Improving Compositional Visual Question-Answering with Chain-of-Thought Reasoning and a Specialized Language Model

Pearl Liu pearlliu@seas.upenn.edu

Olivia Maltz oliviamz@sas.upenn.edu

Hanson Siu siuk@seas.upenn.edu

King Long Tang tangkl@seas.upenn.edu

Abstract

Visual question-answering (VQA) systems often struggle with compositional reasoning. We investigated whether chain-of-thought (CoT) reasoning could improve VQA performance by leveraging specialized language capabilities from a large language model (LLM). Using a subset of the GQA dataset, we implemented two approaches: (1) a multi-agent system combining Gemma for generating sub-questions with LLaVA for visual reasoning, and (2) fine-tuning LLaVA on chain-of-thought examples generated by our multi-agent system. Our best-performing approach improved exact accuracy from 26.79% to 51.56% on our test set, demonstrating that specialized language capabilities from LLMs can enhance VQA systems' compositional reasoning abilities. This work provides insights into improving VQA performance through CoT reasoning without requiring extensive computational resources.

1 Introduction

Our project explored how we can use a large language model (LLM) to improve the performance of a visual question-answering (VQA) system on questions involving relationships between entities (objects or people). We applied and extended established prompt engineering techniques as well as fine-tuned a late fusion vision-language model (VLM) to explore whether we could transfer the specialized language capability of LLM to a VLM.

VQA models like LLaVA-1.5-7b are designed to respond to questions about visual inputs. However, they are far from reliable. To highlight an example, we asked a question about the image in Figure 1: "What is the girl eating?" LLaVA gave the inaccurate answer "Pizza." However, there is no pizza in the image. The girl is eating a fry that she is holding with her hand, and with her other hand she is holding a fork over a salad on a plate in front of her.[1] This example highlights a limitation of VQA models in reasoning about complex scenes, especially about cluttered images or questions requiring discernment of fine-grained detail or synthesizing compositional relationships within an image.

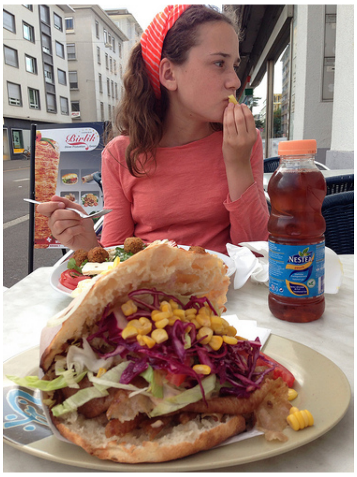

Figure 1: An image of a girl bringing a fry to her mouth with one hand and holding a fork above a plate of salad with her other hand. A bottle of ice tea is also in front of her. On the other side of the table is a sandwich on a plate. The VQA model incorrectly predicts "Pizza" as the answer to the question "What is the girl eating?" for the given image.

Using a subset of the GQA dataset involving questions about relationships between entities in images for evaluation, we wanted to increase the percentage of questions for which our VQA system provided the same answer as the GQA gold label. We observed that LLaVA was unable to perform CoT reasoning by itself, and we wanted to explore whether we could draw on the specialized language capability of an LLM to improve LLaVA's CoT reasoning and thereby its capacity for VQA.

While LLaVA-1.5-7b is a relatively small VLM with noticeable limits on both visual and language reasoning, we wanted to develop a method to improve a VQA system that would not require a huge

[1] The gold label from the GQA dataset for the image is "meal," which summarizes the combination of fries, salad, and beverage that the girl is eating.

amount of time or computing resources. While the approaches that we present do not reach state-of-the-art performance on GQA, they still represent a way to improve, on a limited time and computing budget, a small VLM's performance when prompted to perform chain-of-thought reasoning about compositional visual questions.

2 Literature Review

Recent advances in large multimodal models (LMMs) have shown remarkable progress in VQA tasks. However, these models often struggle with complex reasoning, particularly in zero-shot scenarios where fine-tuning isn't possible. This literature review examines four recent papers that propose novel approaches to enhance LMMs' reasoning capabilities through various CoT and multi-agent methodologies. These works represent significant advances in improving visual reasoning without requiring extensive fine-tuning or annotated data.

In "Compositional Chain-of-Thought Prompting for Large Multimodal Models" (2024), Mitra et al. [1] introduce a zero-shot prompting method called "compositional chain-of-thought prompting" (CCoT). The LMM is first prompted to generate a JSON format scene graph based on an image and task prompt. Then, the LMM is prompted with the image, task prompt, and generated scene graph. Mitra et al. found that CCoT improved the models' performance on compositional and multimodal reasoning benchmarks. Mitra et al. also found that the JSON format of the generated scene graph and the scene graph structure (compared to image captions, which are less structured and condensed) both improved model performance.

Jiang et al. [2], in "Multi-Agent VQA: Exploring Multi-Agent Foundation Models in Zero-Shot Visual Question Answering" (2024), present a multi-agent VQA system. This approach leverages GPT-4V as the main VLM, complemented by specialized agents, like Grounded Segment Everything and CLIP-Count, to handle tasks where the main model falters. If the primary model fails to answer a question, these agents provide additional context that the primary model uses in another attempt. The system showed significant performance gains, with a 20% accuracy drop when the multi-agent pipeline was removed, underscoring the importance of this sequential approach.

Qiu et al.'s "Explainable Knowledge Reasoning via Thought Chains for Knowledge-Based Visual Question Answering" (2024) introduces MuKCoT [3], a model that utilizes large language models to create structured "thought chains" for complex questions. The model guides its reasoning process by breaking down multi-step queries into subquestions. MuKCoT significantly improved relational question accuracy compared to traditional methods. Ablation studies confirmed that the absence of these steps diminished the model's ability to interpret complex relational queries.

In "Visual CoT: Unleashing Chain-of-Thought Reasoning in Multi-Modal Language Models" (2024), Shao et al. [4] propose a visual CoT reasoning method called VisCoT. VisCot involves training a multi-modal language model to select a bounding box that is relevant to answering a question, creating a localized image for detailed analysis. VisCoT outperformed baseline models, demonstrating substantial improvements on datasets like SROIE and DocVQA. Accurate bounding box prediction proved crucial for performance, highlighting how visual attention and structured reasoning bolster LMM capabilities.

These studies showcase diverse yet complementary approaches to enhancing visual reasoning in LMMs. Mitra et al. focus on structured scene graph generation as an intermediate CoT reasoning step, Jiang et al. utilize a multi-agent system with specialized tools, Qiu et al. emphasize thought chains for knowledge-based reasoning, and Shao et al. highlight visual attention through bounding box prediction. Collectively, these methods suggest that combining structured intermediate representations and step-by-step reasoning significantly enhances LMM performance on complex visual tasks.

3 Experimental Design

3.1 Data

Our dataset is derived from the GQA dataset [5], which is designed for visual reasoning and compositional question answering (Hudson and Manning 2019). While the GQA dataset also includes scene graphs, we used only the question and image parts of the GQA dataset. The question data comes in a JSON format with information including image IDs, question types, and answers. GQA was constructed by normalizing Visual Genome Scene Graph annotations, developing a question engine to create a variety of compositional questions based on scene graphs and structural patterns, and generating 22M questions, which was later downsampled to 1.7M balanced questions.

From GQA, we selected questions of the structural type “query” and semantic type “relation” because we wanted to improve our VQA system’s performance on open-ended questions about relationships between objects in images. We randomly selected 8000 questions and images for our training set from GQA’s "train balanced questions," although we ended up using only 5000 examples when fine-tuning LLaVA. We also randomly selected 1000 questions each for dev and test sets from "testdev balanced questions," ensuring no overlap between our dev and test sets. (See table 1.) We excluded questions with missing images.

Train	Dev	Test
5000	1000	1000

Table 1: Dataset Size (# Examples)

Along with using a few GQA training examples in our few-shot prompts in our strong baseline and Extension 1 experiments, we passed our subset of the GQA training data through Extension 1 pipeline to generate the training dataset used to fine-tune LLaVA in Extension 2.

3.2 Evaluation Metric

3.2.1 Exact Accuracy

Our main evaluation metric is the GQA dataset’s accuracy metric, a standard VQA metric which we will refer to as **Exact Accuracy** [5]. Exact Accuracy is the percentage of predictions that exactly match the gold label after normalization. While the strict matching requirement ensures high precision, it also penalizes some correct answers that do not match the annotation. It is calculated as:

$$\textbf{Exact Accuracy =}$$
$$\left(\frac{\text{Number of predictions that exactly match gold label}}{\text{Total number of questions}}\right) * 100\%.$$

3.2.2 Substring Accuracy

To account for a range of correct possible answers, we introduce **Substring Accuracy** as a secondary metric. Substring accuracy is the percentage of predictions that contain the gold label as a substring after normalization. This metric helps us identify cases where the model provides more specific answers (e.g. “handbag” for “bag”) or a sentence instead of GQA’s single word or phrase format. Although substring accuracy can mark incorrect answers as correct, supplementing Exact Accuracy with this metric can offer greater insight into our models’ behavior. It is calculated as:

$$\textbf{Substring Accuracy =}$$
$$\left(\frac{\text{Number of predictions with gold label as a substring}}{\text{Total number of questions}}\right) * 100\%.$$

3.3 Simple Baseline

For our simple baseline, we simply prompted LLaVA with the image and question. The exact accuracy was 26.79% and the substring accuracy was 52.90%.[2] The higher substring accuracy shows that the model often contains the expected answer without being exactly the expected answer, which could be due to the model responding to the question in a sentence instead of GQA’s required single word/phrase format.

4 Experimental Results

4.1 Published Baseline

To establish our strong baseline, we prompted LLaVA to generate and answer sub-questions before answering the original question about an image. First, we prompted LLaVA to generate subquestions for a question.[3] Then, we prompted LLaVA to answer one sub-question about the image at a time. Finally, we prompted LLaVA to answer the question based on the image, subquestions, and answers to sub-questions. For questions where LLaVA did not generate sub-questions when prompted, we set the prediction to an empty string when evaluating the results. The best performing strong baseline prompt had an exact accuracy of 38.95% and a substring accuracy of 44.31%.

4.2 Extensions

4.2.1 Extension 1

Our first extension was similar to the strong baseline, but instead of only prompting LLaVA to perform CoT reasoning, which was ineffective, we used an instruction-tuned LLM, gemma-1.1-2b-it of the Gemma model family, to generate sub-

[2] Adding a formatting instruction to the prompt resulted in improved performance. See appendix A for the full results.

[3] Along with using the same manually created in-context example as in our best-performing extension 1 experiment, we included an explicit instruction since without it, LLaVA would answer the question instead of generating sub-questions. See Table 7 in the Appendices for the strong baseline experiment results and prompts.

Experiment	Exact Acc	Substring Acc
00	44.64 %	50.00 %
01	39.73 %	45.42 %
02	43.08 %	47.66 %
03	33.33 %	**66.67 %**
04	39.40 %	47.21 %
05	41.63 %	48.55 %
06	41.63 %	48.10 %
07	41.07 %	47.77 %
08	**44.92 %**	49.21 %
09	33.71 %	45.87 %
10	37.53 %	48.95 %

Table 2: Extension 1 Experiment Results

questions. We experimented with varying the content of in-context examples, number of in-context examples, number of sub-questions, Gemma generation length, and whether an explicit instruction for sub-question generation was included.

Our best performing variation had one in-context example, a maximum of 50 new tokens for Gemma, and a single-turn conversation format. It outperformed the strong baseline on both metrics with an exact accuracy of 44.94% and a substring accuracy of 49.21%. See Table 2 for the full results and Appendix D for the prompts.

4.2.2 Extension 2

Our second extension focused on enhancing the model's reasoning capabilities through fine-tuning. To create the fine-tuning training data, we put our subset of the GQA training data through our Extension 1 pipeline to generate sub-questions and their answers. The image, question, generated sub-questions, and generated answers were used to fine-tune the VLM to perform CoT reasoning without reliance on a LLM.

After fine-tuning using 8 epochs, we experimented with variations of the most effective Extension 1 prompt. See Appendix E for the prompts. The results, summarized in Table 3, demonstrate a noticeable improvement compared to the baseline and previous extension (as can be seen in Table 4). The best-performing prompt achieved an exact accuracy of 51.56% and a substring accuracy of 53.79%.

Experiment	Exact Acc	Substring Acc
0	41.63%	43.64%
1	45.87%	48.33%
2	44.87%	47.54%
3	49.22%	51.12%
4	48.88%	51.00%
5	**51.56%**	**53.79%**

Table 3: Extension 2 Experiment Results

Experiment Stage	Exact Acc	Substr Acc
Best-Performing Str B	38.95%	44.31%
Best-Performing Ext 1	44.92%	49.21%
Best-Performing Ext 2	51.56%	53.79%

Table 4: Performance Comparison of Best-Performing Strong Baseline, Extension 1, and Extension 2 on Test Set

4.3 Error Analysis

Table 5 presents the model's performance across various VQA task categories. We categorized the questions into four distinct groups: Person Identification, Object Identification, Location, and Other.[4] The model achieves high accuracy on "Other" questions (83%) but shows lower accuracy in Person Identification (38%). This indicates that the model struggles with questions requiring the distinction of individuals within an image. Several factors contribute to the model's errors:

Question Type	**Exact Accuracy**
Person Identification	38%
Object Identification	39%
Location	45%
Other	83%

Table 5: Accuracy across different question types. The model achieves its lowest accuracy in Person Identification.

Focus on Incorrect Entity The model sometimes focuses on the wrong entity within an image. One question asked about the image in Figure 2, "*What is the window covered with?*" The answer is "*blinds*", but the model answered "*brick*", which is in another part of the image.

Dataset Limitations Many errors of our best performing system are due to dataset limitations. For example, it was unclear which entity was referred

[4] A single question can sometimes belong to multiple question types. For example, 'Who is holding the red object?' could be classified as both Person Identification and Object Identification.

to by some questions. One question asked about the image in Figure 3, "*What is the man to the right of the bike wearing?*" There is a man in the background wearing "*shorts*" (the gold label), but the model answered "*jeans*".

Language Understanding Limitations For other questions, the model demonstrated a lack of language capacity. When asked "*What device is on the table?*" about an image with a remote and magazines on a table, the model answered "*magazine*", although it is not a device.

Improvements Over the Strong Baseline Our fine-tuned model demonstrates improvement in certain location-sensitive scenarios compared to our strong baseline. When asked "*What vegetable is to the left of the rice?*" about the image in Figure 4, our strong baseline incorrectly answers "*tomato*", which is in a different location. The fine-tuned model correctly responds with "*lettuce*".

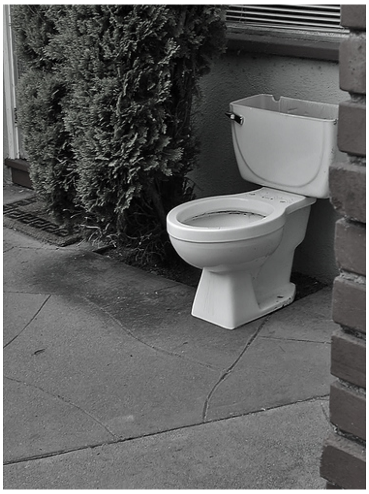

Figure 2: The model incorrectly labels the window covering as "brick" rather than "blinds".

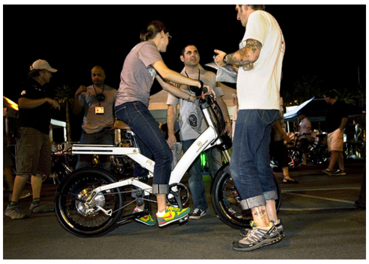

Figure 3: The model answered the question 'What is the man to the right of the bike wearing?" with "jeans" instead of the gold label "shorts".

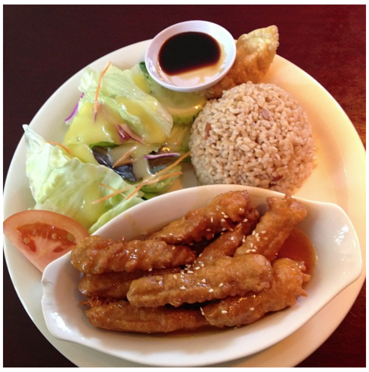

Figure 4: Our model successfully identifies "lettuce" as the vegetable to the left of the rice, outperforming the strong baseline, which predicted "tomato".

5 Conclusions

Our research demonstrates the effectiveness of leveraging an LLM's language capability to improve VQA performance. Starting from a simple baseline of 26.79% exact accuracy, our strong baseline using subquestion-guided reasoning and prompt engineering improved performance to 38.95%. Through two extensions, we achieved significant improvements on our strong baseline. By implementing combining Gemma for sub-question generation with LLaVA for visual reasoning, we improved the exact accuracy to 44.92%, which is 5.97% higher than our strong baseline. By fine-tuning LLaVA on CoT examples generated by our multi-agent system, we further increased the exact accuracy to 51.56%, which is 12.61% higher than our strong baseline.

While our best-performing system falls short of the current state-of-the-art accuracy of 76.04% on the GQA leaderboard [6], our approach offers valuable insights into improving VQA performance with limited computational resources. Our error analysis revealed that the model particularly struggles with object identification (39% accuracy) and person identification (38% accuracy), often due to ambiguous entity references and difficulty in precise object recognition.

The success of our approach suggests that specialized language capabilities from LLMs can effectively enhance VQA systems' compositional reasoning abilities. However, several challenges remain, including handling inconsistent annotations in datasets, improving entity identification, and dealing with semantic nuances between predictions and ground truth answers.

Future work should focus on expanding the training dataset, incorporating visual attention mechanisms, developing evaluation metrics that account for semantic similarity, and improving the model's handling of ambiguous references. These improvements could help bridge the gap between our performance and state-of-the-art VQA systems.

Acknowledgements

We thank Artemis Panagopoulou, our TA, for inspiring and guiding our project, as well as suggesting the Substring Accuracy evaluation metric. We also thank Manu Romero for creating the code for fine-tuning LLaVA that we used in our second extension. [7]

References

[1] Chancharik Mitra et al. "Compositional Chain-of-Thought Prompting for Large Multimodal Models". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. June 2024, pp. 14420–14431. URL: https://openaccess.thecvf.com/content/CVPR2024/html/Mitra_Compositional_Chain-of-Thought_Prompting_for_Large_Multimodal_Models_CVPR_2024_paper.html.

[2] Bowen Jiang et al. "Multi-Agent VQA: Exploring Multi-Agent Foundation Models in Zero-Shot Visual Question Answering". In: *arXiv preprint arXiv:2403.14783* (2024). URL: https://arxiv.org/pdf/2403.14783v1.

[3] Chen Qiu et al. "Explainable Knowledge reasoning via thought chains for knowledge-based visual question answering". In: *Information Processing and Management* (2024). URL: https://doi.org/10.1016/j.ipm.2024.103726.

[4] Hao Shao et al. "Visual CoT: Advancing Multi-Modal Language Models with a Comprehensive Dataset and Benchmark for Chain-of-Thought Reasoning". In: *arXiv preprint arXiv:2403.16999* (2023). URL: https://arxiv.org/pdf/2403.16999.

[5] Drew A Hudson and Christopher D Manning. "GQA: A New Dataset for Real-World Visual Reasoning and Compositional Question Answering". In: *Conference on Computer Vision and Pattern Recognition (CVPR)* (2019). URL: https://arxiv.org/pdf/1902.09506.

[6] EvalAI. *GQA Real-World Visual Reasoning Challenge*. URL: https://eval.ai/web/challenges/challenge-page/225/leaderboard/733.

[7] Manu Romero. *Fine-tune LlaVa-1.5-7B using HuggingFace TR*. URL: https://colab.research.google.com/github/mrm8488/shared_colab_notebooks/blob/master/fine_tune_VLM_LlaVa.ipynb.

[8] Haotian Liu et al. *Improved Baselines with Visual Instruction Tuning*. 2024. arXiv: 2310.03744 [cs.CV]. URL: https://arxiv.org/abs/2310.03744.

[9] Takeshi Kojima et al. *Large Language Models are Zero-Shot Reasoners*. 2023. arXiv: 2205.11916 [cs.CL]. URL: https://arxiv.org/abs/2205.11916.

[10] Mirac Suzgun et al. "Challenging BIG-Bench Tasks and Whether Chain-of-Thought Can Solve Them". In: *Findings of the Association for Computational Linguistics: ACL 2023*. Ed. by Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki. Toronto, Canada: Association for Computational Linguistics, July 2023, pp. 13003–13051. DOI: 10.18653/v1/2023.findings-acl.824. URL: https://aclanthology.org/2023.findings-acl.824.

[11] Laria Reynolds and Kyle McDonell. *Prompt Programming for Large Language Models: Beyond the Few-Shot Paradigm*. 2021. arXiv: 2102.07350 [cs.CL]. URL: https://arxiv.org/abs/2102.07350.

[12] Prompt Engineering Guide. *Prompt Chaining*. URL: https://www.promptingguide.ai/techniques/prompt_chaining.

A Appendix: Simple Baseline

In consultation with our TA and based on a review of the literature on language and multimodal CoT prompting, we experimented with different methods of eliciting an answer in GQA's single word/phrase format. We tried several literature-based formatting instructions: the LLaVA-1.5 paper's "Answer the question using a single word or phrase" formatting prompt [8], Kojima et al's "The answer is" answer extraction phrase [9], and a "Task: Answer the question using a single word or phrase. Question:” format inspired by Mitra et al [1]. Following our TA’s suggestions, we also tested a “short answer” formatting prompt and setting the length penalty to -1 for one of the prompts. However, applying the length penalty to the LLaVA-1.5 paper-based prompt yielded the same performance on both evaluation metrics. See table 6 for the simple baseline experiment results.

Formatting Prompt	Exact Acc (%)	Substr Acc (%)
No formatting prompt	26.79	52.90
“{question} Provide a short answer”	**51.79**	53.57
“Answer the question using a single word or phrase. {question}”	51.23	52.57
“{question} The answer is”	49.67	**54.13**
“Task: Answer the question using a single word or phrase. Question: {question}”	51.12	52.79

Table 6: Simple Baseline Experiment Results

B Appendix: Strong Baseline

See table 7 for the strong baseline experiment results. See below for the prompts.

Experiment	Exact Acc	Substring Acc
0	0.89%	1.12%
1	36.72%	**44.31%**
2	36.61%	41.96%
3	37.61%	43.08%
4	37.95%	42.97%
5	**38.95%**	**44.31%**
6	37.61%	42.86%

Table 7: Strong Baseline Experiment Results

The difference between experiments 1 and 4, 2 and 5, and 3 and 6 is how the sub-questions were parsed from LLaVA’s first response. The former experiments had the sub-questions parsed the same way as Extension 1: treating any sequence of text that ends in a period or question mark as a sub-question. The latter experiments had an additional step of removing sequences of text that end in a period or question mark but only contained the phrase “Sub-questions:”, whitespace, newline, and numbers (i.e. did not actually contain a substantive question or statement). For example, whereas the former parsing method would treat “Sub-questions:\n\n1. What is the tomato inside of?” as two sub-questions, “Sub-questions:\n\n1.” and "What is the tomato inside of?”, the latter parsing method would discard that first “sub-question.” [5] We slightly modified from Extension 1 the way that the sub-questions were parsed from LLaVA’s response since LLaVA was not providing its sub-questions in the format demonstrated by the in-context example without an e.

B.1 Strong Baseline Experiment 0 Sub-question Generation Prompt

first_prompt = (

“USER: <image>\n”

“Question: What piece of furniture is to the right of the pink blanket?” “Sub-questions: Where in the image is the pink blanket? Describe what is to the right of the pink blanket. Where to the right of the pink blanket is the piece of furniture? Describe the piece of furniture. What piece of furniture is it?\n”

“Question: {question} ASSISTANT:”

)

B.2 Strong Baseline Experiments 1 and 4 Sub-question Generation Prompt

first_prompt = (

“USER: <image>\n”

“Question: What piece of furniture is to the right of the pink blanket?” “Sub-questions: Where in the image is the pink blanket? Describe what is to the right of the pink blanket. Where to the right of the pink blanket is the piece of furniture? Describe the piece of furniture. What piece of furniture is it?\”

“Following the format of the above example, break down the following question into a list of sub-questions. Question: {question} ASSISTANT:”

)

B.3 Strong Baseline Experiments 2 and 5 (best-performing) Sub-question Generation Prompt

first_prompt = (

“USER: <image>\n”

“Question: What piece of furniture is to the right of the pink blanket?” “Sub-questions: Where in the image is the pink blanket? Describe what is to the right of the pink blanket. Where to the right of the pink blanket is the piece of furniture? Describe the piece of furniture. What piece of furniture is it?\n”

[5] This is a sample of the output from LLaVA in strong baseline experiment 1.

"Following the format of the above example, generate sub-questions for the following question. Question: {question} ASSISTANT:"

)

B.4 Strong Baseline Experiments 3 and 6 Sub-Question Generation Prompt

first_prompt = (

"USER: <image>\n"

"Question: What piece of furniture is to the right of the pink blanket?" "Sub-questions: Where in the image is the pink blanket? Describe what is to the right of the pink blanket. Where to the right of the pink blanket is the piece of furniture? Describe the piece of furniture. What piece of furniture is it?\"

"Following the format of the above example, break down the following question into sub-questions. Question: {question} ASSISTANT:"

)

B.5 Strong Baseline Sub-question Answering Prompt

intermediate_prompt = "<image>\nQuestion: {sub-question}"

B.6 Strong Baseline Final Answer Prompt

final_prompt = "Question: Use the context: {sub-questions each followed by respective answer to sub-question}. Answer the following question: {question} Provide a short answer."

C Appendix: A Note About the Original Strong Baseline Experiments

The prompts used in our final round of strong baseline experiments were a modified version of the prompts in an earlier round of strong baseline experiments that we performed before working on our first extension. In our initial round of strong baseline experiments, we tried different zero-shot and few-shot CoT prompts that were inspired by language and multimodal CoT prompting literature. Along with trying a standard zero-shot CoT technique based on Kojima et al [9], Mitra et al [1], and Suzgun et al [10], we tried several zero-shot and few-shot prompts intended to elicit sub-question generation and answering as the intermediate CoT reasoning step. These prompts were inspired by Mitra et al [1], Reynolds and McDonell's discussion of metaprompting [11], and guide to prompt chaining [12]. Like Kojima et al [9], we separated reasoning extraction and answer extraction into different prompts. In addition to trying a 2-step prompt format similar to Kojima et al, we tried a 3-step prompt format in which we broke down the reasoning extraction step into two prompts: one for sub-question generation and one for sub-question answering. While we cannot actually use the results from this earlier round of experiments as a baseline due to major differences from our extensions, these experiments influenced our extensions and eventual strong baseline, as well as demonstrated to us that LLaVA could not perform CoT reasoning by itself.

D Appendix: Extension 1

D.1 Gemma Prompts

D.1.1 Extension 1 Experiment 00 Gemma Prompt: 3 in-context examples and default (50) max length for Gemma, single-turn conversation

prompt_template = (

"Question: The cards are on what? "

"Sub-questions: Where in the image are the cards? Describe the cards and its surroundings. Describe what the cards are on.\n"

"Question: What type of bag has the same color as the suitcase? "

"Sub-questions: Where in the image is the suitcase? What color is the suitcase? Where in the image are the bags? Which bag has the same color as the suitcase? Describe the bag. What type of bag is it?\n"

"Question: What piece of furniture is to the right of the pink blanket? "

"Sub-questions: Where in the image is the pink blanket? Describe what is to the right of the pink blanket. Where to the right of the pink blanket is the piece of furniture? Describe the piece of furniture. What piece of furniture is it?\n"

"Question: {question}"

)

D.1.2 Extension 1 Experiment 01 Gemma Prompt: 2 in-context examples and 256 max length for Gemma, single-turn conversation

prompt_template = (

"Question: What type of bag has the same color as the suitcase? "

"Sub-questions: Where in the image is the suitcase? What color is the suitcase? Where in the image the bags? Which bag has the same color as the suitcase? Describe the bag. What type of bag is it?\n"

"Question: What piece of furniture is to the right of the pink blanket? "

"Sub-questions: Where in the image is the pink blanket? Describe what is to the right of the pink blanket. Where to the right of the pink blanket is the piece of furniture? Describe the piece of furniture. What piece of furniture is it?\n"

"Question: {question}"

)

D.1.3 Extension 1 Experiment 02 Gemma Prompt: 1 in-context example and 256 max length for Gemma, single-turn conversation

prompt_template = (

"Question: What piece of furniture is to the right of the pink blanket? "

"Sub-questions: Where in the image is the pink blanket? Describe what is to the right of the pink blanket. Where to the right of the pink blanket is the piece of furniture? Describe the piece of furniture. What piece of furniture is it?\n"

"Question: {question}"

)

D.1.4 Extension 1 Experiment 03 Gemma Prompt: 5 in-context examples, default (50) max length for Gemma, explicit task instructions, different questions, single-turn conversation

prompt_template = (

"The goal is to decompose a complex question into multiple smaller, precise subquestions to enhance clarity and understanding. Do not include instructions, explanations, or any additional text. Here are examples:\n\n "

"Question: The cards are on what?\n"

"Sub-questions: Where are the cards located? What object are the cards placed on? What is the surface supporting the cards? What is holding the cards in place?\n\n"

"Question: What type of bag has the same color as the suitcase?\n"

"Sub-questions: What color is the suitcase? What bags are visible in the image? Which bag matches the suitcase's color? What type of bag is the matching one?\n\n"

"Question: What bag has the same color as the hat the man is wearing?\n"

"Sub-questions: What color is the man's hat? What bags are visible in the image? Which bag matches the hat's color? What type of bag is the matching one?\n\n"

"Question: Who is wearing the clothes?\n"

"Sub-questions: What clothes are visible in the image? Who is wearing these clothes?\n\n"

"Question: Where is the policeman?\n "

"Sub-questions: What objects or landmarks are near the policeman? Where is the policeman located?\n\n"

"Question: {question}\n"

"Sub-questions:"

)

D.1.5 Extension 1 Experiment 04 Gemma Prompt: revise experiment 00 (3 in-context examples) prompt with fewer sub-questions (3 sub-questions per question) and 256 max length for Gemma, single-turn conversation

prompt_template = (

"Question: The cards are on what? "

"Sub-questions: Where in the image are the cards? Describe the cards and its surroundings. Describe what the cards are on.\n"

"Question: What type of bag has the same color as the suitcase? "

"Sub-questions: What color is the suitcase? Which bag has the same color as the suitcase? Describe the bag.\n"

"Question: What piece of furniture is to the right of the pink blanket? "

"Sub-questions: Describe what is to the right of the pink blanket. Where to the right of the pink blanket is the piece of furniture? Describe the piece of furniture.\n"

"Question: {question}"

)

D.1.6 Extension 1 Experiment 05 Gemma Prompt: 4 in-context examples and 256 max length for Gemma, single-turn conversation

prompt_template = (

"Question: The cards are on what? "

"Sub-questions: Where in the image are the cards? Describe the cards and its surroundings. Describe what the cards are on.\n"

"Question: What type of bag has the same color as the suitcase? "

"Sub-questions: Where in the image is the suitcase? What color is the suitcase? Where in the image are the bags? Which bag has the same color as the suitcase? Describe the bag. What type of bag is it?\n"

"Question: What piece of furniture is to the right of the pink blanket? "

"Sub-questions: Where in the image is the pink blanket? Describe what is to the right of the pink blanket. Where to the right of the pink blanket is the piece of furniture? Describe the piece of furniture. What piece of furniture is it?\n"

"Question: What is the girl eating?"

"Sub-questions: Where in the image is the girl? Describe what the girl is eating.\n"

"Question: {question}"

)

D.1.7 Extension 1 Experiment 06 Gemma Prompt: redo experiment 00 (3 in-context examples) with 256 max length for Gemma, single-turn conversation

[Same as experiment 00]

D.1.8 Extension 1 Experiment 07 Gemma Prompt: slightly different 3 in-context examples (last 3 examples from experiment 05) and 256 max length for Gemma, single-turn conversation

prompt_template = (

"Question: What type of bag has the same color as the suitcase? "

"Sub-questions: Where in the image is the suitcase? What color is the suitcase? Where in the image are the bags? Which bag has the same color as the suitcase? Describe the bag. What type of bag is it?\n"

"Question: What piece of furniture is to the right of the pink blanket? "

"Sub-questions: Where in the image is the pink blanket? Describe what is to the right of the pink blanket. Where to the right of the pink blanket is the piece of furniture? Describe the piece of furniture. What piece of furniture is it?\n"

"Question: What is the girl eating? "

"Sub-questions: Where in the image is the girl? Where in the image is what the girl is eating? Describe what the girl is eating. What is the girl eating?"

"Question: {question}"

)

D.1.9 Extension 1 Experiment 08 Gemma prompt: redo experiment 02 (1 in-context example) with default (50) max length for Gemma, single-turn conversation

[Same as experiment 02]

D.1.10 Extension 1 Experiment 09 Gemma prompt: redo experiment 08 (1 in-context example) with 256 max length for Gemma, multi-turn conversation

[Same as experiment 02]

D.1.11 Extension 1 Experiment 10 Gemma prompt: redo experiment 08 (1 in-context example) with default (50) max length for Gemma, multi-turn conversation

[Same as experiment 02]

D.2 LLaVA Prompts

D.2.1 Extension 1 Sub-question Answering Prompt

prompt = "<image>\nQuestion: {subquestion}"

D.2.2 Extension 1 Single-Turn Final Answer Prompt

prompt = "Question: Use the context: {subquestions each followed by respective answer to sub-question}. Answer the following question: {question} Provide a short answer."

E Appendix: Extension 2

E.1 Extension 2 Model Parameters

E.1.1 Extension 2 Experiment 0 Model Parameters

Model Parameters for Prompt 1: max_new_tokens=200, do_sample=True, temperature=0.7, top_p=0.9

Model Parameters for Prompt 2: max_new_tokens=200, do_sample=True, temperature=0.7, top_p=0.9

Model Parameters for Prompt 3: max_new_tokens=50, do_sample=True, temperature=0.7, top_p=0.9

E.1.2 Extension 2 Experiment 1 Model Parameters

Model Parameters for Prompt 1: max_new_tokens=200, do_sample=True, temperature=0.7, top_p=0.9

Model Parameters for Prompt 2: max_new_tokens=100, do_sample=True, temperature=0.7, top_p=0.9

Model Parameters for Prompt 3: max_new_tokens=50, do_sample=True, temperature=0.7, top_p=0.9

E.1.3 Extension 2 Experiment 3 Model Parameters

Model Parameters for Prompt 1: default parameters

Model Parameters for Prompt 2: default parameters

Model Parameters for Prompt 3: default parameters

E.1.4 Extension 2 Experiment 4 Model Parameters

[same as Experiment 3]

E.1.5 Extension 2 Experiment 5 Model Parameters

[same as Experiment 3]

E.2 Extension 2 Prompts

E.2.1 Extension 2 Experiment 0 Prompt

prompt1 = (

"USER: <image>Examples:"

"Question: What piece of furniture is to the right of the pink blanket?\n"

"Sub-questions and sub-question answers: Where in the image are the blankets? The blankets are at the bottom of the image. "

"Where in the image is the pink blanket? The pink blanket is in the bottom left corner of the image. "

"Where to the right of the pink blanket is the piece of furniture? There is a piece of furniture to the right of the pink blanket, in the bottom right corner of the image.\n"

"Following the format of the above examples, generate a set of sub-questions for the question and then answer the sub-questions using the provided image.\n" "Question {question}\nSub-questions: ASSISTANT:"

)

prompt2 = (

"USER: <image>\nAnswer the following questions about the image:{response_to_prompt1}\nASSISTANT:"

)

prompt3 = (

"USER: <image>\n{response_to_prompt2}\n"

"Use the image and above text as context and answer the following question:\n{question}\n"

"Provide a short answer.\nASSISTANT:"

)

E.2.2 Extension 2 Experiment 1 Prompt

prompt1 = (

"USER: <image>\n"

"Question: What piece of furniture is to the right of the pink blanket?\n"

"Sub-questions: Where in the image are the blankets?\n"

"Where in the image is the pink blanket?\n"

"Where to the right of the pink blanket is the piece of furniture?\n"

"Following the format of the above examples, generate a set of sub-questions for the question and then answer the sub-questions using the provided image.\n"

"Question: {question}\nSub-questions: ASSISTANT:"

)

prompt2 = (

"USER: <image>\nQuestion: {sub_question}\nASSISTANT:"

) // Used for each subquestion

prompt3 = (

"USER: <image>\nContext:\n{Q: subquestion1 A: answer1 Q: subquestion2 A: answer2 etc}\n"

"Use the image and above context to answer the following question:\n{question}\n"

"Provide a short answer.\nASSISTANT:"

)

E.2.3 Extension 2 Experiment 2 Prompt

prompt1 = (

"USER: <image>\n"

"Generate subquestions using the provided image and question.\n"

"Question: {question}\nSub-questions: ASSISTANT:"

)

prompt2 = (

"USER: <image>\nQuestion: {sub_question}\nASSISTANT:"

) // used for each subquestion

prompt3 = (

"USER: <image>\nContext:\nQ: subquestion1 A: answer1 Q: subquestion2 A: answer2 etc }\n"

"Use the image and above context to answer the following question:\n{question}\n"

"Provide a short answer.\nASSISTANT:"

)

E.2.4 Extension 2 Experiment 3 Prompt

prompt1 = (

"USER: <image>\n"

"Question: What piece of furniture is to the right of the pink blanket?\n"

"Sub-questions: Where in the image are the blankets?\n"

"Where in the image is the pink blanket?\n"

"Where to the right of the pink blanket is the piece of furniture?\n"

"Following the format of the above examples, generate a set of sub-questions for the question and then answer the sub-questions using the provided image.\n"

"Question: {question}\nSub-questions: ASSISTANT:"

)

prompt2 = (

"USER: <image>\nQuestion: {sub_question}\nASSISTANT:"

) // used for each subquestion

prompt3 = (

"USER: <image>\nContext:\n{Q: subquestion1 A: answer1 Q: subquestion2 A: answer2 etc}\n"

"Use the image and above context to answer the following question:\n{question}\n"

"Provide a short answer.\nASSISTANT:"

)

E.2.5 Extension 2 Experiment 4 Prompt

prompt1 = (

"USER: <image>\n" "Question: What piece of furniture is to the right of the pink blanket? "

"Sub-questions: Where in the image is the pink blanket? Describe what is to the right of the pink blanket. Where to the right of the pink blanket is the piece of furniture? Describe the piece of furniture. What piece of furniture is it?\n"

"Question: {question}\nSub-questions: ASSISTANT:"

)

prompt2 = (

"USER: <image>\nQuestion: {subquestion}\nASSISTANT:"

) // used for each subquestion

prompt3 = (

"USER: <image>\nContext:{Q: subquestion1 A: answer1 Q: subquestion2 A: answer2 etc}\n"

"Use the image and above context to answer the following question:\n {question}\n"

"Provide a short answer.\nASSISTANT:"

)

E.2.6 Extension 2 Experiment 5 Prompt (best-performing)

prompt1 = (

"USER: <image>\n"

"Question: What piece of furniture is to the right of the pink blanket?\n"

"Sub-questions: Where in the image are the blankets?\n"

"Where in the image is the pink blanket?\n"

"Where to the right of the pink blanket is the piece of furniture?\n"

"Following the format of the above examples, generate a set of sub-questions for the question and then answer the sub-questions using the provided image.\n"

"Question: {question}\nSub-questions: ASSISTANT:"

)

prompt2 = (

"USER: <image>\nQuestion: {subquestion}\nASSISTANT:"

) // used for each subquestion

prompt3 = (

"USER: <image>\nContext:\n{Q: subquestion1 A: answer1 Q: subquestion2 A: answer2 etc}\n"

"Use the image and above context to answer the following question:\n{question}\n"

"Provide a short answer.\nASSISTANT:"

)