their end-to-end foundation model counterparts on common benchmarks, likely due to the complicated pipelines and lack of video-specific design.

## 5 Conclusions

We’ve presented *VideoAgent*, a multimodal tool-use agent that reconciles several foundation models with a novel unified memory mechanism for video understanding. Compared to end-to-end multimodal LLMs and tool-use agent counterparts, *VideoAgent* adopts a minimalist tool-use pipeline and does not require expensive training, while offering comparable or better empirical results on challenging long-form video understanding benchmarks including EgoSchema, Ego4D NLQ, WorldQA and NExT-QA. Possible future direction includes more exploration of real-world applications in robotics, manufacturing, and augmented reality.

## Acknowledgements

We thank the anonymous reviewers for their constructive suggestions. Their insights have greatly improved the quality and clarity of our work. This work was partly supported by the National Science and Technology Major Project (2022ZD0114900).

## References

1. 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. *Advances in Neural Information Processing Systems (NeurIPS)* (2022)
2. Gao, Z., Du, Y., Zhang, X., Ma, X., Han, W., Zhu, S.C., Li, Q.: Clova: A closed-loop visual assistant with tool usage and update. *Conference on Computer Vision and Pattern Recognition (CVPR)* (2023)
3. Gong, R., Huang, Q., Ma, X., Vo, H., Durante, Z., Noda, Y., Zheng, Z., Zhu, S.C., Terzopoulos, D., Fei-Fei, L., et al.: Mindagent: Emergent gaming interaction. *arXiv preprint arXiv:2309.09971* (2023)
4. 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: *Conference on Computer Vision and Pattern Recognition (CVPR)* (2022)
5. Gupta, T., Kembhavi, A.: Visual programming: Compositional visual reasoning without training. In: *Conference on Computer Vision and Pattern Recognition (CVPR)* (2023)
6. Han, T., Xie, W., Zisserman, A.: Temporal alignment networks for long-term video. In: *Conference on Computer Vision and Pattern Recognition (CVPR)* (2022)
7. Hou, Z., Ji, L., Gao, D., Zhong, W., Yan, K., Li, C., Chan, W.K., Ngo, C.W., Duan, N., Shou, M.Z.: Groundnlq@ ego4d natural language queries challenge 2023. *arXiv preprint arXiv:2306.15255* (2023)
8. Jia, B., Chen, Y., Huang, S., Zhu, Y., Zhu, S.c.: Lemma: A multi-view dataset for learning multi-agent multi-task activities. In: *European Conference on Computer Vision (ECCV)* (2020)
9. Jia, B., Lei, T., Zhu, S.C., Huang, S.: Egotaskqa: Understanding human tasks in egocentric videos. *Advances in Neural Information Processing Systems (NeurIPS)* (2022)
10. Korbar, B., Xian, Y., Tonioni, A., Zisserman, A., Tombari, F.: Text-conditioned resampler for long form video understanding. *arXiv preprint arXiv:2312.11897* (2023)
11. Li, J., Li, D., Savarese, S., Hoi, S.: Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. In: *International Conference on Machine Learning (ICML)* (2023)
12. Lin, B., Zhu, B., Ye, Y., Ning, M., Jin, P., Yuan, L.: Video-llava: Learning unified visual representation by alignment before projection. *arXiv preprint arXiv:2311.10122* (2023)
13. Liu, H., Li, C., Wu, Q., Lee, Y.J.: Visual instruction tuning. *Advances in Neural Information Processing Systems (NeurIPS)* (2024)

14. Maaz, M., Rasheed, H., Khan, S., Khan, F.S.: Video-chatgpt: Towards detailed video understanding via large vision and language models. arXiv preprint arXiv:2306.05424 (2023)
15. Mangalam, K., Akshulakov, R., Malik, J.: Egoschema: A diagnostic benchmark for very long-form video language understanding. *Advances in Neural Information Processing Systems (NeurIPS)* (2024)
16. Miech, A., Zhukov, D., Alayrac, J.B., Tapaswi, M., Laptev, I., Sivic, J.: Howto100m: Learning a text-video embedding by watching hundred million narrated video clips. In: *International Conference on Computer Vision (ICCV)* (2019)
17. OpenAI: Gpt-4 technical report. arXiv preprint arXiv:2303.08774 (2023)
18. Oquab, M., Darcet, T., Moutakanni, T., Vo, H., Szafraniec, M., Khalidov, V., Fernandez, P., Haziza, D., Massa, F., El-Nouby, A., et al.: Dinov2: Learning robust visual features without supervision. arXiv preprint arXiv:2304.07193 (2023)
19. Radford, A., Kim, J.W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., et al.: Learning transferable visual models from natural language supervision. In: *International Conference on Machine Learning (ICML)* (2021)
20. Schick, T., Dwivedi-Yu, J., Dessì, R., Raileanu, R., Lomeli, M., Hambro, E., Zettlemoyer, L., Cancedda, N., Scialom, T.: Toolformer: Language models can teach themselves to use tools. *Advances in Neural Information Processing Systems (NeurIPS)* (2024)
21. Shafiq, N.M.M., Paxton, C., Pinto, L., Chintala, S., Szlam, A.: Clip-fields: Weakly supervised semantic fields for robotic memory. arXiv preprint arXiv:2210.05663 (2022)
22. Song, E., Chai, W., Wang, G., Zhang, Y., Zhou, H., Wu, F., Chi, H., Guo, X., Ye, T., Zhang, Y., et al.: Moviechat: From dense token to sparse memory for long video understanding. In: *Conference on Computer Vision and Pattern Recognition (CVPR)* (2024)
23. Surís, D., Menon, S., Vondrick, C.: Vipergpt: Visual inference via python execution for reasoning. In: *International Conference on Computer Vision (ICCV)* (2023)
24. Tapaswi, M., Zhu, Y., Stiefelhagen, R., Torralba, A., Urtasun, R., Fidler, S.: Movieqa: Understanding stories in movies through question-answering. In: *Conference on Computer Vision and Pattern Recognition (CVPR)* (2016)
25. Tay, Y., Dehghani, M., Abnar, S., Shen, Y., Bahri, D., Pham, P., Rao, J., Yang, L., Ruder, S., Metzler, D.: Long range arena: A benchmark for efficient transformers. arXiv preprint arXiv:2011.04006 (2020)
26. Team, G., Anil, R., Borgeaud, S., Wu, Y., Alayrac, J.B., Yu, J., Soricut, R., Schalkwyk, J., Dai, A.M., Hauth, A., et al.: Gemini: a family of highly capable multimodal models. arXiv preprint arXiv:2312.11805 (2023)
27. Tong, S., Liu, Z., Zhai, Y., Ma, Y., LeCun, Y., Xie, S.: Eyes wide shut? exploring the visual shortcomings of multimodal llms. In: *Conference on Computer Vision and Pattern Recognition (CVPR)* (2024)
28. Wang, Y., He, Y., Li, Y., Li, K., Yu, J., Ma, X., Li, X., Chen, G., Chen, X., Wang, Y., et al.: Internvid: A large-scale video-text dataset for multimodal understanding and generation. arXiv preprint arXiv:2307.06942 (2023)
29. Wang, Y., Li, K., Li, Y., He, Y., Huang, B., Zhao, Z., Zhang, H., Xu, J., Liu, Y., Wang, Z., et al.: Internvideo: General video foundation models via generative and discriminative learning. arXiv preprint arXiv:2212.03191 (2022)
30. Wang, Y., Yang, Y., Ren, M.: Lifelongmemory: Leveraging llms for answering queries in egocentric videos. arXiv preprint arXiv:2312.05269 (2023)



31. Wang, Y., Wang, Y., Wu, P., Liang, J., Zhao, D., Zheng, Z.: Lstp: Language-guided spatial-temporal prompt learning for long-form video-text understanding. arXiv preprint arXiv:2402.16050 (2024)
32. Wiles, O., Carreira, J., Barr, I., Zisserman, A., Malinowski, M.: Compressed vision for efficient video understanding. In: Asian Conference on Computer Vision (ACCV) (2022)
33. Wu, C.Y., Krahenbuhl, P.: Towards long-form video understanding. In: Conference on Computer Vision and Pattern Recognition (CVPR) (2021)
34. Wu, C., Yin, S., Qi, W., Wang, X., Tang, Z., Duan, N.: Visual chatgpt: Talking, drawing and editing with visual foundation models. arXiv preprint arXiv:2303.04671 (2023)
35. Xiao, J., Shang, X., Yao, A., Chua, T.S.: Next-qa: Next phase of question-answering to explaining temporal actions. In: Conference on Computer Vision and Pattern Recognition (CVPR) (2021)
36. Yang, A., Miech, A., Sivic, J., Laptev, I., Schmid, C.: Zero-shot video question answering via frozen bidirectional language models. Advances in Neural Information Processing Systems (NeurIPS) (2022)
37. Yang, Z., Chen, G., Li, X., Wang, W., Yang, Y.: Doraemongpt: Toward understanding dynamic scenes with large language models (exemplified as a video agent). In: International Conference on Machine Learning (ICML) (2024)
38. Ye, Q., Xu, H., Xu, G., Ye, J., Yan, M., Zhou, Y., Wang, J., Hu, A., Shi, P., Shi, Y., et al.: mplug-owl: Modularization empowers large language models with multimodality. arXiv preprint arXiv:2304.14178 (2023)
39. Yu, S., Cho, J., Yadav, P., Bansal, M.: Self-chained image-language model for video localization and question answering. Advances in Neural Information Processing Systems (NeurIPS) (2024)
40. Zhang, C., Lu, T., Islam, M.M., Wang, Z., Yu, S., Bansal, M., Bertasius, G.: A simple llm framework for long-range video question-answering. arXiv preprint arXiv:2312.17235 (2023)
41. Zhang, C., Lu, T., Islam, M.M., Wang, Z., Yu, S., Bansal, M., Bertasius, G.: A simple llm framework for long-range video question-answering. arXiv preprint arXiv:2312.17235 (2023)
42. Zhang, H., Li, X., Bing, L.: Video-llama: An instruction-tuned audio-visual language model for video understanding. arXiv preprint arXiv:2306.02858 (2023)
43. Zhang, H., Sun, A., Jing, W., Zhou, J.T.: Span-based localizing network for natural language video localization. arXiv preprint arXiv:2004.13931 (2020)
44. Zhang, S., Peng, H., Fu, J., Luo, J.: Learning 2d temporal adjacent networks for moment localization with natural language. In: AAAI Conference on Artificial Intelligence (AAAI) (2020)
45. Zhang, Y., Sun, P., Jiang, Y., Yu, D., Weng, F., Yuan, Z., Luo, P., Liu, W., Wang, X.: Bytetrack: Multi-object tracking by associating every detection box. In: European Conference on Computer Vision (ECCV) (2022)
46. Zhang, Y., Zhang, K., Li, B., Pu, F., Setiadharm, C.A., Yang, J., Liu, Z.: Worldqa: Multimodal world knowledge in videos through long-chain reasoning. arXiv preprint arXiv:2405.03272 (2024)
47. Zhao, H., Cai, Z., Si, S., Ma, X., An, K., Chen, L., Liu, Z., Wang, S., Han, W., Chang, B.: Mmicl: Empowering vision-language model with multi-modal in-context learning. arXiv preprint arXiv:2309.07915 (2023)
48. Zhao, Y., Lv, W., Xu, S., Wei, J., Wang, G., Dang, Q., Liu, Y., Chen, J.: Detrs beat yolos on real-time object detection. In: Conference on Computer Vision and Pattern Recognition (CVPR) (2024)

49. Zhao, Y., Misra, I., Krähenbühl, P., Girdhar, R.: Learning video representations from large language models. In: Conference on Computer Vision and Pattern Recognition (CVPR) (2023)
50. Zhu, C., Xiao, F., Alvarado, A., Babaei, Y., Hu, J., El-Mohri, H., Culatana, S., Sumbaly, R., Yan, Z.: Egoobjects: A large-scale egocentric dataset for fine-grained object understanding. In: International Conference on Computer Vision (ICCV) (2023)