Video Question Answering with Object- and Action-level Insights

Zitian Tang, Shangyang Min, Yiyang Nan, Zhangyi Shen, Ziyu Wang, Siyang Zhang
Brown University

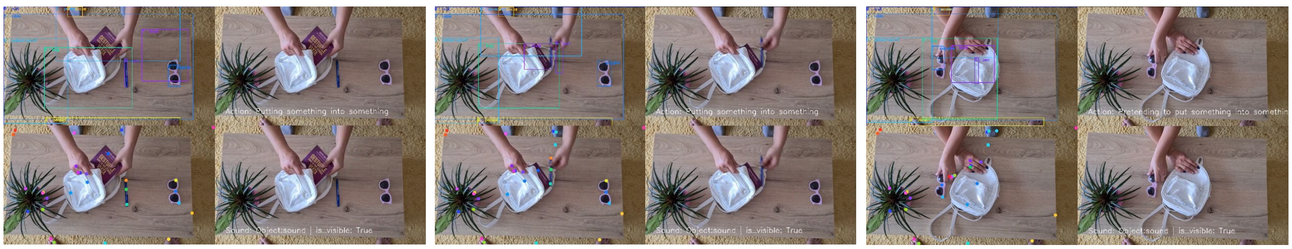


Figure 1. **Perception Test video sample.** Annotations of object tracking (top left), point tracking (bottom left), action localization (top right), and sound localization (bottom right) are provided in the Perception Test benchmark.

Abstract

Video Question Answering (VideoQA) traditionally relies on latent video representations that often fail to capture crucial details necessary for answering complex questions. In this paper, we propose a novel approach that enhances the capabilities of Vision-Language Models (VLMs) by transcending the limitations of traditional latent video representations. Our method utilizes multiple external vision models, including object tracking, point tracking, action localisation, and sound localisation, to transform video information into detailed natural language descriptions. This enhancement allows for the capture of essential details, thereby improving the VLMs' ability to interpret video content more effectively. We demonstrate that our approach significantly improves performance on the Perception Test benchmark in zero-shot scenarios. The results suggest that leveraging detailed textual descriptions offers a promising direction for enhancing video understanding without extensive prior training.

1. Introduction

Video Question Answering (VideoQA) aims to answer questions based on a given video. Solving VideoQA requires models to understand video content and textual questions, and reason over the perceived information. With the exceptional reasoning capability of LLMs, recent video-language models (VLMs) [11, 22] feed latent video representations into Large Language Models (LLMs) to tackle this task. Moreover, some other works [18, 20, 23] deliver not only the video representations but also the captions of video segments to LLMs, making better use of its reasoning ability over text.

However, the video representations used in these methods only summarize the video content at high-level. While video captioning cannot describe every detail in a video, there is also evidence suggesting that some visual encoders like CLIP [15] may lose fine-grained information [17]. In some video questions, accurately capturing the object movement might be required. For example, in a cups-game, a small object is hidden under one of several cups turned upside down. After shuffling the cups, the question asks under which cup the object is. If we can track all the cups in the video, the question can be answered easily based on their trajectories. However, none of the video representations used by current VLMs can accurately capture this detailed object movement.

To address this issue, we propose to use external vision models to explicitly extract information from a video and then aggregate these information into LLMs to answer the questions. In this work, we focus on a multi-choice VideoQA benchmark, Perception Test [13]. It includes questions requiring strong perception capability of the model. Besides, as shown in Figure 1, it provides annotations of object tracking, point tracking, action localization, and sound localization for the videos. All these types of object- and action-level visual information are essential for video understanding. For example, point tracking can tell us whether an object is rotating; Action localisation includes the temporal relations between actions; Sound localisation may reveal whether two objects have touched each other. In this work, we use off-the-shelf models to extract these types of information and then format them in natural language. Finally, they're passed on to LLMs to answer questions about the video. Our method utilizes both the perception ability of vision models and reasoning ability of LLMs.

Furthermore, the availability of manual annotations of such information allows oracle experiments. By replacing the vision models with annotations, we can investigate if the LLMs can answer the questions given nearly perfect visual perceptions.

In the experiments, we show that our method is competitive on Perception Test Benchmark. While employing closed-source LLMs achieves exceptional performance, fine-tuning open-source LLMs can improve the capability in our method. The oracle experiments show that the perception capability of the external vision models is crucial for our method. Finally, we analyze the strengths and weaknesses of our proposed methods.

2. Related Work

2.1. Video Question Answering Models

Recent VideoQA models are developed in two mainstream ways – video-language joint training and visually-augmented LLMs. The video-language joint-training. Video-language joint training models [6, 19] perform masked token modeling on visual and language tokens jointly, and use contrastive learning to align the video and language feature spaces. Visually-augmented LLMs [8, 9, 11, 22, 25] learn a feature adapter to project video features into the language token embedding space. They leverage the strong reasoning capability of LLMs and fine-tune them to enable video-language reasoning. Moreover, some methods employ image and video captioning models to describe the video content and feed them to LLMs to compensate the visual inputs [18, 20, 23]. They demonstrate that language is an effective video representation for video-language models besides latent video embeddings.

2.2. Object-level Inputs for VLMs

There are several works integrating object-level information into language models [3, 14, 16, 27]. STOA-VLP [27] fuses latent representations of detected objects into video representations in video-language pre-learning. BiLL-VTG [14] employs LLMs to reason over scene graphs extracted from video frames. LLM-Driver [3] transforms vector representation (location, velocity, etc.) of a driving scenario into natural language and uses LLMs to make decisions. LanguageRefer [16] feeds the bounding boxes of 3D objects to a language model to perform 3D visual grounding.

Recently, some works enable the image VLMs to receive and generate object bounding boxes [2, 12]. They show that bounding boxes help VLMs to localize and recognize object, and hence enhance their image reasoning ability. While Shikra [2] formats bounding boxes in textual coordinates, PerceptionGPT [12] verifies that encoding them into latent representations improves their comprehensibility to VLMs. Different from them, we are the first to utilize LLMs to reason over object tracking and action localization information on videos.

3. Method

3.1. Overview

In this work, we solve multi-choice Video Question Answering by utilizing the perception ability of vision models and reasoning ability of LLMs. Given a video V, we first employ a few external vision models to extract visual information from the video. Specifically, we use a object tracking model f_{OT}, a point tracking model f_{PT}, an action localisation model f_{AL}, and a sound localisation model f_{SL}. After that, we describe the question Q, options $C_{1:k}$, and all the extracted information above to form a prompt in natural language. Finally, the prompt is delivered to the LLM and the LLM gives a reponse indicating the predicted option. Formally, the proposed pipeline is

$$\begin{aligned} y = & \text{LLM}(\\ & \text{Prompt}(Q, C_{1:k}, f_{OT}(V), f_{PT}(V), f_{AL}(V), f_{SL}(V)) \\ &), \end{aligned}$$

which is also illustrated in Figure 2.

In the following, we will describe how we develop vision models, prompt the LLMs, and optionally fine-tune the LLMs.

3.2. Vision Models

In this section, we introduce the external vision models used in our framework.

Object tracking. The baseline model of object tracking in Perception Test [13] is SiamFC [1]. However, its original version from 8 years ago has outdated code, and the new version in PyTorch requires training with a dataset of 73 GB. After viewing the data, we found that it has less relevance with our training purpose. Hence, we employ a state-of-art model of object detection: YOLOv8 developed by Ultralytics, which enables pre-trained development and custom training.

First, we enable the pre-trained *YOLOv8l(large)* on the sample set to verify its correctness. We set *vid_stride=30*, *iou=0.1*, and *conf=0.3*. The model perform generally well in tracking and detection. However, since the pre-trained model has never learned from the video in Perception Test, some of the objects or shapes do not exist in its dictionary list, resulting in some missing or mis-detection.

Then, we annotate over 1000 frame samples from Perception Test training set with the online Yolo Robflow platform and fine-tune YOLOv8. We only use the object categories that are top-200 frequent in the training set of Perception test. We set *patience=100* and *imgsz=640* while leaving other

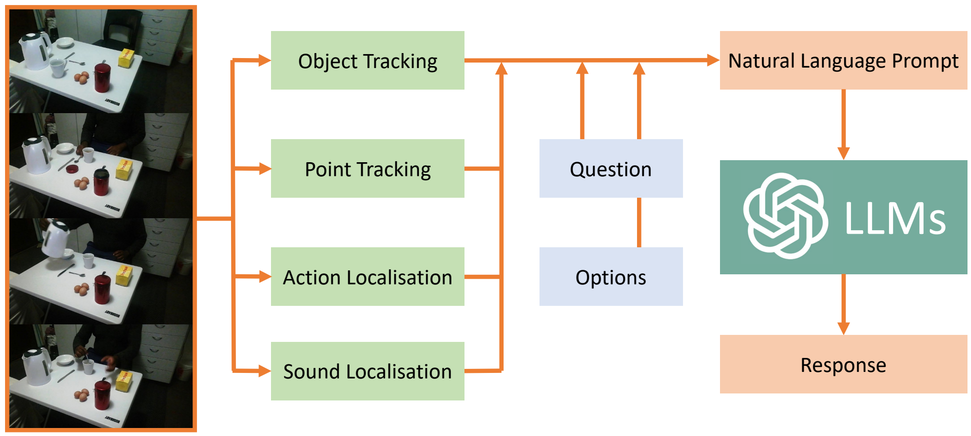


Figure 2. **Overview of our method.** To answer a question about a given video, we first use external vision models to extract object tracking, point tracking, action localisation, and sound localisation from the video. After that, we use a textual template to describe the perceived visual information in natural language. It it delivered to LLMs together with the question and options. The LLMs reason over the given information and answer the question.

hyper-parameters default. With the trained model, we perform object tracking on the Perception Test videos. Some samples are visualized in Figure 3. The detection accuracy and boxing robustness are improved by fine-tuning, but there are still missing or mis-detection when some shapes are ambiguous and some colors are close. Besides, the appearance of some blurred objects and occlusions will affect the performance.

Point tracking. We use a pretrained TAPIR [5] as the point tracking model. It grants the ability to tracks any queried points on any physical surface throughout a video. TAPIR achieves state-of-the-art performances on TAP-vid-Knetics and TAP-vid-Davis [4] benchmarks. And it has in general good performance on the tracking-any-point test. It can both locate the coordinates of point queries and indicate their visibility in each video frame. Such information will provide useful prior for the language model we use for VideoQA task.

TAPIR starts with a global comparison between the query point features in a reference frame and those in every other frame to derive an initial tracking prediction with estimated uncertainty. Then, the model extracts features from a local spatial neighborhood around each point and compare them to the query feature in a higher resolution, post-processing the similarities with a temporal depthwise-convolutional network to get an updated prediction. The updated position is then fed back into the next iteration of refinement. The refinement can be done in an iterative manner so as to enhance the performance.

In our framework, we provide a set of point queries for each video in the first frame. TAPIR tracks all the queried points throughout the video. Thus, we can know the location of any queried point in every frame and its visibility. A qualitative visualization is in Figure 4. The visibility of the points provide essential information in VideoQA. For example, if there is a video in which a person moves glasses on the table, and there are occlusions among the glasses, it is important to know whether a glass is visible in any frame, especially when we ask questions about the blocked object in the last frame of the video.

Figure 3. **Visualization of object tracking with fine-tuned YOLOv8.** The model tracks the objects throughout the video and gives the bounding boxes of them in every frame.

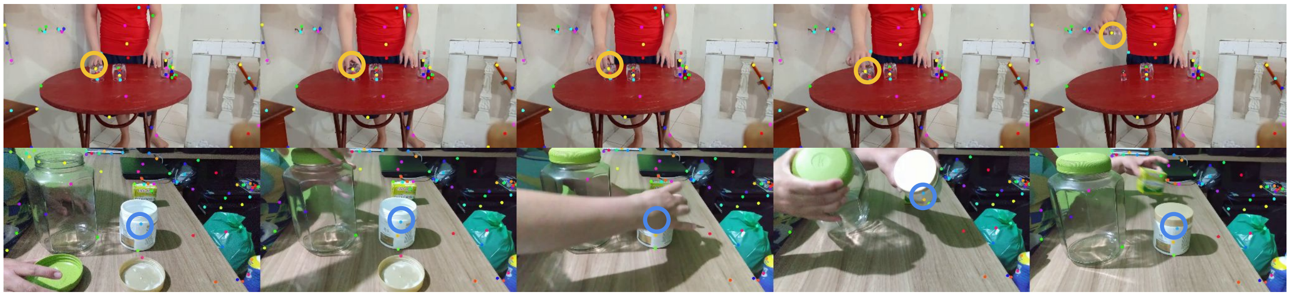


Figure 4. **Visualization of point tracking with TAPIR.** We mannually select key points from the first frame, and then the model will predict their corresponding coordinates and visibility throughout all following video frames. **First row:** The yellow point within the yellow circle; the point is visible during the entire video. **Second row:** The blue point within the blue circle; it is not visible in the third frame because it is blocked by the human hand.

Action localisation. Action localisation involves detecting and tracking specific actions or activities within a video. This task requires a model to possess both spatial and temporal understanding. We employ the current state-of-the-art framework, ActionFormer [24], which is the action localisation baseline of Perception Test. ActionFormer utilizes a self-attention transformer-based architecture to analyze videos, outputting bounding boxes and class labels for the actions it identifies.

The primary components of ActionFormer include a video encoder, a transformer encoder, and an action decoder. The video encoder employs a 3D convolutional neural network to extract spatial and temporal features, while the transformer encoder captures long-range dependencies within the video. The action decoder predicts the bounding boxes and class labels for the detected actions. Unlike many previous methods that use various modalities and ensembles of multiple architectures, we use only an ActionFormer module to gather all the necessary contextual information. This streamlined design allows the model to infer the relationships between different actions in a video, making it particularly suitable for the Perception Test [13].

We first train ActionFormer on the Perception Test training data and then get the validation set localization information by running inference with the trained model.

Sound localisation. Similar to action localisation, we use ActionTransformer for sound localisation. Sound localisation is to estimate the temporal intervals of sounds in a video. Although the ActionTransformer model was originally designed for recognizing and localizing human actions in video, it can also be adapted to sound localization. The input audio is first converted into a time-frequency representation, spectrogram, to preserve both spatial and temporal information about the sound sources. This spectrogram is then treated as a unique kind of video that humans cannot readily interpret. Same as action localisation, we train ActionFormer on the Perception Test training set and conduct inference on the validation set.

3.3. Prompt Design for LLMs

In designing prompts for LLMs, it is crucial to use clear and concise natural language to ensure the model understands the information that we provide. Therefore, as shown in Figure 5, we use a natural language template to describe the perception results of object tracking, point tracking, action localisation, and sound localisation models. We provide the object and action names, tracking bounding boxes, and start and end timestamps of localisation. For object tracking and point tracking, we down-sample the frame rates to 1 FPS. Moveover, we employ a two-stage prompt to improve the model accuracy. The first stage enables chain of thoughts [21], where the model processes the information and give a response with a logical sequence of reasoning. In the second stage, the model is limited to a simplified its output to only a single choice among ’A,’ ’B,’ and ’C’. This structured prompt design not only guides the LLMs through complex reasoning but also prevent them from failing to choose a valid option.

3.4. Fine-tuning LLMs

Closed-source LLMs such as GPT-4 has outstanding reasoning capability. However, they can only be employed in a zero-shot manner. As our inputs to the LLMs are not common in the pretraining data, their potential in our task might not been fully realized. With this in mind, we fine-tune open-source LLMs such as Vicuna-1.5 [26] on the training set of Perception Test. Specifically, we frist format the manual annotations in the training set as the way described in Sectoin 3.3, but disable the chain of thoughts and prompt the model to directly give a letter answer. Then we fine-tuning the LLMs using the next token prediction task, while the training loss is only computed on the answer tokens. We hope fine-tuning LLMs can help them understand the object-

```
USER:
In the following, I will describe the content of a video, including object tracking, point tracking, action
localisation, and sound localisation. I will give you a three-choice question and please answer the question
based on the video content.
Question: How many containers did the person try to cover?
Options:
A.6
B.3
C.4
Video metadata: The video has in total 432 frames with a frame rate of 30 FPS. Its resolution is 1080 by
1920.
Object tracking: There are in total 23 objects in the video. For each object, I will give its bounding boxes
at certain frames. The bounding boxes are in format [x1, y1, x2, y2] with each value between 0 and 1. [x1,
x2] is the boundary along the video width, from the left to the right. [y1, y2] is the boundary along the
video height, from the top to the bottom.
Object 0 is person. Its bounding boxes are - frame 0: [0.0, 0.0, 0.62, 0.8]; frame 30: [0.0, 0.0, 0.62,
0.83]; frame 60: [0.01, 0.0, 0.62, 0.79]; frame 90: [0.0, 0.0, 0.62, 0.78]; ... frame 420: [0.0, 0.0, 0.61,
0.79];
... ...
Object 22 is rice-cooker. Its bounding boxes are - frame 0: [0.04, 0.46, 0.13, 0.56]; frame 30: [0.05, 0.43,
0.12, 0.56]; frame 60: [0.05, 0.46, 0.12, 0.57]; frame 90: [0.05, 0.46, 0.13, 0.57]; ... frame 420: [0.04,
0.46, 0.12, 0.57];
Point tracking: There are in total 69 tracked points in the video. For each point, I will give its positions
at certain frames. The positions are in format [x, y] with each value between 0 and 1. x is along the video
width from the left to the right, while y is along the video height from the top to the bottom.
Point 0 is person-top of object 0 person. Its positions are - frame 0: [0.04, 0.49]; frame 30: [0.03, 0.49];
frame 60: [0.04, 0.49]; frame 90: [0.04, 0.49]; ... frame 420: [0.08, 0.52];
... ...
Point 68 is rice-cooker-middle of object 22 rice-cooker. Its positions are - frame 0: [0.52, 0.1]; frame 56:
[0.52, 0.1]; frame 86: [0.52, 0.1]; ... frame 401: [0.53, 0.1];
action localisation: There are in total 3 actions in the video. For each action, I will give its start and
end frame indices.
Action 0 is Covering something with something. It involves object 1 bottle and object 9 cap. It lasts from
frame 27 to frame 111;
Action 1 is Pretending to cover something. It involves object 3 box and object 10 cap. It lasts from frame
139 to frame 224;
Action 2 is Covering something with something. It involves object 2 bottle and object 8 router. It lasts from
frame 259 to frame 352;
Sound localisation: There are in total 6 sounds in the video. For each sound, I will give its start and end
frame indices.
Sound 0 is Object:sound. It involves object 9 cap and object 13 table. It lasts from frame 25 to frame 32;
... ...
Sound 5 is Interaction:hitting. It involves object 8 router and object 2 bottle. It lasts from frame 317 to
frame 332;
LLM: [Response]
USER: [Response]. The text above is an answer to a three-choice question with options A, B, and C. Which of A,
B, and C is the final answer? Please output only one letter among A, B, and C.
LLM: [Answer]
```

Figure 5. **An example of our natural language prompt.** We use a template to describe the four types of visual information respectively, and employ a two-stage prompt to enable chain of thoughts.

Method	Input	Adaption	AT	CM	CT	LO	OP	OR	OG	ST	Overall
Baselines											
Random	-	-	33.3	33.3	33.3	33.3	33.3	33.3	33.3	33.3	33.3
SeViLA	Video	Zero-shot	40.0	40.0	0.0	40.0	60.0	80.0	60.0	40.0	45.0
SeViLA	Video	Fine-tuned	60.0	40.0	80.0	100	40.0	100	80.0	40.0	67.5
Ours											
Vicuna-1.5	Vision Oracle	Zero-shot	100	20.0	100	20.0	20.0	0.0	20.0	60.0	42.5
Vicuna-1.5	Vision Oracle	Fine-tuned	20.0	60.0	60.0	40.0	60.0	60.0	80.0	20.0	50.0
GPT-4	Vision Oracle	Zero-shot	60.0	100	80.0	80.0	60.0	80.0	20.0	60.0	67.5
GPT-4	Vision Models	Zero-shot	60.0	60.0	60.0	80.0	40.0	80.0	0.0	80.0	57.5

Table 1. **Evaluation results of the models.** Our method outperforms the Perception Test baseline SeViLA in zero-shot setting. With vision oracle (annotations), GPT-4 can achieve the same accuracy as fine-tuned SeViLA. Moreover, fine-tuning can improve the performance of the LLMs with our inputs on the task.

and action-centric information and improve the model performance.

Implementation details. We fine-tune Vicuna-1.5 using LoRA [7] with a rank of 128. The model is trained for 1 epoch with a batch size of 16. We linearly warm up the learning rate to 2×10^{-4} in the first 3% steps and then employ cosine annealing. AdamW [10] is used as the optimizer. As some of the prompts are extremely long, we truncate all the model inputs to at most 2,048 tokens.

4. Experiments

4.1. Evaluation Settings

Test set. To reduce the time cost of model evaluation, we manually select 40 video-question pairs from the validation set of Perception Test. These questions fall into eight categories – action (AT), camera movement (CM), counting (CT), letter order (LO), object position (OP), object recognition (OR), occlusion game (OG), and stability (ST). Each category includes five questions. This enables us to measure the model performance for different question types respectively.

Metric. The performance of the model is indicated by accuracy. As there are three options in each question, a random algorithm on the task can achieve an accuracy of 33.3%.

Models. For comparison, we use a VideoQA model SeViLA [22] as a baseline, which is the official baseline of Perception Test. We evaluate its performance in both zero-shot and fine-tuning settings. As for our method, we explore two LLMs, open-source model Vicuna-1.5 [26] and closed-source model GPT-4. For Vicuna-1.5, we use the oracle visual information (annotations) as model input, and test it in both zero-shot and fine-tuning scenarios. For GPT-4, we evaluate it with either oracle visual information and that extracted by our vision models described in Section 3.2. The checkpoint of GPT-4 is *gpt-4-turbo-2024-04-09* and the temperature is set to be zero.

4.2. Main Results

We show the evaluation results in Table 1. The zero-shot SeViLA baseline using video input achieves an overall accuracy of 45.0%, while fine-tuning enhances its performance to 67.5%. Vicuna-1.5, with zero-shot adaptation and Vision Oracle as input, achieves comparable results with the zero-shot SeViLA, while it improves to 50% accuracy and outperforms zero-shot baselines on most metrics after fine-tuning, especially in the subareas of object recognition and occlusion game. This verifies that fine-tuning can indeed improve the understanding of LLMs on out language-formatted inputs. As the reasoning capability of GPT-4 is much stronger than Vicuna-1.5, it achieves an accuracy of 67.5% in the zero-shot setting using oracle vision information and 57.5% when employing our external vision models. This shows that LLMs can tackle VideoQA given object- and action-level information without direct access to the video. We qualitatively investigate the performance decrease from vision oracle to vision models in Section 4.4.

4.3. Effect of Each Information Type

In our method, four types of information are included – object tracking (OT), ponit tracking (PT), action localisation (AL), and sound localisation (SL). Different types of information contribute to the model performance from different aspects. To explore the effect of each information type, we test our method in cases where only one type of information is delivered to the LLMs. With GPT-4 as the LLM and our vision model results as the input, the results are in Table 2. The integration of different information types has varying effects on the performance of the GPT-4 model across different sub-areas, with the combination of all information types (OT, PT, AL, SL) resulting in the highest overall accuracy of 57.5%. When utilizing individual types of information, using

Method	OT	PT	AL	SL	AT	CM	CT	LO	OP	OR	OG	ST	Overall
GPT-4	✓				40.0	80.0	60.0	60.0	20.0	80.0	20.0	60.0	52.5
GPT-4		✓			60.0	80.0	20.0	60.0	60.0	100	20.0	0.0	50.0
GPT-4			✓		40.0	0.0	20.0	80.0	40.0	40.0	20.0	60.0	37.5
GPT-4				✓	40.0	0.0	60.0	40.0	40.0	40.0	20.0	80.0	40.0
GPT-4	✓	✓	✓	✓	60.0	60.0	60.0	80.0	40.0	80.0	0.0	80.0	57.5

Table 2. **Effect of infomation types.** The most informative type of information is object tracking. And the model performs the best when all the types are integrated.

Method	Prompt	Chain of Thoughts	AT	CM	CT	LO	OP	OR	OG	ST	Overall
GPT-4	JSON	✓	80.0	60.0	20.0	80.0	40.0	40.0	40.0	20.0	47.5
GPT-4	Language		60.0	40.0	20.0	80.0	60.0	60.0	20.0	80.0	52.5
GPT-4	Language	✓	60.0	60.0	60.0	80.0	40.0	80.0	0.0	80.0	57.5

Table 3. **Ablation studies on GPT-4 prompts.** Both natural language prompt and chain of thoughts improve the performance of GPT-4.

object tracking information alone yields the highest accuracy of 52.5%, whereas using sound localization alone results in an accuracy of 37.5%, marginally better than random guessing. These outcomes align with our expectations, given that the majority of questions pertain to objects rather than sounds. The presence of adversarial actions in the videos makes it more challenging for the model to provide accurate localization.

4.4. Qualitative Results

We show a few responses of GPT-4 with vision oracle and vision models in Figure 6, 7, and 8.

In Figure 6, a person is placing a coat and a pair of shoes on a chair to create a distraction while querying about the spatial relations of the items on the table. Both GPT-4 with oracle annotations and external vision models successfully provide the correct answer. They analyze the bounding box locations of various items, demonstrating their ability in detecting static objects and analyzing spatial relations.

However, in Figure 7, the video shows a sequence of actions involving placing objects such as books, pens, and laptops into a bag and then removing them. The question pertains to counting the interactions with the bag. While the response from the oracle model is accurate based on detailed annotations of different objects, our model misses one object, indicating potential issues with the external object detection and action localisation models.

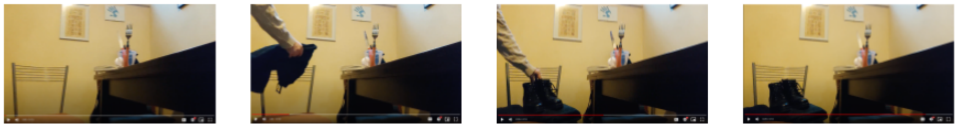
Lastly, in Figure 8, a person displays various items such as cups and pens with different colors. The question inquires about the order and colors of the displayed items. Both the oracle model and our model with external vision modules fail to answer correctly. Since the input lacks color information of the objects, the response from the oracle model involves an educated guess based on the object categories. Conversely, our model, confused by the object tracking information and lacking color information, admits not knowing the correct answer and resorts to random guessing.

4.5. Ablation Studies

In this section, we conduct ablation studies on how we prompt GPT-4 on our task. Visual information extracted by our vision models are used in the following experiments.
Prompt design. As GPT-4 is a language model, formatting the visual information in natural language should ease its understanding. To verify it, we evaluate our method with JSON prompt – delivering the visual information in a JSON format, which is the original format of the Perception Test annotations. As shown in Table 3, JSON prompt degrades the model performance by 10% compared to natural language prompt.
Chain of thoughts. As described in Section 3.3, we enable chain of thoughts by allowing GPT-4 to first freely answer the question and then summarize its response into a single letter. To measure the contribution of this technique, we evaluate our method by forcing GPT-4 to directly give the final answer without showing any reasoning process. As Table 3 suggests, chain of thoughts can indeed improve the QA accuracy of GPT-4 on our task.

5. Limitations and Future Works

While GPT-4 sometimes provides consistent answers when presented with clear textual descriptions, there are numerous instances where it delivers different outputs for the same input. These variations manifest as different reasoning pathways leading to the same answer, incorrect answers derived from similar logic paths, or correct answers following reevaluation of initially incorrect responses. This behavior underscores a critical limitation in the model's ability to process



Question: Which of the following statements describe the scene better?

A. There is a fork and a knife inside a cup on the chair.

B. There are two forks inside a cup on the table.

C. There is a fork and a knife inside a cup on the table.

Oracle: Based on the detailed description provided, the correct answer to the question "Which of the following statements describes the scene better?" is: C. There is a fork and a knife inside a cup on the table. This conclusion is drawn from the object tracking data which indicates the presence of a fork (Object 15) and a butter-knife (Object 16) with their bounding boxes located in a region that overlaps with the table (Object 11). The bounding boxes for the fork and knife are consistently within the coordinates that suggest they are on the table, not inside a cup or on a chair. Additionally, the description does not mention any cup explicitly, but the presence of these utensils on the table aligns with option C as the most accurate description of the scene.

Ours: Based on the detailed description of the video content provided, the correct answer to the question "Which of the following statements describes the scene better?" is: C. There is a fork and a knife inside a cup on the table. This conclusion is drawn from the object tracking data which indicates the presence of a fork (Object 5) and a knife (Object 7, referred to as a hairdryer but likely a mislabel based on context) with bounding boxes that overlap with the table (Object 0) and a cup (Object 20). The bounding boxes for these objects are all located in a manner that suggests they are on the table, not on a chair or with two forks as the other options suggest.,

Figure 6. **Video example of showing spatial relations and object recognition.** Both GPT-4 with vision oracle and its counterpart with external vision models answer this question correctly.

Question: How many objects were put in the backpack throughout the video?

A. 2

B. 6

C. 5

Oracle: Based on the detailed description of the video content and the actions listed, the objects put into the backpack (object 2, bag) throughout the video are: Object 4 (book) during Action 6, Object 5 (book) during Action 7, Object 7 (pen) during Action 8, Object 6 (pen) during Action 9, Object 8 (tshirt) during Action 10, Object 3 (laptop) during Action 11. This totals to 6 objects being put into the backpack. Therefore, the correct answer is: B. 6

Ours: Based on the detailed description provided, the question asks how many objects were put in the backpack throughout the video. The relevant information to answer this question can be found in the "Action localisation" section, where actions described as "Putting something into something" are listed. Here are the relevant actions: Action 0: Putting something into something. It lasts from frame 451 to frame 624. Action 2: Putting something into something. It lasts from frame 818 to frame 900. Action 3: Putting something into something. It lasts from frame 241 to frame 324. Action 5: Putting something into something. It lasts from frame 48 to frame 124. Action 6: Putting something into something. It lasts from frame 651 to frame 777. Counting these actions, there are 5 instances where an object is put into something, presumably the backpack. Therefore, the answer to the question is: C. 5

Figure 7. **Video example of object counting and containment.** GPT-4 answers the question correctly when oracle visual information is provided. However, it fails when the oracle is replaced with vision models.

and interpret input consistently. The instability of GPT-4 might make our evaluation inaccurate and influence the method decision.

Model-wise, the language-formatted object and tracking information might be too long for LLMs to understand. We may train a sequence model to encoder tracking bounding boxes and coordinates and fine-tune the LLMs to understand the encoded tracking information. We hope it can improve the model performance. Further, in our current method, the raw videos are not accessible for the LLMs, which may lose important visual details such as object colors. For future work, we may add vision inputs to the LLMs by training a feature adapter to transform latent visual features into language embedding space.

Data-wise, the Perception Test only contains a small amount of data. We may train on larger VideoQA datasets

Question: Which statement contains a reference to the third distinct object shown by the person in the video?

A. This cup has the colour of the side that moves second in a chess game.

B. This pen partially shares its colour with the person's tshirt.

C. The dominant colour of this pen is the same as the colour of the side that moves first in a chess game.

Oracle: Based on the video content and object tracking information provided, the third distinct object shown by the person in the video is object 3, which is a pen. The bounding boxes and action localization confirm that object 3 is involved in an action of "Showing something to the camera" from frame 191 to frame 320. Given the options provided: A. This cup has the colour of the side that moves second in a chess game. B. This pen partially shares its colour with the person's tshirt. C. The dominant colour of this pen is the same as the colour of the side that moves first in a chess game. The correct answer is: C. The dominant colour of this pen is the same as the colour of the side that moves first in a chess game. This statement refers to object 3, which is the third distinct object shown in the video, and it is a pen. The reference to the colour of the side that moves first in a chess game (typically white) suggests a description of the pen's colour.

Ours: Based on the object tracking information provided, the third distinct object shown by the person in the video is the "mobile-phone" (Object 3). The bounding boxes for this object are provided at various frames, indicating its presence and location in the video. Given the three options provided: A. This cup has the colour of the side that moves second in a chess game. B. This pen partially shares its colour with the person's tshirt. C. The dominant colour of this pen is the same as the colour of the side that moves first in a chess game. The correct answer is not directly related to the third object (mobile-phone) based on the descriptions given. However, if we consider the typical scenario of showing objects in a video, none of the options directly describe the mobile-phone. Therefore, it seems there might be a misunderstanding or misalignment in the question and the options provided. If we strictly follow the object tracking data, none of the options A, B, or C correctly describe the third object (mobile-phone). Hence, based on the information provided, none of the options correctly answer the question regarding the third distinct object shown in the video.

Figure 8. **Video example of sequencing and visual discrimination.** GPT-4 fails the question with either vision oracle or vision models.

or design custom QAs on other object tracking datasets. General object videos from anywhere will have more complicated scenarios and more unseen dynamics, which may enhance the model if we train on them.

Finally, we may develop or adapt the model for real-time VideoQA tasks, which involves optimizing the model architecture for faster processing speeds without a significant drop in accuracy, enabling its use in live scenarios.

6. Conclusion

Our approach to VideoQA leverages multiple vision models to transform video data into textual representations, aiming to enhance the reasoning capabilities of VLMs. According to our experiments, in zero-shot settings our approach outperforms the advanced SeViLA model, underscoring its effectiveness when no fine-tuning is applied. However, when competing against the fine-tuned SeViLA, our method shows a lower performance, indicating a space for improvement of our approach. Notably, our method excels in scenarios requiring detailed visual comprehension, such as object and action recognition, but shows variability in performance based on the type and complexity of the questions posed.

Despite the encouraging results, our approach encounters limitations, both in external vision models and GPT-4. Several avenues are worth exploring in the future. We can optimize the way to deliver the visual information to LLMs, such as encoding tracking bounding boxes and adding latent visual representations. Besides, expanding our training regimen to larger VideoQA datasets is expected to improve the reasoning capability of our model in more diverse and complicated scenarios.

References

[1] Luca Bertinetto, Jack Valmadre, Joao F Henriques, Andrea Vedaldi, and Philip HS Torr. Fully-convolutional siamese networks for object tracking. In *ECCV Workshops*, 2016. 2

[2] Keqin Chen, Zhao Zhang, Weili Zeng, Richong Zhang, Feng Zhu, and Rui Zhao. Shikra: Unleashing multimodal llm's referential dialogue magic. *arXiv preprint arXiv:2306.15195*, 2023. 2

[3] Long Chen, Oleg Sinavski, Jan Hünermann, Alice Karnsund, Andrew James Willmott, Danny Birch, Daniel Maund, and Jamie Shotton. Driving with llms: Fusing object-level vector modality for explainable autonomous driving. *arXiv preprint arXiv:2310.01957*, 2023. 2

[4] Carl Doersch, Ankush Gupta, Larisa Markeeva, Adria Recasens Continente, Kucas Smaira, Yusuf Aytar, Joao Carreira, Andrew Zisserman, and Yi Yang. Tap-vid: A benchmark for tracking any point in a video. In *NeurIPS Datasets Track*, 2022. 3

[5] Carl Doersch, Yi Yang, Mel Vecerik, Dilara Gokay, Ankush Gupta, Yusuf Aytar, Joao Carreira, and Andrew Zisserman. Tapir: Tracking any point with per-frame initialization and temporal refinement. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10061–10072, 2023. 3

[6] Tsu-Jui Fu, Linjie Li, Zhe Gan, Kevin Lin, William Yang Wang, Lijuan Wang, and Zicheng Liu. VIOLET: End-to-End Video-Language Transformers with Masked Visual-token Modeling. In *arXiv:2111.1268*, 2021. 2

[7] Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2022. 6

[8] Kunchang Li, Yali Wang, Yinan He, Yizhuo Li, Yi Wang, Yi Liu, Zun Wang, Jilan Xu, Guo Chen, Ping Luo, Limin Wang, and Yu Qiao. Mvbench: A comprehensive multi-modal video understanding benchmark, 2023. 2

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

[10] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2019. 6

[11] Muhammad Maaz, Hanoona Rasheed, Salman Khan, and Fahad Shahbaz Khan. Video-chatgpt: Towards detailed video understanding via large vision and language models. *arXiv:2306.05424*, 2023. 1, 2

[12] Renjie Pi, Lewei Yao, Jiahui Gao, Jipeng Zhang, and Tong Zhang. Perceptiongpt: Effectively fusing visual perception into llm, 2023. 2

[13] Viorica Pătrăucean, Lucas Smaira, Ankush Gupta, Adrià Recasens Continente, Larisa Markeeva, Dylan Banarse, Skanda Koppula, Joseph Heyward, Mateusz Malinowski, Yi Yang, Carl Doersch, Tatiana Matejovicova, Yury Sulsky, Antoine Miech, Alex Frechette, Hanna Klimczak, Raphael Koster, Junlin Zhang, Stephanie Winkler, Yusuf Aytar, Simon Osindero, Dima Damen, Andrew Zisserman, and João Carreira. Perception test: A diagnostic benchmark for multimodal video models. In *Advances in Neural Information Processing Systems*, 2023. 1, 2, 4

[14] Ji Qi, Kaixuan Ji, Jifan Yu, Duokang Wang, Bin Xu, Lei Hou, and Juanzi Li. Bill-vtg: Bridging large language models and lightweight visual tools for video-based texts generation. *arXiv preprint arXiv:2310.10586*, 2023. 2

[15] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. PMLR, 2021. 1

[16] Junha Roh, Karthik Desingh, Ali Farhadi, and Dieter Fox. Languagerefer: Spatial-language model for 3d visual grounding. In *Conference on Robot Learning*, pages 1046–1056. PMLR, 2022. 2

[17] Shengbang Tong, Zhuang Liu, Yuexiang Zhai, Yi Ma, Yann LeCun, and Saining Xie. Eyes wide shut? exploring the visual shortcomings of multimodal llms. *arXiv preprint arXiv:2401.06209*, 2024. 1

[18] Shijie Wang, Qi Zhao, Minh Quan Do, Nakul Agarwal, Kwonjoon Lee, and Chen Sun. Vamos: Versatile action models for video understanding, 2023. 1, 2

[19] Yi Wang, Kunchang Li, Yizhuo Li, Yinan He, Bingkun Huang, Zhiyu Zhao, Hongjie Zhang, Jilan Xu, Yi Liu, Zun Wang, Sen Xing, Guo Chen, Junting Pan, Jiashuo Yu, Yali Wang, Limin Wang, and Yu Qiao. Internvideo: General video foundation models via generative and discriminative learning. *arXiv preprint arXiv:2212.03191*, 2022. 2

[20] Ying Wang, Yanlai Yang, and Mengye Ren. Lifelongmemory: Leveraging llms for answering queries in long-form egocentric videos, 2024. 1, 2

[21] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 2022. 4

[22] Shoubin Yu, Jaemin Cho, Prateek Yadav, and Mohit Bansal. Self-chained image-language model for video localization and question answering. In *NeurIPS*, 2023. 1, 2, 6

[23] Ce Zhang, Taixi Lu, Md Mohaiminul Islam, Ziyang Wang, Shoubin Yu, Mohit Bansal, and Gedas Bertasius. A simple llm framework for long-range video question-answering, 2023. 1, 2

[24] Chen-Lin Zhang, Jianxin Wu, and Yin Li. Actionformer: Localizing moments of actions with transformers. In *European Conference on Computer Vision*, pages 492–510, 2022. 4

[25] 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

[26] Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric. P Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. Judging llm-as-a-judge with mt-bench and chatbot arena, 2023. 4, 6

[27] Weihong Zhong, Mao Zheng, Duyu Tang, Xuan Luo, Heng Gong, Xiaocheng Feng, and Bing Qin. Stoa-vlp: spatial-temporal modeling of object and action for video-language pre-training. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 3715–3723, 2023. 2