Unmasking Deceptive Visuals: Benchmarking Multimodal Large Language Models on Misleading Chart Question Answering

Anonymous ACL submission

Abstract

Misleading chart visualizations, which intentionally manipulate data representations to support specific claims, can distort perceptions and lead to incorrect conclusions. Despite decades of research, misleading visualizations remain a widespread and pressing issue. Recent advances in multimodal large language models (MLLMs) have demonstrated strong chart comprehension capabilities, yet no existing work has systematically evaluated their ability to detect and interpret misleading charts. This paper introduces the Misleading Chart Question Answering (Misleading ChartQA) Benchmark, a large-scale multimodal dataset designed to assess MLLMs in identifying and reasoning about misleading charts. It contains over 3,000 curated examples, covering 21 types of misleaders and 10 chart types. Each example includes standardized chart code, CSV data, and multiple-choice questions with labeled explanations, validated through multi-round MLLM checks and exhausted expert human review. We benchmark 16 state-of-the-art MLLMs on our dataset, revealing their limitations in identifying visually deceptive practices. We also propose a novel pipeline that detects and localizes misleaders, enhancing MLLMs' accuracy in misleading chart interpretation. Our work establishes a foundation for advancing MLLM-driven misleading chart comprehension. We publicly release the sample dataset to support further research in this critical area[1].

1 Introduction

Misleading visualizations have long been a significant concern in chart comprehension and data communication (Tufte and Graves-Morris, 1983). As early as the 1950s, the influential book *How to Lie with Statistics* exposed how selectively constructed charts can distort data representations to manipulate public perception (Huff, 2023). Despite growing awareness, misleading chart designs continue to have a profound impact today. For instance, in 2020, the Georgia Department of Public Health published a bar chart of COVID-19 cases across five counties, sorting the data from highest to lowest rather than by date (fig. 6 A). This misleading arrangement falsely suggested a downward trend in case numbers (McFall-Johnsen, 2020). Another widely recognized example is the standard world map (fig. 6 B). Many people are unaware that the commonly used Mercator Projection distorts the relative sizes of countries, exaggerating landmasses near the poles while shrinking those near the equator (Kennedy et al., 2000; O'Brien, 2024). These real-world cases demonstrate how charts can ma-

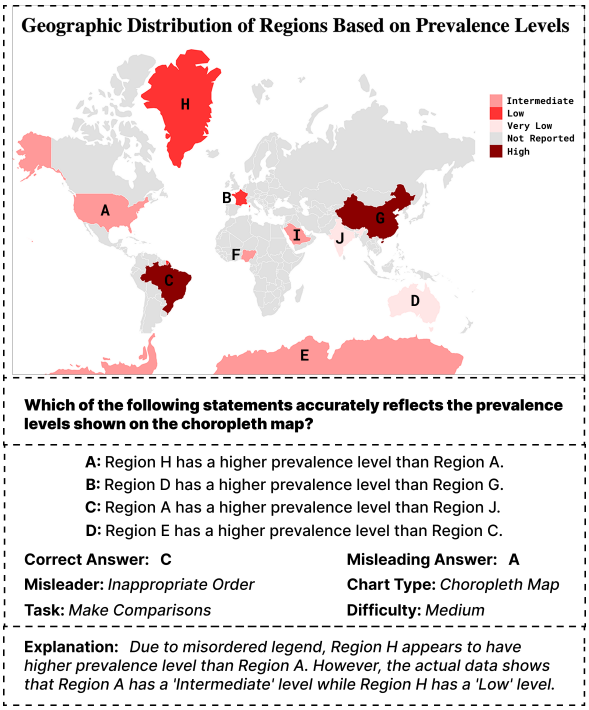


Figure 1: An example multiple-choice question (MCQ) from our benchmark. Each MCQ includes a misleading chart, a question, multiple answer options, the correct answer and a set of labels. A detailed explanation is also provided to illustrate the chart's misleading aspects.

[1]https://anonymous.4open.science/r/Misleading-ChartQA/

nipulate perception and mislead audiences, leading to significant consequences.

Recent advances in multimodal large language models (MLLMs) have demonstrated strong capabilities in chart comprehension tasks, like chart question answering(Xia et al., 2024; Masry et al., 2022), chart captioning(Huang et al., 2023; Rahman et al., 2023), and chart extraction (Chen et al., 2024a). However, existing studies primarily focus on factual tasks such as information extraction, and summarization, leaving the critical challenge of evaluating how well MLLMs can detect and interpret complex misleading charts largely unexplored.

To bridge this gap, we introduce the Misleading Chart Question Answering (Misleading ChartQA) Benchmark, a large-scale multimodal dataset designed to evaluate MLLMs' ability to identify and interpret misleading chart visualizations. Our work builds upon theoretical foundations that classify misleading visualization features (misleaders) (Börner et al., 2019; Lo et al., 2022; Lan and Liu, 2024) and standardized multiple-choice question (MCQ) tests used to assess human interpretation of misleading charts (Lee et al., 2016; Cui et al., 2023; Ge et al., 2023).

We collaborated with data visualization experts to develop a comprehensive misleader taxonomy (fig. 2), expanding the design space to 60 unique (misleader, chart type) pairs, covering 21 misleaders and 10 chart types (fig. 7). For each (misleader, chart type) combination, experts carefully crafted seed MCQs with detailed labels and explanations (fig. 1). These were then transformed into standardized D3.js (Bostock et al., 2011) visualizations, with corresponding CSV data and JSON files with labels. Through automated expansion and iterative human verification, we constructed a high-quality benchmark featuring over 3,000 unique misleading chart MCQs. Evaluating 16 state-of-the-art MLLMs, we found significant limitations in their ability to interpret misleading charts, underscoring the urgent need for improvement. Furthermore, we designed and evaluated a novel approach, Region-Aware Misleader Reasoning, which enhances MLLMs' performance on this complex task. In summary, our main contributions are as follows:

(i) We construct the Misleader Taxonomy, systematically summarizing and categorizing prevailing misleaders across multiple chart types.

(ii) We introduce a large-scale dataset with over 3,000 curated samples, covering 10 chart types, 21 misleaders, and 60 (misleader, chart type) pairs.

(iii) We conduct extensive assessment and analysis of 16 leading MLLMs, measuring their ability to interpret misleaders. Our evaluation highlights their limitations and key areas for future research.

(iv) We propose Region-Aware Misleader Reasoning, a novel method designed to enhance MLLMs' ability on the proposed task. We also explore future directions to improve MLLMs for more effective and reliable chart comprehension.

2 Misleading ChartQA Benchmark

In this section, we outline the construction process of the Misleading ChartQA dataset, which consists of four main stages:(1) Misleader Taxonomy Construction, (2) Seed MCQ Design, (3) Automated Expansion and Iterative Refinement, and (4) Human Evaluation and Final Refinement.

2.1 Misleader Taxonomy Construction

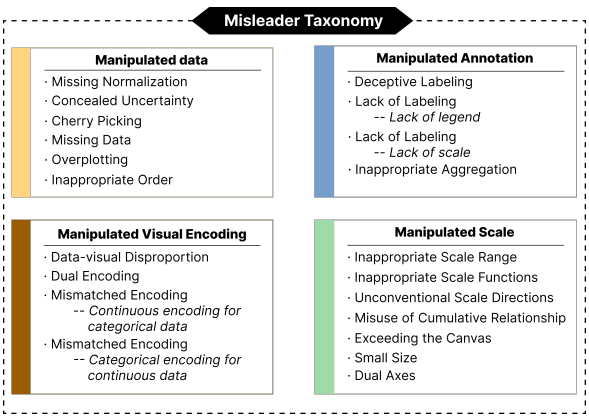


Figure 2: The proposed Misleader Taxonomy, classifying 21 identified misleaders into four main categories based on distinct chart manipulation techniques.

To capture the diverse ways in which visualizations can mislead readers, we constructed a Misleader Taxonomy by consolidating known deceptive strategies from both the literature and real-world examples (Lo et al., 2022; Börner et al., 2019; Lan and Liu, 2024). Four data visualization experts—including two postdoctoral researchers and two senior PhD students—independently reviewed these sources to compile an initial list of common misleaders. Through collective discussion, they refined the list by merging overlapping items, clarifying ambiguous definitions, and removing overly narrow cases, ultimately identifying 21 distinct misleader types. Next, our experts iteratively mapped each misleader to the most relevant chart types

where these deceptive tactics are most commonly observed. This process resulted in a final set of 10 unique chart types and 60 distinct (misleader, chart type) pairings, ensuring broad and representative coverage within our benchmark dataset. A comprehensive definition of each misleader, along with its corresponding chart type mappings, is provided in fig. 7. Finally, we structured these misleaders into the Misleader Taxonomy (fig. 2), establishing clear guidelines for subsequent data expansion.

2.2 Seed Multiple-Choice Question Design

Building on our Misleader Taxonomy and the 60 (misleader, chart type) pairings, we continued to collaborate with the experts to evaluate reference questions from previous studies (Lee et al., 2016; Cui et al., 2023; Ge et al., 2023) and design new misleading chart QA questions for the uncovered (misleader, chart type) pairs. Through multiple rounds of discussion and refinement, we established an initial set of "seed questions". As shown in fig. 1, each seed question includes: (1) a misleading visualization, (2) a corresponding question, (3) answer choices, (4) correct and misleader answers, and (5) metadata such as chart type, task, difficulty, and an explanation of the misleading aspects. Once all seed questions were finalized, we transformed them into a standardized format, comprising:

- **Misleading Chart Code Implementation.** To ensure flexibility in generating misleading chart visualizations and their variations, we implemented each chart in the seed question using D3.js (Bostock et al., 2011), a JavaScript library for highly customizable data visualizations. The code was structured in formatted HTML to facilitate easy rendering while maintaining a consistent coding style, optimized for generating variations.
  CSV Data and JSON QA Specification. Alongside the D3.js code, we curated CSV datasets for each chart, aligning them with misleader scenarios. For example, a scatter plot with the *Cherry Picking* misleader might display a selective data subset to exaggerate a trend. We also converted multiple-choice questions with labeled metadata into a JSON format for future compatibility.
- **Chart Figure Generation.** Using the code and data, we rendered charts and built a labeling tool for experts to annotate misleading areas with bounding boxes. Labeled and raw JPEG images were then exported with standardized dimensions for consist expansion.

These carefully designed seed examples ensure each misleader–chart type pair is matched with a high-quality, well-formatted MCQ, enabling seamless dataset expansion.

2.3 MCQs Expansion and Iterative Refinement

Using seed MCQs for each misleader–chart type pair, we leverage MLLMs to expand our dataset through a novel workflow that generates MCQ variations while preserving misleading features. The next section outlines the core structure of this workflow, with detailed prompt templates for each MLLM component provided in appendix A.7.1.

For each seed question, the annotated chart image, code, data, and JSON QA specification serve as core inputs to the MLLM module. We use the ChatGPT-4o model for its strong performance and efficiency. The question expansion process consists of two main steps: *Chart Variation* and *QA Generation*, followed by an *Automated Evaluation, Feedback, and Refinement Loop* for quality control.

- **Chart Variation:** In the first stage (fig. 3-A), we modify the chart code and dataset. MLLMs perturb the seed code by adjusting color scales, layout, and contextual features. The CSV dataset is also altered by modifying numeric values and category names while ensuring a similar data distribution. This process generates new instances that retain the original misleader but present it in varied formats.
- **QA Generation:** After modifying the CSV data and chart code, our workflow (fig. 3-B) automatically launches a web server to render and capture a screenshot of the generated chart. This image, along with the seed question chart and QA specification, is then passed to the next MLLM module for QA generation, ensuring the new question remains aligned with the original misleader.
- **Automated Evaluation, Feedback, and Refinement Loop:** To assess the quality of the generated content, the newly generated QA specification and chart image are evaluated by an additional MLLM module (fig. 3-B). This module verifies whether the question-chart pair accurately reflects the intended misleader. If the generated MCQ fails, the system provides targeted refinement instructions for the chart code, dataset, and QA specification, which are fed back into the generation mod-

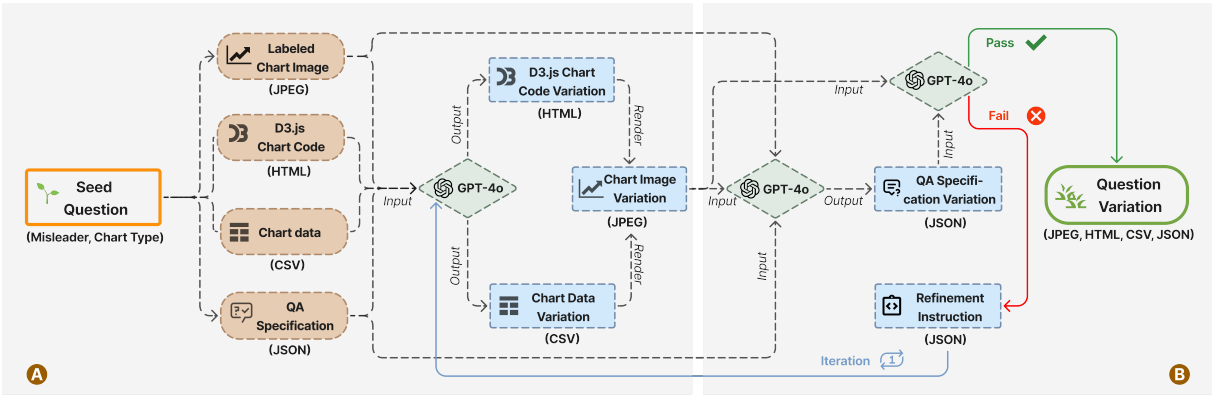


Figure 3: The proposed Automated MCQ Expansion and Iterative Refinement workflow. (A) The *Chart Variation* component, which takes the seed question elements as input and utilizes GPT-4o to modify the code and data while preserving the intended misleader. (B) *QA Generation* and *Automated Evaluation, Feedback, and Refinement Loop*. A separate GPT-4o module generates misleading QAs and explanations for the generated chart, while an evaluator module assesses the outputs and provides revision feedback on code, data, and QA aspects for failed cases.

ule for another iteration. This cycle repeats until the generated variations meet the evaluation criteria. Finally, domain experts manually review the output to ensure quality, refine explanations, and remove problematic samples.

2.4 External Human Evaluation

While automated evaluation and expert review eliminate most inconsistencies, external human evaluators add an additional layer of scrutiny. We recruited twenty PhD students specializing in data visualization to manually examine the dataset, ensuring all variations remained aligned with the curated seed exemplars. Evaluators were compensated at 30 USD per hour, and any erroneous samples were flagged for removal. This process resulted in a high-quality dataset of $3,026$ MCQs, along with corresponding code, data, QA specifications, and labeling metadata. A detailed dataset composition and a diversity comparison with existing benchmarks are provided in table 3.

3 Experiments and Evaluation

In this section, we present a comprehensive evaluation of 16 state-of-the-art MLLMs on our Misleading ChartQA dataset.

3.1 Experimental Setup

Our experiments were conducted on 8 NVIDIA A800 GPUs (80GB each) using PyTorch 2 and Python 3. Given the task's complexity, we selected only the most advanced versions of each model type and evaluated them across different parameter sizes. Due to computational constraints, we randomly sampled 20% (605 cases) from the dataset, ensuring a balanced distribution across misleader and chart types for representativeness. The following sections address five key research questions:

(RQ1) How do different MLLMs perform on the Misleading ChartQA task?

(RQ2) How effectively do MLLMs identify misleaders, and to what extent do they fall into "traps"?

(RQ3) How do misleader types and chart types influence MLLM performance?

(RQ4) What are the common failure patterns?

(RQ5) What strategies can enhance MLLMs' ability to identify and comprehend misleading ChartQA questions?

3.2 Overall Results (RQ1-RQ2)

The overall results are presented in table 1, from which we can make the following observations:

(1) **The Misleading ChartQA task presents a significant challenge**, with the best-performing model achieving only 47.95% baseline accuracy. This contrasts sharply with other chart-related QA benchmarks, where state-of-the-art models typically reach around 90% accuracy. Prior research on the general public's performance reports a similar accuracy, averaging 39% (SD = 16%) (Ge et al., 2023). This suggests that MLLMs trained on general corpora lack sufficient exposure to misleading chart data, underscoring the need for a dedicated corpus tailored to misleading chart comprehension.

(2) **Most models exhibit similar rates of "falling into the trap" relative to their overall**

	Model	Baseline			Zero-shot CoT			Pipeline		
		W. O.	W. M.	Acc.	W. O.	W. M.	Acc.	W. O.	W. M.	Acc.
	Random Guess	49.24	25.38	25.38	49.24	25.38	25.38	49.24	25.38	25.38
Closed Source	GPT-4o	26.65	38.44	34.91	25.56	37.75	36.69	27.72	33.20	39.08
	GPT-o1	30.04	35.57	34.39	24.43	37.44	38.13	23.29	34.02	**42.69**
	Claude-3.5-Sonnet	36.30	29.54	34.16	27.64	35.34	37.02	25.80	35.90	38.30
	Gemini-2.0-Flash	43.48	25.47	31.05	47.03	18.04	34.93	42.61	20.73	36.66
Open Source	DeepSeek-VL2-Tiny	28.54	40.57	30.90	32.88	37.90	29.22	31.74	45.89	22.37
	DeepSeek-VL2-Small	26.65	43.63	29.72	34.70	33.33	31.96	27.40	43.15	29.45
	DeepSeek-VL2	26.48	43.61	29.91	30.37	34.70	34.93	24.43	38.64	36.02
	Qwen2.5-VL-3B	35.14	30.66	34.20	36.99	29.22	33.79	34.70	27.63	37.67
	Qwen2.5-VL-7B	27.40	34.93	37.67	29.22	33.11	37.67	27.63	31.74	40.64
	Qwen2.5-VL-72B	29.48	29.48	41.04	28.77	28.77	42.47	31.51	25.11	43.38
	InternVL2.5-1B-MPO	64.16	14.38	21.46	98.86	0.68	0.46	85.71	0.00	14.29
	InternVL2.5-2B-MPO	29.68	37.90	32.42	31.05	38.81	30.14	30.82	34.93	34.25
	InternVL2.5-4B-MPO	24.20	39.73	36.07	28.77	33.33	37.90	26.48	36.07	37.44
	InternVL2.5-8B-MPO	19.86	38.36	41.78	22.61	34.70	42.69	18.72	36.53	44.75
	InternVL2.5-26B-MPO	20.78	36.76	42.47	29.25	29.68	41.07	18.49	38.81	42.69
	InternVL2.5-78B-MPO	20.09	31.96	**47.95**	16.89	36.76	46.35	18.95	32.88	**48.77**

Table 1: Summary of experimental results. (A) For the Baseline, we used zero-shot prompt settings aligned with the model publishers' configurations for related chart benchmarks (Chen et al., 2024b). (B) The Zero-shot CoT experiments evaluated the impact of prompt strategies while maintaining baseline alignment, showing performance gains in both closed-source and large-scale open-source models. (C) Our pipeline, tested under similar settings, demonstrated effectiveness across both closed-source and large-scale open-source models, achieving the highest performance score of 48.77%. Detailed prompt templates are provided in appendices A.7.2 and A.7.3.

accuracy. For most models, the combined proportion of correctly answered cases and *Wrong due to Misleader* errors exceeds 75%, while *Wrong due to Others* errors remain around 20%. This suggests that although MLLMs effectively recognize general distractors, they struggle to detect and interpret misleaders, revealing a critical gap in their reasoning when faced with misleading chart elements.

(3) **Closed-source models exhibit lower performance compared to recent open-source models.** The GPT series outperforms other closed-source models, with 4o and o1 achieving similar accuracy. In contrast, recent open-source MLLMs surpass closed-source models, with performance improving as model size increases. With the exception of the DeepSeek-VL2 series, Qwen2.5-VL (7B & 72B) and InternVL2.5-MPO (8B, 26B, & 78B) surpass the GPT series, highlighting recent advancements in complex chart reasoning.

3.3 MLLMs' Performance Across Different Misleaders and Chart Types (RQ3-RQ4)

Our experiments show that MLLMs' ability to interpret misleading charts is heavily influenced by both misleader and chart types.

3.3.1 MLLMs' Performance Under Different Misleader Groups and Types

The summarized results for overall model performance across different misleader groups and specific misleader types are presented in table 2-(Baseline). Among the four misleader groups, **Manipulated Data** has the lowest overall *Accuracy* and the highest *Wrong due to Misleader* rate, while MLLMs perform best in the **Manipulated Visual Encoding** group. **Manipulated Data** misleaders primarily distort visual representations by inaccurately mapping or obscuring the true data distribution. In contrast, **Manipulated Visual Encoding** misleaders introduce visual errors through incorrect graphical displays. These results suggest that current MLLMs are more proficient at detecting visual discrepancies than reasoning about misleading data distributions, likely due to their training emphasis on aligning language models with visual models. Consequently, MLLMs find it easier to identify surface-level visual inconsistencies than to detect misleading patterns hidden within data structure or distribution. Examples of questions from these two misleader groups can be found in appendices A.5 and A.6.

Misleader		Wrong due to Others	Wrong due to Misleader	Accuracy
MANIPULATED DATA	Cherry Picking	16.07	52.21	31.72
	Missing Data	29.70	44.85	25.45
	Overplotting	33.18	40.24	26.58
	Inappropriate Order	32.46	33.43	34.12
	Missing Normalization	27.01	44.27	28.72
	Concealed Uncertainty	30.96	37.40	31.64
	Category Overall (normalized)	**28.23**	**42.07**	29.71
MANIPULATED ANNOTATION	Deceptive Labeling	21.05	45.88	33.07
	Lack of Labeling Lack of legend	34.66	33.08	32.26
	Lack of Labeling Lack of scales	30.19	32.72	37.09
	Inappropriate Aggregation	36.67	13.89	49.44
	Category Overall (normalized)	30.64	31.39	37.97
MANIPULATED VISUAL ENCODING	Dual Encoding	32.12	32.12	35.76
	Data-visual Disproportion	40.70	18.60	40.70
	Mismatched Encoding Continuous encoding	27.39	29.37	43.24
	Mismatched Encoding Categorical encoding	28.66	27.17	44.17
	Category Overall (normalized)	32.22	**26.82**	**40.97**
MANIPULATED SCALE	Small Size	35.70	23.44	40.86
	Dual Axes	31.27	35.65	33.08
	Exceeding the Canvas	36.22	26.44	37.33
	Inappropriate Scale Range	37.68	33.33	28.99
	Inappropriate Scale Functions	28.04	27.76	44.20
	Unconventional Scale Directions	8.63	66.89	24.48
	Misuse of Cumulative Relationship	35.56	29.78	34.67
	Category Overall (normalized)	30.44	34.76	34.23

Table 2: Overall statistics for different misleader groups and types. The **Manipulated Data** group exhibits both the highest *Wrong due to Misleader* rate and the lowest *Wrong due to Others* rate. In contrast, the **Manipulated Visual Encoding** group achieves the highest accuracy and the lowest *Wrong due to Misleader* rate, indicating that MLLMs are generally more adept at detecting visual discrepancies than performing in-depth reasoning on aspects such as data distribution. This may be attributed to the core training focus of MLLMs.

3.3.2 MLLMs' Performance Under Different Chart Types

On the other hand, MLLMs' overall performance across different chart types is summarized in 4. Among all chart types, **Line Chart** achieves the highest accuracy. Other basic charts, such as **Area Chart**, **Pie Chart**, and **Bar Chart**, generally receive better performance than more complex charts like **Choropleth Map**, **Stacked Area Chart**, and **Scatterplot**. This suggests that MLLM performance in identifying misleaders is significantly influenced by chart complexity.

However, the *Wrong due to Misleader* rate does not strictly correlate with chart complexity. For example, the **Choropleth Map** exhibits both the highest *Wrong due to Misleader* rate and the second-highest *Wrong due to Others* rate, likely due to its inherent complexity. In contrast, the **Stacked Area Chart** has the lowest *Wrong due to Misleader* rate but a very high *Wrong due to Others* rate, whereas the **Stacked Bar Chart** ranks among the top three in *Wrong due to Others* but has the second-lowest *Wrong due to Misleader* rate. This observation indicates that state-of-the-art MLLMs may still struggle with fundamental reasoning in stacked series charts, even in the absence of misleaders. The limited coverage of stacked series charts in existing chart-related benchmarks (Masry et al., 2022; Han et al., 2023; Wu et al., 2024) suggests that further research and dataset development are needed.

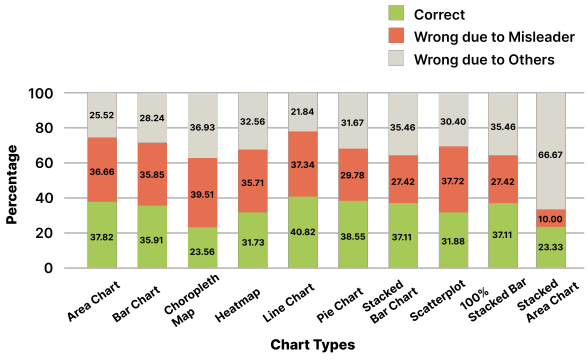


Figure 4: Overall statistics for each chart type. MLLMs demonstrate relatively weak comprehension of complex stacked series charts, while simpler charts, such as line charts, exhibit higher susceptibility to misleaders.

Furthermore, simple chart types can sometimes be more misleading than complex ones. For instance, basic charts like **Line Chart**, **Area Chart**,

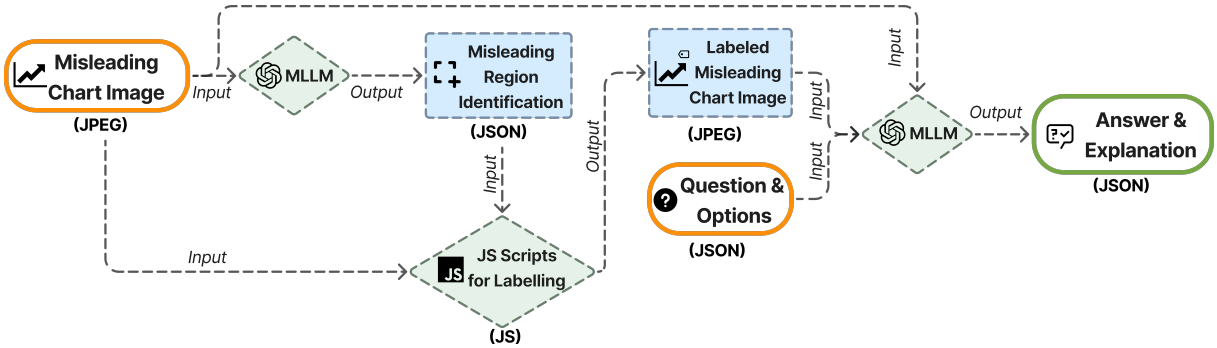


Figure 5: The proposed Region-Aware Misleader Reasoning pipeline. In the first stage, the MLLM component analyzes the misleading chart image using a checklist of common erroneous components and outputs the coordinates of potential misleading regions along with explanations. In the bridge stage, an additional JavaScript script takes these coordinates as input and overlays labels onto the original chart image. In the final stage, both the original chart image and question, as well as the labeled chart with explanations, are provided to the MLLM component, with the labeled version serving as a potential reference for generating the final answers.

and **Bar Chart** exhibit a 10% higher likelihood of *Wrong due to Misleader* compared to more complex charts like **Stacked Bar Chart** and 100% **Stacked Bar Chart**. This suggests that their simplicity, rather than aiding accuracy, may actually increase susceptibility to misleaders. Models are more likely to overlook manipulated components, displaying behavior similar to humans and making them more prone to interpreting these charts as intended by the chart authors.

3.4 Region-Aware Misleader Reasoning Pipeline (RQ5)

To enhance MLLMs' performance in the Misleading ChartQA task, we propose a multi-stage pipeline named Region-Aware Misleader Reasoning, designed to mimic the chart evaluation process used by domain experts. Our approach proposes to first identify and localize deceptive areas before conducting in-depth reasoning, incorporating additional scripts to facilitate the misleader identification process and enhance step-by-step reasoning.

As shown in fig. 5, rather than directly feeding the MLLMs a misleading chart question, the first stage of our pipeline prompts the model to analyze the chart image independently, using a checklist of the most commonly erroneous chart components. The MLLMs attempt to identify and highlight regions that might be misleading, producing a JSON file that contains the coordinates of the detected misleading regions along with explanations. These coordinates are then passed to an additional JavaScript script, which overlays bounding boxes onto the original chart image.

In the second stage, the newly labeled misleading chart image and its corresponding explanations are combined with the original question and answer options for processing by an additional MLLM component. To mitigate potential mislabeling errors, we also provide the original chart image and prompt MLLMs to treat the labeled version as a reference rather than a definitive input, acknowledging possible inaccuracies.

Since our proposed pipeline draws inspiration from the Chain-of-Thought (CoT) method and employs external scripts to facilitate step-by-step reasoning, we also conduct a series of experiments evaluating CoT performance. The detailed prompt templates for both experiments are provided in appendices A.7.2 and A.7.3. To ensure consistency with the baseline setting, we maintain a zero-shot setting in both the CoT process (Kim et al., 2023; DeepLearning.AI, 2025; Chen et al., 2024b) and our pipeline process. As shown in table 1-Pipeline, our pipeline demonstrates its effectiveness, outperforming both the baseline and zero-shot CoT methods across closed-source and large-scale open-source models, achieving the highest performance score of 42.69% and 48.77% respectively.

4 Discussion

Separately Enhancing MLLMs' Misleading Detection and Interpretation Abilities. Our experiments and analysis show that the proposed Region-Aware Misleader Reasoning pipeline improves MLLMs' comprehension of misleading charts. While the pipeline boosts the best closed-source model's performance by 8%, both the zero-

shot CoT and pipeline settings only marginally enhanced open-source models, improving baseline performance by approximately 1%–2%, with the highest accuracy still below 50%.

Through manual inspection of failure cases in our pipeline, we identified two primary issues: 1). *MLLMs incorrectly localized the misleading region in more than half of the cases*, leading to errors in subsequent reasoning. 2). *Among MLLMs that correctly localized the misleading region and provided explanations, one-third still failed to answer the QA correctly in the final stage.* These findings highlight the intrinsic challenges of the Misleading ChartQA task. For future work, we suggest separately improving MLLMs' ability to identify and localize misleading elements, as well as enhancing their interpretation and reasoning capabilities, as a promising approach to boosting performance on the Misleading ChartQA.

5 Related Works

5.1 Chart Reasoning and Related Benchmarks for MLLMs

Chart Reasoning has emerged as a key area of focus within the vision-language community, with several benchmarks developed to assess models' abilities to interpret and reason about charts. Early datasets such as ChartQA(Masry et al., 2022) and PlotQA(Methani et al., 2020) primarily evaluated basic chart understanding, focusing on three common chart types. These datasets were relatively straightforward for recent MLLMs to solve. Subsequent benchmarks have either expanded chart type coverage (Han et al., 2023; Xia et al., 2024; Xu et al., 2023) or refined the complexity of tasks, distinguishing between high-level tasks (e.g., chart captioning, chart summarization (Kantharaj et al., 2022; Rahman et al., 2023; Cheng et al., 2023; Huang et al., 2023; Liu et al., 2022)) and low-level tasks (e.g., extracting numerical values (Kahou et al., 2017; Wu et al., 2024)). Some works have also introduced more complex tasks such as chart structure extraction (Chen et al., 2024a). A detailed comparison of chart variety with existing benchmarks is provided in table 3 and fig. 8.

5.2 Misleading Chart Visualizations and Data Communication

Misleading chart visualizations have long been a significant topic in data visualization and human-computer interaction (King, 1986). Several standardized tests have been designed to evaluate human chart understanding and reasoning abilities (Lee et al., 2016; Boy et al., 2014; Börner et al., 2019). Recent efforts have evolved to emphasize critical thinking in chart comprehension, identifying around 10 categories of common misleaders in charts and formulating nuanced questions for human testing (Ge et al., 2023; Cui et al., 2023). However, these question sets consist of only about 40 questions, each addressing one or two examples of (misleader, chart type) combinations, which limits their effectiveness for evaluating MLLMs. Other latest studies have attempted to summarize common misleading visualization practices (Lo et al., 2022; Lan and Liu, 2024), but these focus on broad visualization design issues that do not directly apply to chart understanding tasks.

5.3 Empirical Explorations on MLLMs in Understanding Misleading Charts

Several recent studies have empirically evaluated MLLMs' performance in understanding misleading chart visualizations by testing them on existing standardized tests designed for humans (Bendeck and Stasko, 2024; Hong et al., 2025; Lo and Qu, 2024; Zeng et al., 2024). These studies typically involved a limited number of models and questions, making it difficult to draw reliable conclusions about MLLMs' ability. In contrast, our work constructs a diverse benchmark dataset with over 3,000 samples, covering a broad range of misleaders and chart types. Through a comprehensive evaluation of 16 state-of-the-art MLLMs, we establish a strong foundation for this task first-ever.

6 Conclusions

We propose *Misleading ChartQA*, the first benchmark designed to evaluate MLLMs' ability to comprehend misleading chart visualizations—a prevalent and significant real-world challenge. Our findings reveal that while the latest MLLMs exhibit some improvement, their performance on this task remains limited, achieving results comparable to those of the general public. Additionally, our analysis highlights that different types of misleaders, chart formats, and their combinations significantly influence MLLMs' performance. Further research is needed to separately enhance MLLMs' capabilities in both misleader identification and interpretation to improve their overall effectiveness in Misleading ChartQA.

Limitations

Limited Visual Prompt Design and Comparison In line with the original models publishers' approaches (e.g., Qwen, DeepSeek, and InternVL series), which primarily use zero-shot methods for ChartQA benchmark testing, our evaluation also adopts a zero-shot approach. While this alignment facilitates comparison, it is likely that MLLMs' performance could be further enhanced through few-shot learning methods. Future work could explore this by incorporating few-shot techniques to potentially improve the models' capabilities in handling misleading chart detection tasks.

Lack of Fine-Tuning on MLLMs We did not explore fine-tuning methods to improve MLLMs' performance on this task. The main reason for this is our goal of first obtaining a comprehensive understanding of the performance of the latest generation of MLLMs on Misleading Chart QA. Based on the results of our experiments, future research could investigate fine-tuning, particularly with the InternVL2-5-78B-MPO model, which exhibited the strongest performance among all the models tested.

References

Alexander Bendeck and John Stasko. 2024. An empirical evaluation of the gpt-4 multimodal language model on visualization literacy tasks. *IEEE Transactions on Visualization and Computer Graphics*.

Katy Börner, Andreas Bueckle, and Michael Ginda. 2019. Data visualization literacy: Definitions, conceptual frameworks, exercises, and assessments. *Proceedings of the National Academy of Sciences*, 116(6):1857–1864.

Michael Bostock, Vadim Ogievetsky, and Jeffrey Heer. 2011. D^3 data-driven documents. *IEEE transactions on visualization and computer graphics*, 17(12):2301–2309.

Jeremy Boy, Ronald A Rensink, Enrico Bertini, and Jean-Daniel Fekete. 2014. A principled way of assessing visualization literacy. *IEEE transactions on visualization and computer graphics*, 20(12):1963–1972.

Jinyue Chen, Lingyu Kong, Haoran Wei, Chenglong Liu, Zheng Ge, Liang Zhao, Jianjian Sun, Chunrui Han, and Xiangyu Zhang. 2024a. Onechart: Purify the chart structural extraction via one auxiliary token. In *Proceedings of the 32nd ACM International Conference on Multimedia*, pages 147–155.

Zhe Chen, Weiyun Wang, Yue Cao, Yangzhou Liu, Zhangwei Gao, Erfei Cui, Jinguo Zhu, Shenglong Ye, Hao Tian, Zhaoyang Liu, et al. 2024b. Expanding performance boundaries of open-source multimodal models with model, data, and test-time scaling. *arXiv preprint arXiv:2412.05271*.

Zhi-Qi Cheng, Qi Dai, and Alexander G Hauptmann. 2023. Chartreader: A unified framework for chart derendering and comprehension without heuristic rules. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 22202–22213.

Yuan Cui, W Ge Lily, Yiren Ding, Fumeng Yang, Lane Harrison, and Matthew Kay. 2023. Adaptive assessment of visualization literacy. *IEEE Transactions on Visualization and Computer Graphics*.

DeepLearning.AI. 2025. ChatGPT Prompt Engineering for Developers - DeepLearning.AI.

Lily W Ge, Yuan Cui, and Matthew Kay. 2023. Calvi: Critical thinking assessment for literacy in visualizations. In *Proceedings of the 2023 CHI conference on human factors in computing systems*, pages 1–18.

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

Jiayi Hong, Christian Seto, Arlen Fan, and Ross Maciejewski. 2025. Do llms have visualization literacy? an evaluation on modified visualizations to test generalization in data interpretation. *IEEE Transactions on Visualization and Computer Graphics*.

Kung-Hsiang Huang, Mingyang Zhou, Hou Pong Chan, Yi R Fung, Zhenhailong Wang, Lingyu Zhang, Shih-Fu Chang, and Heng Ji. 2023. Do lvlms understand charts? analyzing and correcting factual errors in chart captioning. *arXiv preprint arXiv:2312.10160*.

Darrell Huff. 2023. *How to lie with statistics*. Penguin UK.

Samira Ebrahimi Kahou, Vincent Michalski, Adam Atkinson, Ákos Kádár, Adam Trischler, and Yoshua Bengio. 2017. Figureqa: An annotated figure dataset for visual reasoning. *arXiv preprint arXiv:1710.07300*.

Shankar Kantharaj, Rixie Tiffany Ko Leong, Xiang Lin, Ahmed Masry, Megh Thakkar, Enamul Hoque, and Shafiq Joty. 2022. Chart-to-text: A large-scale benchmark for chart summarization. *arXiv preprint arXiv:2203.06486*.

Melita Kennedy, Steve Kopp, et al. 2000. *Understanding map projections*, volume 8. Esri Redlands, CA.

Seungone Kim, Se June Joo, Doyoung Kim, Joel Jang, Seonghyeon Ye, Jamin Shin, and Minjoon Seo. 2023. The cot collection: Improving zero-shot and few-shot learning of language models via chain-of-thought fine-tuning. *arXiv preprint arXiv:2305.14045*.

Gary King. 1986. How not to lie with statistics: Avoiding common mistakes in quantitative political science. *American Journal of Political Science*, pages 666–687.

Xingyu Lan and Yu Liu. 2024. "i came across a junk": Understanding design flaws of data visualization from the public's perspective. *IEEE Transactions on Visualization and Computer Graphics*.

Sukwon Lee, Sung-Hee Kim, and Bum Chul Kwon. 2016. Vlat: Development of a visualization literacy assessment test. *IEEE transactions on visualization and computer graphics*, 23(1):551–560.

Fangyu Liu, Francesco Piccinno, Syrine Krichene, Chenxi Pang, Kenton Lee, Mandar Joshi, Yasemin Altun, Nigel Collier, and Julian Martin Eisenschlos. 2022. Matcha: Enhancing visual language pretraining with math reasoning and chart derendering. *arXiv preprint arXiv:2212.09662*.

Leo Yu-Ho Lo, Ayush Gupta, Kento Shigyo, Aoyu Wu, Enrico Bertini, and Huamin Qu. 2022. Misinformed by visualization: What do we learn from misinformative visualizations? In *Computer Graphics Forum*, volume 41, pages 515–525. Wiley Online Library.

Leo Yu-Ho Lo and Huamin Qu. 2024. How good (or bad) are llms at detecting misleading visualizations? *IEEE Transactions on Visualization and Computer Graphics*.

Ahmed Masry, Do Xuan Long, Jia Qing Tan, Shafiq Joty, and Enamul Hoque. 2022. Chartqa: A benchmark for question answering about charts with visual and logical reasoning. *arXiv preprint arXiv:2203.10244*.

Morgan McFall-Johnsen. 2020. A 'cuckoo' graph with no sense of time or place shows how Georgia bungled coronavirus data as it reopens.

Nitesh Methani, Pritha Ganguly, Mitesh M Khapra, and Pratyush Kumar. 2020. Plotqa: Reasoning over scientific plots. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 1527–1536.

Lotti O'Brien. 2024. Maps of world 'completely misleading' as true size of Europe, China and Africa revealed.

Raian Rahman, Rizvi Hasan, Abdullah Al Farhad, Md Tahmid Rahman Laskar, Md Hamjajul Ashmafee, and Abu Raihan Mostofa Kamal. 2023. Chartsumm: A comprehensive benchmark for automatic chart summarization of long and short summaries. *arXiv preprint arXiv:2304.13620*.

Edward R Tufte and Peter R Graves-Morris. 1983. *The visual display of quantitative information*, volume 2. Graphics press Cheshire, CT.

Yifan Wu, Lutao Yan, Leixian Shen, Yunhai Wang, Nan Tang, and Yuyu Luo. 2024. Chartinsights: Evaluating multimodal large language models for low-level chart question answering. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 12174–12200.

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

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

Xingchen Zeng, Haichuan Lin, Yilin Ye, and Wei Zeng. 2024. Advancing multimodal large language models in chart question answering with visualization-referenced instruction tuning. *IEEE Transactions on Visualization and Computer Graphics*.

A Appendix

A.1 Real-world examples of misleading visualizations

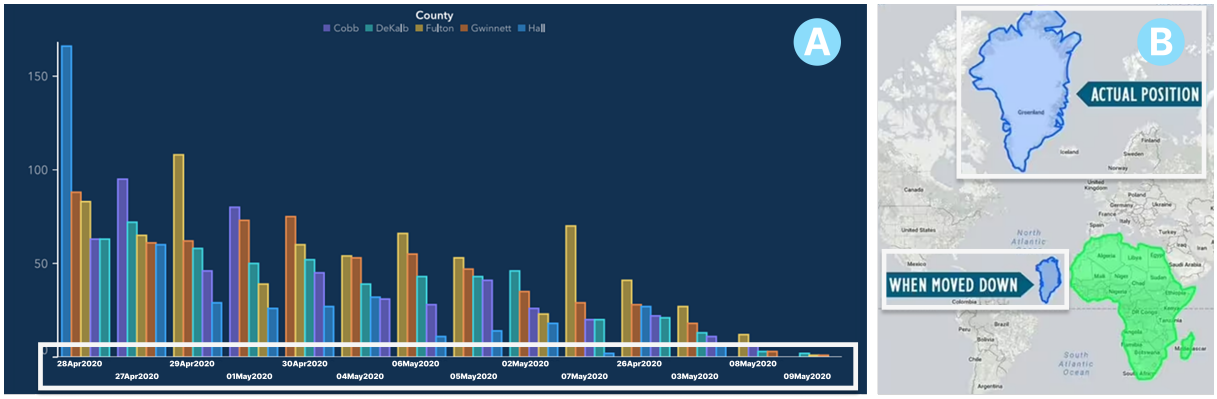


Figure 6: Two real-world examples of misleading chart visualizations. **(A)** A bar chart of COVID-19 cases across five counties, sorted by case count rather than by date, creating the false impression of a declining trend unless viewers carefully examine the x-axis. **(B)** The commonly used world map projection, which misrepresents Greenland as being the same size as Africa, despite Africa being significantly larger.

A.2 Misleader Definition

	Misleader Name	Definition	Area Chart	Bar Chart	Choropleth Map	Heatmap	Line Chart	Pie Chart	Stacked Area Chart	Stacked Bar Chart	Scatterplot	100% Stacked Bar Chart
Manipulated Data	**Missing Normalization**	Displaying unnormalized absolute values when relative or normalized comparisons would be more appropriate for interpretation.			✓							
	Concealed Uncertainty	Omitting uncertainty in visualizations can misrepresent the reliability of underlying data. In predictive contexts, this may lead viewers to develop an unjustified sense of confidence in the conclusions.		✓	✓						✓	
	Cherry Picking	Selecting only a subset of data to display, potentially misleading viewers by implying conclusions about the entire dataset.					✓				✓	
	Missing Data	Presenting a visual representation that suggests data exists when, in reality, it is missing.			✓							
	Overplotting	Overcrowding a visualization with excessive data points or elements, making it difficult to discern meaningful patterns.									✓	
	Inappropriate Order	Manipulating the order of data by manipulating axis labels or legend items in a way that misleads viewers or creates a false impression of trends.	✓	✓	✓						✓	✓
Manipulated Annotation	**Deceptive Labeling**	Using annotations or labels that contradict the data or make the visualization difficult to interpret.		✓			✓	✓				
	Lack of Labeling: *Lack of legend*	Omitting a legend that explains colors, symbols, or other encodings, leaving viewers uncertain about the meaning of the visualization.							✓	✓		
	Lack of Labeling: *Lack of scales*	Failing to provide axis scales or units of measurement, which can oversimplify or obscure the interpretation of the data.	✓	✓			✓					
	Inappropriate Aggregation	Combining or summarizing data in a way that distorts the true distribution or relationships, leading to inaccurate conclusions.	✓	✓			✓				✓	✓
Manipulated Visual Encoding	**Data-visual Disproportion**	Creating a visual representation where the graphical elements (e.g., bar heights) do not accurately correspond to the actual data values, leading to misinterpretation.		✓			✓	✓			✓	
	Dual Encoding	Using multiple visual channels (e.g., both width and height) to encode the same variable, which exaggerates the data's visual impact.		✓				✓				
	Mismatched Encoding: *Continuous encoding for categorical data*	Applying continuous encoding methods (e.g., color gradients or line connections) to categorical data, which can mislead viewers into perceiving relationships that do not exist.	✓				✓	✓				
	Mismatched Encoding: *Categorical encoding for continuous data*	Representing continuous data using discrete categories, potentially distorting trends and relationships.			✓	✓						
Manipulated Scale	**Inappropriate Scale Range**	Altering the scale of axes or legends by stretching, truncating, or using inconsistent binning, which distorts data representation.		✓	✓		✓		✓			✓
	Inappropriate Scale Functions	Applying arbitrary or misleading non-linear transformations to the scale of an axis, affecting how viewers perceive the relationships within the data.		✓			✓	✓				
	Unconventional Scale Directions	Using non-standard axis or legend orientations, such as inverting scales, which can confuse viewers and misrepresent relationships.	✓	✓	✓		✓				✓	
	Misuse of Cumulative Relationship	Incorrectly combining or accumulating data elements that do not logically sum or relate, distorting the true relationships.							✓	✓		✓
	Exceeding the Canvas	Allowing data points, labels, or visual elements to extend beyond the display area, causing loss of critical information.	✓	✓			✓					
	Small Size	Using excessively small text or graphical elements that hinder readability and make it difficult to interpret data.			✓		✓				✓	
	Dual Axes	Incorporating multiple axes in a way that complicates comparisons and forces viewers to mentally align different scales.					✓					

Figure 7: List of misleaders categorized under each misleader group, along with their detailed definitions and corresponding chart types. In total, there are 60 (misleader, chart type) pairings.

A.3 Comparison with the existing benchmarks for chart-related evaluations.

Task Focus	Datasets	#-Chart Types	# Chart	# Task type	Metadata?	Chart Code?	Chart Data?
Basic understanding	ChartQA	3	4.8k	4	N	N	N
	PlotQA	3	224k	1	N	N	N
Summarization/ captioning	ChartLlama	10	11k	7	N	N	N
	ChartBench	11	2.1k	4	N	N	N
	Chart-to-text	6	44k	3	N	N	N
	Chartsumm	3	84k	1	Y	N	N
Data/structure extraction	ChartInsights	7	2k	10	Y	N	N
	FigureQA	5	120K	6	N	N	N
Misleading Chart Comprehension	**Misleading ChartQA**	**10**	**3k**	**21**	**Y**	**Y**	**Y**

Table 3: Comparison of the Misleading ChartQA dataset with existing benchmarks. Misleading ChartQA is the first dataset specifically designed for the misleading chart comprehension task. It also features a diverse range of chart types and task types, along with rich metadata, chart code, and chart data.

A.4 Distribution of Chart Types in the Misleading ChartQA Dataset

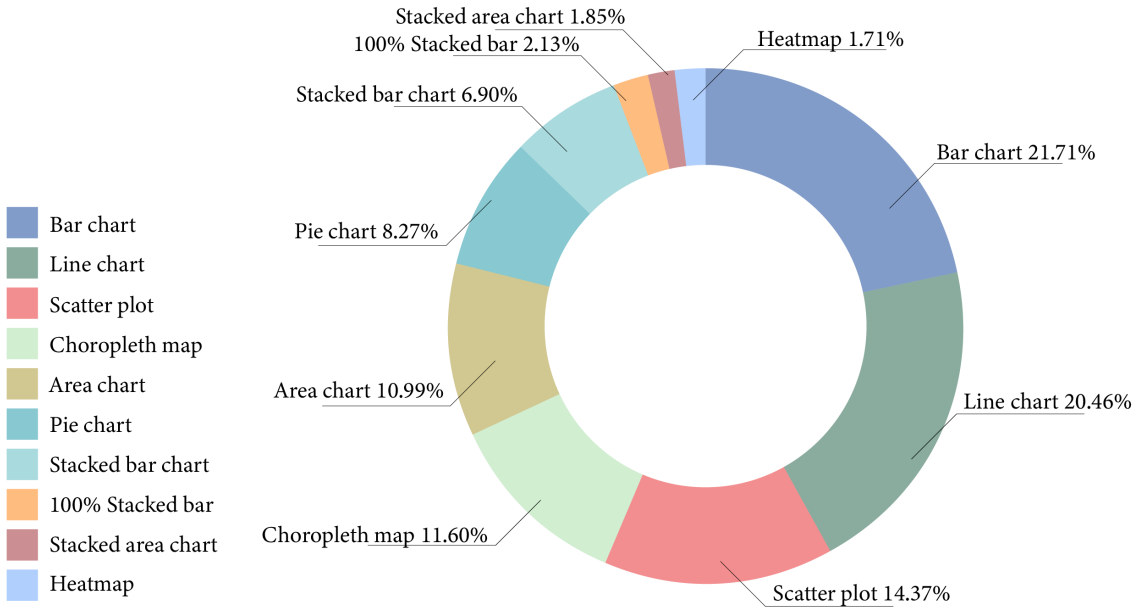
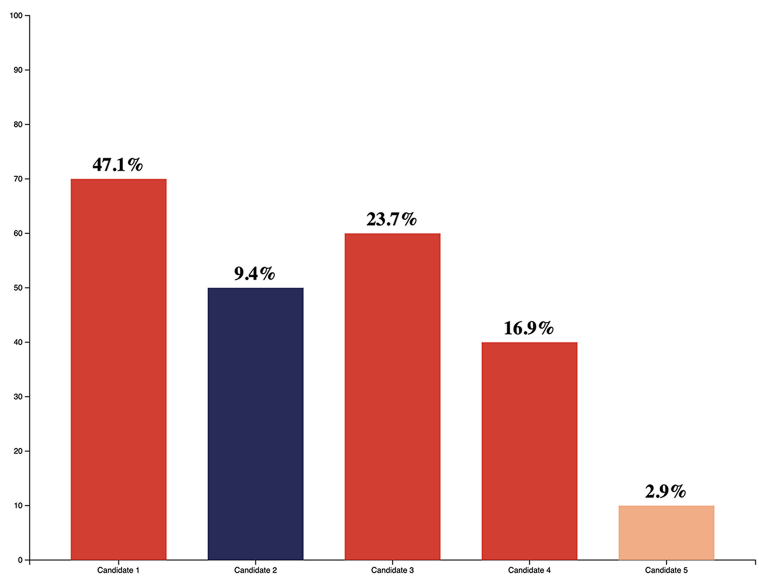


Figure 8: Breakdown of Chart Types in the Misleading ChartQA Dataset.

A.5 Examples of Questions from Manipulated Visual Encoding Group



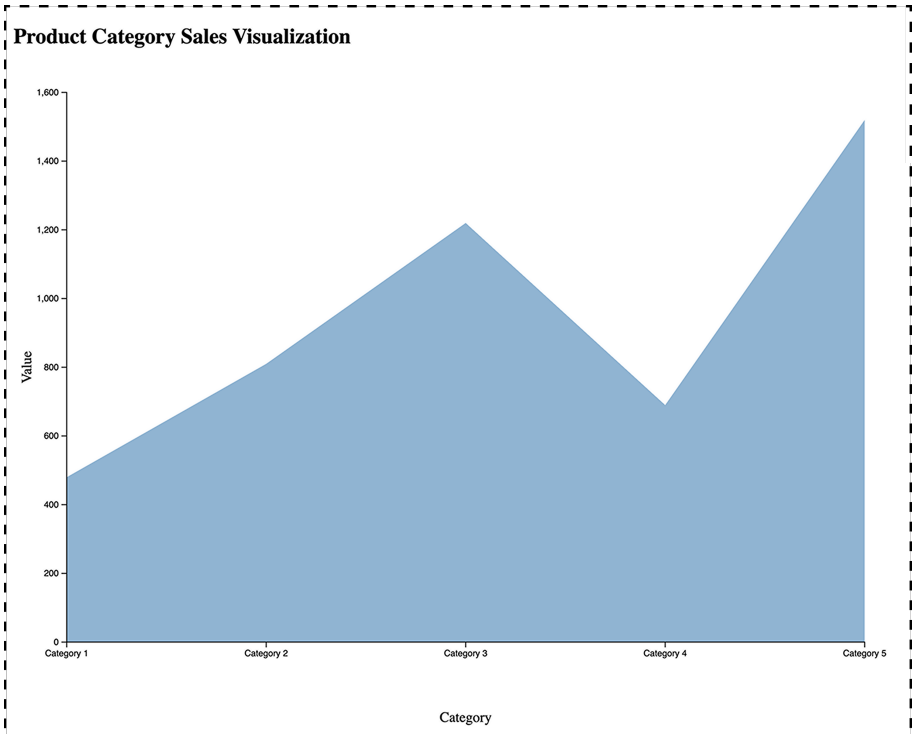
The chart displays the percentage of votes received by five candidates in a local election. Which of the following statements accurately describes the candidate with the lowest percentage of votes?

A: Candidate 1 received exactly 40% of the total votes.
B: Candidate 2 is in second place.
C: Candidate 5 received the least votes.
D: Candidate 3 and Candidate 4 have nearly the same percentage of votes.

Correct Answer: C | **Misleading Answer:** B
Misleader: *Data-Visual Disproportion* | **Chart Type:** *Bar Chart*
Task: *Data Interpretation* | **Difficulty:** *Low*

Explanation: *The bar chart clearly shows that Candidate 5 received the smallest percentage of votes (2.9%), making option C correct. Option A is incorrect as Candidate 1 received 47.1% of the votes. Option B is incorrect because Candidate 2 is not in second place, with only 9.4% of the vote rate.*

Figure 9: An example question from the **Manipulated Visual Encoding** group, categorized under *Data-Visual Disproportion* and presented as a *Bar Chart*.



The area chart displays values for five different categories. Which of the following interpretations of the data is misleading based on how the chart is presented?

A: There is a sequential trend in product performance over time.

B: Category 3 has the highest cumulative value among the products.

C: Each category is distinct without any implied temporal relationship.

D: All categories are represented by the same type of data.

Correct Answer: A

Misleading Answer: C

Misleader: *Mismatched Encoding: Continuous encoding for categorical data*

Chart Type: *Area Chart*

Difficulty: *Medium*

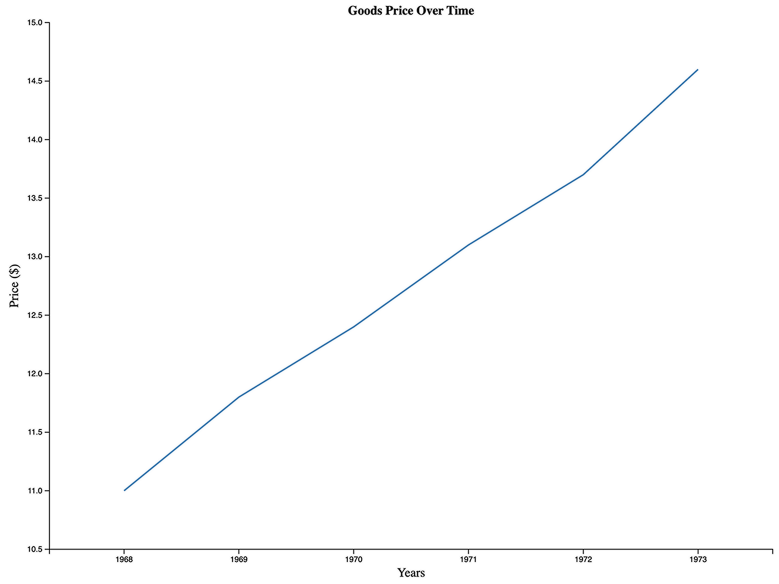
Task: *Chart Interpretation*

Explanation: *The chart incorrectly uses an area chart, leading to misleading interpretations of trends in categorical data which should not be interpreted as sequential values over time.*

Figure 10: An example question from the **Manipulated Visual Encoding** group, categorized under *Mismatched Encoding: Continuous encoding for categorical data* and presented as a *Area Chart*.

A.6 Examples of Questions from Manipulated Data Group

Goods Price Visualization



What is the trend of the price changes from 1968 to 1980 based on the provided chart data?

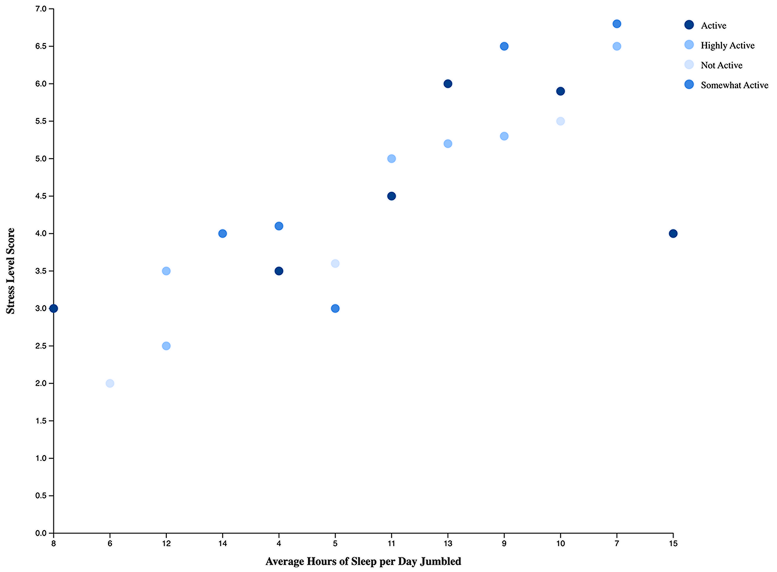
A: Generally increasing.
B: Generally decreasing.
C: Generally stable.
D: Cannot be inferred.

Correct Answer: D
Misleading Answer: A
Misleader: *Cherry Picking*
Chart Type: *Line Chart*
Task: *Correlations/Trends*
Difficulty: *Medium*

Explanation: *The chart shows an increasing trend during the selected period (1968-1973). However, when considering the broader data set (1973-1980), this snippet actually excludes important periods of decline, leading to a misleading interpretation.*

Figure 11: An example question from the **Manipulated Data** group, categorized under *Cherry Picking* and presented as a *Line Chart*.

Sleep Hours vs. Stress Level for People in Highly Active Neighborhood



Based on the chart above, which of the following statements about the activeness levels across different sleep hours is accurate?

A: People sleeping between 6-7 hours are mostly 'Highly Active'.

B: Individuals with 10-11 hours of sleep tend to be 'Not Active'.

C: The 'Activeness' trends are easier to interpret if presented in a logical order.

D: None of the above.

Correct Answer: C

Misleading Answer: A

Misleader: *Inappropriate Order*

Chart Type: *Scatterplot*

Task: *Make Comparisons Find Correlations/Trends*

Difficulty: *High*

Explanation: *The legend for 'Activeness' is misleading due to its inappropriate order. The categories are not presented in a logical sequence, making it harder to interpret trends.*

Figure 12: An example question from the **Manipulated Data** group, categorized under *Inappropriate Order* and presented as a *Scatterplot*.

A.7 Prompt Templates

A.7.1 Automated MCQ Expansion and Iterative Refinement workflow

The following are the prompts for each components in the proposed Automated MCQ Expansion and Iterative Refinement workflow (fig. 3).

Chart Variation

Generate HTML Variation

You are generating misleading HTML-based charts for a QA benchmark using D3.js. The goal is to modify the visualization to reflect the misleader $\{misleader\}$ by adjusting the chart's visual representation while maintaining core structure and labels.
Requirements:
1. The base HTML provided serves as the primary reference. Maintain the same overall structure, styles, and chart components. The generated HTML must be directly runnable.
2. Retain the following from the base HTML:
    - Chart dimensions (fixed at 1000x750 pixels).
    - Titles, legends, axis labels, and grid lines.
    - D3.js visualization logic.
3. Modify the D3 chart to apply the misleader.
4. Ensure the chart reads data from the updated CSV path: $\{csv\_path\_in\_html\}$.
    - Ensure there are no extra or duplicated closing parentheses ')' in the 'd3.csv' function call.
5. Prevent overflow by adjusting margins and ensuring all chart elements fit within the canvas.
6. Use the labelled JPEG sample as a visual guide to ensure the misleader effect is accurately represented.
7. Remove all unnecessary comments, such as:
    - Descriptive comments like "Here's the complete and executable HTML page..."
    - Markdown syntax (e.g., "'html, "').
8. **Ensure the chart title reflects the new chart topic but do not infer the misleader in the chart title**:
    - The title should match the description of the relevant CSV columns. Make sure do not infer the misleaders in chart title. Keep the same
9. **Ensure axis labels dynamically update**:
    - Use the column names from the CSV data for axis labels whenever appropriate. Make sure do not infer the misleaders in the axis labels.
Returns: str: The generated HTML content only.

Misleader: $\{misleader\}$
Misleader Description: $\{misleader\_description\}$
Chart Type: $\{chart\_type\}$
CSV Data (Driving the Chart): $\{csv\_data\}$
Base HTML (Reference for Structure and Style): $\{base\_html\}$
JPEG (Labelled Misleader):
    - Refer to the attached JPEG for visual alignment. Path to JPEG: $\{jpeg\_path\}$
Ensure the full visualization code (chart headings, legends, titles, axes) is preserved:
Return the output as a complete and executable HTML page in the following format:

```
<!DOCTYPE HTML>
<html lang="en">
```

```
    <head>
        <meta charset="UTF-8">
        <meta name="viewport" content="width=device-width, initial-scale=1.0">
        <script src="https://d3js.org/d3.v6.min.js"></script>
        <style>
            #chart {{
                width: 1000px;
                height: 750px;
                margin: 60px auto;
            }}
            .axis path, .axis line {{
                stroke: black;
            }}
            .dot {{
                fill: steelblue;
                stroke: black;
                stroke-width: 1px;
            }}
            .avg-line {{
                stroke: black;
                stroke-dasharray: 4,4;
            }}
            .annotation {{
                font-size: 12px;
                font-weight: bold;
                fill: black;
            }}
        </style>
    </head>
    <body>
        <h1> // // Insert appropriate chart heading like the base HTML,
        ensure do not disclose the misleader information here </h1>
        <div id="chart"></div>
        <script>
            // Insert D3.js visualization logic extracted from base HTML here
        </script>
    </body>
    </html>
     ```
```

- Ensure that the returned HTML page preserves the full chart functionality and visualization logic from the base HTML.
- Implement the misleader described above by modifying axis scaling, bar order, or annotation placement.
- The goal is to introduce subtle distortions that create misleading visual interpretations while retaining the core chart layout.
```

Generate CSV Variation

You are modifying CSV data for a $\{chart\_type\}$ visualization that reflects the misleader $\{misleader\}$.
Instructions:
1. Keep the same number of columns ($\{expected\_num\_columns\}$) as the original CSV.
2. Ensure each column has the same data type (e.g., int, float, string) as the original CSV.
3. Modify column names and data values to reflect the misleader effect:
    - $\{misleader\_description\}$
4. Return only the modified CSV content with no additional comments or metadata.

Original CSV Data: $\{csv\}$

QA Generation

Generate QA Variation

You are generating Q&A content for a misleading chart which is generated as a variation of the sample example. Please strictly follow the style of the sample (in which a chart with labeled misleading region and the corresponding Q&A is provided). The goal is to craft a question that highlights the misleading aspect of the variation chart accordingly.

Requirements:
1. Follow the structure of the provided JSON file exactly.
2. Frame the question to reflect the misleading aspect of the chart.
3. Adjust the options (A, B, C, D) to ensure one option aligns with the misleader.
4. Indicate the correct answer clearly.
5. Choose the most misleading option as "wrongDueToMisleaderAnswer" to highlight the most plausible incorrect option caused by the misleading chart.
6. Reference the JPEG-labelled chart and Q&A sample to ensure the explanation correctly addresses the visual misleader.
7. Set the "ifLabelled" field to "False" to indicate the chart is not labelled.

Misleader: $\{misleader\}$
Misleader Description: $\{misleader\_description\}$
Chart Type: $\{chart\_type\}$
CSV Data (Driving the Variation Chart): $\{csv\_data\}$
The target Misleading Chart (Variation Chart): $\{chart\_variation\}$
Sample Q&A JSON (Structure Reference): $\{base\_json\}$
Sample Chart JPEG (with Labelled Misleader):
    - Refer to the attached JPEG for visual alignment.
    - Path to JPEG: $\{jpeg\_path\}$
Return the output in this strict format:

```
 ```json
{{
  "question": "Based on the chart, what is the approximate average sales for Q1
   2023 in Restaurant X?",
   "options": {{
```
```

```
      "A": "120",
      "B": "180",
      "C": "220",
      "D": "250"
    }},
    "correctAnswer": "B",
    "misleader": "{misleader}",
    "chartType": "{chart_type}",
    "task": "Aggregate Values",
    "explanation": "The chart annotation shows 'Reference: 220', but the true
    average is 180. Misleading annotations cause users to misjudge the data.",
    "difficulty": "Medium",
    "ifLabelled": "False",
    "wrongDueToMisleaderAnswer": "C"
  }}
  ```
```
```

Automated Evaluation & Feedback & Refinement Loop

Variation Evaluation

You are tasked with evaluating and refining a visualization QA sample for a misleading chart.

** Inputs **
- **QA Content**: $\{qa\_content\}$
- **Misleader Description**: $\{misleader\_desc\}$
- **Misleadering Chart Image**: $\{chart\_image\}$
- **CSV Variation Check**: $\{csv\_variation\_status\}$
- **Generated CSV **: $\{generated\_csv\}$
- **Original CSV **: $\{original\_csv\}$

** Task **
Evaluate the chart (visualization), question, QA options, correct answer, wrong-Due-To-Misleader-Answer all match the misleader description. If you find anything wrong, try to identify the corresponding errors in the CSV, QA, and HTML components based on the below guidelines and commen issues.
Ensure:
- Make sure to double check the visualization indeed represents the intended misleader as described in the misleader description!
- Make sure to check if the QA content matches the misleader and visualization.
- Make sure to double check the correctness of the correct answer and wrongDueToMisleaderAnswer based on the misleader description and the chart figure!
- Make sure to check if the generated CSV introduces meaningful variations compared to the original CSV.
- Make sure to double check the items in the list of "Some common issues include" below.

** Guidelines **
Evaluate the chart (visualization), question, QA options, correct answer, wrong-Due-To-Misleader-Answer, and alignment with the misleader description. Provide status as 'correct' or 'incorrect':
- "correct": No refinement needed.
- "incorrect": Refinement needed, provide comments and instructions.

- If the sample is correct, set "status": "correct" and leave "comments", "revision_instructions", and "updated_content" fields empty or as "No issues" and "null".
- If the sample requires refinement, set "status": "incorrect" and provide detailed comments and specific revision instructions for each component ("csv", "qa", "html").

** For the updated_content for "qa", directly provided the revised content in JSON format. **
** For the updated_content for "csv" and "HTML", provide very detailed samples and do not include the whole code. **

** Some common issues include: **
CSV:
- The data values have no changes (no small variations) with the original data. Only changed the column names.

- Incorrect or missing data values.

QA:
- Mismatched question context (e.g., question does not align with the chart's content).
- Mismatched options (e.g., no correct answer choices exist).
- Missing or incorrect correct answers (e.g., no correct option, or wrong answer marked as correct).
- Incorrect explanations (e.g., explanation does not match the chart or the misleader description).
- Incorrect or missing wrongDueToMisleaderAnswer (e.g., wrong answer does not align with the misleader).

HTML:
- The CSV data path in the D3.js code is incorrect. Ensure the path in the D3.js code is path: $\{csv\_path\_in\_html\}$.
- Disclose the misleader in the visualization title (e.g., title implies it is a misleading visualization).
- Not specified by misleader description, but still missing labels or legend.
- Have any annotations to indicate misleading nature. Need to remove them.

** Output Format **
Return a JSON object with the following structure:

```json
{{
    "status": "<correct/incorrect>",
    "comments": {{
        "csv": "<Comment for CSV refinement or 'No issues'>",
        "qa": "<Comment for QA refinement or 'No issues'>",
        "html": "<Comment for HTML refinement or 'No issues'>"
    }},
    "revision_instructions": {{
        "csv": "<Specific instructions for revising the CSV or
        'No revision required'>",
        "qa": "<Specific instructions for the revised QA or
        'No revision required'>",
        "html": "<Specific instructions for revising the HTML or
        'No revision required'>"
    }},
    "updated_content": {{
        "csv_data": "<Updated CSV content if applicable or null>",
        "qa_content": "<Updated QA content if applicable or null>",
        "html_content": "<Updated HTML content if applicable or null>"
    }}
}}
```

Revision Loop: CSV

You are tasked with revising a CSV file to address specific issues. If you find no issues mentioned in the Comments and Instructions or they are unclear, please directlty output the Current CSV Content $\{csv\_content\}$ without any changes.

*** Comments:
$\{comments\}$

*** Instructions:
$\{instructions\}$

*** Current CSV Content:
$\{csv\_content\}$

*** Revised CSV Sample:
$\{revised\_csv\_sample\}$

*** Task
Make the necessary revisions to the CSV file according to the Comments, Instructions and Revised CSV Sample. Return the updated content as a valid CSV file.

Revision Loop: HTML

You are tasked with revising an HTML file to address specific issues. If you find no issues mentioned in the Comments and Instructions or they are unclear, please directlty output the Current HTML Content $\{html\_content\}$ without any changes.

*** Comments:
$\{comments\}$

*** Instructions:
$\{instructions\}$

*** Current HTML Content:
$\{html\_content\}$

*** Task
Make the necessary revisions to the HTML file and return the updated content as valid and executable HTML.

Ensure the full visualization code (chart headings, legends, titles, axes) is preserved:

Make sure to replace the CSV path in the D3.js code with the correct path $\{csv\_path\_in\_html\}$.

Make sure to remove any annotations or titles in the visualization that disclose the misleader! (e.g., should not have some extra titles indicating the potential misleader)

Make sure the visualization represents the misleader as intended.

Make sure to not change the other parts of the visualization code.

Return the output as a complete and executable HTML page in the following format:

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <script src="https://d3js.org/d3.v6.min.js"></script>
    <style>
        #chart {{
            width: 1000px;
            height: 750px;
            margin: 60px auto;
        }}
        .axis path, .axis line {{
            stroke: black;
        }}
        .dot {{
            fill: steelblue;
            stroke: black;
            stroke-width: 1px;
        }}
        .avg-line {{
            stroke: black;
```

```
            stroke-dasharray: 4,4;
        }}
        .annotation {{
            font-size: 12px;
            font-weight: bold;
            fill: black;
        }}
    </style>
</head>
<body>
    <h1> // Insert appropriate chart heading like the base HTML, ensure do not
    disclose the misleader information here </h1>
    <div id="chart"></div>
    <script>
        // D3.js visualization logic
        d3.csv("{csv_path_in_html}")
            .then(function(data) {{
                // Chart logic here
            }})
            .catch(function(error) {{
                console.error('Error loading CSV data:', error);
            }});
    </script>
</body>
</html>
```

Revision Loop: Q&A

You are tasked with revising a QA JSON file to address specific issues. If you find no issues mentioned in the Comments and Instructions or they are unclear, please directlty output the Current QA Content $\{qa\_content\}$ without any changes.

*** Comments:
$\{comments\}$

*** Instructions:
$\{instructions\}$

*** Current QA Content:
$\{qa\_content\}$

*** Revised QA Recommendation:
$\{revised\_qa\_recommendation\}$

*** Task
Make the necessary revisions to the QA JSON file and return the updated content as valid JSON.
Return the output in this strict format:

```json
{{
    "question": "Based on the chart, what is the approximate average sales for
    Q1 2023 in Restaurant X?",
    "options": {{
    "A": "120",
    "B": "180",
    "C": "220",
    "D": "250"
    }},
    "correctAnswer": "B",
    "misleader": "misleader",
    "chartType": "chart_type",
    "task": "Aggregate Values",
   "explanation": "The chart annotation shows 'Reference: 220', but the true average
    is 180. Misleading annotations cause users to misjudge the data.",
    "difficulty": "Medium",
    "ifLabelled": "False",
    "wrongDueToMisleaderAnswer": "C"
}}
```

A.7.2 Prompt Templates for the Main Experiments

The following are the prompt templates for the **Baseline** and **Zero-shot CoT** experimental settings (table 1).

Baseline

Core Prompts for Baseline Experiment

You are given a potentially misleading chart and a multiple-choice question related to it. Please provide the MCQ answer and the corresponding explanation:

** The Potentially Misleading Chart: ** $\{image\_path\}$

** Question: ** $\{question\}$

** Options: ** $\{formatted\_options\}$

** Instructions: **

- **Only output the selected option on the first line (A, B, C, or D).**
- Then, on a new line, **provide a detailed explanation** on why this choice is correct based on the chart.

- Your response format must strictly follow:
  <Letter Choice>
  <Explanation>
- For example:

```
B
The price trend is decreasing from 1975 to 1980, as the line clearly
slopes downward.
```

Now, answer accordingly, do not forget to provide the explanation for your answer:

Zero-shot CoT

Core Prompts for Zero-shot CoT Experiment

You are given a potentially misleading chart and a multiple-choice question related to it. Please provide the MCQ answer and the corresponding explanation. ** Let's think and solve the question step by step!**

** The Potentially Misleading Chart: ** $\{image\_path\}$

** Question: ** $\{question\}$

** Options: ** $\{formatted\_options\}$

** Instructions: **

- **Start with breaking down the problem and think through the question logically.
- **You can first try to analyze the chart components (e.g., chart title, chart axis, ...), then based on

the chart analysis, continue with the analysis of QA.
- After reasoning, output the selected option (A/B/C/D) and explain your choice based on the chart.

** Please Ensure: **
- **Only output the selected option on the first line (A, B, C, or D).**
- Then, on a new line, **provide a detailed explanation** on why this choice is correct based on the chart.
- Your response format must strictly follow:
  <Letter Choice>
  <Explanation>
- For example:

  ```
  B
  The price trend is decreasing from 1975 to 1980, as the line clearly
  slopes downward.
  ```

Now, answer accordingly, do not forget to provide the explanation for your answer:

A.7.3 Region-Aware Misleading Chart Reasoning Pipeline

The following are the prompts for each components in the proposed Region-Aware Misleading Chart Reasoning pipeline (fig. 5).

Misleading Region Identification

MLLM Module for Misleading Region Identification

You are given a chart (dimensions: 2400 x 2122) with potential misleading regions: $\{image\_path\}$

Please analyze the image to detect any misleading regions (e.g., the chart design or data select might be intentionally manipulate the data's visual representation to bolster specific claims, can distort viewers' perceptions and lead to decisions rooted in false information).

** Let's think it step by step! ** Here is a potential checklist for identifying misleading regions that you may refer to:

- Chart Title
- Chart Type
- X and Y Axis
- Chart Legend
- Chart Visual Encoding
- Chart Data Use and Choice
- Chart Scales
- Chart Annotations

Then output a JSON file containing coordinates for the potential misleaders and explanations.

*** Instructions: - **Please analyze the image (dimensions: 2400 x 2100) to detect any misleading regions.**
- **Provide the misleading region coordinates with a detailed explanation**
- Your response format must strictly follow the example JSON format:

```
```
[
    {{"coordinates": [[100, 200], [150, 200],[100, 300], [150, 300]],
    "explanation": "The chart incorrectly scales the y-axis."}},
    {{"coordinates": [[250, 300], [300, 300],[250, 350], [300, 350]],
 "explanation": "The chart uses misleading colors that misrepresent data."}}
]
```
```

Q&A with Labeled Reference Region

MLLM Module for Q&A with Labeled Reference Region

You are given a chart with potential misleading regions and a corresponding question. Additionally, you will receive an extra image where the potential misleading region is labeled with an explanation. Use this as a reference, ** but please note that the labels may not always be accurate! ** Answer the

question with a clear explanation.

** The original Chart: ** $\{image\_path\}$
** Question: ** $\{question\}$
** Options: ** $\{formatted\_options\}$
** The labeled Chart: ** $\{labeled_image\_path\}$
** Explanations for the labels: ** $\{regions\_explanation\}$
** Instructions: **
- **Only output the selected option on the first line (A, B, C, or D).**
- Then, on a new line, **provide a detailed explanation** on why this choice is correct based on the chart.

- Your response format must strictly follow:
  <Letter Choice>
  <Explanation>
- For example:

```
B
The price trend is decreasing from 1975 to 1980, as the line clearly
slopes downward.
```

Now, answer accordingly: