VisReas: Complex Visual Reasoning with Unanswerable Questions

Syeda Nahida Akter[1], **Sangwu Lee**[2], **Yingshan Chang**[1], **Yonatan Bisk**[1], **Eric Nyberg**[1]
Language Technologies Institute, Carnegie Mellon University, Pittsburgh, PA, United States[1]
Department of Computer Science, University of Rochester, Rochester, NY, United States[2]
`{sakter,yingshac,ybisk,ehn}@cs.cmu.edu`, `slee232@u.rochester.edu`

Abstract

Verifying a question's validity before answering is crucial in real-world applications, where users may provide imperfect instructions. In this scenario, an ideal model should address the discrepancies in the query and convey them to the users rather than generating the best possible answer. Addressing this requirement, we introduce a new compositional visual question-answering dataset, **VisReas**, that consists of answerable and unanswerable visual queries formulated by traversing and perturbing commonalities and differences among objects, attributes, and relations. **VisReas** contains 2.07M semantically diverse queries generated automatically using Visual Genome scene graphs. The unique feature of this task, ***validating question answerability with respect to an image before answering***, and the poor performance of state-of-the-art models inspired the design of a new modular baseline, **Logic2Vision** that reasons by producing and executing pseudocode ***without any external modules*** to generate the answer. Logic2Vision outperforms generative models in VisReas (+4.82% over LLaVA-1.5; +12.23% over InstructBLIP) and achieves a significant gain in performance against the classification models.[1]

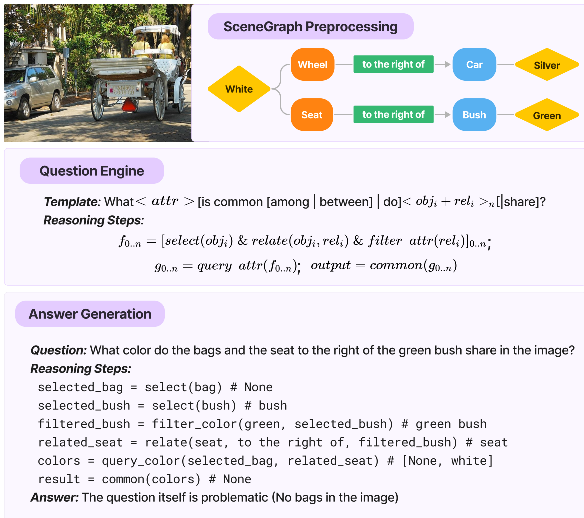


Figure 1: Overview of VisReas dataset construction process. Using scene graphs, we cluster objects (**orange**), relations, and attributes of the related objects (**blue**) based on the attribute of the corresponding objects (**orange**). Then the question engine takes each template as input and traverses all possible clusters to generate the query as well as the reasoning steps. Each function in the reasoning steps can return **NONE** if any object, attribute, or relation is absent in the image.

1 Introduction

In visual question answering (VQA), validating question authenticity with the corresponding image and then reasoning over it is an important requirement in real-world application dynamics where users may make errors in judgment, leading to invalid queries. Confirming a question's validity becomes pivotal to maintaining consistency, rectifying mistakes, and preventing misguided responses (Rajpurkar et al., 2018). Following the prior VQA datasets' (Goyal et al., 2017; Krishna et al., 2016; Hudson and Manning, 2019b) focus on answerable questions only, a system trained solely for answerable questions may exhibit unstable behaviors when faced with unanswerable queries. For instance, a delivery robot receiving an incorrect address but a valid instruction like "`place the package by the yellow door`" might overlook the error unless prompted to reevaluate its decision. In contrast, presuming the correctness of the query would likely lead to unpredictable behaviors. Therefore, a reliable and responsible system should be able to question the validity of the instruction it receives before acting upon it.

However, aligning questions with the region of interest in the image breaks down visual reasoning task into perception (object detection and scene

[1]Code and data at https://github.com/RE-N-Y/visreas.git

Findings of the Association for Computational Linguistics ACL 2024, pages 6735–6752

representation learning) and reasoning (question interpretation and inference grounded in the scene). Datasets and models proposed to date have shown significant improvement in the detection task which therefore improved the perception system (Goyal et al., 2017; Krishna et al., 2016; Tan and Bansal, 2019), but they face critical vulnerabilities due to the lack of generalities in the datasets (Zhang et al., 2016a; Agrawal et al., 2016). Recent datasets (Johnson et al., 2017; Selvaraju et al., 2020; Hudson and Manning, 2019b) encourage reasoning beyond surface-level object recognition and focus on multi-step inference. But they tend to reason about object relations *(often questions revolving around single object)* instead of reasoning over clusters of objects in the image that share common attributes or relations. Reasoning over general sets of objects requires both identifying objects and understanding their attributes and relations.Where prior scene-graph based work assumes reasoning follows from traversing a single path to generate an answer, our goal is to establish a multi-hop approach of identifying cliques with shared properties.

Bridging the gap in prior benchmarks, we introduce a new dataset, **VISREAS** (**Vis**ual **Reas**oning), for studying reasoning over commonalities and differences across objects. The unnatural assumption in the current VQA datasets - *"a correct answer for every question"* causes models to produce an answer even when the question is inapplicable and has no possible answer. To ensure that models verify the consistency of question text with the image before answering, we curate questions that have no answer given the image by altering relations and attributes among the objects. We design a question generation engine that takes the information about objects, attributes, and relations from the Visual Genome scene graphs (Krishna et al., 2016) and finds common features shared among multiple objects. Based on this retrieved information, we generate 2.07M unique questions covering vast semantic variations. Each question is paired with a scene graph and a semantic program that specifies the series of reasoning steps needed to be performed to produce the answer. Our generated questions require visual reasoning abilities such as comparing, differentiating, counting, clustering objects, and performing logical reasoning. Most importantly, unlike other VQA datasets, VISREAS enforces the VQA models to verify the information in the question with the image in each reasoning step before predicting an answer.

We find existing VQA models less robust in the reasoning and unanswerable settings presented by VISREAS. Motivated by the shortcomings of existing models, we propose a new architecture, **LOGIC2VISION** that has been trained to produce logical reasoning steps from the query at first and then predict answers based on the reasoning steps and the image. Unlike prior generative models, LOGIC2VISION is compute and cost-efficient as it does not require any external expensive APIs or modules and solely relies on the reasoning capabilities of visual language models (VLM). Experiments on VISREAS shows that LOGIC2VISION outperforms the current fine-tuned VQA models: obtaining **66.20%** (+4.82% over LLaVA-1.5 (Liu et al., 2023), +12.23% over InstructBLIP (Dai et al., 2023)) accuracy on VISREAS.

In short, our contributions are twofold:

- We introduce **VISREAS**, a dataset containing complex yet natural reasoning. Our dataset makes the first step towards developing reliable VLM adaptable to real-world scenarios where user instructions may not always be impeccable.
- We present **LOGIC2VISION**, that aims to handle spatial reasoning by executing consecutive pseudocode with verification in each step.

2 Related Works

Recent years have witnessed tremendous progress in visual understanding. Multiple attempts have been made to mitigate the systematic biases of VQA datasets (Goyal et al., 2017; Zhang et al., 2016b; Agrawal et al., 2018; Johnson et al., 2017), but they fall short in providing an adequate solution: Some approaches operate over constrained and synthetic images (Zhang et al., 2016b; Johnson et al., 2017), neglecting the realism and diversity natural photos provide. Suhr et al. (2019) introduced a dataset for reasoning about semantically-diverse natural language descriptions of images in the form of a classification task. While the dataset exhibits diverse semantic phenomena, this task rarely requires much beyond a single type of object recognition and its associated relation and attribute. Unlike these datasets, VISREAS is open-ended and consists of both unanswerable and answerable queries based on the similarity/dissimilarity of multiple objects in the image. VISREAS jointly evaluates VQA models' alignment, multihop reasoning, and verification ability which cannot be approximated by simply finding the most likely object/relation/attribute to answer the question.

Recent transformer-based models have (Tan and Bansal, 2019; Lu et al., 2020; Nguyen et al., 2022) achieved promising performance on visual reasoning tasks. Yet, these models are prone to reproducing spurious correlations without accurately learning true causal relations (Agrawal et al., 2016; Jia and Liang, 2017; Tenenbaum, 2018). Neural-symbolic methods (Andreas et al., 2016; Hu et al., 2017; Hudson and Manning, 2018, 2019a) explicitly perform symbolic reasoning on the object and language representations. These models offer modularity and interpretability in the reasoning process. However, as module parameters are usually derived solely from end-task supervision, there is a potential for the program to deviate from accurately explaining the model's behavior (Ross et al., 2017; Jain and Wallace, 2019; Subramanian et al., 2020).

Conversely, a recent approach to modularity leverages Large Language Models (LLM) to craft code or Python programs using expensive APIs (Chen et al., 2021; Surís et al., 2023; Gupta and Kembhavi, 2023; Subramanian et al., 2023). However, these approaches outsource basic aspects of the reasoning to external components rather than performing reasoning as part of the model itself. For example, prior works outsource basic cognitive abilities such as recognizing objects, counting, and even arithmetic operations. Focusing on these limitations, our proposed LOGIC2VISION aims to leverage single VLM to address complex reasoning in a modular approach that shows promising performance across models of three different categories.

3 VISREAS: Visual Reasoning

The VISREAS dataset is an attempt towards better aligning model capabilities with real application circumstances. In parallel, VISREAS aims to develop complex compositional reasoning into the machine that involves consideration of relations among multiple objects and verification of alignment between information provided in the question and the image. In the following sections, we provide details about the VISREAS data generation pipeline and a comprehensive analysis of the VISREAS dataset. In the supplementary material, we conduct a detailed comparative study between VISREAS and the well-established GQA dataset, followed by details of the human evaluation process using Mechanical Turk.

3.1 Data Generation

Our dataset is constructed in three major steps: (1) Process scene graphs, (2) Define templates and reasoning functions that the question will involve, (3) Automatically generate corresponding reasoning steps in pseudocodes along with the final answer from each query as shown in Fig. 1. Finally, to prevent models from learning statistical biases in attribute, reasoning, or answer type distributions, we meticulously balance the VISREAS dataset across three distinct paradigms (Appendix A).

3.1.1 Scene Graph Processing

To begin with the data construction process, we run two phases of processing on the scene graphs before passing them to the question engine.

First Phase. We clean up the scene graphs by removing opposite attributes and discarding object nodes with similar names that share similar attributes and relations. Our processed scene graphs contain 1703 distinct objects, 14 attributes, and 114 relationships. It is also observed that one object name in the image might correspond to multiple object IDs and bounding boxes in the scene graph. This will cause ambiguity in the later question-generation process. Thus, we merge bounding boxes corresponding to the same object name with a high IoU (> 0.7). In addition, there can be images where a bigger bounding box contains multiple small bounding boxes, which can be either parts of the object represented by the bigger bounding box (e.g., a cat (bigger bounding box) has a tail, ear, face (small bounding boxes), etc.) or they can collectively represent the object in the bigger bounding box (e.g., lime and apple can together be mentioned as fruits). These overlapping bounding boxes will be problematic while clustering objects based on similar attributes (e.g., fruits and lime are all green; for *'What has the same color as the lime?'* the answer generation module will produce: fruits and apple - which is ambiguous). To discard these cases, we measure the ratio of intersection area vs individual bounding box area and check whether the smaller objects are subclasses of the bigger one using Wordnet (Miller, 1994). If the ratio is high and the larger object is a superclass of the smaller one, we discard the larger bounding box during preprocessing to avoid ambiguity.

Second Phase. We cluster the scene graphs based on the common attributes and relations among the objects in each image and create several

Attribute	Templates	Train	Validation
Color	12	1326086	1500
Cleanliness	8	7794	900
Material	15	368337	1500
Size	4	116438	1500
Pose	18	36687	1500
Height	10	9894	1200
Weather	6	31376	1500
Length	11	45764	1500
Tone	11	37184	1500
Shape	15	30119	1500
Activity	21	15639	1500
Sport Activity	21	13215	1500
Age	12	19594	1500
Pattern	18	14313	1500
Total	**182**	**2072440**	**20100**

Table 1: Question-template distribution over attributes

	# QA	# Images	AveQLen	ListAns	Grounding	Counting	RealWorld	No Answer
VQA	614,163	204,721	6.2 ± 2.0	✓		✓	✓	
Visual7W	327,939	47,300	6.9 ± 2.4	✓	✓	✓	✓	
CLEVR	853,554	100,000	18.3 ± 3.5		✓	✓		
GQA	1,750,623	113,000	7.9 ± 3.1		✓		✓	
VisReas	2,072,437	113,000	19.4 ± 4.6	✓	✓	✓	✓	✓

Table 2: Comparisons on existing VQA datasets. VisReas covers a wide variety of reasoning along with ***No Answer*** cases. The average question length is also higher in VisReas compared to others.

sub-graphs as seeds for the question engine. Initially, we cluster objects based on a single relation or attribute, later we merge the clusters recursively if there are objects with multiple attributes or relations in common. Finally, each cluster represents a collection of objects that share a similar set of attributes and relations and the question engine exhaustively traverses all clusters to generate questions. For each object in a cluster, we also store other objects that are related to that object along with their relation name. This information is used to populate nested compositional references for multi-hop relation traversal.

3.1.2 Question Engine

For question generation from the clusters, we manually create 182 templates on different attributes (Table 1). Our templates cover five categories of reasoning (*query*, *count*, *compare*, *verify*, and *choose*) which can be further broken down into nine broad categories of reasoning mentioned in Appendix. For some categories, we have list answers and no-answer cases. All of our templates are formulated considering clusters of objects to facilitate multi-object comparison. To generate no-answer cases, we apply two approaches: (1) We either add an outlier (object not present in the image) to the cluster or include an object that exists in the image but not in the cluster and has different relations and attributes from the objects in the cluster. (2) We perturb the existing relation/attribute of an object inside a cluster (e.g., change *'apple to the left of knife'* to *'apple to the right of knife'*) which derives no-answer cases.

3.1.3 Answer Generation

The answer generation step involves two consecutive phases. *Initially,* we formulate the reasoning steps in pseudocode (Figure 1) and produce the intermediate results for each line of code using our designed parser (Figure 9). For each question template and reasoning type, we have hand-coded the basic reasoning steps necessary to answer the query. Based on the number of objects, relations, and attributes, our parser generates all intermediate reasoning steps along with the answers. *Finally,* we combine all intermediate results to come up with the answer. If any intermediate reasoning step results in **'NONE'**, the final answer becomes `'the question itself is problematic'` indicating some objects, relations, or attributes mentioned in the question text cannot be found in the image.

3.2 Dataset Analysis and Comparison

The VisReas dataset consists of 113K images from the Visual Genome where each image is annotated with dense descriptions of the scene stored in the scene graphs. We refine the existing scene graphs and generate $2,072,437$M unique questions, twice the size of current VQA datasets (Table 1), that combine features of multiple objects and their relations and require the implementation of consecutive complex reasoning skills with an in-depth understanding of object attributes and relations in the image. Our dataset covers 14 different attributes and 114 diverse relations among 1703 different objects from real-life images. We define five major types of reasoning (Figure 2) while generating the corpus based on the overall nature of the query template. Figure 5 shows details of the query structures along with examples. However, the intermediate reasoning steps that are necessary to answer the query can be diverse and can combine all five types of reasoning for a single query (as in Figure 1). We balance the dataset combinedly across 14 attributes and 5 reasoning types (Appendix A).

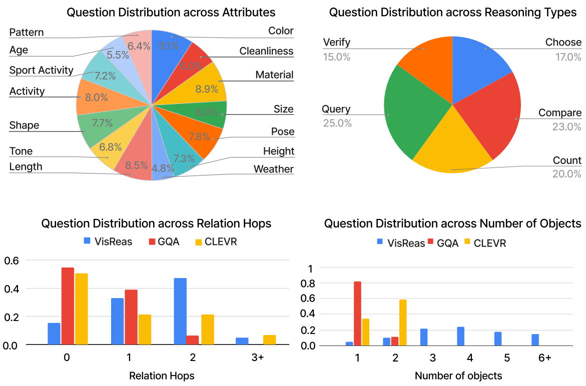


Figure 2: Overview of VISREAS statistics. **(Top left)** The dataset covers 14 attributes in a balanced ratio. **(Top right)** It consists of five reasoning types of queries in a balanced distribution. **(Bottom left)** Comparison of multi-hop relation traversal for different VQA datasets. Majority questions of VISREAS require multi-hop traversal compared to others. **(Bottom right)** Comparison of number of objects mentioned in the question for different datasets where VISREAS questions contain larger amount of objects.

Compared to existing VQA tasks, VISREAS emphasizes creating longer reasoning chains (multi-hop) with a larger number of objects (Figure 2). The average number of reasoning hops for VISREAS is 1.42 (95% CI: $[1.415, 1.417]$), significantly higher than GQA (mean: 0.52; 95% CI: $[0.517, 0.519]$) and CLEVR (mean: 0.84; 95% CI: $[0.839, 0.843]$). However, to limit the question length and increase human readability (Figure 6), the majority of the questions require at most two hops relation traversal for each object.

Reflecting on human clustering ability based on commonalities, VISREAS consists of queries that require consideration of multiple objects based on their attribute or relation similarities. Therefore, unlike existing datasets, the majority of VISREAS queries are composed of more than three objects from the image. The average objects per question for VISREAS is 3.91, which is higher than both GQA (1.12) and CLEVR (1.63). Hence, VISREAS requires multiple object detection and consecutive reasoning to answer a single query (Figure 2). In addition, each query can have multiple attributes associated with it (Figure 7a). For example, in question, `'What is the common material among the silver and blue utensils?'`, both `<material>` and `<color>` attributes are needed to be considered for answer generation that involves multiple attribute filtering along with the associated objects.

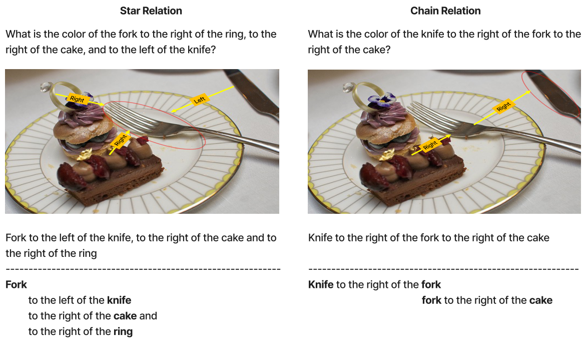


Figure 3: VISREAS contains two types of relation traversals. **Star relation** states a single object that shares multiple relations with other objects (Left). **Chain relation** states multiple objects that share a single relation with each other (Right).

In contrast to other spatial reasoning datasets that focus primarily on one-hop relation traversals (Bottom left of Figure 2), we explore two ways of novel traversals: (1) ***Star Relation:*** The target object shares multiple relations with other objects (e.g. is the center of the star and other objects are connected to it with a relation – Figure 3 left), and (2) ***Chain Relation***: The target object is related to an object that is related to another object and the relation traversal is linear (Figure 3 right). The inclusion of these traversals adds multi-hop complexity to the corpus and makes the *each-step verification process* harder for unanswerable questions (as Figure 10).

4 LOGIC2VISION

In recent years, LLMs combined with code generation and chain-of-thought prompting have shown impressive performance in complex reasoning by generating intermediate reasoning steps before inferring the answer (Zhang et al., 2023a; Surís et al., 2023). However, these frameworks are often prone to hallucinations of LLMs and are too restricted in terms of reasoning they can perform and dependent on expensive external modules to execute the reasoning (Zhang et al., 2023b; Surís et al., 2023). To address these limitations and elicit the reasoning capability of VLMs, we propose LOGIC2VISION, a two-stage VQA framework that (1) plans the necessary reasoning steps using the question and (2) executes the plan with the help of an image leveraging the SOTA VLM (Figure 4).

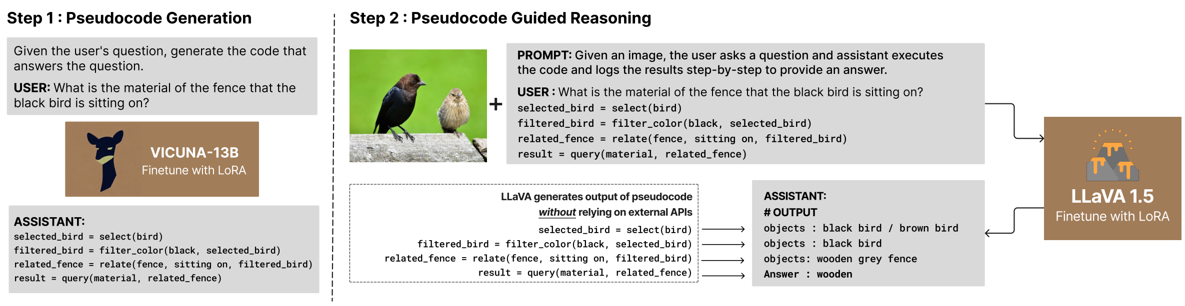


Figure 4: Overview of LOGIC2VISION. In **Pseudocode Generation** phase, we generate pseudocode which outlines the reasoning steps. During **Pseudocode-Guided Reasoning**, the pseudocodes along with the question and image are provided to the model. The model executes all intermediate pseudocodes to arrive at the final answer.

4.1 Stage 1: Pseudocode Generation

Given a natural language question, this module generates a consecutive set of reasoning steps as pseudocodes. For training our pseudocode generation model, we take advantage of the existing VQA dataset: GQA as it provides a semantic string that decomposes the question into a sequence of reasoning steps. For instance, the semantic string for the question `'Is there a red apple on the table?'` would be `'select: table → relate: on, subject, apple → exist: ?'`. We build a custom parser (Figure 9) that converts each line of GQA semantic string to pseudocode and extracts all the intermediate expected outputs along with the final answer from the scene graph. The parsed `(pseudocode, output)` pairs serve as a rationale to solve the question (Figure 8). For the pseudocode generation, we use an instruction finetuned VICUNA-13B (Chiang et al., 2023) model which has shown good performance across various language tasks including code generation. We finetune VICUNA using LoRA on `(question, pseudocode)` pairs (Hu et al., 2022) to generate the pseudocode for a given question. The fine-tuned model achieves 98.6% METEOR (Banerjee and Lavie, 2005) score and 96.3% ROUGE-L (Lin, 2004) score against ground-truth code parsed from GQA semantic strings.

4.2 Stage 2: Pseudocode-Guided Reasoning

Since the Pseudocode Generation module outlines the necessary steps to answer the question, the remaining task is to perform pseudocode-guided sequential reasoning on the image. For this stage, we choose state-of-the-art VLM, LLaVA-1.5 (Liu et al., 2023), due to its impressive performance in diverse reasoning tasks. As LLaVA-1.5 was not trained to reason with pseudocode and image, we fine-tuned it to generate an answer by executing sequential reasoning with the pseudocode and the image. To adapt this framework in our case, we rearrange the instruction as below:

> USER:<Image> *Executes the code and logs the results step-by-step to provide an answer to the question.*
> **Question**: {Question}
> **Code**: {Code}
> ASSISTANT:
> **Logs**: {Logs}
> **Answer**: {Answer}

Here LOGIC2VISION takes the image, question, and the corresponding sequential pseudocodes as input and produces all intermediate outputs of the codes as logs along with the final answer. Therefore, during fine-tuning, LOGIC2VISION not only learns to generate the final answer but also must predict all intermediate responses correctly. This includes predicting `NONE` when there is no answer possible in any intermediate step. The ability to produce intermediate outputs as logs makes LOGIC2VISION more explainable compared to others. As each line of the pseudocode requires a different reasoning ability (e.g., `select`, `compare` or `relate`), we can detect which reasoning task the model is failing by simply tracking the logs. The essential training details of this stage can be found in subsection C.3.

5 Experiments and Analysis

In the subsequent sections, we conduct a comprehensive analysis of the VISREAS dataset and assess the performance of various benchmarks including LOGIC2VISION, GPT-4V (OpenAI, 2023), and human participants, revealing a notable disparity from

		Model	Accuracy (%) GQA	VisReas
ZS	GEN	BLIP-2 (2023)	44.70	35.16
		InstructBLIP (2023)	49.50	36.84
		LLaVA-1.5 (2023)	63.3*	38.98
	Code-GEN	ViperGPT (2023)	48.10	10.31
		VisProg (2023)	50.50	20.82
FT	CLS	LXMERT (2019)	60.05	50.15
		ViLBERT (2020)	60.65	53.05
		CRF (2022)	72.10	53.56
	Logic-GEN	Logic2Vision	**60.32**	**66.20**

Table 3: Performance comparison among baseline models on GQA and VisReas. (*) GQA trainset images were used during training.

human performance.

5.1 Baseline Experiments

To analyze the complexity and generalizability of our dataset and model, we run experiments with models trained on both classification and generative tasks. We cover two types of generative models: **GEN** (relies on pretrained visual-language alingment module) and **Code-GEN** (generates a program and utilizes external APIs to solve VQA tasks). We categorize Logic2Vision as **Logic-GEN** as it produces intermediate logical reasoning steps before answering. All model configurations can be found in Appendix C. To make the training and inference consistent, we define our own prompt for all generative models (as subsection C.4). Table 3 shows the results of different baselines on both GQA and VisReas. All baseline models perform worse on VisReas than on GQA, highlighting the unique challenge provided by VisReas. Table 4 presents the performance on VisReas across diverse baselines along with GPT-4V and human accuracy. We break down the performance along two axes: the reasoning type and answerability. We finetune models in the CLS and the GEN groups to obtain stronger baseline results. We could not finetune models in the Code-GEN group due to their close-sourced weights. Logic-GEN outperforms all others baselines at a significant margin.

[CLS] For models trained with classification task, we finetune and evaluate on both GQA and VisReas. From the fine-tuning results of the CLS models, it is obvious that VisReas proposes a different task than GQA that can not be easily solved by scaling the model size or changing the pretraining corpus. Furthermore, the higher performance gap of the models between GQA and VisReas tasks suggests the inefficacy of the existing CLS models on our proposed spatial reasoning task.

[GEN] From generative domain, we select three SOTA models, BLIP-2, InstructBLIP, and LLaVA-1.5, that try to leverage the LLMs using two types of vision-language alignment modules: Q-Former and MLP cross-modal connector. We evaluate the models on zero-shot GQA and VisReas to probe the relevance of our proposed task to their training domain. We notice that BLIP-2 performs poorly on our task compared to GQA where InstructBLIP and LLaVA-1.5 shows higher accuracy. Both LLaVA-1.5 and InstructBLIP are instruction tuned on diverse downstream tasks which allows them to excel in VQA tasks compared to BLIP-2. However, LLaVA-1.5 gains the highest zero-shot accuracy in this category due to its training set images being overlapped with VisReas. Yet, it shows a significant drop (-24.32%) in ZS accuracy compared to GQA, which proves that VisReas highlights a novel reasoning task that can not be generalized using GQA. Furthermore, the smaller performance gap among these models on VisReas suggests the inefficacy of the current VLMs on our proposed spatial reasoning task.

[Code-GEN] From modular Code Generation models, we analyze recent works - ViperGPT and VisProg. These models employ an LLM to generate an executable program that utilizes a pre-defined API, including functions such as `detect(image, obj_category)` or `segment(image, obj_category)`. VisProg also utilizes the "in-context learning" abilities of LLMs, enabling the model to respond to new queries with just a few examples of input and output behavior. Zero-shot evaluations of Code-GEN models on GQA and VisReas reveal that current models are struggling with our task more than GQA, where both corpora use similar images. We find these models heavily biased to answerable setting that they tend to ignore the discrepancies between the question and the image. Furthermore, the codes generated by these models are often incomplete or runs into error when passed to the compiler. We term these cases as incorrect responses for consistent evaluation. We believe that problematic questions can be handled better with modified prompts which would require additional expensive few-shot prompting. However, their poor performance in *Non-Problematic* questions denotes the inability of these models to reason with longer relational hops and cluster multiple objects based on commonali-

Metric	CLS LXMERT	CLS ViLBERT	CLS CRF	GEN BLIP-2	GEN InstructBLIP	GEN LLaVA-1.5	Code-GEN ViperGPT	Code-GEN VisProg	Logic-GEN LOGIC2VISION	GPT-4V	Humans
Choose	74.23	82.91	83.30	71.21	78.50	84.11	10.37	15.86	82.54	82.61	91.30
Compare	65.62	69.86	71.87	28.72	53.29	67.75	5.97	26.09	59.25	68.33	86.12
Count	45.32	47.80	49.59	25.88	49.86	43.08	7.85	6.02	39.47	39.52	85.78
Query	44.05	47.65	48.11	41.55	47.77	50.31	4.35	19.30	63.79	58.78	81.78
Verify	76.10	82.18	83.03	70.77	49.48	81.27	3.10	44.18	84.54	82.16	93.94
Problematic	67.54	77.08	78.41	25.39	64.68	68.04	0.25	0.16	55.34	70.18	90.29
Non-Problematic	56.11	59.16	61.60	51.41	52.25	60.31	11.97	24.17	67.94	55.47	84.89
Accuracy (%)	**50.15**	**53.05**	**53.56**	**47.81**	**53.97**	**61.38**	**10.38**	**20.82**	**66.20**	**62.83**	**87.21**

Table 4: Accuracy breakdown of baseline models and humans on VISREAS across different reasoning types. **Problematic** type consists of questions that contain certain relation, attribute, or object that is missing/ not consistent with the image. In contrast, **Non-Problematic** questions have correct answers as the question is consistent with the image. Except for the Code-GEN models, we provide **fine-tuned results** on VISREAS for all other models.

ties.

5.2 Analysis

According to Table 4, all the models including GPT-4V struggle in `Compare`, `Count`, and `Query` question-types which require grounding, clustering, and verifying the existence of multiple objects, relations, and attributes. Specifically in `Query`, the performance gap between humans and the models is significantly higher which demonstrates the limitation of current models to perform complex multi-hop reasoning. LOGIC2VISION, on the other hand, shows a promising result in `Query` questions. We hypothesize that structured pseudocode helps the model consider each object and its corresponding attributes and relations before answering while the other models try to learn from the surface-level word distribution. In addition, `Query` questions are in general lengthier than other types of questions which makes it easier for the models to lose attention to the details (Figure 7b).

In contrast, GPT-4V outperforms all generative models in `Problematic` questions. After analyzing the predictions, we find that GPT-4V excels at identifying problematic questions that involve an object not present in the image or an object with a false attribute. However, when the question becomes problematic due to an incorrect relation, GPT-4V consistently struggles to recognize it which also holds for other models. This signifies the uniqueness of our corpus that emphasizes understanding relations beyond simple object detection. It is also notable that GPT-4V often denies to answer questions related to a person and sometimes just ignores questions by saying *'I'm sorry, but I can't assist with identifying or making assumptions about people in images.'* For fair comparison with other models, we report all these occurrences as incorrect answers.

Model	Choose	Compare	Count	Query	Verify	Prob.	Non-Prob.	All
7B	81.20	54.90	35.13	59.24	82.75	55.38	63.92	**62.74**
13B	82.54	59.25	39.47	63.79	84.45	55.34	67.94	**66.20**

Table 5: Breakdown of accuracies on VISREAS for LOGIC2VISION's VICUNA model size. We observe that VICUNA's model size improves performance in most question-types except the problematic ones.

To investigate the effect of LLM's scale on the VQA task, we test two versions of LLMs (VICUNA 7B and 13B) within VISREAS architecture. Table 5 breaks down the performance of LOGIC2VISION in the presence of different LLMs. We observe that increasing LLM's size dramatically increases the accuracy of longer questions (Figure 7b) such as `Non-Problematic`, `Count`, `Query`, and `Compare` instances and marginally improves performance on question categories such as `Choose` and `Verify`. This finding reassures the ability of larger LLM to reason with longer context. However, for problematic questions, increasing LLM size has no impact. As this category requires verification and grounding of information with image, both LLM and vision-language alignment need to be strong to excel in this domain.

6 Conclusion

We introduce the VISREAS dataset, for real-world complex and multihop visual reasoning and compositional question answering. The dataset emphasizes object commonalities, differences, and relational aspects, necessitating validation of question-text relevance with the image before answering. We describe the dataset curation process along with the performance of SOTA models from three different domains in our task. Addressing the shortcomings in grounding and clustering in recent models, we propose a novel LOGIC2VISION baseline that deconstructs questions into pseudocodes and sequentially executes them using images to generate answers. We anticipate that this dataset and model will catalyze advancements in VQA research, pushing it toward complex semantic comprehension, robust reasoning, and addressing unanswerability when the provided context is not sufficient.

7 Discussion and Future Work

Solving VQA tasks via code generation and external APIs has gained attention due to its capability to perform complex reasoning and planning in a modular manner. However, code generation has limitations: a fixed set of operations limits models to specific types of questions and heavy use of external modules prevents end-to-end training. While modularity encourages specialization, in practice it requires managing multiple environments and heavy GPU memory usage as multiple large models are used to carry out visual and cognitive tasks like detection and captioning. In addition, current code generation methods (Surís et al., 2023; Gupta and Kembhavi, 2023) rely on OpenAI's API to generate executable code which hinders the accessibility of benchmarking due to its high costs[2] and fluctuations of OpenAI models over time (Chen et al., 2023) which makes it hard to diagnose whether certain performance gains come from OpenAI model or improvements in other components. In contrast, our model and dataset suggest that one can use a single VLM model that combines both the strength of structured reasoning and train it in a simple end-to-end manner. VISREAS requires many operations such as `select`, `filter`, `relate`, and `query` which are limited to cognitive skills to standard VQA tasks and spatial reasoning. Therefore, models trained on VISREAS may not generalize well for visual-language tasks such as visual storytelling and image captioning which goes beyond the scope of our dataset. A natural future direction would be to incorporate other visual-language tasks into the dataset as well.

[2]Evaluation with VisProg requires approximately 2,500 tokens per question including in-context examples, prompts, and outputs. Using original `text-davinci-003` model used in original code would cost $(0.0200/1000 \text{ tokens}) \cdot 2500 \text{ tokens} \cdot 17171 \text{ instances} \approx 858$ USD.

References

Aishwarya Agrawal, Dhruv Batra, and Devi Parikh. 2016. Analyzing the behavior of visual question answering models. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1955–1960, Austin, Texas. Association for Computational Linguistics.

Aishwarya Agrawal, Dhruv Batra, Devi Parikh, and Aniruddha Kembhavi. 2018. Don't just assume; look and answer: Overcoming priors for visual question answering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4971–4980.

Jacob Andreas, Marcus Rohrbach, Trevor Darrell, and Dan Klein. 2016. Neural module networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 39–48.

Satanjeev Banerjee and Alon Lavie. 2005. METEOR: An automatic metric for MT evaluation with improved correlation with human judgments. In *Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization*, pages 65–72, Ann Arbor, Michigan. Association for Computational Linguistics.

Lingjiao Chen, Matei Zaharia, and James Zou. 2023. How is chatgpt's behavior changing over time?

Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde de Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, et al. 2021. Evaluating large language models trained on code. *arXiv preprint arXiv:2107.03374*.

Wei-Lin Chiang, Zhuohan Li, Zi Lin, Ying Sheng, Zhanghao Wu, Hao Zhang, Lianmin Zheng, Siyuan Zhuang, Yonghao Zhuang, Joseph E. Gonzalez, Ion Stoica, and Eric P. Xing. 2023. Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality.

Wenliang Dai, Junnan Li, Dongxu Li, Anthony Meng Huat Tiong, Junqi Zhao, Weisheng Wang, Boyang Li, Pascale Fung, and Steven Hoi. 2023. Instructblip: Towards general-purpose vision-language models with instruction tuning.

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 *Conference on Computer Vision and Pattern Recognition (CVPR)*.

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 (CVPR)*, pages 14953–14962.

Edward J Hu, yelong shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*.

Ronghang Hu, Jacob Andreas, Marcus Rohrbach, Trevor Darrell, and Kate Saenko. 2017. Learning to reason: End-to-end module networks for visual question answering. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*.

Drew Hudson and Christopher D Manning. 2019a. Learning by abstraction: The neural state machine. *Advances in Neural Information Processing Systems*, 32.

Drew A Hudson and Christopher D Manning. 2018. Compositional attention networks for machine reasoning. In *International Conference on Learning Representations (ICLR)*.

Drew A Hudson and Christopher D Manning. 2019b. Gqa: A new dataset for real-world visual reasoning and compositional question answering. *Conference on Computer Vision and Pattern Recognition (CVPR)*.

Sarthak Jain and Byron C. Wallace. 2019. Attention is not Explanation. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 3543–3556, Minneapolis, Minnesota. Association for Computational Linguistics.

Robin Jia and Percy Liang. 2017. Adversarial examples for evaluating reading comprehension systems. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2021–2031, Copenhagen, Denmark. Association for Computational Linguistics.

Justin Johnson, Bharath Hariharan, Laurens Van Der Maaten, Li Fei-Fei, C Lawrence Zitnick, and Ross Girshick. 2017. Clevr: A diagnostic dataset for compositional language and elementary visual reasoning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2901–2910.

Kushal Kafle and Christopher Kanan. 2017. Visual question answering: Datasets, algorithms, and future challenges. *Computer Vision and Image Understanding*, 163:3–20.

Ranjay Krishna, Yuke Zhu, Oliver Groth, Justin Johnson, Kenji Hata, Joshua Kravitz, Stephanie Chen, Yannis Kalantidis, Li-Jia Li, David A Shamma, Michael Bernstein, and Li Fei-Fei. 2016. Visual genome: Connecting language and vision using crowdsourced dense image annotations.

Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. 2023. BLIP-2: bootstrapping language-image pre-training with frozen image encoders and large language models. In *ICML*.

Chin-Yew Lin. 2004. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain. Association for Computational Linguistics.

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

Jiasen Lu, Vedanuj Goswami, Marcus Rohrbach, Devi Parikh, and Stefan Lee. 2020. 12-in-1: Multi-task vision and language representation learning. In *The IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

George A. Miller. 1994. WordNet: A lexical database for English. In *Human Language Technology: Proceedings of a Workshop held at Plainsboro, New Jersey, March 8-11, 1994*.

Binh X Nguyen, Tuong Do, Huy Tran, Erman Tjiputra, Quang D Tran, and Anh Nguyen. 2022. Coarse-to-fine reasoning for visual question answering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4558–4566.

OpenAI. 2023. Gpt-4 technical report.

Pranav Rajpurkar, Robin Jia, and Percy Liang. 2018. Know what you don't know: Unanswerable questions for SQuAD. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 784–789, Melbourne, Australia. Association for Computational Linguistics.

Andrew Slavin Ross, Michael C. Hughes, and Finale Doshi-Velez. 2017. Right for the right reasons: Training differentiable models by constraining their explanations. In *Proceedings of the 26th International Joint Conference on Artificial Intelligence*, IJCAI'17, page 2662–2670. AAAI Press.

Ramprasaath R Selvaraju, Purva Tendulkar, Devi Parikh, Eric Horvitz, Marco Tulio Ribeiro, Besmira Nushi, and Ece Kamar. 2020. Squinting at vqa models: Introspecting vqa models with sub-questions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10003–10011.

Sanjay Subramanian, Ben Bogin, Nitish Gupta, Tomer Wolfson, Sameer Singh, Jonathan Berant, and Matt Gardner. 2020. Obtaining faithful interpretations from compositional neural networks. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5594–5608, Online. Association for Computational Linguistics.

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. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics*, Toronto, Canada. Association for Computational Linguistics.

Alane Suhr, Stephanie Zhou, Ally Zhang, Iris Zhang, Huajun Bai, and Yoav Artzi. 2019. A corpus for reasoning about natural language grounded in photographs. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 6418–6428, Florence, Italy. Association for Computational Linguistics.

Dídac Surís, Sachit Menon, and Carl Vondrick. 2023. Vipergpt: Visual inference via python execution for reasoning. *arXiv preprint arXiv:2303.08128*.

Hao Tan and Mohit Bansal. 2019. Lxmert: Learning cross-modality encoder representations from transformers. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*.

Josh Tenenbaum. 2018. Building machines that learn and think like people. In *Proceedings of the 17th International Conference on Autonomous Agents and MultiAgent Systems*, AAMAS '18, page 5, Richland, SC. International Foundation for Autonomous Agents and Multiagent Systems.

Peng Zhang, Yash Goyal, Douglas Summers-Stay, Dhruv Batra, and Devi Parikh. 2016a. Yin and Yang: Balancing and answering binary visual questions. In *Conference on Computer Vision and Pattern Recognition (CVPR)*.

Peng Zhang, Yash Goyal, Douglas Summers-Stay, Dhruv Batra, and Devi Parikh. 2016b. Yin and yang: Balancing and answering binary visual questions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5014–5022.

Zhuosheng Zhang, Aston Zhang, Mu Li, Hai Zhao, George Karypis, and Alex Smola. 2023a. Multimodal chain-of-thought reasoning in language models. *arXiv preprint arXiv:2302.00923*.

Zhuosheng Zhang, Aston Zhang, Mu Li, Hai Zhao, George Karypis, and Alex Smola. 2023b. Multimodal chain-of-thought reasoning in language models. *arXiv preprint arXiv:2302.00923*.

A Data Balancing

A primary concern with current VQA datasets is the prevalence of question-conditional biases, enabling models to make informed guesses without a genuine grasp of the underlying images. Nevertheless, precise rendering of question semantics could offer enhanced control over these biases, holding the potential to significantly mitigate the issue (Zhang et al., 2016b; Kafle and Kanan, 2017). Motivated by this observation, we perform a rigorous balancing based on question categories, attribute/relation types, and answer distribution.

Adopting the balancing approach outlined in previous research (Hudson and Manning, 2019b), we employ a clustering strategy based on a fusion of two labels: `<attr/rel_type>` and `<res_type>`. The former denotes attributes or relation names (e.g., *red* or *right*), while the latter signifies reasoning types (e.g., *verify.rel*). We refine the question set within each cluster, filtering out questions that encompass overlapping sets of objects in their texts or that contain subsets of objects already covered by other questions with complete sets. We prioritize questions featuring larger sets of objects and multihop relations, provided their length stays below 25. Finally, we introduce an additional label `<answer>` and equilibrate the question sampling through the answer distribution. After executing this balancing in an iterative manner on 2.07M questions, we generate a balanced corpus of 72,244 questions with images.

B Overview and Analysis of the VisReas

This section provides an in-depth examination of the VisReas dataset, focusing on various aspects of question types and their characteristics. It encompasses an overview of question types, the distribution of semantic lengths, question readability scores, average question lengths per reasoning type, the relationship between question frequency and the number of attributes, and human accuracy on attributed questions.

B.1 Questions Types and Templates

The VisReas dataset features a diverse array of question types that challenge multimodal reasoning and compositional understanding. These question types include `query`, `count`, `compare`, `verify`, and `choose`, each requiring a unique approach to answer. Depending on how the clusters are made, each question type can further be broken down into `attr` and `rel` subtypes. Therefore, in total, there can be nine categories of questions. Figure 5 gathers all templates and examples from the dataset to offer insights into the intricacies of these question categories.

B.2 Distribution of Relation Hops and Readability

A comprehensive analysis of the distribution of relation hops in VisReas questions reveals a predominant trend toward questions that involve about two reasoning hops. These hops can entail tracking object relations, identifying attributes, or executing logical operations. We conduct a readability test using the workers from Amazon Mechanical Turk.

Type	Question	Answer	List	No Answer	Example
<query_attr>	a. What <attr> do the <objs> have common? b. What [is|are] the <attr> of the <objs>? (can be more specific depending on the attribute type)	a. <attr | attrs> b. <attr | attrs>	a. yes b. yes	a. yes b. yes	What material do the pole and the bike next to the road have common?
<count_attr>	a. Among <objs>, how many of them have [a particular <attr> | multiple <attrs> | a particular <attr> common and one of the object has another <attr>]? b. [Is | Are] there [less than | greater than] <num> objects that share [a particular <attr> | multiple <attrs>]? c. [Is | Are] any <num> of the following things, <objs> <attr | opp_attr>?	a. <num> b. <yes | no> c. <yes | no>	a. no b. no c. no	a. no b. no c. no	Among the shirt, the pant and the hat, how many are red? Are there less than three objects that share red color?
<compare_attr>	a. [Does a particular attribute | Do multiple attributes] of <obj | objs> match with the <attr | attrs> of <obj | objs>? b. Do <objs> share the same <attr> in the image? c. Are <objs> similar in <attr>?	a. <yes | no> b. <yes | no> c. <yes | no>	a. no b. no c. no	a. yes b. yes c. yes	Does the shape of the plate on the table match with the shape of the steel pan and the orange box?
<verify_attr>	Do you see any [<obj | objs> of <attr>| <attr> <obj | objs>] in the image?	<yes | no>	no	no	Do you see any red bike and red helmet?
<query_obj>	[What [object | objects]| Who] in the image [has the same | is doing the same] <attr> as <obj | objs>?	<obj | objs>	yes	yes	Who in the image is the same activity as the boy wearing blue jeans?
<choose_attr>	[Is | Are | Do | Does] <obj | objs> [| look | apprear] <attr> or <opp_attr>?	<attr>	no	yes	Do the driver and the passenger look younger or older?
<query_rel>	On which side of the <attr> <obj | objs> in the image the <attr> <obj | objs> are located?	<rel>	no	yes	On which side of the red car in the image the trees, the metal pole and the silver wire located?
<verify_rel>	[Is | Are] the <attr> <obj | objs> <rel> <attr | attrs> <obj | objs>?	<yes | no>	no	yes	Are the man on the road and the boy with red hair wearing a blue jacket?
<choose_rel>	[Is | Are] the <attr> <obj> [located | positioned] <rel> <attr | attrs> <obj | objs> or <opp_rel>?	<rel>	no	yes	Are the buses located to the left of the road or to the right of the road?

Figure 5: Overview of types of questions along with some templates and examples from the VISREAS corpus.

Our analysis reveals that questions with larger relation hops demonstrate a noticeable decline in readability, emphasizing the complexity associated with extended reasoning (Figure 6). To enhance the quality of the dataset so that it can reflect the real-world day-to-day life questions, we choose to keep the relation hop within two.

B.3 Average Question Length per Reasoning Type

By dissecting question lengths across different reasoning categories in Figure 7b, we observe a consistent trend: query questions tend to be longer than other reasoning types. This phenomenon is particularly apparent due to the inclusion of multiple objects sharing similar attributes and their corresponding relations.

B.4 Question Frequency and Attribute Usage

The VISREAS corpus has been generated using the clusters of objects that share similar relation or attribute. However, clusters based on shared attributes/relations can share objects that possess all of those attributes/relations. For example, a table and a chair have the `color` *brown* and `material` *wood* in an image. Initially, we have two clusters with *brown* and *wood*. Now, if both clusters share some objects, we again create a new cluster based on *brown+wood* adding the shared objects (i.e., table and chair). Using this approach, we create clusters that share multiple attributes and relations and generate questions that involve filtering multiple attributes/relations along with the identification of objects of interest. Figure 7a shows the distribution of questions in VISREAS with respect to the number of attributes/relations. As the number of attributes/relations goes higher, the number of clusters also decreases resulting in decreasing number of questions.

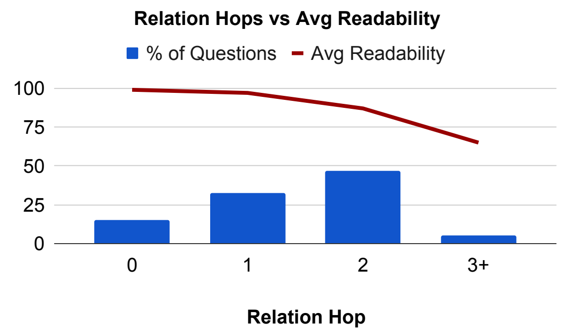


Figure 6: Distribution of VISREAS questions semantic length (number of computation steps to arrive at the answer) as well as the readability scores for each semantic step type. We can see that most questions require at most two reasoning steps, where each step may involve tracking a relation between objects, an attribute identification, or a logical operation. At the same time, questions with larger semantic steps are difficult to read.

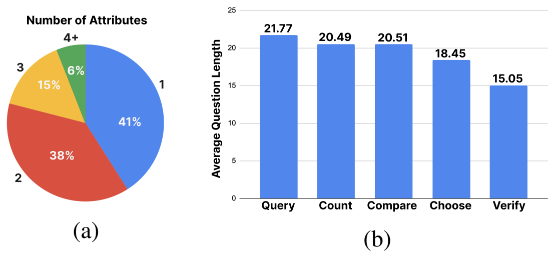


Figure 7: (a) Question distribution across the number of attributes in a query. The question complexity increases with the number of attributes or relations. (b) Average question length per reasoning type in VISREAS corpus. Query questions are lengthier than other reasoning categories as these questions contain multiple objects of similar attributes with their relations.

B.5 Human Accuracy on Attributed Questions

The final facet of our exploration delves into human accuracy when answering attributed questions from the VISREAS dataset. By assessing the performance of human subjects across different question types and attributes, we gain a deeper understanding of the challenges inherent to this multimodal reasoning task. Figure 11 breaks down the human accuracy across different attribute types. It is noticeable that `color` and `material` questions have the lowest accuracy, as they contain a higher amount of questions compared to other attributes.

In summary, this section offers a comprehensive overview and analysis of the VISREAS dataset, encompassing question types, semantic lengths, question readability, average lengths per reasoning type, attribute-based question distribution, and human

GQA semantic string format	VisReas Psuedocode format
What are the children on?	**What are the children on?**
select : children →	selected_children = select(children)
relate : on, object, children →	related_object = relate(_, on, o, selected_children)
query : name	result = query(name, related_object)
Where in this photo are the green chairs, top or bottom?	**Where in this photo are the green chairs, top or bottom?**
select : chairs →	selected_chairs = select(chairs)
filter color : green →	filtered_green = filter_color(green, selected_chairs)
choose vertical position : top | bottom	result = choose_vposition(top|bottom, filtered_green)

Figure 8: **Pseudocode format.** Our method restructures the format of GQA semantic string to pseudocode to better leverage Code-LLMs without adding any auxiliary information.

```
# question: contains question, answer, and semantic string
# scene: collection of objects in the scene
def parse(question, scene):
    codes, outputs = [], []
    for line in question["semantic"]:
        code, output = run(line, scene, history)
        codes.append(code), outputs.append(output)
    return codes, outputs
# line: single line of semantic string
# scene: collection of objects in the scene
# history: list of previous outputs
def run(line, scene, history):
    # formats semantic string to pseudocode, operator, and its arguments using regex
    # IN "select:box(43)" OUT "selected_box = select("box")", "select", { id:"43" }
    pseudocode, operator, args = parse_semantic_string(line)
    match operator:
        case "select":
            return scene.objects[arg.id]
        case "relate":
            { relation, subject, object } = args
            objects = scene.objects.find(o => o.relations[subject.id] == relation)
            return objects
        # implement other operators
        case "...":
            ...
```

Figure 9: **Semantic string parser.** For every line of semantic string, we use regex and string manipulation to extract operator and its arguments. We represent scene-graph in adjacency list format and run the parsed operator to get formatted pseudocode and its expected output.

accuracy. These insights contribute to a holistic understanding of the dataset's intricacies and its potential to advance the field of visual reasoning and question answering.

C Baseline Configuration

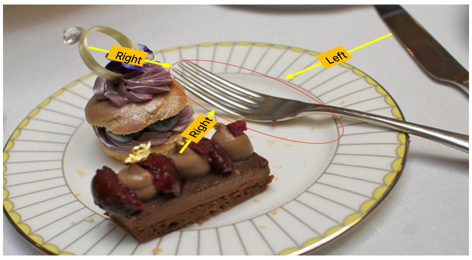
All baselines follow default settings provided by the original author evaluation script. All configurations for model, optimizer, scheduler, and training follow default parameters from Pytorch and Huggingface library. For generative models, all inference is done using default settings without temperature tuning, nucleus sampling, repetition penalty, etc. Specific settings used for zeroshot and finetuning are presented below:

C.1 VisProg

The original VisProg script uses `text-davinci-003` model which is around 10 times more expensive than `gpt-3.5-turbo` model. To cut evaluation costs, we use the

Star Relation

What is the color of the fork to the right of the ring, to the right of the cake, and to the left of the knife?



Fork to the left of the knife, to the right of the cake and to the right of the ring

----

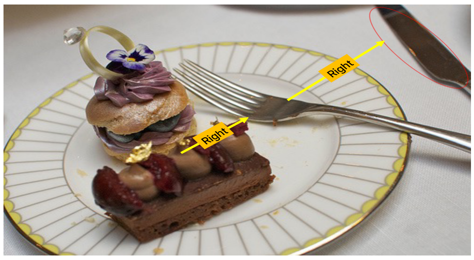
Fork
to the left of the **knife**
to the right of the **cake** and
to the right of the **ring**

Star relation in Pseudocode

```
selected_fork = select(fork)
related_knife = relate(knife, to the left of, o, selected_fork)
related_cake = relate(cake, to the right of, o, selected_fork)
related_ring = relate(ring, to the right of, o, selected_fork)
related_fork = selected_fork && exists(related_knife, related_cake, related_ring)
```

Chain Relation

What is the color of the knife to the right of the fork to the right of the cake?



Knife to the right of the fork to the right of the cake

Knife to the right of the **fork**
fork to the right of the **cake**

Chain Relation in Pseudocode

```
selected_cake = select(cake)
related_fork = relate(fork, to the right of, s, selected_cake)
related_knife = relate(knife, to the right of, related_fork)
```

Figure 10: Overview of pseudocodes for two different traversal types in the VisReas corpus.

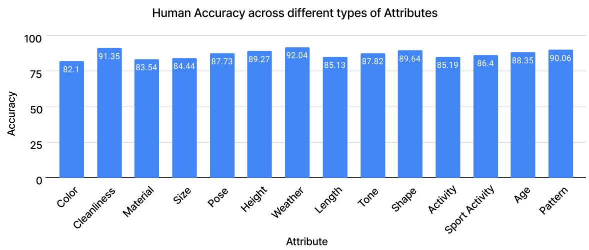


Figure 11: Human accuracy on different attributed questions

`gpt-3.5-turbo` model instead. All 20 examples are found in GQA evaluation script for code generation.

C.2 ViperGPT

For similar reason as VisProg, we use `gpt-3.5-turbo` model for code generation to reduce costs. Since generated code doesn't always return functional Python code, we return either "`None`" or "`ERROR`" in these cases. In cases where the code throws an error, the answer defaults to "`ERROR`". In cases where the code didn't have a return statement, the answer defaults to "`None`".

C.3 Logic2Vision

The effective batch size is kept at 4 across experiments. LoRA modules are only attached to query and value linear layers in attention layers. The batch size and gradient accumulation steps are adjusted accordingly. Due to memory requirements, we set batch size to 1 on each GPU and set gradient accumulation steps to 4. We've have used 2-4 A6000 GPUs with distributed data parallel (DDP) strategy for multi-GPU training. Training Logic2Vision on VisReas takes around 13 hours using 2 A6000 with LLaVA-1.5 backbone.

Hyperparameters	Values
Effective batch size	4
Learning rate	5e-6 (GQA), 2e-5 (VisReas)
Precision	bfloat16
Optimiser	AdamW
Schedule	Linear warmup with cosine decay
Warmup steps	128
Epoch	1

Table 6: Hyperparameters for Logic2Vision model

Hyperparameters	Values
Rank	8
Alpha	16
Dropout	0.05

Table 7: LoRA configurations

C.4 InstructBLIP / BLIP-2 / LLaVA-1.5

On GQA, we use identical configuration as Logic2Vision for LLaVA-1.5. For InstructBLIP and BLIP-2, we observe that batch size of 4 causes the model to output repetitive tokens during inference. For that reason, we increase the effective batch size to 8. We use the same original prompt that the authors have reported in their original papers.

On VISREAS, we again use identical configuration as LOGIC2VISION for LLaVA-1.5. For InstructBLIP and BLIP-2, we lower the learning rate to 5e-6 and increase the effective batch size to 8 for the same reason above.

C.5 LXMERT / ViLBERT / CRF

For all three models trained with the classification task, we used the default hyperparameters that have been used to finetune on GQA corpus for consistency. As GQA and VISREAS share the same image and scenegraphs, using the same model with the same configuration should produce different results if the two tasks are different. And the result section reflects the distinction between GQA and VISREAS.

Hyperparameters	LXMERT	ViLBERT	CRF
Learning rate	1e-5	0.00004	1e-4
Optimizer	BertAdam	AdamW	BertAdam
Schedule	Linear Warmup	Linear Warmup	Linear Warmup
Epoch	4	20	13

Table 8: Hyperparameters of all CLS baselines

D Effect of pseudocode finetuning

We study the effect of finetuning a VLM to perform VQA through pseudocode-guided reasoning. Table 9 demonstrates that finetuning LLaVA-1.5 to follow pseudocode consistently improves performance on VISREAS for both 7B and 13B models.

Model size	Without Pseudocode	With Pseudocode
7B	57.36	62.74
13B	61.38	66.20

Table 9: Effect of pseucode finetuning on LLaVA-1.5

E Examples from VISREAS and GQA

In Figure 12, we show example questions from VISREAS and GQA using the same image. In general, VISREAS tends to have longer questions compared to GQA. Additionally, VISREAS questions involve more than two objects, whereas GQA primarily centers on one or two objects.

F Mechanical Turk Details

To evaluate human performance, we used Amazon Mechanical Turk to collect human responses for 5000 random questions, taking a majority vote among three workers for each question. We limited our pool of crowdworkers to individuals located in the US or Canada, requiring a minimum of 1,000 previously approved HITs with a 95% approval rate. Additionally, participants had to achieve a minimum score of 70% or higher on our qualification task before gaining access to our main task. In the subsequent sections, we provide details of this response collection process.

F.1 Qualification Test for Worker Selection

To secure accurate human assessments, we carefully designed a qualification test using Amazon Mechanical Turk interfaces (Figure 13). This test aimed to select proficient workers capable of accurately completing the VISREAS task: (1) The qualification test encompassed two distinct tasks. The initial task focused on careful comprehension of instructions. Workers were required to attentively read the instructions and subsequently answer a set of multiple-choice questions to assess their grasp of the task's nuances. (2) Upon successful completion of the first task, the qualified workers proceeded to the task proficiency evaluation stage. Here, a series of ten questions, each accompanied by an image, were presented. The workers' task was to select the correct answer from a dropdown list of 2013 entries. The selection process for the final evaluation cohort prioritized workers who achieved correct answers for more than seven out of the ten questions.

F.2 Human Accuracy Assessment Interfaces

After gathering qualified workers who are aware and proficient in our task, we move to the final stage of the evaluation process (Figure 14). For each Human Intelligence Task (HIT), an image and the corresponding question were provided. Workers were tasked with selecting the correct answer from the same dropdown list used for the worker selection stage. Furthermore, we requested workers to rate the complexity and structural integrity of the presented question, thereby acquiring insights into the inherent challenges posed by various question types.

To facilitate a deeper understanding of the potential issues with the queries, we encouraged workers

VisReas

Question: Are the doll and the soda bottle found sitting on or standing on the armchair in the image?
Answer: The question itself problematic
Explanation: There is no armchair present in the picture.
Category: verify.rel

GQA

Question: What kind of furniture is the doll to the left of the figurine sitting on?
Answer: Table
Category: query.obj

VisReas

Question: What is the common attribute of the pole, the road sign and the leaves which are to the right of the store in the picture?
Answer: Green
Category: query.attr

GQA

Question: Are the cars on the left or on the right side of the photo?
Answer: Right
Category: choose.rel

VisReas

Question: Among the floor, the doorway to the left of the red graffiti and the door, how many things are made of concrete?
Answer: Two
Category: count.attr

GQA

Question: What is the floor made of?
Answer: Concrete
Category: qurey.attr

VisReas

Question: Do the drawer and the floor to the right of the white shoes and to the left of the white dishwasher share the same material?
Answer: The question itself is problematic
Explanation: The floor is to the left of the white shoes and to the left of the white dishwasher
Category: compare.attr

GQA

Question: What is common to the drawer and the floor?
Answer: Material
Category: query.attr

VisReas

Question: Among knife, napkin, crust and wall, what object in the image has the same color as the plate and the coffee cup?
Answer: Napkin
Category: query.obj

GQA

Question: Is the coffee cup tall and white?
Answer: Yes
Category: verify.attr

VisReas

Question: Do you see any tiny stop sign on the large and metal post and any large flower?
Answer: No
Category: verfy.attr

GQA

Question: What's on the post?
Answer: Stop sign
Category: query.obj

VisReas

Question: What are the soap bottle, the bench and the pole in front of the brown trees made of in the image?
Answer: The question itself is problematic
Explanation: There are no soap bottle and bench present in the photo
Category: query.attr

GQA

Question: Are there any fences?
Answer: Yes
Category: verify.obj

Figure 12: Example questions from the VISREAS and the GQA corpuses.

to provide additional details about any perceived problems. If a worker identified a problematic aspect within the question, they were encouraged to rephrase or rewrite the query to address the issue. This dynamic engagement aimed to uncover underlying complexities and refine the evaluation process.

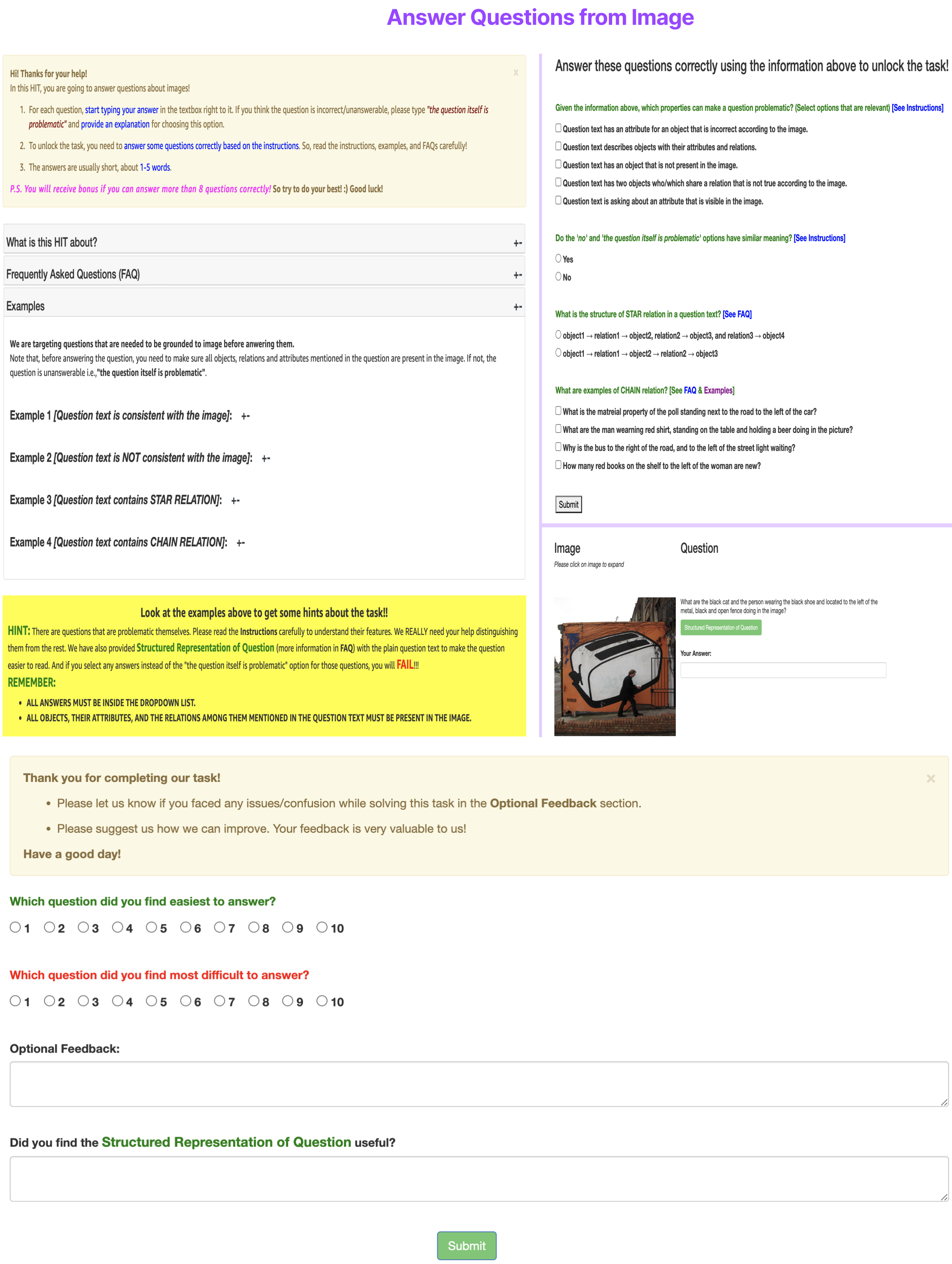
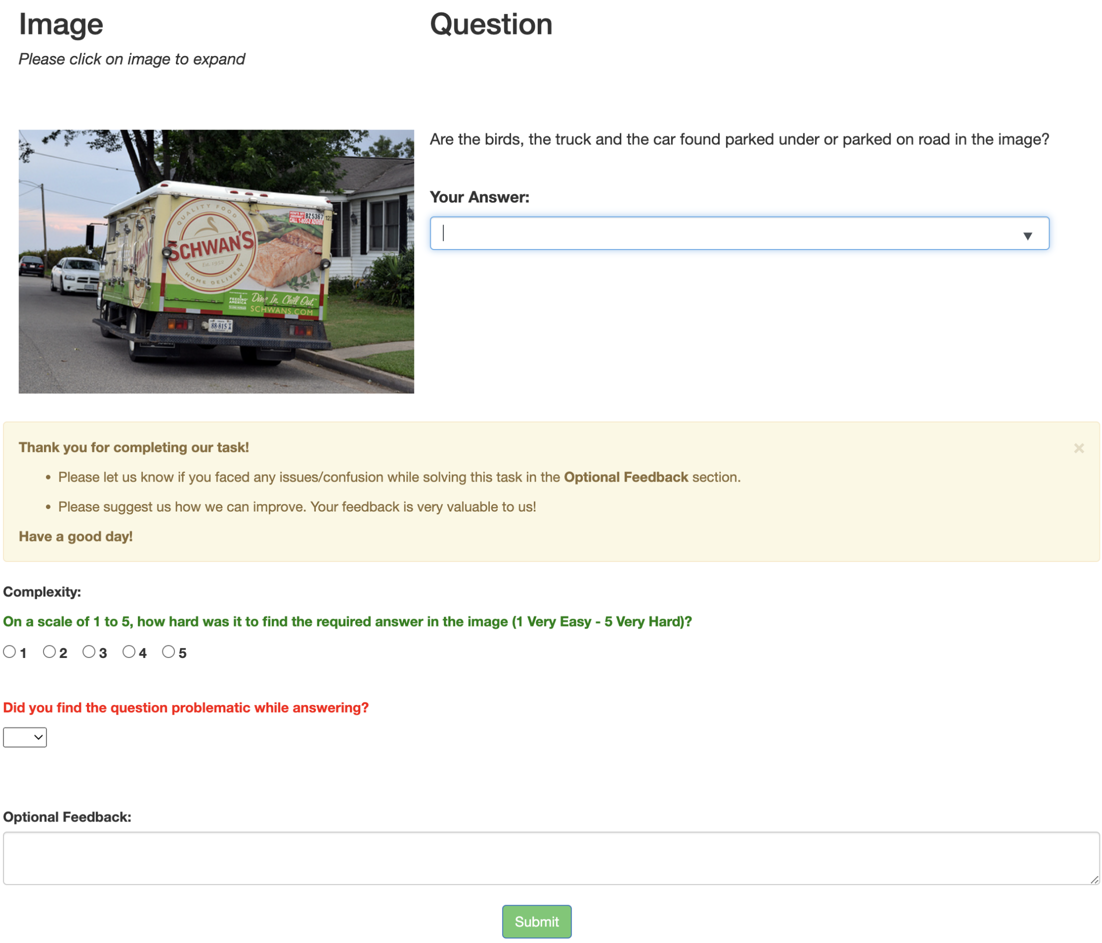
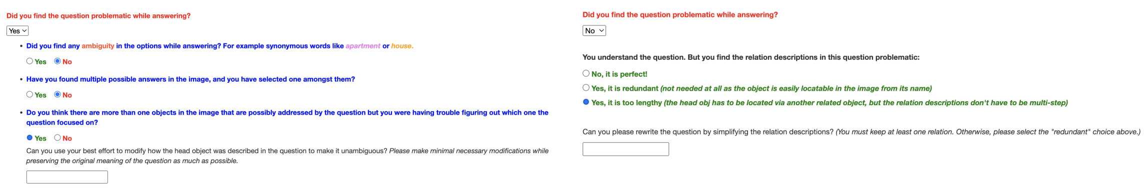


Figure 13: Amazon Mechanical Turk interfaces used for Qualification Test to choose the right workers for human accuracy assessment on VISREAS task. We study the workers by deploying two tasks. In the first task, we ask the workers to read the instructions carefully (**Top left**) and answer some multiple-choice questions (**Top right**). After passing this task, ten questions with images will be presented and the final task would be to choose the right answer from the answer dropdown list (**Bottom right**). We choose the workers for the final evaluation who have correctly predicted more than seven answers out of ten questions.



(a)



(b)

Figure 14: Amazon Mechanical Turk interfaces for human accuracy assessment on VISREAS task using the qualified workers. (a) For each HIT, we provide an image and a question that needs to be answered from a dropdown list of 2013 entries. In addition, we ask for rating the complexity and structural soundness of the query and further look for details if any Turker finds the question problematic. (b) To investigate what type of problem the question possesses, we ask for further details from the workers and even encourage them to rewrite the query to remove the problem they faced while answering the query.