ViQAgent: Zero-Shot Video Question Answering via Agent with Open-Vocabulary Grounding Validation

Tony Montes **Fernando Lozano**
The Department of Electrical and Electronics Engineering
Universidad de los Andes
{t.montes, flozano}@uniandes.edu.co

Abstract

Recent advancements in Video Question Answering (VideoQA) have introduced LLM-based agents, modular frameworks, and procedural solutions, yielding promising results. These systems use dynamic agents and memory-based mechanisms to break down complex tasks and refine answers. However, significant improvements remain in tracking objects for grounding over time and decision-making based on reasoning to better align object references with language model outputs as newer models get better at both tasks. This work presents an LLM-brained agent for zero-shot Video Question Answering (VideoQA) that combines a Chain-of-Thought framework with grounding reasoning alongside YOLO-World to enhance object tracking and alignment. This approach establishes a new state-of-the-art in VideoQA and Video Understanding, showing enhanced performance on NExT-QA, iVQA, and ActivityNet-QA benchmarks. Our framework also enables cross-checking of grounding time-frames, improving accuracy and providing valuable support for verification and increased output reliability across multiple video domains. The code is available here.

1. Introduction

In the present year, the evolution of large language models (LLMs) [11, 26, 32] and vision language models (VLMs) [1, 2, 14, 15, 19, 21, 49] has significantly advanced their ability in video understanding, particularly in the video question-answering (VideoQA) task, a significant challenge on computer vision, where the model is provided with a video and a related question [6, 29, 31, 38] that they must answer as accurately as possible. These models are designed to analyze the visual and linguistic data of the video to generate answers based on semantics and dynamics. Despite recent progress, significant limitations still need to be addressed when addressing more complex videos, particularly those with dynamic context and extensive length. Current LLM-based solutions often struggle with reliably capturing content that is crucial for answering questions when scenes are complex or require a high level of contextual and sequential understanding. This highlights the need for more adaptable approaches that can respond to a broader range of video types and question complexities [17, 20, 25, 40].

To address these challenges, agent-based strategies for VideoQA have recently emerged, employing modular reasoning blocks [23, 28], memory-based strategies [7, 23]. Procedural approaches [4, 28], as well as employing foundational vision-language models as tools for solving complex tasks and augmenting the context [4, 8, 34]. This enables models to track relevant content more effectively over time, which results especially useful for structuring tasks and reasoning through multiple steps, enabling more accurate question answering even when handling diverse and dynamic content. A key approach in these frameworks is video grounding [6, 8, 23, 39], wherein specific video segments are identified as containing objects or events essential to answering the question. In this case, Grounding anchors target segments within the video, allowing the model to focus on relevant portions rather than attempting to analyze entire video sequences. These approaches are also increasingly incorporating Chain-of-Thought (CoT) reasoning. In this method, the model explicitly articulates step-by-step reasoning with purposes of evaluating the consistency [12, 27, 39], planning the steps to solve a task [28], or just bringing a more accurate and argued response [8, 23, 24, 50]. CoT reasoning can enhance certainty in the model's responses, improve interpretability by presenting intermediate reasoning that is understandable to humans, and validate model outputs against additional contextual information.

Inspired by these advances, we propose ViQAgent. This framework combines the capabilities of VideoLLMs and state-of-the-art vision-language foundation models to create a structured agent for VideoQA. ViQAgent employs a VideoLLM [32] to identify the key objects (*i.e.*, targets) in the video that are relevant to the question, generating an initial understanding of the main elements and dynamics in a

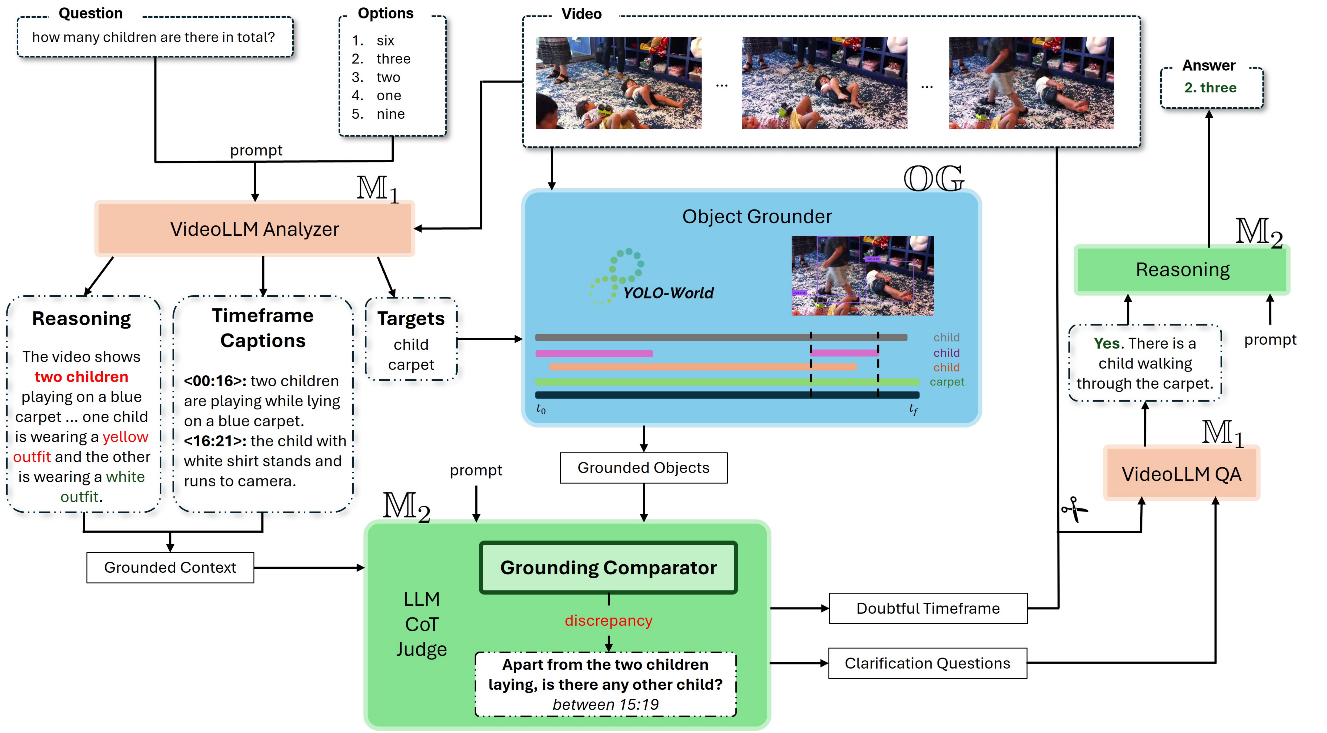


Figure 1. **An overview of our ViQAgent framework**. Through three main modules, we propose an agentic solution for the Video Question-Answering (VideoQA) task by taking advantage of most advanced VideoLLMs capabilities on first-sight zero-shot reasoning, timeframe captioning, and target identification (M_1), and the open-vocabulary capabilities of YOLO-World to ground the given targets/objects in the video (OG) in specific parts of the video in between t_0 and t_f; to finally end with a Chain-of-Thoughts judgment and reasoning layer (M_2) that compares both the grounded context and grounded object detections to determine the confidence of the M_1 answer. In case of discrepancy, the CoT judge defines a set of clarification questions in specific timeframes that go through the VideoLLM again for specific short-ended question-answering. Finally, a reasoning layer takes these answers and the original question to produce a grounded and more accurate answer.

video and a preliminary answer based on observed video content. This initial answer (*i.e.*, *first-sight* response) includes a first attempt to ground it in time and provides a set of key timeframes with their corresponding captions in the video. In the next phase, ViQAgent utilizes YOLO-World [3] to perform object detection based on this list of targets. Here, YOLO-World tracks occurrences of these predefined targets across video frames, returning an accurate timeline of their appearances, a task that sometimes VLMs tend to struggle with, when compared against each other [18]. Unlike traditional object detectors with fixed categories, YOLO-World's tracking is customized by the VideoLLM's selected open-vocabulary targets, allowing it to focus detection on the objects that are most pertinent to solving the question. The timeline created in this step adds a layer of precision, extending the initial detection with frame-by-frame data for each target object and the count of detected objects. In the final phase, the judgment and reasoning layer compares the initial response plus its grounded context, as well as the comprehensive object-tracking data, to determine whether the response is confident or not. This reasoning layer applies Chain-of-Thought (CoT) reasoning to combine both inputs, carefully validating the initial reaction against the grounded data from YOLO-World to produce an answer that is both accurate and substantiated by clear visual evidence. In case of an unconfident answer, a set of expressly framed questions are re-validated through the VideoLLM to provide a final answer then. The CoT approach in this output stage reinforces interpretability by tracing logical steps in reaching the answer and strengthens answer reliability by cross-checking the data from both sources.

Our approach provides multiple advantages as a zero-shot solution, requiring no specialized task-based fine-tuning and thus allowing easy adaptation to new scenarios and question types with just a dataset-specific subprompt. By combining the strengths of VideoLLMs for initial analysis and general video understanding and YOLO-World for open-vocabulary object detailed tracking, it benefits from the complementary capabilities of these models, handling a wide range of video-based questions with minimal configuration. Furthermore, the CoT mechanism contributes to a more nuanced and compelling cross-validation of the reasoning outputs. This grounding structure improves reliability and helps ensure that the content in the video firmly supports all answers. The open nature of ViQAgent's vocabulary for target tracking - dynamically adapted by the VideoLLM layer based on the question - further enhances its flexibility and relevance for various VideoQA tasks.

Empirical results underscore the effectiveness of ViQAgent, as it consistently outperforms current zero-shot solutions across major benchmarks: NExT-QA [37] [22], iVQA [43], ActivityNet-QA [47], and the Egoschema's open subset [22], achieving up to a 4.4% improvement in accuracy. This improvement sets a new state-of-the-art for VideoQA systems, especially in zero-shot scenarios where temporal and spatial reasoning capabilities are essential.

In summary, our contributions are as follows: **(1)** Implementation of ViQAgent, a zero-shot framework that integrates VideoLLMs and vision-language models for effective video grounding, object tracking, and question answering in VideoQA. **(2)** Enhanced interpretability through structured Chain-of-Thought reasoning and grounding outputs, providing interpretable insight into the intermediate steps that inform the final response of the model. **(3)** State-of-the-art zero-shot performance across NExT-QA, iVQA, and ActivityNet-QA benchmarks, highlighting ViQAgent's effectiveness in addressing complex temporal and spatial reasoning tasks with higher accuracy and adaptability.

2. Related Work

Video Question-Answering. Video Question-Answering (VideoQA) has seen significant progress in recent years [6, 10, 17, 20, 25, 40, 45, 51], contributing advancements in both video understanding and natural language processing. Early VideoQA models focused on straightforward tasks, such as frame captioning [45] or simple event identification. Still, these approaches were often limited to fundamental interactions or small, static frameworks [31, 38]. Recent methods have introduced more sophisticated architectures that incorporate attention mechanisms [17, 20, 25, 40], temporal modeling [6, 10], and multi-modal transformers [14, 15, 19, 21, 29, 49] to handle complex questions over temporally extended video sequences. Benchmarks such as NExT-QA [37], iVQA [43], ActivityNet-QA [47], and EgoSchema [22] have challenged models with complex queries that require contextual awareness, reasoning over sequential frames, and an understanding of nuanced interactions. Despite these advancements, current end-to-end approaches struggle with contextual continuity across frames and often require task-specific training to achieve high accuracy. Zero-shot solutions, while desirable, have had limited success in maintaining high generalizability and usually require fine-tuning to increase performance significantly. This highlights the need for more adaptable methods or vision-language tools to respond to diverse question types. ViQAgent seeks to address this gap by utilizing an agent-based approach that grounds key video elements in response to the question, adding a layer of interpretability that can better navigate complex VideoQA challenges in zero-shot scenarios.

LLM Modular Agents. Modular agents built on large language models (LLMs) [4, 7, 8, 23, 28, 34] have gained traction as a solution for decomposing tasks into manageable components (*i.e.*, modules), such as in MoReVQA [23] and TraveLER [28], enabling improved specialization and adaptability in complex problem-solving domains, including VideoQA. These agents use LLMs to orchestrate a series of modular reasoning blocks, each designed to address specific sub-tasks within a broader query. This strategy allows the model to segment complex questions, apply targeted reasoning processes, and sequentially consolidate insights into a final answer. Many modern modular agents also incorporate memory mechanisms, such as VideoAgent [7], TraveLER [28], and MoReVQA [23]. This approach enhances their ability to retain relevant contextual information over time. This is particularly useful in VideoQA, where questions often require remembering objects or events across multiple frames and reasoning about their changes over time. Furthermore, the integration of language-vision foundation models as tools within these modular agents has allowed for more accurate detection, such as in VideoAgent [34], ProViQ [4], and MotionEpic [8]. ViQAgent builds upon this approach by combining VideoLLMs [32] with YOLO-World [3], which adds targeted grounding capabilities to the agent's toolkit, specifically tracking object appearances and interactions as specified by the VideoLLM's outputs. This allows for an additional layer of analysis in complex scenes, enabling the agent to consolidate multiple perspectives and enhance the accuracy and interpretability of its answers.

Chain-of-Thought Reasoning. Chain-of-thought (CoT) reasoning has emerged as a promising approach in complex question-answering, allowing models to break down tasks into sequential, interpretable steps [8, 12, 23, 24, 27, 28, 36, 39, 50]. In VideoQA, where questions often require multi-step reasoning to contextualize events, CoT enables models to articulate intermediate reasoning steps, improving the accuracy of the answer and enhancing the interpretability of the model's outputs. By outputting a transparent sequence of reasoning steps, CoT makes it possible to validate each stage of the decision-making process [23, 50], adding a layer of reliability to model predictions [12, 27, 39], this is particularly observed on the VoT reasoning framework [8]. Grounding, a related concept within CoT frameworks, involves identifying and tracking objects or actions relevant to a given

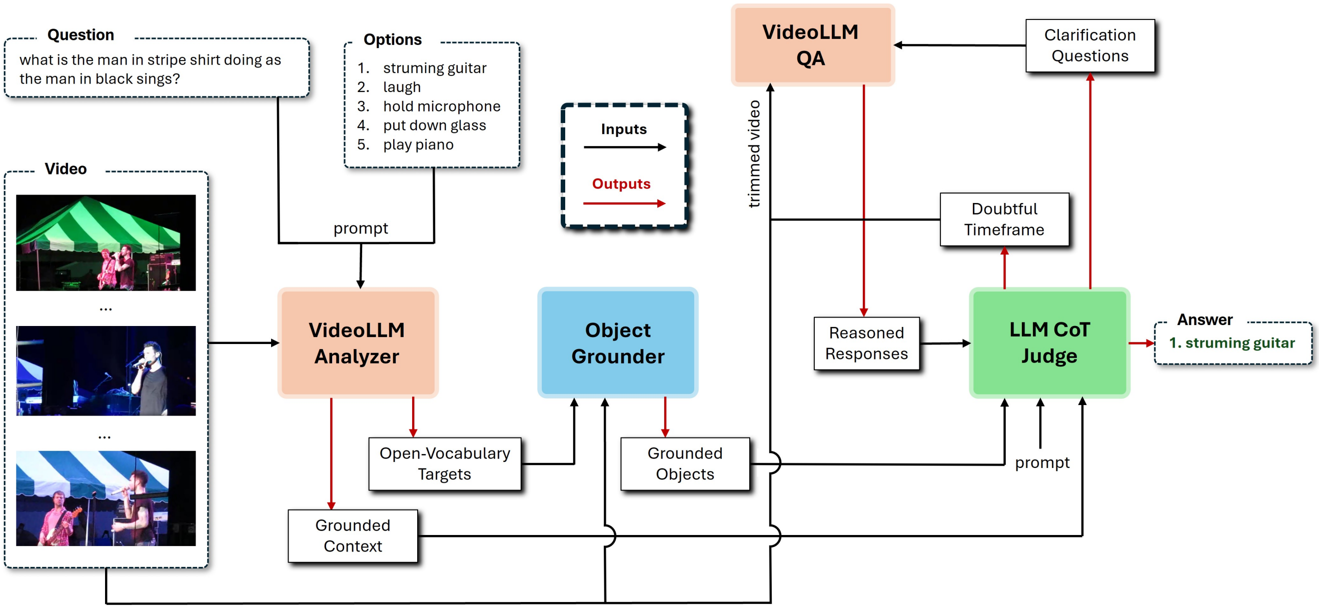


Figure 2. An outline of the black-boxed ViQAgent framework modules **inputs** and **outputs**, and the intermediate representations, that allow to track and understand the final selected answer. The (M_1) **inputs** are the video and the question plus the answer options (namely *prompt*). In contrast, the **outputs** are the open-vocabulary targets, and the reasoning plus timeframe captions (namely *Grounded Context*). The (OG) **inputs** are the targets and the video, and the **output** is the object detection timeline (namely *Grounded Objects*). Finally, the (M_2) first **receives** both ground responses and the prompt, then, if there seem to be inconsistencies, **returns** a doubtful timeframe and a set of clarification questions to make to the VideoLLM from that specific timeframe. The answers are then **re-inputted** to **produce** the final answer.

question across frames [6, 10, 39]. This process creates a temporal map of pertinent elements, providing a more structured basis for the CoT to build upon. ViQAgent leverages CoT by incorporating it within its final judgment and reasoning layer, using CoT steps to align the preliminary answer and grounded object tracking data from YOLO-World, ultimately refining the response based on these validated insights through several validation questions about them. This structured approach ensures that each reasoning step is clear and accessible, supporting the model's interpretability and enhancing its overall robustness in VideoQA tasks.

3. ViQAgent Framework

In this section, we describe in detail the implementation details of each module of the ViQAgent framework (Fig. 2), starting with a general overview and task definition (Sec. 3.1), module implementation details (Sec. 3.2, Sec. 3.3, and Sec. 3.4), to finally end with an algorithm of the full implementation (Sec. 3.5).

3.1. Overview

ViQAgent answers both open- and close-ended questions about video content using three interconnected modules. First, a VideoLLM-based module M_1 provides initial insights into the video's relevance to the question. Next, an open-vocabulary grounding module OG detects and tracks detected targets across frames. Finally, a reasoning module M_2 validates, cross-checks, and refines answers using prior outputs, forming a cohesive chain-of-thought question-based reasoning process. ViQAgent thus identifies, tracks, and reasons over relevant video segments, generating accurate answers with minimal prior knowledge or domain-specific data.

In a VideoQA task, the model is given a question Q and a video V, composed of n frames $V = [v_1, ..., v_n]$, based on the video's frames-per-second ratio. For close-ended questions, the model also receives a set of answer options A_{opt}, ensuring that the output answer must be in that set $A \in A_{opt}$ [23]. Only Q and V are provided for open-ended questions, though A_{opt} is typically used to gauge answer similarity before the ground truth and the model's output. Our pipeline aims to solve the VideoQA task with an intermediate rationale for increasing interpretability. The task can be defined as follows, assuming a solution system S:

$$S(V, Q + [A_{opt}]) \to A \quad (1)$$

3.2. VideoLLM Analyzer

The analyzer module M_1, also referred to as the *first-sight assessment*, employs a VideoLLM to provide a preliminary

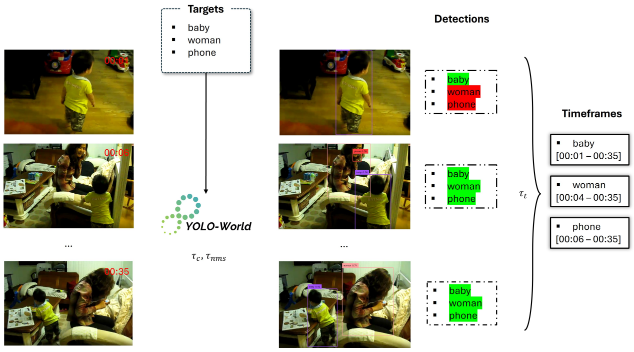


Figure 3. **A more detailed overview of the internal process of the *OG* module.** The process begins by extracting all frames from the input video V. For each frame v_i, the YOLO-World model detects specified target classes within the frame, using predetermined confidence and NMS thresholds (τ_c, τ_{nms}). After detection, these classes are tracked across all frames to establish the exact time intervals during which they are present. If a detected object is absent from subsequent frames for a specified duration τ_t, it is assumed to have exited the scene, marking the end of its appearance.

interpretation of the video for the question. Given the question Q, and A_{opt}, and the entire video V, this module generates three outputs:

1. Open-Vocabulary Targets: The VideoLLM identifies a set of open-vocabulary targets $\{T_1, T_2, ..., T_m\}$ essential to solving the question. These targets are derived from both the question context and the video content, forming an initial roadmap of entities or events likely to contribute to the answer.

2. Preliminary Answer and Reasoning: Based on its analysis, the VideoLLM proposes an initial answer A_1 to the question, with a detailed rationale R_1 that articulates why this answer might be appropriate, and that will later aid in the validation process.

3. Scene-Segmented Timeframes: The VideoLLM segments the video into discrete scenes based on changes it identifies as relevant to the question and generates a caption describing in detail what happens in that scene. These segmented timeframes serve as markers for different key moments in the video, where one significant scene ends and another begins, establishing a structured temporal foundation for subsequent grounding and reasoning steps. These segments can be modeled as a set of pairs of timeframe-caption $\{(t_i, t_f) : c\}$.

3.3. Open-Vocabulary Object Grounding

The grounding module OG (Fig. 3) performs an object-tracking task using the open-vocabulary targets identified by the analyzer module M_1. This module is responsible for detecting and locating in time target objects within the video frames and is structured as follows:

1. Object Detection: Utilizing YOLO-World [3] or a similar open-vocabulary object detection model, this phase processes each video frame v_i to identify instances of the relevant targets. Applying a confidence threshold to τ_c ensures that only credible detections are retained. Additionally, an NMS (non-maximum suppression) threshold τ_{nms} is used to filter redundant or overlapping detections, preserving the most significant object representations and avoiding overlapping noise in the results.

2. Timeframe Extraction: With object detections in place, this phase extracts timeframes where targets are identified, considering a time threshold τ_t that reduces the risk of short-lived false negatives (*e.g.*, objects momentarily obscured or blurred). This time threshold is highly beneficial to ensure precise temporal localization so that detected objects remain consistent with the scene continuity without responding to minor, transient distortions. As a result, the grounding module outputs a set of well-defined timeframes where each target is reliably tracked. These extractions are a list of timeframes for a particular target $\{T_1 : [(t_{i0}, t_{f0}), ..., (t_{in}, t_{fn})], ...\}$.

These three hyperparameters $(\tau_c, \tau_{nms}, \tau_t)$ were fine-tuned under a set of tests before running the benchmarks and are crucial to achieving balanced and precise object grounding having been optimized to ensure comprehensive target coverage without excessive or unreliable detections.

3.4. CoT Judgment

The final module M_2 performs in-depth reasoning and cross-validation, combining outputs from the previous modules to deliver a definitive answer to the question. This module acts first as a judge, evaluating and consolidating all prior insights through a chain-of-thought approach, and reutilizes the M_1 logic in case of inconsistencies, then, with the additional information, acts as a reasoner to provide a final answer:

1. Comparison judgment First, both the M_1 reasoning output and the scene-segmented timeframes are merged into a single *grounded context*, that contains information regarding the overall video and a first answer to be judged. This context is compared against the *OG grounded object* timeline. The main output from this comparison judgment is a decision of whether they are consistent and, if not, a specific reasoning of why and where they are inconsistent.

2. Question Generation: If the outputs are effectively discrepant, with the aid of the why/where specifications of the discrepancy, and the original question Q and answer options A_{opt}, a set of one or more clarification questions are formulated for the given timeframe, in order to obtain additional information for being able to finally validate the confidence of the answer candidate proposed by M_1. These clarification questions are then fed to the M_1's VideoLLM instance for simple question-answering. The answers are then analyzed

in a further step.
3. Answer Refinement: Based on the clarification questions created in the previous step and their corresponding answers given by the VideoLLM, the module adjusts or refines the initial answer where necessary, ensuring that the final output reflects a coherent understanding of the video as well as consistency with both visual and semantic information.

3.5. ViQAgent

The end-to-end ViQAgent pipeline integrates these modules in a sequential, end-to-end manner to provide an accurate, reliable answer for VideoQA tasks. The algorithm of the whole framework can be seen in Algorithm 1.

By structuring each module's operations this way, ViQAgent provides a robust VideoQA pipeline capable of answering complex questions in a zero-shot setting. It employs open-vocabulary grounding and modular reasoning to handle diverse visual-linguistic queries.

4. Experiments

This section outlines the setup for benchmarking and evaluating ViQAgent's performance. We detail the datasets and metrics used, compare them against relevant baselines, and discuss modification studies that inform key hyperparameter choices in our model. Implementation details and a comprehensive analysis of our results are also provided.

4.1. Datasets and Metrics

We evaluated ViQAgent on four widely recognized video question answering (VideoQA) benchmarks, each representing unique video types, question formats, and challenges in the domain.
1. NExT-QA [37]: This dataset tests reasoning over causal, temporal, and descriptive question types. In our experiments, we used the validation split containing 4,996 video-question pairs. Each question is close-ended, presenting five answer options, with ViQAgent tasked with selecting the correct one.
2. iVQA [43]: iVQA comprises instructional video clips from the HowTo100M dataset, lasting 7-30 seconds. Each video clip includes a question and an annotated set of ground truth answers, with ViQAgent evaluated on the test set, which consists of 1,879 clips. Notably, iVQA is an open-ended VideoQA task, requiring ViQAgent to generate responses without candidate options.
3. ActivityNet-QA [47]: This dataset includes 5,800 videos, each with ten annotated question-answer pairs, covering actions, objects, locations, and events. Similar to iVQA, ActivityNet-QA is an open-ended VideoQA dataset. For consistency with previous works, we report results on the test split, utilizing an evaluation based on a large language model (LLM) comparison between ViQAgent's response and ground-truth answers (see Appendix C.1).

Algorithm 1 ViQAgent framework algorithm, after initializing the models YoloWorld, VideoLLM$_{1,2,3,4}$, LLM$_{1,2,3}$

```
**Hyperparameters:** $\tau_c, \tau_{nms}, \tau_t$
**Input:** $V, Q, [A_{opt}]$
$prompt \leftarrow Q + [A_{opt}]$
                                                        ▷ $M_1$ starts
$A, R_1 \leftarrow$ VideoLLM$_1(V, prompt)$             ▷ Rationale
$TC \leftarrow$ VideoLLM$_2(V)$                         ▷ Timeframes
$T \leftarrow$ VideoLLM$_3(V, prompt)$                  ▷ Targets
$\{T_1, T_2, ..., T_m\} \leftarrow T$
                                                        ▷ $OG$ starts
YoloWorld.set_classes($T_1, T_2, ..., T_m$)
$D \leftarrow \{T_1 : \varnothing, T_2 : \varnothing, ..., T_m : \varnothing\}$    ▷ Detections
for $v_i$ in $V$ do
    $d \leftarrow$ YoloWorld.detect($v_i, \tau_c, \tau_{nms}$)
    for $T_i$ in $T$ do
        $D[T_i] \leftarrow D[T_i] + d[T_i]$
    end for
end for
$t \leftarrow 0$
$TG \leftarrow \{T_1 : \varnothing, T_2 : \varnothing, ..., T_m : \varnothing\}$   ▷ Timeframes
for $T_i$ in $T$ do
    $idx \leftarrow 0$
    for $d$ in $D$ do
        $t \leftarrow t + (V\texttt{.fps})^{-1}$
        if $d[T_i] \neq \varnothing$ then
            if $idx + 1 \neq |TG[T_i]|$ then
                $TG[T_i][idx] \leftarrow (t, 0)$
            else
                $TG[T_i][idx][1] \leftarrow t$
            end if
        else
            $\Delta t \leftarrow t - TG[T_i][idx][1]$
            if $\Delta t \geq \tau_t$ then
                $idx \leftarrow idx + 1$
            end if
        end if
    end for
end for
                                                        ▷ $M_2$ starts
$\mu \leftarrow$ LLM$_1(R_1, TC, TG)$                   ▷ Unconfidence
if $\mu$ then
    $W \leftarrow V\texttt{.trim}(\mu\texttt{.times})$
    $qs \leftarrow$ LLM$_2(\mu, prompt)$
    $as \leftarrow \varnothing$
    for $q$ in $qs$ do
        $as \leftarrow as+$VideoLLM$_4(W, q)$
    end for
    $A \leftarrow$ LLM$_3(R_1, prompt, qs, as)$
end if
```

Method	Zero-Shot	Agent	Acc@C	Acc@T	Acc@D	Acc@All
HiTeA	✗	✗	62.4	58.3	75.6	63.1
LLaMa-VQA	✗	✗	72.7	69.2	75.8	72.0
SeViLa (fine-tuned)	✗	✗	-	-	-	73.8
InternVideo	✓	✗	43.4	48.0	65.1	49.1
AssistGPT	✓	✗	60.0	51.4	67.3	58.4
SeViLa	✓	✗	61.3	61.5	75.6	63.6
ProViQ	✓	✓	-	-	-	63.8
ViperGPT+	✓	✓	-	-	-	64.0
JCEF	✓	✓	-	-	-	66.7
LLoVi	✓	✓	69.5	61.0	75.6	67.7
TraveLER	✓	✓	70.0	60.5	78.2	68.2
MoReVQA	✓	✓	-	-	-	69.2
VideoAgent (mem)	✓	✓	60.0	76.0	76.5	70.8
VideoAgent (long)	✓	✓	72.7	64.5	81.1	71.3
MotionEpic	✓	✓	75.8	74.6	83.3	76.0
ViQAgent (ours)	✓	✓	**82.2**	74.5	**86.3**	**80.4**

Table 1. Results of ViQAgent against all the other state-of-the-art solutions in NExT-QA benchmark, in all the casual (C), temporal (T), and descriptive (D) subsets, as well as the overall accuracy. ViQAgent surpasses all the baselines, achieving a new state-of-the-art on VideoQA.

Method	Zero-Shot	Agent	Acc
VideoCoCa	✗	✗	39.0
FrozenBiLM (fine-tuned)	✗	✗	39.7
FrozenBiLM	✓	✗	27.3
BLIP-2	✓	✗	45.8
InstructBLIP	✓	✗	53.8
ProViQ	✓	✓	50.7
JCEF	✓	✓	56.9
MoReVQA	✓	✓	60.9
ViQAgent (ours)	✓	✓	**62.6**

(a) iVQA results.

Method	Zero-Shot	Agent	Acc
FrozenBiLM (fine-tuned)	✗	✗	43.2
Video-ChatGPT	✓	✗	35.2
Video-LLaVa	✓	✗	45.3
VideoChat2	✓	✗	49.1
ViperGPT+	✓	✓	37.1
ProViQ	✓	✓	42.3
JCEF	✓	✓	43.3
MoReVQA	✓	✓	45.3
MotionEpic	✓	✓	54.6
ViQAgent (ours)	✓	✓	**59.9**

(b) ActivityNet-QA results.

Table 2. Open-ended Question-Answering benchmark results.

4. EgoSchema [22]: Focused on long-form, egocentric video understanding, EgoSchema contains 3-minute clips sourced from the Ego4D benchmark. It includes close-ended questions with higher complexity and length than NExT-QA's, assessing a model's ability to handle extended video content. We evaluated ViQAgent on the available 500 samples of the open-answer split, focusing solely on accuracy. This experiment, in particular, is not considered a paper contribution, as we didn't evaluate the whole benchmark, but it still provides an addition to previous results.

Across all benchmarks, accuracy was used as the primary evaluation metric, providing a direct and interpretable measure of ViQAgent's performance on close-ended and open-ended tasks alike.

4.2. Hyperparameter Tuning

To refine ViQAgent's hyperparameters, we performed a series of simple modification studies on the NExT-QA dataset, specifically on each experiment's small, randomly sampled, evenly class-distributed subset of videos. We investigated the effects of varying detection thresholds in the object grounding module and the time threshold to separate appearance timeframes effectively:

Confidence Threshold. We evaluated confidence values of 0.01, 0.05, 0.1, and 0.3 for the object detection model within YOLO-World. A default confidence of **0.05** was found optimal, balancing sensitivity to target objects with tolerance to noise in lower-resolution, motion-filled frames. This value is considerably lower than the 0.3 confidence suggested in

Method	Agent	Acc Subset	Acc
SeViLa	✗	25.7	22.7
ImageViT	✗	40.8	30.9
ShortViViT	✗	49.6	31.0
InternVideo	✗	-	32.1
LongViViT	✗	56.8	33.3
MC-ViT-L	✗	62.6	44.4
Vamos	✗	-	48.3
JCEF	✓	-	50.0
LLoVi	✓	57.6	50.3
MoReVQA	✓	-	51.7
TraveLER	✓	-	53.3
VideoAgent (long)	✓	60.2	54.1
ProViQ	✓	-	57.1
ViQAgent (ours)	✓	**67.87**	-

Table 3. EgoSchema results.

Benchmark	Increase of the SotA
ActivityNet-QA	+5.3%
NExT-QA	+4.4%
iVQA	+1.7%

Table 4. Increase on the state-of-the-art results on the benchmarks.

the original YOLO-World paper [3] due to the challenges posed by low-resolution and high-motion video frames in the benchmarks, which is not the case on image-level object detection. This reduction ensures that important objects are detected, accounting for YOLO-World's open-vocabulary capabilities, which can introduce variability in detection sensitivity based on the complexity of target object sets provided by the first module.

NMS Threshold. The non-maximum suppression (NMS) threshold, controlling overlap in detections, was tested with intersection-over-union (IoU) values of 0.1 and 0.3. An IoU of **0.1** performed best, effectively reducing overlapping false positives while preserving distinct object instances in most of the cases, essential for ViQAgent's open-vocabulary object grounding approach.

Time Threshold. We tested temporal segmentation with 500ms, 1s, and 1.5s time thresholds. A **1.5-second** threshold was chosen because it captured consistent object presence across varying scene dynamics, particularly useful in lengthy benchmark videos, ensuring detection continuity despite video length and quality variances. However, minor changes on this threshold didn't result in significant differences.

4.3. Results and Discussion

ViQAgent not only surpasses prior state-of-the-art results across all evaluated benchmarks (Tab. 4) but also establishes itself as a leading choice in both zero-shot VideoQA and modular, agent-based solutions. In the close-ended NExT-QA benchmark (Tab. 1), ViQAgent demonstrated a significant edge, achieving a 4.4% increase in accuracy over the previous best-performing Video-of-Though (VoT) solution: MotionEpic [8]. One particularly notable finding is that our zero-shot model outperforms fine-tuned models that are optimized specifically for this benchmark, such as SeViLa [46], LLaMa-VQA [13], and HiTeA [44]. This result highlights ViQAgent's capability to generalize effectively without needing task-specific training, underscoring its potential as a more adaptable and resource-efficient solution for VideoQA tasks. This zero-shot superiority not only raises the benchmark for VideoQA performance but also signals the value of robust, generalized VideoLLMs in solving real-world, unseen problems without extensive fine-tuning.

Results across the open-ended benchmarks—iVQA (Tab. 2a), ActivityNet-QA (Tab. 2b)—further validates ViQAgent's robustness and adaptability to diverse video types and question structures, showing its performance as a versatile VideoQA agent capable of handling both close-ended and open-ended tasks, while being able to surpass robust state-of-the-art agentic solutions for VideoQA: MoReVQA [23] (and it's additional contributions JCEF and ViperGPT+), TraveLER [28], ProViQ [4], MotionEpic [8], VideoAgent [7], and VideoAgent [34].

Further, ViQAgent shows promising results on the partial open-answer subset of questions within the EgoSchema benchmark (Tab. 3), expanding its potential applicability as a flexible VideoQA model well-suited for both straightforward and complex agentic tasks.

5. Conclusion

In this paper, we presented ViQAgent, a novel framework that advances the field of video question answering (VideoQA) by leveraging a modular, task-specific approach tailored for zero-shot generalization across diverse VideoQA benchmarks. By employing a strategically layered architecture that integrates open-vocabulary object detection, object grounding, and chain-of-thought reasoning, ViQAgent dynamically adapts to the complexities of various video domains, enabling it to successfully tackle both close-ended and open-ended questions without requiring extensive hours of fine-tuning or training. Through extensive benchmarking on datasets such as NExT-QA, iVQA, and ActivityNet-QA, we demonstrated ViQAgent's superior performance, significantly outperforming existing zero-shot and modular-agent approaches. Our experiments also highlight the adaptability of our framework to low-resolution, high-motion, and lengthy videos, settings that traditionally challenge VideoQA models. By pushing the boundaries of zero-shot VideoQA, setting a new standard for intelligent and efficient multimodal comprehension in versatile video understanding.

References

[1] Xi Chen, Josip Djolonga, Piotr Padlewski, Basil Mustafa, Soravit Changpinyo, Jialin Wu, Carlos Riquelme Ruiz, Sebastian Goodman, Xiao Wang, Yi Tay, Siamak Shakeri, Mostafa Dehghani, Daniel Salz, Mario Lucic, Michael Tschannen, Arsha Nagrani, Hexiang Hu, Mandar Joshi, Bo Pang, Ceslee Montgomery, Paulina Pietrzyk, Marvin Ritter, AJ Piergiovanni, Matthias Minderer, Filip Pavetic, Austin Waters, Gang Li, Ibrahim Alabdulmohsin, Lucas Beyer, Julien Amelot, Kenton Lee, Andreas Peter Steiner, Yang Li, Daniel Keysers, Anurag Arnab, Yuanzhong Xu, Keran Rong, Alexander Kolesnikov, Mojtaba Seyedhosseini, Anelia Angelova, Xiaohua Zhai, Neil Houlsby, and Radu Soricut. Pali-x: On scaling up a multilingual vision and language model, 2023. 1

[2] Xi Chen, Xiao Wang, Soravit Changpinyo, AJ Piergiovanni, Piotr Padlewski, Daniel Salz, Sebastian Goodman, Adam Grycner, Basil Mustafa, Lucas Beyer, Alexander Kolesnikov, Joan Puigcerver, Nan Ding, Keran Rong, Hassan Akbari, Gaurav Mishra, Linting Xue, Ashish Thapliyal, James Bradbury, Weicheng Kuo, Mojtaba Seyedhosseini, Chao Jia, Burcu Karagol Ayan, Carlos Riquelme, Andreas Steiner, Anelia Angelova, Xiaohua Zhai, Neil Houlsby, and Radu Soricut. Pali: A jointly-scaled multilingual language-image model, 2023. 1

[3] Tianheng Cheng, Lin Song, Yixiao Ge, Wenyu Liu, Xinggang Wang, and Ying Shan. Yolo-world: Real-time open-vocabulary object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 16901–16911, 2024. 2, 3, 5, 8

[4] Rohan Choudhury, Koichiro Niinuma, Kris M. Kitani, and László A. Jeni. Zero-shot video question answering with procedural programs. *CoRR*, abs/2312.00937, 2023. 1, 3, 8, 12

[5] Wenliang Dai, Junnan Li, Dongxu Li, Anthony Tiong, Junqi Zhao, Weisheng Wang, Boyang Li, Pascale Fung, and Steven Hoi. InstructBLIP: Towards general-purpose vision-language models with instruction tuning. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. 12

[6] Shangzhe Di and Weidi Xie. Grounded question-answering in long egocentric videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 12934–12943, 2024. 1, 3, 4

[7] 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 *Computer Vision – ECCV 2024*, pages 75–92, Cham, 2025. Springer Nature Switzerland. 1, 3, 8, 12

[8] Hao Fei, Shengqiong Wu, Wei Ji, Hanwang Zhang, Meishan Zhang, Mong-Li Lee, and Wynne Hsu. Video-of-thought: Step-by-step video reasoning from perception to cognition. In *Forty-first International Conference on Machine Learning*, 2024. 1, 3, 8, 12

[9] Difei Gao, Lei Ji, Luowei Zhou, Kevin Qinghong Lin, Joya Chen, Zihan Fan, and Mike Zheng Shou. Assistgpt: A general multi-modal assistant that can plan, execute, inspect, and learn, 2023. 12

[10] Xin Gu, Heng Fan, Yan Huang, Tiejian Luo, and Libo Zhang. Context-guided spatio-temporal video grounding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 18330–18339, 2024. 3, 4

[11] Albert Q. Jiang, Alexandre Sablayrolles, Antoine Roux, Arthur Mensch, Blanche Savary, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Emma Bou Hanna, Florian Bressand, Gianna Lengyel, Guillaume Bour, Guillaume Lample, Lélio Renard Lavaud, Lucile Saulnier, Marie-Anne Lachaux, Pierre Stock, Sandeep Subramanian, Sophia Yang, Szymon Antoniak, Teven Le Scao, Théophile Gervet, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. Mixtral of experts, 2024. 1

[12] Zaid Khan and Yun Fu. Consistency and uncertainty: Identifying unreliable responses from black-box vision-language models for selective visual question answering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10854–10863, 2024. 1, 3

[13] Dohwan Ko, Ji Lee, Woo-Young Kang, Byungseok Roh, and Hyunwoo Kim. Large language models are temporal and causal reasoners for video question answering. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 4300–4316, Singapore, 2023. Association for Computational Linguistics. 8, 12

[14] Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models, 2023. 1, 3, 12

[15] KunChang Li, Yinan He, Yi Wang, Yizhuo Li, Wenhai Wang, Ping Luo, Yali Wang, Limin Wang, and Yu Qiao. Videochat: Chat-centric video understanding, 2024. 1, 3

[16] 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, 2024. 12

[17] Zhaohe Liao, Jiangtong Li, Li Niu, and Liqing Zhang. Align and aggregate: Compositional reasoning with video alignment and answer aggregation for video

question-answering. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 13395–13404, 2024. 1, 3

[18] Christian Limberg, Artur Gonçalves, Bastien Rigault, and Helmut Prendinger. Leveraging YOLO-world and GPT-4v LMMs for zero-shot person detection and action recognition in drone imagery. In *First Workshop on Vision-Language Models for Navigation and Manipulation at ICRA 2024*, 2024. 2

[19] 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, 2024. 1, 3, 12

[20] Fan Ma, Xiaojie Jin, Heng Wang, Yuchen Xian, Jiashi Feng, and Yi Yang. Vista-llama: Reducing hallucination in video language models via equal distance to visual tokens. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 13151–13160, 2024. 1, 3

[21] Muhammad Maaz, Hanoona Rasheed, Salman Khan, and Fahad Shahbaz Khan. Video-chatgpt: Towards detailed video understanding via large vision and language models, 2024. 1, 3, 12, 13

[22] Karttikeya Mangalam, Raiymbek Akshulakov, and Jitendra Malik. Egoschema: A diagnostic benchmark for very long-form video language understanding. In *Advances in Neural Information Processing Systems*, pages 46212–46244. Curran Associates, Inc., 2023. 3, 7, 12

[23] 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 (CVPR)*, pages 13235–13245, 2024. 1, 3, 4, 8, 12, 13

[24] Chancharik Mitra, Brandon Huang, Trevor Darrell, and Roei Herzig. Compositional chain-of-thought prompting for large multimodal models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 14420–14431, 2024. 1, 3

[25] David Mogrovejo and Thamar Solorio. Question-instructed visual descriptions for zero-shot video answering. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 9329–9339, Bangkok, Thailand, 2024. Association for Computational Linguistics. 1, 3

[26] OpenAI. Gpt-4 technical report, 2024. 1

[27] Peng Qi, Zehong Yan, Wynne Hsu, and Mong Li Lee. Sniffer: Multimodal large language model for explainable out-of-context misinformation detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 13052–13062, 2024. 1, 3

[28] Chuyi Shang, Amos You, Sanjay Subramanian, Trevor Darrell, and Roei Herzig. TraveLER: A modular multi-LMM agent framework for video question-answering. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 9740–9766, Miami, Florida, USA, 2024. Association for Computational Linguistics. 1, 3, 8, 12

[29] Enxin Song, Wenhao Chai, Guanhong Wang, Yucheng Zhang, Haoyang Zhou, Feiyang Wu, Haozhe Chi, Xun Guo, Tian Ye, Yanting Zhang, Yan Lu, Jenq-Neng Hwang, and Gaoang Wang. Moviechat: From dense token to sparse memory for long video understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 18221–18232, 2024. 1, 3

[30] 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 (ICCV)*, pages 11888–11898, 2023. 12

[31] Yunlong Tang, Jing Bi, Siting Xu, Luchuan Song, Susan Liang, Teng Wang, Daoan Zhang, Jie An, Jingyang Lin, Rongyi Zhu, Ali Vosoughi, Chao Huang, Zeliang Zhang, Pinxin Liu, Mingqian Feng, Feng Zheng, Jianguo Zhang, Ping Luo, Jiebo Luo, and Chenliang Xu. Video understanding with large language models: A survey, 2024. 1, 3

[32] Gemini Team. Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context, 2024. 1, 3, 13

[33] Shijie Wang, Qi Zhao, Minh Quan Do, Nakul Agarwal, Kwonjoon Lee, and Chen Sun. Vamos: Versatile action models for video understanding, 2024. 12

[34] Xiaohan Wang, Yuhui Zhang, Orr Zohar, and Serena Yeung-Levy. Videoagent: Long-form video understanding with large language model as agent. In *Computer Vision – ECCV 2024*, pages 58–76, Cham, 2025. Springer Nature Switzerland. 1, 3, 8, 12

[35] 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, 2022. 12

[36] Yan Wang, Yawen Zeng, Jingsheng Zheng, Xiaofen Xing, Jin Xu, and Xiangmin Xu. VideoCoT: A video chain-of-thought dataset with active annotation tool. In *Proceedings of the 3rd Workshop on Advances in Language and Vision Research (ALVR)*, pages 92–101, Bangkok, Thailand, 2024. Association for Computational Linguistics. 3

[37] Junbin Xiao, Xindi Shang, Angela Yao, and Tat-Seng Chua. Next-qa:next phase of question-answering to explaining temporal actions, 2021. 3, 6, 12

[38] Junbin Xiao, Nanxin Huang, Hangyu Qin, Dongyang Li, Yicong Li, Fengbin Zhu, Zhulin Tao, Jianxing Yu, Liang Lin, Tat-Seng Chua, and Angela Yao. Videoqa in the era of llms: An empirical study, 2024. 1, 3

[39] Junbin Xiao, Angela Yao, Yicong Li, and Tat-Seng Chua. Can i trust your answer? visually grounded video question answering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 13204–13214, 2024. 1, 3, 4

[40] Jiaqi Xu, Cuiling Lan, Wenxuan Xie, Xuejin Chen, and Yan Lu. Retrieval-based video language model for efficient long video question answering, 2023. 1, 3

[41] Shen Yan, Tao Zhu, Zirui Wang, Yuan Cao, Mi Zhang, Soham Ghosh, Yonghui Wu, and Jiahui Yu. Video-coca: Video-text modeling with zero-shot transfer from contrastive captioners, 2023. 12

[42] Antoine Yang, Antoine Miech, Josef Sivic, Ivan Laptev, and Cordelia Schmid. Zero-shot video question answering via frozen bidirectional language models, 2022. 12

[43] Antoine Yang, Antoine Miech, Josef Sivic, Ivan Laptev, and Cordelia Schmid. Learning to answer visual questions from web videos, 2022. 3, 6, 12, 13

[44] Qinghao Ye, Guohai Xu, Ming Yan, Haiyang Xu, Qi Qian, Ji Zhang, and Fei Huang. Hitea: Hierarchical temporal-aware video-language pre-training. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 15405–15416, 2023. 8, 12

[45] Shuhong Ye, Weikai Kong, Chenglin Yao, Jianfeng Ren, and Xudong Jiang. Video question answering using clip-guided visual-text attention. In *2023 IEEE International Conference on Image Processing (ICIP)*, pages 81–85, 2023. 3

[46] Shoubin Yu, Jaemin Cho, Prateek Yadav, and Mohit Bansal. Self-chained image-language model for video localization and question answering. In *Advances in Neural Information Processing Systems*, pages 76749–76771. Curran Associates, Inc., 2023. 8, 12

[47] Zhou Yu, Dejing Xu, Jun Yu, Ting Yu, Zhou Zhao, Yueting Zhuang, and Dacheng Tao. Activitynet-qa: A dataset for understanding complex web videos via question answering, 2019. 3, 6, 12

[48] 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, 2024. 12

[49] Hang Zhang, Xin Li, and Lidong Bing. Video-llama: An instruction-tuned audio-visual language model for video understanding, 2023. 1, 3

[50] Shitian Zhao, Zhuowan Li, Yadong Lu, Alan Yuille, and Yan Wang. Causal-cog: A causal-effect look at context generation for boosting multi-modal language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 13342–13351, 2024. 1, 3

[51] Yaoyao Zhong, Wei Ji, Junbin Xiao, Yicong Li, Weihong Deng, and Tat-Seng Chua. Video question answering: Datasets, algorithms and challenges. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 6439–6455, Abu Dhabi, United Arab Emirates, 2022. Association for Computational Linguistics. 3

ViQAgent: Zero-Shot Video Question Answering via Agent with Open-Vocabulary Grounding Validation

Supplementary Material

This supplementary material provides detailed information about the ViQAgent framework implementation. It includes a case study showcasing the outputs at each stage of the framework and analyzing observations from each step (App. A). Additionally, the prompts and schemas used in the solution are detailed (App. B), along with the baselines and benchmark configurations (App. C). Finally, insights and implications are discussed in greater depth (App. D).

A. Case Study

To illustrate the internal outputs and their overall contribution to the final answer, we present a comprehensive step-by-step case study. This example demonstrates a scenario where the VideoLLM fails to answer the question directly but provides hints that facilitate the detection of subsequent inconsistencies, ultimately leading to a correct answer. The case study is detailed in Figs. 4–10. These steps align with the methodology and sequence outlined in Algorithm 1.

B. Prompts and Schemas

In the ViQAgent framework, multiple LLMs are utilized in both the M_1 and M_2 modules. Module 1 is responsible for independently extracting relevant information and reasoning from the video, while Module 2 compiles and evaluates the final answer based on the information gathered.

Module 1 directly interacts with the video through VideoLLMs, whereas Module 2 leverages LLMs for reasoning using the pre-computed information without direct access to the video. The prompts and output schemas for all VideoLLM$_{1,2,3,4}$ (Module 1) and LLM$_{1,2,3}$ are presented in Tabs. 5 - 18, following the procedure shown previously in the Algorithm 1. Notably, for VideoLLM$_1$ (Tab. 5) and LLM$_3$ (Tab. 17), a space is left at the end. This is because these two submodules are specifically responsible for providing direct answers to the query. As the question formats depend on the specific benchmark, an additional subinstruction is appended in this section to accommodate the requirements of the respective benchmark.

C. Baselines and Benchmarks

For each of the evaluated benchmarks, we conducted a thorough review of the most relevant state-of-the-art solutions from recent years that have reported results on these benchmarks, focusing on those that achieved notably high accuracy or are widely recognized baselines evaluated against previous work.

NExT-QA [37]: We compare against fine-tuned solutions that achieved high performance across both specific subsets (Casual, Temporal, and Descriptive) and the overall subset. The solutions include HiTeA [44], LLaMa-VQA [13], and a fine-tuned version of SeViLa [46]. Additionally, several zero-shot solutions emerged, as NExT-QA is a significant benchmark for evaluating Video Question-Answering solutions. We classified these solutions into agentic and non-agentic categories. The non-agentic solutions include InternVideo [35], AssistGPT [9], and SeViLa [46]. The agentic solutions, which are more pertinent for comparison with our framework, include ProViQ [4], ViperGPT+, JCEF, MoReVQA (all introduced in Min et al. [23], with ideas from Surís et al. [30]), LLoVi [48], TraveLER [28], VideoAgent [7], VideoAgent [34], and MotionEpic [8] with its Video-of-Thought (VoT) framework.

iVQA [43]: For this dataset, only two fine-tuned solutions were identified, as reported in Min et al. [23]: VideoCoCa [41] and a fine-tuned version of FrozenBiLM [42]. Its non-fine-tuned counterpart, alongside BLIP-2 [14] and InstructBLIP [5], forms the zero-shot non-agentic baselines for comparison. Regarding the agentic solutions, which are particularly relevant to our framework, the notable baselines include ProViQ [4], JCEF, and MoReVQA [23].

ActivityNet-QA [47]: For the ActivityNet-QA benchmark, the fine-tuned group comprises only the FrozenBiLM solution [42]. The non-agentic zero-shot baselines include Video-ChatGPT [21], Video-LLaVa [19], and VideoChat2 [16]. For agentic solutions, the primary baselines are ViperGPT+, JCEF, MoReVQA [23], ProViQ [4], and MotionEpic [8], which employs the VoT framework.

EgoSchema [22]: Although EgoSchema is not a primary benchmark for our solution, several baselines were compared with our partial results. It is noteworthy that fine-tuning is not feasible for this dataset, as only a subset of the answers is accessible. Therefore, only non-agentic and agentic solutions were included in the evaluation. Among the non-agentic baselines, the most relevant are SeViLa [46], InternVideo [35], and Vamos [33]. For agentic solutions, previously established baselines were used, including LLoVi [48], TraveLER [28], VideoAgent [34], ProViQ [4], JCEF, and MoReVQA [23].

C.1. ActivityNet-QA evaluation

ActivityNet-QA [47] open-ended benchmark provides only a single correct answer for each question, requiring LLM-based evaluation of responses, as shown in prior work

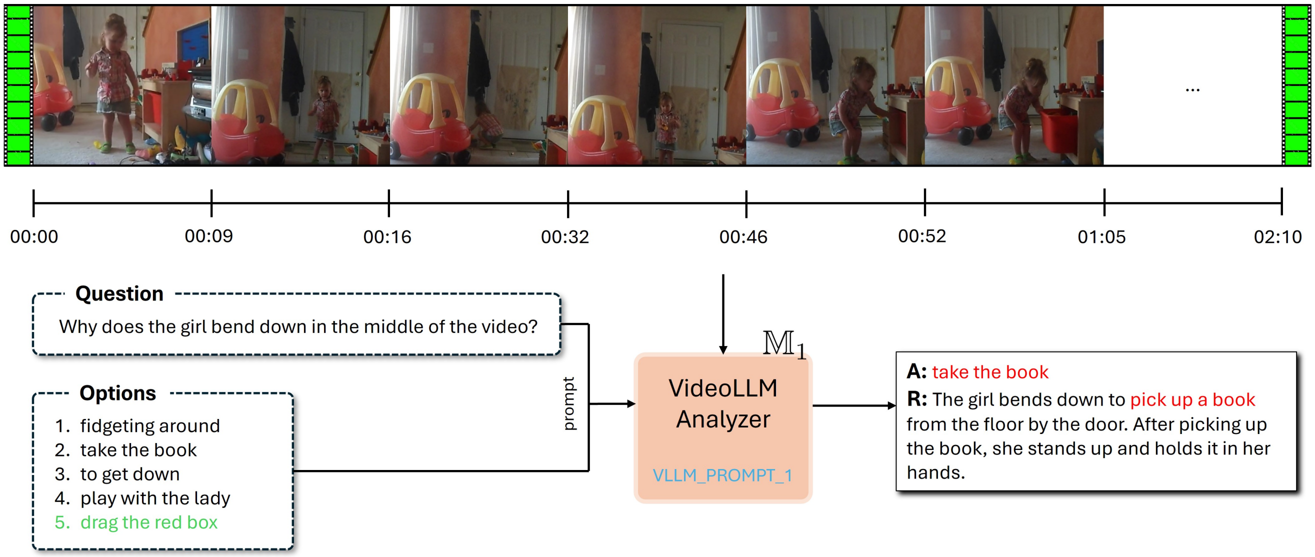
[21, 23, 32]. Notably, this process was not required for iVQA [43], which offers a set of potential answers. For this evaluation, we used the following prompt:

```
Evaluate whether the predicted answer
/reasoning are correct based on the real
answer to the question. Only output ’yes’
or ’no’, don’t provide an explanation.

Question: {q}
Real answer: {a}
Predicted answer: {p}
Predicted reasoning: {r}
Output (yes/no):
```

D. Additional Analysis

As demonstrated in the benchmark results, the ViQAgent framework shows significant promise for the Video Question Answering task and, in general, tool-using agents, establishing a new state-of-the-art by integrating VideoLLMs, LLMs, and computer vision foundation models to address questions based on video content. However, it is important to acknowledge that, as LLMs currently cannot be seeded, certain random factors in their output remain uncontrollable, even though all experiments were conducted with a temperature parameter set to 0.0. Furthermore, the ViQAgent framework proves most effective when the object or target of the query is visibly present in the video. In cases where the YOLO-World model cannot detect the object, confusion may arise. Nevertheless, the framework is robust enough to mitigate this confusion through the CoT judge module (M_2), resulting in improved performance while preserving the primary advantages of Zero-Shot VideoLLM-based question answering.



Question

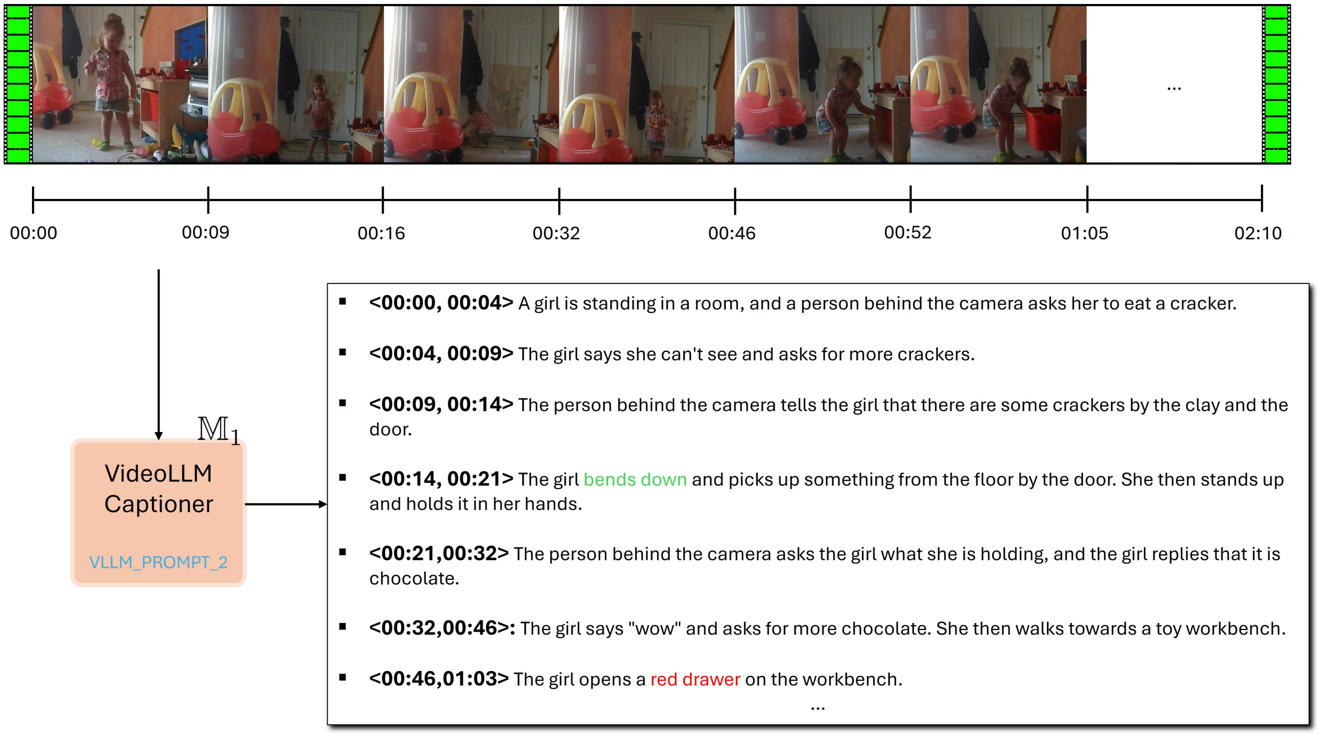
Why does the girl bend down in the middle of the video?

Options

1. fidgeting around
2. take the book
3. to get down
4. play with the lady
5. drag the red box

A: take the book
R: The girl bends down to pick up a book from the floor by the door. After picking up the book, she stands up and holds it in her hands.

Figure 4. **VideoLLM Analyzer** (VideoLLM$_1$): Given the full video, and the ”prompt” (question plus answer options, if available), the VideoLLM Analyzer submodule provides a first-sight response with a reasoning text of why that answer is correct.



- **<00:00, 00:04>** A girl is standing in a room, and a person behind the camera asks her to eat a cracker.
- **<00:04, 00:09>** The girl says she can't see and asks for more crackers.
- **<00:09, 00:14>** The person behind the camera tells the girl that there are some crackers by the clay and the door.
- **<00:14, 00:21>** The girl bends down and picks up something from the floor by the door. She then stands up and holds it in her hands.
- **<00:21,00:32>** The person behind the camera asks the girl what she is holding, and the girl replies that it is chocolate.
- **<00:32,00:46>:** The girl says "wow" and asks for more chocolate. She then walks towards a toy workbench.
- **<00:46,01:03>** The girl opens a red drawer on the workbench.

...

Figure 5. **VideoLLM Captioner** (VideoLLM$_2$): Given the full video, but not the question (to avoid bias), the VideoLLM Captioner submodule provides a comprehensive set of event-separated timeframes with a description (*i.e*. caption) of what is happening in every part of the video. This is the first grounding output, used then for comparison against YOLO-World object grounding.

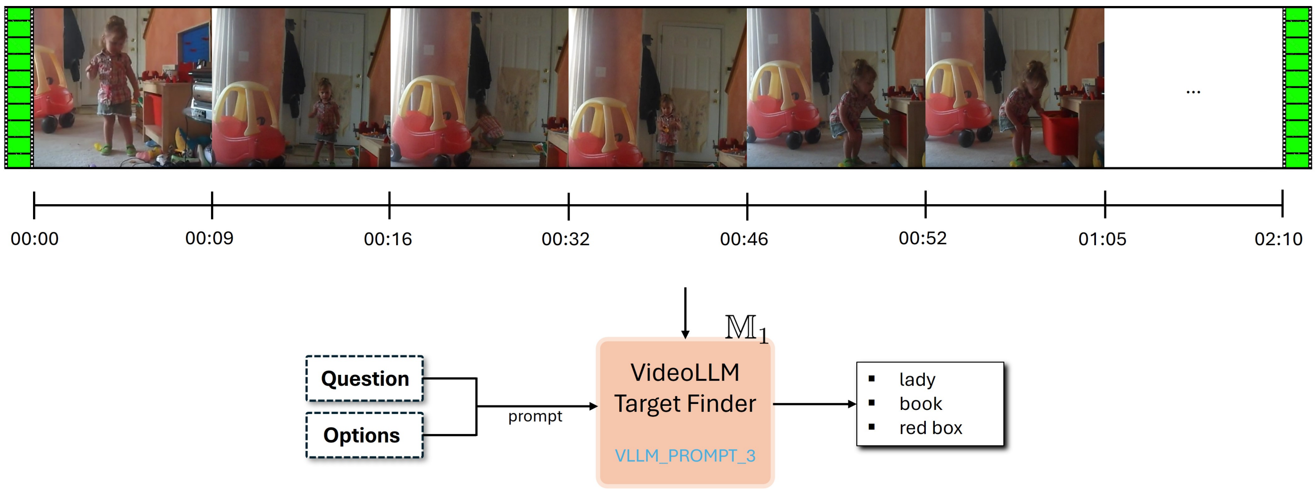
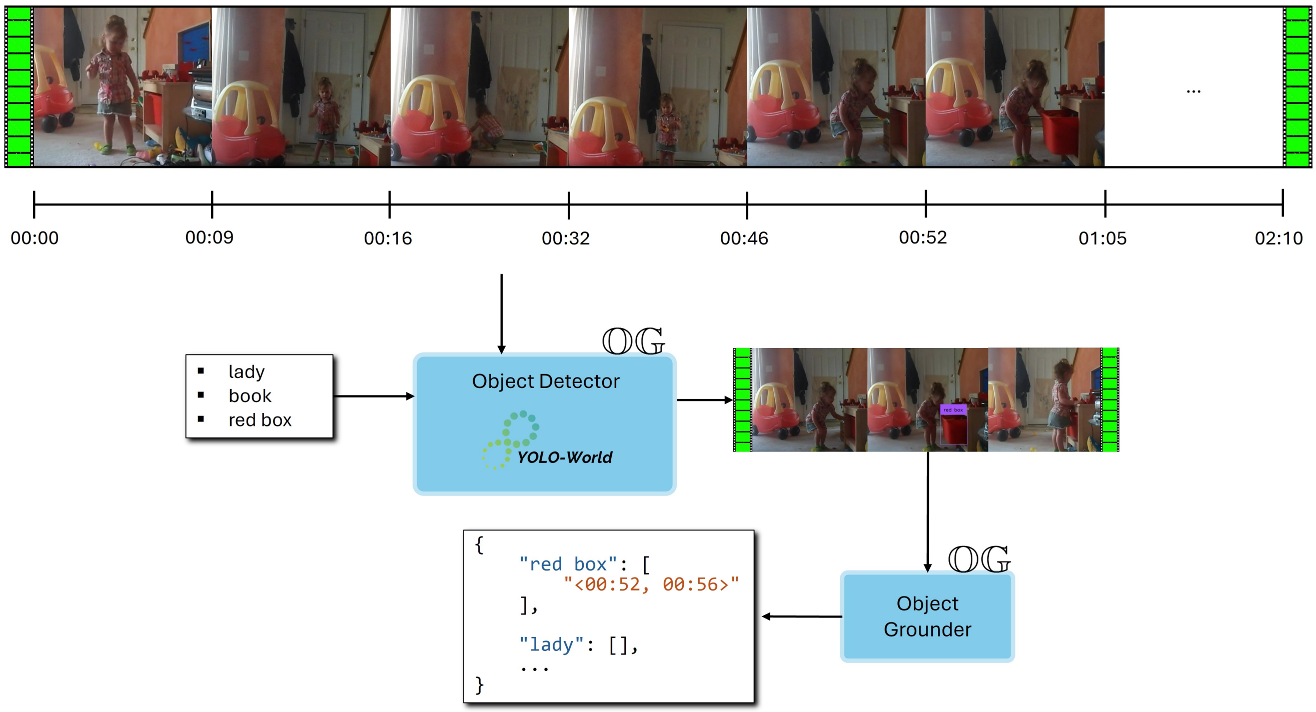
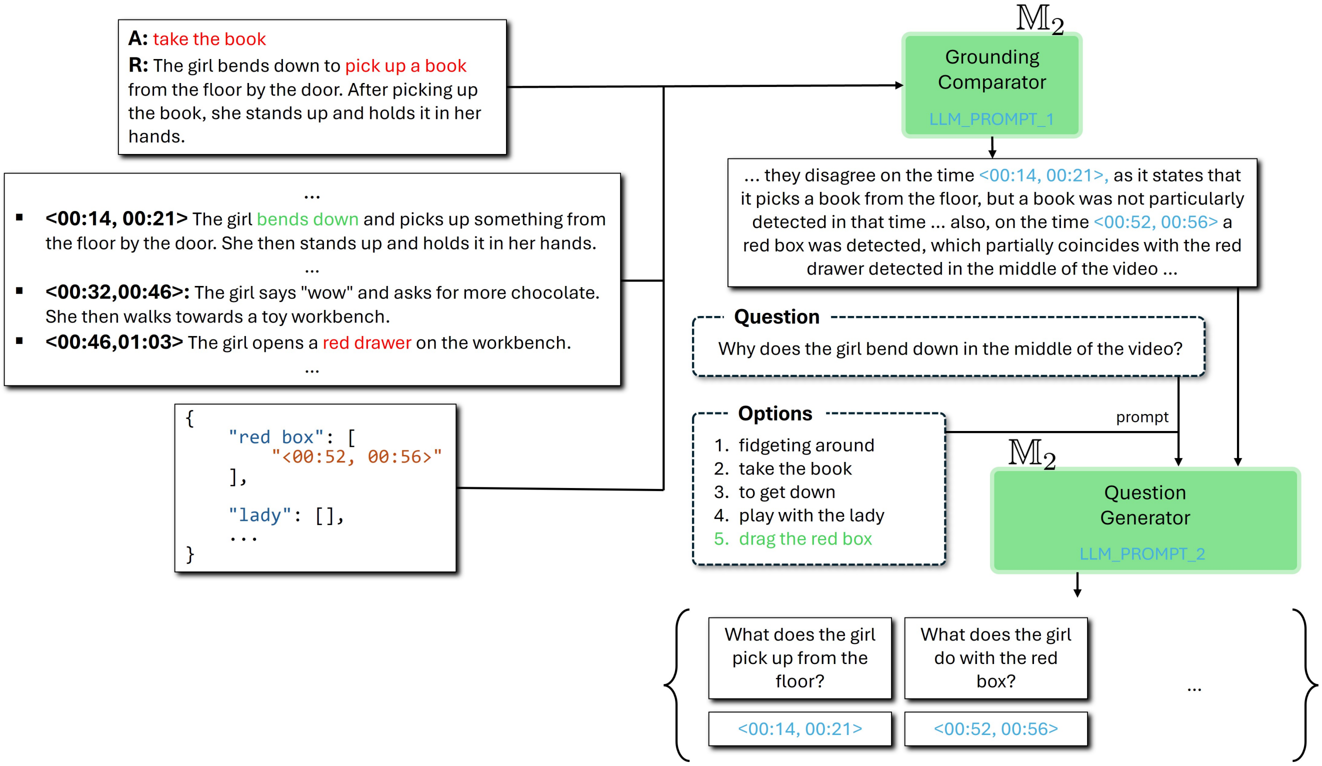


Figure 6. **VideoLLM Target Finder** (VideoLLM$_3$): Given the full video, and the ”prompt” (question plus answer options, if available), the VideoLLM Target Finder submodule is very straightforward and simple, yet effective on finding up to 4 relevant targets to identify in the video with the YOLO-World model. These objects/targets are selected based on both the video content and the relevant targets mentioned in the questions and answers.



```
{
    "red box": [
        "<00:52, 00:56>"
    ],
    "lady": [],
    ...
}
```

Figure 7. (*OG* Module): First, given the full video and the selected targets, the **Object Detector Model** (YOLO-World) extracts the detections from the targets in the video, which are then passed to the **Object Grounder** to consolidate into a dictionary that contains all the timeframes in which each target is detected.



A: take the book
R: The girl bends down to pick up a book from the floor by the door. After picking up the book, she stands up and holds it in her hands.

...
- **<00:14, 00:21>** The girl bends down and picks up something from the floor by the door. She then stands up and holds it in her hands.

...
- **<00:32,00:46>:** The girl says "wow" and asks for more chocolate. She then walks towards a toy workbench.
- **<00:46,01:03>** The girl opens a red drawer on the workbench.

...

```
{
    "red box": [
        "<00:52, 00:56>"
    ],
    "lady": [],
    ...
}
```

... they disagree on the time <00:14, 00:21>, as it states that it picks a book from the floor, but a book was not particularly detected in that time ... also, on the time <00:52, 00:56> a red box was detected, which partially coincides with the red drawer detected in the middle of the video ...

Question

Why does the girl bend down in the middle of the video?

Options

1. fidgeting around
2. take the book
3. to get down
4. play with the lady
5. drag the red box

What does the girl pick up from the floor?

<00:14, 00:21>

What does the girl do with the red box?

<00:52, 00:56>

...

Figure 8. **LLM Grounding Comparator** (LLM$_1$): Given the grounding output from the captioning submodule (VideoLLM$_2$), the object grounding, and the first-sight reasoning (VideoLLM$_1$), the grounding comparator determines whether there are inconsistencies or uncertain parts within them, determines, and explains in which parts are these inconsistencies, which are then fed to the **Question Generator** (LLM$_2$) along with the prompt (original question plus answer options, if available), to state up to 3 clean questions and their doubtful timeframes, to be then asked to a VideoLLM.

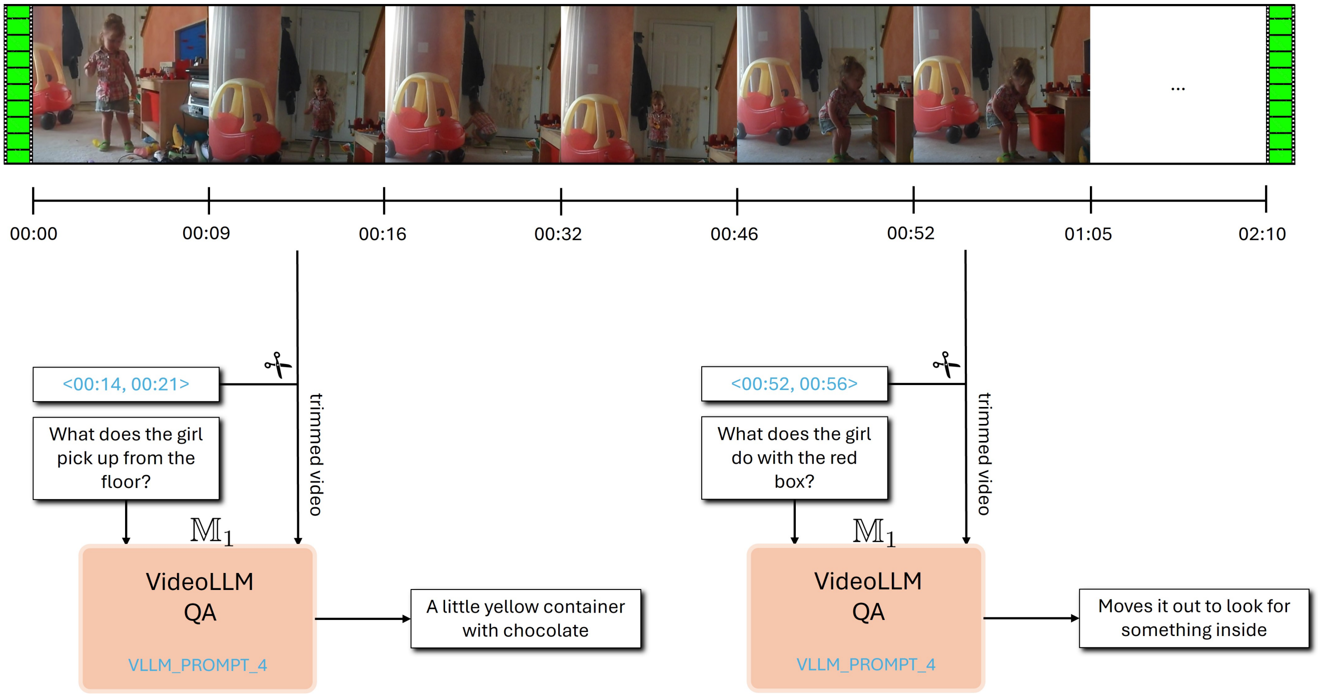
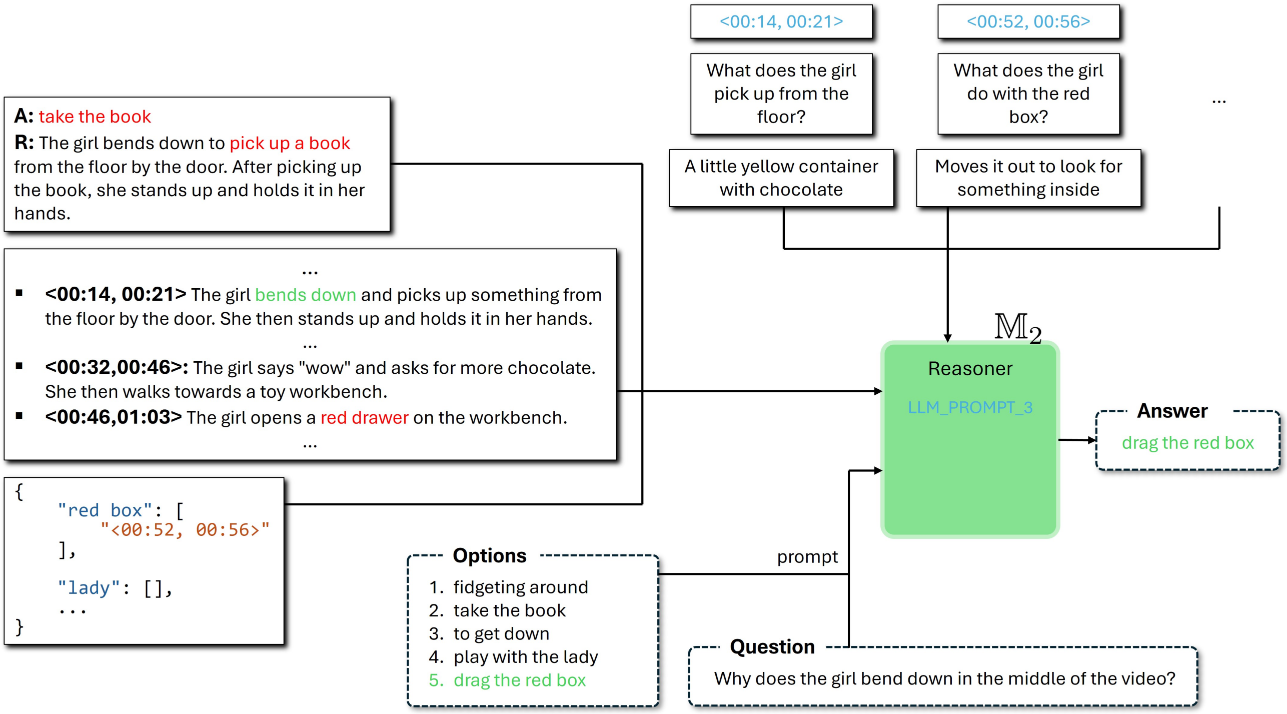


Figure 9. The stated questions from the previous step are then fed to the **VideoLLM QA** (VideoLLM$_4$) submodule to simply answer them.


A: take the book
R: The girl bends down to pick up a book from the floor by the door. After picking up the book, she stands up and holds it in her hands.

...
- **<00:14, 00:21>** The girl bends down and picks up something from the floor by the door. She then stands up and holds it in her hands.
...
- **<00:32,00:46>:** The girl says "wow" and asks for more chocolate. She then walks towards a toy workbench.
- **<00:46,01:03>** The girl opens a red drawer on the workbench.
...

```
{
    "red box": [
        "<00:52, 00:56>"
    ],
    "lady": [],
    ...
}
```

Options
1. fidgeting around
2. take the book
3. to get down
4. play with the lady
5. drag the red box

Question
Why does the girl bend down in the middle of the video?

<00:14, 00:21>
What does the girl pick up from the floor?
A little yellow container with chocolate
<00:52, 00:56>
What does the girl do with the red box?
Moves it out to look for something inside
...
$\mathbb{M}_2$
Reasoner
LLM_PROMPT_3
prompt
Answer
drag the red box

Figure 10. Once all the intermediate outputs are generated, they are fed to a **Final Reasoner** (LLM$_3$) submodule that answers the question with the new information.

Table 5. VideoLLM Analyzer Prompt (VideoLLM$_1$)

Prompt: VideoLLM Analyzer

```
VLLM_PROMPT_1 = """
Based on the provided video, select or provide the correct answer for the user
question. Break down your reasoning into clear, logical steps, and arrive at
the most accurate answer.

To ensure accuracy, follow this step-by-step reasoning process:
1. Restate or reframe the question for clarity.
2. Consider key events, actions, or objects relevant to the question.
3. If answer options are provided, assess each option in relation to the
video’s content. If no options are given, logically derive an answer.
4. Provide a clear and concise response based on your reasoning.

You must provide the index of the selected answer or the answer itself, and a
brief explanation of your reasoning.

"""
```

Table 6. VideoLLM Analyzer Schema (VideoLLM$_1$)

Output Schema for VideoLLM Analyzer

```
VLLM_SCHEMA_1 = {
    "type": "object",
    "properties": {
        "reasoning": { "type": "string" },
        "answer": { "type": "string" }
    }
}
```

Table 7. VideoLLM Captioner Prompt (VideoLLM$_2$)

Prompt: VideoLLM Captioner

```
VLLM_PROMPT_2 = """
Based on the provided video and the given question (and answer options if
available), capture a list of the main timeframes in the video in the format
<<mm0:ss0,mm1:ss1>>: {description}, where ’description’ is a detailed
description of what is happening in that particular timeframe.

Follow these steps to generate your response:
1. Carefully analyze the question and the video content to identify the key
events or actions that are relevant to the question.
2. Identify key events, actions, or transitions that represent meaningful
changes or notable moments in the video.
3. Break the video into distinct timeframes where these events occur.
4. For each identified timeframe, provide a clear, detailed description of the
action or scene in that segment.
5. Ensure that each description is specific, concise, and accurately reflects
the action within the timeframe.
"""
```

Table 8. VideoLLM Captioner Schema (VideoLLM$_2$)

Output Schema for VideoLLM Captioner

```
VLLM_SCHEMA_2 = {
    "type": "object",
    "properties": {
        "timeframes": {
            "type": "array",
            "items": { "type": "string" }
        }
    }
}
```

Table 9. VideoLLM Target Finder Prompt (VideoLLM$_3$)

Prompt: VideoLLM Target Finder

```
VLLM_PROMPT_3 = """
Based on the provided video and the given question (and answer options if
available), your task is to capture a **list of objects/targets** that are
involved in the video and are relevant to the question. These targets will
be used for object detection and grounding via a YOLO model. Please follow
these steps:

1. Understand the question and its context within the video, along with any
answer options provided.
2. Focus on the most relevant objects or targets that are involved in the
video’s key actions or scenes. Ensure that these targets directly relate to
the question.
3. Choose no more than 4 targets, ideally 3 or fewer. Consider only the
objects that are clearly present and essential to answering the question,
and that are not too complex to identify (not too large as well), but not
too general for the particular video.
4. Ensure that the targets are also directly related to the answer options,
if provided.
5. Provide a short list of targets, ensuring each description is clear and
relevant (e.g., ’player in white outfit’, ’spoon’, etc.).
"""
```

Table 10. VideoLLM Target Finder Schema (VideoLLM$_3$)

Output Schema for VideoLLM Target Finder

```
VLLM_SCHEMA_3 = {
    "type": "object",
    "properties": {
        "targets": {
            "type": "array",
            "items": { "type": "string" }
        }
    }
}
```

Table 11. VideoLLM QA Prompt (VideoLLM$_4$)

Prompt: VideoLLM QA

```
VLLM_PROMPT_4 = """
Based on the provided video, answer the user question in the VERY SPECIFIC
given timeframe.

Only provide the final, concise answer, directly related to the question.
Base your answer ONLY on the information in the video, and do not add any
information. If the answer is not present in the video, state ’unanswerable’.
For example, if the question is ’What color is the car?’, and the car is not
shown in the video timeframe, the answer should be ’unanswerable’.
"""
```

Table 12. VideoLLM QA Schema (VideoLLM$_4$)

Output Schema for VideoLLM QA

```
VLLM_SCHEMA_4 = {
    "type": "object",
    "properties": {
        "answer": { "type": "string" }
    }
}
```

Table 13. LLM Grounding Comparator Prompt (LLM$_1$)

Prompt: LLM Grounding Comparator

```
LLM_PROMPT_1 = """
You will be provided with reasoning for an answer to a question, along with
two grounding pieces of information:
1. **VideoLLM-extracted grounding captions**: These describe the key events
and timeframes within the video (e.g., <<mm0:ss0,mm1:ss1>>: {description}).
2. **YOLO object grounding**: This identifies the specific objects/targets
and their appearances in different video timeframes.

Your task is to analyze if there is any disagreement between the grounding
information (both the captions and object grounding) and the reasoning for
the answer. Disagreements may occur if the reasoning implies events or objects
appearing in timeframes that are inconsistent with the grounding.

Please output a "disagree" boolean indicating if there is any disagreement at
all, and a detailed but concise explanation of the specific timeframes where
the grounding information does not align with the reasoning. Only include
timeframes where discrepancies occur, and keep the explanation short but clear.
If no disagreement is found, simply explain that there is no disagreement.

Disagreements should be highlighted by timeframe (<<mm0:ss0,mm1:ss1>>) and why
the reasoning conflicts with the provided grounding information.
"""
```

Table 14. LLM Grounding Comparator Schema (LLM$_1$)

Output Schema for LLM Grounding Comparator

```
LLM_SCHEMA_1 = {
    "type": "object",
    "properties": {
        "reasoning": { "type": "string" },
        "disagree": { "type": "boolean" }
    }
}
```

Table 15. LLM Question Generator Prompt (LLM$_2$)

Prompt: LLM Question Generator

```
LLM_PROMPT_2 = """
You will be provided the following:
1. A question (and answer options if available) related to a video.
2. A text explaining the set of discrepancies found in previous studies of the
video. These indicate specific timeframes in the video where the grounding
information does not align with the reasoning. These timeframes and the reasons
for the discrepancies are provided.

Your task is to generate a set of up to 3 concise questions to ask a VideoLLM
to clarify and provide a more grounded, precise answer. The goal is to resolve
the discrepancies and improve the grounding for the question at hand.

- Each question should focus on a specific timeframe where a discrepancy was
found.
- Each question should be concise and relevant to the timeframe, and
particularly relevant to answer the question.
- Ensure that each question includes the timeframe where the clarification
is needed, formatted as <<mm0:ss0,mm1:ss1>>.
- The timeframe must be very precise in time, covering only the specific
segment where the discrepancy occurred.
- Do not include any unnecessary details, just the specific query for
clarification.
- If there are not CONSIDERABLE discrepancies, you may return an empty list!

Generate between 0 and up to 3 questions based on the discrepancies identified.
"""
```

Table 16. LLM Question Generator Schema (LLM$_2$)

Output Schema for LLM Question Generator

```
LLM_SCHEMA_2 = {
    "type": "object",
    "properties": {
        "questions": {
            "type": "array",
            "items": { "type": "string" }
        }
    }
}
```

Table 17. LLM Final Reasoner Prompt (LLM$_3$)

Prompt: LLM Final Reasoner

```
LLM_PROMPT_3 = """
You will be provided the following:
1. A question (and answer options if available) related to a video.
2. An initial reasoning made for a possible answer, along with an
explanation of why it was chosen. This reasoning was done BEFORE
knowing the grounding information, and clarification questions.
3. The **grounding information**:
    - **VideoLLM grounding**: Timeframes and event descriptions from the
    video.
    - **YOLO object grounding**: Objects/targets identified in the video
    and their corresponding appearing timeframes.
4. A set of clarification questions asked about discrepancies in the
grounding, and their responses.

Your task is to:
1. Analyze all the provided information and reasoning.
2. Select or provide the correct answer for the user question, based on the
new clarifications from the questions and grounding data.
3. Provide the final, most accurate specific answer, as well as a reasoning
for it.

Remember to stick to the information provided, and ensure that your answer
is accurate and well-supported by the grounding information and reasoning
provided. If none of the answer options are correct, select the most
appropiate based on the new information and reasoning.

"""
```

Table 18. LLM Final Reasoner Schema (LLM$_3$)

Output Schema for LLM Final Reasoner

```
LLM_SCHEMA_3 = {
    "type": "object",
    "properties": {
        "reasoning": { "type": "string" },
        "answer": { "type": "string" }
    }
}
```