UpstreamQA: A Modular Framework for Explicit Reasoning on Video Question Answering Tasks

Jason Nguyen
Lincoln North Star High School

Ameet Rao
The Charter School of Wilmington

Alexander Chang
Greenwich High School

Ishaan Kumar
Santa Susana High School

Erin Tan
UC Berkeley

Abstract

Video Question Answering (VideoQA) demands models that jointly reason over spatial, temporal, and linguistic cues. However, the task's inherent complexity often requires multi-step reasoning that current large multimodal models (LMMs) perform implicitly, leaving their internal decision process opaque. In contrast, large reasoning models (LRMs) explicitly generate intermediate logical steps that enhance interpretability and can improve multi-hop reasoning accuracy. Yet, these models are not designed for native video understanding, as they typically rely on static frame sampling. We propose UpstreamQA, a modular framework that disentangles and evaluates core video reasoning components through explicit upstream reasoning modules. Specifically, we employ multimodal LRMs to perform object identification and scene context generation before passing enriched reasoning traces to downstream LMMs for VideoQA. We evaluate UpstreamQA on the OpenEQA and NExTQA datasets using two LRMs (o4-mini, Gemini 2.5 Pro) and two LMMs (GPT-4o, Gemini 2.5 Flash). Our results demonstrate that introducing explicit reasoning can significantly boost performance and interpretability of downstream VideoQA, but can also lead to performance degradation when baseline performance is sufficiently high. Overall, UpstreamQA offers a principled framework for combining explicit reasoning and multimodal understanding, advancing both performance and diagnostic transparency in VideoQA in several scenarios.

1 Introduction

Video Question Answering (VideoQA) is a challenging task that requires large multimodal models (LMMs) to have a comprehensive understanding of video inputs and be capable of answering various questions about them [1]. These models must be capable of inferring semantic, spatial, temporal, and causal relationships between different entities in the video.

Despite significant advancements, LMMs continue to face substantial limitations regarding weak grounding of question-specific frames, high sensitivity to adversarial perturbations, and disproportionate over-reliance on certain modalities when performing VideoQA tasks [2, 3]. Traditionally, VideoQA has relied on end-to-end architectures, however their black-boxed nature hinders the transparency of their internal reasoning processes [4].

39th Conference on Neural Information Processing Systems (NeurIPS 2025) Workshop: Vision Language Models: Challenges of RealWorld Deployment.

Recently, a new type of model has been introduced: Large Reasoning Models (LRMs). These models rely on System 2 thinking—logical, deliberate decision-making—as opposed to the quick, intuitive reasoning of System 1 thinking [5]. LRMs can be leveraged in VideoQA to improve accuracy, as they are capable of facilitating intermediate reasoning of questions and incorporating a deeper understanding of spatial, temporal, and causal relationships of video content.

We perform an exploratory study of how explicit reasoning influences VideoQA performance. We introduce a framework for evaluating various upstream tasks processed by LRMs can influence the downstream VideoQA performance. Concretely, our contributions are as follows:

1. We introduce UpstreamQA, a novel framework for evaluating explicit reasoning as upstream modules for VideoQA, providing more insight on the intermediate processes in such a complex task.
2. We perform experiments across two upstream tasks—object identification and scene context generation—as well as two LRMs and two LMMs, reporting results of their effect on VideoQA performance.
3. We find that although explicit reasoning improves interpretability of logical decision making processes, performance differences are model- and dataset-dependent.

2 Related Works

In recent years, Large Multimodal Models (LMMs) that effectively combine visual and linguistic data have driven significant improvements on VideoQA benchmarks [6]. SOTA approaches on these benchmarks improve factual grounding by incorporating strategies such as Retrieval-Augmented Generation (RAG) [7, 8]. However, these systems are commonly trained end-to-end and process the entire task in one continuous pipeline, with no separation between different stages like retrieval, reasoning, and answer generation. This often makes it difficult to analyze why a model fails or succeeds.

Recently, many approaches have worked to modularize frameworks, including TraveLER [9] and LLoVi [10], in an attempt to decompose the complex task of VideoQA into several smaller subtasks. These approaches integrate auxiliary large language models (LLMs) to perform tasks such as refining the question or captioning frames [11, 10, 4]. ENTER is another modular framework that demonstrates better interpretability in the reasoning process through generated event graphs [12]. MoReVQA introduces a multi-stage system which produces intermediate outputs applied to specific tasks at each stage [4].

Our framework takes a similar modularized approach to evaluate explicit reasoning on particular upstream tasks. We simplify these multi-stage frameworks into a linear, two-stage process in order to better isolate the effect of the reasoning module on downstream VideoQA performance. Our approach is also designed to be task-agnostic, where the drop-in reasoning module can be evaluated over any upstream reasoning task.

2.1 Explicit Reasoning in LLMs

LRMs are a class of LLMs designed to perform multi-step reasoning rather than directly producing answers [5]. These models output intermediate logical reasoning steps to make their thinking process more transparent. Current state-of-the-art (SOTA) LRMs include OpenAI's GPT-5, Google DeepMind's Gemini 2.5 Pro, and Anthropic's Claude Opus 4.1 [13, 14, 15].

Recent works demonstrate that enabling explicit reasoning via chain-of-thought, tree-of-thought, or reinforcement learning-based reasoning, substantially boosts task performance and interpretability [16]. This is largely because introducing explicit reasoning facilitates stronger factual grounding and reduces hallucinations. ReasVQA uses a pipeline that generates explicit reasoning, refines the reasoning, then learns from it [17]. They find that noisy or incorrect reasoning hinders performance. Similarly, Zang et al. [18] focus on causal reasoning, removing confounders (irrelevant information) by modeling causal graphs. UpstreamQA presents a novel evaluation framework to evaluate explicit reasoning for VideoQA, which, to the best of our knowledge, has not been done.

3 Methods

Our method follows a two-stage pipeline. First, we employ reasoning modules to perform distinct upstream video analysis tasks. Specifically, we input 50 uniformly-sampled frames along with an upstream task objective into a multimodal LRM to generate a structured reasoning output. Second, the output is provided to an LMM, that performs the equivalent base VideoQA task but with additional upstream reasoning. We then evaluate the effect of upstream reasoning modules on VideoQA performance using metrics based on accuracy. In this work, we focus specifically on object identification and scene context generation as the upstream tasks, although our framework itself is designed to be task-agnostic. The object identification upstream task focuses on generating a structured inventory of the objects, their attributes (e.g., color, material, etc.), and spatial relationships with one another in a given video [19]. Additionally, the scene context generation upstream task is aimed at recognizing the overall scene category (e.g., kitchen) and generating a comprehensive description of the environment (e.g., environmental details, ambiance, etc.) [20, 21].

The outputs of these subtasks are then passed to an LMM along with the original video-question pair, producing our final VideoQA output. Figure 1 illustrates the entire pipeline.

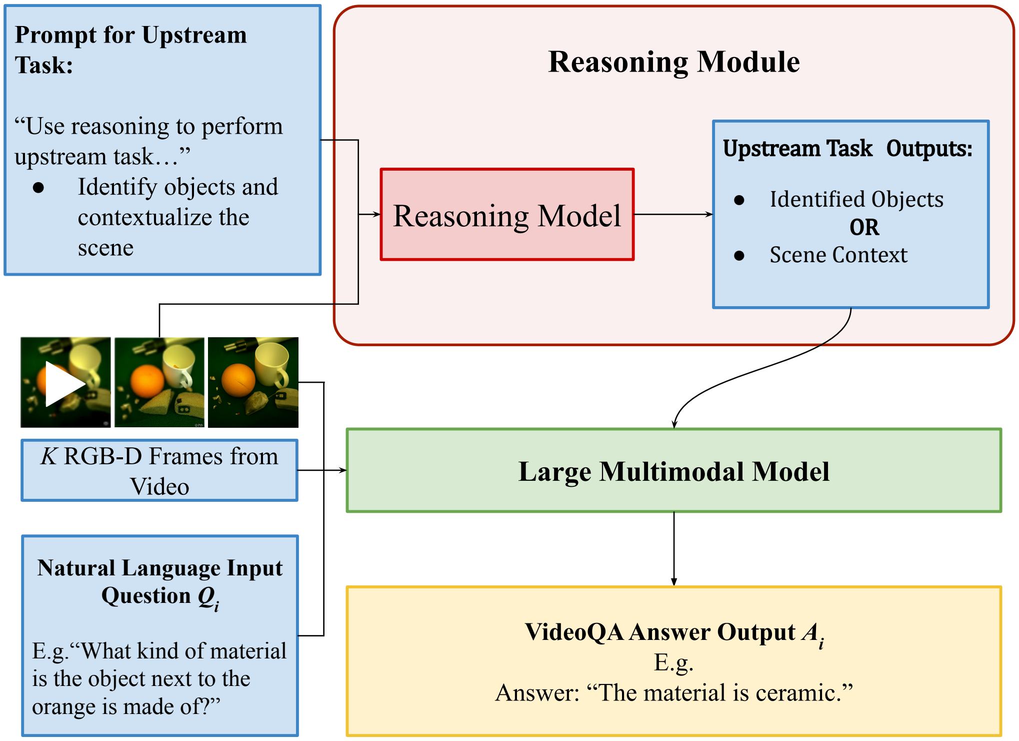

Prompt for Upstream Task:
"Use reasoning to perform upstream task…"
Identify objects and contextualize the scene
Reasoning Module
Reasoning Model
Upstream Task Outputs:
Identified Objects
OR
Scene Context
K RGB-D Frames from Video
Large Multimodal Model
Natural Language Input Question Q_i
E.g."What kind of material is the object next to the orange is made of?"
VideoQA Answer Output A_i
E.g.
Answer: "The material is ceramic."

Figure 1: **Overview of our OpenEQA and NExTQA framework**. An explicit reasoning model is used to perform a specific upstream task, and the output is passed to the LMM along with the original video and question/answers.

3.1 Datasets

Our experiments are conducted across two datasets: NExTQA and OpenEQA [22, 23]. Both datasets are preprocessed by uniformly sampling 50 frames from each video in the dataset, paired alongside the audio data for the video [24]. The LMM receives a combined input of the sampled frames, audio data, the output from the processed upstream task, and a natural language question prompt. We utilize zero-shot prompting to evaluate the models. All prompts are provided in Section B of the Appendix.

NExTQA NExTQA is a VideoQA dataset containing 5,440 videos and around 52,000 manually annotated question-answer pairs in the form of either multiple-choice QA or open-ended QA. Videos in the dataset show object interactions in daily life, taken from the VidOR dataset [25, 26].

In this work, we experiment on only the multiple-choice subset, and evaluate performance using accuracy (percentage of correct answers selected). Due to computational constraints, we limited the question set to 2,500 questions, each of which correspond to one of the 298 videos containing the fewest frames in the NExTQA dataset [22]. All of the questions from the filtered dataset are between 10-20 seconds long.

OpenEQA OpenEQA is a modern dataset for the task of Embodied Question Answering (EQA), where an agent must navigate a 3D environment and answer natural language questions about it [23, 27].

In this work, we specifically use the Episodic Memory EQA (EM-EQA) variant of OpenEQA, which uses episode histories collected from two real-world sources: HM3D and ScanNet [28, 29]. EM-EQA presents pre-recorded first-person walkthroughs of various environments, which more directly align with a standard VideoQA task, as they consist of continuous sequences that can be processed as standard video input.

In OpenEQA, videos of the given environments are captured in RGB-D frames, allowing for spatial memory. Included are 1,636 human-generated questions regarding 187 scenes based on real-world environments.

We utilize the same evaluation method and correctness metric (LLM-Match) introduced by OpenEQA [23]. To evaluate outputs, an independent LLM (GPT-4) is used to score outputted answers compared to the ground truth [30]. Scores are aggregated using the following formula:

$$C = \frac{1}{N} \sum_{i=1}^{N} \frac{\sigma_i - 1}{4} \times 100\% \tag{1}$$

Given a question Q_i, a human-annotated answer A_i^*, and model output A_i, the LLM is prompted to provide a score $\sigma_i \in \{1, \ldots, 5\}$. On this scale, a score of 1 denotes an incorrect response, while a score of 5 denotes a correct response. Any intermediate value represents similarity to the upper or lower bound of the scale. This process continues for N questions, where N represents the total number of questions.

3.2 Experiments

Our experiments are conducted across two different LRMs, two LMMs, and two distinct VideoQA datasets. All models are used off-the-shelf with no additional training or fine-tuning.

Baselines We evaluated baselines by using standalone LMMs. The models used were GPT-4o and Gemini 2.5 Flash [31, 14]. Both models take text, images, and audio inputs, while Gemini 2.5 Flash additionally supports video input natively. The baselines are evaluated on “vanilla" VideoQA, meaning no upstream task outputs are provided. All prompts are provided in section B of the Appendix.

Experiments We evaluated the addition of explicit reasoning modules by introducing them along with the base LMMs to help with upstream tasks. The reasoning modules used were o4-mini—a smaller, lightweight LRM—and Gemini 2.5 Pro, which is a larger, more powerful LRM [32, 14]. LRMs were evaluated on different upstream tasks and their contribution to performance on downstream VideoQA tasks, being used as a “drop in" module to assist the LMM. Essentially, additional context is passed into the baseline, providing more information to answer the questions effectively.

4 Results

Our results show that the effect of upstream reasoning on downstream VideoQA is dependent on the dataset and base model used. Table 1 reports the overall results from our experiments across all LRM and base model combinations, as well as both upstream tasks.

LMM	LRM	OpenEQA	NExTQA
GPT-4o	———	67.7	62.32%
Gemini 2.5 Flash	———	58.8	78.32%
Object Identification			
GPT-4o	o4-mini	55.7	**67.48%**
GPT-4o	Gemini 2.5 Pro	59.7	**67.08%**
Gemini 2.5 Flash	o4-mini	**63.6**	77.44%
Gemini 2.5 Flash	Gemini 2.5 Pro	**67.1**	78.00%
Scene Context			
GPT-4o	o4-mini	48.1	**67.68%**
GPT-4o	Gemini 2.5 Pro	47.8	**64.96%**
Gemini 2.5 Flash	o4-mini	**66.7**	77.20%
Gemini 2.5 Flash	Gemini 2.5 Pro	**67.8**	77.16%

Table 1: **Results on the OpenEQA and NExTQA datasets with distinct LMM and LRM pairs**. The LRMs perform an upstream task of either object recognition or scene context. Performance on OpenEQA is presented through an LLM-Match score, and performance on NExTQA is evaluated by percentage of correct answers. Scores for OpenEQA and NExTQA are provided in the table, with notable score improvements shown in bold.

On OpenEQA, we find that introducing explicit reasoning significantly improves performance when Gemini 2.5 Flash is used as the base LMM. Specifically, Gemini 2.5 Flash achieves a baseline overall LLM-Match Score of 58.8, which increases to 67.1 and 67.8 when using Gemini 2.5 Flash for object identification and scene context generation, respectively. We observe comparable improvements when using o4-mini as the LRM.

Interestingly, GPT-4o—which has an estimated 1.8 trillion parameters—scored significantly worse than the baseline when combined with an LRM [33]. GPT-4o already exhibited strong performance on standard VideoQA, outscoring Gemini 2.5 Flash by 8.9 points on the baseline.

In NExTQA, we observe performance gains from upstream reasoning in GPT-4o, but not for Gemini 2.5 Flash. GPT-4o achieves a baseline accuracy of 62.32%, and combining GPT-4o with o4-mini for the object identification upstream task increases the accuracy to 67.48%. Using Gemini 2.5 Pro as the reasoning model for the same object identification upstream task increases accuracy to 67.08%. For scene context generation, GPT-4o and o4-mini combined achieve an improved accuracy of 67.68% (+5.36%).

In contrast, Gemini 2.5 Flash begins with a baseline accuracy of 78.32%, much higher than GPT-4o. Combining with the reasoning models for object identification leads to slight degradation in overall accuracy, to 77.44% for o4-mini (-0.88% accuracy) and 78.00% for Gemini 2.5 Pro (-0.32%).

These results over both datasets show that introducing upstream explicit reasoning does not always improve downstream VideoQA performance, and that the impact of reasoning is primarily dependent on the dataset and choice of base LMM. When baseline task performance is sufficiently high, we find that introducing explicit reasoning is not always helpful and can even lead to performance degradation.

4.1 Impacts by Question Type

In this section, we present additional analysis on OpenEQA to evaluate the performance of the LRMs on their assigned upstream task [23]. In particular, we hypothesize that the performance of the LRM on the upstream task plays a critical role in determining downstream impacts.

We utilize the change in performance across two question categories—object recognition and world knowledge—as a proxy for the relative performance difference between the base LMM and the LRM on the relevant upstream task. OpenEQA does not provide ground-truth labels for object identification or scene context generation; instead, questions are categorized into seven distinct question types which can be used to evaluate a model's performance on specific facets of VideoQA.

Object recognition questions evaluate a model's ability to recognize objects that appear in the episodic history, whereas world knowledge questions assess a model's ability to leverage external background knowledge about the world [23]. We select object recognition and world knowledge because they most directly align with our upstream tasks of object identification and scene context generation, respectively. In our analysis, we stratify LLM-Match scores for the object recognition and world knowledge question types to better isolate the effects that reasoning modules have on OpenEQA performance.

Figure 2 reports the results stratified by question type for Gemini 2.5 Flash on OpenEQA. The full set of results for both models are presented in Table 2 in the Appendix. For object identification, we find that performance on object recognition questions improves significantly when reasoning is used. As a control, we also provide the change in scores for world knowledge questions when object identification is the upstream task and find that performance stays relatively stagnant. Curiously, we observe similar trends for scene context generation, wherein object recognition performance improves while world knowledge remains stagnant. These results suggest that the benefit of explicit upstream reasoning largely lies in its ability to provide factual grounding and not so much *understanding*; for more structured questions such as object recognition, reasoning is helpful, whereas it may not be as helpful for broader questions about world knowledge.

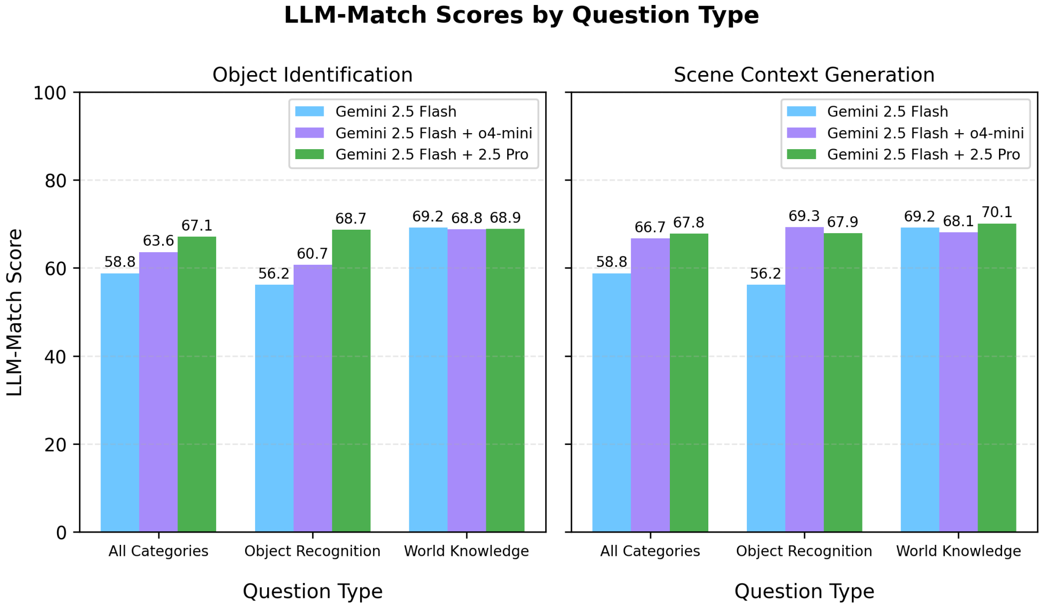


Figure 2: **LLM-Match Scores stratified by question category on OpenEQA using Gemini 2.5 Flash as the base model**. Results are reported for both upstream tasks: object identification (left) and scene context generation (right). Upstream reasoning improves overall accuracy and accuracy on object recognition questions while world knowledge question performance remains relatively stagnant.

5 Discussion

In this work, we introduce a novel two-stage framework for evaluating explicit reasoning in VideoQA across the NExTQA and OpenEQA datasets. The modularity of our framework allows for greater flexibility, allowing for experimentation with different model and task combinations. Furthermore, our framework also allows for greater interpretability of the models and by exposing intermediate reasoning steps rather than relying solely on end-to-end evaluation metrics.

The results of our experiments reveal the effect of our framework on VideoQA accuracy on certain tasks, while leading to performance degradation on others. This indicates that selective integration of modular reasoning via UpstreamQA can yield substantial improvements in task-specific VideoQA performance.

Limitations & Future Work Our work presents promising preliminary findings for better understanding the role of explicit reasoning models in improving complex tasks like VideoQA. Additional experiments are necessary to account for various confounding factors, such as upstream task performance. The scope of our evaluation was limited to zero-shot VideoQA, although comparisons to other methods such as chain-of-thought prompting are also necessary.

Future expansions of this work may consider exploring further usage of modular reasoning to encompass other core video reasoning components and their effect on VideoQA performance. Moreover, subsequent research may additionally investigate why performance degradation occurs on certain models while significant performance improvements are observed on others.

References

[1] Yaoyao Zhong, Wei Ji, Junbin Xiao, Yicong Li, Weihong Deng, and Tat-Seng Chua. Video question answering: Datasets, algorithms and challenges. In Yoav Goldberg, Zornitsa Kozareva, and Yue Zhang, editors, *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 6439–6455, Abu Dhabi, United Arab Emirates, December 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.emnlp-main.432. URL `https://aclanthology.org/2022.emnlp-main.432/`.

[2] 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, 2025. URL `https://arxiv.org/abs/2408.04223`.

[3] T. Winterbottom, S. Xiao, A. McLean, and N. Al Moubayed. On modality bias in the tvqa dataset. In *Proceedings of the British Machine Vision Conference (BMVC)*, 2020.

[4] Juhong Min, Shyamal Buch, Arsha Nagrani, Minsu Cho, and Cordelia Schmid. Morevqa: Exploring modular reasoning models for video question answering, 2025. URL `https://arxiv.org/abs/2404.06511`.

[5] Zhong-Zhi Li, Duzhen Zhang, Ming-Liang Zhang, Jiaxin Zhang, Zengyan Liu, Yuxuan Yao, Haotian Xu, Junhao Zheng, Pei-Jie Wang, Xiuyi Chen, Yingying Zhang, Fei Yin, Jiahua Dong, Zhiwei Li, Bao-Long Bi, Ling-Rui Mei, Junfeng Fang, Xiao Liang, Zhijiang Guo, Le Song, and Cheng-Lin Liu. From system 1 to system 2: A survey of reasoning large language models, 2025. URL `https://arxiv.org/abs/2502.17419`.

[6] Junxiao Xue, Quan Deng, Fei Yu, Yanhao Wang, Jun Wang, and Yuehua Li. Enhanced multimodal rag-llm for accurate visual question answering, 2024. URL `https://arxiv.org/abs/2412.20927`.

[7] Soyeong Jeong, Kangsan Kim, Jinheon Baek, and Sung Ju Hwang. Videorag: Retrieval-augmented generation over video corpus, 2025. URL `https://arxiv.org/abs/2501.05874`.

[8] Xubin Ren, Lingrui Xu, Long Xia, Shuaiqiang Wang, Dawei Yin, and Chao Huang. Videorag: Retrieval-augmented generation with extreme long-context videos, 2025. URL `https://arxiv.org/abs/2502.01549`.

[9] Chuyi Shang, Amos You, Sanjay Subramanian, Trevor Darrell, and Roei Herzig. Traveler: A modular multi-lmm agent framework for video question-answering, 2024. URL `https://arxiv.org/abs/2404.01476`.

[10] 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. URL `https://arxiv.org/abs/2312.17235`.

[11] Xinxin Dong, Baoyun Peng, Haokai Ma, Yufei Wang, Zixuan Dong, Fei Hu, and Xiaodong Wang. Leadqa: Llm-driven context-aware temporal grounding for video question answering, 2025. URL `https://arxiv.org/abs/2507.14784`.

[12] Hammad Ayyubi, Junzhang Liu, Ali Asgarov, Zaber Ibn Abdul Hakim, Najibul Haque Sarker, Zhecan Wang, Chia-Wei Tang, Hani Alomari, Md. Atabuzzaman, Xudong Lin, Naveen Reddy Dyava, Shih-Fu Chang, and Chris Thomas. Enter: Event based interpretable reasoning for videoqa, 2025. URL `https://arxiv.org/abs/2501.14194`.

[13] OpenAI. Gpt-5 system card. `https://cdn.openai.com/gpt-5-system-card.pdf`, August 2025.

[14] Google DeepMind. Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities, 2025. URL `https://arxiv.org/abs/2507.06261`.

[15] Anthropic. System card addendum: Claude opus 4.1. Technical report, Anthropic, August 2025. URL `https://assets.anthropic.com/m/4c024b86c698d3d4/original/Claude-4-1-System-Card.pdf`.

[16] Fengli Xu, Qianyue Hao, Zefang Zong, Jingwei Wang, Yunke Zhang, Jingyi Wang, Xiaochong Lan, Jiahui Gong, Tianjian Ouyang, Fanjin Meng, Chenyang Shao, Yuwei Yan, Qinglong Yang, Yiwen Song, Sijian Ren, Xinyuan Hu, Yu Li, Jie Feng, Chen Gao, and Yong Li. Towards large reasoning models: A survey of reinforced reasoning with large language models, 2025. URL `https://arxiv.org/abs/2501.09686`.

[17] Jianxin Liang, Xiaojun Meng, Huishuai Zhang, Yueqian Wang, Jiansheng Wei, and Dongyan Zhao. Reasvqa: Advancing videoqa with imperfect reasoning process, 2025. URL `https://arxiv.org/abs/2501.13536`.

[18] Chuanqi Zang, Hanqing Wang, Mingtao Pei, and Wei Liang. Discovering the real association: Multimodal causal reasoning in video question answering. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 19027–19036, 2023. doi: 10.1109/CVPR52729.2023.01824.

[19] Ranjay Krishna, Yuke Zhu, Oliver Groth, Justin Johnson, Kenji Hata, Joshua Kravitz, Stephanie Chen, Yannis Kalantidis, Li-Jia Li, David A. Shamma, Michael S. Bernstein, and Fei-Fei Li. Visual genome: Connecting language and vision using crowdsourced dense image annotations, 2016. URL `https://arxiv.org/abs/1602.07332`.

[20] Bolei Zhou, Agata Lapedriza, Jianxiong Xiao, Antonio Torralba, and Aude Oliva. Learning deep features for scene recognition using places database. In Z. Ghahramani, M. Welling, C. Cortes, N. Lawrence, and K.Q. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 27. Curran Associates, Inc., 2014. URL `https://proceedings.neurips.cc/paper_files/paper/2014/file/19ea3982b415d7bb3363917eb3d60c4a-Paper.pdf`.

[21] Subhashini Venugopalan, Huijuan Xu, Jeff Donahue, Marcus Rohrbach, Raymond Mooney, and Kate Saenko. Translating videos to natural language using deep recurrent neural networks. In Rada Mihalcea, Joyce Chai, and Anoop Sarkar, editors, *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1494–1504, Denver, Colorado, May–June 2015. Association for Computational Linguistics. doi: 10.3115/v1/N15-1173. URL `https://aclanthology.org/N15-1173/`.

[22] Junbin Xiao, Xindi Shang, Angela Yao, and Tat-Seng Chua. Next-qa: Next phase of question-answering to explaining temporal actions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9777–9786, June 2021.

[23] Arjun Majumdar, Anurag Ajay, Xiaohan Zhang, Pranav Putta, Sriram Yenamandra, Mikael Henaff, Sneha Silwal, Paul Mcvay, Oleksandr Maksymets, Sergio Arnaud, Karmesh Yadav, Qiyang Li, Ben Newman, Mohit Sharma, Vincent Berges, Shiqi Zhang, Pulkit Agrawal, Yonatan Bisk, Dhruv Batra, Mrinal Kalakrishnan, Franziska Meier, Chris Paxton, Sasha Sax, and Aravind Rajeswaran. Openeqa: Embodied question answering in the era of foundation models. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024.

[24] OpenAI. Openai cookbook: Examples guides for building with the openai api. `https://cookbook.openai.com/`, 2025.

[25] Xindi Shang, Donglin Di, Junbin Xiao, Yu Cao, Xun Yang, and Tat-Seng Chua. Annotating objects and relations in user-generated videos. In *Proceedings of the 2019 on International Conference on Multimedia Retrieval*, pages 279–287. ACM, 2019.

[26] Bart Thomee, David A Shamma, Gerald Friedland, Benjamin Elizalde, Karl Ni, Douglas Poland, Damian Borth, and Li-Jia Li. Yfcc100m: The new data in multimedia research. *Communications of the ACM*, 59(2):64–73, 2016.

[27] Abhishek Das, Samyak Datta, Georgia Gkioxari, Stefan Lee, Devi Parikh, and Dhruv Batra. Embodied question answering, 2017. URL `https://arxiv.org/abs/1711.11543`.

[28] Santhosh K. Ramakrishnan, Aaron Gokaslan, Erik Wijmans, Oleksandr Maksymets, Alex Clegg, John Turner, Eric Undersander, Wojciech Galuba, Andrew Westbury, Angel X. Chang, Manolis Savva, Yili Zhao, and Dhruv Batra. Habitat-matterport 3d dataset (hm3d): 1000 large-scale 3d environments for embodied ai, 2021. URL `https://arxiv.org/abs/2109.08238`.

[29] Angela Dai, Angel X. Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. Scannet: Richly-annotated 3d reconstructions of indoor scenes, 2017. URL `https://arxiv.org/abs/1702.04405`.

[30] OpenAI. Gpt-4 technical report, 2024. URL `https://arxiv.org/abs/2303.08774`.

[31] OpenAI. Gpt-4o system card, 2024. URL `https://arxiv.org/abs/2410.21276`.

[32] OpenAI. Introducing openai o3 and o4-mini. `https://openai.com/index/introducing-o3-and-o4-mini/`, April 2025.

[33] Trevor Lin, Ryan T Lin, Rahul Mhaskar, and Curtis E Margo. Evaluating the accuracy of advanced language learning models in ophthalmology: A comparative study of chatgpt-4o and meta ai's llama 3.1. *Advances in Ophthalmology Practice and Research*, 5(2):95–99, 2025. doi: 10.1016/j.aopr.2025.01.002.

A Extended Results

LMM	LRM	Overall	Object Recognition	World Knowledge
GPT-4o	———	67.7	71.4	62.4
Gemini 2.5 Flash	———	58.8	56.2	69.2
Object Identification				
GPT-4o	o4-mini	55.7	56.7	60.7
GPT-4o	Gemini 2.5 Pro	59.7	62.6	64.9
Gemini 2.5 Flash	o4-mini	**63.6**	**60.7**	68.8
Gemini 2.5 Flash	Gemini 2.5 Pro	**67.1**	**68.7**	68.9
Scene Context				
GPT-4o	o4-mini	48.1	43.7	54.8
GPT-4o	Gemini 2.5 Pro	47.8	43.1	55.5
Gemini 2.5 Flash	o4-mini	**66.7**	**69.3**	68.1
Gemini 2.5 Flash	Gemini 2.5 Pro	**67.8**	**67.9**	70.1

Table 2: **LLM-Match scores on the OpenEQA dataset with distinct LMM and LRM pairs**. The LRMs perform an upstream task of either object recognition or scene context. The scores are for all questions (overall), object recognition questions, and world knowledge questions. The object recognition and world knowledge questions are stratified from the OpenEQA dataset, which originally has seven questions types. Both question types are stratified since we hypothesize that they will display the most significant score differences through our framework. Notable score improvements are represented in bold.

B Prompts

In this section we present the prompts used in our experiments. Figure 3 represents the baseline prompt passed into GPT-4o and Gemini 2.5 Flash without any upstream task augmentation. {question} in Figure 3 denotes where the question sourced from the dataset is input.

The prompts for the two upstream tasks, object identification and scene context generation, are presented in Figures 4 and 5, respectively. Figure 6 shows the prompt used for processing QA questions along with upstream reasoning, which is passed into the LMM. In figure 6, {upstream_task} represents the upstream input, {question} represents the question sourced from the dataset, and {upstream_task_placeholder} represents a task-differentiating variable, since the same prompt was used for different upstream tasks.

Example outputs for an object identification or scene context generation task are presented in Figures 8 and 7, respectively.

Baseline VideoQA Prompt
You are an embodied AI assistant. Your task is to answer a question about a given environment from images using your own knowledge.
*Primary Goal:** Apply your general and functional knowledge to reason about what is possible or true based on the context using your given inputs.

CRITICAL RULES: - **Be direct and brief.** Your answer should be as short as possible.
- For questions that can be answered with "Yes" or "No", you MUST answer with only "Yes" or "No".
- **Do not explain your reasoning or mention the provided context.** Avoid phrases like "Based on the context..." or "Observing the images...".
- **Don't overanalyze, many of the answers are simple and are not extremely detailed or have a lot of adjectives**
- **Very short statements are also acceptable**

User Query: {question} A:

Figure 3: Prompt for standalone LMMs to run baselines.

Object Identification Prompt
Note This is an upstream task of object identification and their spatial layout.
Use reasoning to analyze the provided sequence of images from a first-person perspective. Your goal is to generate a comprehensive, structured, and factual description of the object inventory and spatial layout. Be as detailed as possible.
Provide your analysis in the following structured format:

Object Inventory:
Major Items: List prominent furniture and appliances. For each item, specify its attributes (color, material, shape) and its location relative to the room and other objects (e.g., "A rectangular wooden desk is against the far wall, to the left of the window").
Minor Items: Detail smaller objects such as decorations, electronics, containers, or personal items found on surfaces or shelves. Describe their key features and placement.

Figure 4: Upstream task prompt for LRMs to **identify objects** in the inputted frames.

Scene Context Prompt
Note This is an upstream task of scene context.
Use reasoning to analyze the provided sequence of images from a first-person perspective. Your goal is to generate a comprehensive, structured, and factual description of the scenes. Be as detailed as possible.
Provide your analysis in the following structured format:

Scene Overview:
Identification and Purpose: Identify the type of room or space (e.g., office, bedroom, kitchen). Describe its likely purpose.
Architectural Details: Note the materials and styles of the floor, walls, and ceiling. Mention significant structural elements like windows, doors, or stairs.
Ambiance: Briefly describe the overall condition and atmosphere (e.g., tidy, cluttered, modern, rustic, well-lit, dimly lit).

Figure 5: Upstream task prompt for LRMs to generate **scene context** of the inputted frames.

VideoQA Prompt with Included Upstream Reasoning
You are an embodied AI assistant. Your task is to answer a question by synthesizing information from images, an upstream task, and your own knowledge.

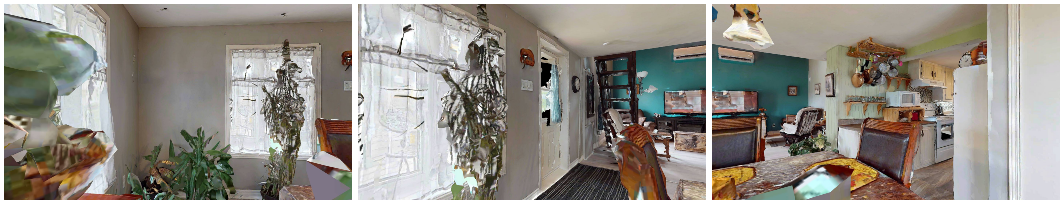
Primary Goal: Use the upstream task as a helpful source for the current state of the environment.
*Secondary Goal:** Apply your general and functional knowledge to reason about what is possible or true based on the context using your given inputs.

CRITICAL RULES:
- Be direct and brief. Your answer should be as short as possible
- For questions that can be answered with "Yes" or "No", you MUST answer with only "Yes" or "No"
- Do not explain your reasoning or mention the provided context. Avoid phrases like "Based on the context..." or "Observing the images..."
- Don't overanalyze, many of the answers are simple and are not extremely detailed or have a lot of adjectives
- Very short statements are also acceptable
- The upstream task is designed to help you answer questions—not give you the answer
- Important information may be missing from the upstream task, so make sure you still use your general and functional knowledge to reason

Upstream Task:
This is the upstream task of {upstream_task_placeholder} for the given scenes.
{upstream_task}

User Query: {question}
A:

Figure 6: Upstream task prompt for LRMs to generate **scene context** of the inputted frames.



Example Scene Context Output

Scene Overview:

Identification and Purpose

Open-plan living space combining a small dining area, a kitchenette, and a lounge.

Functions include meal preparation and dining (nearest the kitchen), casual seating and media viewing (farther in), and general circulation to a loft via a narrow staircase.

Architectural Details

Flooring: Light-toned, wide-plank wood or wood-look laminate laid continuously throughout.

Walls: Predominantly light gray paint with white trim; one full accent wall in the lounge is painted teal.

Ceiling: Flat, painted white over the dining and kitchen zone; vaulted wood-paneled section over the lounge.

Windows & Doors:

Two double-hung windows dressed with semi-sheer lace curtains in the dining corner.

A white paneled exterior door (with glazed upper section) opens near a side wall.

Interior doorways (one near the lounge leads to a bedroom) feature white trim.

A shallow closet with bifold white doors sits adjacent to the kitchen entrance.

Staircase: A compact, dark-stained wooden ladder-style stair rises against the teal wall to a loft or mezzanine level.

Kitchen Fixtures: Light oak cabinetry, open shelving for glassware and cookware, a wall-mounted pot rack made of rough-hewn wood, white range and refrigerator, microwave on a countertop.

Ambiance

Lighting: Bright natural daylight filters through the lace curtains; ambient light is supplemented by a pendant lamp over the dining table and floor lamps in the lounge.

Furnishings & Decor:

Dining: A small rectangular table with two high-back chairs and leaf-patterned placemats.

Lounge: Two upholstered glider chairs, a leather-look loveseat, a low TV stand with flat-screen, and a rustic storage trunk used as a coffee table.

Accessories: Several potted plants cluster by the window, framed prints on the walls, a wall clock, and a decorative wall plaque.

Condition & Style: The space feels cozy and lived-in, leaning toward a rustic-eclectic style. It is generally tidy, though numerous decorative items and plants create a slightly busy, homey atmosphere.

Figure 7: Example output of generated **scene context**.

Example Object Identification Output

Based on the sequence of images, here is a detailed analysis of the object inventory and spatial layout of the interconnected kitchen, dining, and living areas.

Object Inventory

Major Items:

Refrigerator: A tall, white refrigerator is located at the edge of the kitchen area, to the left of a bifold closet door.

Stove/Oven: A white, freestanding electric range with an oven is positioned to the right of the main kitchen counter space.

Dining Table: A large rectangular dining table with a mottled brown and gray tabletop sits between the kitchen and living areas.

Dining Chairs: At least two wooden dining chairs with dark brown frames, brown leather-like upholstered seats, and high, slatted backs are positioned at the dining table.

Sofa: A plush, brown fabric sofa with multiple throw pillows is situated against the far right wall of the living room.

Armchair: A matching plush, brown fabric armchair is located in the corner to the left of the TV console.

Rocking Chairs: Two rocking chairs with dark wood frames are present. One, with a white/beige cushioned seat and a white throw blanket, is next to the sofa. Another, with similar light-colored cushions, is next to a tall shelving unit.

TV Console / Fireplace: A dark brown wooden media console with glass-front cabinets and a central electric fireplace insert is placed against the teal accent wall, beneath the television.

Tall Shelving Unit: A very tall, dark wood, open-back shelving unit with multiple shelves stands against the gray wall, acting as a partial divider between the dining and living areas.

Trunk Coffee Table: A vintage-style rectangular trunk, light-colored with a dark printed design (possibly a map or historical illustration), serves as a coffee table in front of the TV console.

Kitchen Cabinets: Off-white or cream-colored kitchen cabinets are visible, with a set of upper and lower cabinets to the right of the stove.

Minor Items:

Microwave: A white microwave oven sits on a small wooden stand on the kitchen countertop.

Pot Rack: A large, wooden, ceiling-mounted pot rack is installed above the kitchen peninsula. Numerous pots and pans, including copper and stainless steel ones, hang from it.

Wall Shelves:** Two wooden shelves are mounted on the light green kitchen wall below the pot rack. The lower shelf holds a neat row of clear glass jars with lids.

Potted Plants: Several potted plants are throughout the space. A large, leafy green plant is in the corner of the dining area by a window. Another tall, thinner plant is next to it. A smaller plant is on the floor near the kitchen peninsula.

Television: A large, widescreen flat-panel TV is mounted on the teal accent wall above the media console.

Air Conditioner: A white, ductless mini-split air conditioner unit is mounted high on the teal wall, above and to the right of the TV.

Lighting: A floor lamp with a silver-toned base and a white upward-facing shade stands between the tall shelving unit and the armchair. A ceiling-mounted light fixture with a bell-shaped glass shade is visible above the dining area.

Clocks: A round, black-rimmed analog clock is on the gray wall near the exterior door. Another round, white-faced clock is visible on top of the refrigerator.

Wall Decor: Several framed pictures adorn the walls. Two small, square-framed pictures are on the teal wall. A larger, rectangular framed picture of a winter scene hangs on the off-white wall next to a doorway. A small, irregularly shaped piece of wood art is on the gray wall by the dining area.

Curtains: Sheer, white, gathered curtains are on the windows in the dining area. Darker gray or black curtains are on the window in the living area corner.

Small Tables: A small, slatted wooden coffee table is next to the armchair by the tall shelf. A small end table with a crisscross base is next to the sofa.

Rugs: A small, dark, striped rug is on the floor by the exterior door. A small area rug with a dark border is under the trunk coffee table.

Closet Door: A white, two-panel bifold door is located between the refrigerator and the dining area wall.,

Figure 8: Example output of generated **object identification**.