

# **Towards Agentic Vision-Language Models for Question Answering on Interactive Dashboards**

**Aaryaman Kartha**

**A THESIS SUBMITTED TO THE FACULTY OF GRADUATE STUDIES IN  
PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF  
MASTER OF SCIENCE**

**Graduate Program in Electrical Engineering and Computer Science**

**York University  
Toronto, Ontario**

**December 2025**

**©Aaryaman Kartha, 2026**

## Abstract

Multimodal models, specifically Vision Language Models (VLMs), have shown increasing capabilities in data visualization oriented downstream tasks, achieving performance saturation in shorter intervals of time. Consequently, focus has shifted to assessing their potential towards new frontiers, specifically interactive environments. Various benchmarks center around data visualization question answering tasks on static visualizations, and such rudimentary approaches don't reflect real world analysis scenarios where vast decision making is required. Dashboards, while being commonplace tools in various industries, have had limited work done into evaluating the capabilities of VLMs to traverse and reason with them. To tackle these limitations, this thesis presents DashboardQA, a novel benchmark for interactive dashboard question answering. Overall, **292** tasks encompassing **405** QA pairs are presented from **5** diverse category types, with **112** carefully chosen dashboards represented. Experimental results show this benchmark is a challenge for various types of VLMs assessed, with the best model achieving 38.69 %.

## **Acknowledgments**

I extend my gratitude to many people who have helped me throughout my master's degree. I would like to first thank my supervisor, **Dr. Enamul Hoque Prince**, for his constant guidance and the wisdom he has shared throughout my masters degree, even with his hectic work schedule. He shaped me as a researcher and made me feel confident to progress in the highly competitive world of academia. **Dr. Shafiq Joty** has also provided valuable feedback and motivated my efforts, and I am extremely grateful for his help. I am grateful to all members of my research group for all their help.

I would like to give a special thanks to **Dr. Manos Papagelis**, for being a great teacher and for serving as a committee member, as well as **Dr. Zissis Poulos** for serving as the internal committee member.

I would like to thank my parents for all the love and support they gave to me throughout my life, and for instilling a passion for education onto me. I dedicate this thesis to them.

## Preface

The following work is based primarily on **DashboardQA: Benchmarking Multimodal Agents for question answering on Interactive Dashboards**, which has been accepted to EACL 2026. It is important to note that DashboardQA has built upon the previous work of **ChartQAPro: A More Diverse and Challenging Benchmark for Chart Question Answering**, which is why various parallels are seen in the dataset creation process.

A majority of the work was done **Aaryaman Kartha**, whose contributions include of idea and task formulation, dashboard curation, designing and setting up the entire human-VLM collaboration framework for DashboardQA, creating more than half of the questions in the dataset, dashboard metadata collection, helping with task configuration for the OSWorld setup, helping solve any possible issues with the generated results and refinement, and creating the goal-breakdown-workaround taxonomy as well as collecting those relevant results. **Ahmed Masry** helped greatly by running the experiments and setting up the OSWorld framework as well as collecting the evaluation results, seen in chapter 4 and sub chapters 5.1 to 5.3 respectively. My supervisor Professor **Enamul Hoque Prince**, as well as Professor **Shafiq Joty** provided assistance throughout this work and would constantly give insightful comments and feedback. **Shadikur Rahman** helped with the quantitative analysis seen in figure 5.6. The co-authors of the DashboardQA research paper contributed towards the dataset creation.

AI (GPT4o) was used for correcting any punctuation, grammatical, or structural errors in the writing of this thesis.

# Contents

	<b>Page</b>
Abstract . . . . .	ii
	<b>Page</b>
Acknowledgements . . . . .	iii
	<b>Page</b>
Preface . . . . .	iv
Contents . . . . .	v
<b>List of Tables</b>	<b>ix</b>
List of Figures . . . . .	xi
<b>1 Introduction</b>	<b>1</b>
1.1 Motivation . . . . .	1
1.2 Our Approach . . . . .	4
1.2.1 Research questions . . . . .	5
1.3 Thesis contributions . . . . .	6
1.4 Thesis organization . . . . .	7
<b>2 Literature Review</b>	<b>8</b>
2.1 Interactive Dashboard Features and Applications . . . . .	8

2.2	Vision Language Models . . . . .	10
2.3	Visual Question Answering . . . . .	11
2.3.1	Interactive Question Answering . . . . .	12
2.3.2	Pragmatic Oriented Visual Question Answering . . . . .	12
2.3.3	Evaluation Metrics . . . . .	13
2.4	Data Visualization Oriented Downstream Tasks . . . . .	15
2.4.1	Agentic Frameworks . . . . .	17
2.5	Agentic Environments . . . . .	18
2.6	Human VLM Benchmarking Collaboration . . . . .	19
2.7	Empirical Studies of Data Analysis Workflows . . . . .	20
<b>3</b>	<b>The DashboardQA Benchmark</b>	<b>22</b>
3.1	Dataset Creation . . . . .	22
3.1.1	<b>Phase 1 - Dashboard Curation</b> . . . . .	23
3.1.2	<b>Phase 2 - QA Pair Creation</b> . . . . .	23
3.1.3	<b>Phase 3 - QA Peer Review</b> . . . . .	26
3.2	Dataset Analysis . . . . .	31
3.2.1	Dashboard Analysis . . . . .	32
3.2.2	Question Answer Pair Complexity Analysis . . . . .	33
3.3	Summary . . . . .	36
<b>4</b>	<b>Methodology</b>	<b>37</b>
4.1	Task Defintion . . . . .	37
4.2	Task Setup . . . . .	37
4.3	Prompt Setups . . . . .	38
4.4	Action and Observation Spaces . . . . .	39

4.4.1	Action Space . . . . .	40
4.4.2	Observation Space . . . . .	41
4.5	Summary . . . . .	42
<b>5</b>	<b>Evaluation</b>	<b>43</b>
5.1	Baselines . . . . .	43
5.2	Evaluation Metric . . . . .	44
5.3	Main Results . . . . .	45
5.4	Qualitative Analysis . . . . .	46
5.4.1	Strengths . . . . .	46
5.4.2	Weaknesses . . . . .	47
5.4.3	Breakdowns and Solutions . . . . .	50
5.5	VLM Reasoning Behavior Quantitative Analysis . . . . .	52
5.6	Ablation Study . . . . .	58
5.7	Summary . . . . .	58
<b>6</b>	<b>Conclusion and Future Work</b>	<b>60</b>
6.1	Conclusion . . . . .	60
6.2	Limitations . . . . .	61
6.3	Future Work . . . . .	62
	<b>Bibliography</b>	<b>65</b>
	Appendices . . . . .	80
A	Prompts . . . . .	80
B	Sample Dashboards . . . . .	82
C	Sample Questions and Answer Workflows . . . . .	84
D	Sample Errors, Breakdowns, and Workaround Strategies . . . . .	86

E	Additional breakdown workaround mapping quantification . . . . .	88
F	Code for breakdown and associated workaround identification . . . . .	89

# List of Tables

2.1	Comparison of visualization characteristics across DVQA datasets. . . . .	17
2.2	Comparison of QA pair characteristics across DVQA datasets. . . . .	17
3.1	Counts of question creation methods through Human, VLM or mixed use in our collaboration framework. . . . .	25
3.2	Example prompt given for factoid QA pair generation . . . . .	26
3.3	Prevalence, count, and average number of respective visualization types across all dashboards, along with overall average of all visualizations. . . . .	34
3.4	Prevalence, count, and average number of respective navigation tools across the dashboards curated. . . . .	35
3.5	Distribution of views/states of dashboards required for the respective question categories. . . . .	35
3.6	Distribution of difficulty levels and corresponding number of steps across different question categories in DashboardQA. . . . .	36
4.1	Prompt Templates for Each Question Category. . . . .	39
4.2	GUI interaction and VM control actions . . . . .	40

5.1	Accuracy (%) on DashboardQA across different Observation Types (main headers) and Question Categories (sub-headers). Each Observation block includes five question types along with an Overall score. Color shading indicates model category: closed models (grey), hybrid models (blue), and fully open-source models (orange). The highest score within each category is shown in bold. . . . .	45
5.2	Goals, Breakdowns, and Workarounds seen in VLM-based interactive dashboard environments . . . . .	53
5.3	Ablation results for Gemini Pro 2.5 with an a11y tree and Jedi 7B W GPT4o on DashboardQA across 3 task difficulty levels: EASY, MEDIUM, and HARD. . . . .	58
1	Examples of Prompt Categories in DashboardQA for single turn question single dashboard questions . . . . .	80
2	Examples of Prompt Categories in DashboardQA involving multiple dashboards or multi-turn question answering . . . . .	81

# List of Figures

- 1.1 Illustrating the evolution of data visualization question answering tasks over time. These tasks have progressed from basic arithmetic and logical reasoning on single charts seen in subfigure (a) to multiple charts seen in (b). From there DashboardQA has shifted to interactive interfaces which entail multiple visualizations to reason from multiple views that must be traversed through keyboard/mouse actions, as shown in (c) . . . . . 2
  
- 3.1 DashboardQA creation process, from curating dashboards in tableau public to creation using a Human-VLM collaboration framework, and lastly reviewing. 22
- 3.2 DashboardQA focuses on 5 categories of questions that require exploration of dynamic dashboards. Questions are formulated so as to require dashboard exploration and reasoning between multiple views or visualizations. . . . . 24
- 3.3 Example hypothetical question featured in DashboardQA, depicting future forecasting. . . . . 27
- 3.4 DashboardQA QA pair generation and refinement process. Starting with a carefully crafted prompt given with dashboard UI screenshot decomposition with multiple views appended, generated qa pairs are filtered and then refined with a separate annotator. . . . . 28
- 3.5 Example conversational question in DashboardQA . . . . . 30

3.6	Example multiple dashboard question in DashboardQA . . . . .	31
3.7	Dashboard topic breakdown by author type. For the 8 dashboard topics present, the nested pie chart shows the entire composition of the topics (outer loop) to their respective author source type (inner loop). . . . .	32
4.1	An overview of the DashboardQA task execution pipeline within the OS-World environment, where (a) illustrates the overall setup of the VLM agent, and (b) shows an example interaction trajectory required to find an answer to a given question. The VLM, acting within an Ubuntu virtual environment, takes visual (Screenshot) or structural (a11y tree) input at each step and guides the navigation actions by generating corresponding <code>pyautogui</code> commands at every step. The action space of our agent consists of moving the mouse to a particular location, scrolling, and clicking. . . . .	40
4.2	Example dual observation space given in (ii). On the left a dashboard view screenshot is given, and to the right is the a11y tree representation of the dashboard UI elements. . . . .	41
5.1	DashboardQA Error Analysis: Illustrative cases of VLM failures, including plan-tracking errors, incorrect trend reasoning, and loss of intermediate values during interactive dashboard analysis. . . . .	47
5.2	Example incorrect reasoning trajectories as done by GPT4o. . . . .	49
5.3	Examples visualization tool usage failure and environment breakdown. The former shows mouse position recalibration for accurate cross filtering on a bar chart, and the latter waiting for the case of dashboard loading latency. . .	51
5.4	Prompt used to find VLM goals, breakdowns, and workarounds . . . . .	54
5.5	Showing multiple different breakdowns and workarounds for a sample task execution by Gemini Pro 2.5 . . . . .	55

5.6	Comparison of reasoning behaviors between Gemini-Pro-2.5 and Jedi-7B w/ GPT-4o on the DashboardQA benchmark. The plots illustrate (a, b) the dis- tribution of the number of interaction steps taken to answer questions and (c, d) the lengths of intermediate reasoning thoughts. . . . .	56
5.7	Heatmap representing breakdown-workaround pairs found for Gemini Pro 2.5 task executions . . . . .	57
1	Example default views of Tableau dashboards collected from various institu- tions, organizations, and individuals. . . . .	82
2	Sample dashboards chosen for all of the topics in DashboardQA . . . . .	83
3	Example MCQ question and task workflow in DashboardQA . . . . .	84
4	Example factoid question and task workflow in DashboardQA . . . . .	85
5	Example planning failure breakdown seen. Jedi is unable to utilize visual tab titles to directly go to required state, instead inefficiently sequentially moving along the tabs. Additionally, perceptual failure is seen when the VLM is unable to realize they have traversed through the right state. . . . .	86
6	Examples environment breakdown and visualization tool usage failure . . . . .	87
7	Heatmap representing breakdown-workaround pairs found for JEDI task ex- ecutions . . . . .	88

# Chapter 1

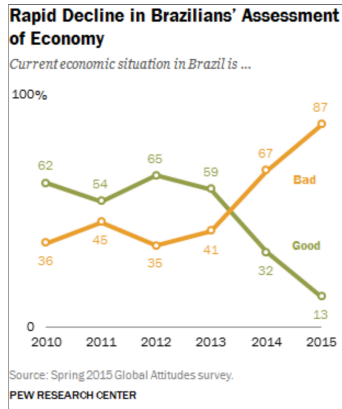
## Introduction

### 1.1 Motivation

Interactive dashboards are powerful tools that combine visualizations such as charts, maps, and tables with interactive features in a singular interface, allowing for a smooth analysis and obtainment of granular details [1]. Within various industries, dashboards are catalysts for decision making, serving as interfaces for streamlined insight extraction and extrapolation [2–4]. However, despite their ability to compress information well, interactive dashboards are still susceptible to information overload [5]. With their high volume of data, often compressed visuals, and navigational requirements between and within different states, reasoning over dashboards poses an inherently tedious and often demanding task [6], in particular with less technically proficient stakeholders. This challenge, along with the increasing capabilities of modern Vision Language Models (VLMs) to handle complex multi-modal question tasks, notably in question answering, has attracted interest in leveraging AI models to aid in analysis and ultimately decision making.

The fast paced improvements of modern VLMs have led to developments towards more complex question answering tasks on data visualizations [7, 8]. In order to gauge their true

Static chart reasoning

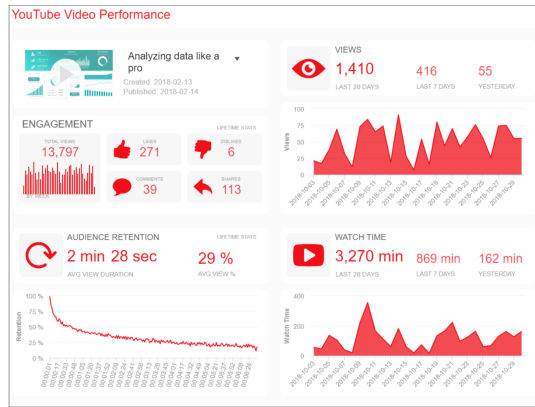


Q: Which year has the most divergent opinions about Brazil's economy?

A: 2015

(a)

Static Multi-chart reasoning

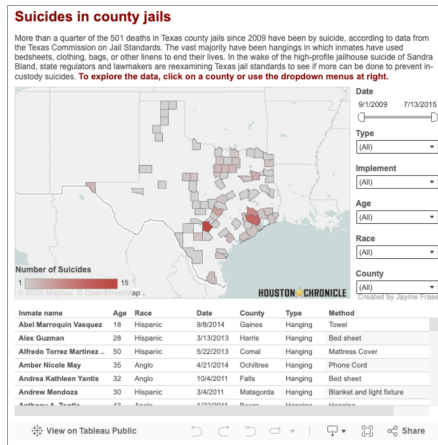


Q: Do the views overtime and watch time always follow the same pattern?

A: No

(b)

Dynamic dashboard reasoning



Q: Among the African-American, Anglo, and Hispanic inmates who died by suicide in Texas county jails, which group had the highest number of unique counties reporting 3 or more suicides?

A: Anglo

(c)

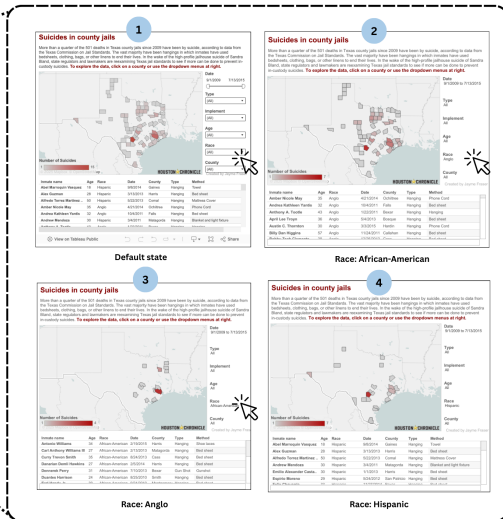


Figure 1.1: Illustrating the evolution of data visualization question answering tasks over time. These tasks have progressed from basic arithmetic and logical reasoning on single charts seen in subfigure (a) to multiple charts seen in (b). From there DashboardQA has shifted to interactive interfaces which entail multiple visualizations to reason from multiple views that must be traversed through keyboard/mouse actions, as shown in (c)

capabilities, they have been assessed through various downstream tasks, primarily question answering, on charts, infographics, and static dashboard snapshots [9]. What has started

with one-dimensional tasks over basic synthetic charts has progressed quickly into composite reasoning over multiple intricate, real-world visualizations. Nevertheless, previous work revolving around data visualization question answering has a common limitation: a focus solely around static visualizations. Such tasks are not practical as real world settings often engage with more dynamic visualizations that offer interaction and contain multiple possible views [10]. For these models, this involves not just complex arithmetical and logical reasoning between the views, but showing great GUI grounding abilities through executing carefully crafted sequences of GUI operations of inter-view and intra-view navigation operations. The former consists of operations to move to new states such as changing dropdown menu options or changing tabs, while the latter include of “Details on demand” activities to reveal hidden information in charts or tables such as hovering over them to extract information from a tool tip [11]. To effectively handle such complex tasks therefore requires careful planning, detailed visual reasoning, and self-reflection in their trajectory. The evolution of data visualization question answering over time is depicted in 1.1.

Navigation based interactive question answering has been the focus of multiple benchmarks, featuring tasks that require traversing through web and desktop applications [12]. Various environments have been created to streamline interactive benchmarking, notably OS-World [13], OfficeBench [14], and AndroidWorld [15]. The results obtained showed that even the best models achieved poor results in GUI based tasks. This has prompted the creation of new agentic approaches to navigate computer environments. For example, OpenAI have released their Computer Using Agent (CUA), a gpt-4o powered agent that can traverse the web. GUI oriented open source models such as UI - TARS [16] and JEDI [17] which are fine tuned, supplemental models to help navigate GUIs. As agentic VLM systems are showing increasing capabilities in complex tasks like navigating environments for question answering, it is fitting to assess VLMs true capabilities through more challenging practical tasks akin to

human reasoning. Nevertheless, this has not been looked at towards interactive dashboards, despite its common usage. Natural language question answering on dashboards can emulate real world data-driven meetings, where dashboards are used to drive decision making [18].

## 1.2 Our Approach

This thesis presents DashboardQA, an agentic VLM based, exploration grounded benchmark for advanced interactive dashboard question answering. This benchmark evaluates a range of open source, closed source, and hybrid model approaches over **292** tasks encompassing **405** question answer pairs encompassing **5 categories** of questions crafted through a custom human-vlm collaboration framework are represented: Multiple Choice Questions (MCQs), Factoid, Hypothetical, Multiple-Dashboard (Multi-Dashboard), and Conversational. Overall, **112** diverse, carefully curated interactive dashboards from tableau public are represented.

To evaluate such a benchmark, we utilize a custom framework on the OSWorld multi-modal agentic evaluation environment. Experimental results achieved show that this is a challenging task for all VLMs across all categories, with the best model approach of Gemini Pro 2.5 being aided with an accessibility tree achieving the best accuracy of 38.69%. Beyond accuracy, logged VLM reasoning steps are utilized in order to grasp the analyze the strengths and weaknesses of VLM agents in dashboard reasoning and navigation. Motivated by HCI literature, this thesis extends our examination of the model’s task workflows towards workaround approaches when facing breakdowns, helping to gauge how robust SOTA models are. This work ultimately serves to show the viability of VLM integration into data science workflows and create foundations to build upon creating effective agentic systems.

### 1.2.1 Research questions

The overall goal of this thesis is to understand general VLM capabilities in interactive dashboard reasoning tasks. In answering the main research question, we seek to also answer the following questions:

**RQ1: What types of reasoning questions do VLMs struggle or perform well in?**

5 discrete categories of questions are created to assess the diverse inferential abilities of various models. By obtaining the accuracy for each category for every model, we are able to shine light on what types of tasks each model may be suited towards or need refinement in.

**RQ2: What distinctions in performance can be made between open source, closed source, and hybrid models?**

A central theme of this paper revolves around analyzing the variation in results achieved between the proprietary closed source models against open source models, and of hybrid approaches. Open source models improve accessibility, increase transparency, and show promise in various applications [19]. Thus, it is relevant to examine their potential to be integrated into agentic workflows for dashboard analysis tasks, looking at both standalone and hybrid collaboration frameworks.

**RQ3: To what extent does supplemental information (a11y trees) help with navigational based dashboard question answering?**

We host VLMs in an online environment where the task executions occur in tableau public discover webpages featuring embedded dashboards. With this setting, a11y trees are able to be extracted and given as supplemental aids for VLMs to navigate. Thus, for evaluation each of the closed source proprietary models run 2 rounds of trials with the respective observation spaces: (1) Screenshots, (2) Screenshot + a11y trees. This helps evaluate VLMs abilities as standalone models and examine how much of a boost additional material may present.

**RQ4: What types of errors, breakdowns, and workarounds are commonly seen in**

### **traversing and reasoning across the dashboards?**

It is crucial to understand why VLMs typically achieve low results in interactive environments, and look at what type of errors they make. Additionally, breakdowns (disruptions of a trajectory) occur and require workaround solutions [20] that require self-reflection in order to resolve. It is important to note the various approaches different VLMs take and the impact on their performance.

## **1.3 Thesis contributions**

This thesis comprises of four core components:

1. We present a novel Question Answering dataset, DashboardQA, to assess VLM agents on their ability to navigate and reason with interactive dashboards. We provide 5 categories of questions to enable a holistic assessment of reasoning capabilities in situations with different inferential requirements.
2. An evaluation of various open, closed, and hybrid models on DashboardQA through a streamlined custom setup utilizing OSWorld and the tableau public website.
3. A thorough examination of the results achieved and key findings of performances by the different VLMs in the different categories, including of a qualitative analysis and quantitative analysis featuring insights into strengths and weaknesses of the models
4. We further examine logged VLM reasoning steps among SOTA models to identify and quantify reasoning strategies taken when dealing with breakdowns.

## **1.4 Thesis organization**

The remainder of this thesis is organized as follows: Chapter 2 outlines relevant literature that gave inspiration and guided the development of this work. Chapter 3 details on the dataset construction process and gives a multi-dimensional analysis of the diversity of the dashboards curated and QA pairs. Chapter 4 clarifies the task at hand and explains the process created for evaluation, including of the action and observation spaces. Chapter 5 presents an overview of the results achieved and gives a qualitative and quantitative analysis of the VLM task executions. Chapter 6 concludes the thesis with an overall summarization of the results observed, examining limitations of this work, and proposing future research directions.

# Chapter 2

## Literature Review

Our literature review begins by outlining interactive dashboard features and applications, and highlighting their importance. From there VLMs and their improvements over time, as well as towards interactive applications, are looked at. The next three sections look holistically at the how question answering benchmarks have branched to different focuses over time and become more complex, and outline the changes in evaluation methods. The next two sections look at the development of agentic environments and agentic applications in data visualization downstream tasks. Human-VLM collaboration efforts are looked at next, and finally HCI literature are examined for data analysis workflows.

### 2.1 Interactive Dashboard Features and Applications

Interactive dashboards are commonplace tools for data analysis that integrate multiple visual and interactive features and tools into one compact and usable interface [21]. Interactive dashboards have both navigation tools and built-in interactive features for visualization interaction. The primary set of navigation tools consists of dropdown menus (Multiple option and single option), range-sliders, tabs, and radio buttons, which all help users move to dif-

ferent dashboard states. Various visualization types are represented in dashboards, such as bar charts, line charts, heatmaps, and cross tabs (tables), and often dashboards have multiple visualizations in a single view. Interactive features are implemented to allow users to drill down on data visualizations; users can find specific data point values through tool tips and utilize space well with cross filtering and cross highlighting. Cross filtering in a view with multiple visualizations allows a user to filter surrounding visualizations to have data on the same topic [22]. Similarly, cross highlighting allows a user to highlight a certain data value in a visualization and related data in other visualizations by dimming the other data points [22].

Dashboard usage extends to various fields, notably in time-critical, data-heavy areas such as healthcare and finance, among many others [4, 23, 24]. In healthcare dashboard applications can be seen on individual and broader scales. The former involves usage to effectively gauge a patients data and thus treat the patient, while the latter focuses on evaluating larger population metrics such as infection or obesity rates for example, ultimately guiding public health policy [23]. Within the finance sector, dashboards serve to highlight market trends, study customer behavior, and overall help reduce risk management [4]. In both the time sensitive industries mentioned, it is seen that with built-in mechanisms for real time data updates, dashboards become more efficient alternatives for decision making than static reports [25]. Beyond these fields, various governments, organizations, and institutions have utilized dashboards for demographic analysis, social justice, and as educational tools [26, 27].

DashboardQA presents dashboards that reflect the interactive diversity that would be present in real world dashboard analysis scenarios. Additionally, in order to assess the prospect of VLM agents within appropriate domains, dashboards presented are carefully curated to help reflect their usage in various fields, most notably in the aforementioned fields of

healthcare, finance/business, and in shaping public policy.

## 2.2 Vision Language Models

Vision language models (VLMs) merge techniques from Computer vision (CV) and Natural Language Processing (NLP) to create a multi-modal system that can comprehend both text and image inputs [28]. Rapid progress has been made in enhancing the capabilities of these models in handling complex tasks. Notably, novel proprietary models such as GPT4o [29] and Gemini Pro 2.5 [30] have shown State of the art (SOTA) performance in many tasks, prompting further research on their viability as 'agents' - entities mimicking human activity to be integrated in various work environments within multiple domains [31]. To gauge their capabilities in various fields, VLMs are assessed on a range of carefully crafted benchmarks. In order to exploit the reasoning power of models, and thus obtain results that best present their true capabilities, various prompting approaches have been proposed. Chain-of-thought prompting serves to elicit reasoning through generating intermediate steps, breaking down complex task requirements [32].

With newer models showing ever increasing performance saturation on static visual benchmarks, a new focus on gauging the interaction abilities of multi-modal models in interactive environments. In order to supplement the capabilities of VLMs in these environments, various GUI oriented models have been introduced to enable integration into high-end multi-agentic framework pipelines, such as UI-Tars [16] and Jedi [17]. These models are built on top of the QWEN VL model series [33], and fine-tuned, respectively, on custom made datasets. High-end and more standalone capable models for open ended, interactive reasoning tasks have also been created, notably through the utilization of a reinforcement learning framework. GUI-Owl [34] follows a similar construction process to UI-Tars and Jedi, but also uses reinforcement learning to enhance abilities in downstream tasks. Additionally, OpenAI

released a computer using agent (CUA) model, enhancing the power of GPT4o’s abilities to traverse environments with advanced reinforcement learning [35].

Despite all the progress made, even State of the art (SOTA) VLMs have shown poor performance in open-ended tasks simulating real world computer environments [13]. In spite of agentic VLM frameworks increasingly being integrated in many phases of the data science pipeline [36], their effectiveness of reasoning and navigating with dashboards is not well understood. This motivates the need to evaluate, and further analyze the responses, of both standalone models, closed and open source, as well as hybrid models on the tasks created in DashboardQA.

## **2.3 Visual Question Answering**

Visual Question Answering (VQA) broadly refers to the task of generating responses to inputs of images accompanied with a natural language question [37]. Various benchmarks have been developed to assess multi-modal models in reasoning between both textual and visual modalities. Over time, many of these benchmarks have evolved to simulate tasks more akin to real world human tasks, incorporating more linguistic principles, as well as focusing on interactive environments.

VQA benchmarks have branched to many different focuses over time. Agrawal et al. [38] introduced one of the earliest large scale VQA benchmarks, featuring open ended MCQs. Open ended visual question answering has also been done by Ging et al.[39], with a focus on a granular evaluation of the images. Another focus has been on multi hop reasoning which looks at multiple levels or pieces of information being reasoned through together [40]. Tran et al. [41] introduced ReasonVQA, a dataset focused on multi-hop visual question answering. This entails inputs of images of real world objects, coupled with a natural language question, that involve combining multiple pieces of external information given to answer.

### **2.3.1 Interactive Question Answering**

Recently, shifting interest has come into interactive question answering, whereby given continuous screenshots of dynamic GUI states models must navigate and reason through computer environments. Such tasks require more cognitive efforts - necessitating additional planning and forethought, GUI grounding and self-reflection. An example is seen in AppWorld [42], where a subset of the benchmark presents question answering tasks that require traversing commonly used applications such as gmail or spotify.

DashboardQA extends interactive question answering to tableau dashboards hosted on the tableau public website [43]. In our dataset and evaluation analysis, various navigational properties are quantified to show interactive necessity.

### **2.3.2 Pragmatic Oriented Visual Question Answering**

A core facet of multi-turn Natural Language Processing (NLP) tasks is pragmatics, which looks at how context affects the meaning of language [44]. The primary means for demonstrating core pragmatics principle is through conversational question answering, which requires utilizing contextual information from previous queries and responses in order to answer follow up questions [45]. To do as such, models must leverage pragmatic principles to handle any ambiguity seen [45]. This presents a task more in tune with natural human dialogue.

Prior literature has focused on evaluating the abilities of LLMs to understand pragmatic principles. Benchmarks focusing on fine grained analysis on a singular principle has been done for implicature [46] and presupposition [47]. Sravanthi et al. combine many ideas to [48] provide a benchmark featuring MCQ tasks on 4 pragmatics principles: Implicature, Presupposition, Reference, and Deixis. These approaches have been purely text based, but slowly have been integrated into visual question answering tasks. Specifically, with the grow-

ing popularity of data visualization comprehension tasks, various pragmatics understanding tasks related to their comprehension have been proposed. Before the deep learning era, approaches towards pragmatics oriented question answering were more rule and logic based [49]. This evolution has extended towards data visualizations comprehension tasks. For example, Hoque et al. [50] present Envizeon, a system built for multi-turn interactions, or conversations, with data visualization and has a focus on pragmatics. Support for handling such tasks are embedded into the system to help locate anaphoric, deictic, and visual property references, starting from an initial utterance. Following this more deep learning oriented benchmarks came. ChartQA Pro [8] features a conversational question category in its questions revolving around chart, infographics, and static dashboards.

The interest in visualization question answering and pragmatics comprehension, but lack of attention to interactive visualization, has prompted this effort in DashboardQA towards evaluating conversational question answering on tableau dashboards.

### **2.3.3 Evaluation Metrics**

Evaluation of generated answers in VQA benchmarks have evolved with different metrics over time, with both human and automatic evaluation approaches being utilized. Changes in evaluation methods are the result of benchmarks containing more categories of questions that require different means of assessment, and also due to stricter rules towards higher end models. Additionally, differences in evaluation exist depending on the whether the answer is fixed or more open ended. For example, a custom human evaluation approach have been proposed for more open-ended and subjective VQA by Agrawal et al. [38], where 10 workers are asked to answer the question at hand and compared with the model's response through  $\min(n/3,1)$ , where  $n$  represents the number of agreements with different annotators. However, an inherent flaw is that questions with minimal agreement, specifically below 3, limit

the accuracy score that can be achieved [37].

Large scale benchmarks pose inherent issues when relying on human annotations, posing a time consuming and costly challenge, and consequently prompting the need for more automated evaluation frameworks [51]. Strict exact match assessment is used particularly for the cases of MCQs or true/false questions [8, 52, 53]. Less strict approaches include BLEU [54] and ROGUE [55], and METEOR [56], which all compare generated responses with ground truths through different variants of n-gram (Sequences of n words) comparisons. In order to account for textual information within images while not heavily penalizing minor reasoning mistakes in Optical Character Recognition use tasks, Biten et al [57] introduce the Average Normalized Levenshtein Similarity, which uses the Normalized Levenshtein metric comparing two strings [58] to give a score between 0.5 and 1. In equation (2.1),  $N$  represents the number of questions being evaluated on, and the ground truth string  $a_{ij}$  is being compared to the predicted answer string  $o_{ij}$ . The function  $s$  shown in (2.2) calculates the Normalized Levenshtein, and a threshold  $\tau$  set to be 0.5 to be strict in cases where over half of the string’s characters are incorrect [57].

$$\text{ANLS} = \frac{1}{N} \sum_{i=0}^N \left( \max_j s(a_{ij}, o_{q_i}) \right) \tag{2.1}$$

$$s(a_{ij}, o_{q_i}) = \begin{cases} 1 - NL(a_{ij}, o_{q_i}), & \text{if } NL(a_{ij}, o_{q_i}) < \tau, \\ 0, & \text{if } NL(a_{ij}, o_{q_i}) \geq \tau. \end{cases} \tag{2.2}$$

DashboardQA was built in order to reflect the growing question answering landscape and present a more complex, interactive task for VLMs. Furthermore, given the diverse categories of questions contained with the dataset, a multifaceted evaluation framework reflecting

many ideas of the diverse approaches summarized above is utilized.

## 2.4 Data Visualization Oriented Downstream Tasks

The AI landscape has seen rapid improvements in VLM agentic capabilities within data visualization comprehension tasks. In particular, the task of Data Visualization Question Answering (DVQA) has grown in popularity [10]. What started with value extrapolation and comparisons from simple charts has evolved toward arithmetic and logical reasoning on multi-chart visualizations for more advanced comprehension [59, 60]. Various datasets surround Data Visualization comprehension have been created as such. Tables 2.1 and 2.2 present, in chronological order, prominent datasets created for data visualization question answering tasks and analyzes them to show respective refinements made over time. This evaluation is split by a image/visualization analysis in table 2.1 and QA pair analysis in table 2.2 respectively.

Before ChartQA primarily template based questions were in focus in various datasets [59, 61–64], with a lack of complex categories of questions (i.e hypothetical, unanswerable etc) that usually would entail composite reasoning involving multiple steps/techniques to answer. Additionally, the visualizations either had synthetic data or were made synthetically. The main changes seen were additional question templates, clearer categorization of questions, and even a move from synthetic data to real world data, but still synthetic visualizations remained.

The new era of visualization comprehension was pioneered with ChartXiv [65], ChartQA [7] and its extension ChartQAPro [8]. They focused on shifting away from template based questions, and incorporating human made or machine generated question answer pairs on both real world data and visualizations. ChartQA created both machine generated and human made questions. They utilized a T5 (Text to Text Transfer Transformer) model [66]

for question and answer generation and trained the model on SQuAD’s (passage,answer) to create the question dataset [67]. Shortly after ChartQA was released the advent of GPT and various other MLLMs showed great capabilities in creating a wide variety of questions for chart comprehension. In turn, Wang et al. introduced ChartXiv [65] to increase the complexity of chart understanding tasks for MLLMs, and found flaws in SOTA MLLMs systems. They create a stress test and find that MLLMs performance deteriorates when dealing with perturbations, or “small visual or textual changes in charts”. With the eventual performance saturation Masry et al. [8] introduce ChartQAPro, featuring 5 diverse categories of questions. The visualization scope was also extended to feature dashboards and infographics, and some of the visualizations presented were accompanied with supplemental text to create a more multi-modal challenge for the VLMs to work with. Additionally, the QA pairs generated were either through humans or utilizing three popular VLMs (GPT4o, Claude Sonnet 3.7, or Gemini 2.5 pro). Specifically, VLMs were prompted such as to elicit question answer pairs from the respective categories of questions, which then were either refined/corrected or kept as given. Multi-ChartQA by Zhu et al. [60] was created slightly before ChartQAPro, but nevertheless focused on the issue that only singular charts were used before in existing benchmarks, aiming to then put in multi-hop reasoning that is the type of complexity seen in the real world. Beyond Question Answering, ChartInstruct [9] presents a broad range of instruction following downstream tasks, from creating basic summaries and future forecasting to coding abilities.

The overarching attribute to all of the datasets created were that the visualizations used were static images. This is not how visualizations are utilized with in the real world where they are more interactive in nature [10]. This in turn would prompt the desire for the creation of DashboardQA which features only dynamic visualizations with questions requiring exploration.

Name	Real/ Synthetic data	Real/ Synthetic visualizations	# of visualizations	Static/ Dynamic	Basic charts	Infographics	Dashboards
FigureQA[61]	Synthetic	Synthetic	100k	Static	✓	✗	✗
DVQA[59]	Synthetic	Synthetic	300k	Static	✓	✗	✗
LEAF-QA[62]	Real	Synthetic	240k	Static	✓	✗	✗
LEAFQA++ [64]	Real	Synthetic	244k	Static	✓	✗	✗
PlotQA [? ]	Real	Synthetic	224k	Static	✓	✗	✗
ChartQA [7]	Real	Real	20882	Static	✓	✗	✗
ChartXiv[65]	Real	Real	2323	Static	✓	✗	✗
ChartQAPRO [8]	Real	Real	1341	Static	✓	✓	✓
Multi-chartQA [60]	Real	Real	2000	Static	✓	✗	✗
DashboardQA	Real	Real	112	Dynamic	✓	✗	✓

Table 2.1: Comparison of visualization characteristics across DVQA datasets.

Name	# of questions	Template/Human/VLM based	Factoid	MCQ	Conversational	Hypothetical	Multi-visualization	Unanswerable
FigureQA[61]	1 million	Template	✓	✗	✗	✗	✗	✗
DVQA[59]	3,487,194	Template	✓	✗	✗	✗	✗	✗
LEAF-QA[62]	2 million	Template	✓	✗	✗	✗	✗	✗
LEAFQA++[64]	2.6 million	Template	✓	✗	✗	✗	✗	✗
PlotQA[? ]	28.9 million	Template	✓	✗	✗	✗	✗	✗
ChartQA	32.7k	Template/Human based	✓	✗	✗	✗	✗	✗
ChartXiv[65]	5000	Human	✓	✗	✗	✓	✓	✓
ChartQAPRO[8]	1948	Human/VLM based	✓	✓	✓	✓	✓	✓
Multi-chartQA[60]	1370	Human	✓	✓	✗	✗	✓	✗
DashboardQA	500	Human/VLM based	✓	✓	✓	✓	✓	✓

Table 2.2: Comparison of QA pair characteristics across DVQA datasets.

## 2.4.1 Agentic Frameworks

The advent of multi-agentic reasoning frameworks has shown in various fields that they have a great potential to autonomously solve complex tasks with little to no human intervention [68]. Through task delegation, collaboration efforts, and reflective strategies, multi-agentic systems are able to grasp flexibility to undertake tasks in new environments [69]. Multi-agentic frameworks have been extended towards data visualization comprehension tasks. To tackle the issue of hallucination within chart question answering, Goswami et al. [70] create a multi-agentic framework called ChartCitor to supplement chart question answering with fine grained bounding box annotations on pertinent chart elements. Another paper by Suri et al. [71] looks at visual attribution through set-of-marks prompting [72] as a means to avoid hallucinations and increase faith in the VLMs response.

Additionally, work on multi-agentic frameworks for data visualization downstream tasks serve as inspiration for testing the capabilities of hybrid models on dashboard question-

answering, utilizing proprietary models for reasoning and GUI-oriented models for navigation. This reflects a planner navigator framework, which will in turn be evaluated against standalone model approaches.

## 2.5 Agentic Environments

Multiple benchmarking environments and respective benchmarks have been created for the assessment of multi-modal models in a variety of interactive tasks mimicking real world computer use.

Wang et al. [14] introduce OfficeBench, featuring various large scale tasks requiring traversing and working on multiple work applications such as Microsoft word, excel, and email. This designed to mimic typical larger scale tasks with multiple applications being used that take place in the office environment. Over time, we see larger scale environments to encompass more OS related features. Xie et al. introduced OSWorld, a specialized environment that allows for evaluation of multi-modal agents on computer tasks [13]. This includes of tasks like creating scripts, editing documents, and file management, among others. Overall, 369 are featured, and they look at both single and multi-app workflows that feature various popular applications and require either Graphical User Interfaces (GUI) or Command Line Interface (CLI) usage. However, the agentic capabilities on computer tasks done through the OSWorld environment were reported to have very low scores, with a majority of models hovering well below 30 percent success rate. Noticing a lack of agentic work done for mobile applications even with its commonplace usage, Rawles et al. presented AndroidWorld [15] for testing autonomous agents in an Android OS emulator environment. In order to reflect the dynamic nature of real world tasks, AndroidWorld randomly instantiates the task parameters in order to extend the number of tasks it can work on.

In this work we utilize the agentic oriented OSWorld environment for streamlined and

scalable testing on virtual machines, and also create our own novel framework for doing so. Specifically, we utilize agentic VLM frameworks as a means to traverse the dashboards within the tableau public website.

## **2.6 Human VLM Benchmarking Collaboration**

Benchmark creation can be tedious, time-consuming, and costly. In the case of large scale dataset, these are seemingly inevitable. To alleviate these problems, many benchmarks have utilized VLM assisted approaches that involve model generation and human refinement. K-viscuit [73] introduces a Korean cultural understanding benchmark created through a human-VLM collaboration whereby models, given guidelines and few shot learning, output is corrected by native Korean speakers [73]. Additionally, ChartQA Pro [8] presents usage of a human VLM collaboration framework to generate 5 categories of questions revolving around standalone image representations of charts, dashboards, and infographics. A common attribute between the aforementioned benchmarks is a focus on static question answering.

Nevertheless, there has been research done to utilize VLMs in automated GUI trajectory creation, focusing on both formulaic and exploration based approaches. Noticing the scarcity of GUI training data, AgentTrek [74] creates a scalable framework utilizing automated collection of web tutorials coupled with human-assessed VLM evaluators to create structured GUI trajectories. Explorer [75] focuses on a more exploration-based automated trajectory creation process, with a clearer usage of a multi-agentic framework including of task delegation and collaboration. From an initial snapshot to generate a starting task trajectory and first action, a refiner agent iteratively tunes the trajectory based on new views given. An accompanying task description is given by a summarizer agent after analyzing action history and screenshots. To evaluate alignment between the task description and the trajectory given, a task verifier agent is featured [75]. Ultimately, they utilized the power of

multi-agentic frameworks to help automate their benchmark creation.

In tune with the evolving use of Human VLM collaboration strategies in various domains, this thesis builds on ideas presented in the aforementioned approaches, focusing on utilizing an expert guided human VLM collaboration framework for dashboard exploration oriented QA pair generation.

## **2.7 Empirical Studies of Data Analysis Workflows**

Human-computer interaction (HCI) research has often focused on analyzing data science workflow practices of professional and non-professional data workers in various domains, helping create empirical foundations to refine systems [20]. Notably, studies have examined strategies data workers have taken when dealing with general obstacles and broad solution spaces during their tasks. For example, Boukhelifa et al. [76] use interviews to gauge how various domain experts handle uncertainty in their data analysis process, categorizing and quantifying strategies employed for different goals they set. Similarly, Liu et al. [77] examine how data workers from various domains choose among various "alternatives" - a broadly defined solution space encompassing tools, sources of information, algorithms, hypotheses etc. The results obtained from semi-structured interviews ultimately helps create a clear multi-dimensional characterization of the respective roles of the alternatives [77]. However, these examinations as a whole have seldom taken dashboard usage into account, even with their widespread usage in modern data workflows. To fill this gap, Tory et al. [20] through an interview study examined how dashboard users make sense of their dashboards and handle any difficulties, or "breakdowns", encountered when completing any of their assigned tasks. Breakdowns were classified into various high and low-level tasks and quantified to illustrate how often dashboard users encountered the problem.

This thesis extends the analysis of dashboard sense making and hardship management

in dashboard reasoning capabilities to modern vision language models. Such an extension is deemed fitting given the shift from humans to VLM agents in the automation revolution of data analysis workflows [24]. Additionally, given the 'Jack of all trades, masters of none' capabilities of AI systems seen in data analysis tasks [78], many parallels can be seen with the study conducted by Tory et al. [20] whereby non-professionals data workers are examined on their dashboard sensemaking processes.

# Chapter 3

## The DashboardQA Benchmark

This chapter outline how the DashboardQA dataset was made, looking at the dashboard curation and QA pair creation process for DashboardQA. Additionally, a holistic analysis of our multi-modal dataset is presented through a topical, navigational, and visual diversity analysis of the dashboards collected.

### 3.1 Dataset Creation

DashboardQA’s construction framework encompasses three phases : (1) Interactive Dashboard curation, (2) QA pair generation, and (3) QA peer review.

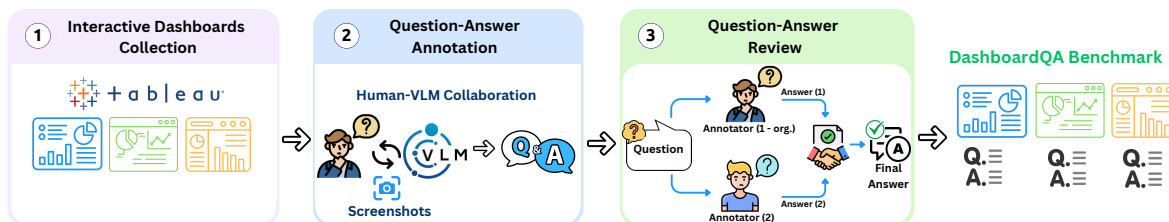


Figure 3.1: DashboardQA creation process, from curating dashboards in tableau public to creation using a Human-VLM collaboration framework, and lastly reviewing.

### **3.1.1 Phase 1 - Dashboard Curation**

All of the dashboards were sourced from Tableau Public - a free to use cloud-based platform for creating and sharing tableau dashboards [43]. Tableau public was chosen because of the vast range of easily accessible dashboards available. Additionally, there are various qualities of the dashboards that make it appealing for interactive based evaluation; they often support rich interactivity features such as coordinated views, cross-filtering, and cross-highlighting. Beyond that, a diverse set of navigation tools and chart types are commonly featured.

We extracted real world, interactive dashboards based on various pertinent social, economic, and political issues. Multiple pre-defined topics of interest were put into search terms and then filtered in terms of relevancy or view count. On top of topic diversity, both the visual and navigational diversity was also examined. Noticing similar types of authors tend to use recurring layout patterns, we looked at various different individuals, organizations, and corporations in our search. Notable dashboard sources include of U.S Census Bureau [27], United Nations [79], and USDA[80].

### **3.1.2 Phase 2 - QA Pair Creation**

DashboardQA features 5 categories of questions: (i) Multiple Choice Questions (MCQs), (ii) factoid, (iii) hypothetical, (iv) conversational, (v) multi-dashboard. Group members, consisting of researchers well versed in the fields of HCI, NLP, and data analytics and visualization, cooperated to create the dataset through a human-VLM collaboration framework. Measures taken to avoid bias by humans and VLMs involved utilizing 3 different VLMs to create 3 different methods, representing different combinations of human-VLM interactions, of QA creation. We specifically used state of the art models available - GPT-4o, Gemini Pro 2.5, and Claude Sonnet. 3 different approaches were used to create the QA pairs, and are detailed as follows:

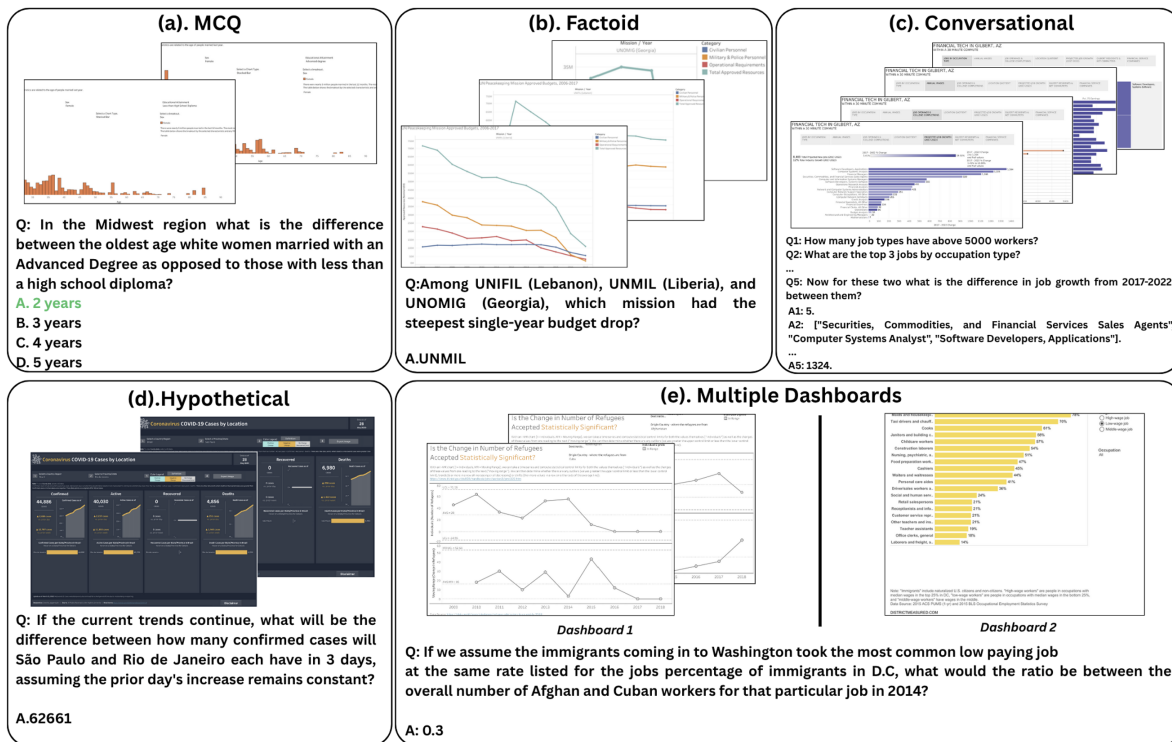


Figure 3.2: DashboardQA focuses on 5 categories of questions that require exploration of dynamic dashboards. Questions are formulated so as to require dashboard exploration and reasoning between multiple views or visualizations.

- **Curating seed QA pairs:** After scrutinizing the dashboard at hand, the author created their own meaningful QA pair without any VLM assistance. DashboardQA was set up so that at least 25 percent of all examples here human created. Aside from adding diversity and mitigating VLM bias, the human-made examples offer more authentic means of evaluating human thinking [81].
- **VLM generated:** Purely VLM generated questions are those that were taken directly as given by the VLM, and were rare occurrences. Obtaining an ideal question answer pair without any visual or semantic errors was a challenge for all of the VLMs.
- **Human refinement:** When unsatisfactory output given by the VLM requires refinement, modifications and corrections are made by the author. This includes of correcting

errors in the answers generated, along with fixing punctual and grammatical errors in the questions. VLM assisted questions created represent the vast majority of qa pairs.

Table 3.1: Counts of question creation methods through Human, VLM or mixed use in our collaboration framework.

Question Category	Human-made	VLM Assisted			VLM-Only		
		GPT4o	Claude	Gemini	GPT4o	Claude	Gemini
MCQ	39	15	10	7	5	1	3
Factoid	34	11	0	6	3	0	13
Hypothetical	28	19	8	6	1	0	0
Multi-Dashboard	18	8	8	10	2	0	0
Conversational	9	7	6	6	0	0	0

For the questions that involved VLM use, contributors were provided with carefully crafted textual prompts, seen in A, tailored to each respective question category, and directed to change the examples at the end of each prompt to help diversify the QA pair created. Additionally, they were instructed to decompose the dashboard(s) into screenshots to add for visual context. These screenshots include of the dashboards default view, additional views in different states, and snippets of revealed dropdown menu options upon expanding. Notably, if there are multiple tabs in the dashboard, then more views are to be given as they have a tendency to have different structural layouts. The views are given to help the VLM understand semantic layouts commonalities, whereas the dropdown menu options are given to help show possible trajectory options to create multi-view questions. As was instructed in the prompts, 5 examples were to be generated and the annotator was to choose the best to either keep or refine. In the case of poor generated QA pairs, authors were allowed to follow up with the VLM in a conversational approach to get better output. To streamline VLM evaluation and inter-annotater agreements, all answers were required to be concise, either true/ false, yes/no, a numerical value, or a label taken from the dashboard itself. List options were also allowed.

As seen in table 3.1, VLM assisted questions as a whole represent the largest share of

questions. GPT4o was the most chosen model, and the reason for variations between the models was due to inference limitations with free trials as well as time restrictions.

Table 3.2: Example prompt given for factoid QA pair generation

Category	Prompt
Factoid	<p>I am giving you multiple snapshots/views of an interactive dashboards, including of the respective default views, options available for navigation through the dashboard’s navigation tools (i.e Dropdown menus, radio buttons, scrollers, etc.), and more views from when particular options are selected. Since not every possible snapshot / view of each dashboard is given, the selection options available and the example views for each dashboard should together help you guess how the views for the rest of the options would be structurally (i.e types and content of the charts or textual data available) for that particular dashboard.</p> <p>Using all of these, please generate 5 diverse and challenging factoid reasoning questions that involve arithmetic and logical reasoning. They must have multiple operators (i.e sum, ratio, etc.) in the question. Additionally, make sure the answers required are succinct, where they are either a numerical value, True or False, Yes or No, unanswerable or simply a label taken from the dashboard itself. Each question must require 3 to 5 views (Cumulative) of the dashboards to answer. Just the questions are to be given now from your end, in the next prompt the required views will be given for you to answer.</p> <p>Here are some sample questions to give some inspiration, but make sure to create the questions with lexical and semantic differences.</p>

### 3.1.3 Phase 3 - QA Peer Review

In order to assure quality and accuracy in our question answer pairs, all QA pairs were peer-assessed by a separate co-author working on a different question category. This also served to

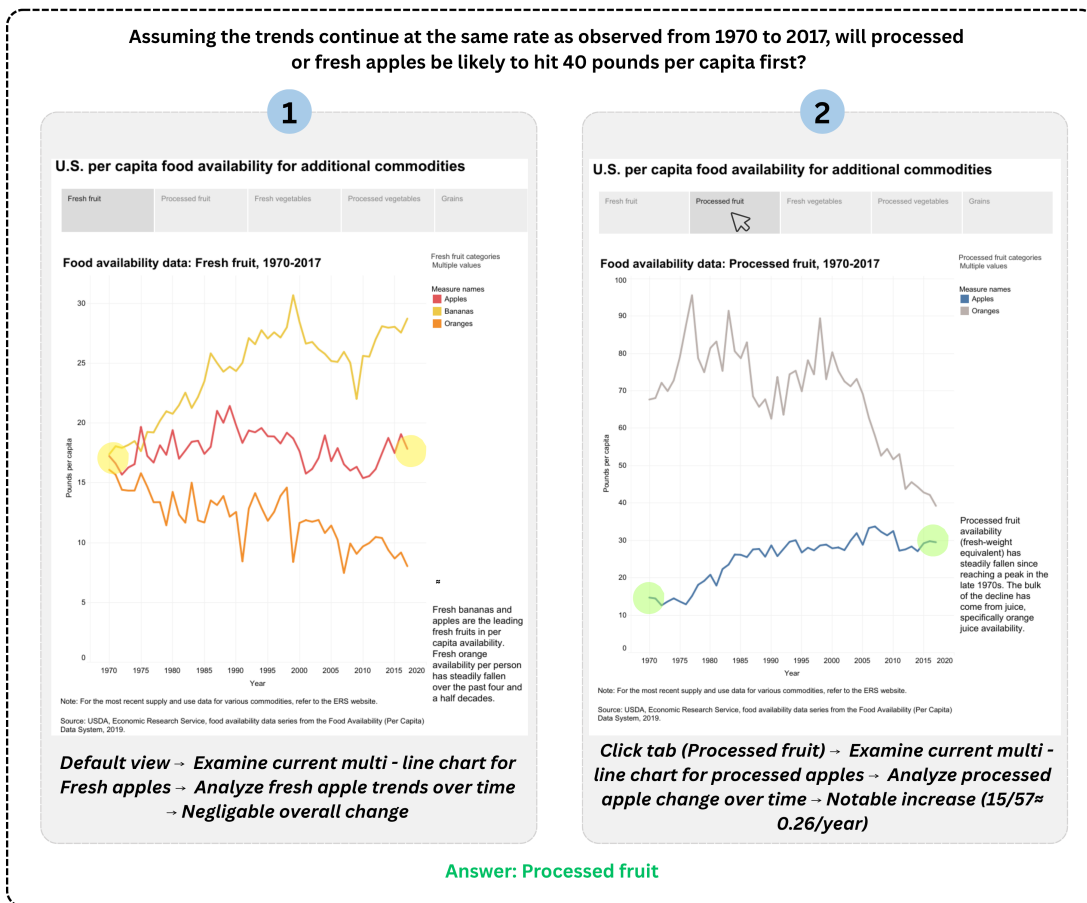


Figure 3.3: Example hypothetical question featured in DashboardQA, depicting future forecasting.

highlight any unnecessary complexity that a question may have had. In the case of questions that require estimations, a percentage difference of less than 0.5 would be deemed acceptable. Overall, an inter-annotator agreement rate of 74.93%, with the remaining 25.07% needing further refinement. While a lot of DashboardQA is based on ChartQAPro, there is a clear focus on agentic reasoning on dynamic visualizations as opposed to static. Every question is created such that there must be multiple views traversed and reasoned over to obtain an answer. The following question categories are covered:

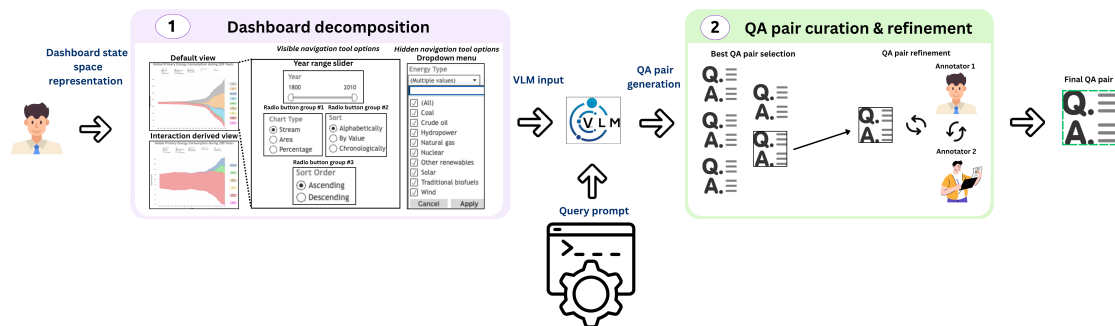


Figure 3.4: DashboardQA QA pair generation and refinement process. Starting with a carefully crafted prompt given with dashboard UI screenshot decomposition with multiple views appended, generated qa pairs are filtered and then refined with a separate annotator.

**MCQ:** When given 4 or 5 candidates, the VLM must reason with information from multiple views and elements of the dashboard to find which answer is the best fit. Many of these questions feature cases where the options are separated by small margins, thus requiring precise information extraction and accurate computation. An example is given in 3.

**Factoid:** These focus on information retrieval, pattern recognition, and extrapolation to derive answers on questions that involve visual perception and logical reasoning towards the dashboard in question. Emphasis is placed to make sure each question involves multiple arithmetic and logical operations. A standard example is seen in 4

**Hypothetical:** Hypothetical questions requires reasoning beyond what is seen in the dashboard itself. These consist of counterfactual (i.e looking at what might happen in different scenarios) and future prediction related questions involving trend analysis. Figure 3.4 shows an example future forecasting question.

**Conversational:** Conversational question answering assesses the ability of VLMs to show linguistic proficiency through recognizing contextual dependencies and handling coreference resolution. Essentially, they must show the ability to understand references to ideas in the previous question to resolve later question ambiguities. Additionally, they must utilize previous questions answers (intermediate calculations) in follow-up questions. An example conversational task and answer workflows is given in 3.5. In Dashboard QA, each conversation contains anywhere from 3 to 7 factoid style questions, and only the final question in the conversation is given to be answered to account for VLM inference costs.

**Multi-Dashboard:** Multi-Dashboard question answering involves reasoning between multiple dashboards. This focuses on an inherently more challenging task that involves both inter-dashboard and intra-dashboard analysis, requiring an understanding of the visual and semantic diversity between dashboards along with a higher degree of planning and self-reflection. The action space increases as we not factor in moving back and forth between the dashboards. This, in turn, leads to a propensity of erroneous results. Given the high inference costs and substantial reasoning requirements of working with many dashboards, each multi-dashboard question is confined to two dashboards. Figure 3.6 shows an example question and answer workflow.

Additionally, each of the single-turn categories of questions have a unique sub-category of unanswerable questions. These require reasoning with information not available in the dashboard, and are a good way to assess a models robustness by evaluating their capabilities in handling adversarial tasks [82]. Examples of QA pairs from each category are shown in Figures 2 and 3. Figure 2 represents single dashboard single turn questions, whereas in figure 3 the categories that either require multi-turn or multiple dashboards are shown, representing

**Q1: How many centuries are featured in the timeline?**

**1.1**

Global Primary Energy Consumption during 200 Years

Drag the mouse pointer to the top right corner - Click and drag the year range slider to identify years and centuries - Initial value is 1800 to 2010 - Century 1 = 1800

**1.2**

Global Primary Energy Consumption during 200 Years

Click and drag the year range slider value from 1800 to 1900 - Century 2 = 1900

**1.3**

Global Primary Energy Consumption during 200 Years

Click and drag the year range slider value from 1900 to 2000 - Century 3 = 2000 - Final Answer is 3

**Answer: 3.**

**Q2: Comparing only between Coal, Natural gas, and Crude oil, which century shows the most one sided energy consumption distribution?**

**2.1**

Global Primary Energy Consumption during 200 Years

Select 'Coal' on the right side of the dashboard and hover on the 'Coal' button - It's 19<sup>th</sup> Century for 'Coal'

**2.2**

Global Primary Energy Consumption during 200 Years

Now select 'Crude Oil' on the right side of the dashboard and hover on the 'Crude Oil' button - It's 19<sup>th</sup> Century for 'Crude Oil'

**2.3**

Global Primary Energy Consumption during 200 Years

Finally, select 'Natural Gas' on the right side of the dashboard and hover on the 'Natural Gas' button - It's 19<sup>th</sup> Century for 'Natural Gas' - Final Answer is 19<sup>th</sup> Century

**Answer: 19th century.**

**Q3: Which century shows the most uniform distribution then?**

**3.1**

Global Primary Energy Consumption during 200 Years

Click and drag the year range slider to identify the first century (19<sup>th</sup>) starting from 1801 - 1900 and check the distribution

**3.2**

Global Primary Energy Consumption during 200 Years

Click and drag the year range slider to identify the second century (20<sup>th</sup>) starting from 1901 - 2000 and check the distribution

**3.3**

Global Primary Energy Consumption during 200 Years

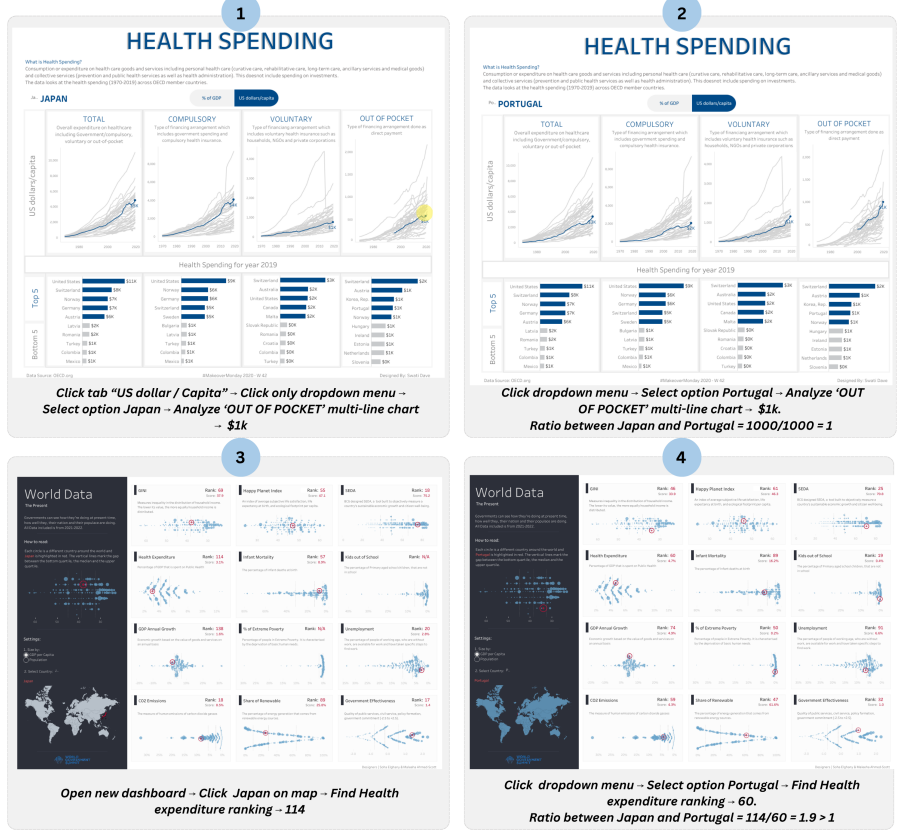
Click and drag the year range slider to identify the last century (21<sup>st</sup>) starting from 2001 - 2010 - The 21<sup>st</sup> Century shows the most uniform distribution, so the answer is 21<sup>st</sup> Century.

**Answer: 21st century.**

Figure 3.5: Example conversational question in DashboardQA

what we believe is a more challenging problem.

Compare the ratio of the most recent per capita Out-of-pocket health spending between Japan and Portugal versus the ratio of their Health Expenditure Ranking. Which is larger?



Answer: Health Expenditure Ranking

Figure 3.6: Example multiple dashboard question in DashboardQA

## 3.2 Dataset Analysis

In analyzing our dataset we examine the dashboards for their topical, visual, and navigational/ interactivity diversity. Additionally, we look at the complexity of the answer requirements for the QA pairs created.

### 3.2.1 Dashboard Analysis

**Topic distribution:** In order to see patterns in topic choices between the types of authors, we examine their topic distributions. Figure 3.7 shows categorized distributions of the 9 topics seen in curated dashboards, with respective spreads displayed across for organizations, institutions, and individual authors. Individual and organization created dashboards encompass all the topics seen, whereas institutions encompassed half. This can be due to the restrictions and specializations that institutions have. Additionally, institutions showed the most skewed distribution with a particular leaning towards politics, comprising 54% of the dashboards. Organizations and individual authors showed the most uniform distribution of dashboard topics, with individuals marginally more balanced. Organizational dashboards focused mostly on economics with a 37% share, while individuals focused the most on health related topics with a 25% share.

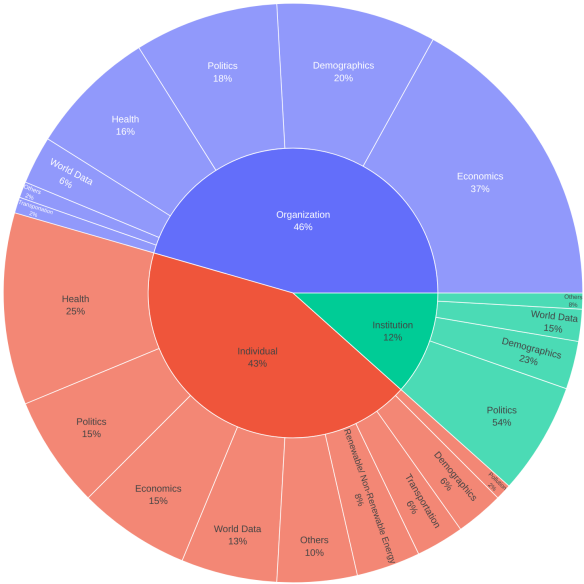


Figure 3.7: Dashboard topic breakdown by author type. For the 8 dashboard topics present, the nested pie chart shows the entire composition of the topics (outer loop) to their respective author source type (inner loop).

**Visualization components analysis:** Overall, 13 visualization types were seen across the dashboards collected, consisting of cross tabs maps, and 11 chart sub-types. As depicted in table 3.3, basic charts such as line and bar charts were the most prevalent, with more complex charts such as heat maps, bubble charts, and bump charts being less common. In tune with the instructions given to the authors, the average number of navigation tool components per dashboard was around 2.66, showing the emphasis placed on having multiple exploration means in the dashboards selected.

**Navigation tools analysis:** Our analysis included of a prevalence count seeing how many respective dashboards included the tool, and of an average per dashboard which sums the total number of the navigation tool groups seen and divides by the total number of dashboards (112). As seen from table 3.4, dropdown menus were seen in almost every dashboard with a 92.31% prevalence rate, and had multiple of them with 2.38 on average. Tabs come in second with 23.08%. Radio Buttons and Range Sliders hold an equal presences of 7.69%. However, radio button groups were seen more often when indeed present, having a higher average per dashboard.

### 3.2.2 Question Answer Pair Complexity Analysis

To gauge the difficulty and level of effort required to answer the questions, we examine the minimum number of required states to traverse to answer, as well as the number of steps required.

**State navigation requirements:** In DashboardQA, a state represents the current configuration of selected navigation tool options of the dashboard. Often, state and view might be used interchangeably, they don't always represent the same thing. A view refers to a snap-

Table 3.3: Prevalence, count, and average number of respective visualization types across all dashboards, along with overall average of all visualizations.

<b>Visualization</b>	<b>Prevalence</b>	<b>Count</b>	<b>Avg. per dashboard</b>
Line Chart	48.51%	86	0.77
Bar Chart	42.57%	79	0.71
Map	20.79%	25	0.22
Area Chart	16.83%	31	0.28
Scatter Plot	10.89%	30	0.27
Crosstab	10.89%	12	0.11
Pie Chart	6.93%	9	0.08
Bubble Chart	5.94%	7	0.06
Heat Map	5.94%	7	0.06
Tree Map	4.95%	7	0.06
Donut Chart	1.98%	2	0.02
Histogram	1.98%	2	0.02
Bump Chart	0.99%	1	0.01
<b>Total</b>	<b>100%</b>	<b>298</b>	<b>2.66</b>

shot at the current time, so when data visualizations are interacted with to reveal tooltips for example, the view changes but the dashboards state doesn't. Table 3.5 displays the distribution of dashboard states required to traverse and reason over to complete all of the tasks. The most common number of views was 2, with 7 or more views being the least. Notably, the more complex question types (Hypothetical, Multi-Dashboard, and Conversational) contained higher shares of questions requiring 6 or more states to traverse. This is the case as these tasks were intentionally framed to be more challenging with more multi-view reasoning, as opposed to MCQs and Factoids which had more of a focus on simple exploration abilities. Among all the categories, factoids had the most uniform distribution for the number of views required to answer their respective questions. The most skewed distribution was for Multi-Dashboard with 2 to 4 states representing close to 84.78 % of the distribution.

Table 3.4: Prevalence, count, and average number of respective navigation tools across the dashboards curated.

<b>Navigation Tool dashboard</b>	<b>Prevalence</b>	<b>Count</b>	<b>Avg. per</b>
Dropdown Menus	92.31%	266	2.38
Tabs	23.08%	103	0.92
Radio Buttons	7.69%	26	0.23
Range Sliders	7.69%	9	0.08

**Step requirements:** In the context of DashboardQA, a step is defined as a round of mouse or keyboard actions that will either change the configuration of a navigation tool or drill down on a particular data visualization to reveal specific data values. From table 3.6 it is once again clear that the easier categories have their question distribution skewed towards 'Medium' and 'Hard' questions whereas conversational and Multi-Dashboard questions are skewed towards 'Hard'. Overall, 'Medium' level questions represent the largest share of questions, and 'Hard' the least.

Table 3.5: Distribution of views/states of dashboards required for the respective question categories.

<b>Category Type</b>	<b>1</b>	<b>2</b>	<b>3</b>	<b>4</b>	<b>5</b>	<b>6</b>	<b>7+</b>
MCQ	0	41	25	7	3	4	0
Factoid	5	14	21	21	9	2	2
Hypothetical	4	30	19	7	2	2	1
Multi-Dashboard	0	9	17	13	0	5	2
Conversational	0	4	7	8	4	2	2

Table 3.6: Distribution of difficulty levels and corresponding number of steps across different question categories in DashboardQA.

Question Category	Easy		Medium			Hard			Total
	1	2	3	4	5	6	7	8+	
MCQ	14	24	15	11	6	7	3	0	80
Factoid	10	21	19	12	3	6	1	2	74
Hypothetical	18	13	12	8	6	4	3	1	65
Multi-Dashboard	3	2	7	8	9	8	1	8	46
Conversational	1	6	6	4	4	2	1	4	27
<b>Total</b>	<b>46</b>	<b>66</b>	<b>59</b>	<b>43</b>	<b>28</b>	<b>27</b>	<b>9</b>	<b>15</b>	<b>292</b>

### 3.3 Summary

In this chapter we outlined how the DashboardQA benchmark was created through a human-VLM collaboration framework. Furthermore, for the sake of transparency on the dataset’s diversity, we present a fine grained multi-dimensional breakdown of the dashboards and QA pairs created.

# Chapter 4

## Methodology

This chapter focuses on the task setup used to evaluate the VLM agents for the tasks in DashboardQA. Starting with a formal definition, we then describe the overall custom framework setups and detail on the action and observation spaces.

### 4.1 Task Definition

Every task in DashboardQA is framed as exploration and required dashboard navigation. The entire dataset can formally be defined as  $D = (d_i, q_i, a_i)_{i=1}^N$ , where each triple comprises of an interactive dashboard  $d_i$ , associated question  $q_i$ , and ground truth answer  $a_i$ . Overall, there are 292 tasks representing 405 questions, where all categories are single turn except for conversational which contain anywhere from 3 to 7 questions.

### 4.2 Task Setup

To help streamline our evaluation for our interactive setting, we utilized a custom framework with OSWorld’s environment and the tableau public. For each task, the entire evaluation

process begins through launching an Ubuntu OS virtual machine and opening up the question’s associated tableau public URL to show the default view of the embedded interactive dashboard. This is done in a google chrome browser, where the page is set to full screen at 1920 x 1080. The task for the VLM agent is initiated when given a triple of an initial observation of the web view captured from the virtual environment, instruction prompt, and the dashboard’s corresponding question. The agent then processes this information to plan accordingly and give an initial reasoning thought accompanied with a proposed action execution code sequence. Following the environment’s execution of the given code sequence, another observation is taken and given to the VLM to reason over. This back and forth continues until either the model signals task completion or acknowledges failure, or if the environment sees the process going over the predefined step limit. The final step involves parsing the answer from the completed log files, where the structured format answer given by the VLM is extracted and assessed. Figure 4.1 displays the architecture used and provides an example demonstration of task workflow executions.

### **4.3 Prompt Setups**

For each question category a specific prompt is given to instruct them on the type of task they have and the rules to be followed. This is shown in table 4.1. All prompts were setup in order to elicit succinct answers within brackets so as to make for easy parsing and automatic evaluation, and Chain of Thought (COT) prompting is used in order to see the thought process the VLMs take and to enable further qualitative and quantitative analysis on the logged steps.

Table 4.1: Prompt Templates for Each Question Category.

Category	Example Prompt
<b>Factoid</b>	You are given a factoid question about an interactive Tableau dashboard that you need to navigate to answer the question. You need to think step-by-step, but your final answer should be a single word, number, or phrase. Do not generate units. But if numerical units such as million, m, billion, B, or K are required, use the exact notation shown in the dashboard. Remember to navigate the dashboard, think step-by-step, and put the final answer between these brackets <answer></answer>
<b>MCQ</b>	Question: <question> You are given a question about an interactive Tableau dashboard along with different possible answers. You need to navigate the dashboard to select the correct answer from them. You need to think step-by-step, but your final answer should be one of the options letters only (without any additional text). Remember to navigate the dashboard, think step-by-step, and put the final answer between these brackets <answer></answer>.
<b>Hypothetical</b>	Question: <question> You are given a hypothetical question about an interactive Tableau dashboard that you need to navigate to answer the question. You need to think step-by-step, but your final answer should be a single word, number, or phrase. Do not generate units. But if numerical units such as million, m, billion, B, or K are required, use the exact notation shown in the dashboard. Remember to navigate the dashboard, think step-by-step, and put the final answer between these brackets <answer></answer>
<b>Multi-dashboard</b>	Question: <question> You are given a question about two interactive Tableau dashboards that are open in two tabs in the browser. You need to navigate them to answer the question. You need to think step-by-step, but your final answer should be a single word, number, or phrase. Do not generate units. But if numerical units such as million, m, billion, B, or K are required, use the exact notation shown in the dashboard. Remember to navigate the dashboard, think step-by-step, and put the final answer between these brackets <answer></answer>
<b>Conversational</b>	Question: <question> You are given a multi-turn conversation, and your job is to answer the final question based on the conversation history and the information in the provided interactive Tableau dashboard that you need to navigate. You need to think step-by-step, but your final answer should be a single word, number, or phrase. Do not generate units. But if numerical units such as million, m, billion, B, or K are required, use the exact notation shown in the dashboard. Remember to navigate the dashboard, think step-by-step, and put the final answer between these brackets <answer></answer>  Question: <question with conversation history>

## 4.4 Action and Observation Spaces

A clear set of definitions of the action and observation spaces is given as follows:

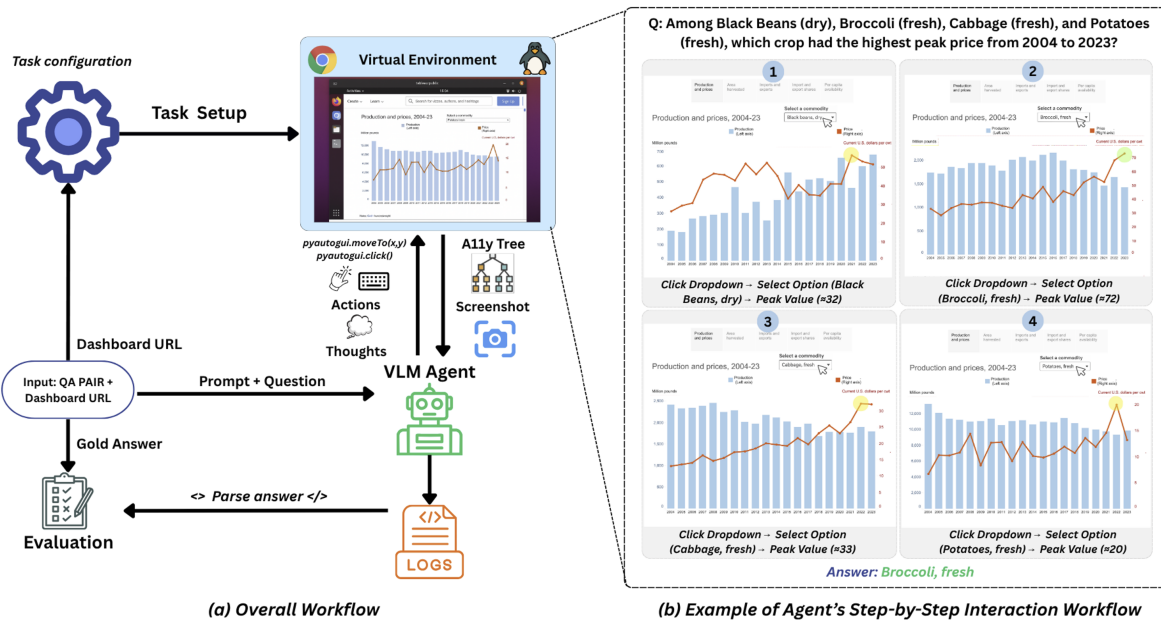


Figure 4.1: An overview of the DashboardQA task execution pipeline within the OSWorld environment, where (a) illustrates the overall setup of the VLM agent, and (b) shows an example interaction trajectory required to find an answer to a given question. The VLM, acting within an Ubuntu virtual environment, takes visual (Screenshot) or structural (a11y tree) input at each step and guides the navigation actions by generating corresponding `pyautogui` commands at every step. The action space of our agent consists of moving the mouse to a particular location, scrolling, and clicking.

### 4.4.1 Action Space

The action space, in accordance with OSWorld’s framework, encompasses of fundamental mouse movements and keyboard interactions executed through either `pyautogui`, and virtual machine control done by special meta actions. Termination of the task execution occurs in three scenarios: (1) The model has found the answer, (2) The agent acknowledges task failure, (3) The agent has reached the predefined step limit of 25.

Category	Actions
Mouse and keyboard actions	Move position, Click, Drag, Scroll
VM control actions	Wait, Fail, Done

Table 4.2: GUI interaction and VM control actions

## 4.4.2 Observation Space

For the evaluation each VLM undergoes 2 rounds of trials on the entire dashboardQA dataset, with each batch having different observation types given for environment representation:

- i) **Screenshot:** A raw pixel-level representation of the dashboard interface, showcasing the most difficult comprehension challenge whereby models must handle all reasoning and GUI grounding without supplementary materials. This requires extracting precise mappings and understanding of dashboard layouts, and ultimately is the best means of assessment for raw model capabilities.
- ii) **Screenshot + accessibility (a11y) tree:** An a11y, represented in XML format, containing meta information about UI components seen in the webpage [83] is obtained through the ATSPI framework [13] and appended to the screenshot input. The given information entails the component type, title, location coordinates, and size dimensions. This helps to evaluate how useful supplemental inputs to raw screenshots are bypassing common limitations seen in navigating GUI interfaces [13].

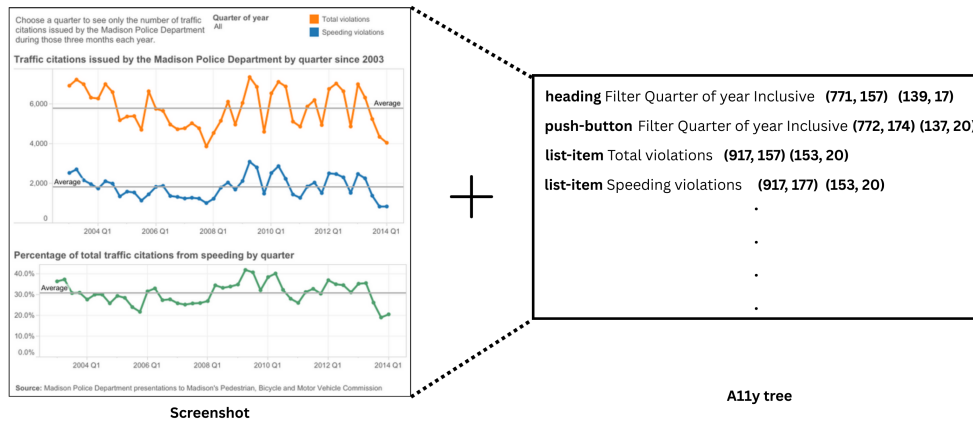


Figure 4.2: Example dual observation space given in (ii). On the left a dashboard view screenshot is given, and to the right is the a11y tree representation of the dashboard UI elements.

## 4.5 Summary

This section has outlined the framework for testing the VLMs on the tasks in DashboardQA with the use of OSWorld and the tableau public website. The environment setup, prompt setups for each question category, action space and two types of observation spaces used have been given.

# Chapter 5

## Evaluation

This section summarizes the results achieved for all the open, closed, and hybrid model setups on the DashboardQA benchmark, and analyzes how the reasoning behavior contributes to the model's performances; after summarizing the models evaluated, outlining the evaluation framework, and examining the results achieved, a qualitative and quantitative analysis of task workflows by the models are given.

### 5.1 Baselines

Overall, 5 closed source models, 3 standalone open source models, and 2 hybrid approaches were assessed on the dataset. The open source setups involved testing various independently crafted models with various parameter values. UI-TARS 2B, UI-TARS 1.5 -7B, and GUI-OWL 7B were chosen as the standalone open source models, whereas the hybrid approaches entailed Jedi-3b and 7b with GPT4o. The closed source models include of Gemini-Pro-2.5, GPT4o and it's smaller size alternatives of O4-mini and GPT5-mini, as well as of OPENAI's CUA.

## 5.2 Evaluation Metric

To calculate the accuracy of the responses we utilize the relaxed accuracy metric used in ChartQA Pro [8] in order to encompass all the respective categories of questions. This is shown in the piece-wise function in (1). Multiple different evaluation styles seen in various VQA datasets are utilized to account for the different requirements required for different answer types. Given the difficulty of the task at hand, an error margin of 5 percent is given for a prediction  $p$  against ground truth  $t$  for each category. Notably, we utilize the Average Normalized Levenshtein similarity (ANLS) metric [57], specifically made for evaluating visual question answering where images have textual information. List-based answers were evaluated by parsing the solution list, comparing with the ground truth, and finding the ratio of correct entries to total entries. The overall accuracy is calculated as the accumulated correctness sum divided by the total number of tasks.

$$C(p,t) = \left\{ \begin{array}{ll} EXACTMATCH(p,t), & \text{if MCQ or the answer involves years,} \\ 1, & \text{if numeric and } \left| \frac{p-t}{t} \right| \leq 0.05, \\ 0, & \text{if numeric and } \left| \frac{p-t}{t} \right| > 0.05, \\ ANLS(p,t), & \text{Otherwise} \end{array} \right\} \quad (5.1)$$

$$C_{\text{list}}([p_1, p_2, \dots, p_n], [t_1, t_2, \dots, t_n]) = \frac{1}{N} \sum_{i=1}^N C(t_i, p_i). \quad (5.2)$$

$$\text{Accuracy} = \frac{1}{M} \sum_{j=1}^M C_j \quad (5.3)$$

Table 5.1: Accuracy (%) on DashboardQA across different Observation Types (main headers) and Question Categories (sub-headers). Each Observation block includes five question types along with an Overall score. Color shading indicates model category: closed models (grey), hybrid models (blue), and fully open-source models (orange). The highest score within each category is shown in bold.

Model	Screenshot						Screenshot + A11y Accessibility Tree					
	Factoid	MCQ	Convers.	Hypoth.	Multidash.	Overall	Factoid	MCQ	Convers.	Hypoth	Multidash	Overall
<i>Closed-Source Models</i>												
GPT4-o	8.88	20.00	14.81	<b>7.69</b>	4.35	11.50	27.25	35.00	17.86	13.85	10.87	22.94
O4-mini	1.35	1.25	0.00	0.00	0.00	0.68	7.92	22.50	0.00	2.82	2.17	9.14
GPT5-mini	1.35	2.50	0.00	0.00	0.00	1.03	13.11	27.50	7.41	16.55	0.00	15.28
Gemini-Pro-2.5	10.33	15.00	22.22	<b>7.69</b>	<b>8.70</b>	11.86	<b>40.03</b>	<b>46.25</b>	<b>51.20</b>	<b>39.27</b>	<b>15.22</b>	<b>38.69</b>
OpenAI CUA	<b>26.45</b>	<b>42.50</b>	<b>29.63</b>	7.22	0.00	<b>22.69</b>	22.24	38.75	16.93	3.08	0.00	18.50
<i>Open-Source Models</i>												
Jedi-3B w/GPT4o	32.70	40.00	44.85	23.41	<b>6.67</b>	29.73	N/A	N/A	N/A	N/A	N/A	N/A
Jedi-7B w/GPT4o	<b>35.33</b>	<b>42.50</b>	<b>49.52</b>	<b>34.00</b>	2.17	<b>33.09</b>	N/A	N/A	N/A	N/A	N/A	N/A
UI-Tars-2B	0.75	0.00	0.00	0.00	0.00	0.19	N/A	N/A	N/A	N/A	N/A	N/A
UI-Tars-1.5-7B	8.98	<b>12.50</b>	0.00	2.40	0.00	6.23	N/A	N/A	N/A	N/A	N/A	N/A
GUI-OWL-7B	<b>17.12</b>	1.25	<b>25.93</b>	<b>17.75</b>	<b>0.37</b>	<b>11.12</b>	N/A	N/A	N/A	N/A	N/A	N/A

### 5.3 Main Results

As shown in Table 5.1 all models demonstrated poor overall performance, struggling in every question category. While noticeable improvements are seen when models are aided with a11y trees or utilizing hybrid approaches, the overall accuracy percentages never break the 40 % threshold. Most of the variations in results between the SOTA, hybrid models, and standalone open source models were as expected. Gemini pro 2.5 shows the best overall performance (38.69%) when aided with an a11y tree, while Jedi-7B with GPT4o shows the best performance with sole screenshot (33.09%). GPT4’s smaller size counterparts showed very poor performance, with O4-mini achieving 0.68% and GPT5-mini 1.03%. displaying results comparable to standalone open-source models. Notably, OPENAI CUA shows a decline in results when given an a11y tree, indicating that the reinforcement learning framework it was built on did not comply with supplemental meta data approaches.

Comparatively, most open source models underperformed compared to the best performing approaches, but exhibited comparable performance to smaller GPT4 variants. UI-

Tars-2B showed the worst results overall (0.19%), with the upgraded variant showing substantial increase (6.23%). GUI-OWL-7B performed substantially better than its open source peers in standalone approaches, displaying inference abilities to reason beyond just GUI grounding and abilities to act as a sole agent.

Multiple Dashboard questions proved to be the most challenging category, obtaining the lowest results in all cases except for Gemini Pro 2.5. Additionally, hypothetical questions proved to be exceptionally difficult as well, but with notable increases when aided with a11y trees. Conversational questions went better than expected, attributable to the simplified evaluation approach involving just the final question to be answered. MCQs and factoids showed to be the best performing category for the models with their direct reasoning requirements.

## **5.4 Qualitative Analysis**

25 samples were taken from (1) Gemini pro 2.5, (2) GPT4o, and (3) Jedi-7B w/GPT-4o to find strengths and weaknesses of the VLMs, and what common patterns were seen in the gold correct and erroneous samples. Their runtime logs were examined thoroughly to look at the reasoning steps and trajectories taken for the various categories of questions.

### **5.4.1 Strengths**

Upon inspection, the SOTA models displayed competence in trajectory planning, extracting and extrapolating information from the visualizations present in the dashboard, and utilizing the navigation tools available. In particular, hybrid approaches like using Jedi 7B with GPT4o showed sound ability in not just navigating to relevant new views, but in attempting to extract hidden details from visualizations shown through hovering over certain data points to reveal exact values, and utilizing self-reflection to re-locate to the correct points when needed. This

demonstrates that GUI-oriented models additions help enable dashboard GUI grounding.

## 5.4.2 Weaknesses

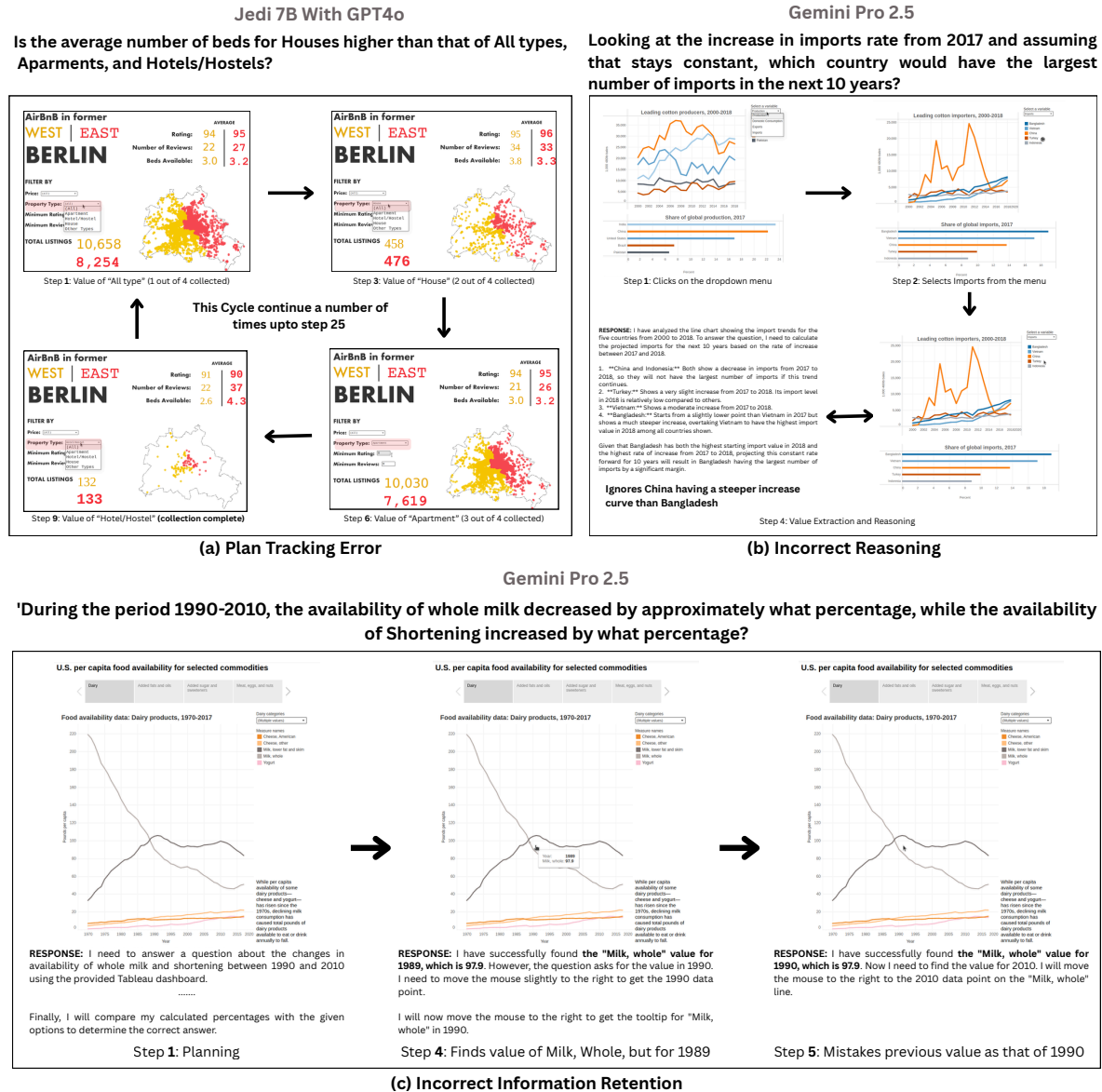


Figure 5.1: DashboardQA Error Analysis: Illustrative cases of VLM failures, including plan-tracking errors, incorrect trend reasoning, and loss of intermediate values during interactive dashboard analysis.

Ultimately, the poor results obtained by the VLMs can be explained through common mistakes. The following types of errors in reasoning were found:

**Plan tracking errors:** While the models showed good trajectory mappings in its initial planning, a notable fault was their inability to utilize the information retrieved in the first few steps. At times, the models were observed to forget important intermediate values retrieved, causing them to revert back to the same states. Often, this cycle would repeat in a loop until the maximum number of steps was achieved. This is seen in figure X for example with Jedi 7B with GPT4o, whereby all the 4 required states were visited by the 9th step, but the system kept looping over them and with the 25 step limit being reached no answer was ultimately given.

**Incorrect interpretations/hallucinations:** Hallucinations, representing mismatches between the visuals given and text analysis generated by the models [84], were another common error seen. An example of this is seen in subfigure (b) in 5.1 where the VLM is required to compare trajectories of various countries cotton imports as depicted in a multi-line chart. The model was unable to notice the steeper increase from 2017 by China and gives Bangladesh as the answer. Notably, tiny perturbations to what would otherwise be standard questions were not recognized at times. Figure 6 depicts another example on the left where the unanswerable question given simply gives a out of bounds date as compared to what is given in the dashboard, but the VLM doesn't recognize the issue.

**Inconsistencies in information retention:** Occasionally the models in latter steps would incorrectly recall intermediate values needed. Subfigure c in 5.1 shows this where an incorrect date value, which was in fact noted as faulty by the VLM, was taken into computation. This, along with the aforementioned plan tracking errors, showed that on occasion the longer context tasks involved in DashboardQA imposed too much of a mix of memory and reasoning demands to handle.

**Incorrect parsing:** Despite being explicitly instructed to give their answers as structured

tags, there were cases where the VLMs were unable to parse their own answer, leading to repeated task executions with either repeated answers or new ones eventually being calculated. Although there were cases where the models had got the correct result and failed to parse the answer (leading to either a new incorrect calculation or an exhaustion of the step limit).

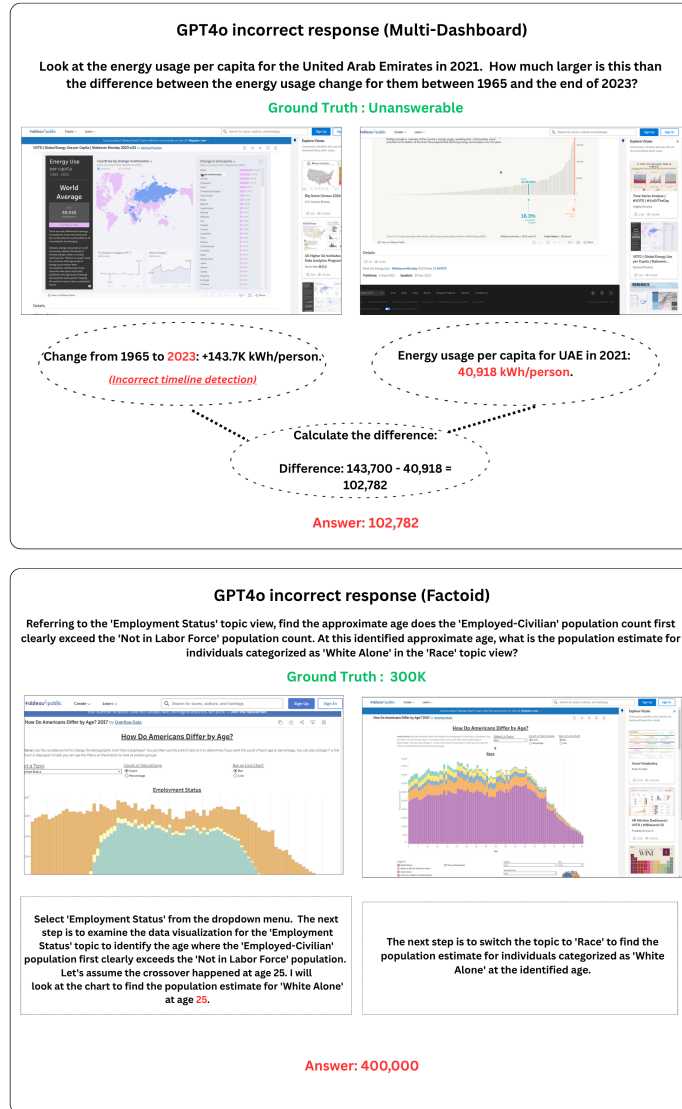


Figure 5.2: Example incorrect reasoning trajectories as done by GPT4o.

Overall, while there was impressive ability in tackling many sub-tasks required for

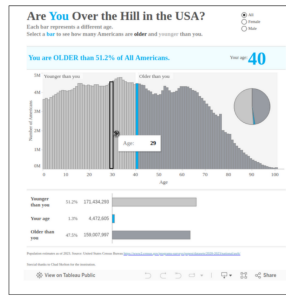
the problem, there were issues in synthesizing all the information collected efficiently and maintaining a correct reasoning trajectory. The primary difficulty with DashboardQA with the final answer evaluation is that a single mistake can offset all other sound efforts.

### 5.4.3 Breakdowns and Solutions

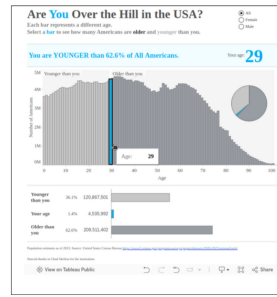
Upon inspection of the logged reasoning steps for completing DashboardQA tasks, many breakdowns were seen while navigating and reasoning with tableau dashboards, attributable to their visual, functional, and interactive complexity and diversity. In this context, a breakdown is specifically defined as any situation where the model faced any impediments in completing any particular action in dashboard interface grounding and trajectory reasoning. To tackle such challenges, a diverse set of workarounds taken by the VLMs were observed, displaying VLM capabilities of handling adversity and utilizing self-reflection. This ranged from repetitive re-calibrations to completely new routes taken to achieve objectives. Some of the following breakdown and workaround were seen:

- **Visualization usage failure:** Often, locating items whose bounding box is not explicitly given through the ally tree would cause difficulties in pinning exact locations. Example cases are seen in figures 5.3 and 6 when attempting to locate exact data points in charts to hover over requiring precision clicking in dense visualizations. At times when tooltips were not showing up to display exact data values, the VLM would attempt to utilize downloadable data such as CSV files to find underlying data. An example case of this is seen in steps 12 and 15 in 5.7.

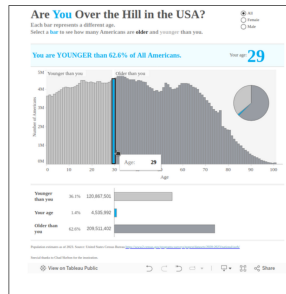
## Visualization usage failure



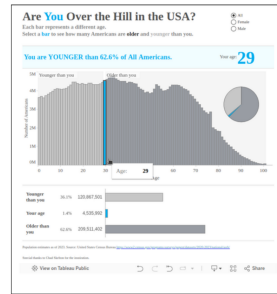
Step 1: I need to click the "30" age bar to see the relevant data. Click on the age 30 bar in the histogram  
 pyautogui.click(651, 477, duration=1)



Step 3: The age 29 bar is currently selected in the histogram. I need to select the age 30 bar.  
 pyautogui.click(649, 522, duration=1)



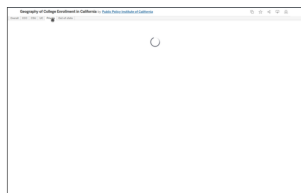
Step 5: The age is set to 29, not 30. I need to set the age to 30  
 pyautogui.click(649, 540, duration=1)



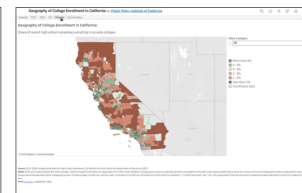
Step 6: The age is set to 29. I need to change it to 30.  
 pyautogui.click(652, 561, duration=1)

Result: Fail

## Environment breakdown



The "Private" tab is currently loading. The map and data have not yet appeared. I need to wait for the "Private" tab to load



The map now shows private college enrolment by county with various color codes.

Figure 5.3: Examples visualization tool usage failure and environment breakdown. The former shows mouse position recalibration for accurate cross filtering on a bar chart, and the latter waiting for the case of dashboard loading latency.

- **Navigation tool usage failure:** These were mostly the case for multi-option drop-down menus and range sliders. For the former, wrong options, or wrong dropdown menus would be clicked on. Approaches to tackling these errors would primarily center around re-calibrating the positions of the mouse through spatial reasoning or, if possible to manually type the option for isolation and making it easier to click on. Often, the models would decide to leave the multi-option menu as a whole and focus on

isolating the particular (topic) through using the legends and the "keep only" attribute available. This is seen in steps 3,5, and 6 in 5.7. Additionally, when using range sliders and not able to drag to the correct positions the VLM would resort to manual input or using arrow icons.

- **Planning failure:** In some cases the VLMs misjudged the required states to answer the question, and would have to reroute. However, sometimes they would not be able to utilize the information available and inefficiently traverse the dashboard. Figure 5 shows this where both the tabs metadata text is not used and the VLM skips over the correct state, while going back to incorrect states.
- **Environment bottleneck:** Often, an OS or webpage related bottleneck would appear. From the OS side, when the VLM would take too long they would be logged out. The VLM would simply type the password and login. With respect to the web pages, sometimes there would be dashboard loading latency or a 404 error. To tackle this, the models would either simply wait or refresh the page.

In order to elucidate the strategies taken by the agents in various contexts, a taxonomy of the high level goals, breakdowns, and workarounds/fixes are shown in table 5.2. An example task workflow by Gemini Pro 2.5 demonstrating some of the workaround/fixes to various breakdowns is presented in figure 5.7.

## 5.5 VLM Reasoning Behavior Quantitative Analysis

The logged reasoning thoughts of the VLMs are quantitatively evaluated through three dimensions: step count, step lengths, and workaround breakdown pairings. The first two categories are done to assess their efficiency levels. Specifically, this helps to gauge how often models would either converge effectively, or diverge off track or loop due to erroneous and redundant

Table 5.2: Goals, Breakdowns, and Workarounds seen in VLM-based interactive dashboard environments

High level goal	Description
Environment/interface use	Able to successfully interact with tableau public’s dashboard interface through Ubuntu .
Data retrieval	Finding a specific data value in dense visualizations.
Dashboard navigation	Utilizing navigation tools to go to new dashboard states.
Information synthesis/Calculation	Compound operations on extracted or given values pertaining to the given question’s requirements.
Breakdown Type	Description
Navigation tool usage failure	Unable to click or configure widgets / navigation tools correctly for navigation.
Visualization usage failure	Unable to extract required information from data visualizations through interaction.
Environment/Interface bottleneck	Unable to use dashboard due to webpage or OS bottlenecks.
Planning failure	Unable to gather relevant/required results from planned path into dashboard state.
Workaround/Fix	Description
Re-calibrate position	Move mouse position (x,y) for precision clicks.
Tool re-orientation	Use the same navigation tool in a different manner.
Change tool/widget	Move to different widget.
Modality shift	Move from data visualizations to underlying data files to retrieve data. .
Environment / Interface reorientation	Refresh webpage/research URL, change to previous webpage tab.
Wait	Wait for state to re-orientate itself.

reasoning. The last category is to see what types of workaround strategies lean to and stray away from.

Figures 5.7 and 7 quantifies the workaround strategies taken by Gemini Pro 2.5 (aided

by an A11Y tree) and GPT4o with Jedi 7B. In order to gather these results, 292 tasks for were analyzed through the use of few-shot prompting with the openAI API utilizing gpt4o. Specifically, the VLM’s reasoning logs were parsed and given as input, along with the instructions of finding all breakdown-workaround correspondences. The prompt used is given in 5.4. Overall, most issues faced stemmed from dashboard navigation and data retrieval issues, associated with GUI grounding capabilities. The results show how hybrid approaches are less prone to breakdowns in general and typically have a less varied workaround solution space. However, when looking at the heatmaps and comparing that with the performance results seen where Gemini with an a11y tree outperforms GPT4o with Jedi 7B, Gemini Pro 2.5 is shown to be a very robust model, persevering to ultimately obtain their answer with a higher degree of accuracy.

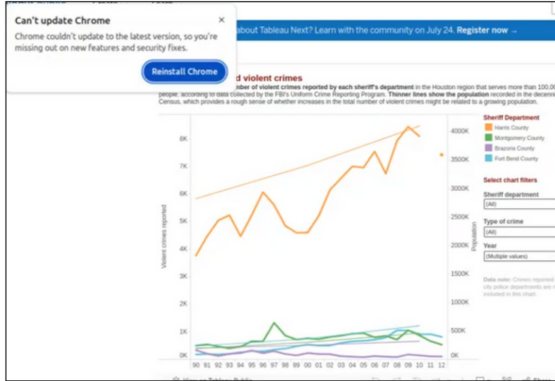
```

Role:
You are an analyst summarizing logged task executions taken by Vision Language Models in response to an interactive dashboard question answering task. Given a series of reasoning steps taken by the VLM, you must identify all the breakdowns and workarounds seen when accomplishing specific goals.
Rules:
-You must output in JSON format only.
-The breakdown, workarounds, and goals must be taken from the following lists:
Breakdowns: f{breakdowns}
Workarounds: f{workarounds}
Goals: f{goals}
-The evidence strings must be taken directly as given in the reasoning steps.

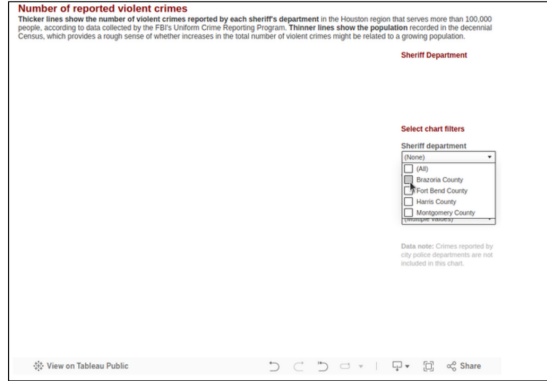
Contents:
I am giving you the VLM generated reasoning steps here: {thoughts}.
Note: Don't mistake common task workflows as a breakdown, the first view not being in the right state and parsing issues are not breakdowns. Breakdowns occur only when there are general impediments to completing and reasoning with the task at hand.
If there are none at all, just return an empty list for extractions.
Output only JSON using this design format, replacing variable placeholders as needed:
{{ "File": "{task_num}_runtime.log",
  "extractions": [ {{
    "goal": "string for goal 1",
    "breakdown": "issue_faced",
    "evidence_breakdown": "log_quote",
    "workarounds": [
      {{ "action": "workaround_x",
        "workaround_num_attempts": "N (Must be an integer)",
        "evidence_action_1": "string A from reasoning logs",
        "workaround_num_attempts": "P (Must be an integer)",
        "evidence_action_1": "string A from reasoning logs." }},
      {{ "action": "workaround_y",
        "workaround_num_attempts": "Z (Must be an integer)",
        "evidence_action_2": "string B from reasoning logs" }}
    ]
  }
  ], }}}}

```

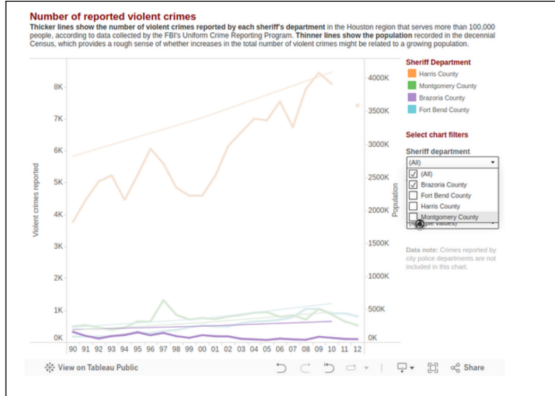
Figure 5.4: Prompt used to find VLM goals, breakdowns, and workarounds



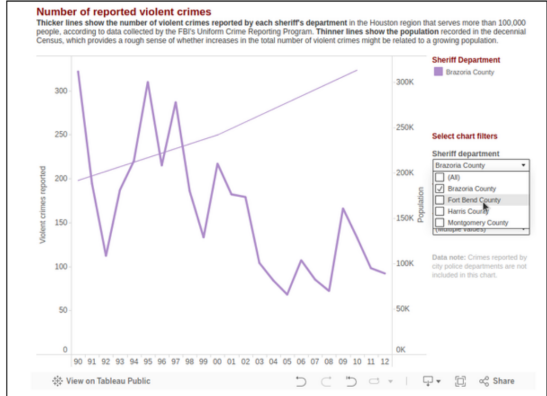
Step 1: First, I see a "Can't update Chrome" pop-up that is obscuring part of the view. I will close this pop-up first to have a clear view of the dashboard.



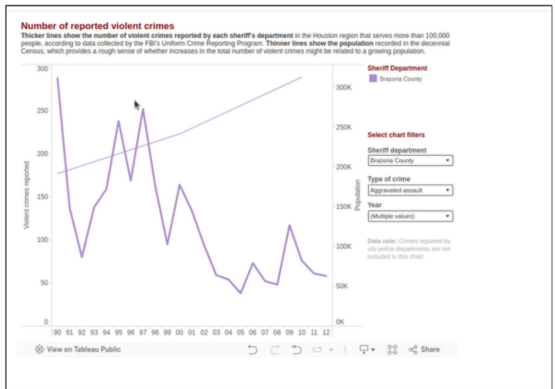
Step 3: I have opened the "Sheriff department" filter dropdown. Now I need to select only "Brazoria County". To do this, I will first deselect "(All)" to clear the current selections, and then I will select "Brazoria County".



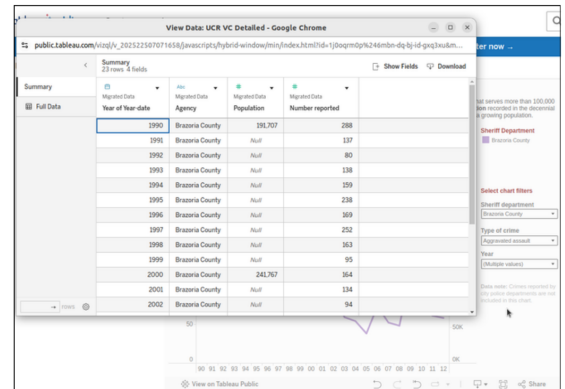
Step 5: My previous action of deselecting "(All)" and selecting "Brazoria County" seems to have resulted in an empty chart. Instead of deselecting all and then selecting one, which failed, I will deselect the other counties one by one, leaving only "Brazoria County" checked.



Step 6: I made an error in my previous step by clicking on a location that was part of the open "Sheriff department" filter dropdown, instead of the "Type of crime" filter. My immediate next step must be to close this dropdown so I can access the "Type of crime" filter.



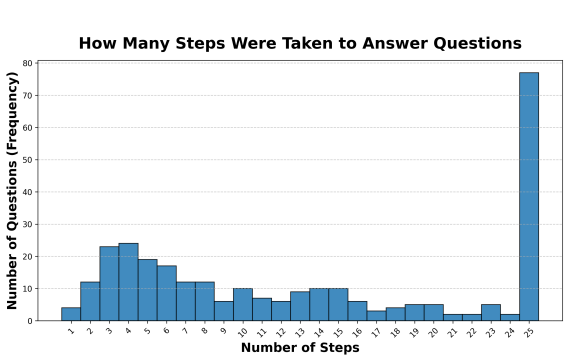
Step 12: I have moved the mouse over the peak of the line chart, but the tooltip with the data did not appear. I have tried hovering and clicking on the peak of the line chart to



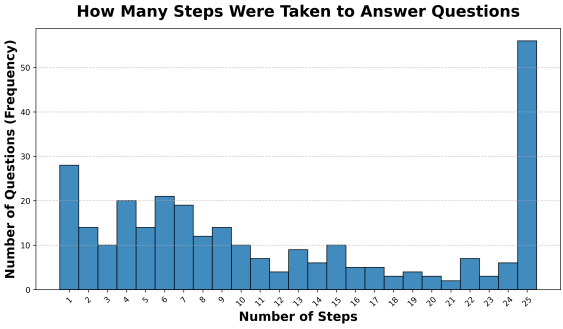
Step 15: The most promising option is to use the "Download" feature. I have successfully opened the data view, which provides a table with the exact numbers for

Figure 5.5: Showing multiple different breakdowns and workarounds for a sample task execution by Gemini Pro 2.5

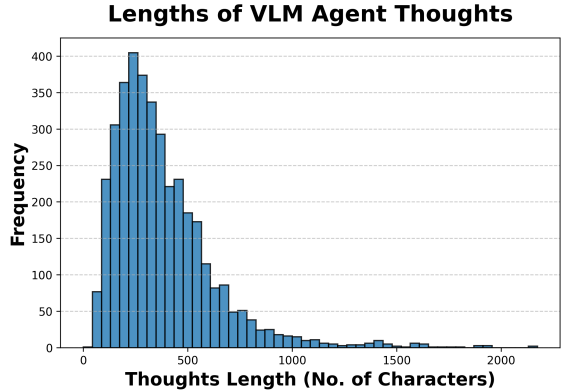
Figure 7 illustrates the distribution of steps and lengths of the highest achieving agents thoughts for completing the problems given in DashboardQA. Quantitatively, Gemini Pro 2.5 required additional steps and in carrying out the tasks of dataset compared to our hybrid model setup. In particular, the model appeared to reach the maximum step limit more often. Notably, the hybrid approach showed a far more skewed distribution towards its reasoning steps lengths, with a much larger range as well.



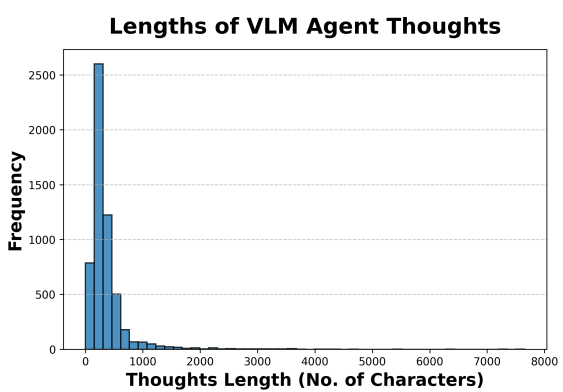
(a) Distribution of steps taken by Gemini-Pro-2.5.



(b) Distribution of steps taken by Jedi-7B w/ GPT-4o.



(c) Lengths of reasoning thoughts generated by Gemini.



(d) Lengths of reasoning thoughts generated by Jedi-7B.

Figure 5.6: Comparison of reasoning behaviors between Gemini-Pro-2.5 and Jedi-7B w/ GPT-4o on the DashboardQA benchmark. The plots illustrate (a, b) the distribution of the number of interaction steps taken to answer questions and (c, d) the lengths of intermediate reasoning thoughts.

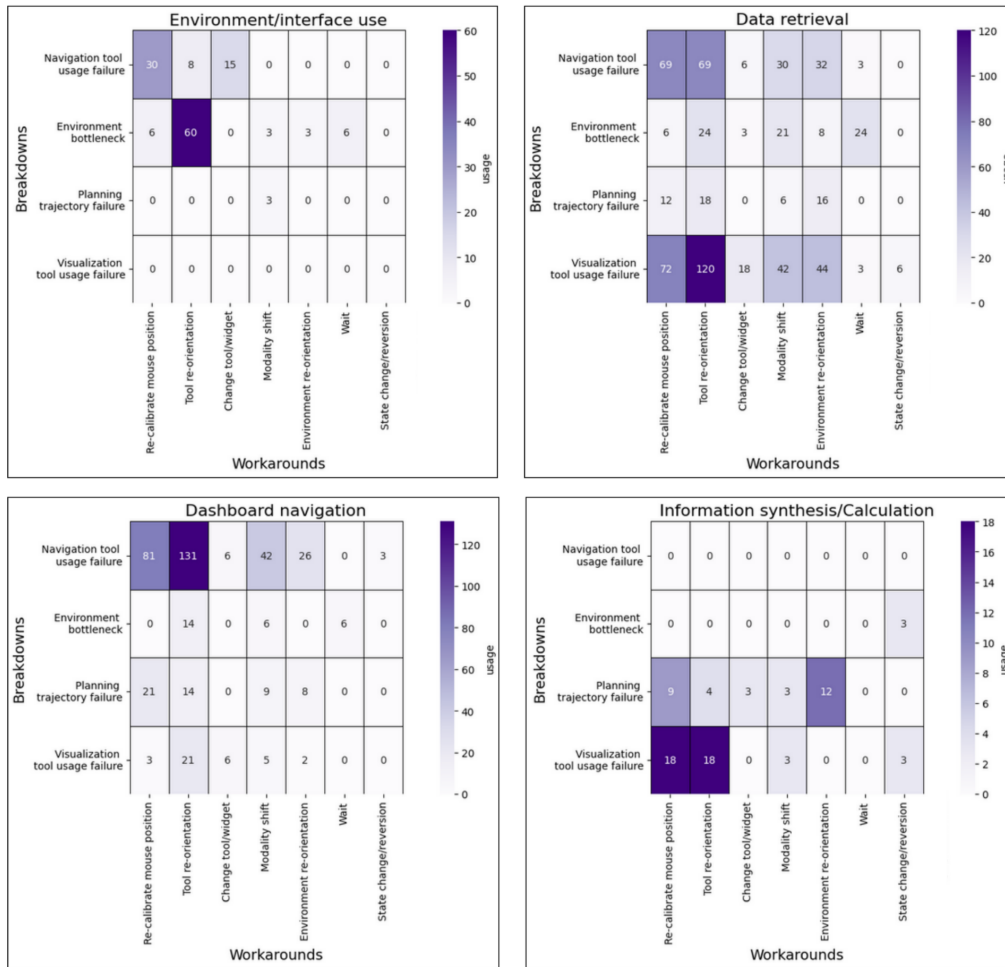


Figure 5.7: Heatmap representing breakdown-workaround pairs found for Gemini Pro 2.5 task executions

## 5.6 Ablation Study

To gauge the effect of task complexity on SOTA model performance, a benchmark ablation study is conducted where questions are categorized into 'Easy' (1-2), 'Medium'(3-5), or 'Hard'(6+) - based on the number of steps required to answer from. Specifically, top performing model observation space combos from 5.1 are chosen - Gemini Pro 2.5 when given a screenshot + a11y tree and Jedi-7B with GPT4o with just a screenshot. From table 5.6, there is a clear performance decline among both models in more difficult tasks, and with Gemini outperforming in easier and harder tasks, but the hybrid approach just surpassing in Medium tasks. This shows that Gemini Pro 2.5 as a model is relatively competent in long horizon planning tasks.

Table 5.3: Ablation results for Gemini Pro 2.5 with an a11y tree and Jedi 7B W GPT4o on DashboardQA across 3 task difficulty levels: EASY, MEDIUM, and HARD.

<b>Task Difficulty</b>	<b>Gemini Pro 2.5 (Screenshot + A11y)</b>	<b>Jedi-7B w/ GPT4o (Screenshot)</b>
Easy	50.64	40.40
Medium	32.17	32.31
Hard	28.86	18.74
<b>Overall</b>	<b>38.69</b>	<b>33.09</b>

## 5.7 Summary

This chapter has outlined the evaluation framework used and analyzed the results achieved by various types of VLMs. Additionally, an analysis of the strengths and weaknesses of VLMs

is given. Further analysis of the reasoning workflows, specifically the workaround strategies taken when dealing with breakdowns, is given through a taxonomy and quantified by their respective goals. Finally, a benchmark ablation study is conducted to elucidate the effect of the question difficulty, as highlighted by step ranges, towards the performance levels.

# Chapter 6

## Conclusion and Future Work

### 6.1 Conclusion

This work aimed to gauge the capabilities of modern Vision Language Models to navigate and reason among interactive dashboards through various categories of question answering tasks. To do so, we introduced DashboardQA, a novel benchmark for assessment on carefully curated dashboards. Additionally, a custom framework was created for evaluation through OS-World, and with the results retrieved an analysis of the performances of closed, open-sourced, and hybrid models was given. Finally, following the creation of a customized taxonomy of breakdowns and workarounds/fixes for respective goals, their mappings were quantified for the highest performing models.

Our study showed that VLMs face significant challenges in exploration based question answering with interactive dashboards, lacking the ability to combine accurate GUI grounding with multi-state reasoning throughout their trajectories. Notably, while VLMs showed promise with good initial planning and good understanding of task requirements, they struggled with maintaining correct reasoning throughout their trajectories. Given that these multi-hop question answering tasks require compound operations, their answers were susceptible

to error propagation.

Most of the variations in results between the models achieved were as expected. Hybrid models performed the best when given standalone screenshots, and Gemini Pro 2.5 with a11y trees was the best performing approach overall. When aided with a11y trees or using a higher parameter model variant, models generally had great improvements in their results. With respect to dealing with breakdowns, it appears that standalone proprietary models like Gemini pro 2.5 (aided with a a11y tree) face more breakdowns than hybrid approaches like Jedi with GPT4o, but are quite robust with their workarounds and fixes as seen by a better accuracy rate. This can also serve as indicators that open source GUI models need revamping if approaches with a11y trees can obtain better results.

Ultimately, this work serves as a foundation to understand how VLMs could be used in real world applications to expedite and enhance the process of utilizing and comprehending dashboard content. The empirical analysis built in this paper serve can help to build more robust agentic VLM frameworks.

## 6.2 Limitations

Even though a great effort was made to develop a holistic benchmark, inherent limitations of the dataset, as well as of the evaluation setup, must be noted.

**Tableau oriented:** Firstly, all of the dashboards curated were from tableau. This limits the spatial, topic, and navigational diversity that would exist otherwise if visualizations from alternative visualization platforms such as Power BI and Zoho analytics were included. In addition, the website URLs on tableau public were utilized as they allowed for accessibility tree access, allowing for smooth, no fault agentic navigation evaluation. This is not available directly for desktop apps/offline use.

**Expert annotation bias:** For the purely human and human with VLM assistance questions

made, only those proficient in the fields of data analytics and NLP were used in the dataset generation process. While in a professional sense this assesses the dashboards well, this does not take into consideration the context of industries with stakeholders of varying knowledge levels on the topic at hand and dashboard usage proficiencies. Thus, the semantic, linguistic, and complexity diversity of the QA pairs may be lacking.

**Conversational and Multi-Dashboard question answering scope limitations:** While our conversational and multi-dashboard categories proved to be challenging for all the VLMs, there were restrictions placed on them. The evaluation of the conversational category only focused on the final question’s answer. This was done due to the high inference costs that would be involved in calculating all turns. Such a limited approach overlooks the true challenge of conversational question answering to maintain accuracy throughout; marginal errors in initial questions can propagate through the conversation and have compounding effects. This occurs as intermediate values obtained in earlier questions would be included in composite calculations later. For similar reasons, the multi-dashboard setup only involved 2 dashboards per question, when real-world setups could involve analyzing between a larger number of dashboards.

## 6.3 Future Work

The rapid advancements of the AI landscape have given various research directions to expand upon this work of assessing VLM capabilities with interactive visualizations. Primarily, this pertains to developments to our existing benchmark and evaluation framework, the use of specialized multi-agentic frameworks in evaluation, and looking at various other dashboard related downstream tasks. Example future work applications include:

- There are more possible categories of questions that could be added. In particular,

Retrieval Augmentation Generation (RAG) has shown increasing popularity in tackling VLM hallucinations and addressing limitations to pretraining knowledge by utilizing information retrieval techniques on external corpus of information to append to prompts for more accurate, contextual answer generation [85]. In our context, most of the dashboards represented in DashboardQA contain source links to additional metadata about the topic at hand, representing another exploration domain.

- Evaluation could be done in offline settings (i.e with the tableau desktop app). A mixed offline and online testing setup would align more with the complexities data analysts could face in the real world.
- DashboardQA was evaluated through a single or double agent hybrid framework. This does not allow for gauging the enhanced abilities of VLMs achieved through larger scale collaboration with more specific task delegation, and through the use of layered Chain of thought prompting [86].
- Recent research has looked into VLMs as judges approach to gauge their effectiveness in analyzing responses in chart question answering , which can be utilized to bypass the sheer cost, effort, and time required for human made qualitative analysis [87]. This can be extended upon to DashboardQA to assess capabilities of models in evaluating long-term reasoning in exploration. Specifically, ideas from workarounds and breakdowns can be used to supplement automated trajectory analysis.
- Many downstream tasks exist aside from general question answering. This includes of retrieval, coding (i.e generating code for dashboard generation or dashboard analysis scripts), or summarization tasks. It would be interesting to expand upon assessing, evaluating, and improving the abilities of VLMs in more niche areas. In turn, this can help in the integration with multi-agentic VLM frameworks for data analysis pipelines.

- Previous HCI work with chart comprehension has focused on making data visualizations more accessible to those who are visually impaired [88]. It would be useful to expand upon this to the case of dashboard usage. For example, generated output from VLMs can be turned into auditory output to help visually impaired users gain insights about the contents.

# Bibliography

- [1] Jaspersoft, “What is an interactive dashboard?.” <https://www.jaspersoft.com/articles/what-is-an-interactive-dashboard#:~:text=A%20key%20benefit%20of%20an,drilling%20down%2C%20and%20examining%20it>. Accessed: 2025-05-28.
- [2] A. Pakdaman, M. Karajizadeh, M. Nasiri, and R. Sharifian, “Business intelligence dashboard in healthcare: Lesson learned,” *Health Management & Information Science*, vol. 9, no. 3, pp. 184–191, 2022.
- [3] E. M. M. Yusof and M. S. Othman, “A review on the dashboard characteristics for manufacturing organizations,” *Journal of information systems research and innovation*, vol. 2, pp. 28–34, 2012.
- [4] H. P. Kothandapani, “Drivers and barriers of adopting interactive dashboard reporting in the finance sector: an empirical investigation,” *Reviews of Contemporary Business Analytics*, vol. 2, no. 1, pp. 45–70, 2019.
- [5] J. I. Amundsen, “Dashboards and information overload: Challenges and design guidelines for health information management systems in developing countries,” master’s thesis, University of Oslo, 2019.

- [6] M. Alhamadi, O. Alghamdi, S. Clinch, and M. Vigo, “Data quality, mismatched expectations, and moving requirements: The challenges of user-centred dashboard design,” in *Nordic Human-Computer Interaction Conference*, NordiCHI ’22, p. 1–14, ACM, 2022.
- [7] A. Masry, D. X. Long, J. Q. Tan, S. Joty, and E. Hoque, “ChartQA: A benchmark for question answering about charts with visual and logical reasoning,” in *Findings of the Association for Computational Linguistics: ACL 2022*, (Dublin, Ireland), pp. 2263–2279, 2022.
- [8] A. Masry, M. S. Islam, M. Ahmed, A. Bajaj, F. Kabir, A. Kartha, M. T. R. Laskar, M. Rahman, S. Rahman, M. Shahmohammadi, M. Thakkar, M. R. Parvez, E. Hoque, and S. Joty, “ChartQAPro: A more diverse and challenging benchmark for chart question answering,” in *Findings of the Association for Computational Linguistics: ACL 2025*, pp. 19123–19151, 2025.
- [9] A. Masry, M. Shahmohammadi, M. R. Parvez, E. Hoque, and S. Joty, “ChartInstruct: Instruction tuning for chart comprehension and reasoning,” in *Findings of the Association for Computational Linguistics: ACL 2024*, pp. 10387–10409, 2024.
- [10] E. Hoque, P. Kavehzadeh, and A. Masry, “Chart question answering: State of the art and future directions,” *Computer Graphics Forum*, vol. 41, no. 3, pp. 555–572, 2022.
- [11] B. Shneiderman, “The eyes have it: a task by data type taxonomy for information visualizations,” in *Proceedings 1996 IEEE Symposium on Visual Languages*, pp. 336–343, 1996.

- [12] W. Huang, Z. Gu, C. Peng, J. Liang, Z. Li, Y. Xiao, L. Wen, and Z. Chen, “Auto-Scraper: A progressive understanding web agent for web scraper generation,” in *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, (Miami, Florida, USA), pp. 2371–2389, 2024.
- [13] T. Xie, D. Zhang, J. Chen, X. Li, S. Zhao, R. Cao, T. J. Hua, Z. Cheng, D. Shin, F. Lei, *et al.*, “Osworld: Benchmarking multimodal agents for open-ended tasks in real computer environments,” *Advances in Neural Information Processing Systems*, vol. 37, pp. 52040–52094, 2024.
- [14] Z. Wang, Y. Cui, L. Zhong, Z. Zhang, D. Yin, B. Y. Lin, and J. Shang, “Officebench: Benchmarking language agents across multiple applications for office automation,” *arXiv preprint arXiv:2407.19056*, 2024.
- [15] C. Rawles, S. Clinckemahillie, Y. Chang, J. Waltz, G. Lau, M. Fair, A. Li, W. E. Bishop, W. Li, F. Campbell-Ajala, D. K. Toyama, R. J. Berry, D. Tyamagundlu, T. P. Lillicrap, and O. Riva, “Androidworld: A dynamic benchmarking environment for autonomous agents,” in *The Thirteenth International Conference on Learning Representations*, 2025.
- [16] Y. Qin, Y. Ye, J. Fang, H. Wang, S. Liang, S. Tian, J. Zhang, J. Li, Y. Li, S. Huang, W. Zhong, K. Li, J. Yang, Y. Miao, W. Lin, L. Liu, X. Jiang, Q. Ma, J. Li, X. Xiao, K. Cai, C. Li, Y. Zheng, C. Jin, C. Li, X. Zhou, M. Wang, H. Chen, Z. Li, H. Yang, H. Liu, F. Lin, T. Peng, X. Liu, and G. Shi, “Ui-tars: Pioneering automated gui interaction with native agents,” *arXiv preprint arXiv:2501.12326*, 2025.
- [17] T. Xie, J. Deng, X. Li, J. Yang, H. Wu, J. Chen, W. Hu, X. Wang, Y. Xu, Z. Wang, Y. Xu, J. Wang, D. Sahoo, T. Yu, and C. Xiong, “Scaling computer-

- use grounding via user interface decomposition and synthesis,” *arXiv preprint arXiv:2505.13227*, 2025.
- [18] T. Broughton, A. M. Weggelaar, and S. Sülz, “From data to improvement: Social mechanisms as a key to understanding dashboard adoption,” *American journal of medical quality : the official journal of the American College of Medical Quality*, vol. 40, 2025.
- [19] J. Manchanda, L. Boettcher, M. Westphalen, and J. Jasser, “The open source advantage in large language models (llms),” *arXiv preprint arXiv:2412.12004*, 2024.
- [20] M. Tory, L. Bartram, B. Fiore-Gartland, and A. Crisan, “ Finding Their Data Voice: Practices and Challenges of Dashboard Users ,” *IEEE Computer Graphics and Applications*, vol. 43, no. 01, pp. 22–36, 2023.
- [21] M. Alhamadi, “Challenges, strategies and adaptations on interactive dashboards,” in *Proceedings of the 28th ACM Conference on User Modeling, Adaptation and Personalization*, UMAP ’20, p. 368–371, Association for Computing Machinery, 2020.
- [22] Microsoft, “Filters and highlighting in power bi reports - power bi.” <https://learn.microsoft.com/en-us/power-bi/create-reports/power-bi-reports-filters-and-highlighting>. Accessed: 2025-11-06.
- [23] F. L. Meier, L. Jäger, O. Senn, S. Markun, and J. M. Burgstaller, “Effectiveness of interactive dashboards as audit and feedback tools in primary care: A systematic review,” *PLOS One*, vol. 20, 2025.
- [24] M. Rahman, A. Bhuiyan, M. Saidul Islam, M. T. R. Laskar, R. Mahbub, A. Masry,

- S. Joty, and E. Hoque, “Llm-based data science agents: A survey of capabilities, challenges, and future directions,” *arXiv preprint arXiv:2510.04023*, 2025.
- [25] DOMO, “Dashboard vs report: Choosing the right bi tool for your data needs.” <https://www.domo.com/learn/article/dashboard-vs-report-choosing-the-right-bi-tool-for-your-data-needs>. Accessed: 2025-02-01.
- [26] K. Williamson and R. Kizilcec, “A review of learning analytics dashboard research in higher education: Implications for justice, equity, diversity, and inclusion,” in *LAK22: 12th International Learning Analytics and Knowledge Conference*, LAK22, p. 260–270, Association for Computing Machinery, 2022.
- [27] U.S. Census Bureau, “U.s. census bureau dashboards on tableau public.” <https://public.tableau.com/app/profile/us.census.bureau/vizzes>, 2024. Accessed: April 2025.
- [28] H. Nghiem, H. Du, X. Wu, G. Shi, and Z. Li, “Benchmark evaluations, applications, and challenges of large vision language models: A survey,” *arXiv preprint arXiv:2501.02189*, 2025.
- [29] OpenAI, “Gpt-4 technical report,” *arXiv preprint arXiv:2303.08774*, 2023.
- [30] G. Comanici, E. Bieber, M. Schaekermann, I. Pasupat, and N. S. et al., “Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities,” *arXiv preprint arXiv:2507.06261*, 2025.
- [31] X. Liu, T. Zhang, Y. Gu, I. L. Iong, S. XiXuan, Y. Xu, S. Zhang, H. Lai, J. Sun, X. Yang, Y. Yang, Z. Qi, S. Yao, X. Sun, S. Cheng, Q. Zheng, H. Yu, H. Zhang, W. Hong, M. Ding, L. Pan, X. Gu, A. Zeng, Z. Du, C. H. Song, Y. Su, Y. Dong,

- and J. Tang, “Visualagentbench: Towards large multimodal models as visual foundation agents,” in *The Thirteenth International Conference on Learning Representations*, 2025.
- [32] J. Wei, X. Wang, D. Schuurmans, M. Bosma, b. ichter, F. Xia, E. Chi, Q. V. Le, and D. Zhou, “Chain-of-thought prompting elicits reasoning in large language models,” in *Advances in Neural Information Processing Systems*, vol. 35, pp. 24824–24837, Curran Associates, Inc., 2022.
- [33] J. Bai, S. Bai, S. Yang, S. Wang, S. Tan, P. Wang, J. Lin, C. Zhou, and J. Zhou, “Qwen-vl: A versatile vision-language model for understanding, localization, text reading, and beyond,” *arXiv preprint arXiv:2308.12966*, 2023.
- [34] J. Ye, X. Zhang, H. Xu, H. Liu, J. Wang, Z. Zhu, Z. Zheng, F. Gao, J. Cao, Z. Lu, J. Liao, Q. Zheng, F. Huang, J. Zhou, and M. Yan, “Mobile-agent-v3: Fundamental agents for gui automation,” *arXiv preprint arXiv:2508.15144*, 2025.
- [35] OpenAI, “Computer-using agent.” <https://openai.com/index/computer-using-agent/>, 2025. Accessed: 2025-10-17.
- [36] P. Wang, Y. Yu, K. Chen, X. Zhan, and H. Wang, “Large language model-based data science agent: A survey,” *arXiv preprint arXiv:2508.02744*, 2025.
- [37] R. Kabir, N. Haque, M. S. Islam, and M.-E. Marium-E-Jannat, “A comprehensive survey on visual question answering datasets and algorithms,” *arXiv preprint arXiv:2411.11150*, 2024.
- [38] S. Antol, A. Agrawal, J. Lu, M. Mitchell, D. Batra, C. L. Zitnick, and D. Parikh, “Vqa: Visual question answering,” in *Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV)*, ICCV ’15, (USA), p. 2425–2433, 2015.

- [39] S. Ging, M. A. Bravo, and T. Brox, “Open-ended vqa benchmarking of vision-language models by exploiting classification datasets and their semantic hierarchy,” in *The Twelfth International Conference on Learning Representations (ICLR)*, 2024. Spotlight.
- [40] V. Mavi, A. Jangra, and A. Jatowt, “Multi-hop question answering,” *Found. Trends Inf. Retr.*, vol. 17, no. 5, p. 457–586, 2024.
- [41] D. T. Tran, T.-K. Tran, M. Hauswirth, and D. Le Phuoc, “Reasonvqa: A multi-hop reasoning benchmark with structural knowledge for visual question answering,” in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 18793–18803, 2025.
- [42] H. Trivedi, T. Khot, M. Hartmann, R. Manku, V. Dong, E. Li, S. Gupta, A. Sabharwal, and N. Balasubramanian, “AppWorld: A controllable world of apps and people for benchmarking interactive coding agents,” in *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 16022–16076, 2024.
- [43] Tableau Software, “Tableau public.” <https://www.tableau.com/products/public>, 2025. Accessed: 2025-06-02.
- [44] B. Ma, Y. Li, W. Zhou, Z. Gong, Y. J. Liu, K. Jasinskaja, A. Friedrich, J. Hirschberg, F. Kreuter, and B. Plank, “Pragmatics in the era of large language models: A survey on datasets, evaluation, opportunities and challenges,” in *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 8679–8696, 2025.
- [45] M. Zaib, W. E. Zhang, Q. Z. Sheng, A. Mahmood, and Y. Zhang, “Conversational

- question answering: a survey,” *Knowledge and Information Systems*, vol. 64, no. 12, p. 3151–3195, 2022.
- [46] A. Louis, D. Roth, and F. Radlinski, ““I’d rather just go to bed”: Understanding indirect answers,” in *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, (Online), pp. 7411–7425, Association for Computational Linguistics, Nov. 2020.
- [47] N. Kim, P. M. Htut, S. R. Bowman, and J. Petty, “(QA)<sup>2</sup>: Question answering with questionable assumptions,” in *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 8466–8487, 2023.
- [48] S. Sravanthi, M. Doshi, P. Tankala, R. Murthy, R. Dabre, and P. Bhattacharyya, “PUB: A pragmatics understanding benchmark for assessing LLMs’ pragmatics capabilities,” in *Findings of the Association for Computational Linguistics: ACL 2024*, pp. 12075–12097, 2024.
- [49] J. Weizenbaum, “Eliza—a computer program for the study of natural language communication between man and machine,” *Commun. ACM*, vol. 9, no. 1, p. 36–45, 1966.
- [50] E. Hoque, V. Setlur, M. Tory, and I. Dykeman, “Applying pragmatics principles for interaction with visual analytics,” *IEEE Transactions on Visualization and Computer Graphics*, vol. 24, no. 1, pp. 309–318, 2018.
- [51] K. Kafle and C. Kanan, “Visual question answering: Datasets, algorithms, and future challenges,” *Computer Vision and Image Understanding*, vol. 163, p. 3–20, 2017.

- [52] Y. Zhu, O. Groth, M. Bernstein, and L. Fei-Fei, “ Visual7W: Grounded Question Answering in Images ,” in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4995–5004, IEEE Computer Society, 2016.
- [53] Y. Zhang, Y. Su, Y. Liu, X. Wang, J. Burgess, E. Sui, C. Wang, J. Aklilu, A. Lozano, A. Wei, L. Schmidt, and S. Yeung-Levy, “Automated generation of challenging multiple-choice questions for vision language model evaluation,” in *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 29580–29590, 2025.
- [54] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu, “Bleu: a method for automatic evaluation of machine translation,” in *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics, ACL ’02*, p. 311–318, Association for Computational Linguistics, 2002.
- [55] C.-Y. Lin, “ROUGE: A package for automatic evaluation of summaries,” in *Text Summarization Branches Out*, pp. 74–81, Association for Computational Linguistics, 2004.
- [56] S. Banerjee and A. Lavie, “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*, pp. 65–72, Association for Computational Linguistics, 2005.
- [57] A. F. Biten, R. Tito, A. Mafla, L. Gomez, M. Rusiñol, C. Jawahar, E. Valveny, and D. Karatzas, “Scene text visual question answering,” in *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 4290–4300, 2019.

- [58] L. Yujian and L. Bo, “A normalized levenshtein distance metric,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 29, no. 6, pp. 1091–1095, 2007.
- [59] K. Kafle, B. Price, S. Cohen, and C. Kanan, “Dvqa: Understanding data visualizations via question answering,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 5648–5656, 2018.
- [60] Z. Zhu, M. Jia, Z. Zhang, L. Li, and M. Jiang, “MultiChartQA: Benchmarking vision-language models on multi-chart problems,” in *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pp. 11341–11359, 2025.
- [61] S. E. Kahou, V. Michalski, A. Atkinson, Á. Kádár, A. Trischler, and Y. Bengio, “Figureqa: An annotated figure dataset for visual reasoning,” *arXiv preprint arXiv:1710.07300*, 2017.
- [62] R. Chaudhry, S. Shekhar, U. Gupta, P. Maneriker, P. Bansal, and A. Joshi, “Leafqa: Locate, encode & attend for figure question answering,” in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pp. 3512–3521, 2020.
- [63] N. Methani, P. Ganguly, M. M. Khapra, and P. Kumar, “PlotQA: Reasoning over Scientific Plots,” in *2020 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pp. 1516–1525, IEEE Computer Society, 2020.
- [64] H. Singh and S. Shekhar, “Stl-cqa: Structure-based transformers with localization and encoding for chart question answering,” in *Proceedings of the 2020*

*Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 3275–3284, 2020.

- [65] Z. Wang, M. Xia, L. He, H. Chen, Y. Liu, R. Zhu, K. Liang, X. Wu, H. Liu, S. Malladi, A. Chevalier, S. Arora, and D. Chen, “Charxiv: Charting gaps in realistic chart understanding in multimodal llms,” in *Advances in Neural Information Processing Systems*, vol. 37, pp. 113569–113697, Curran Associates, Inc., 2024.
- [66] C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, and P. J. Liu, “Exploring the limits of transfer learning with a unified text-to-text transformer,” *J. Mach. Learn. Res.*, vol. 21, Jan. 2020.
- [67] P. Rajpurkar, J. Zhang, K. Lopyrev, and P. Liang, “SQuAD: 100,000+ questions for machine comprehension of text,” in *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pp. 2383–2392, Association for Computational Linguistics, Nov. 2016.
- [68] P. Pajo and Others, “Multi-agentic platforms: Architectures, applications, and emerging research frontiers in collaborative ai systems.” [https://www.researchgate.net/publication/392728233\\_Multi-Agentic\\_Platforms\\_Architectures\\_Applications\\_and\\_Emerging\\_Research\\_Frontiers\\_in\\_Collaborative\\_AI\\_Systems](https://www.researchgate.net/publication/392728233_Multi-Agentic_Platforms_Architectures_Applications_and_Emerging_Research_Frontiers_in_Collaborative_AI_Systems), 2025.
- [69] Y. Zhang, Z. Ma, Y. Ma, Z. Han, Y. Wu, and V. Tresp, “Webpilot: a versatile and autonomous multi-agent system for web task execution with strategic exploration,” in *Proceedings of the Thirty-Ninth AAAI Conference on Artificial Intelligence and Thirty-Seventh Conference on Innovative Applications of Artificial Intelligence and Fifteenth Symposium on Educational Advances in Artificial Intelligence*, AAAI’25/IAAI’25/EAAI’25, AAAI Press, 2025.

- [70] K. Goswami, P. Mathur, R. Rossi, and F. Deroncourt, “Chartcitor: Answer citations for chartqa via multi-agent llm retrieval,” WWW ’25, p. 1668–1671, Association for Computing Machinery, 2025.
- [71] M. Suri, P. Mathur, N. Lipka, F. Deroncourt, R. A. Rossi, and D. Manocha, “ChartLens: Fine-grained visual attribution in charts,” in *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 22447–22462, 2025.
- [72] J. Yang, H. Zhang, F. Li, X. Zou, C. Li, and J. Gao, “Set-of-mark prompting unleashes extraordinary visual grounding in gpt-4v,” *arXiv preprint arXiv:2310.11441*, 2023.
- [73] C. Park, Y. Baek, J. Kim, Y.-J. Heo, D.-S. Chang, and J. Choo, “Evaluating visual and cultural interpretation: The k-viscuit benchmark with human-VLM collaboration,” in *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 21960–21974, 2025.
- [74] Y. Xu, D. Lu, Z. Shen, J. Wang, Z. Wang, Y. Mao, C. Xiong, and T. Yu, “Agent-trek: Agent trajectory synthesis via guiding replay with web tutorials,” in *International Conference on Representation Learning*, vol. 2025, pp. 79822–79843, 2025.
- [75] V. Pahuja, Y. Lu, C. Rosset, B. Gou, A. Mitra, S. Whitehead, Y. Su, and A. H. Awadallah, “Explorer: Scaling exploration-driven web trajectory synthesis for multimodal web agents,” in *Findings of the Association for Computational Linguistics: ACL 2025*, pp. 6300–6323, 2025.
- [76] N. Boukhelifa, M.-E. Perrin, S. Huron, and J. Eagan, “How data workers cope

with uncertainty: A task characterisation study,” in *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*, CHI ’17, p. 3645–3656, Association for Computing Machinery, 2017.

- [77] J. Liu, N. Boukhelifa, and J. R. Eagan, “Understanding the role of alternatives in data analysis practices,” *IEEE Transactions on Visualization and Computer Graphics*, vol. 26, no. 1, pp. 66–76, 2020.
- [78] J. Kocoń, I. Cichecki, O. Kaszyca, M. Kochanek, D. Szydło, J. Baran, J. Bielaniewicz, M. Gruza, A. Janz, K. Kanclerz, A. Kocoń, B. Kopyra, W. Mieleśczenko-Kowszewicz, P. Milkowski, M. Oleksy, M. Piasecki, Radliński, K. Wojtasik, S. Woźniak, and P. Kaziienko, “Chatgpt: Jack of all trades, master of none,” *Information Fusion*, vol. 99, p. 101861, 2023.
- [79] United Nations Office on Drugs and Crime, Research and Analysis Branch, “Unodc rab dashboards on tableau public.” <https://public.tableau.com/app/profile/unodc.rab/vizzes>, 2024. Accessed: April 2025.
- [80] Economic Research Service, “Economic research service dashboards on tableau public.” <https://public.tableau.com/app/profile/economic.research.service/vizzes>, 2024. Accessed: April 2025.
- [81] L. Ying, K. M. Collins, L. Wong, I. Sucholutsky, R. Liu, A. Weller, T. Shu, T. L. Griffiths, and J. B. Tenenbaum, “On benchmarking human-like intelligence in machines,” *arXiv preprint arXiv:2502.20502*, 2025.
- [82] M. Bartolo, T. Thrush, R. Jia, S. Riedel, P. Stenetorp, and D. Kiela, “Improving question answering model robustness with synthetic adversarial data generation,”

in *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, Association for Computational Linguistics, 2021.

- [83] MDN Contributors, “Accessibility tree — glossary — mdn.” [https://developer.mozilla.org/en-US/docs/Glossary/Accessibility\\_tree](https://developer.mozilla.org/en-US/docs/Glossary/Accessibility_tree), 2025. Last modified July 11, 2025; Accessed September 18, 2025.
- [84] F. Liu, W. Wang, Y. Zhu, X. Wang, Y. Li, and H. Li, “A survey on hallucination in large vision-language models,” *arXiv preprint arXiv:2402.00253*, 2024.
- [85] P. Lewis, E. Perez, A. Piktus, F. Petroni, V. Karpukhin, N. Goyal, H. Küttler, M. Lewis, W.-t. Yih, T. Rocktäschel, S. Riedel, and D. Kiela, “Retrieval-augmented generation for knowledge-intensive nlp tasks,” in *Proceedings of the 34th International Conference on Neural Information Processing Systems, NIPS ’20*, Curran Associates Inc., 2020.
- [86] M. Sanwal, “Layered chain-of-thought prompting for multi-agent llm systems: A comprehensive approach to explainable large language models,” *arXiv preprint arXiv:2501.18645*, 2025.
- [87] M. T. R. Laskar, M. S. Islam, R. Mahbub, A. Masry, M. Rahman, A. Bhuiyan, M. T. Nayeem, S. Joty, E. Hoque, and J. Huang, “Judging the judges: Can large vision-language models fairly evaluate chart comprehension and reasoning?,” in *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 6: Industry Track)*, pp. 1203–1216, 2025.
- [88] M. Z. I. Alam, S. Islam, and E. Hoque, “Seechart: Enabling accessible visualizations through interactive natural language interface for people with visual

impairments,” in *Proceedings of the 28th International Conference on Intelligent User Interfaces*, IUI '23, p. 46–64, ACM, 2023.

# Appendices

## A Prompts

Table 1: Examples of Prompt Categories in DashboardQA for single turn question single dashboard questions

---

Category	Example Prompt
<b>Factoid</b>	I am giving you multiple snapshots/views of an interactive dashboards, including of the respective default views, options available for navigation through the dashboard’s navigation tools (i.e Dropdown menus, radio buttons, scrollers, etc.), and more views from when particular options are selected. Since not every possible snapshot / view of each dashboard is given, the selection options available and the example views for each dashboard should together help you guess how the views for the rest of the options would be structurally (i.e types and content of the charts or textual data available) for that particular dashboard. Using all of these, please generate 5 diverse and challenging factoid reasoning questions that involve arithmetic and logical reasoning. They must have multiple operators (i.e sum, ratio, etc.) in the question. Additionally, make sure the answers required are succinct, where they are either a numerical value, True or False, Yes or No, unanswerable or simply a label taken from the dashboard itself. Each question must require 3 to 5 views (Cumulative) of the dashboards to answer. Just the questions are to be given now from your end, in the next prompt the required views will be given for you to answer.
<b>Hypothetical</b>	I am giving you multiple snapshots/views of an interactive dashboards, including of the respective default views, options available for navigation through the dashboard’s navigation tools (i.e Dropdown menus, radio buttons, scrollers, etc.), and more views from when particular options are selected. Since not every possible snapshot / view of each dashboard is given, the selection options available and the example views for each dashboard should together help you guess how the views for the rest of the options would be structurally (i.e types and content of the charts or textual data available) for that particular dashboard. Using all of these, please generate 5 diverse and challenging hypothetical questions that involve arithmetic and logical reasoning. This may involve looking into future trends, extrapolation tasks, or looking at counterfactuals for example. They must have multiple operators (i.e sum, ratio, etc.) in the question. Additionally, make sure the answers required are succinct, where they are either a numerical value, True or False, Yes or No, unanswerable or simply a label taken from the dashboard itself. Each question must require from 2 to 5 views of the dashboard to answer. Just the questions are to be given now from your end, in the next prompt the required views will be given for you to answer.
<b>MCQ</b>	I am giving you multiple snapshots/views of an interactive dashboard, including of the respective default views, options available for navigation through the dashboard’s navigation tools (i.e Dropdown menus, radio buttons, scrollers, etc.), and more views from when particular options are selected. Since not every possible snapshot / view of each dashboard is given, the selection options available and the example views for each dashboard should together help you guess how the views for the rest of the options would be structurally (i.e types and content of the charts or textual data available) for that particular dashboard. Using all of these, please generate 5 diverse and challenging MCQ reasoning questions that involve arithmetic and logical reasoning. There may be 4-5 answer options available, but only one of them is correct. They must have multiple operators (i.e sum, ratio, etc.) in the question. Additionally, make sure the answers required are succinct, where they are either a numerical value, True or False, Yes or No, unanswerable or simply a label taken from the dashboard itself. Each question must require 2 to 5 views of the dashboard to answer. Just the questions are to be given now from your end, in the next prompt the required views will be given for you to answer. Remember to not be biased towards just the views given, and to reason as to how the others would look like as to help your questions.

---

Table 2: Examples of Prompt Categories in DashboardQA involving multiple dashboards or multi-turn question answering

Category	Example Prompt
<b>Multi-dashboard</b>	<p>I am giving you multiple snapshots/views respectively for multiple interactive dashboards, including of their respective default views, options available for navigation through the dashboard’s navigation tools (i.e Dropdown menus, radio buttons, scrollers, etc.), and more views from when particular options are selected. Since not every possible snapshot / view of each dashboard is given, the selection options available and the example views for each dashboard should together help you guess how the views for the rest of the options would be structurally (i.e types and content of the charts or textual data available) for that particular dashboard.</p> <p>Using all of these, please generate 5 diverse and high level questions surrounding all of the dashboards given that involve arithmetic and logical reasoning. This could include having multiple operators (i.e sum, ratio, etc.) in the question. Additionally, make sure the answers required are succinct, where they are either a numerical value, True or False, Yes or No, unanswerable or simply a label taken from the dashboard itself. Each question must require from 2 to 5 views of the dashboard to answer, and a majority of the follow up questions should require utilizing one or more of the previous questions to help. Just the questions are to be given now from your end, in the next prompt the required views will be given for you to answer.</p> <p>Here are some sample questions to give some inspiration, but make sure to create the questions with lexical and semantic differences.</p>
<b>Conversational</b>	<p>I am giving you multiple snapshots/views of an interactive dashboard, including the default view, the options available for navigation through the dashboard’s navigation tools (i.e Dropdown menus, radio buttons, scrollers, etc.), and more views from when particular options are selected. Since not every possible snapshot / view are going to be given, the selection options available and the example views should together help you guess how the views for the rest of the options would be structurally (i.e types and content of the charts or textual data available for that particular dashboard).</p> <p>Using all of these, please generate a conversation of 4-7 questions that analyzes the contents of the dashboard and requires multiple operations (i.e sum, ratio etc.). With a focus on arithmetic and logical reasoning, make sure that a majority of the questions follow up on each other. Use ambiguous pronouns in the latter questions when making references to previous questions. Additionally, make sure the answers required are succinct, where they are either a numerical value, True or False, Yes or No, or simply a label taken from the dashboard itself. The conversation must require from 2 to 5 views of the conversation to answer, where each individual question can either require just one to three views to answer respectively. Just the questions are to be given now from your end, in the next prompt the required views will be given for you to answer.</p> <p>Here are some sample conversations to give some inspiration, but make sure to create the conversations with different lexical and semantic differences. The respective dashboard images are not given for these samples, they are just to get an idea (Not purely representative) of the style of conversations.</p>

# B Sample Dashboards

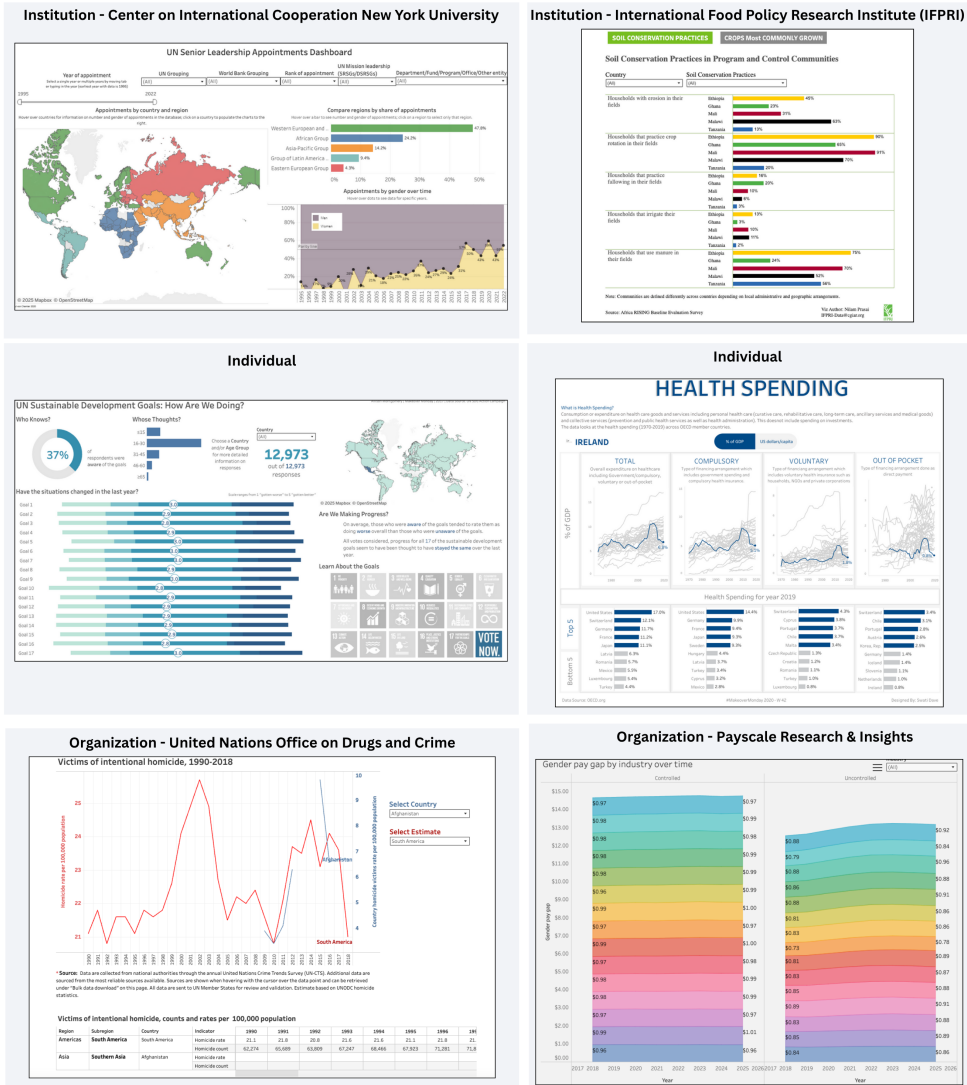


Figure 1: Example default views of Tableau dashboards collected from various institutions, organizations, and individuals.

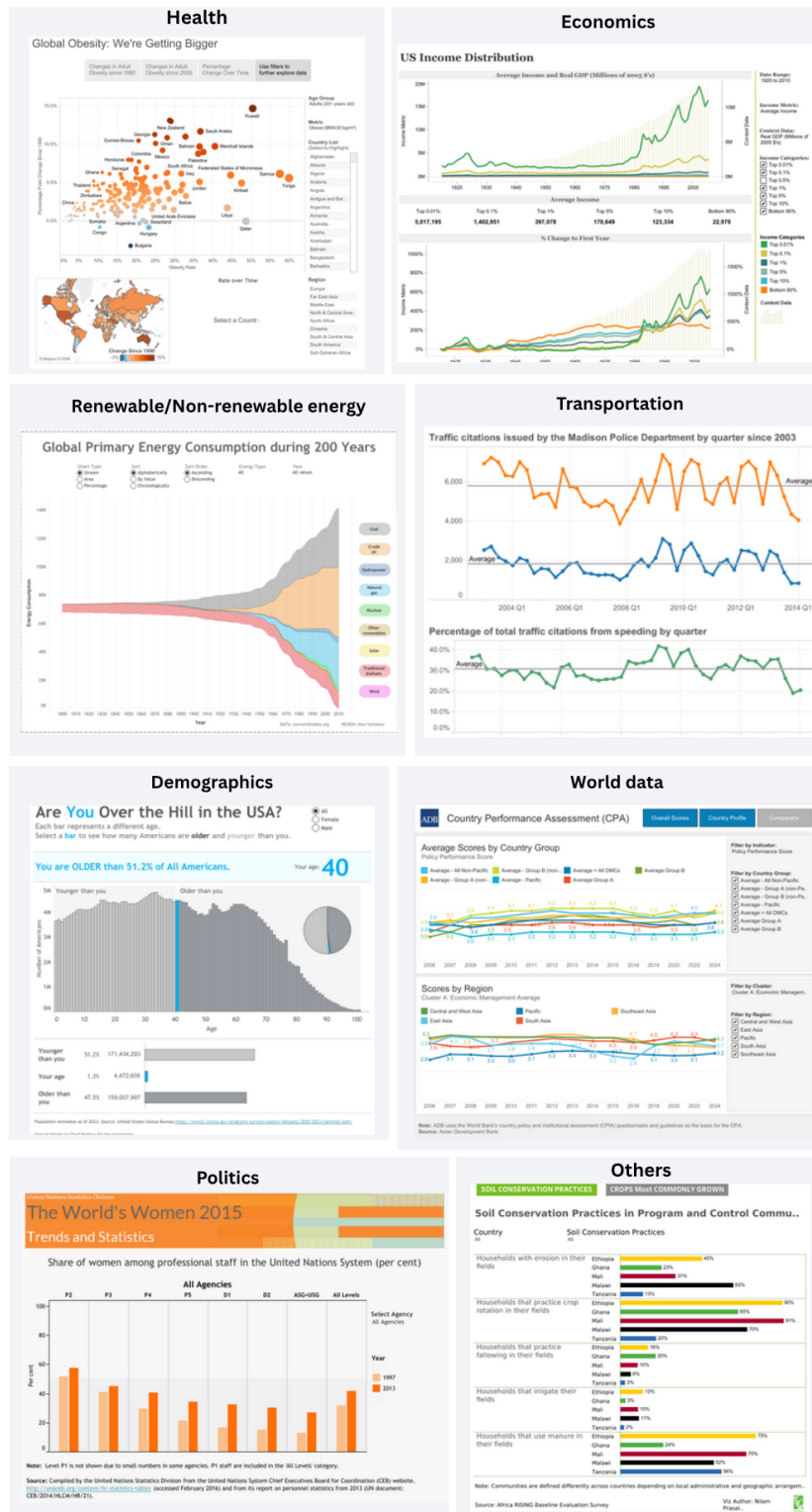


Figure 2: Sample dashboards chosen for all of the topics in DashboardQA

## C Sample Questions and Answer Workflows

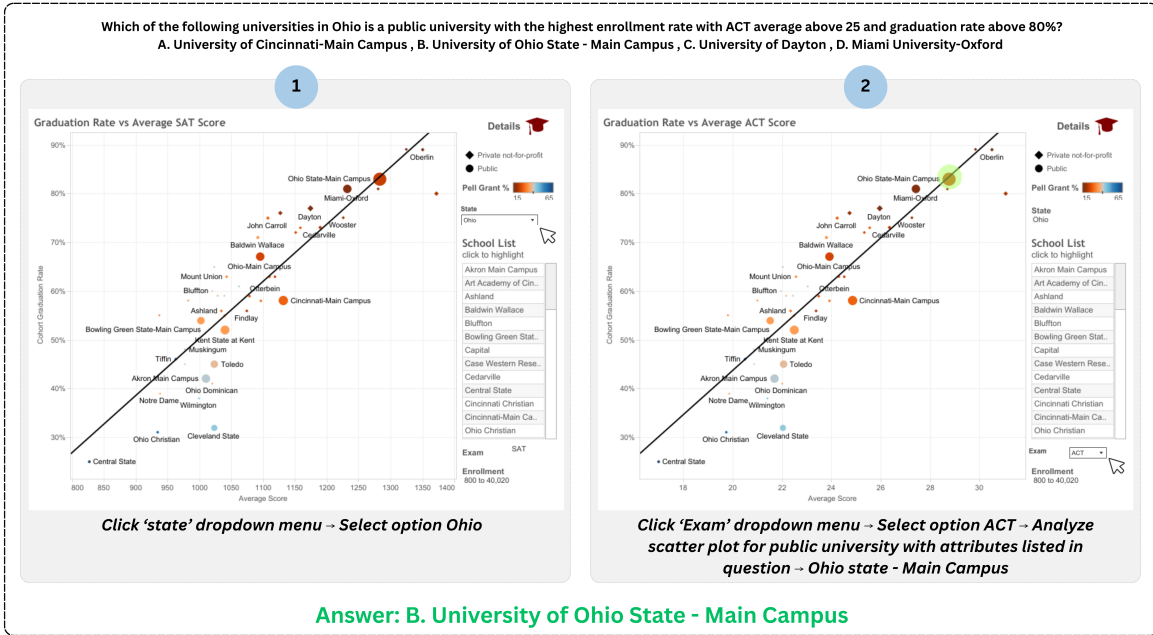
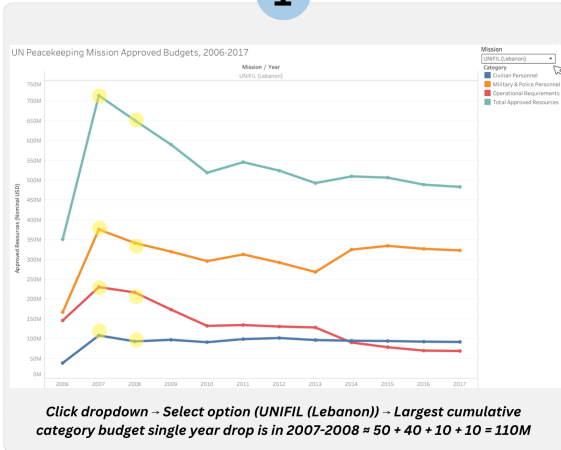


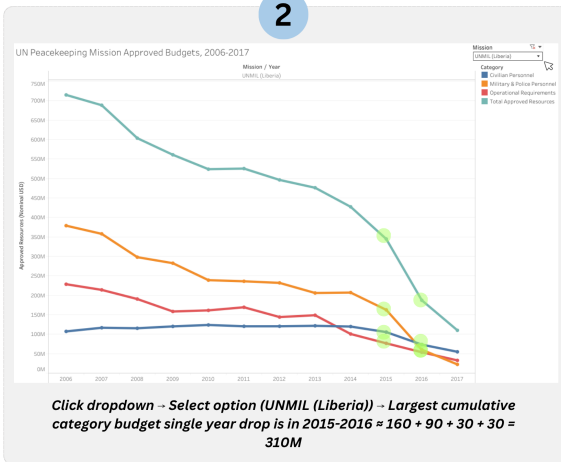
Figure 3: Example MCQ question and task workflow in DashboardQA

Among UNIFIL (Lebanon), UNMIL (Liberia), and UNOMIG (Georgia), which mission had the steepest single-year budget drop?

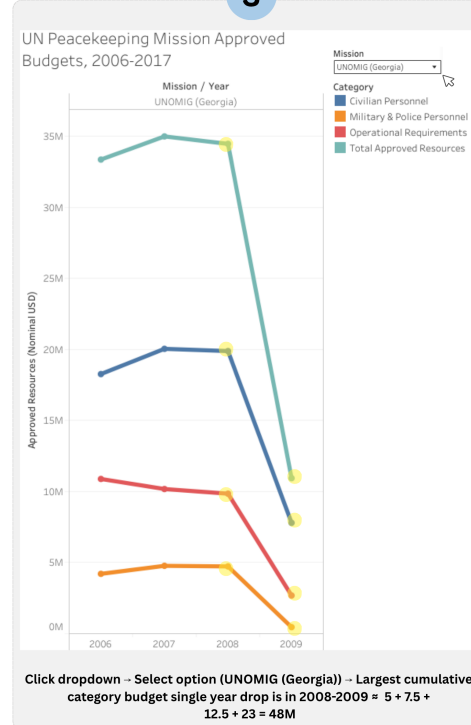
1



2



3



Answer: UNMIL (Liberia)

Figure 4: Example factoid question and task workflow in DashboardQA

## D Sample Errors, Breakdowns, and Workaround Strategies

### Planning and perceptual failure

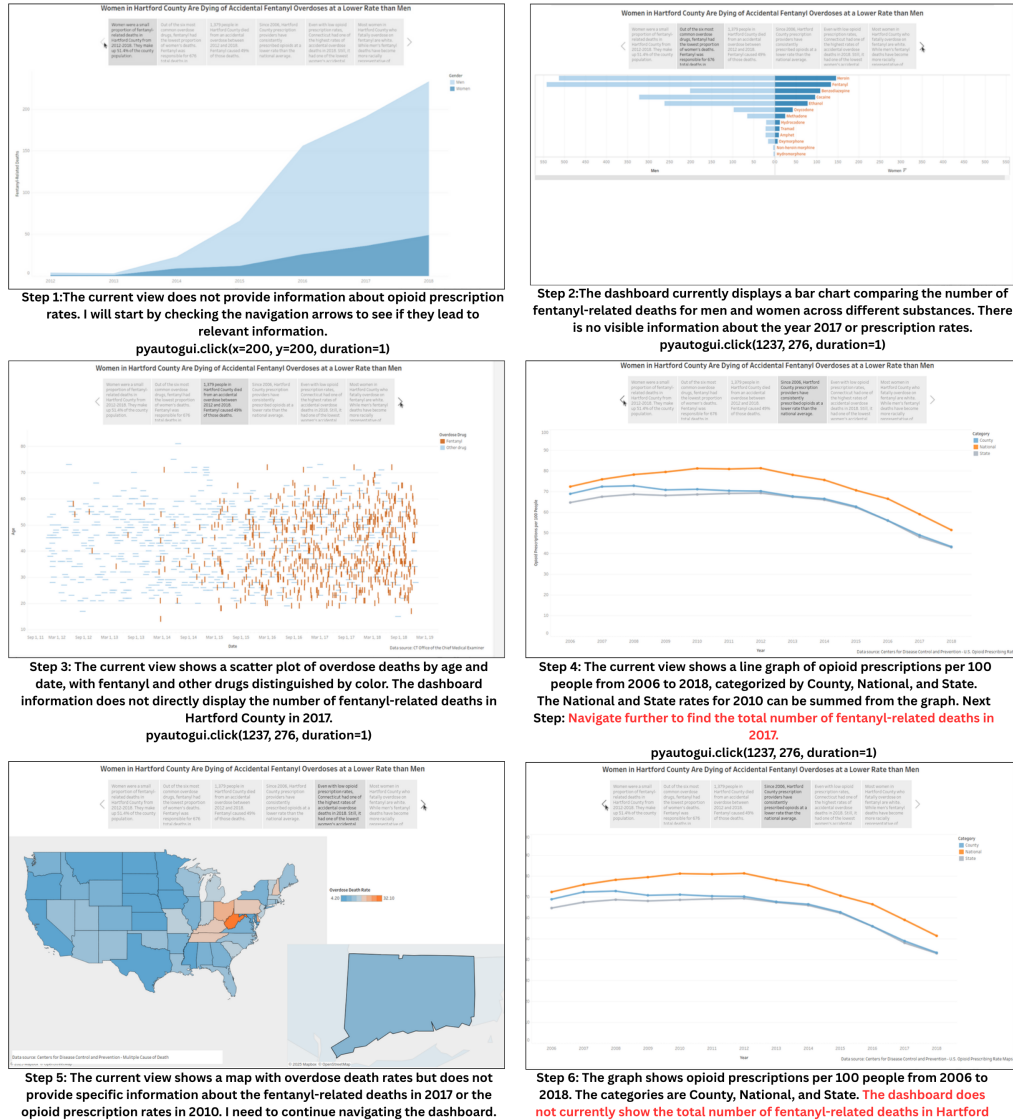
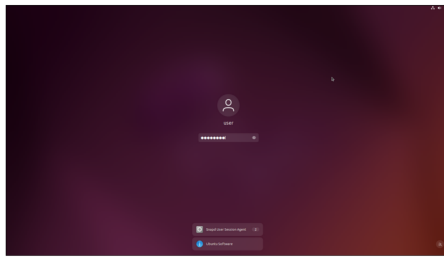
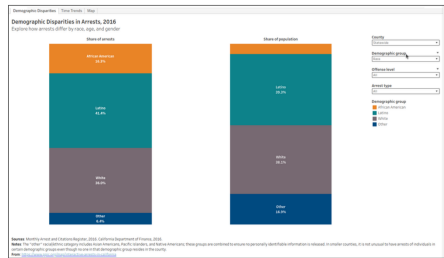


Figure 5: Example planning failure breakdown seen. Jedi is unable to utilize visual tab titles to directly go to required state, instead inefficiently sequentially moving along the tabs. Additionally, perceptual failure is seen when the VLM is unable to realize they have traversed through the right state.

## Environment breakdowns

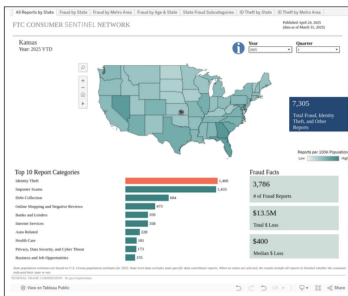


I'm unable to complete the task as the screen is currently locked. I need to log in to access the dashboard.  
`pyautogui.typewrite('password', interval=0.1)`

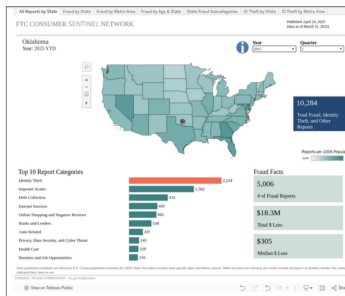


After logging in, I will navigate to the Tableau dashboard.

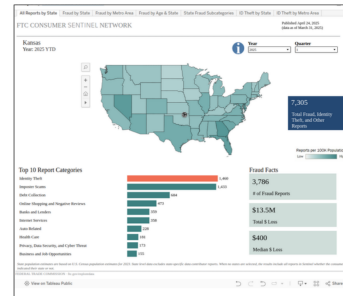
## Visualization usage failure



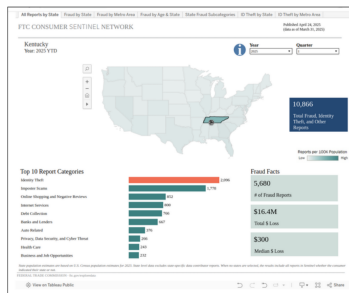
Step 2: The map is currently showing data for a different state, not Alabama. I need to select Alabama to view its specific data.  
`pyautogui.click(751, 465, duration=1)`



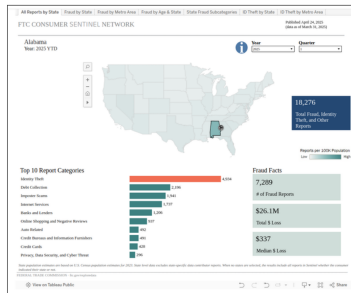
Step 4: I need to correctly select Alabama to view its specific fraud data  
`pyautogui.click(743, 496, duration=1)`



Step 6: The map is currently focused on Kansas. I need to select Alabama to view its specific fraud data.  
`pyautogui.click(759, 476, duration=1)`



Step 8: I need to correctly select Alabama on the map to view its specific fraud data.  
`pyautogui.click(841, 496, duration=1)`



Step 10: I need to accurately select Alabama on the map to view its specific fraud data.  
`pyautogui.click(867, 518, duration=1)`

Figure 6: Examples environment breakdown and visualization tool usage failure

## E Additional breakdown workaround mapping quantification

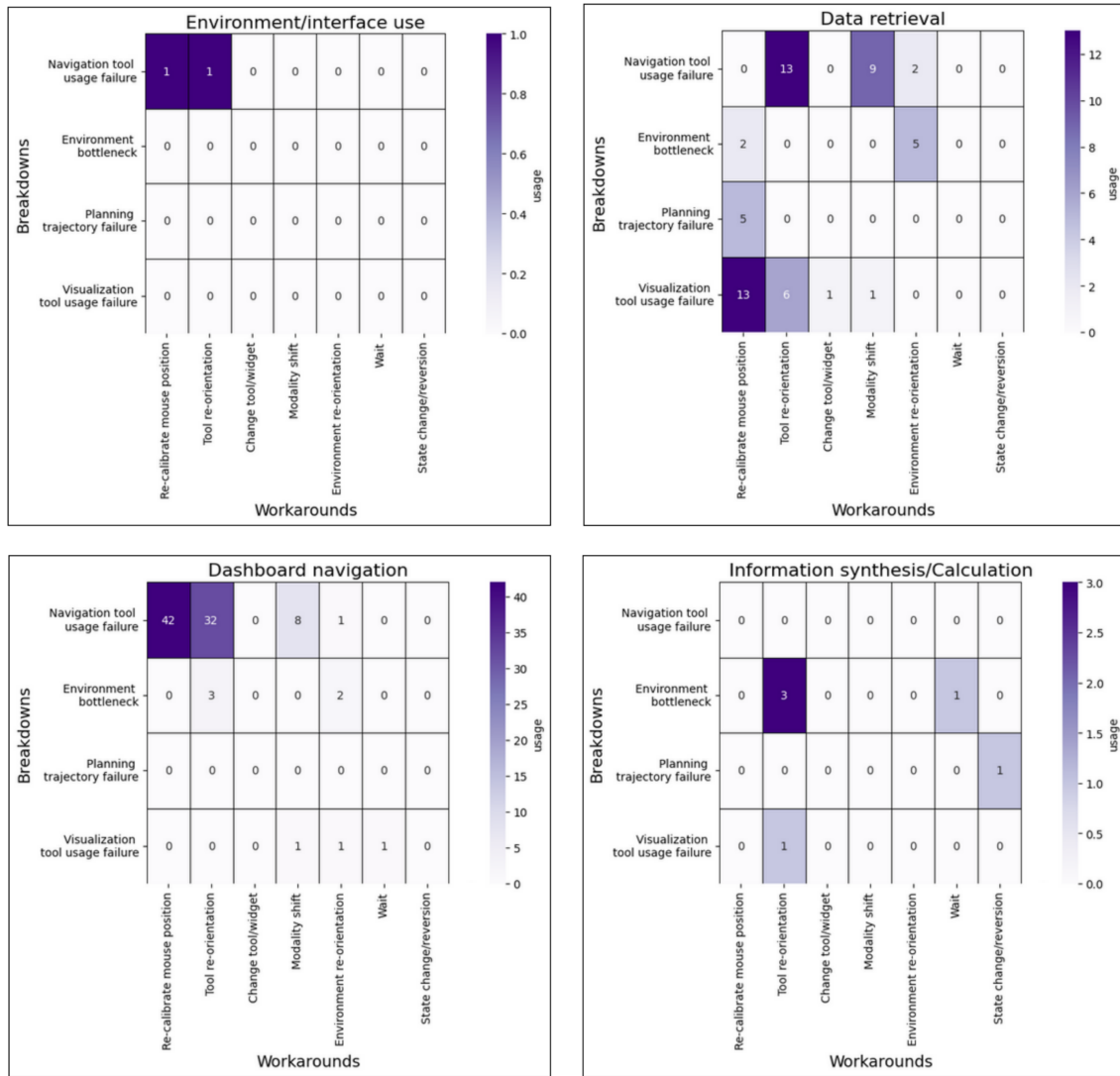


Figure 7: Heatmap representing breakdown-workaround pairs found for JEDI task executions

## **F Code for breakdown and associated workaround identification**

The script for breakdown/workaround identification and quantification is released at:  
[https://github.com/aaryamankartha/VLM-reasoning-open-coding/blob/main/Agent\\_breakdown\\_quantification.ipynb](https://github.com/aaryamankartha/VLM-reasoning-open-coding/blob/main/Agent_breakdown_quantification.ipynb)