FlowVQA: Mapping Multimodal Logic in Visual Question Answering with Flowcharts

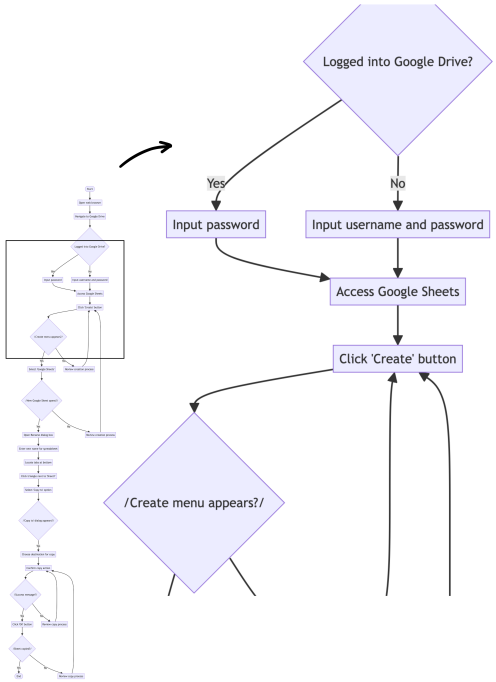
Shubhankar Singh[1†], Purvi Chaurasia[2†], Yerram Varun[3*], Pranshu Pandya[4*]
Vatsal Gupta[4*], Vivek Gupta[5‡], Dan Roth[5]
[1]Mercer Mettl, [2]IGDTUW New Delhi, [3]Google Research
[4]Indian Institute of Technology Guwahati, [5]University of Pennsylvania
shubhankar.singh@mercer.com, purvi069btcsai21@igdtuw.ac.in,
vyerram@google.com, {p.pandya,g.vatsal}@iitg.ac.in,
{gvivek, danroth}@seas.upenn.edu

Abstract

Existing benchmarks for visual question answering lack in visual grounding and complexity, particularly in evaluating spatial reasoning skills. We introduce FlowVQA, a novel benchmark aimed at assessing the capabilities of visual question-answering multimodal language models in reasoning with flowcharts as visual contexts. FlowVQA comprises 2,272 carefully generated and human-verified flowchart images from three distinct content sources, along with 22,413 diverse question-answer pairs, to test a spectrum of reasoning tasks, including information localization, decision-making, and logical progression. We conduct a thorough baseline evaluation on a suite of both open-source and proprietary multimodal language models using various strategies, followed by an analysis of directional bias. The results underscore the benchmark's potential as a vital tool for advancing the field of multimodal modeling, providing a focused and challenging environment for enhancing model performance in visual and logical reasoning tasks.

1 Introduction

Since the inception of Vision Language Models (VLMs), tasks and benchmarks for visual question answering (VQA) and reasoning have received significant attention. Most benchmarks evaluate pretrained model extraction capabilities, neglecting their ability to comprehend complex spatial relationships and visual logical reasoning. Research on spacial path following or visual sequential reasoning in VLMs is limited. Current benchmarks for assessing VLM reasoning abilities mainly fall under VQA, a task formalized by Goyal et al. (2017), which involves generating responses to questions based on a given image. These works evaluate the spatial reasoning and visual information extraction abilities of VLMs.



Q. Derek wants to ensure that the sheet was successfully copied before reporting back to Melissa. What should Derek see or do next to ensure the task was completed correctly?

A. He should look for a success message and dismiss the dialogue by clicking 'OK'.

Figure 1: A zoomed-in section of a flowchart in our resource set along with a correspnding QA pair. wiki00203: "How To Convert an Old Google Spreadsheet to Google Sheets." A detailed example of a flowchart along with its question-answer pairs is outlined in Appendix A.

Visual Grounding (VG) of a visual question-answering system evaluates models' abilities to attribute their generations to different image regions referenced in the query (Reich et al., 2023). The absence of VG has been a frequent issue among the current VQA systems, manifesting in an over-reliance on irrelevant parts of images or a complete disregard for the visual modality. Existing benchmarks (Yue et al., 2023) require models to rely on pre-trained knowledge to answer queries posed on image contexts. In this work, we aim to test the capabilities of VLMs in following visual information

*, † contributed equally, ‡ primary mentor & corresponding author

Findings of the Association for Computational Linguistics ACL 2024, pages 1330–1350
August 11-16, 2024

without any pre-existing knowledge. To accomplish this, we delve into the realm of flowcharts, as depicted in figure 1, which entail intricate structural configurations and path reconstruction, a considerably more challenging task compared to mere image comprehension.

Flowcharts **emphasize sequential and logical reasoning**, as they necessitate traversal of steps or decisions in a specific sequence. Flowcharts are **inherently visual**, and provide a clear and structured method for representing processes, decision paths, and flows. Unlike traditional text, which flows linearly, flowcharts require an understanding of **directional logic**; their flow is often multi-directional, representing various paths that can be taken based on certain conditions or decisions. Despite being long and complex, flowcharts have *compact*, *systematic* representations and provide insights regarding information in a step-by-step manner.

In this paper, we set out to answer a crucial question: *"Can modern Vision Language Models effectively handle challenges that demand understanding both structural and semantic aspects, along with capturing macroscopic and granular context within visually complex yet straightforward flowcharts?"* To tackle this question, we introduce **FlowVQA**, a novel benchmark comprising intricate structural and path-based questions posed on lengthy flowchart images. We propose a novel approach to Visual Question Answering (VQA) on Flowchart tailored for VLM, with a focus on harnessing flowcharts as the primary contextual framework for visual logic and spatial reasoning.

FlowVQA consists of 2,272 Mermaid.js flowchart scripts generated with human input, sourced from process workflow articles like Instructables and WikiHow, as well as Code. Accompanying these are 22,413 Q/A pairs covering various reasoning skills like information localization, fact retrieval, scenario deductions, flow reasoning, and topological understanding. The creation process involves a meticulous multi-step machine generation and human verification to discard up to 51% of samples, ensuring they meet high standards of challenge, coherence, and insightfulness. This rigorous process grounds the flowchart reasoning in textual domain, enriching the visual task complexity. Extensive experimentation revealed that both closed and open-source Vision Language Models (VLMs), equipped with a range of prompting strategies and fine-tuning techniques, struggled to execute visual and spatial reasoning tasks within the FlowVQA dataset. Moreover, our findings highlighted a directional bias and non-uniform performance pattern across flowcharts of varying lengths exhibited by these VLMs. Our contributions are the following:

- Introduction of VQA for FlowCharts, focusing on visual logic and spatial reasoning, filling a gap in previous benchmarks.
- Development of a detailed framework for generating intricate VQA samples transitioning from text to visual domains, ensuring quality, complexity, and accuracy via rigorous verification.
- Introducing the novel benchmark FlowVQA, consisting of 2,272 high-quality Flowchart Images and 22,413 Q/A samples spanning four distinct question types, created using the framework.
- Thorough evaluation of closed and open-source VLMs, employing various prompting strategies and fine-tuning methods. An analysis of directional bias and non-uniform performance across different flowchart lengths.

The FlowVQA dataset, along with modeling and evaluation scripts, generation pipeline and prompts, and the human verification platform, can be accessed at https://flowvqa.github.io/.

2 Proposed FlowVQA Resource

In this section, we will see the details of the construction of **FlowVQA**. We outline the process of collection of raw data from wild sources, multi-step generation of mermaid scripts and flowchart images and complex Q/A creation.

2.1 Flowchart Sources

We draw input texts from three primary sources: **WikiHow** articles, **Instructables** DIY blogs, and **FloCo** (Shukla et al., 2023a) code snippets. WikiHow and Instructables provide *step-by-step instructions* for everyday tasks, while the FloCo dataset, *a flowchart-to-code* resource, features low-complexity code samples. We categorize all the WikiHow articles, Instructables DIY based on the domains of these articles. FloCo code snippets are categorized into *code* category. The distribution across categories is outlined in Appendix B.

We manually select high-quality code snippets from FloCo to ensure uniformity in our pipeline across all text sources. FloCo image samples enable us to iteratively compare the generated

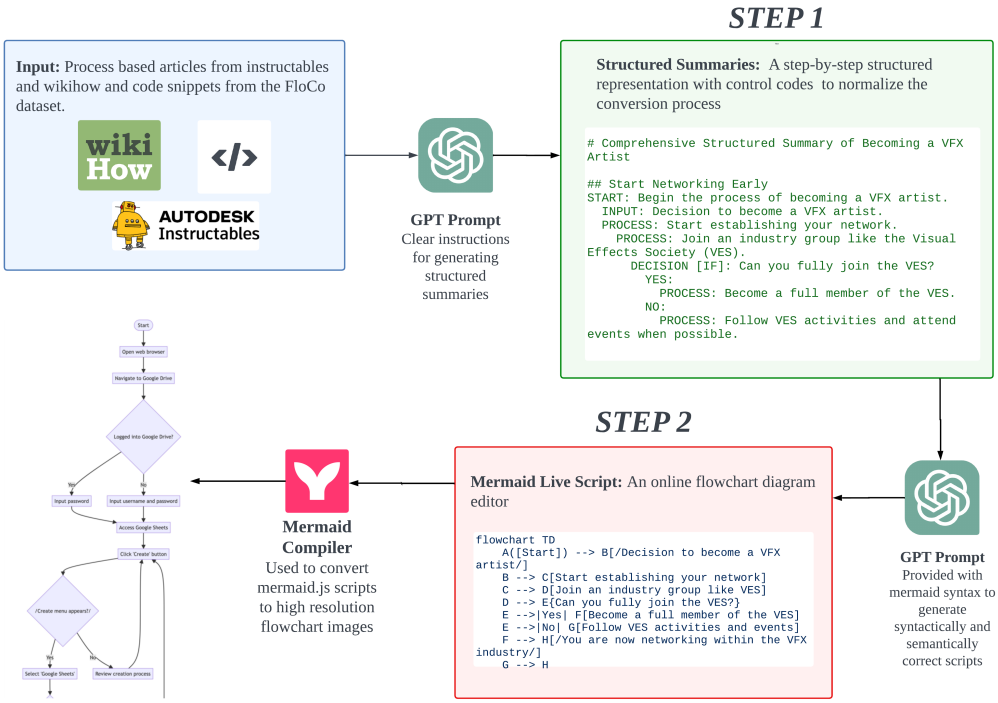


Figure 2: Our dataset's generation pipeline encompasses the creation of flowcharts. As previously outlined, we employ a comprehensive two-step process to derive high-quality flowcharts from source texts. Additionally, to guarantee accurate generation, a cross-verification mechanism is implemented.

flowcharts with the original samples. This step was crucial as it helped perfect our prompts and allow applicability to the WikiHow and Instructables set. We sample 1,268 WikiHow articles, 789 Instructables blogs, and 475 FloCo examples as an input to our human verification pipeline.

	WikiHow	Instructables	FloCo
Source Texts	1,914	943	700
Mermaid.js Scripts	1,500	792	575

Table 1: FlowVQA Generation resources.

Generation and Filteration. GPT-4 based data generation of data and benchmarks is prevalent (Han et al., 2023) in prior works. Machine generation method for flowcharts and Q/A has several advantages to crowdsourcing: (i) The complex and intricate process of creating flowcharts and Q/A pairs constitutes a laborious, efficient and a time-intensive task for human workers, (ii) Using GPT-4 for the generation of structured representations and subsequent conversion into flowcharts and Q/A pairs enables rapid scaling, (iii) The Stochastic nature of LLMs helps in the creation of an unbiased and diverse Q/A dataset. To produce Flowchart and Q/A Samples, we employ an automated 'generate-and-test' approach, where we exhaustively generate questions of multiple reasoning types and apply rigorous filtration to maintain the quality, hardness, and correctness of samples through effective prompting with GPT-4. Our meticulous verification through experts and rubrics, along with our custom-built annotation platform, ensures a thorough and impartial evaluation of both flowcharts and Q/A pairs.

2.2 Flowchart Generation

Our primary supposition for flowchart creation is that *any process-based workflow, regardless of domain, can be converted to a flowchart which highlights key aspects of the process in a detailed step-by-step fashion.* We treat the conversion of source article to flowchart Mermaid Scripts as a two-step soft-syntax summarization task. Ideally, we would use real-world flowcharts from external sources such as books and documents, but the availability of such structured data is extremely sparse. Initially, we aimed to use real-life flowcharts, but the scarcity of standardized flowcharts and the lack of sufficient open-source examples made it unfeasible to create a dataset as comprehensive as ours. We decouple the structured summarization into a flowchart script to implement this two-step process.

First Step. We query GPT-4 with the source text

Source	# Samples	Avg. NPF	Avg. EPF	Avg. Width	Avg. Height	Ratio	# Qs.
Wikihow	1,121	21.83	24.04	1568.0	5551.81	1 : 3.54	11,957
Instructables	701	19.76	21.18	1568.0	6629.80	1 : 4.23	6,893
Code	450	9.87	10.85	1568.0	2738.15	1 : 1.75	3,563
Full	2,272	18.82	20.54	1568.0	5327.13	1 : 3.40	22,413

Table 2: FlowVQA Source-wise Statistics: Number of Flowchart Samples, Average Nodes Per Flowchart, Average Edges per Flowchart, Average Image Width (Pixels), Average Image Height (Pixels), Aspect Ratio and Number of Questions. (The flowchart image render is set for a constant width factor)

to generate a step-by-step structured representation of the text annotated with functional control tags (e.g., "START," "PROCESS," "DECISION"). This step converts the source text into a tagged textual representation suitable for converting into mermaid flowchart scripts. For FloCo-sourced texts, we generate pseudocode for the code scripts as the input to the next step.

Second Step. In this step, we generate the Mermaid.js flowchart script(top-down) using the output of the *first step* by querying GPT-4 with a template Mermaid.js script. The control tags facilitate mapping the steps to the node types used in the script. Constraining points are provided alongside both prompts for improved normalization. The Mermaid.js scripts are then compiled to create high-resolution PNG images.

Table 1 represents the number of samples after the two-step conversion process. We exclude the scripts and representations with minor syntactical and rendering errors. Figure 2 showcases the generation pipeline of the flowcharts in our dataset. Appendix D.1 lists the prompts used in *first step* and *second step*.

2.3 Question Answer Creation

We curate four question types designed to analyze and test different aspects: Fact Retrieval, Applied Scenario, Flow Referential and Topological Question and Answer. First three can be broadly categorized under granular flowchart comprehension while topological tests structural information.

T1. Fact Retrieval: These simple questions involve the localization and retrieval of direct factual information from flowchart's nodes. Despite being simple, they still necessitate image analysis and retrieving relevant cues that localize the final answer.

T2. Applied Scenario: These questions describe a real-life scenario and test the models' application of the flowchart to a practical problem. These questions capture reasoning skills used by humans parsing flowcharts in day-to-day life. It leads to interesting puzzle-like word problems that test the understanding of decision steps, content, and reasoning in the presence of distractor context, which needs to be filtered to understand the question.

T3. Flow Referential: In these questions, A random sub-graph/section of the flowchart, usually involving a decision node, is considered, and a question is formulated on backward-forward flow with decision-based logic. It assesses granular path dynamics in a flowchart.

T4. Topological: This question type addresses the larger topology of a flowchart, requiring analysis of the flowchart at a more macroscopic level to give an answer related to the structural topology of the graph. These questions are created by parsing Mermaid.js scripts to convert them into an adjacency matrix representing the flowchart in the form of a graph. It generates template-based questions that usually have quantitative correct answers.

Statistics		Train	Test	Total
Total Flowcharts		1,319	953	2,272
Avg. Nodes		18.63	19.09	18.82
QA	Fact Retrieval	2,654	1,878	4,532
	Applied Scenario	2,640	1,936	4,576
	Flow Referential	2,128	1,585	3,713
	Topological	5,516	4,076	9,592
Total QA		12,938	9,475	22,413

Table 3: QA Resource Split Statistics.

Q/A Generation. We construct a prompt to query GPT-4 using the tagged textual representation, Mermaid.js script and text-only few-shot examples to generate high quality Q/A pairs of types, T1, T2 and T3 (listed in Appendix D.2). For each question, we generate three paraphrased gold answers, which allows us to evaluate models irrespective of their generation syntactics and semantics. As part of text-only few-shot examples we pass a variety of creative high-quality examples. Topological Q/A pairs (T4) are generated by parsing the Mermaid script, converting the graph into an adjacency matrix, and creating template-based questions. Answers are usually quantitative. After formulating the template-based answers, we

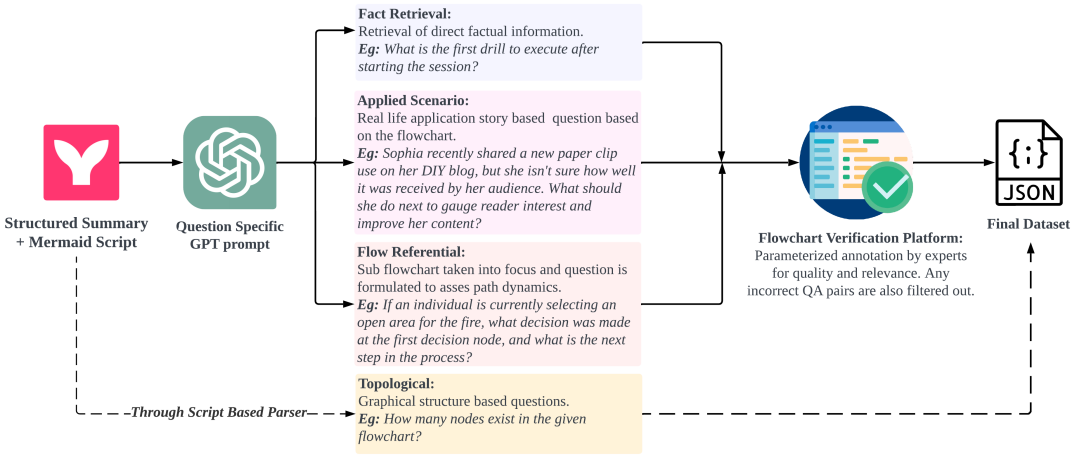


Figure 3: Our dataset incorporates a question creation pipeline tailored to accommodate various question types. As previously noted, each question type undergoes generation via a carefully crafted prompt, meticulously designed to achieve the specific objectives associated with that type of question

obtain two additional paraphrased answers for each template answer to achieve three gold-standard answers, thus maintaining the standard with the other question type for three gold short answers.

2.4 Human Verification Pipeline and Platform

To ensure strong validity of our work, we establish a robust human verification pipeline for our models and flowcharts. All generated outputs for flowcharts and subsequent Q/A pairs undergo a rigorous quality check by a team of five expert annotators. As we adhere to a "Generate-and-test" paradigm (section 2), we provide detailed rubrics for both flowchart and Q/A pair verification and annotation, with parameters such as logical flow, complexity, context alignment and more, for flowcharts and Q/A pairs which allow the annotators be strict and thorough. To assist with their work and eliminate any bias and stress, we also provide them with a detailed, custom-built annotation platform to provide scores, filter out, etc. This custom platform enables parallel viewing.

	# Samples	# T1	# T2	# T3
Pre	2,532	8,932	9,138	7,262
Post	2,272	4,532	4,576	3,713
% drop	10.3%	49.3%	50%	48.9%

Table 4: FlowVQA Annotation-based filtering stats pre and post-verification and filtration for number of flowchart samples and QA Types *T1*, *T2* and *T3*

.

Annotation Platform. Our custom-built annotation platform consists of UI, where we pass the flowchart and Q/A pairs together so they can be viewed simultaneously. The annotators provide quality scores[1] for all components of the dataset and a final holistic score[2]. We filter out flowcharts below a fixed quality threshold and Q/A pairs which rate below average. Topological questions are not passed into the platform as they are hard-template-based and obtained via scripting. All verification products are verified with two separate supervising experts who ensure the quality of annotations is consistent and scores remain unbiased. The verification lasts for ten days from start to end.

The final samples, see Appendix A, ensure appropriate complexity and correctness of flowcharts, questions and corresponding answers. Figure 3 showcases the complete question-answer generation pipeline used to create the dataset.

3 Experimental Evaluation

We address the following research questions through our experiments:

RQ1. Does the introduced visual multimodal dataset present a significant challenge to current multimodal language learning models (VLMs), and can it provide valuable insights that could contribute to their future advancement?

RQ2. Is the efficacy of VLMs influenced by factors such as (a) the source of flowcharts, (b) the type of questions posed, and (c) the level of complexity inherent in the flowcharts?

RQ3. Are there ways to enhance the perfor-

[1]captures the consistency, correctness and complexity.
[2]captures the relevancy between the components.

mance of visual question answering tasks related to flowcharts through the use of specific directives tailored to flowcharts? Moreover, does the process of fine-tuning these models with the train split of FlowVQA dataset improve their proficiency in handling questions tied to flowchart-based data?

RQ4. Is there an observable directional bias in existing VLMs when they are applied to flowchart analysis?

Limitations of Smaller Models. FlowVQA represents a complex multimodal challenge that requires visual logic and reasoning across large-scale high-resolution images. In our assessment of several widely utilized open-source multimodal language learning models (VLMs) – including **LLaVA** (Liu et al., 2023b), **OpenFlamingo** (Awadalla et al., 2023), **BLiPv2** (Li et al., 2023a), **mPLUG-OWL** (Ye et al., 2023b), Sphinx (Lin et al., 2023) — we observe that their performance on our test dataset **is notably subpar (<10%)**. These multimodal language learning models (VLMs) lack a sizable vision encoder, leading to the internal distortion of flowchart images with high aspect ratios when passed into the vision encoder. Furthermore, even if they can interpret the image a bit, their inadequate reasoning abilities render them extremely ineffective for any further analysis utilizing this resource.

Models for Comparison. We perform evaluations on FlowVQA with **five** different VLMs. We employ **GPT-4V** (OpenAI, 2023) and **Gemini Pro** (Anil et al., 2023)[3] to test the visual understanding capabilities of best proprietary (closed) models available. We also employ three open-source models. **CogAgent-VQA** (Hong et al., 2023) is an 18- billion-parameter visual language model (VLM) specializing in GUI understanding and navigation (fine tuned on smaller VQA Tasks). This model supports inputs at the resolutions of 1120x1120, enabling it to recognize tiny page elements and text in the flowcharts.

InternLM-X-Composer2 (Dong et al., 2024) uses a novel approach (PLORA) that applies additional LoRA parameters exclusively to image tokens to ensure that linguistic abilities are not affected, striking a balance between precise vision understanding and text composition. **Qwen-VL-chat** (Bai et al., 2023) is the instruction tuned model in the Qwen-VL series. Its *position-aware vision language adapter* ensures that, even though the images are resized to a fixed resolution long image feature contexts are captured effectively by the model. We summarize the base language models and visual models used in our baselines in Table 5.

[3]We use the preview version for Gemini Pro at Vertex API (Vertex). Gemini Ultra is/was not made public yet.

Open Model	LM	VM	Norm. Res.
CogAgent-VQA	Vicuna-7B	ViT-4.4B	1120x1120
InternLM$_{\text{X-Comp.2}}$	Intern-LM2-7B	ViT-304M	490x490
Qwen-VL-chat	Qwen-VL-7B	ViT-1.9B	448x448

Table 5: Open Baseline Models. VLMs are composed of a Language model that encodes text and a visual model that encodes the images. LM: Language Model, VM: denotes vision model.

3.1 Baseline Evaluation

We evaluate the baseline models under multiple settings:

1. **Zero-Shot**: Given a flowchart, we prompt the VLM to answer the question with a small instruction and provide a short concise answer.
2. **Zero-Shot CoT**: Given a flowchart, we prompt the VLM with the question to first elicit a rationale and then deduce the final answer (Wei et al., 2023).
3. **Text Only Few-Shot CoT with Reasoning Directives**: We create a custom prompt outlining the reasoning steps involved in answering questions specific to flowcharts. We scrutinize the areas where improved prompting is necessary for the models and draw inspiration from (Zhang et al., 2023), (Li et al., 2023b), and (Kojima et al., 2023) to devise a text-only few-shot CoT approach with directional stimulus and step-by-step reasoning. The central objective is to deconstruct complex questions, identify which elements to map, and determine the answer. Each example, or "shot," encompasses four key components: The Question, Directional Stimulus Tags, Step-by-Step Rationale, and the Answer. These distinct parts aid in breaking down the question into relevant segments, offering a logical, step-by-step analysis, and concluding with an answer. We develop this strategy based on its potential effectiveness for flowcharts, with its actual efficacy demonstrated ahead. The few-shot samples we give are dynamic in nature, i.e the each question type gets more similar samples from our train set annotated samples samples for the method.
4. **Fine-Tuning**: We fine-tune the VLM on the

train split of FlowVQA, and then prompt the VLM to answer the question.[4]

3.2 Evaluation Method

Our methodology adopts an *"AI as an Evaluator"* approach similar to Fu et al. (2023); Lin and Chen (2023); Chiang and Lee (2023). We employ three evaluator models—GPT-3.5 (Ye et al., 2023a), Llama-2 70B (Touvron et al., 2023), and Mixtral 8*7B (Mixtral-of-Experts) (Jiang et al., 2024) —to assess the model-generated responses, which are compared against three gold standard short answers and the question (context excluded). The evaluators' task is to dissect and align the responses, eliciting a detailed rationale that demonstrates Chain of Thought behavior, and then assigning a binary label to indicate whether the response is correct or incorrect. This process essentially boils down the evaluation into a "length-invariant" paraphrase detection task for short text responses, surpassing traditional similarity metrics and rule-based matching in effectiveness. We determine the final label via a majority vote among the evaluator models.

Fine-tuning Settings. We fine-tune Qwen-VL-chat $_{\text{FT}}$ using LORA (Hu et al., 2022) strategy on 2xNVIDIA A100 40GB GPUs. We train with an effective batch size of 8 using a cosine-based learning scheduler with a warmup. We set a higher warmup to ensure no loss of pretraining knowledge in the base model.

3.3 Baseline Results and Discussion

Table 6 tabulates the results of model evaluations across multiple strategies, with the scores split across various question types and text sources. Figure 5 in Appendix C provides a horizontal bar chart that compiles the results from the table.

FlowVQA is sufficiently hard. The dataset resource presents a challenging task, with all the models. The evaluations highlight a scope for improvement for all the models. Our Best performing model with the top performing strategy, i.e. GPT-4 prompted with Few-shot directive-based prompting achieves 68.42% Majority voting across all the evaluators.

Few-Shot Directives are helpful. In the evaluation of most of our models, we observe that text-only few-shot CoT with reasoning directives outperforms other prompting strategies. We observe 7% improvement in GPT-4 evaluation and 12% improvement in Gemini-Pro with this strategy. CogAgent-VQA , however does not show an improvement with few-shot directives. We observe in our initial experiments that it was unable to generate directives and hence it could not make use of reasoning directives.

Proprietary models perform better than open-source models. We observe that proprietary models heavily outperform the open-source models. GPT-4 with few-shot directives outperforms Qwen-VL-chat by a significant 30%.

Fine-tuning helps. We fine-tune Qwen-VL-chat and evaluate by prompting with Zero-Shot and Zero-Shot CoT strategies. We see an improvement of 3% from Zero-Shot prompting and 11% improvement from Zero-Shot COT. This improvement emphasises the lack of flowchart understanding in original pretraining mixtures of these VLMs. The improvement in T2, T3 and T4 (10%) being more significant than T1 (5%), can be attributed to the fact that fact-retrieval is a simpler task and does not need in-depth understanding of the flowchart structure. The fine-tuned model outperforms all other existing open-source models, which highlights the fact that *FlowVQA* can be effectively used to introduce visual logic and reasoning in existing VLMs.

Question Types. We present the question-wise metrics in Table 5. It is evident from the table that all models consistently perform better on *Fact Retrieval (T1)* and *Applied Scenario (T2)* based based questions than on *Flow-Referential (T3)* and *Topological (T4)*. Outlined in Sec. 2.3, *T3* and *T4* question types require thorough understanding of the flowchart and complex reasoning over the visual modality.

Number of Nodes. Using the Mermaid.js scripts, we obtain the count of nodes in each flowchart. We categorize the flowchart by binning the number of nodes present in them. A Large number of nodes implies a more complex representation of visual information, and hence the flowchart is harder to reason upon. The results in the Table 7 confirms this fact. Figure 7 in Appendix C shows the decline of performance of models with increase in number of nodes.

[4]Due to resource constraints we only Fine-Tune on Qwen-VL-Chat through LoRA Finetuning

Model	Strategy	MV_{Total}	MV_{T1}	MV_{T2}	MV_{T3}	MV_{T4}	MV_{Wiki}	$MV_{Instruct}$	MV_{Code}
GPT-4V	**Zero-Shot**	61.22	**90.72***	82.24	63.79	40.62	60.98	60.78	62.65
	Zero-Shot COT	65.57	72.79	69.94	73.50	**58.25***	**67.84***	70.89	47.71
	Few-Shot COT_D	**68.42***	89.02	**89.92***	**81.41***	46.72	63.33	**72.25***	**64.83***
Gemini-Pro-V	**Zero-Shot**	49.57	80.08	70.29	35.34	33.86	48.84	48.27	54.36
	Zero-Shot COT	58.76	81.21	78.39	62.14	41.99	54.23	57.57	63.81
	Few-Shot COT_D	61.41	84.96	81.83	77.69	43.60	54.12	60.12	61.41
CogAgent-VQA	**Zero-Shot**	37.17	55.27	52.68	26.56	27.23	37.45	36.80	36.96
	Zero-Shot COT	38.84	58.73	57.95	27.51	26.98	40.01	37.47	37.64
	Few-Shot COT_D	25.13	33.93	34.26	16.76	21.67	34.62	29.65	22.37
$InternLM_{X\text{-}Comp.2}$	**Zero-Shot**	37.47	49.47	49.79	24.16	32.15	35.67	38.26	41.90
	Zero-Shot COT	43.35	58.85	**65.58#**	33.86	31.39	43.24	41.48	47.16
	Few-Shot COT_D	45.09	58.96	64.80	38.56	32.64	45.05	**43.03#**	**47.74#**
Qwen-VL-chat	**Zero-Shot**	33.67	48.83	46.64	20.19	26.89	32.92	34.02	35.47
	Zero-Shot COT	36.19	49.84	53.82	22.65	28.13	36.01	35.41	38.32
	Few-Shot COT_D	38.44	57.21	57.00	25.13	27.98	40.76	37.75	32.94
Qwen-VL-chat $_{FT}$	**Zero-Shot**	36.84	56.95	49.86	25.75	25.77	39.64	34.63	32.51
	Zero-Shot COT	**47.13#**	**61.55#**	59.78	**43.34#**	**36.02#**	**50.10#**	42.14	47.67

Table 6: Majority Vote Accuracy on All Models and Strategies broken down Question Type Wise (*T1, T2, T3, T4*) as in Sec 2.3 and Source-Wise (Instruct, Wiki, Code) as in Table 2. The highest value for each column is highlighted and marked with * in Closed Source Models and with # in Open Source Models.

Number of Nodes	Average Accuracy
0-8	51.73
8-17	45.74
17-26	44.60
35-44	38.99
26-35	40.35

Table 7: Number of Nodes comparison (Average across all models and strategies). Performance decreases as number of nodes increases.

3.4 Directional Bias

To study ***RQ4***, we parse the mermaid scripts of the FlowVQA flowcharts and systematically invert them to produce a inverted flowchart **"Bottom Top"** set. Bottom Top analysis helps further evaluate the Visual and Sequential nature of our resource. The Bottom Top Flowcharts look directionally counter-intuitive with the start nodes at the bottom and end at the top. We perform this inversion on 1,500 flowchart-question pairs on which all evaluators evaluate to "True" (correct response for all). We evaluate a the top-performing models and strategies obtained in Section 3.1 on the inverted flowchart set to detect any presence of directional bias in the VLMs.

Table 8 highlights the fact that our best performing models do *suffer from a directional bias* in understanding and reasoning over flowcharts. We see a significant 15% drop in majority voting accuracy thorough with GPT-4.

Model (Strategy)	Top-Down	Bottom-Up
GPT-4V $_{(CoT)}$	100.00	85.71
Qwen-VL-chat $_{(CoT)}$	100.00	76.09

Table 8: Directional Bias test, we evaluate on two models using CoT approach on 1500 flowchart-QA pairs.

Analysis. The directional bias evaluation underlines an important lacking of existing VLMs. They suffer from biases introduced in pretraining mixture and do not ground their inferences in the context images which leads to a significant drop in their evaluation performances. Strategies like augmenting pretraining mixtures with counterfactual examples might help alleviating these issues, which we leave for future study.

4 Related Work

Vision language models have made large progress in diverse vision-language applications (Bai et al., 2023; Liu et al., 2023a; Xia et al., 2024) with multiple benchmarks being proposed to aid effective evaluation of visual and textual grounding capabilities of these models. The MMMU benchmark (Yue et al., 2023) is designed to assess the model's inherent "subject-specific" knowledge and reasoning abilities across various subjects (such as Technology, Humanities, Health, and more).

Benchmarks like TextVQA and DocVQA (Singh et al., 2019; Mathew et al., 2021b; Zellers et al.,

2019; Park et al., 2020; Lu et al., 2022; Hudson and Manning, 2019) evaluate the models' fine-grained transcription abilities on low-resolution images. More complex multimodal reasoning tasks, such as MathVista (Lu et al., 2024), examine the models' abilities to integrate visual and mathematical logic. Benchmarks focusing on spatial multimodal reasoning include ChartQA (Masry et al., 2022; Xu et al., 2023; Methani et al., 2020) and InfographicVQA (Mathew et al., 2021a). ChartQA is aimed at evaluating straightforward chart understanding and analysis, while InfographicQA poses direct logical questions about data visualizations and charts.

Prior Flowchart Works. To our knowledge, there exists a study on Flowchart QA (Tannert et al.), that suffers from major limitations. (i) Synthetically generated flowcharts with randomized scripts, (ii) Primarily poses structural questions and (iii) Uses multiple choice-based questions to evaluate weaker existing models. Other research in this domain addresses issues like Flowchart Object Recognition and Flowchart to Code/Script conversion, where a modest parallel flowchart resource is paired with corresponding code or script (Liu et al., 2022; Shukla et al., 2023b; Thean et al., 2012; Sun et al., 2022). However, notable limitations here include poor flowchart image quality, niche or overly complex context, structural imbalance (only linear or excessively complex), lack of ground truth scripts for flowcharts, and insufficient context for effective Q/A or practical tasks. In contrast to these works, we construct a complex benchmark suitable to test practical applicability of existing VLMs. The complex and diverse QA types ensure an effective and just evaluation over multi-modal visual and textual understanding.

5 Conclusion and Future Work

In conclusion, this study evaluates the effectiveness of existing Multimodal Large Language Models (VLMs) in reasoning upon a complex visual, sequential logical reasoning based task, *FlowVQA*. We introduce the novel dataset resource, *FlowVQA*, consisting of 2,272 Flowchart images, Mermaid.js scripts, 22,413 Q/A pairs with gold standard answers. Our extensive evaluation on these models with multiple strategies and scenarios highlights the need for advancements in **architecture** and **prompting** strategies in existing VLMs. We also study the presence of any *directional* bias in the flowcharts by re-evaluating the test sets with an inverted flowchart subset. We find that both proprietary and open-source models suffer from directional bias due to lack of visual grounding and complex structural reasoning required for flowchart reasoning.

Future Work. Our work and resources give rise to many research avenues in (a) **Flowchart Reasoning**: *FlowVQA* can be used to enhance the visual logic and reasoning capabilities of the models. Constructing VLMs that are flowchart specific is also a encouraging research direction. (b) **Graph-Encoder Models**: In this study, we consider the graph nature of flowcharts solely to generate topological questions. This consideration can also be taken into account while designing model architectures and inference strategies to enhance structural reasoning in the base models. (c) **Adversarial and Counterfactual probes**: We provide questions of four different types which can be augmented with multiple probe sets like negative path following, counter-intuitive questions and noisy-graph based questions. (d) **Complex Subtasks**: The parallel nature of *FlowVQA* allows us to formulate multiple subtasks using the resource. Primary task of *FlowVQA* is the *Flowchart→Q/A*. We can create multitude of tasks: *article→Q/A*, *Mermaid.js→Q/A*, *Flowchart→Mermaid.js*. The tasks can then act as an additional resource for training LLMs and VLMs. (e) **NeuroSymbolic AI Approaches** like in Trinh et al. (2024) can also be considered to enhance performance and training on our resource as flowcharts are inherently symbolic and sequential structures.

Limitations

There are a few notable limitations to our work. Primarily, the inability to fine-tune all models under consideration due to financial and computational resource constraints has led to a potential underrepresentation of the capabilities of various NLP models beyond our primary focus. Moreover, the language limitations encountered in this research, particularly the focus on English for generating Visual Question Answering (VQA) methods, underscore the need for linguistic diversity in NLP applications to ensure broader applicability and inclusivity. Given the novelty of the task at hand, it is also important to acknowledge that the insights provided may not be exhaustive, highlighting the potential for future research.

Ethics Statement

We, the authors of this work, confirm that our research adheres to the highest ethical standards in both research and publication. Throughout this study, we have diligently considered and addressed various ethical issues to ensure the responsible and equitable use of computational linguistics methodologies. To promote the reproducibility of our results, we provide comprehensive information, including sharing code, datasets (we use publicly available datasets and comply with the ethical standards set by their authors), and other relevant resources. This enables the research community to validate and extend our work. The claims presented in this paper are consistent with our experimental results. However, given the inherent stochasticity of *black-box* large language models, we have minimized variability by maintaining a fixed temperature. We thoroughly detail the annotations, dataset splits, models used, and prompting methods employed, ensuring the reproducibility of our work.

Acknowledgements

Research was sponsored by the Army Research Office and was accomplished under Grant Number W911NF-20-1-0080. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the Army Research Office or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation herein. This work was partially funded by ONR Contract N00014-19-1-2620. We extend our gratitude to the annotators who verified our flowcharts and corresponding question answer pairs. Lastly, we extend our appreciation to the reviewing team for their insightful comments.

References

Rohan Anil, Sebastian Borgeaud, Yonghui Wu, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M. Dai, Anja Hauth, Katie Millican, David Silver, Slav Petrov, Melvin Johnson, Ioannis Antonoglou, Julian Schrittwieser, Amelia Glaese, Jilin Chen, Emily Pitler, Timothy P. Lillicrap, Angeliki Lazaridou, Orhan Firat, James Molloy, Michael Isard, Paul Ronald Barham, Tom Hennigan, Benjamin Lee, Fabio Viola, Malcolm Reynolds, Yuanzhong Xu, Ryan Doherty, Eli Collins, Clemens Meyer, Eliza Rutherford, Erica Moreira, Kareem Ayoub, Megha Goel, George Tucker, Enrique Piqueras, Maxim Krikun, Iain Barr, Nikolay Savinov, Ivo Danihelka, Becca Roelofs, Anaïs White, Anders Andreassen, Tamara von Glehn, Lakshman Yagati, Mehran Kazemi, Lucas Gonzalez, Misha Khalman, Jakub Sygnowski, and et al. 2023. Gemini: A family of highly capable multimodal models. *CoRR*, abs/2312.11805.

Anas Awadalla, Irena Gao, Josh Gardner, Jack Hessel, Yusuf Hanafy, Wanrong Zhu, Kalyani Marathe, Yonatan Bitton, Samir Yitzhak Gadre, Shiori Sagawa, Jenia Jitsev, Simon Kornblith, Pang Wei Koh, Gabriel Ilharco, Mitchell Wortsman, and Ludwig Schmidt. 2023. Openflamingo: An open-source framework for training large autoregressive vision-language models. *CoRR*, abs/2308.01390.

Jinze Bai, Shuai Bai, Shusheng Yang, Shijie Wang, Sinan Tan, Peng Wang, Junyang Lin, Chang Zhou, and Jingren Zhou. 2023. Qwen-vl: A versatile vision-language model for understanding, localization, text reading, and beyond. *arXiv preprint arXiv:2308.12966*.

Cheng-Han Chiang and Hung-yi Lee. 2023. Can large language models be an alternative to human evaluations? In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 15607–15631, Toronto, Canada. Association for Computational Linguistics.

Xiaoyi Dong, Pan Zhang, Yuhang Zang, Yuhang Cao, Bin Wang, Linke Ouyang, Xilin Wei, Songyang Zhang, Haodong Duan, Maosong Cao, Wenwei Zhang, Yining Li, Hang Yan, Yang Gao, Xinyue Zhang, Wei Li, Jingwen Li, Kai Chen, Conghui He, Xingcheng Zhang, Yu Qiao, Dahua Lin, and Jiaqi Wang. 2024. Internlm-xcomposer2: Mastering free-form text-image composition and comprehension in vision-language large model. *CoRR*, abs/2401.16420.

Jinlan Fu, See-Kiong Ng, Zhengbao Jiang, and Pengfei Liu. 2023. Gptscore: Evaluate as you desire.

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.

Yucheng Han, Chi Zhang, Xin Chen, Xu Yang, Zhibin Wang, Gang Yu, Bin Fu, and Hanwang Zhang. 2023. Chartllama: A multimodal LLM for chart understanding and generation. *CoRR*, abs/2311.16483.

Wenyi Hong, Weihan Wang, Qingsong Lv, Jiazheng Xu, Wenmeng Yu, Junhui Ji, Yan Wang, Zihan Wang, Yuxuan Zhang, Juanzi Li, Bin Xu, Yuxiao Dong, Ming Ding, and Jie Tang. 2023. Cogagent: A visual language model for GUI agents. *CoRR*, abs/2312.08914.

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 *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net.

Drew A. Hudson and Christopher D. Manning. 2019. Gqa: A new dataset for real-world visual reasoning and compositional question answering.

Albert Q. Jiang, Alexandre Sablayrolles, Antoine Roux, Arthur Mensch, Blanche Savary, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Emma Bou Hanna, Florian Bressand, Gianna Lengyel, Guillaume Bour, Guillaume Lample, Lélio Renard Lavaud, Lucile Saulnier, Marie-Anne Lachaux, Pierre Stock, Sandeep Subramanian, Sophia Yang, Szymon Antoniak, Teven Le Scao, Théophile Gervet, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2024. Mixtral of experts.

Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. 2023. Large language models are zero-shot reasoners.

Junnan Li, Dongxu Li, Silvio Savarese, and Steven C. H. Hoi. 2023a. BLIP-2: bootstrapping language-image pre-training with frozen image encoders and large language models. In *International Conference on Machine Learning, ICML 2023, 23-29 July 2023, Honolulu, Hawaii, USA*, volume 202 of *Proceedings of Machine Learning Research*, pages 19730–19742. PMLR.

Zekun Li, Baolin Peng, Pengcheng He, Michel Galley, Jianfeng Gao, and Xifeng Yan. 2023b. Guiding large language models via directional stimulus prompting.

Yen-Ting Lin and Yun-Nung Chen. 2023. Llm-eval: Unified multi-dimensional automatic evaluation for open-domain conversations with large language models.

Ziyi Lin, Chris Liu, Renrui Zhang, Peng Gao, Longtian Qiu, Han Xiao, Han Qiu, Chen Lin, Wenqi Shao, Keqin Chen, Jiaming Han, Siyuan Huang, Yichi Zhang, Xuming He, Hongsheng Li, and Yu Qiao. 2023. SPHINX: the joint mixing of weights, tasks, and visual embeddings for multi-modal large language models. *CoRR*, abs/2311.07575.

Haotian Liu, Chunyuan Li, Yuheng Li, and Yong Jae Lee. 2023a. Improved baselines with visual instruction tuning. *CoRR*, abs/2310.03744.

Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. 2023b. Visual instruction tuning. *CoRR*, abs/2304.08485.

Zejie Liu, Xiaoyu Hu, Deyu Zhou, Lin Li, Xu Zhang, and Yanzheng Xiang. 2022. Code generation from flowcharts with texts: A benchmark dataset and an approach. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 6069–6077, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Pan Lu, Hritik Bansal, Tony Xia, Jiacheng Liu, Chunyuan Li, Hannaneh Hajishirzi, Hao Cheng, Kai-Wei Chang, Michel Galley, and Jianfeng Gao. 2024. Mathvista: Evaluating mathematical reasoning of foundation models in visual contexts.

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. In *Advances in Neural Information Processing Systems*, volume 35, pages 2507–2521. Curran Associates, Inc.

Ahmed Masry, Do Xuan Long, Jia Qing Tan, Shafiq Joty, and Enamul Hoque. 2022. Chartqa: A benchmark for question answering about charts with visual and logical reasoning.

Minesh Mathew, Viraj Bagal, Rubèn Pérez Tito, Dimosthenis Karatzas, Ernest Valveny, and C. V Jawahar. 2021a. Infographicvqa.

Minesh Mathew, Dimosthenis Karatzas, and C. V. Jawahar. 2021b. Docvqa: A dataset for vqa on document images.

Nitesh Methani, Pritha Ganguly, Mitesh M. Khapra, and Pratyush Kumar. 2020. Plotqa: Reasoning over scientific plots. In *IEEE Winter Conference on Applications of Computer Vision, WACV 2020, Snowmass Village, CO, USA, March 1-5, 2020*, pages 1516–1525. IEEE.

OpenAI. 2023. GPT-4 technical report. *CoRR*, abs/2303.08774.

Jae Sung Park, Chandra Bhagavatula, Roozbeh Mottaghi, Ali Farhadi, and Yejin Choi. 2020. Visualcomet: Reasoning about the dynamic context of a still image. In *Computer Vision – ECCV 2020*, pages 508–524, Cham. Springer International Publishing.

Daniel Reich, Felix Putze, and Tanja Schultz. 2023. Measuring faithful and plausible visual grounding in VQA. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 3129–3144, Singapore. Association for Computational Linguistics.

Shreya Shukla, Prajwal Gatti, Yogesh Kumar, Vikash Yadav, and Anand Mishra. 2023a. Towards making flowchart images machine interpretable. In *Document Analysis and Recognition - ICDAR 2023 - 17th International Conference, San José, CA, USA, August 21-26, 2023, Proceedings, Part V*, volume 14191 of *Lecture Notes in Computer Science*, pages 505–521. Springer.

Shreya Shukla, Prajwal Gatti, Yogesh Kumar, Vikash Yadav, and Anand Mishra. 2023b. Towards making flowchart images machine interpretable. In *Document Analysis and Recognition - ICDAR 2023: 17th International Conference, San José, CA, USA, August 21–26, 2023, Proceedings, Part V*, page 505–521, Berlin, Heidelberg. Springer-Verlag.

Amanpreet Singh, Vivek Natarajan, Meet Shah, Yu Jiang, Xinlei Chen, Dhruv Batra, Devi Parikh, and Marcus Rohrbach. 2019. Towards vqa models that can read.

Lianshan Sun, Hanchao Du, and Tao Hou. 2022. Fr-detr: End-to-end flowchart recognition with precision and robustness. *IEEE Access*, 10:64292–64301.

Simon Tannert, Marcelo Feighelstein, Jasmina Bogojeska, and Joseph Shtok. Flowchartqa. https://document-intelligence.github.io/DI-2022/files/di-2022_final_11.pdf.

Andrew Thean, Jean-Marc Deltorn, Patrice Lopez, and Laurent Romary. 2012. Textual summarisation of flowcharts in patent drawings for clef-ip 2012. In *Conference and Labs of the Evaluation Forum*.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. 2023. Llama 2: Open foundation and fine-tuned chat models.

Trieu H. Trinh, Yuhuai Wu, Quoc V. Le, He He, and Thang Luong. 2024. Solving olympiad geometry without human demonstrations. *Nature*, 625(7995):476–482.

Google Vertex. Gemini pro api. Accessed on Feb 4, 2024.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc Le, and Denny Zhou. 2023. Chain-of-thought prompting elicits reasoning in large language models.

Renqiu Xia, Bo Zhang, Hancheng Ye, Xiangchao Yan, Qi Liu, Hongbin Zhou, Zijun Chen, Min Dou, Botian Shi, Junchi Yan, and Yu Qiao. 2024. Chartx & chartvlm: A versatile benchmark and foundation model for complicated chart reasoning. *CoRR*, abs/2402.12185.

Zhengzhuo Xu, Sinan Du, Yiyan Qi, Chengjin Xu, Chun Yuan, and Jian Guo. 2023. Chartbench: A benchmark for complex visual reasoning in charts. *CoRR*, abs/2312.15915.

Junjie Ye, Xuanting Chen, Nuo Xu, Can Zu, Zekai Shao, Shichun Liu, Yuhan Cui, Zeyang Zhou, Chao Gong, Yang Shen, Jie Zhou, Siming Chen, Tao Gui, Qi Zhang, and Xuanjing Huang. 2023a. A comprehensive capability analysis of gpt-3 and gpt-3.5 series models.

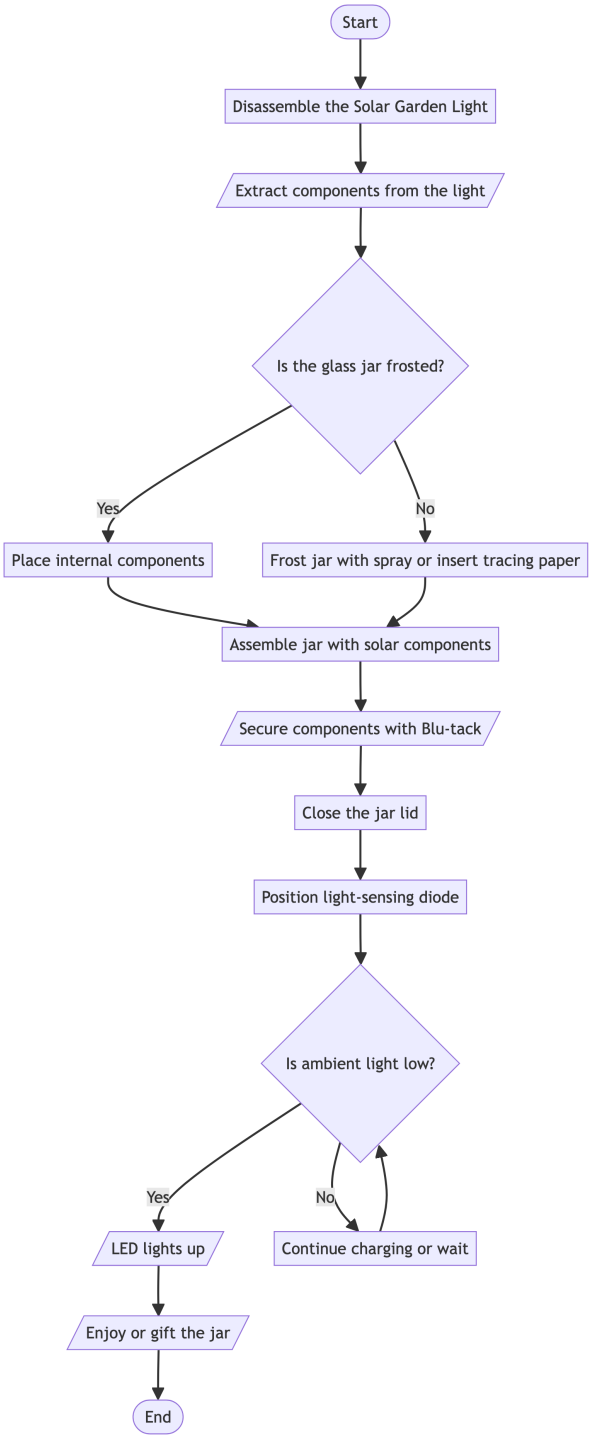
Qinghao Ye, Haiyang Xu, Guohai Xu, Jiabo Ye, Ming Yan, Yiyang Zhou, Junyang Wang, Anwen Hu, Pengcheng Shi, Yaya Shi, Chenliang Li, Yuanhong Xu, Hehong Chen, Junfeng Tian, Qian Qi, Ji Zhang, and Fei Huang. 2023b. mplug-owl: Modularization empowers large language models with multimodality. *CoRR*, abs/2304.14178.

Xiang Yue, Yuansheng Ni, Kai Zhang, Tianyu Zheng, Ruoqi Liu, Ge Zhang, Samuel Stevens, Dongfu Jiang, Weiming Ren, Yuxuan Sun, Cong Wei, Botao Yu, Ruibin Yuan, Renliang Sun, Ming Yin, Boyuan Zheng, Zhenzhu Yang, Yibo Liu, Wenhao Huang, Huan Sun, Yu Su, and Wenhu Chen. 2023. Mmmu: A massive multi-discipline multimodal understanding and reasoning benchmark for expert agi.

Rowan Zellers, Yonatan Bisk, Ali Farhadi, and Yejin Choi. 2019. From recognition to cognition: Visual commonsense reasoning. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6713–6724.

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

A Flowchart QA Example



T1: Fact Retrieval

Q: What should be done if the glass jar is not frosted?
A: Frost the jar with spray or insert tracing paper.

Q: What triggers the LED to light up in the solar jar?
A: Low ambient light causes the LED to light up.

T2: Applied Scenario

Q: Jason is disassembling a solar garden light for a DIY project but is unsure about how to safely extract the internal components including the solar panel, circuitry, LED, and battery housing. What tools should he use and how should he proceed with the disassembly?
A: Jason should use a utility knife and screwdriver to carefully disassemble the solar garden light and extract the necessary components.

Q: While attempting to create a homemade solar-powered LED lighted cookie jar, Michael realized he forgot to frost his Ikea glass jar. He doesn't have any frosting spray on hand but remembers he has some tracing paper. How should he proceed to achieve the necessary frosted effect?
A: Michael should cut a strip of tracing paper to fit inside the jar to achieve the frosted effect.

T3: Flow Referential

Q: Assuming the glass jar was already frosted, what are the next two steps I must take in sequence?
A: You would place the internal components and then assemble the jar with solar components.

Q: If I have just completed frosting the jar with spray or inserting tracing paper, what is the next immediate step in the process?
A: The next step is to assemble the jar with solar components.

T4: Topological

Q: How many nodes exist in the given flowchart?
A: 15

Q: Is the node "Is ambient light low?" direct predecessor of the node "Place internal components"?
A: No

B Dataset Distribution

Figure 4 illustrates how our data is distributed among various sources and types of questions.

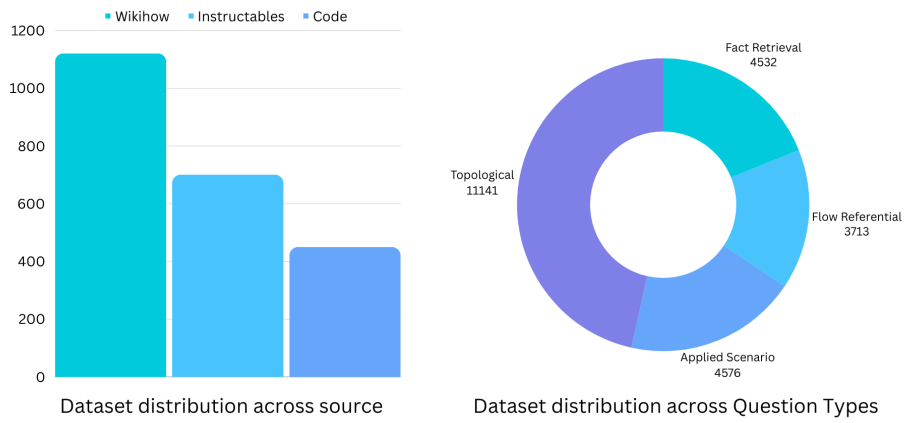


Figure 4: The figure shows the distribution of our data across different sources as well as across different types of questions.

C Additonal Results

Figure 5 shows the performance of FlowVQA dataset on various modelling strategies as outlined in Section 3. Table 9 shows VLMs across the three evaluator models - GPT, Llama and Mixtral over the various categories in the FlowVQA dataset. Figure 6 show category wise distribution of majority score for GPT-4V model. We also measure the average performance vs number of nodes in the flowcharts . The average is across all models and strategies and the graph is created after smoothening with an exponential weighted moving average ($\alpha = 0.4$), as shown in figure 7.

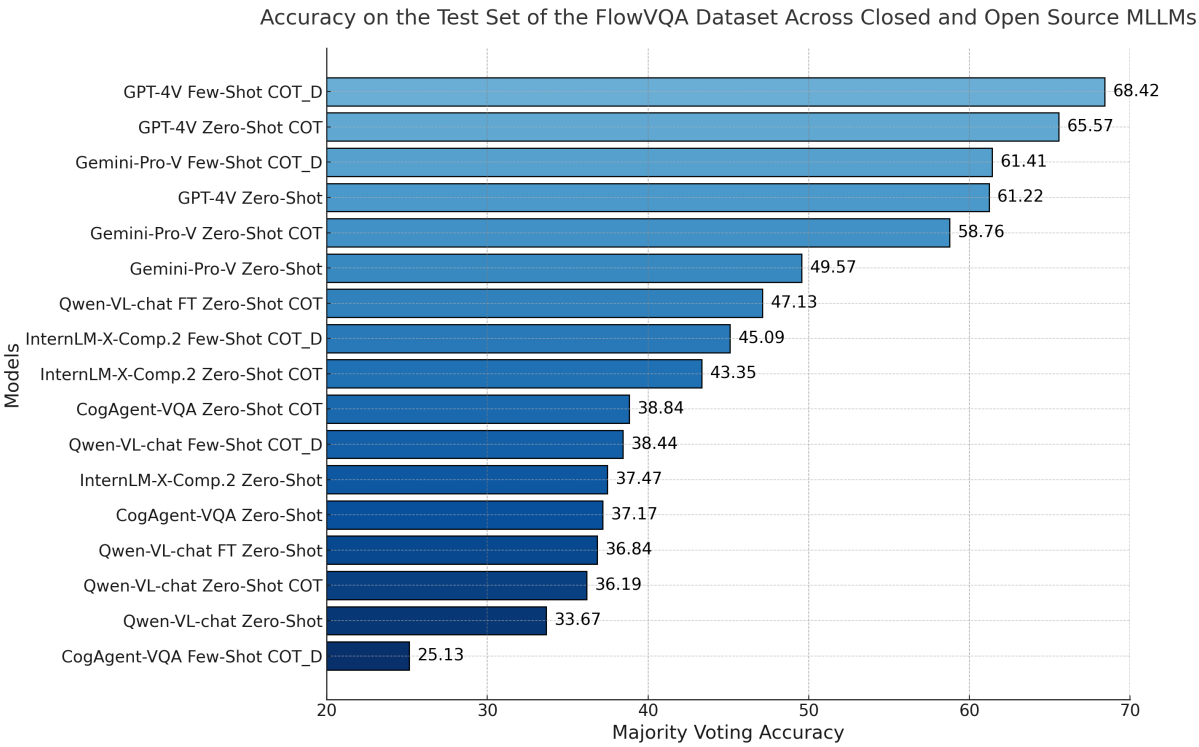


Figure 5: The horizontal bar chart shows the performance of FlowVQA dataset on various modelling strategies outlined in Section 3.

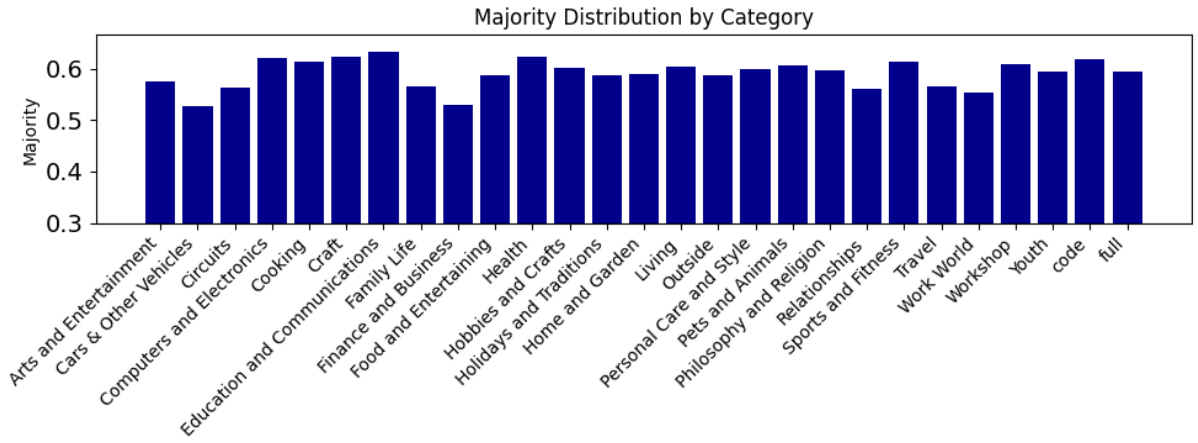


Figure 6: Category wise distribution of majority score for GPT-4V

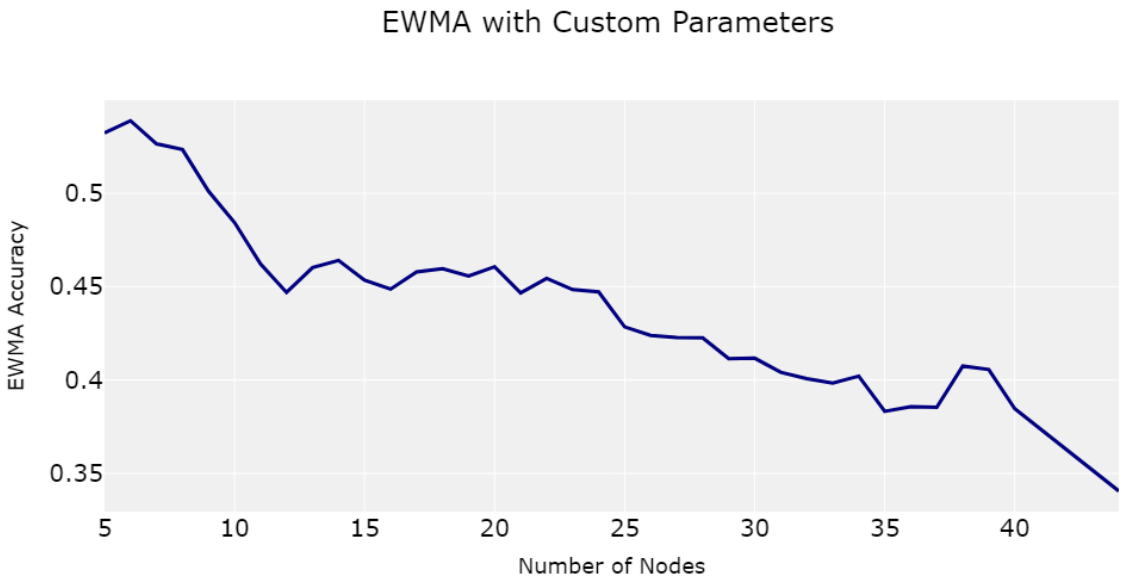


Figure 7: Average performance vs number of nodes. We measure the average across all models and strategies and the grpah is created after smoothening with an exponential weighted moving average ($\alpha = 0.4$)

Category	Majority Voting	GPT	LLAMA	Mixtral
Arts and Entertainment	57.4	57.4	58.2	59.3
Cars & Other Vehicles	52.8	54.3	53.6	53.4
Circuits	56.3	57.3	57.0	61.1
Computers and Electronics	62.1	61.8	61.4	63.6
Cooking	61.3	62.8	60.4	64.2
Craft	62.3	63.9	62.6	64.5
Education and Communications	63.4	64.4	60.2	64.4
Family Life	56.4	57.8	57.1	58.4
Finance and Business	53.1	54.6	53.4	53.8
Food and Entertaining	58.7	58.3	56.4	61.4
Health	62.4	64.8	60.4	62.8
Hobbies and Crafts	60.1	59.1	59.5	62.1
Holidays and Traditions	58.6	59.1	60.3	60.3
Home and Garden	59.0	59.8	57.0	60.5
Living	60.5	60.3	60.3	63.4
Outside	58.8	61.0	56.5	61.5
Personal Care and Style	59.9	59.9	60.3	62.9
Pets and Animals	60.7	62.1	60.3	64.7
Philosophy and Religion	59.6	58.2	58.7	60.9
Relationships	56.1	56.1	54.8	59.2
Sports and Fitness	61.3	62.9	60.2	60.9
Travel	56.6	57.0	55.3	58.3
Work World	55.3	54.3	53.2	56.7
Workshop	60.9	60.6	57.7	65.4
Youth	59.3	58.8	58.8	59.3
code	61.7	63.3	62.9	63.8

Table 9: Baselines across the three evaluator models—GPT, Llama and Mixtral over the various categories in the FlowVQA dataset.

D Prompts for Generation

In this section we lists the prompts we use to query GPT-4 in various steps outlined in Section 2

D.1 Flowchart Creation

First Step: Breaking down source text to structured summaries

Please provide a comprehensive structured summary, detailed step-by-step representation of the blog post below. Each step in the representation summary should be labeled with specific control codes that define its nature in the system. These codes include:
START: Marks the first step. There must be only one start step and the whole summary representation must follow a single step-by-step structure.
PROCESS: Indicates an ongoing process step.
DECISION [IF] [ELSE]: Denotes a conditional decision-making step, with outcomes being either 'Yes' or 'No'. For steps with multiple outcomes, break them down into smaller decision steps.
INPUT: Introduces new variables or elements, like ingredients in a recipe.
OUTPUT: Highlights the results, outputs or products of a step
END: Marks all terminal points where the process ends or cannot go any further.
! Treat the blog instructions as a system. The system has some inputs and some output. Describe the entire detailed summary in that particular format. Be it the working of an ATM machine or the steps to create pizza from raw ingredients everything can be looked at like a system or pseudocode. Make sure not to miss any critical points in processes.
! Try to retain context and structure it well.
! Important. Design the decision/conditional steps to have only 'Yes' or 'No' outcomes and treat their text like questions.
! Start from a single start point, do not have multiple parallel starts, make sure things remain step-wise with conditionals, loops etc.
Make the steps comprehensive and detailed, final output in markdown.

Second step: Converting structured summaries to Mermaid Scripts

Here is a detailed step-by-step summary tagged with detailed control codes for a blog post. Treat the step-wise summary as a system or a detailed pipeline. For this create a Mermaid Live Flowchart Script (flowchart TD) that is detailed, does not miss any key points, and captures all integral nodes perfectly. Treat the blog instructions and the flowchart as a system representation. Be it the working of an ATM machine or the steps to create pizza from raw ingredients everything can be looked at like a system.
Objective: Convert Passed Structured Summary to detailed Mermaid Live Flowchart (flowchart TD)
Control Codes for Assistance:
START: Marks the first step. There must be only one start step and the whole summary representation must follow a single step-by-step structure.
PROCESS: Indicates an ongoing procedure or action. Rectangle Shape.
DECISION [IF] [ELSE]: Denotes a conditional decision-making step, with outcomes being either 'Yes' or 'No'. For multiple outcomes, decompose into smaller decisions. Diamond Shape.
INPUT: Introduces new elements or variables, akin to ingredients in a recipe. Parallelogram Shape.
OUTPUT: Results, Outputs or end-products of a step. Parallelogram Shape.
END: All points of no further go terminal. Oval Shape.
Important Points
1. Treat the blog post instructions as a single system workflow or pipeline.
2. The system should include I/O, processes, decisions and terminals.
3. Ensure that the flowchart accurately depicts a real-life system flowchart, it should be contextually rich and practical for reference.
4. Maintain an optimal length for the flowchart not too long not too short, if there are multiple process steps in sequence you may consider combining them if the flowchart is too long.
5. Important! Design the decision steps to have only 'Yes' or 'No' outcomes. For steps with multiple outcomes, break them down into smaller decision steps.
6. Ensure a singular flow for the system, with all subroutines being direct components of the main system.
7. Ensure use of all flowchart symbols like rectangles, ovals, diamonds, circle, arrows etc.
8. Ensure the actual control codes are not mentioned in the flowchart nodes.
9. Verify flowchart syntax carefully

Sample of a small mermaid flowchart TD for reference:
flowchart TD
A(["Start"]) –> B["Process 1"]
B –> C"Decision?"
C –>|"Yes"| D["Process 2"]
D –> E["Process 3"]
E –> C
C –>|"No"| F[/"Output or Input"/]
F –> G(["End")
Make sure to verify each point above before your output.

D.2 Question Generation

Fact Retrieval

Task: You will analyze a step-by-step structured summary and Mermaid Flowchart Representation of a blog post or code script. The blog post includes specific steps for handling tasks.

Your Role: As a fact-extractor and question creator, your objective is to locate factual content within the summary. Your goal is to construct several question-answer pairs that each relate to distinct and critical facts presented in the summary.

Guidelines for Question Development:
1. Begin by determining the presence and quantity of direct facts in the summary. If there are multiple concrete facts, especially quantitative ones, generate questions for each. If fewer facts are present, create fewer questions. The ideal question range is 2-4 questions. 2-3 for fewer facts and 3-4 for ones with more facts.
2. Focus on specific and relevant facts, asking questions like Who? What? Why? How much? How many? Emphasize quantitative facts over qualitative ones.
3. Questions should be straightforward, with answers in the summary. Avoid direct references to the summary or the blog post in your questions.
4. Ensure each question highlights a different fact from the summary.

Answer Guidelines:
1. Provide brief and clear answers.
2. Answers must be definitive, avoiding open-endedness.
3. Offer several paraphrased answers for each question. (A1, A2, A3)

Output Format: Present your questions and answers in a structured JSON format, following the provided example.
Example Structure:
- Output JSON:
{
"1": {
"Q": "First Fact-based Question here",
"A1": "",
"A2": "",
"A3": "",
},
"2": {
"Q": "",
"A1": "",
"A2": "",
"A3": "",
},
}

Sample Question-Answer Pairs:
1. What is the correct temperature for preheating the oven?
A1. 80 Degrees Celsius
A2. Preheat the oven to 80 degrees Celsius
A3. ...

2. How long should crayons be left in the oven to melt?
A1. 20 Minutes
A2. Leave the crayons in the oven for about 20 minutes

3. What might tempt someone to peek?
A1. Gifts
A2. The temptation to peek at Christmas gifts

4. At what angle should the target be struck for full extension?
A1. A 90-degree Angle

5. How long should the cork be left to cure?
A1. Overnight
A2. Cure the cork overnight

PS: Your Answers should be BRIEF, definitive and must offer three paraphrased versions A1, A2, A3. Make sure the questions are not too open ended and concrete.

Also DO NOT MENTION THE BLOG/STRUCTURED SUMMARY/SCRIPT IN THE QUESTION.

Applied Scenario

Task: You will analyze a step-by-step structured summary and Mermaid Flowchart Representation of a blog post or code script. The blog post includes specific steps for handling tasks.

Your Role: As a complex situational question-answer generator, your task is to focus on the most interesting parts of the blog post's structured summary. Create 2-4 Complex Question-Answer Pairs. Each pair should correspond to a different, interesting area of the structured summary of the blog post.

Guidelines for Question Development:
- Focus on specific, relevant / crucial steps of the structured summary such as decisions, loops and other critical steps.
- Craft situational questions that are creative, practical, and likely to occur in real life.
- Ensure each question is directly related to a specific step mentioned in the blog post summary.
- Important: The question must be created in a way that the answer to the question can be directly obtained or inferred from the structured summary but no logical thinking should be done to further process the information in steps. The blog post should only be used to construct the context of the situation, not to generate the question itself.
- Important: Don't explicitly mention the structured summary or blog post in the question. Assume the person answering can reference it. Create long complex situations and questions.
- Provide suitable distractors in the question, complex stories, unique names, etc. Anything that makes the question more interesting, yet, answerable.
- Make sure all questions attend separate parts of the structured summary.

Answer Guidelines:
- Provide short, concise answers.
- Answers should be definitive and not open-ended.
- Offer several paraphrased answers for each question. (A1, A2, A3)

Output Format: Present your questions and answers in a structured JSON format, following the provided example.
Example Structure:
- Output JSON:
{ "1": {
"Q": "First Applied Scenario Based Question",
"A1": "Concise Answer 1",
"A2": "",
"A3": "",
},
"2": {
"Q": "",
"A1": "",
"A2": "",
"A3": "",
},
... More Q/A Pairs here
}

Sample Questions:
1. Ram, aged 45 years old, was going home from the office in his Minivan and his Minivan broke down on the way. He now wants to find a Minivan mechanic to get it repaired. He was trying to follow the given article, but being a little forgetful, he could not remember the age of his Minivan. He thought his warranty documents could help, Where should he try to find them?
2. Alice has decided to make custom fabric paint for a set of cotton t-shirts. She mixed equal parts of acrylic paint and a transparent gloss medium, but after testing on a swatch of cotton, the paint soaked through. What adjustment should she make to the paint mixture?
3. Selena has recurrent tonsil stones and her doctor has prescribed a course of antibiotics to address the issue. Unfortunately, the antibiotics weren't successful and Selena hasn't experienced any side effects or a relapse. What would her doctor's advice likely be at this stage?
4. Mark, an aspiring VFX artist, is enthusiastic about networking to enhance his opportunities in the field. He wants to join an industry group like the Visual Effects Society (VES). However, he is uncertain about the number of VES members and their global distribution. How can Mark find this information to ensure the group's relevance to his networking goals?

PS: Your Answers should be BRIEF, definitive and must offer three paraphrased versions A1, A2, A3. Make sure the questions are not too open ended and concrete.

Also DO NOT MENTION THE BLOG/STRUCTURED SUMMARY/SCRIPT IN THE QUESTION.

Flow Referential

Task: You will analyze a step-by-step structured summary and Mermaid Flowchart Representation of a blog post or code script. This post details specific steps to handle certain tasks.

Your Role: As a capable flowchart path and flow analyzer your task is to focus on critical sub-areas of the processes and flowchart and create path-based questions from that subflowchart.

Guidelines for Question Development:
- The first step is to decide on how many questions to create: If the flowchart is long and complex, break it down into smaller areas and create more questions (3). If the flowchart is short create fewer (2-3) but still good quality questions that would not be easy to answer directly. Focus on specific, relevant / crucial paths of the structured flowchart script and summary.
- Create questions based on node information looking FORWARD, BACKWARD, IN THE MIDDLE, etc. Questions about crucial decisions taken in a possible path.
- Craft questions about paths that are creative and hard but MUST HAVE A SINGLE DEFINITIVE TRUE ANSWER.
- Important: Don't explicitly mention the structured summary or flowchart in the question. Assume the person answering can reference it. Create long complex situations and questions.
- Create questions about backtracking, future paths, conditionals, nodes or steps in the middle, etc. Anything that is interesting in a flowchart path.
- IMPORTANT! It is very important that the current node/step or the node/path in question later is mentioned clearly. The rules for counting must be clearly mentioned.
Look at the sample questions below to create questions.

Answer Guidelines:
- Provide concise direct answers that are relevant to the question asked.
- Answers should be definitive.
- Offer several paraphrased answers for each question. (A1, A2, A3)

Output Format: Present your questions and answers in a structured JSON format, following the provided example.
Example Structure:
- Output JSON:
{
"1": {
"Q": "First Path Based Question",
"A1": "Concise Answer 1",
"A2": "",
"A3": ""
},
"2": {
"Q": "",
"A1": "",
"A2": "",
"A3": ""
},
}
Sample Questions:
1. What is the second step, given my zeroeth step is taking a negative decision at "Bostik Spritzkork 3070 Available?"?
2. If I currently have to fill the mold with plaster, what decision must have I taken a few steps back and what is the condition present at that node?
3. What is the minimum number of steps required to reach 'Final Inspection' from the "change job?" conditional?
4. Given the current zeroeth step is to close the top of the lid, what is the fifth step that I will be completing if I take the affirmative decision at any conditional present in between?
5. If at the current step the bathtub is not yet full and requires more water, what are the labels or descriptions of the fifth and seventh steps encountered when following the affirmative path from the current decision node?
6. How many steps are there from the initial "Start" node up to, but not including, the first decision point? In this count, the "Start" node is to be considered as the initial node or the 'zeroeth' step.
7. Alice is preparing for a rock-themed party and recalls Scarlet's unique style. She decides to start with a band T-shirt but is unsure whether to buy it online or at a concert. Given her limited budget, what should Alice's decision be based on?
8. If a patient's eligibility for tonsillectomy is currently being evaluated and they proceed with tonsillectomy following a positive recommendation, what would be the immediate next step, and what decision must have been made directly prior to this step?

Answers:
9. If I am currently at the 'Choose Show Audio Animation or press Control-A' step, what was the decision made at the first decision point, and what is the immediate next step?
A1: "The decision made was 'Yes' at the 'Decision to edit audio effects?' node, and the immediate next step is 'Audio effects editing mode activated'.
A2: "At the 'Decision to edit audio effects?' node, a positive decision was taken, leading to the next step of activating the audio effects editing mode.
A3: "The first decision point led to a 'Yes' outcome, and the following step is to activate the audio effects editing mode.

PS: Your Answers should be BRIEF, definitive and must offer three paraphrased versions A1, A2, A3. Make sure the questions are not too open ended and concrete.
Also DO NOT MENTION THE BLOG/STRUCTURED SUMMARY/ FLOWCHART SCRIPT IN THE QUESTION.

E Prompts for Question-Answering

In this section we lists the prompts we use to query models

E.1 Few-Shot-COT-with-Directives

Few-Shot COT_D

Examine the provided flowchart to answer the given question below. Here are some illustrative examples accompanied by a sequence of reasoning directives intended to stimulate analytical thought and elicit a rationale. These guidelines should facilitate the development of a rationale. Once a rationale has been formulated, proceed to present a conclusive final answer as in the examples.
Question - Answer pairs with Tags. The exemplar questions given depend upon the type of question we are asking
(Fact Retrieval/Applied Scenario/Flow Referential/Topological)
Example:

⋮
Q1. What temperature should the oven be preheated to for making the cake?
Tags: Temperature, Oven, Preheating, Cake Reasoning: Take it step-by-step. Look for a node or cluster of nodes, in the flowchart that mention preheating the oven. Identify node / nodes that mention a 'preheating'. After locating relevant nodes extract final answer if already present or reason further to deduce the correct answer.
A. 325 degrees Fahrenheit.
⋮

Concerned Question to Ask
Example:
What action is taken when the 'file' is found to be not empty?