From the Least to the Most: Building a Plug-and-Play Visual Reasoner via Data Synthesis

Chuanqi Cheng[1,2*] **Jian Guan**[2*] **Wei Wu**[2†] **Rui Yan**[1†]
[1]Gaoling School of Artificial Intelligence, Renmin University of China [2]Ant Group
{chengchuanqi, ruiyan}@ruc.edu.cn
{jianguanthu, wuwei19850318}@gmail.com

Abstract

We explore multi-step reasoning in vision-language models (VLMs). The problem is challenging, as reasoning data consisting of multiple steps of visual and language processing are barely available. To overcome the challenge, we first introduce a least-to-most visual reasoning paradigm, which interleaves steps of decomposing a question into sub-questions and invoking external tools for resolving sub-questions. Based on the paradigm, we further propose a novel data synthesis approach that can automatically create questions and multi-step reasoning paths for an image in a bottom-up manner. Our approach divides the complex synthesis task into a few simple sub-tasks, and (almost entirely) relies on open-sourced models to accomplish the sub-tasks. Therefore, the entire synthesis process is reproducible and cost-efficient, and the synthesized data is quality guaranteed. With the approach, we construct 50k visual reasoning examples. Then, we develop a visual reasoner through supervised fine-tuning, which is capable of generally enhancing the reasoning abilities of a wide range of existing VLMs in a plug-and-play fashion. Extensive experiments indicate that the visual reasoner can consistently and significantly improve four VLMs on four VQA benchmarks. Our code and dataset are available at `https://github.com/steven-ccq/VisualReasoner`.

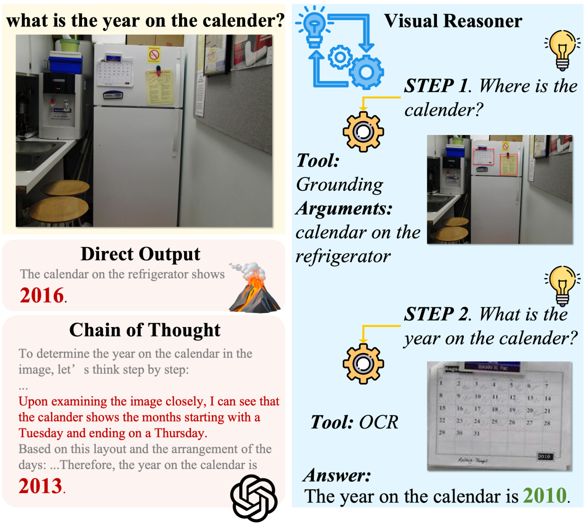


Figure 1: **Top Left:** An example from TextVQA (Singh et al., 2019) with “2010” as the ground truth. **Middle Left:** Response from LLaVA-NeXT-13B (Liu et al., 2024a). **Bottom Left:** Response from GPT-4o with the prompt “{*question*} please think step by step and answer the question”. **Right:** Response given by the proposed method, which is also the only correct answer.

1 Introduction

Large language models (LLMs) (Brown et al., 2020; Achiam et al., 2023) have demonstrated remarkable performance across various tasks in natural language processing (NLP). Encouraged by the success, the artificial intelligence community is now enthusiastically exploring ways to enable LLMs to process information from modalities beyond language, leading to a recent surge in the development of large multimodal models (LMMs) (Alayrac et al., 2022; Liu et al., 2024b; Team et al., 2023; Dai et al., 2024; Bai et al., 2023; Ormazabal et al., 2024). By adopting a pre-training and instruction tuning paradigm, representations of multiple modalities are effectively fused in deep architectures, bringing substantial advancements in tasks, such as image captioning (Young et al., 2014), visual question answering (VQA) (Goyal et al., 2017), and optical character recognition (OCR) (Mishra et al., 2019), among others.

In this work, we explore vision language models (VLMs) as a typical example of LMMs. Despite the rapid progress, state-of-the-art VLMs still face challenges in reasoning over visual content. As exemplified in Figure 1, intuitively, the question can be correctly solved following a “least-to-most” paradigm (Zhou et al., 2022), in which the ques-

* Equal Contribution.
† Corresponding Authors.

Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing, pages 4941–4957
November 12-16, 2024

tion is decomposed into a series of sub-questions, and an answer is deduced by resolving the sub-questions step by step. However, existing VLMs are inept at performing such multi-step reasoning because (1) Multi-step reasoning paths, like the one shown in Figure 1 (right), are rarely included in the training data (Dai et al., 2024)[1]. The VLMs have few opportunities to develop the reasoning capability from the subsequent post-training. And (2) different from reasoning over text, solving questions in a vision-language context may require manipulating the input image (e.g., marking a specific area) and deducting the next steps from both textual and visual intermediate results. The requirement, however, is difficult to accomplish for most VLMs, whether open-sourced or proprietary.

We present *least-to-most visual reasoning*, a general paradigm to guide VLMs to decompose a given question into sub-questions and invoke tools to resolve each sub-question for handling diverse visual reasoning tasks. While there have been extensive studies for LLMs regarding chain-of-thought reasoning (Wei et al., 2022b; Yao et al., 2022, 2024; Wang et al., 2022) and tool-invoking (Schick et al., 2024; Qin et al., 2023b), the techniques are less explored in the context of VLMs. Since data scarcity is a major obstacle, we propose a novel data synthesis approach, dubbed *least-to-most synthesis*, to automatically generate a "(question, reasoning path)" tuple for a given image in a bottom-up manner. Specifically, the pipeline of *least-to-most synthesis* consists of four steps: (1) *Entity Recognition:* recognizing all entities in an image; (2) *Node Construction:* constructing three types of nodes, each aggregating an image with a few entities and some textual features; (3) *Reasoning Process Synthesis:* synthesizing a reasoning path from a sampled chain of nodes. Based on the nodes, the reasoning path is formed by connecting a sequence of sub-questions and tool arguments generated by an LLM; and (4) *Question Synthesis:* generating the main question by recursively combining the sub-questions in the reasoning path through an LLM. Our approach (almost entirely) relies on open-sourced models. Therefore, it offers several advantages, including cost-efficiency, reproducibility, and ensured data quality, over the common practice where data are obtained by querying powerful proprietary LMMs like GPT-4V (Qi et al., 2024; Li et al., 2024).

Based on *least-to-most synthesis*, we build a large scale **VI**sual **RE**as**O**ning dataset (VIREO) with 50k examples, and tailor LLaVA-1.5-7B (Liu et al., 2023a) as a visual reasoner through supervised fine-tuning on VIREO. The reasoner can be generally applied to off-the-shelf VLMs in a plug-and-play fashion to enhance their reasoning capabilities. We conduct experiments with four representative VLMs as showcases. Evaluation results across four VQA benchmarks indicate that the reasoner can consistently improve all VLMs over all tasks, with absolute performance gains ranging from 0.71% to 39%.

Our contributions are three-fold:

I. We introduce the *least-to-most visual reasoning* paradigm to synergize question-decomposition and tool-invoking in VLMs for solving complex vision-language tasks.

II. We propose *least-to-most synthesis*, a reproducible, cost-efficient, and data quality-assured algorithm for automatically creating multi-step visual reasoning data (almost) using open-source models.

III. We use *least-to-most synthesis* to construct the VIREO dataset of 50k examples for fine-tuning a reasoner model[2]. Extensive experiments illustrate that the reasoner can consistently and significantly enhance existing VLMs in a plug-and-play fashion across five VQA benchmarks.

2 Related Works

2.1 Vision-Language Models

Building LMMs aims to enable foundation models to seamlessly handle multimodal signals, such as language, vision, and audio. Among the efforts, significant attention has been focused on jointly modeling vision and language, known as VLMs. Recent work on VLMs can be broadly categorized into two groups, according to how visual information is incorporated into the models. The first line integrates visual information into LLMs via a vision-language connector. For example, LLaVA series (Liu et al., 2023b,a) exploit a linear transformation or an MLP to transform outputs from a vision encoder into inputs of a language model. BLIP-2 (Li et al., 2023a) and InstructBLIP (Dai et al., 2024) rely on Query Transformers to achieve vision-language alignment.

[1] A few datasets, such as A-OKVQA (Schwenk et al., 2022), VCR (Zellers et al., 2019), and ScienceQA (Lu et al., 2022), contain rationales. However, the rationales merely provide explanations for the answers and thus differ significantly from the multi-step reasoning data we study in the work.

[2] We have released a larger version of VIREO comprising 500k examples to spur research in the visual reasoning area.

Similarly, Qwen-VL (Bai et al., 2023) and mPLUG-Owl (Ye et al., 2023) use learnable tokens to take visual information into account. CogVLM (Wang et al., 2023a) maps vision embedding to the space of word embedding by an MLP adapter, and then enables deep vision-language feature alignment via a visual expert module. The second group strives to train VLMs natively from image tokens and textual tokens. Among the representative models, Flamingo (Alayrac et al., 2022) inserts gated cross-attention dense blocks into a pre-trained LM. BEIT-3 (Huang et al., 2024) utilizes Multiway Transformers as the backbone to encode various modalities. KOSMOS-1 (Huang et al., 2024) and KOSMOS-2 (Peng et al., 2023) interleave image tokens with textual tokens in the input sequence through a designed format. Gemini family (Team et al., 2023; Reid et al., 2024) take multimodal signals as input and can natively generate images using discrete image tokens. Instead of building a new VLM, we target enhancing the multi-step reasoning ability of existing VLMs. Our data synthesis approach enables us to develop a visual reasoner that can be generally applied to various VLMs, leading to consistent improvements over various VQA tasks.

2.2 Reasoning and Tool Use

Reasoning is an important emergent ability of LLMs (Wei et al., 2022a). With appropriate exemplars or prompts, LLMs can demonstrate "chain-of-thought" (CoT) behavior and solve problems through multi-step reasoning (Wei et al., 2022b; Kojima et al., 2022). Encouraged by the observation, significant efforts have been made to improve the reasoning capabilities of LLMs. For example, Zhou et al. (2022) propose least-to-most prompting for complex reasoning; Yao et al. (2022) extend the ability of LLMs by interleaving reasoning and acting. Wen et al. (2024) generate code-form plans before low-level reasoning. In addition to innovations in methodology, reasoning abilities have also proven effective in various applications, such as table understanding (Wang et al., 2023c), math problem solving (Wei et al., 2022b), question answering (Guan et al., 2024), and decision making (Yao et al., 2022). Very recently, the research community begins to investigate the problem in multimodal models (Qi et al., 2024; Yang et al., 2023; Wang et al., 2024). Different from existing work, we focus on data synthesis with open-sourced models, and develop a plug-an-play visual reasoner.

Our work also relates to the efforts on facilitating LLMs to leverage tools (Qin et al., 2023b; Schick et al., 2024; Shen et al., 2024; Qin et al., 2023a). The difference is that our tools are specially selected for visual processing, with the goal of enhancing the reasoning capabilities of VLMs.

2.3 Task Decomposition for Visual Reasoning

Various approaches have been proposed to tackle complex vision-language tasks through task decomposition and step-by-step reasoning. For example, Visual Programming (Gupta and Kembhavi, 2023) utilizes LLMs to perform visual reasoning by breaking down tasks into subroutines. Similarly, ViperGPT (Surís et al., 2023) and CodeVQA (Subramanian et al., 2023) decompose tasks to generate executable code. However, these methods heavily rely on the strong instruction-following capabilities of LLMs, making them less effective with smaller models. Furthermore, these approaches are susceptible to instability arising from prompt designs, choice and ordering of demonstrations, and LLM selection, even when employing powerful models like GPT-3 (Zhao et al., 2021). Our work deviates significantly from these methods by focusing on fine-tuning VLMs on large-scale synthesized datasets, systematically enhancing the model's capability to handle complex tasks. This approach enables us to achieve competitive performance using smaller, open-source models (e.g., 7B or 13B parameters), making our method more cost-effective, accessible, and suitable for practical deployment with fewer computational resources.

3 Method

We elaborate on our method for multi-step reasoning in VLMs. First, we formalize *least-to-most visual reasoning* that delineates how a visual reasoner solves a complex problem according to an image (§3.1). Then, we present details of *least-to-most synthesis* by which a VLM can be tuned as the visual reasoner to perform reasoning (§3.2).

3.1 Least-to-Most Visual Reasoning

We formalize the *least-to-most visual reasoning* paradigm as follows: Given an image I and a question Q, a visual reasoner $\mathcal{M}_R$ deduces a multi-step reasoning path R, where each step either performs operations on I (e.g., marking an area with a red box) or asks an off-the-shelf VLM $\mathcal{M}$ to conclude a final answer. To this end, we represent R as a chain of invoking tools from a pre-defined pool $\mathcal{T} = \{t^i | i = 1, 2, \cdots, T\}$ step by step, where each tool refers to a class of operation on images or the VLM $\mathcal{M}$, as illustrated in Figure 2 (right).

Reasoning Process. At the k-th step, $\mathcal{M}_R$ proposes a sub-question q_k and selects a tool t_k from $\mathcal{T}$ based on an image I_k and previous steps:

$$q_k, t_k = \mathcal{M}_R(I_k, Q, \{q_{<k}\}), \quad (1)$$

where t_k is a textual description specifying the invoked tool and the corresponding arguments. Then, the answer to q_k is deduced by t_k, denoted as r_k, based on which I_{k+1} is obtained as follows:

$$I_{k+1} = \begin{cases} r_k, & \text{if } r_k \text{ is an image,} \\ I_k, & \text{otherwise.} \end{cases} \quad (2)$$

Particularly, we set I_1 to I. If I_k includes a red box to mark some area smaller than a threshold α, and t_k intends to infer information from I_k (including the OCR and Answer tools in Table 1), we automatically crop this area from original I_k and enlarge it to the same size as I_k. The above process iterates until t_k refers to the VLM $\mathcal{M}$, in which case we define the final answer $A = r_k$ and terminate the reasoning process. In summary, we formally denote $R = [(I_k, q_k, t_k)]_{k=1}^{K}$,

Tools. Following human experience, we define four tools, each targeting a class of atomic problems and outputs either a modified image or a piece of text. Tab. 1 describes details.

Specifically, we implement Grounding and Highlight using GroundingDino (Liu et al., 2023c), OCR using PaddleOCR[3], and Answer using the VLM $\mathcal{M}$. The reasoner is trained to invoke these tools to dive into the details of the given image. By varying $\mathcal{M}$ in Answer, the reasoner can adapt to different VLMs in a plug-and-play fashion, and significantly enhance their performance across a wide range of tasks, as will be shown in §4.

3.2 Least-to-Most Synthesis

To overcome the data barrier, a common practice is to feed image-question pairs to powerful proprietary LMMs like GPT-4V, and gather the outputs as a dataset (Qi et al., 2024). The top-down approach, however, suffers from several issues: (1) Even powerful proprietary models like GPT-4V still struggle to perform reliable reasoning (Wu et al., 2023), implying that the quality of data obtained in this way is not guaranteed. (2) Utilizing proprietary models incurs high costs, hindering the scalability of the synthesized data. (3) It is difficult for others to reproduce the method since the behavior of proprietary models can vary over time.

In contrast to the top-down approach, we propose a bottom-up pipeline that can synthesize multi-step visual reasoning data using (almost entirely) open-sourced models while ensuring the quality of the synthesized data. The workflow begins with an image, gradually generates sub-questions associated with tools and intermediate results, and ultimately synthesizes a question based on the image and the reasoning path. Specifically, the pipeline comprises four steps: *entity recognition*, *node construction*, *reasoning path synthesis*, and *question synthesis*, where each node aggregates a focused (sub-)image and relevant text-form information. Figure 2 (Left) illustrates the construction process.

Entity Recognition. We employ Deformable DETR (Zhu et al., 2020) to recognize entities in an image, which can identify 1,203 types of entities. In practice, we discard those entities with confidence scores $\leqslant 0.5$.

Node Construction. Given an image, we automatically construct nodes based on the recognized entities. Each node is represented as a pair of an image and a textual profile, where the image is extracted from the given one, and the profile is an attribute-value dictionary about the image. By converting the image into a textual profile, data synthesis can eliminate the reliance on visual signals, allowing the use of more advanced LLMs instead of VLMs in subsequent processes. As indicated in Figure 2, we define three types of nodes spanning various granularities: **(1) Single-Entity Node.** The image for such a node refers to one recognized entity. We utilize specialized tools to extract accurate and fine-grained attributes from multiple dimensions (e.g., color) to form the profile. More details are presented in Appendix A. **(2) Entity-Group Node.** The image of an entity-group node is the aggregation of multiple recognized entities that are close to each other. We employ BLIP[4] to caption the image as the corresponding profile of the node. Each caption contains about 10-20 tokens, illustrating the inter-entity relations in detail. **(3) Whole-Image Node.** This type of node corresponds to the

[3] https://github.com/PaddlePaddle/PaddleOCR

[4] https://huggingface.co/Salesforce/blip-image-captioning-large

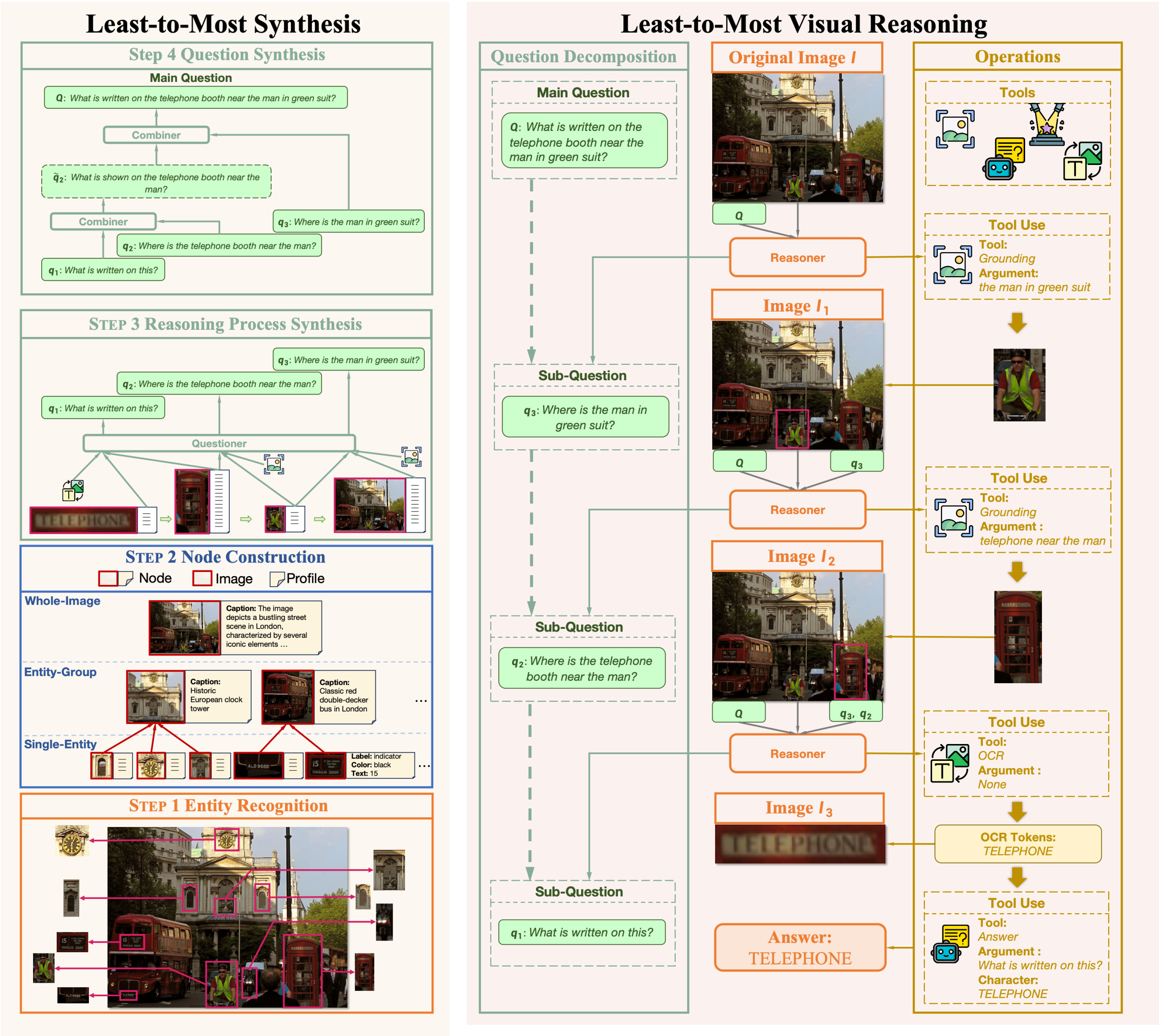


Figure 2: **Left:** the pipeline of *least-to-most synthesis*. **Right:** the process of *least-to-most visual reasoning*.

whole image given in advance. We use LLaVA[5] to caption the image in detail as its profile, typically exceeding 200 tokens in length, which offers comprehensive but coarse-grained information.

Reasoning Process Synthesis. We sample a chain of M nodes from the constructed node set, denoted as $(N_1, N_2, \cdots, N_M)$, which will be connected in turn to form the reasoning process. We craft elaborate rules to ensure nodes can be reasonably connected and the last node N_M is a whole-image node, as detailed in Appendix A. For every two adjacent nodes N_m and N_{m+1} in the chain ($m \leqslant M-1$), based on their profiles and a sampled tool t_m, we exploit an LLM as a `Questioner` to synthesize one sub-question q_m to ask about a certain attribute of the head node N_m conditioned on the tail node N_{m+1}:

$$q_m, \hat{t}_m = \texttt{Questioner}(N_m^P, N_{m+1}^P, t_m), \quad (3)$$

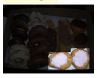
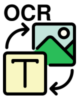
where N_m^P and N_{m+1}^P refer to the profiles of N_m and N_{m+1}, respectively, $\hat{t}_m$ is the argument of the specified tool t_m to solve q_m. Iterating the process w.r.t. m, we obtain $M-1$ sub-questions and synthesize the entire reasoning process R as $[(I_{M-1}, q_{M-1}, \hat{t}_{M-1}), \ldots, (I_1, q_1, \hat{t}_1)]$, where I_m is the image of N_m ($m = 1, \cdots, M-1$).

Question Synthesis. We generate the main question Q by recursively combining the sub-questions using another LLM as a `Combiner`:

$$Q = \tilde{q}_{M-1}, \quad (4)$$

$$\tilde{q}_m = \begin{cases} q_1, & m = 1, \\ \texttt{Combiner}(\tilde{q}_{m-1}, q_m), & m > 1, \end{cases} \quad (5)$$

[5] `https://huggingface.co/llava-hf/llava-v1.6-vicuna-13b-hf`

Tool	Argument		Output	Purpose
Grounding 	Image 	Target Entity `The woman in a purple ski suit.`	Image 	Frame an area from the Image corresponding to the Target Entity
Highlight 	Image 	Target Entity `Pink Cookie`	Image 	Highlight all entities corresponding to the Target Entity in the Image
OCR 	Image 		List of Text `[“CURLY DICK RD”, “Tarana 13”, “Oberon 39”]`	Recognize all pieces of text from the Image

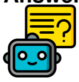
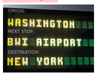
Answer 	Image 	Question
		`What is the next stop?`
		Character
		`[“NEXT STOP”, “BWI AIRPORT”]`
			Answer
			`As shown in the image, the next stop is “BWI STOP”.`
				Answer the Question based on the Image and recognized Character. And Character can be empty.

Table 1: Tools used in our reasoner.

where $\tilde{q}_m$ is an intermediate result for the m-th step. Note that $\{\tilde{q}_m\}_{m=2}^{M-2}$ are excluded from R, as they are just used for synthesis of Q.

We implement `Questioner` and `Combiner` by fine-tuning LLaMA-3-8B-Instruct[6]. Due to limited resources, we obtain the training data by querying GPT-4 (Wang et al., 2023b) using a few high-quality demonstrations as seeds following the self-instruct pipeline (Wang et al., 2023b). Since the two tasks are relatively simple, querying GPT-4 only 10k times is enough to achieve satisfactory performance. More details are in Appendix B.

Compared to the top-down approach, our *least-to-most synthesis* is reproducible and significantly more cost-efficient. Moreover, every step in the pipeline is atomic, ensuring the performance of the open-sourced models on these simple tasks. Therefore, the synthesized data is guaranteed to be of high quality, as will be demonstrated in §4.

4 Experiments

To assess the efficacy of the proposed method, we fine-tune an LLaVA-1.5-7B[7] as the Reasoner, and plug the Reasoner into four representative VLMs with various sizes and conduct experiments on four standard VQA benchmarks. In the implementation, we obtain 10k training examples for `Question` and `Combiner`, respectively, by querying GPT-4. Subsequently, we synthesize a visual reasoning dataset (VIREO) with 50k examples following the least-to-most synthesis approach, and perform instruction tuning on VIREO to obtain the Reasoner. More implementation details are presented in Appendix B.

4.1 Models for the Answer Tool

To illustrate the versatility of our Reasoner, we employ several VLMs with different sizes and architectures as the Answer tool, including **(1) BLIP-2** (Li et al., 2023a): It utilize a Q-Former module to integrate visual and textual information. We use the 2.7B version[8], which is the smallest model in our experiments. **(2) InstructBLIP** (Dai et al., 2024): It expands BLIP-2 by incorporating instruction prompts into the Q-Former module. We use the 7B[9] and 13B[10] versions in our experiments. **(3) LLaVA** (Liu et al., 2024a): It hinges on a basic projection layer to align image and text representations. We use the 13B version[11].

4.2 Evaluation Datasets

We conduct experiments on the following datasets: **(1) GQA** (Hudson and Manning, 2019): It is a VQA dataset constructed from knowledge graphs, primarily focusing on inter-entity attribute relationships. **(2) TextVQA** (Singh et al., 2019) and **ST-VQA** (Biten et al., 2019): The two datasets include textual information within images, used to evaluate

[6] https://huggingface.co/meta-llama/Meta-Llama-3-8B-Instruct

[7] https://huggingface.co/llava-hf/llava-1.5-7b-hf

[8] https://huggingface.co/Salesforce/blip2-opt-2.7b

[9] https://huggingface.co/Salesforce/instructblip-vicuna-7b

[10] https://huggingface.co/Salesforce/instructblip-vicuna-13b

[11] https://huggingface.co/llava-hf/llava-v1.6-vicuna-13b-hf

the capability to understand text in pictorial form. **(3) TallyQA** (Acharya et al., 2019): It is widely used to assess the counting ability, divided into a simple subset and a complex subset. The complex subset of TallyQA involves more fine-grained attributes of the entities in the images than the simple subset. In our experiments, we denote these subsets as Tally-S and Tally-C, respectively.

Since the VLMs used for the Answer tool are general-purpose generative models and not fine-tuned on the evaluation datasets, they may produce correct answers but include additional information (e.g., "The answer is") beyond the ground truth provided by the datasets. Therefore, we use Exact Matching (EM) as the metric only for TallyQA. For GQA and TextVQA, we use answer recall, i.e., whether the output includes the ground truth, for evaluation. For ST-VQA, we submit results to its official website.

4.3 Baselines

To comprehensively compare our approach with alternative task decomposition strategies, we devise a baseline termed "Tools" inspired by Gupta and Kembhavi (2023). Specifically, we employ a few-shot approach to decompose the initial problem into a sequence of tool-calling operations (e.g., PaddleOCR and GroundingDINO), which are subsequently executed. Notably, we leverage LLaMA-3-8B-Instruct (Dubey et al., 2024) for decomposition, ensuring a comparable parameter count to the LLaVA-7B model used by our Reasoner.

4.4 Main Results

Table 2 reports evaluation results. We find that **(1) The Reasoner consistently improves the performance of all VLMs across all datasets.** By decomposing questions and invoking specialized tools, the Reasoner can improve various off-the-shelf VLMs in a plug-and-play fashion, suggesting its strong generalization ability. **(2) The Reasoner helps better capture complex inter-entity relations.** The Reasoner brings clear improvements on GQA, which involves diverse relations among entities in the images. The improvements could result from the least-to-most synthesis algorithm in which such relations are implicitly modeled. The improvement for LLaVA (3.63%) is more substantial than other weaker VLMs (1.17%-1.95%). It is possibly because stronger VLMs may better realize their potential with the Reasoner considerably alleviating the difficulty in locating the target entity. **(3) The Reasoner boosts the understanding of words in images.** The OCR tool empowers the Reasoner to effectively enhance the performance on both TextVQA and ST-VQA. On TextVQA, we notice that the enhancement is less significant on LLaVA than on weaker VLMs, possibly because LLaVA is less dependent on external tools for understanding the words in images. **(4) The Reasoner improves the counting performance by a large margin.** The benefit of the Reasoner is particularly significant on TallyQA. Without the Reasoner, the base VLMs are sensitive to irrelevant information in the images and thus easily miscount. By utilizing the Highlight tool, the Reasoner can reliably mitigate the impact of such irrelevant information.

4.5 Quality Assessment of VIREO

We conduct a human study to investigate the quality of our synthesized VIREO dataset. Specifically, we randomly sample 200 examples from VIREO, and recruit 3 graduate students majoring in NLP to manually check the correctness of the synthesized question and reasoning process for each instance. The check focuses on the correctness in three aspects: **(1) Sub-Question.** The sub-question must be accurately faithful to the profile of the tail node and can be resolved using the specified tool. **(2) Argument.** Based on the current sub-question, the annotators need to assess whether using the given argument to invoke the tool can yield the expected output. **(3) Main Question.** The annotators judge whether the main question covers all sub-questions and whether it can be decomposed into sub-questions in the correct order. Each example receives 3 labels from each of the three annotators on each aspect. Results show that the three evaluators consistently approve that all examples are correct in all aspects[12] , suggesting the highly guaranteed quality of the synthesized data.

5 Discussions

Although the main results affirmed the high quality of VIREO and the general benefits of the Reasoner to enhance VLMs on multiple benchmarks, we are still curious about the following research questions: **(1) RQ1:** How does the capability of the Reasoner vary w.r.t. the size of instructions for fine-tuning? **(2) RQ2:** What is the relative impact of each integrated tool on the Reasoner's overall capability? **(3)**

[12]This might be surprising, but we double-checked the evaluation and confirmed the results.

Model	Size	GQA	TextVQA	ST-VQA	TallyQA-S	TallyQA-C
BLIP2	**2.7B**	40.85	32.65	18.89	20.93	29.00
+ Tools	-	38.26 (-1.17)	34.54 (+2.11)	-	49.77 (+28.84)	34.46 (+5.46)
+ Reasoner	-	**42.02** (+1.17)	**36.94** (+4.29)	**19.60** (+0.71)	**56.38** (+35.45)	**39.38** (+10.38)
InstructBLIP	**7B**	51.65	38.46	24.85	67.05	38.84
+ Tools	-	50.33 (-1.32)	42.87 (+4.41)	-	68.64 (+1.59)	50.25 (+11.41)
+ Reasoner	-	**53.60** (+1.95)	**43.42** (+4.96)	**26.36** (+1.51)	**74.99** (+7.94)	**57.78** (+18.94)
InstructBLIP	**13B**	52.72	37.61	22.21	58.38	23.46
+ Tools	-	51.92 (-0.80)	**39.65** (+2.04)	-	69.29 (+10.91)	51.37 (+27.91)
+ Reasoner	-	**54.24** (+1.48)	**42.03** (+4.42)	**23.72** (+1.51)	**74.98** (+16.60)	**59.73** (+36.27)
LLaVA	**13B**	57.02	57.04	-	70.29	29.69
+ Tools	-	54.19 (-2.83)	58.91 (+1.87)	-	77.01 (+6.72)	61.67 (+31.98)
+ Reasoner	-	**60.65** (+3.63)	**59.22** (+2.18)	-	**80.28** (+9.99)	**68.65** (+38.96)

Table 2: Evaluation results with different VLMs as the Answer tools. **Size** refers to the number of parameters of the language model in each VLM. We do not provide the result of LLaVA on ST-VQA because its output always includes abundant information, leading to nonsensical low scores using the official evaluation website. We highlight the best results in **bold**.

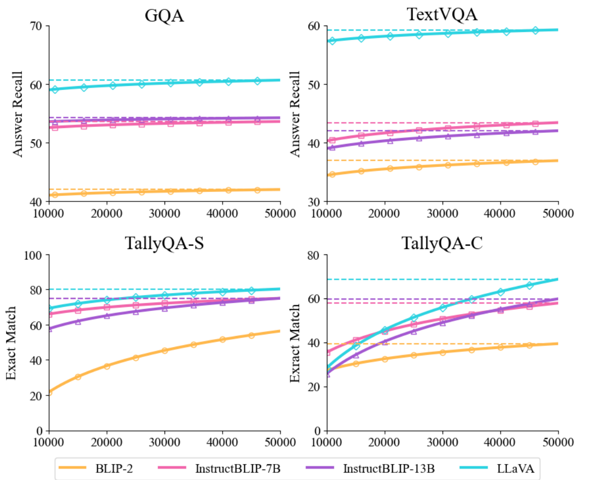


Figure 3: The performance of the Reasoner varies with the size of VIREO. The dashed lines indicate the performance of corresponding models using 50k training examples.

Model	GQA	TextVQA	Tally-S	Tally-C
BLIP2	**42.02**	**36.94**	**56.38**	**39.38**
w/o OCR	40.26	32.30	54.89	37.89
w/o Highlight	39.34	36.30	20.45	28.66
w/o Grounding	40.44	31.98	20.89	28.84
InstBLIP-7B	**53.60**	**43.42**	**74.99**	**57.78**
w/o OCR	50.09	37.20	73.17	55.78
w/o Highlight	48.90	41.78	65.68	38.49
w/o Grounding	51.03	37.84	67.01	38.67
InstBLIP-13B	**54.24**	**42.03**	**74.98**	**59.73**
w/o OCR	50.25	36.56	73.19	55.78
w/o Highlight	48.90	41.70	57.51	25.11
w/o Grounding	51.82	37.46	58.61	25.14
LLaVA	**60.65**	**59.22**	**80.28**	**68.65**
w/o OCR	53.86	55.89	77.28	64.53
w/o Highlight	51.82	37.46	58.61	25.14
w/o Grounding	56.51	58.09	70.22	33.04

Table 3: Evaluation results of removing different tools.

RQ3: Will the Reasoner bring a negative impact on visual tasks other than VQA? **(4) RQ4:** Is the Reasoner still effective on more advanced VLMs? **(5) RQ5:** What are the underlying causes of the Reasoner's errors on the benchmarks?

5.1 RQ1: Influence of Data Size

Figure 3 presents the model performance when we vary the size of VIREO. Overall, the gain in performance gradually increases with the number of training examples. However, we also notice that the marginal benefit is diminishing, suggesting that more diverse data synthesis approaches besides ours are needed to further break through the upper limit in future work.

5.2 RQ2: Ablation Studies Regarding Tools

To gauge the significance of the diverse tools employed in our approach, we conducted a thorough ablation study, systematically investigating the impact of removing each tool on model performance. The findings, presented in Table 3, unequivocally demonstrate that the exclusion of any single tool adversely affects the model's capabilities across various datasets. This observation underscores the multifaceted nature of visual tasks, which often necessitate the synergistic application of multiple sophisticated operations, rendering the reliance on a solitary tool insufficient.

Model	Size	MMMU	POPE
InstructBLIP	**7B**	22.43	84.79
+ Reasoner	-	22.50 (+0.07)	84.79 (+0.00)
InstructBLIP	**13B**	18.75	80.42
+ Reasoner	-	18.23 (-0.52)	80.42 (+0.00)

Table 4: Results on broader visual tasks beyond VQA.

5.3 RQ3: Extending to Broader Visual Tasks

To investigate whether introducing a Reasoner will hurt the performance on other visual tasks, we conduct experiments on MMMU (Yue et al., 2024) and POPE (Li et al., 2023b) as examples. The MMMU benchmark aggregates massive multi-discipline tasks necessitating abundant knowledge to address, which is beyond the scope of our Reasoner. POPE is a discrimination dataset for hallucination detection which requires discriminating whether a certain coarse-grained object is present in a given image without the need for multi-step reasoning. We choose InstructBLIP-7B and -13B as the base VLMs for discussion.

As shown in Table 4, introducing the Reasoner on MMMU has a minimal impact on the final performance, suggesting that the Reasoner does not influence the expression of VLMs' inherent knowledge. On the other hand, the presence or absence of the Reasoner on POPE yields consistent results, indicating that additional reasoning over images does not increase the likelihood of the VLMs generating hallucinations.

5.4 RQ4: Efficacy on More Advanced VLMs

Considering that more advanced VLMs such as Qwen-VL (Bai et al., 2023) have huge potential to inherit the function of external tools such as perceiving bounding boxes in images, we convert VIREO into an end-to-end format by sequentially combining the input and output of each step in order to avoid the computation overhead from explicit tool invocation. Table 5 shows that the Qwen-VL model, fine-tuned on the end-to-end version of VIREO, significantly improves across all datasets. Notably, the application of our Reasoner does not cause a noticeable increase in time cost compared to the vanilla model (1.4s v.s. 1.1s per sample).

5.5 RQ5: Regarding Error Analysis

To analyze the error types of the Reasoner, we randomly sample 100 instances where the model prediction is wrong from all evaluation sets. Then,

Model	GQA	TextVQA	TallyQA-S	TallyQA-C
Qwen-VL	57.95	67.26	75.38	37.92
+ Reasoner	61.73	74.12	81.50	68.40

Table 5: Evaluation results on Qwen-VL.

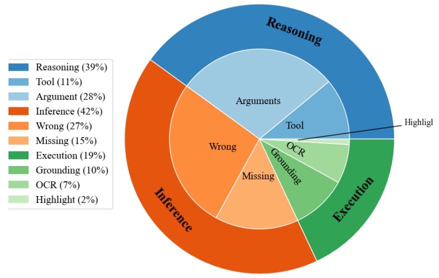


Figure 4: The distribution of different error types.

we analyze their error and categorize the errors into three types: **(1) Reasoning:** The Reasoner uses a wrong tool (*Tool*) or generates wrong arguments for the tool (*Arguments*); **(2) Execution:** The *Grounding*, *OCR*, or *Highlight* tool returns wrong execution results; and **(3) Inference:** The Answer tool outputs wrong answers (*Wrong*) or irrelevant answers with the question (*Missing*).

As shown in Figure 4, the "Inference" error has the largest proportion (42%). Among these, there are even 15% of instances where the Answer tool generates irrelevant answers, indicating the huge room to improve the instruction-following ability of existing VLMs. Additionally, the "Reasoning" error accounts for a significant proportion (39%), and most of the instances (28%) are attributed to wrong arguments. This means VLMs' tool-invoking capability is far from perfect, yet this ability is crucial for VLMs to interact with the environment. The "Execution" error is less frequent, implying the huge potential of building general and powerful VLMs based on specialized visual tools.

6 Conclusions

We propose a data synthesis approach to enhancing multi-step reasoning capabilities of vision-language models. The approach decomposes the synthesis task into several simple sub-tasks, and finish the sub-tasks (almost) with open-sourced models. Based on the approach, we build a large-scale visual reasoning dataset, and develop a visual reasoner by tuning on the dataset. Evaluation results across four VQA benchmarks indicate that the visual reasoner can generally improve the reasoning capabilities of a number of existing VLMs.

Limitations

We select COCO2014 (Lin et al., 2014) as the source of images for our data synthetic process. However, while COCO2014 is a general-purpose image dataset, we do not guarantee that its data can cover all visual tasks. Additionally, our proposed method demonstrates consistent improvements across four VQA benchmarks, but this does not imply that our method will be effective in all visual datasets and scenarios. Furthermore, during our experiments, we only selected four VLMs as base models, and we cannot ensure that our method is capable of bringing improvements to all models.

Ethical Considerations

Images may contain sensitive information. Using or publishing datasets that include such information could pose potential ethical risks. In our experiments, we strictly control the image sources of our data, utilizing only authorized open-source datasets. Furthermore, all the models we used are publicly accessible and adhere to established requirements.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (NSFC Grant No. 62122089), Beijing Outstanding Young Scientist Program NO. BJJWZYJH012019100020098, and Intelligent Social Governance Platform, Major Innovation & Planning Interdisciplinary Platform for the "Double-First Class" Initiative, Renmin University of China, the Fundamental Research Funds for the Central Universities, and the Research Funds of Renmin University of China.

References

Manoj Acharya, Kushal Kafle, and Christopher Kanan. 2019. Tallyqa: Answering complex counting questions. In *Proceedings of the AAAI conference on artificial intelligence*, volume 33, pages 8076–8084.

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*.

Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katherine Millican, Malcolm Reynolds, et al. 2022. Flamingo: a visual language model for few-shot learning. *Advances in neural information processing systems*, 35:23716–23736.

Jinze Bai, Shuai Bai, Shusheng Yang, Shijie Wang, Sinan Tan, Peng Wang, Junyang Lin, Chang Zhou, and Jingren Zhou. 2023. Qwen-vl: A frontier large vision-language model with versatile abilities. *arXiv preprint arXiv:2308.12966*.

Ali Furkan Biten, Ruben Tito, Andres Mafla, Lluis Gomez, Marçal Rusinol, Ernest Valveny, CV Jawahar, and Dimosthenis Karatzas. 2019. Scene text visual question answering. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 4291–4301.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

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.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

Yash Goyal, Tejas Khot, Douglas Summers-Stay, Dhruv Batra, and Devi Parikh. 2017. Making the v in vqa matter: Elevating the role of image understanding in visual question answering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6904–6913.

Jian Guan, Wei Wu, Zujie Wen, Peng Xu, Hongning Wang, and Minlie Huang. 2024. Amor: A recipe for building adaptable modular knowledge agents through process feedback. *arXiv preprint arXiv:2402.01469*.

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.

Shaohan Huang, Li Dong, Wenhui Wang, Yaru Hao, Saksham Singhal, Shuming Ma, Tengchao Lv, Lei Cui, Owais Khan Mohammed, Barun Patra, et al. 2024. Language is not all you need: Aligning perception with language models. *Advances in Neural Information Processing Systems*, 36.

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.

Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. 2022. Large language models are zero-shot reasoners. *Advances in neural information processing systems*, 35:22199–22213.

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

Yifan Li, Yifan Du, Kun Zhou, Jinpeng Wang, Wayne Xin Zhao, and Ji-Rong Wen. 2023b. Evaluating object hallucination in large vision-language models. *arXiv preprint arXiv:2305.10355*.

Zejun Li, Ruipu Luo, Jiwen Zhang, Minghui Qiu, and Zhongyu Wei. 2024. Vocot: Unleashing visually grounded multi-step reasoning in large multi-modal models. *arXiv preprint arXiv:2405.16919*.

Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. 2014. Microsoft coco: Common objects in context. In *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13*, pages 740–755. Springer.

Haotian Liu, Chunyuan Li, Yuheng Li, and Yong Jae Lee. 2023a. Improved baselines with visual instruction tuning.

Haotian Liu, Chunyuan Li, Yuheng Li, Bo Li, Yuanhan Zhang, Sheng Shen, and Yong Jae Lee. 2024a. Llava-next: Improved reasoning, ocr, and world knowledge.

Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. 2023b. Visual instruction tuning.

Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. 2024b. Visual instruction tuning. *Advances in neural information processing systems*, 36.

Shilong Liu, Zhaoyang Zeng, Tianhe Ren, Feng Li, Hao Zhang, Jie Yang, Chunyuan Li, Jianwei Yang, Hang Su, Jun Zhu, et al. 2023c. Grounding dino: Marrying dino with grounded pre-training for open-set object detection. *arXiv preprint arXiv:2303.05499*.

Pan Lu, Swaroop Mishra, Tanglin Xia, Liang Qiu, Kai-Wei Chang, Song-Chun Zhu, Oyvind Tafjord, Peter Clark, and Ashwin Kalyan. 2022. Learn to explain: Multimodal reasoning via thought chains for science question answering. *Advances in Neural Information Processing Systems*, 35:2507–2521.

Anand Mishra, Shashank Shekhar, Ajeet Kumar Singh, and Anirban Chakraborty. 2019. Ocr-vqa: Visual question answering by reading text in images. In *2019 International Conference on Document Analysis and Recognition (ICDAR)*, pages 947–952. IEEE Computer Society.

Aitor Ormazabal, Che Zheng, Cyprien de Masson d'Autume, Dani Yogatama, Deyu Fu, Donovan Ong, Eric Chen, Eugenie Lamprecht, Hai Pham, Isaac Ong, et al. 2024. Reka core, flash, and edge: A series of powerful multimodal language models. *arXiv preprint arXiv:2404.12387*.

Zhiliang Peng, Wenhui Wang, Li Dong, Yaru Hao, Shaohan Huang, Shuming Ma, and Furu Wei. 2023. Kosmos-2: Grounding multimodal large language models to the world. *arXiv preprint arXiv:2306.14824*.

Ji Qi, Ming Ding, Weihan Wang, Yushi Bai, Qingsong Lv, Wenyi Hong, Bin Xu, Lei Hou, Juanzi Li, Yuxiao Dong, et al. 2024. Cogcom: Train large vision-language models diving into details through chain of manipulations. *arXiv preprint arXiv:2402.04236*.

Yujia Qin, Shengding Hu, Yankai Lin, Weize Chen, Ning Ding, Ganqu Cui, Zheni Zeng, Yufei Huang, Chaojun Xiao, Chi Han, et al. 2023a. Tool learning with foundation models. *arXiv preprint arXiv:2304.08354*.

Yujia Qin, Shihao Liang, Yining Ye, Kunlun Zhu, Lan Yan, Yaxi Lu, Yankai Lin, Xin Cong, Xiangru Tang, Bill Qian, et al. 2023b. Toolllm: Facilitating large language models to master 16000+ real-world apis. *arXiv preprint arXiv:2307.16789*.

Machel Reid, Nikolay Savinov, Denis Teplyashin, Dmitry Lepikhin, Timothy Lillicrap, Jean-baptiste Alayrac, Radu Soricut, Angeliki Lazaridou, Orhan Firat, Julian Schrittwieser, et al. 2024. Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context. *arXiv preprint arXiv:2403.05530*.

Timo Schick, Jane Dwivedi-Yu, Roberto Dessì, Roberta Raileanu, Maria Lomeli, Eric Hambro, Luke Zettlemoyer, Nicola Cancedda, and Thomas Scialom. 2024. Toolformer: Language models can teach themselves to use tools. *Advances in Neural Information Processing Systems*, 36.

Dustin Schwenk, Apoorv Khandelwal, Christopher Clark, Kenneth Marino, and Roozbeh Mottaghi. 2022. A-okvqa: A benchmark for visual question answering using world knowledge. In *European Conference on Computer Vision*, pages 146–162. Springer.

Yongliang Shen, Kaitao Song, Xu Tan, Dongsheng Li, Weiming Lu, and Yueting Zhuang. 2024. Hugginggpt: Solving ai tasks with chatgpt and its friends in hugging face. *Advances in Neural Information Processing Systems*, 36.

Amanpreet Singh, Vivek Natarajan, Meet Shah, Yu Jiang, Xinlei Chen, Dhruv Batra, Devi Parikh, and Marcus Rohrbach. 2019. Towards vqa models that can read. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8317–8326.

Sanjay Subramanian, Medhini Narasimhan, Kushal Khangaonkar, Kevin Yang, Arsha Nagrani, Cordelia

Schmid, Andy Zeng, Trevor Darrell, and Dan Klein. 2023. Modular visual question answering via code generation. *arXiv preprint arXiv:2306.05392*.

Dídac Surís, Sachit Menon, and Carl Vondrick. 2023. Vipergpt: Visual inference via python execution for reasoning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 11888–11898.

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*.

Weihan Wang, Qingsong Lv, Wenmeng Yu, Wenyi Hong, Ji Qi, Yan Wang, Junhui Ji, Zhuoyi Yang, Lei Zhao, Xixuan Song, et al. 2023a. Cogvlm: Visual expert for pretrained language models. *arXiv preprint arXiv:2311.03079*.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc V Le, Ed H Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. 2022. Self-consistency improves chain of thought reasoning in language models. In *The Eleventh International Conference on Learning Representations*.

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*.

Yizhong Wang, Yeganeh Kordi, Swaroop Mishra, Alisa Liu, Noah A Smith, Daniel Khashabi, and Hannaneh Hajishirzi. 2023b. Self-instruct: Aligning language models with self-generated instructions. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 13484–13508.

Zilong Wang, Hao Zhang, Chun-Liang Li, Julian Martin Eisenschlos, Vincent Perot, Zifeng Wang, Lesly Miculicich, Yasuhisa Fujii, Jingbo Shang, Chen-Yu Lee, et al. 2023c. Chain-of-table: Evolving tables in the reasoning chain for table understanding. In *The Twelfth International Conference on Learning Representations*.

Jason Wei, Yi Tay, Rishi Bommasani, Colin Raffel, Barret Zoph, Sebastian Borgeaud, Dani Yogatama, Maarten Bosma, Denny Zhou, Donald Metzler, et al. 2022a. Emergent abilities of large language models. *Transactions on Machine Learning Research*.

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

Jiaxin Wen, Jian Guan, Hongning Wang, Wei Wu, and Minlie Huang. 2024. Codeplan: Unlocking reasoning potential in large langauge models by scaling code-form planning. *arXiv preprint arXiv:2409.12452*.

Yang Wu, Shilong Wang, Hao Yang, Tian Zheng, Hongbo Zhang, Yanyan Zhao, and Bing Qin. 2023. An early evaluation of gpt-4v (ision). *arXiv preprint arXiv:2310.16534*.

Zhengyuan Yang, Linjie Li, Jianfeng Wang, Kevin Lin, Ehsan Azarnasab, Faisal Ahmed, Zicheng Liu, Ce Liu, Michael Zeng, and Lijuan Wang. 2023. Mm-react: Prompting chatgpt for multimodal reasoning and action. *arXiv preprint arXiv:2303.11381*.

Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Tom Griffiths, Yuan Cao, and Karthik Narasimhan. 2024. Tree of thoughts: Deliberate problem solving with large language models. *Advances in Neural Information Processing Systems*, 36.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik R Narasimhan, and Yuan Cao. 2022. React: Synergizing reasoning and acting in language models. In *The Eleventh International Conference on Learning Representations*.

Qinghao Ye, Haiyang Xu, Guohai Xu, Jiabo Ye, Ming Yan, Yiyang Zhou, Junyang Wang, Anwen Hu, Pengcheng Shi, Yaya Shi, et al. 2023. mplug-owl: Modularization empowers large language models with multimodality. *arXiv preprint arXiv:2304.14178*.

Peter Young, Alice Lai, Micah Hodosh, and Julia Hockenmaier. 2014. From image descriptions to visual denotations: New similarity metrics for semantic inference over event descriptions. *Transactions of the Association for Computational Linguistics*, 2:67–78.

Xiang Yue, Yuansheng Ni, Kai Zhang, Tianyu Zheng, Ruoqi Liu, Ge Zhang, Samuel Stevens, Dongfu Jiang, Weiming Ren, Yuxuan Sun, et al. 2024. Mmmu: A massive multi-discipline multimodal understanding and reasoning benchmark for expert agi. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9556–9567.

Rowan Zellers, Yonatan Bisk, Ali Farhadi, and Yejin Choi. 2019. From recognition to cognition: Visual commonsense reasoning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 6720–6731.

Zihao Zhao, Eric Wallace, Shi Feng, Dan Klein, and Sameer Singh. 2021. Calibrate before use: Improving few-shot performance of language models. In *International conference on machine learning*, pages 12697–12706. PMLR.

Denny Zhou, Nathanael Schärli, Le Hou, Jason Wei, Nathan Scales, Xuezhi Wang, Dale Schuurmans, Claire Cui, Olivier Bousquet, Quoc V Le, et al. 2022. Least-to-most prompting enables complex reasoning in large language models. In *The Eleventh International Conference on Learning Representations*.

Xizhou Zhu, Weijie Su, Lewei Lu, Bin Li, Xiaogang Wang, and Jifeng Dai. 2020. Deformable detr: Deformable transformers for end-to-end object detection. *arXiv preprint arXiv:2010.04159*.

A Details of Data Construction.

Attributes of Single-Entity Node. Each single-entity node consists of five attributes: *Label, Location, Color, Text, and Size.* Specifically, *Label* and *Location* are directly derived from the recognition results of Deformable DETR. *Color* is determined using ColorThief [13] to analyze the dominant color scheme of the subgraph corresponding to the single-entity node. *Text* is obtained by applying PaddleOCR to recognize textual content within the image. *Size* records the node's proportion in the entire image, considering its width, height, and area.

Construction of Chain. We use tools that produce graphical outputs (e.g., Grounding, Highlight) to connect intermediary nodes and tools that produce textual outputs (e.g., OCR, Answer) to connect terminal nodes.

We construct the chain by sequentially adding nodes. Specifically, we maintain a queue $L = (N_1, N_2, \ldots, N_i)$ consisting of nodes. To add N_{i+1}, we first specify the tool t_i to be used and sample a node in the remaining set $\hat{N} = \{N_j \notin L\}$ that allow the use of this tool. For instance, only nodes with text in images permit using the OCR tool. If $\hat{N} = \emptyset$ or the chain length reaches the limit, the process terminates. When constructing VIREO, we set the maximum chain length to 4. Consequently, the dataset includes reasoning paths with lengths of 2, 3, and 4.

B Implementation Details

Image Source. We use the COCO2014 dataset (Lin et al., 2014) as our source of images.

Questioner. We use LLaMA-3-8B-Instruct as the base model for the `Questioner`. To construct the corresponding training data, we first manually create 5 seed prompts for each combination of head node and tail node ($3 \times 3 = 9$). Then, we generate a total of 10k instances by using GPT-4. During this process, to reduce the bias introduced by the seed prompts, we gradually add the obtained results into the prompt pool. For each instance, three prompts are randomly selected from the prompt pool as demonstrations. For each instance, the input contains 3 fields: the profiles of the head node and tail node, and the specified tool. Output includes 2 fields: the question and the arguments of the tool. We perform instruction fine-tuning on the Questioner using LoRA. The rank is set to 8 and the lora_alpha is set to 8. We adopt AdamW as the optimizer. We set adam_beta1, adam_beta2, and adam_epslion to 0.9, 0.999, and 1e-8, respectively. We use the cosine schedule to warm up the training and set warmup_steps to 0.1. We set the batch size to 4 and fine-tune Questioner for 2 epochs, with each epoch taking around 1 hour. The training process is completed with 8 Nvidia A100 GPUs.

[13] `https://lokeshdhakar.com/projects/color-thief/`

Model	Size	VLM-only	Reasoner
BLIP2	2.7B	0.05	1.40
InstructBLIP	7B	0.19	1.70
InstructBLIP	13B	0.28	1.79
LLaVA	13B	0.57	2.08

Table 6: The average per-sample time cost (seconds) when using only the VLM and applying the Reasoner.

Combiner. Similar to `Questioner`, we also use LLaMA-3-8B-Instruct as the base model for the `Combiner`. We manually create 20 seed prompts and generate a total of 10k instances by using GPT-4. The input includes two questions to be merged, and the output is the merged result. The fine-tuning settings of `Combiner` keep the same as `Questioner`.

Reasoner. We synthesize 50k data examples using the least-to-most synthesis method. Each example includes the current image I, the main question Q, and the previous sub-questions $\{q_{<k}\}$. The label for each data example comprises the current sub-question q_k and the tool t_k that needs to be invoked. We use this data to train LLaVA-1.5-7B as the Reasoner. We perform instruction fine-tuning on the Reasoner using LoRA. The rank is set to 8 and the lora_alpha is set to 8. We adopt AdamW as the optimizer. We set adam_beta1, adam_beta2, and adam_epslion to 0.9, 0.999, and 1e-8, respectively. We use the cosine schedule to warm up the training and set warmup_steps to 0.1. We set the batch size to 8 and fine-tune the Reasoner for 3 epochs, with each epoch taking around 3 hours.

The training process is completed with 1 Nvidia A100 GPU.

C Additional Time Cost

While the Visual Reasoner offers significant performance gains across various tasks, it is crucial to acknowledge the computational overhead associated with its multi-step reasoning process. As each step necessitates additional inference time, the overall inference duration is extended compared to using a vanilla VLM. Table 6 presents a quantitative analysis of the average per-sample time cost using a single Nvidia A100 GPU.

D Case Study

To facilitate understanding of our method, we show some cases in Figure 5, Figure 6, and Figure 7.

E Example of Data Construction

To visualize our least-to-most pipeline, we show the data construction process in Figure 8.

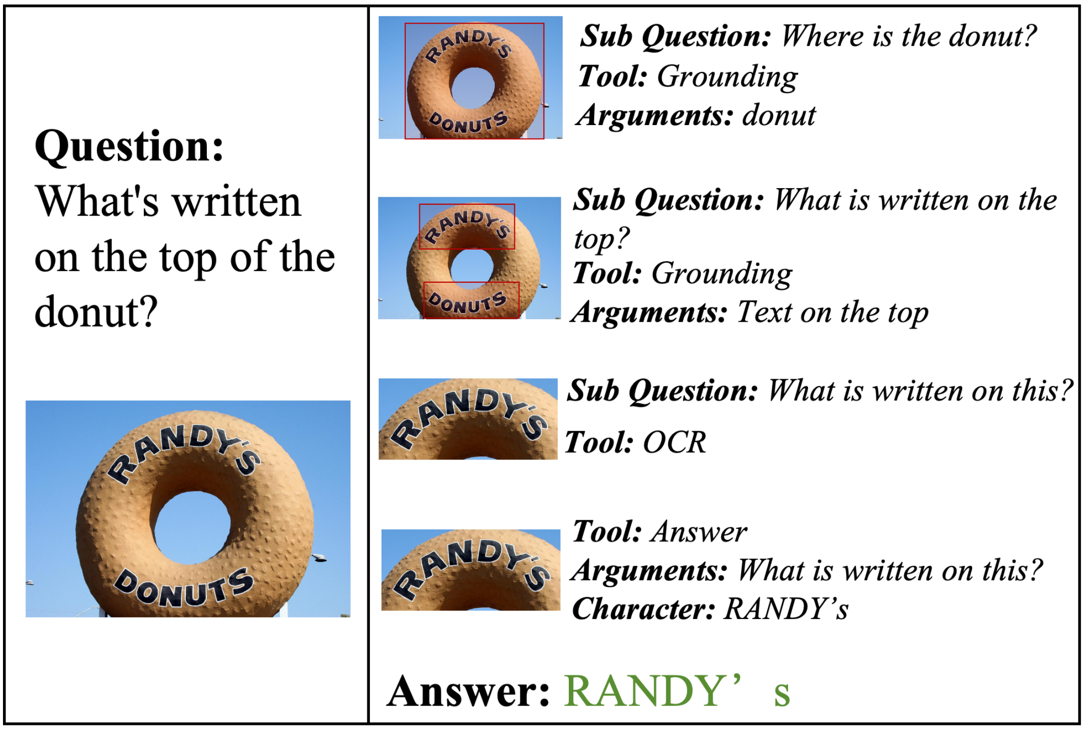


Figure 5: Case 1. This case uses Grounding to locate the donut and uses OCR and Answer to get the final answer.

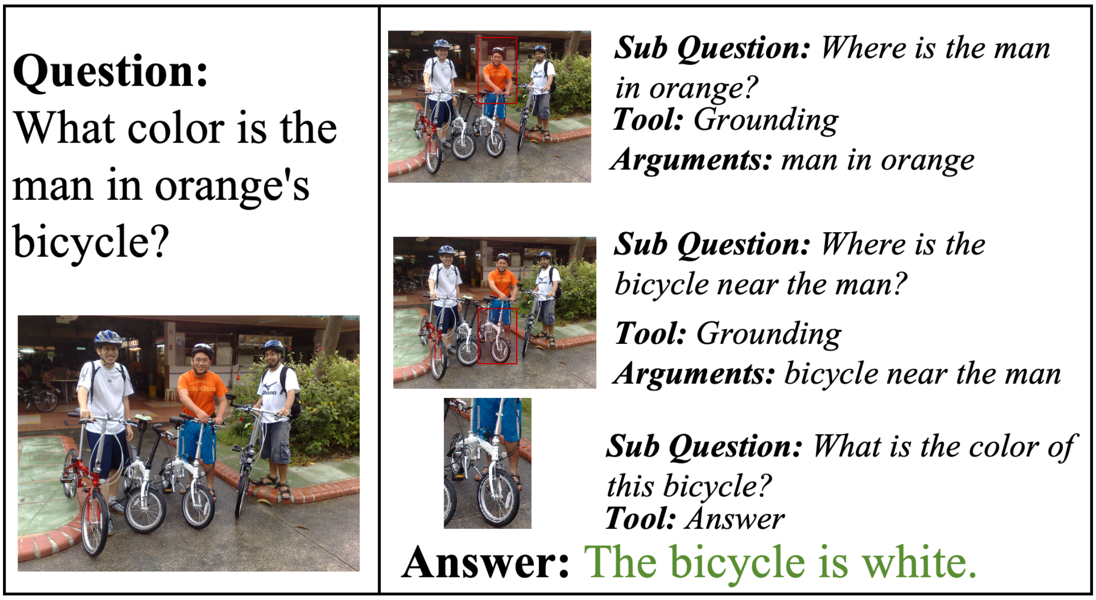


Figure 6: Case 2. In this case, the man wearing orange is first precisely identified, and then attention is directed to the bicycle near him to obtain the answer.

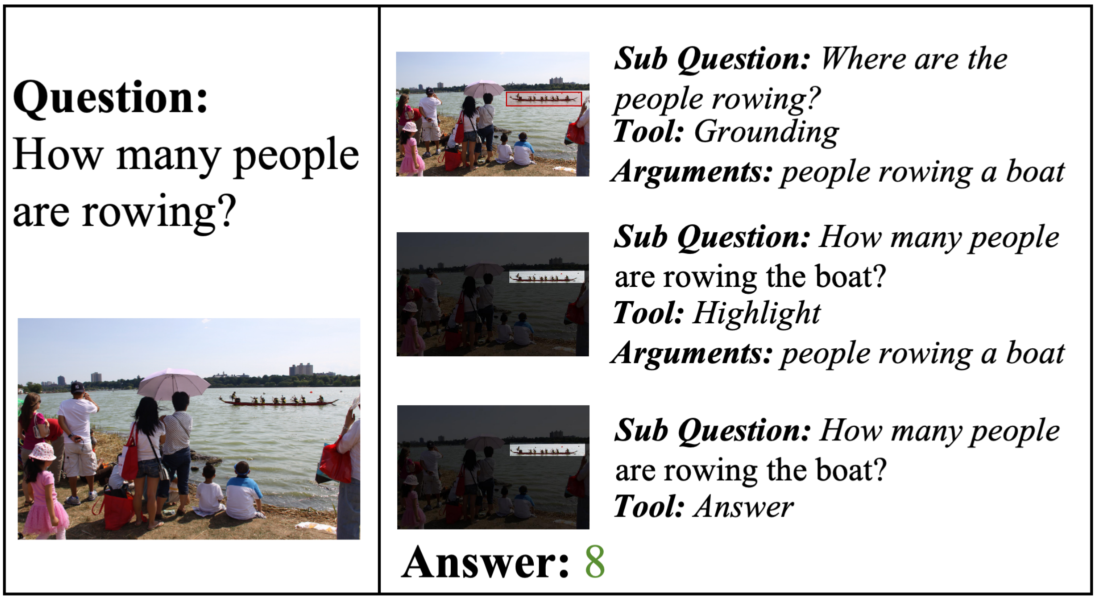


Figure 7: Case 3. This case excludes irrelevant people and highlights the target group, thereby achieving accurate counting results.

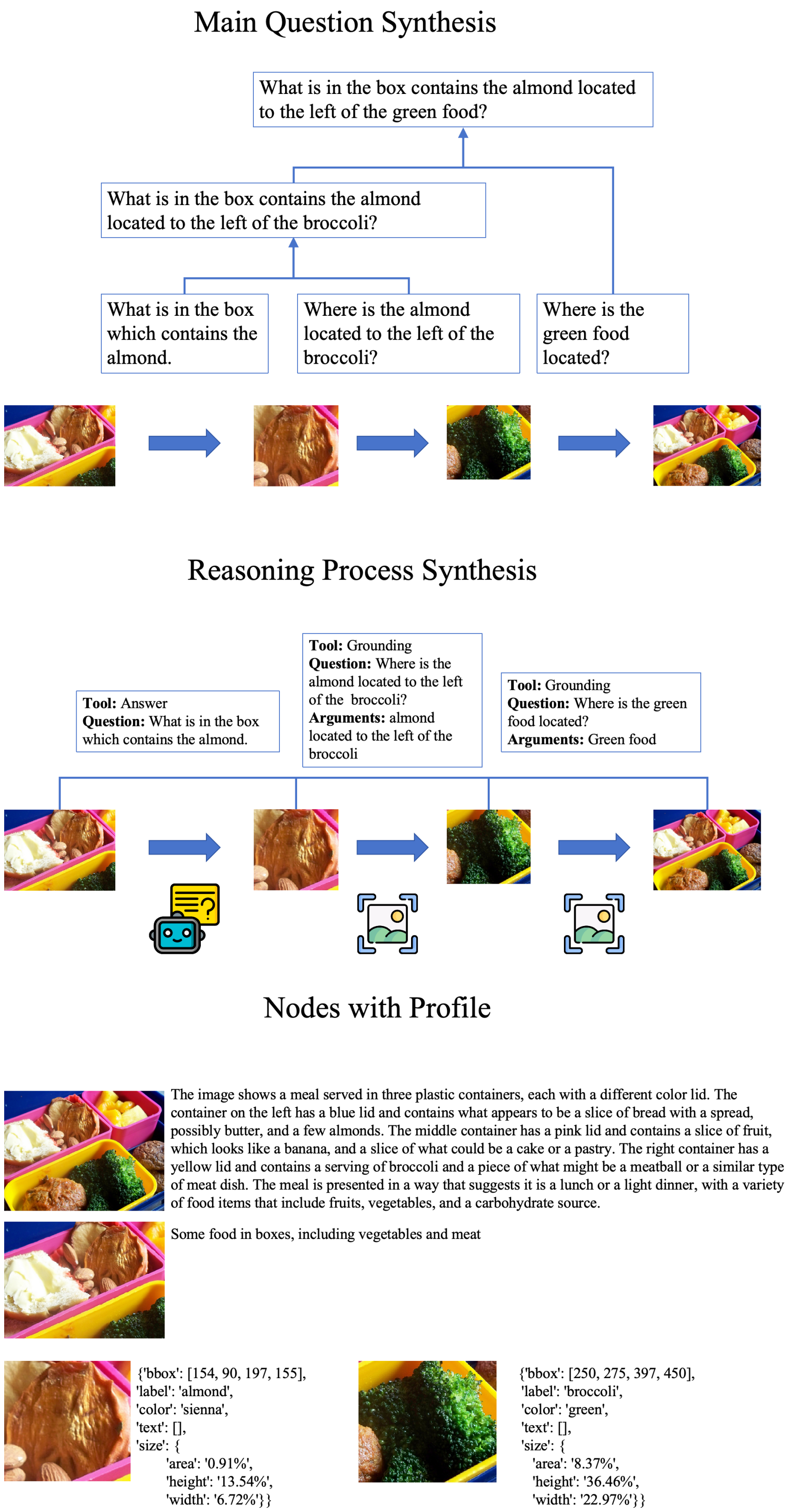


Figure 8: A simple case to demonstrate the data construction process.