Spatial-Aware Visual Program Reasoning for Complex Visual Questions Answering

Anonymous ACL submission

Abstract

Visual Question Answering (VQA) often requires complex multi-hop reasoning encompassing both vision and language. Despite the remarkable performance of Large Multimodal Models (LMMs) in vision-language tasks, they encounter difficulties when faced with challenging scenarios that require complex reasoning and may be susceptible to object hallucination. This paper introduces a novel framework named Spatial-aware Visual Program Reasoning (SVPR). The primary goal of SVPR is to enhance the alignment between vision and language within LMMs, fostering their multi-hop reasoning abilities and ultimately strengthening their capacity to address complex visual reasoning tasks. We first utilize the strong visual understanding abilities of LMMs to generate scene graphs, facilitating coordination between vision and language at semantic levels. Then, we leverage the in-context learning ability of LMMs to generate visual programs, which guide the question decomposition process. Finally, we employ a program solver to execute the programs and derive the final answer. This process makes our approach both explanatory and robust, providing clear explanations of its reasoning process while ensuring the faithfulness of the answer to the visual input. We evaluate our framework on two challenging multi-hop multimodal VQA datasets and show its effectiveness under zero-shot settings. Our code is available: https://anonymous.4open.science/r/SVPR-5BBA

1 Introduction

Large Multimodal Models (LMMs) like GPT-4V (Achiam et al., 2023) and Gemini (Team et al., 2023) have demonstrated remarkable zero-shot capabilities in handling various visual-language tasks. Nevertheless, despite their significant advancements, LMMs demonstrate limited performance in answering complex questions that require multi-hop reasoning across various levels of visual information (Yang et al., 2023c; Ossowski et al., 2024; Wu and Xie, 2023). For instance, consider the image depicted in Table 1. A straightforward question such as *"What color is the building?"* requires only one-hop (one-step) reasoning to determine the color of the building in the image. In contrast, a more complex question like *"On which side of the walkway leading to the San Francisco Civic Center can the American flag be found?"* requires multi-hop reasoning: (i) visually detecting the walkway leading to the building, (ii) visually locating the American flag, and (iii) determining the spatial relationship between the walkway and

Question: *On which side of the walkway leading to the San Francisco Civic Center can the American flag be found?*

Ground Truth: The flag is located on the left side.

GPT-4V: The American flag is located on the **right side** of the walkway leading to the San Francisco Civic Center in the image provided.

GPT-4V+*SVPR*: The American flag [0.25, 0.3, 0.26, 0.35] is located on the **left side** of the walkway leading to the San Francisco Civic Center [0.3, 0.25, 0.7, 0.75].

Table 1: An example of SVPR in answering a visual question that requires spatial reasoning, with correct textual reasoning illustrated in **green** and incorrect textual reasoning illustrated in **red**. Additionally, SVPR provides bounding boxes (highlighted in blue) as visual evidence to provide grounding.

the flag, which involves spatial reasoning.

To facilitate Large Language Models (LLMs) and Large Multimodal models (LMMs) in breaking down the input question into multiple reasoning steps, several techniques have been proposed, such as Chain-of-Thought (Wei et al., 2022), Self-Ask (Press et al., 2023), Least-to-most prompting (Zhou et al., 2022), ReAct (Yao et al., 2022), and others. While these models excel in handling single-hop questions, they encounter challenges when confronted with multimodal multi-hop questions. In such scenarios, the formulation of subsequent questions is influenced by the answers to preceding sub-questions. Moreover, these techniques often do not explicitly facilitate coordination between vision and language and lack spatial awareness. Consequently, there is a discrepancy in semantic granularity between visual and textual information. Unlike textual sentences where each word is distinctly separated, identities within an image lack clear boundaries and aren't isolated in the same explicit manner.

In this paper, we introduce *Spatial-aware Visual Program Reasoning (SVPR)*, a novel framework designed to foster language-vision coordination and enhance the complex reasoning capabilities of LMMs in answering complex visual questions. Specifically, our framework consists of three stages: **(1) Scene graph generation** prompts LMMs to create a structured representation of the image known as a scene graph. This graph encapsulates detailed semantics by explicitly modeling objects, their attributes, and the relationships between pairs of objects; **(2) Visual program generation** decomposes the input question into simpler sub-questions by generating a visual reasoning program. This program is essentially a sequence of sub-tasks aimed at simplifying the overall reasoning process; **(3) Program solver** first answers the formulated sub-questions based on the image using a validator. These sub-questions and their corresponding sub-answers collectively act as rationales for the final reasoning step. Then, LMMs perform reasoning aggregation over the scene graph and rationales to derive the final answer and give justification for their reasoning process.

We evaluate our proposed framework on two challenging datasets that require complex reasoning abilities: WebQA (Chang et al., 2022) and GQA (Hudson and Manning, 2019). Our experiment results demonstrate that SVPR can effectively answer complex questions while providing clear explanations of its reasoning process.

In summary, our contributions are:

1. We introduce a new framework to enhance LMMs' vision-language coordination and multi-hop reasoning ability to answer complex visual questions.
2. Our framework is designed in a way that each step is transparent and consistent, thus providing both explainable and robust answers.
3. We comprehensively evaluate the effectiveness of our method, and the large improvements demonstrate its great potential in complex visual reasoning.

2 Background

Multi-modal Multi-hop Question Answering. Multimodal Multi-hop Question Answering (MMQA) (Chang et al., 2022; Reddy et al., 2022; Talmor et al., 2021) requires answering a question by reasoning over multiple input sources from different modalities. This task often involves multi-step reasoning, wherein one or more intermediate conclusions must be reached before arriving at the final answer (Mavi et al., 2022; Wang et al., 2024). Each intermediate conclusion acts as a necessary premise for the subsequent one. This progression of intermediate and final conclusions is called a reasoning chain. While previous approaches (Chang et al., 2022; Chen et al., 2022; Li et al., 2022; Reddy et al., 2022; Talmor et al., 2021; Yang et al., 2023b) utilizing supervised learning have demonstrated promising outcomes, current attention has pivoted towards MMQA under the zero-shot settings. To solve the zero-shot compositional VQA task, VISPROG (Gupta and Kembhavi, 2023) uses a neural-symbolic approach to perform multi-step reasoning using language models. (Rajabzadeh et al., 2023) utilize a tool-interacting divide-and-conquer approach, empowering large language models (LLMs) to address intricate multimodal multi-hop inquiries. More recently, II-MMR (Kil et al., 2024) employs two distinct prompting techniques to determine a reasoning path leading to its solution. Like the prior approaches, our framework also adopts a decomposition strategy for executing multi-step reasoning. However, our emphasis lies in cultivating visual-language coordination and prioritizing visual cues.

Spatial-Aware Prompting Methods. While LMMs have demonstrated remarkable visual reasoning capabilities, they remain vulnerable to hallucination issues, including object, attribute, or relation hallucination. Previous research has indicated that this issue could largely stem from a lack of visual-language coordination or a robust language prior, causing the model to overlook crucial visual cues. To address these challenges, several visual prompting techniques have been proposed to enhance the visual perception of LMMs. For example, RedCircle (Shtedritski et al., 2023) utilized a circle marker to direct the model's attention toward specific regions for fine-grained classification. Meanwhile, FGVP (Yang et al., 2024), SCAFFOLD (Lei et al., 2024), and SOM (Yang et al., 2023a) investigated prompts for spatial reasoning using dot matrices or pre-trained models. Furthermore, (Wu et al., 2024) introduced a prompting paradigm and toolkit aimed at unlocking the zero-shot object detection capability of LMMs. In contrast, given that multi-hop questions often require a clear comprehension of semantic relationships between objects, we leverage scene graphs (Zhu et al., 2022) to enhance vision-language coordination.

Symbolic-Guided Reasoning. While approaches like Chain-of-Thought (Wei et al., 2022), Self-Ask (Press et al., 2023), and ReAct (Yao et al., 2022) can elicit LLM's step-by-step reasoning capabilities, they perform reasoning directly over natural language, where the intrinsic complexity and ambiguity of natural language could bring undesired issues such as unfaithful reasoning and hallucinations. To address these challenges, several neural-symbolic approaches (Pan et al., 2023b,a; Wang and Shu, 2023; Gupta and Kembhavi, 2023) have been proposed to integrate LLMs with symbolic logic. Our work aligns with the symbolic-guided reasoning paradigm. However, unlike previous studies, we explicitly incorporate scene graph information into the textual prompt to offer visual grounding for LMMs' reasoning processes. The inclusion of structural semantic information in the scene graphs enhances our framework's ability to excel in visual reasoning tasks and provide visual evidence with bounding boxes.

3 Method

As depicted in Figure 1, our model takes a natural language question Q and one or multiple images I linked to the question as inputs. Subsequently, our framework conducts spatial-aware visual reasoning through three distinct stages. In the *scene graph generation* stage, we prompt an LMM to identify the objects using bounding boxes as evidence, as well as to discern the attributes of these objects and the relationships between them. In the *visual program-guided reasoning stage*, we instruct the LMMs with a set of in-context examples to translate the question into a symbolic visual program. Subsequently, a program interpreter is employed to convert the visual program into a set of sub-questions. Finally, in the *program-solving* stage, a validator answers the sub-questions, and these, along with their corresponding sub-answers, collectively form rationales. We then aggregate the scene graph and the rationales to conclude the final answer and provide explanations to justify the decision process.

3.1 Scene Graph Generation

Scene Graph (Zhu et al., 2022) is a structural representation that captures detailed semantics. A scene graph comprises relationship triplets represented as <*subject, relation, object*> or <*object, is, attribute*>, which encapsulate the modeling of objects, attributes of objects, and the relationships between paired objects. Given that multi-hop questions usually revolve around attributes and relationships between objects, the first step involves extracting the scene graph to represent the structural information derived from the input images. In light of the strong visual understanding ability and rich world knowledge of LMMs, we prompt an LMM to fulfill this task. First, we overlay the images with a grid and provide a labeling system to assist LMMs in identifying and referring to specific points within the images. Then, we prompt an LMM to generate the scene graph and provide bounding boxes for objects. Specifically, each bounding boxes are represented as a tuple $[x_{min}, y_{min}, x_{max}, y_{max}]$, where x_{min} and y_{min} are coordinates of the top-left corner of the bounding box; x_{max} and y_{max} are coordinates of the bottom-right corner of the bounding box. The prompt for scene graph generation is listed in Section A in the appendix.

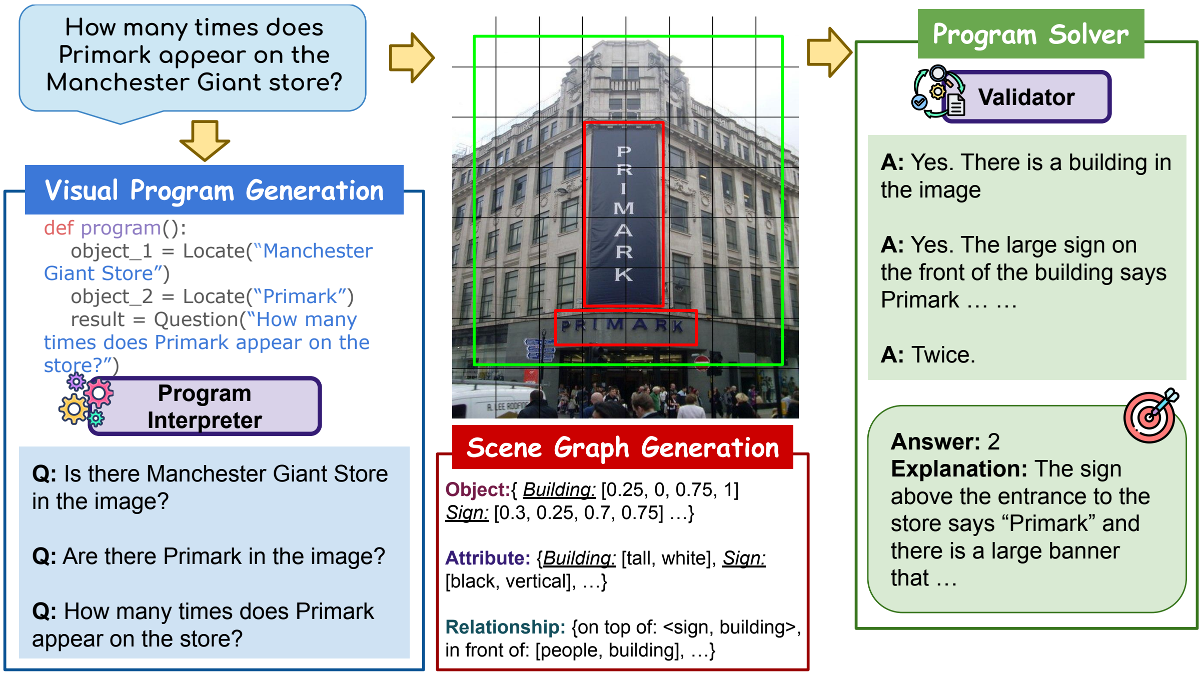

Figure 1: Overview of our *SVPR* framework, which consists of three stages: (i) *SVPR* generates a scene graph and uses it to provide LMMs with structural semantic information of the input images; (ii) *SVPR* then generates symbolic visual programs to represent the multi-step reasoning process and a program interpreter translates the function calls in the program into a set of sub-questions; and (iii) *SVPR* uses a validator to provide answers to the sub-questions and aggregates the reasoning chain to derive the final answer and generate explanations.

3.2 Visual Program-Guided Reasoning

This stage follows a program generation and execution paradigm to translate the natural language question into a symbolic reasoning program.

Program Generation. Given the question and the input images, a planner P generates a reasoning program $P = [S_1, ..., S_n]$ for it, which consists of n sequentially ordered reasoning steps S_i. Each reasoning step $S_i \in P$ is an instruction in controlled natural language that directs S_i to a function that represents a reasoning step. Specifically, we define two functions that the program can invoke during program generation. The `Locate()` function determines the location of objects in the images using bounding boxes, while the `Question()` function poses inquiries regarding the attributes and relationships of objects.

Program Interpreter. The role of the program interpreter is to parse the generated visual programs into a set of sub-questions in natural language. Specifically, each `Locate()` function is translated into *"Is there object in the image? If so, please provide its bounding boxes.*. Once we have obtained the list of sub-questions, a program validator to answer the sub-questions, utilizing the scene graph as visual grounding.

3.3 Program Solver

During this stage, SVPR consolidates the visual cues provided by the scene graph along with the rationales generated by the program validator, to derive the final answer.

Program Validator. The goal of the program validator is to answer the sub-questions generated by the visual programs. For object-level questions generated by the `Locate()` functions, we employ a pre-trained VQA model (Li et al., 2023a) to answer the question. When compared to LMMs, VQA models typically produce shorter answers with fewer hallucinations, making them a pragmatic option. For attribute-level and relation-level queries generated by the `Question()` functions, we leverage LMMs to provide answers due to their strong visual comprehension capabilities.

Answer Prediction. Guided by the scene graph, along with the sub-questions and their corresponding sub-answers, we employ LMMs as reasoning agents to deduce the final answer. To enhance explainability, we instruct the LMMs to offer justifications for their decisions. Additionally, we prompt them to append bounding boxes directly after expressions referencing objects. This approach facilitates the correspondence between entities mentioned in the responses and object instances in the image, thereby providing convenient access to verify the reliability of the output. The prompt for aggregation is included in Section A in the appendix.

4 Experiments

We compare *SVPR* against three baselines on two challenges: Multi-hop Multimodal QA (MMQA) and Compositional QA (CQA). Our experiment settings are described in Section 4.1, 4.2 & 4.3 and we discuss our main results in Section 4.4.

4.1 Dataset

To demonstrate the effectiveness of *SVPR* for MMQA and CQA, we conduct experiments on WebQA and GQA datasets respectively.

WebQA (Chang et al., 2022) is a challenging benchmark for multi-hop multimodal question-answering (MMQA) tasks. This dataset contains questions that are knowledge-seeking and resemble real-world use cases, each question has one or more images as positive evidence associated with it. Each question falls into one of the four categories: color, shape, number (i.e., "how many"), yes/no, and other. To reduce the GPT4-V API costs, we use stratified sampling to select a total of 250 entries from each question category.

GQA (Hudson and Manning, 2019) is a dataset featuring compositional questions over real-world images. Many of the GQA questions involve multiple reasoning skills, spatial understanding, and multi-step inference. We choose the balanced validation set, where the answer distribution for different groups of questions is tightly controlled, in order to prevent educated guess using language and world priors. For the same cost restriction reasons, we sampled 250 entries from the balanced validation set.

4.2 Baselines

We compare our proposed framework against the following three baselines:

Direct This baseline directly prompts LMMs to answer the question based on the input images, establishing a straightforward baseline without any prompt optimization.

Chain-of-Thought (Wei et al., 2022) is a popular approach that guides LMMs to perform step-by-step reasoning before outputting the final answer. This prompting method poses a question to the model and has the model to output a chain of thought before outputting its final answer. The prompt text "Let's think step-by-step" is prepended to the task description.

SCAFFOLD (Lei et al., 2024) is a visual prompting scheme that promotes vision-language coordination in LMMs. Specifically, SCAFFOLD first overlays a dot matrix within the image as visual information anchors and leverages multi-dimensional coordinates as textual positional references. This baseline establishes a scaffold for enhancing vision-language coordination in LMMs and has demonstrated superior performance in spatial and compositional reasoning benchmarks.

4.3 Experiment Settings

LMMs. Our pipeline is training-free and comprises an LMM and a pre-trained VQA model as the validator to answer the sub-questions. Specifically, we choose the following three LMMs, InstructBlip (Dai et al., 2024) is an open-source instruction-tuned LMM that achieves state-of-the-art performance on a wide variety of vision tasks. Specifically, we use the InstructBlip-Vicuna-13B model. We also choose two much larger closed-source LMMs: GPT4-V (Achiam et al., 2023) and Gemini (Team et al., 2023). We utilize Blip2-FlanT5-XXL as the VQA model to answer the sub-questions conditioned on the input image.

Evaluation. Since the answers generated by LMMs are open-ended, traditional metrics such as SQuAD (Rajpurkar et al., 2016) style Exact-Match and F1 do not measure the performance to its fullest. For instance, LLMs excel in generating diverse and contextually relevant responses, which might not always align with exact matches to gold standard answers. Instead, they often provide paraphrases or alternative expressions that convey

	WebQA				GQA			
	Direct	*CoT*	*SCAFFOLD*	*SVPR*	*Direct*	*CoT*	*SCAFFOLD*	*SVPR*
InstructBlip	46.8	45.4	43.6	**52.2**	51.6	50.2	51.4	**55.2**
Gemini	55.2	58.4	61.2	**69.6**	52.4	54.4	56.4	**62.8**
GPT4-V	61.8	62.2	68.4	**71.6**	47.2	51.2	55.4	**65.2**

Table 2: Accuracy of Direct, Chain-of-Thought (CoT), Scaffold, and our method *SVPR* on two challenging visual question answering datasets, WebQA and GQA. We use three unique LMMs for our experiments. The best results within each dataset are highlighted.

the same underlying meaning. This highlights the need for more nuanced evaluation strategies that account for semantic equivalence rather than strict verbatim matches. Therefore following (Lin et al., 2022; Li et al., 2023b; Sun et al., 2024; Wang et al., 2023), we use GPT-4 as a judge to check whether the generated answer has the same meaning as the gold answer. The evaluation prompt is included in Section A in the appendix.

4.4 Main Results

We report the overall results of *SVPR* in Table 2. *SVPR* achieves the best performance on both datasets, demonstrating its effectiveness. Based on the experiment results, we have the following major observations:

Scene graphs improve visual reasoning. On the WebQA dataset, SVPR showcases superior performance over Direct, CoT, and Scaffold by margins of 15.86%, 15.11%, and 4.68% on GPT-4V, respectively. This highlights SVPR's effectiveness in answering multi-modal, multi-hop visual questions. Among the baselines, Scaffold proves to be more effective than Direct and CoT. This implies that integrating dot matrices as visual anchors enhances LLMs' spatial reasoning capabilities. However, since many questions demand not only visual comprehension and vision anchors but also a profound semantic understanding of object attributes and relationships within the scene, scene graphs play a crucial role in providing LLMs with deeper semantic visual understanding. They aid LLMs in achieving more comprehensive comprehension. Similar observations are made on the GQA dataset, suggesting that SVPR performs well not only on multi-hop reasoning tasks but also on compositional visual reasoning tasks. In addition to our primary findings, our analysis also highlights discernible performance variations among various LMMs. Notably, our investigation reveals that GPT-4V and Gemini consistently outperform the smaller-scale InstructBlip model, which relies on Vicuna-13B as its backbone LLM. This observation underscores the significant impact of model architecture and size on overall performance metrics. Furthermore, our comparative analysis demonstrates a slight but consistent advantage held by GPT-4V over Gemini across both datasets evaluated. These findings emphasize the importance of considering model selection criteria tailored to specific task requirements and performance objectives.

Symbolic-guided reasoning can decompose the reasoning chain better. Our *SVPR* method, which uses visual programs to guide the decomposition reasoning approach outperforms CoT and SCAFFOLD baselines on both datasets. This suggests that the visual programs help LMMs to better decompose questions, and result in more accurate reasoning. On both WebQA and GQA, Scaffold exhibits a significant performance boost. Both datasets require intricate reasoning abilities to deconstruct the questions and employ a divide-and-conquer approach to problem-solving. Since Scaffold also actively promotes vision-language coordination, we can infer the performance comes from SVPR's better question decomposition strategy. Overall, SVPR exhibits superior performance compared to the Direct baseline across both datasets. This observation indicates the critical role of question decomposition in complex visual question answering, as Direct does not decompose the questions.

	Color	Shape	Number	Yes/No	Other
GPT-V	54.2	48.2	46.2	82.4	78.2
GPT-V+Scaffold	52.6	48.4	50.4	76.6	82.6
GPT-V+SVPR	66.4	56.2	64.4	86.2	84.6

Table 3: Ablation Study: Impact of Scene Graphs

4.5 The Impacts of Scene Graphs

To deepen our understanding of the role of scene graphs in the decision-making process of LLMs, we conduct an ablation study on the WebQA dataset using GPT4-V. This study involves comparing the performance of Direct, SCAFFOLD, and SVPR approaches. The Direct approach lacks any visual understanding information and solely represents the raw visual understanding capabilities of LMMs. In contrast, SCAFFOLD overlays dot matrices onto the original image and incorporates textual prompts to actively guide LMMs. By utilizing coordinates as vision anchors and reference points, SCAFFOLD promotes coordination between vision and language. In contrast, our SVPR not only incorporates vision anchor points but also integrates deep semantic information from scene graphs. This enables LMMs to engage in structured visual understanding, enhancing their comprehension capabilities. To comprehend the reasoning challenges where scene graphs play the most significant role, we present the performance based on the question category. Table 3 shows the experimental results, indicating that SVPR outperforms both baselines, highlighting its effectiveness. Additionally, we notice that questions categorized as more complex, involving reasoning over relationships between objects such as Yes/No and others, exhibit superior performance on SVPR compared to SCAFFOLD. This underscores the utility of incorporating structured semantic information like scene graphs, particularly in addressing questions necessitating structured reasoning.

4.6 The Impacts of Validators

As discussed in Section 4.4, program-guided reasoning demonstrates superior decomposition of questions compared to CoT-like prompt techniques. However, it's crucial to note that to reach the final correct answer, we must first answer the sub-questions correctly. To evaluate the potential impact of using different validators on the overall performance of SVPR, we conduct the following ablation study. We utilize Gemini to generate the visual programs and employ the following four models as validators. In addition to employing LMMs, we hypothesize that pre-trained VQA models such as (Li et al., 2023a) can mitigate the risk of object hallucination. This refers to the phenomenon where models may generate text describing objects that are not actually present in the image. Given that VQA models typically generate shorter answers compared to LMMs, albeit with fewer instances of hallucinations, they can indeed be considered a viable option for addressing this issue. As shown in Table 4, our experiment results reveal that despite our assumption that pre-trained VQA models like Blip2 would exhibit superior performance and hallucinate less, they do not perform nearly as well as the larger models. This phenomenon can be attributed to two main factors. Firstly, we observe a significant number of questions that possess inherent ambiguity, leading to misunderstandings by Blip2. Secondly, certain questions necessitate a profound visual understanding of the images. These questions inquire about specific details within the images, demanding a heightened visual comprehension to accurately recognize such details. Furthermore, we notice that Blip2 tends to perform better on the GQA dataset, where the sub-questions are generally more straightforward to answer.

	Blip2	**InstructBlip**	**Gemini**	**GPT-V**
WebQA	48.4	52.8	69.6	70.4
GQA	52.4	54.6	62.8	64.2

Table 4: Ablation Study: Impact of Validators

4.7 Taxonomy of Reasons for Failures

As shown in Table 5, upon examining GPT-V's explanations, we identify four primary categories of failure reasons:

Recognition errors occur when the model's visual component misinterprets images, resulting in difficulties in identifying vital details within them. For instance, in subfigure (a), when prompted about the number of distinct railings on the stairs depicted, Gemini failed to detect any railings in the images. Beyond the limited visual comprehension of LMMs regarding image particulars, this could also stem from image-related problems, such as low image quality such as blurriness, or incompleteness, which impairs the model's accuracy in recognizing content.

Reasoning errors emerge when the model misinterprets images due to diverse factors, including unnatural distortions, uncommon perspectives, or logical inaccuracies. For instance, in subfigure (b), the model identifies there is a cross on one of the domes of Saint Peter's Basilica. However, it fails to realize the question asks whether there is a cross on every dome of Saint Peter's Basilica.

(a) Recognition Error	(b) Reasoning Error	(c) Incorrect Rationales	(d) Conflict Rationales
			
Question: How many different railings are on the stairs at the TAMU Psychology Building?	**Question:** Are there crosses on top of every dome of Saint Peter's Basilica?	**Question:** How many points does each star on the obverse of the 1856 double eagle have?	**Question:** How many flags fly outside the Madison Square Garden?
Answer: Three different railings are on the stairs at the TAMU Psychology Building.	**Answer:** No, there are not crosses on top of every dome on top of Saint Peter's Basilica.	**Answer:** On the obverse of the 1856 double eagle, each star has six points.	**Answer:** One flag flies outside the Madison Square Garden.
Gemini+SVPR: The Scene Graph does not list any railings associated with the stairs or building. ... Despite the third rationale mentioning four railings, this is not supported by the Scene Graph or the other Rationals, so we conclude no railings.	**Gemini+SVPR:** Based on the scene graph, there is a dome on top of Saint Peter's Basilica, and the rationales confirm that there is a cross on top of the dome. Additionally ... Therefore, there are crosses on top of every dome.	**Gemini+SVPR:** Based on the rationale provided, there is an obverse of the 1856 double eagle in the image and it is stated that each star has five points. ... Therefore, we can conclude that each star on the obverse of the coin has five points.	**Gemini+SVPR:** The Scene Graph does not mention any flags outside Madison Square Garden, and the Rationals provide conflicting answers. ... We can conclude that there are no flags flying outside Madison Square Garden.

Table 5: Examples demonstrate why GPT4-V fails to answer the questions. We identify four failure reasons: recognition error, reasoning error, incorrect rationales, and conflict rationales.

Incorrect rationales represent a critical challenge for models like SVPR, as they can significantly impact the accuracy and reliability of the final predictions. Subfigure (c) illustrates this phenomenon, showcasing how a cascade error during the aggregation reasoning phase leads the model to acquire an incorrect rationale—specifically, in this case, each star possesses five points. This erroneous rationale, in turn, undermines the model's ability to generate the correct prediction, highlighting the detrimental effects of error propagation within the SVPR pipeline.

Conflicting rationales present a significant challenge for models like SVPR, particularly when they encounter contradictory factual information from multiple rationales. This phenomenon underscores the complexity inherent in aggregating diverse streams of data and reasoning to arrive at a coherent conclusion. Subfigure (d) illustrates how SVPR grapples with this challenge, highlighting its struggle to determine the ultimate answer when faced with competing lines of reasoning. Therefore, improving the accuracy of the validators is a focus of our future work.

5 Conclusion

In this paper, we propose a novel approach to answer complex visual questions using LLMs by eliciting vision-language coordination and symbolic guided reasoning. We introduce SVPR, a visual reasoning method that enhances LMMs' vision-language coordination and multi-hop reasoning ability to answer complex questions. By explicitly incorporating scene graphs with bounding boxes into the textual prompts, SVPR actively integrates visual cues during reasoning and includes visual evidence as part of its explanations. The visual programs are shown to be effective in decomposing complex visual questions into a series of subquestions. Our experiment results show that SVPR demonstrates promising performance on two challenging datasets without any additional training. Additionally, we investigate the impact of visual awareness and program-guided reasoning on the performance of SVPR. The results indicate that SVPR can make accurate predictions and generate explanations while providing visual evidence. The limitations and future work are discussed in the subsequent section.

6 Limitations

We identify two main limitations of SVPR. First, SVPR depends on in-context learning coupled with self-refinement to convert a natural language question into a visual program representation. While this method has proven to be effective, it may face difficulties when dealing with questions with intricate grammar structures and logical structures. This arises from the difficulty in conveying complex grammatical rules to the language model through a limited number of demonstrations within a constrained context size. Second, our aggregation method purely relies on LMMs themselves, which could introduce potential hallucination problems. On the other hand, by using a more robust logic solver could help with the hallucination issues, but there would be a tradeoff between the applicability and the robustness of the model.

7 Ethical Statement

Biases. We acknowledge the possibility of biases existing within the data used for training the language models, as well as in certain factuality assessments. Unfortunately, these factors are beyond our control.

Intended Use and Misuse Potential. Our models have the potential to answer complex visual questions. However, it is essential to recognize that they may also be susceptible to misuse by malicious individuals. Therefore, we strongly urge researchers to approach their utilization with caution and prudence.

Environmental Impact. We want to highlight the environmental impact of using large language models, which demand substantial computational costs and rely on GPUs/TPUs for training, which contributes to global warming. However, it is worth noting that our approach does not train such models from scratch. Instead, we use few-shot in-context learning. Nevertheless, the large language models we used in this paper are likely running on GPU(s).

References

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

Yingshan Chang, Mridu Narang, Hisami Suzuki, Guihong Cao, Jianfeng Gao, and Yonatan Bisk. 2022. Webqa: Multihop and multimodal qa. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16495–16504.

Wenhu Chen, Hexiang Hu, Xi Chen, Pat Verga, and William Cohen. 2022. MuRAG: Multimodal retrieval-augmented generator for open question answering over images and text. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 5558–5570, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Wenliang Dai, Junnan Li, Dongxu Li, Anthony Meng Huat Tiong, Junqi Zhao, Weisheng Wang, Boyang Li, Pascale N Fung, and Steven Hoi. 2024. Instructblip: Towards general-purpose vision-language models with instruction tuning. *Advances in Neural Information Processing Systems*, 36.

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

Drew A Hudson and Christopher D Manning. 2019. Gqa: A new dataset for real-world visual reasoning and compositional question answering. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 6700–6709.

Jihyung Kil, Farideh Tavazoee, Dongyeop Kang, and Joo-Kyung Kim. 2024. Ii-mmr: Identifying and improving multi-modal multi-hop reasoning in visual question answering. *arXiv preprint arXiv:2402.11058*.

Xuanyu Lei, Zonghan Yang, Xinrui Chen, Peng Li, and Yang Liu. 2024. Scaffolding coordinates to promote vision-language coordination in large multi-modal models. *arXiv preprint arXiv:2402.12058*.

Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. 2023a. Blip-2: Bootstrapping language-image pretraining with frozen image encoders and large language models. In *International conference on machine learning*, pages 19730–19742. PMLR.

Junyi Li, Xiaoxue Cheng, Xin Zhao, Jian-Yun Nie, and Ji-Rong Wen. 2023b. HaluEval: A large-scale hallucination evaluation benchmark for large language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 6449–6464, Singapore. Association for Computational Linguistics.

Yongqi Li, Wenjie Li, and Liqiang Nie. 2022. MMCoQA: Conversational question answering over text, tables, and images. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4220–4231, Dublin, Ireland. Association for Computational Linguistics.

Stephanie Lin, Jacob Hilton, and Owain Evans. 2022. TruthfulQA: Measuring how models mimic human falsehoods. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3214–3252, Dublin, Ireland. Association for Computational Linguistics.

Vaibhav Mavi, Anubhav Jangra, and Adam Jatowt. 2022. A survey on multi-hop question answering and generation. *arXiv preprint arXiv:2204.09140*.

Timothy Ossowski, Ming Jiang, and Junjie Hu. 2024. Prompting large vision-language models for compositional reasoning. *arXiv preprint arXiv:2401.11337*.

Liangming Pan, Alon Albalak, Xinyi Wang, and William Wang. 2023a. Logic-LM: Empowering large language models with symbolic solvers for faithful logical reasoning. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 3806–3824, Singapore. Association for Computational Linguistics.

Liangming Pan, Xiaobao Wu, Xinyuan Lu, Anh Tuan Luu, William Yang Wang, Min-Yen Kan, and Preslav Nakov. 2023b. Fact-checking complex claims with program-guided reasoning. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6981–7004, Toronto, Canada. Association for Computational Linguistics.

Ofir Press, Muru Zhang, Sewon Min, Ludwig Schmidt, Noah Smith, and Mike Lewis. 2023. Measuring and narrowing the compositionality gap in language models. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 5687–5711, Singapore. Association for Computational Linguistics.

Hossein Rajabzadeh, Suyuchen Wang, Hyock Ju Kwon, and Bang Liu. 2023. Multimodal multi-hop question answering through a conversation between tools and efficiently finetuned large language models. *arXiv preprint arXiv:2309.08922*.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. SQuAD: 100,000+ questions for machine comprehension of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2383–2392, Austin, Texas. Association for Computational Linguistics.

Revant Reddy, Xilin Rui, Manling Li, Xudong Lin, Haoyang Wen, Jaemin Cho, Lifu Huang, Mohit Bansal, Avirup Sil, S. Chang, Alexander Schwing, and Heng Ji. 2022. Mumuqa: Multimedia multi-hop news question answering via cross-media knowledge extraction and grounding. *Proceedings of the AAAI Conference on Artificial Intelligence*, 36:11200–11208.

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

Lichao Sun, Yue Huang, Haoran Wang, Siyuan Wu, Qihui Zhang, Chujie Gao, Yixin Huang, Wenhan Lyu, Yixuan Zhang, Xiner Li, et al. 2024. Trustllm: Trustworthiness in large language models. *arXiv preprint arXiv:2401.05561*.

Alon Talmor, Ori Yoran, Amnon Catav, Dan Lahav, Yizhong Wang, Akari Asai, Gabriel Ilharco, Hannaneh Hajishirzi, and Jonathan Berant. 2021. Multimodalqa: Complex question answering over text, tables and images. *arXiv preprint arXiv:2104.06039*.

Gemini Team, Rohan Anil, Sebastian Borgeaud, Yonghui Wu, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M Dai, Anja Hauth, et al. 2023. Gemini: a family of highly capable multimodal models. *arXiv preprint arXiv:2312.11805*.

Haoran Wang and Kai Shu. 2023. Explainable claim verification via knowledge-grounded reasoning with large language models. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 6288–6304, Singapore. Association for Computational Linguistics.

Jiaan Wang, Yunlong Liang, Fandong Meng, Zengkui Sun, Haoxiang Shi, Zhixu Li, Jinan Xu, Jianfeng Qu, and Jie Zhou. 2023. Is chatgpt a good nlg evaluator? a preliminary study. *arXiv preprint arXiv:2303.04048*.

Yiqi Wang, Wentao Chen, Xiaotian Han, Xudong Lin, Haiteng Zhao, Yongfei Liu, Bohan Zhai, Jianbo Yuan, Quanzeng You, and Hongxia Yang. 2024. Exploring the reasoning abilities of multimodal large language models (mllms): A comprehensive survey on emerging trends in multimodal reasoning. *arXiv preprint arXiv:2401.06805*.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837.

Penghao Wu and Saining Xie. 2023. Guided visual search as a core mechanism in multimodal llms. *arXiv preprint arXiv:2312.14135*.

Yixuan Wu, Yizhou Wang, Shixiang Tang, Wenhao Wu, Tong He, Wanli Ouyang, Jian Wu, and Philip Torr. 2024. Dettoolchain: A new prompting paradigm to unleash detection ability of mllm. *arXiv preprint arXiv:2403.12488*.

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

Lingfeng Yang, Yueze Wang, Xiang Li, Xinlong Wang, and Jian Yang. 2024. Fine-grained visual prompting. *Advances in Neural Information Processing Systems*, 36.

Qian Yang, Qian Chen, Wen Wang, Baotian Hu, and Min Zhang. 2023b. Enhancing multi-modal multi-hop question answering via structured knowledge and unified retrieval-generation. In *Proceedings of the 31st ACM International Conference on Multimedia*, pages 5223–5234.

Zhengyuan Yang, Linjie Li, Kevin Lin, Jianfeng Wang, Chung-Ching Lin, Zicheng Liu, and Lijuan Wang. 2023c. The dawn of lmms: Preliminary explorations with gpt-4v (ision). *arXiv preprint arXiv:2309.17421*, 9(1):1.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. 2022. React: Synergizing reasoning and acting in language models. *arXiv preprint arXiv:2210.03629*.

Denny Zhou, Nathanael Schärli, Le Hou, Jason Wei, Nathan Scales, Xuezhi Wang, Dale Schuurmans, Claire Cui, Olivier Bousquet, Quoc Le, et al. 2022. Least-to-most prompting enables complex reasoning in large language models. *arXiv preprint arXiv:2205.10625*.

Guangming Zhu, Liang Zhang, Youliang Jiang, Yixuan Dang, Haoran Hou, Peiyi Shen, Mingtao Feng, Xia Zhao, Qiguang Miao, Syed Afaq Ali Shah, et al. 2022. Scene graph generation: A comprehensive survey. *arXiv preprint arXiv:2201.00443*.

A Prompts

Listing 1: Scene Graph Generation Prompt

```
This labeling system is designed to assist you in identifying and referring to specific points within each image.
The image is overlaid with a grid matrix to help you with the task.
The bounding boxes, indicating the position of objects in the image, which are represented as [x_min, y_min, x_max, y_max]
      with floating numbers ranging from 0 to 1.
Coordinates of a bounding box are encoded with four values in pixels: [x_min, y_min, x_max, y_max]. x_min and y_min are
      coordinates of the top-left corner of the bounding box.
x_max and y_max are coordinates of bottom-right corner of the bounding box.

Given the image, please generate the scene graph in the following format:
First identify the objects and provide the bounding boxes in the form of {object: [x1, y1, x2, y2]}.
Then, identify the attributes of the objects in the form of {object: [attribute, attribute]}.
Then, identify the realtionship triplet in the form of {Relationship: <object, object>}.
Here is an example.

Object: {object: [x1, y1, x2, y2], object: [x1, y1, x2, y2], ...}
Attribute: {object: [attribute, attribute], object: [attribute, attribute], ...}
Relationship: {Relationship: <object, object>, Relationship: <object, object>, ...}
```

Listing 2: Visual Program Generation Prompt

```
Given a question, first generate a python-like program that describes the reasoning steps required to answer the question step
      -by-step.
You can call two functions in the program: 1. Question() to answer the question; 2. Locate() to locate an object in the image
      with bounding boxes;
Here are some example.

Question: On which side of the walkway leading to the San Francisco Civic Center can the American Flag be found?
def program():
    object = Locate("Walkway leading to the San Francisco Civic Center")
    object = Locate("American Flag")
    result = Question("Which side of the walkway can the American Flag be found?")

Question: Is the surface of the egg next to the handrail at the Big Egg Hunt in Covent Garden London shiny or dull?
def program():
    object = Locate("Handrail at the Big Egg Hunt in Covent Garden London")
    object = Locate("The egg next to the handrail")
    result = Question("Is the surface of the egg shiny or dull?")

Question: %s
```

Listing 3: Aggregation Prompt

```
This labeling system is designed to assist you in identifying and referring to specific points within each image.
The bounding boxes, indicating the position of objects in the image, which are represented as [x1, y1, x2, y2] with floating
      numbers ranging from 0 to 1.
These values correspond to the bottom left x1, top left y1, bottom right x2, and top right y2.

Your goal is to answer the question based on the following inputs:
(1) Question: this is the question you need to answer.
(2) Scene Graph: this represents the structural information of the image.
(3) Rationals: this is a set of QAs that assist you conclude the final answer.

Please first answer the question based on the inputs, and then provide your explanation.

Question: %s
Scene Graph: %s
Rationals: %s
Your Answer:
```

Listing 4: Evaluation Prompt

```
Given a question and a correct answers. Is the following answer correct? Only reply YES or NO.
Question: %s
Correct Answer: %s
Answer you should evaluate: %s
```