CHARTSEER: A BENCHMARK FOR INDUCTIVE REASONING OVER CHARTS

Anonymous authors
Paper under double-blind review

ABSTRACT

Chart understanding is an important capability of multimodal large language models (MLLMs) for analyzing and reasoning over data visualizations. Existing chart understanding benchmarks largely focus on deductive reasoning, where the question specifies what relation to evaluate, and typically involve charts with relatively few data points. As a result, they largely overlook the evaluation of inductive reasoning over charts, which requires models to discover latent regularities across sufficient data points without being explicitly given the specific regularity. To address this gap, we introduce CHARTSEER, a benchmark for evaluating inductive reasoning over charts. We construct CHARTSEER from 1,438 data-rich charts collected from diverse public datasets and chart repositories, each paired with its underlying tabular data. Based on the underlying data, we identify and verify latent regularities and construct 3,929 questions covering two complementary capabilities: *Rule Induction* for identifying latent regularities in charts and *Example Inference* for generalizing from these regularities to new cases. We evaluate 14 state-of-the-art MLLMs and compare them with humans, finding that current models remain substantially behind human performance. Further analysis reveals that visual perception and rule induction remain two major bottlenecks. We hope CHARTSEER facilitates future research on the inductive reasoning capabilities of MLLMs for chart understanding. [1]

1 INTRODUCTION

Charts are widely used in scientific papers, business reports, and news articles as a direct visual interface to quantitative information and data patterns (Kafle et al., 2018; Masry et al., 2022). However, manually inspecting and interpreting charts can be time-consuming, motivating the need for machines to understand charts automatically (Kafle et al., 2018; Masry et al., 2022). Automatic chart understanding is therefore an important step toward automated data analysis.

Recent advances in multimodal large language models (MLLMs) have demonstrated strong capabilities in visual understanding tasks (Li et al., 2024; Chen et al., 2024). Given an image and a natural-language question, MLLMs can leverage their visual perception and reasoning capabilities to derive the answer. Since charts are data-rich images, they provide a natural testbed for examining whether MLLMs can move beyond general image understanding toward automated data analysis. Therefore, researchers have extended MLLMs to chart understanding tasks and proposed a series of benchmarks to evaluate their performance.

Early benchmarks such as DVQA (Kafle et al., 2018), FigureQA (Kahou et al., 2017), and PlotQA (Methani et al., 2020) primarily evaluate basic chart understanding through value extraction and comparison tasks. More recent benchmarks, including CharXiv (Wang et al., 2024), InfoChartQA (Xie et al., 2025), and ChartMuseum (Tang et al., 2025), substantially broaden the diversity and visual complexity of charts, covering real-world scientific figures and infographic charts. Despite increasingly diverse chart representations, existing benchmarks still largely operate under a deductive reasoning paradigm, where the specific relation or operation to be applied is specified by the question. Moreover, their charts typically contain relatively few data points, limiting the evidence

[1] All experimental data and code are available at: `https://anonymous.4open.science/r/ChartSeer-2FF0/`.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

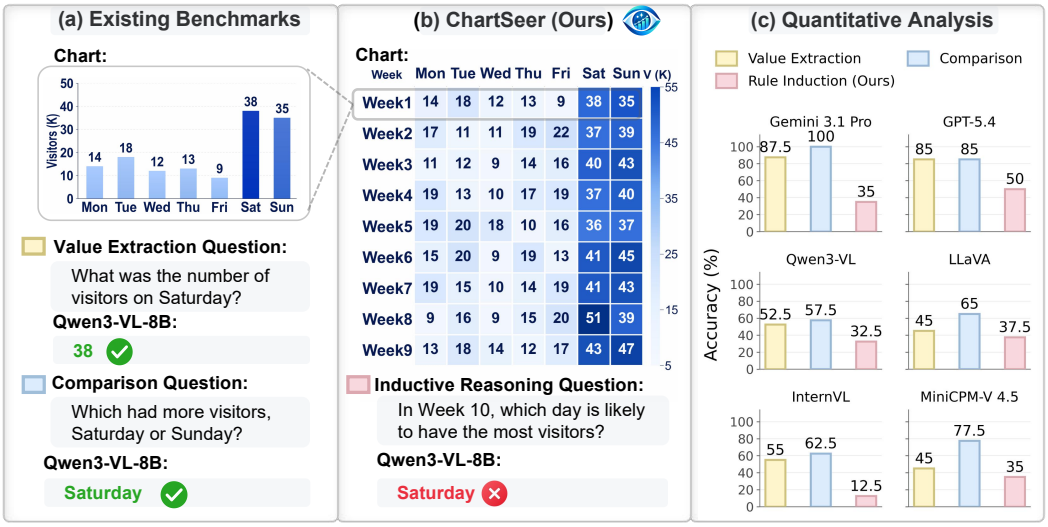


Figure 1: Comparison between representative examples from *(a) existing benchmarks* and *(b)* CHARTSEER *(ours)*, together with *(c) quantitative analysis* of model performance.

available for uncovering latent regularities. Figure 1(a) shows a representative example from existing benchmarks. Given a chart showing weekly visitor counts, an MLLM (i.e., Qwen3-VL-8B) can answer a value extraction question such as "What was the number of visitors on Saturday?" with "38", or a comparison question such as "Which had more visitors, Saturday or Sunday?" with "Saturday", by directly reading and comparing the corresponding bars.

Despite the strong performance of MLLMs on deductive chart reasoning, **their ability to perform inductive reasoning over charts remains largely unexplored**. Figure 1(b) illustrates a chart containing data points from nine weeks. Given this chart, we consider the question: "In week 10, which day is likely to have the most visitors?" To answer this question, a model is expected to identify an empirical pattern such as "Sunday tends to have the highest number of visitors across most weeks" from sufficient data points and extrapolate it to an unseen future period (i.e., week 10). As shown in Figure 1(b), Qwen3-VL-8B incorrectly predicts "Saturday" instead of "Sunday".

To further examine the gap between deductive and inductive chart reasoning, we conduct a small-scale controlled study on 100 systematically sampled charts with underlying numerical data. For each chart, we construct value extraction, comparison, and rule induction questions on the same chart, with further details provided in Appendix B. As shown in Figure 1(c), across six representative MLLMs, the average accuracy reaches 61.7% and 74.6% on value extraction and comparison, respectively, but drops to only 33.8% on rule induction. These results reveal **a substantial gap between deductive and inductive chart reasoning in current MLLMs**.

To systematically evaluate this capability, we introduce CHARTSEER, a benchmark comprising 1,438 data-rich charts collected from diverse public datasets and chart repositories. Each chart is paired with its underlying structured data, which enables us to filter charts based on regularity criteria; the retained charts contain a median of 3,730 data points (9,189 on average). Moreover, CHARTSEER contains 3,929 QA pairs and evaluates two complementary capabilities: inferring latent regularities from observed data (*Rule Induction*) and applying them to new cases (*Example Inference*). All QA pairs are grounded in machine-readable numerical data or verified structured chart metadata, enabling deterministic answer verification. We evaluate 14 state-of-the-art MLLMs and compare their performance with humans. The best-performing models achieve 57.5 on rule induction and 38.1 on example inference, compared with 70.0 and 77.3 for humans, respectively, revealing **a substantial gap between current MLLMs and humans in inductive reasoning over chart data**. To better understand where models fail, we conduct a controlled-input analysis that progressively provides additional information required for reasoning (e.g., tabular data and underlying rules), revealing that **model failures mainly arise in the visual perception and rule induction steps**. These findings provide useful directions for the development of future MLLMs.

Our contributions are summarized as follows:

Category	Dimension	DVQA	ChartQA	CharXiv	Info ChartQA	Chart Museum	ChartSeer (Ours)
Data Source	Real-world data	✗	✓	✓	✓	✓	✓
	Multiple sources	✗	✓	✗	✓	✓	✓
	Source-aligned table	✓	✓̸	✗	✓̸	✗	✓
Chart Style	Plain charts	✓	✓	✓	✓	✓	✓
	Infographic charts	✗	✗	✗	✓	✓	✓
	Multi-panel charts	✗	✗	✓	✓	✓	✓
Chart Regularity	Multiple families	✗	✗	✗	✓̸	✓̸	✓
	Explicit annotation	✗	✗	✗	✗	✗	✓
	Statistical verification	✗	✗	✗	✗	✗	✓
QA Design	Regularity-grounded	✗	✓̸	✓̸	✓̸	✓̸	✓
	Rule induction	✗	✗	✓̸	✓̸	✓̸	✓
	Example inference	✗	✗	✗	✗	✗	✓

Table 1: Comparison between ChartSeer and representative chart-understanding benchmarks. ✓: explicit coverage; ✓̸: indirect coverage (e.g., CharXiv includes trend-description questions, but does not ground QA design in statistically verified regularities); ✗: no coverage.

- We introduce ChartSeer, a benchmark for evaluating inductive reasoning over charts, a capability largely overlooked by existing chart-understanding benchmarks.
- We curate charts with sufficient data points and clear regularities, and use rule induction and example inference to evaluate two complementary inductive reasoning capabilities over charts.
- Extensive experiments on 14 state-of-the-art MLLMs reveal substantial limitations in their ability to perform robust inductive reasoning over charts.

2 Related Work

Multimodal reasoning. Recent MLLMs have demonstrated strong reasoning capabilities over visual inputs (OpenAI, 2026; Li et al., 2024; Chen et al., 2024). General multimodal reasoning benchmarks, such as MathVista (Lu et al., 2024) and MMMU (Yue et al., 2024), evaluate mathematical and logical reasoning across diverse visual inputs. Existing multimodal reasoning studies largely focus on *deductive reasoning*, where models derive conclusions from explicitly provided visual evidence through operations such as comparison, arithmetic computation, and multi-step logical inference. In contrast, *inductive reasoning* requires models to infer general patterns or rules from observed examples and apply them to new cases (Li et al., 2025). Several recent studies (Mirchandani et al., 2023; Gendron et al., 2023; Chia et al., 2024) have begun to evaluate such capabilities in MLLMs, but primarily through abstract visual puzzles and symbolic patterns tasks. Consequently, inductive reasoning over structured visual inputs such as charts remains less systematically studied.

Chart understanding. Chart understanding is an important multimodal reasoning task that involves answering natural language questions based on charts (Yang et al., 2025). Early benchmarks such as DVQA (Kafle et al., 2018) and FigureQA (Kahou et al., 2017) mainly focus on visual attribute recognition and value extraction from chart elements. Subsequent datasets, including PlotQA (Methani et al., 2020) and ChartQA (Masry et al., 2022), introduce reasoning operations such as comparison, arithmetic computation, and extremum detection over observed chart values. More recent benchmarks, such as ChartBench (Xu et al., 2023), ChartX (Xia et al., 2025), InfographicVQA (Mathew et al., 2022), and CharXiv (Wang et al., 2024), further broaden chart understanding to more diverse visualizations and more complex reasoning tasks. Despite these advances, existing chart benchmarks still largely follow a deductive reasoning paradigm, where the relation or operation to be applied is specified by the question. As summarized in Table 1, existing benchmarks generally lack explicit and statistically verified chart regularities, and do not systematically construct regularity-grounded questions for both rule induction and example inference.

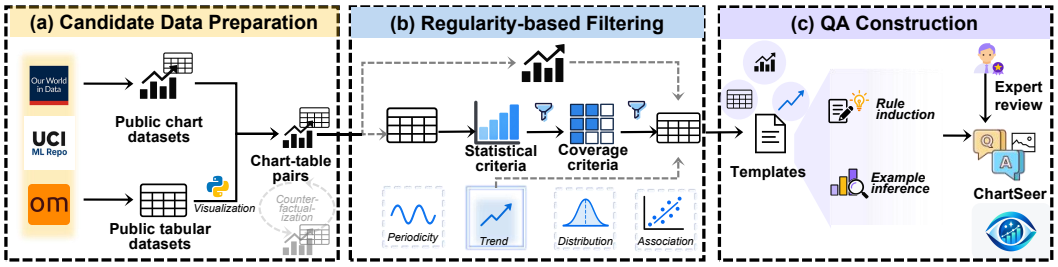


Figure 2: The CHARTSEER benchmark construction pipeline.

3 CHARTSEER BENCHMARK

Types of regularities. To evaluate MLLMs' inductive reasoning in chart understanding, we consider four common categories of data regularities identified in prior visualization and exploratory data analysis studies (Law et al., 2020; Demiralp et al., 2017). (1) *Trend* describes a systematic directional or structural change in values along an ordered axis. (2) *Periodicity* describes a pattern of variation that recurs across successive intervals. (3) *Distribution* describes how values are concentrated, dispersed, or shaped within a data series. (4) *Association* describes systematic dependence or co-variation between two or more data series.

Definition of inductive chart reasoning. We define inductive chart reasoning as inferring a chart-specific regularity from multiple data points and reasoning based on that regularity (e.g., inferring an increasing trend from data points exhibiting an overall upward pattern). Based on this definition, rule induction evaluates whether a model can identify or verify the chart-specific regularity (e.g., determining whether the chart exhibits an increasing trend), while example inference evaluates whether it can generalize the inferred regularity to a new case (e.g., predicting a future value based on the increasing trend).

CHARTSEER construction pipeline. As shown in Figure 2, we construct CHARTSEER in three stages: candidate data preparation, regularity-based filtering, and question-answer construction. We first collect data from diverse public sources and process them into aligned chart–table pairs. We then identify chart-specific regularities from the underlying data using predefined statistical and coverage criteria, retaining candidates with clear and well-supported regularities. Finally, we construct rule induction and example inference questions based on the verified regularities and perform programmatic and human quality control. The resulting benchmark contains 1,438 charts and 3,929 question-answer pairs. Detailed construction procedures are provided in Appendix C.

3.1 CANDIDATE DATA COLLECTION

Data sources. To construct data-rich charts with verifiable regularities, we prioritize sources that contain sufficient data points and provide accessible underlying tabular data. We collect candidate data from 10 source components spanning diverse domains and data characteristics, including Open-Meteo (Zippenfenig, 2023), the UCI Machine Learning Repository (UC Irvine, 2026), the UCR/UEA time-series archives (Dau et al., 2019; Bagnall et al., 2018), Our World in Data (OWID) (Our World in Data, 2026), and ChartQA (Masry et al., 2022). A complete list of data sources is provided in Appendix C.1.1.

Candidate selection. After collecting data from the selected sources, we retain candidates with valid underlying tables and sufficient numerical information for subsequent analysis. For existing chart datasets, we further require reliable alignment between each chart and its paired table or structured metadata, and remove duplicate candidates based on source identifiers and underlying data records. Detailed selection criteria and statistics are provided in Appendix C.1.2.

Chart-table construction. For sources with reliably aligned chart images and underlying tables, we preserve the original visualizations. For tabular-only sources, we render the selected data using predefined visualization templates matched to their data schemas (e.g., line, bar, histogram, and

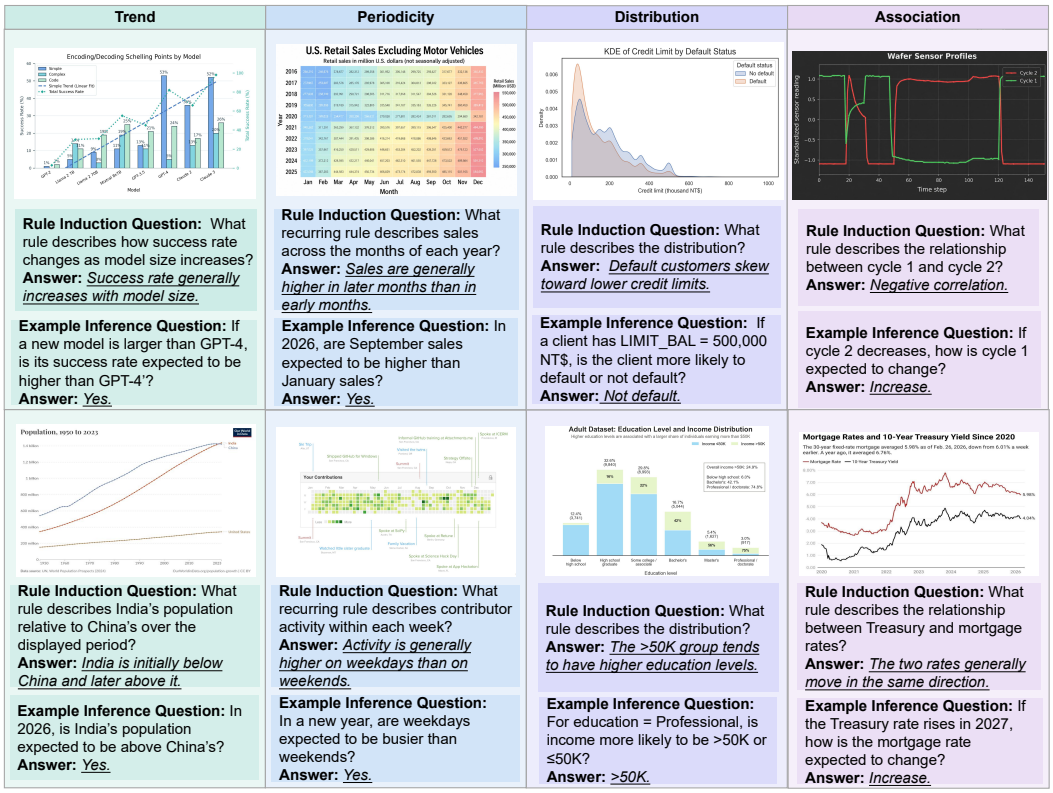


Figure 3: Representative examples from CHARTSEER.

grouped bar charts), while balancing chart types to avoid over-representation of particular visual forms. Detailed visualization templates and chart-type statistics are provided in Appendix C.1.5.

Counterfactual construction. To mitigate potential memorization of public-source data, we construct a counterfactual subset by first identifying potentially memorized charts through masked reconstruction (Deng et al., 2024), and then modifying their underlying values or labels while preserving the data structure and target regularity (Park et al., 2026). The modified charts are re-rendered accordingly, with detailed transformation rules provided in Appendix C.2.4.

3.2 REGULARITY-BASED FILTERING

Statistical criteria. Following prior taxonomies of data facts and visual insights (Law et al., 2020; Demiralp et al., 2017), we define subtype-specific statistical criteria to identify candidate regularities. Each subtype is evaluated using corresponding statistical measures and thresholds (e.g., recurring cycles are evaluated using seasonal strength and cycle similarity).

Coverage criteria. A statistically strong regularity may still be difficult to infer if it is supported by insufficient visible evidence. We therefore impose subtype-specific coverage requirements to ensure that each retained regularity is sufficiently manifested in the chart (e.g., a recurring periodic pattern must span at least two complete cycles).

Filtering based on regularity criteria. For each chart–table pair, we test applicable candidate regularities against the corresponding statistical and coverage criteria, retaining only those that are both sufficiently strong and well supported by the visible data. When multiple regularities satisfy the criteria, we retain the chart only if a single primary regularity can be determined unambiguously. Each retained instance forms a chart–table–regularity triple for subsequent QA construction. Complete definitions of the criteria and filtering procedures are provided in Appendix C.1.6.

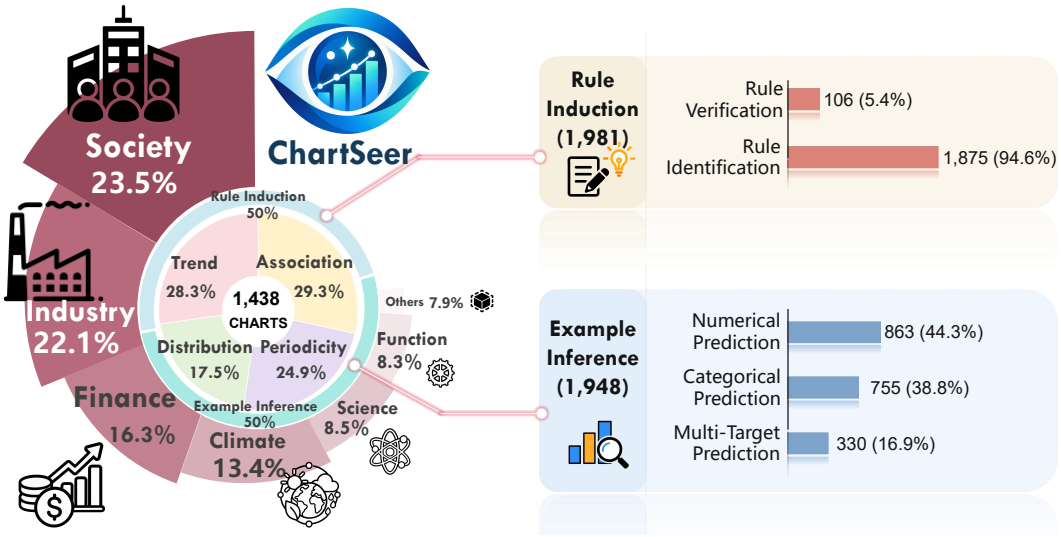


Figure 4: Statistics of the CHARTSEER benchmark.

3.3 QA CONSTRUCTION

Rule induction questions. For each retained chart–table–regularity triple, we construct rule induction questions from the verified target regularity. We consider two question types: *Rule Verification*, which asks whether a candidate formulation of the target regularity holds, and *Rule Identification*, which asks the model to select the chart-supported rule from a set of candidates. For rule identification, distractors are generated from the target regularity and underlying data by varying its direction, parameters, phase, ordering, dependency, or scope.

Example inference questions. Using the same verified regularity, we construct example inference questions that query its consequence in a new case. The ground-truth answer is derived from the verified regularity. We consider three question types: *Numerical Prediction*, *Categorical Prediction*, and *Multi-Target Prediction*. Representative matched examples are shown in Figure 3, with complete templates and instantiation rules provided in Appendix C.2.2.

Quality control. All QAs are programmatically verified against the underlying data and target regularities, and all charts are manually inspected to ensure that the intended regularity is clearly supported by the visible evidence without obvious ambiguity. We further conduct a stratified semantic audit of 400 QA pairs with three expert annotators. QA instances with identified issues, along with the corresponding templates that generate them, are removed from the benchmark. Detailed procedures and results are provided in Appendix D.

3.4 BENCHMARK STATISTICS

Figure 4 summarizes the composition of CHARTSEER. The benchmark contains 1,438 charts spanning four regularity categories and diverse application domains. The four regularities are relatively balanced, accounting for 17.5%–29.3% of the charts, while the domains cover society, industry, finance, climate, science, function-based data, and others.

CHARTSEER contains 3,929 question-answer pairs, nearly evenly split between rule induction (1,981) and example inference (1,948). Within rule induction, 1,875 questions are Rule Identification and 106 are Rule Verification. Example inference includes 863 Numerical Prediction, 755 Categorical Prediction, and 330 Multi-Target Prediction questions, covering both direct numerical answers and option-based predictions.

We partition the benchmark at the chart level into a development set of 288 charts with 780 questions and a test set of 1,150 charts with 3,149 questions. Charts are stratified jointly by regularity and source, and all questions associated with the same chart are kept in the same split. Detailed statistics on data sources, chart types, question templates, and data splits are provided in Appendix E.

Model	Rule Induction					Example Inference				
	All	Trend	Period.	Distrib.	Assoc.	All	Trend	Period.	Distrib.	Assoc.
Baselines										
Text-only (GPT-5.4 (OpenAI, 2026))	22.0	26.0	3.7	29.3	28.9	18.6	3.2	2.3	44.1	25.0
Proprietary Models										
Claude Opus 4.6 (Anthropic, 2026a)	46.2	77.8	21.9	35.9	49.1	27.4	32.8	23.1	22.1	31.6
Claude Sonnet 4.6 (Anthropic, 2026b)	53.3	79.1	21.9	47.4	64.7	26.2	35.1	19.9	26.9	23.0
Gemini 3.1 Flash-Lite (Google DeepMind, 2026a)	50.1	73.2	30.5	40.0	56.9	32.9	43.4	4.1	33.0	**51.0**
Gemini 3.1 Pro (Google DeepMind, 2026b)	44.3	62.5	21.1	23.3	**70.4**	**38.1**	**44.4**	15.2	**59.0**	33.7
GPT-4o (OpenAI, 2024)	39.3	61.2	10.4	31.1	54.4	27.2	21.4	7.0	38.2	42.3
GPT-5.4 (OpenAI, 2026)	53.3	75.6	24.6	**51.1**	61.9	37.7	39.5	11.1	54.8	45.4
o3 (OpenAI, 2025)	**57.5**	**80.2**	**46.8**	46.3	56.9	29.1	32.4	**40.1**	35.1	8.7
o4-mini (OpenAI, 2025)	56.2	77.5	42.8	47.4	57.1	27.8	32.0	38.0	34.7	6.6
Open-Source Models										
InternVL3-8B (Chen et al., 2024)	40.6	**80.6**	12.8	37.4	31.7	31.5	23.9	19.3	51.7	31.1
LLaVA-OV-1.5-8B (Li et al., 2024)	44.4	70.2	23.0	42.2	42.0	29.8	23.8	23.1	42.2	30.1
ChartGemma (Masry et al., 2025)	26.9	32.1	24.1	33.0	18.6	18.9	0.4	23.4	30.0	21.9
MiniCPM-V 4.5 (Yu et al., 2025)	46.9	69.1	**32.9**	**43.3**	42.4	28.9	24.1	16.4	50.0	25.0
TinyChart (Zhang et al., 2024)	27.7	29.3	25.4	33.0	23.2	15.5	6.7	**25.7**	4.6	25.0
Qwen3-VL-8B (Bai et al., 2025)	**48.9**	65.1	31.8	36.7	**62.2**	**33.6**	**30.5**	16.7	**52.5**	**34.7**

Table 2: Evaluation results on CHARTSEER. Bold numbers represent the best in-class performance (open-source or proprietary), and underlined numbers represent the second-best performance. Period.=periodicity, Distrib.=distribution, Assoc.=association.

4 EXPERIMENTS

4.1 MODEL COMPARISON

Models. We evaluate a diverse set of open-source and proprietary models on CHARTSEER. For proprietary models, we include Claude Opus 4.6 (Anthropic, 2026a) and Claude Sonnet 4.6 (Anthropic, 2026b), Gemini 3.1 Flash-Lite (Google DeepMind, 2026a) and Gemini 3.1 Pro (Google DeepMind, 2026b), GPT-4o (OpenAI, 2024), GPT-5.4 (OpenAI, 2026), and o3 and o4-mini (OpenAI, 2025). For open-source models, we include InternVL3-8B (Chen et al., 2024), LLaVA-OneVision-1.5-8B-Instruct (Li et al., 2024), ChartGemma (Masry et al., 2025), MiniCPM-V 4.5 (Yu et al., 2025), TinyChart (Zhang et al., 2024), and Qwen3-VL-8B-Instruct (Bai et al., 2025). Model checkpoints and inference configurations are provided in Appendix I.

Baselines. We include a text-only baseline, denoted as Text-only (GPT-5.4), in which GPT-5.4 is prompted to make a reasonable guess without access to the chart images (see the prompt in Appendix J). This baseline estimates the extent to which models can answer correctly by relying only on common sense or shallow textual cues in the questions. We recruited 10 human participants with expertise in deep learning and visualization to complete the human evaluation (see Appendix L). The evaluation contains 300 human QA responses, with each QA independently answered by two participants. Participants use a fixed chart, question, option order, and answer format, and their responses are evaluated with the same task metrics.

Evaluation metrics. Following prior chart question-answering benchmarks (Wang et al., 2024; Tang et al., 2025; Xie et al., 2025), we report accuracy for all tasks. Multiple-choice questions, including rule verification, rule identification, categorical prediction, and multi-target prediction, use exact match accuracy on the normalized option label. Numerical prediction uses relaxed accuracy following ChartQA (Masry et al., 2022), with each requested value required to be parseable and within a 5% relative error tolerance of the reference answer.

4.2 EXPERIMENTAL RESULTS

Table 2 summarizes the performance of all evaluated MLLMs on CHARTSEER. The task-to-evaluation mapping is provided in Appendix C.2.1. Key observations include:

Chart (Association)



Rule Induction Task

Question: Which option best describes the relationship between the two housing-affordability ratios?

Answer Options in Natural Language

A) The two ratios show no clear correlation from Dec 1987 to Oct 2003, but are positively correlated from Oct 2003 to May 2021.

B) The two ratios are positively correlated from Dec 1987 to Dec 2006, negatively correlated from Dec 2006 to Sep 2011, and positively correlated again from Sep 2011 to May 2021.

C) The mortgage-payment-to-income ratio and the average home-price-to-income ratio are broadly positively correlated overall.

D) The two ratios show no clear correlation from Dec 1987 to Oct 2003, are positively correlated from Oct 2003 to Sep 2011, and then show no clear correlation from Sep 2011 to May 2021.

o3's Reasoning Process

… between 2001 and 2006, both lines rise sharply, **indicating a positive correlation**. From 2006 to 2011, both lines drop drastically, **still maintaining a positive correlation** despite individual trends.

… Option B suggests a negative correlation from 2006 to 2011, but since both ratios fell during that time, **it's actually positive**. Option C states there's a broad positive correlation, **which seems plausible**.

However, Option D appears strongest. This matches my observations about the data, so I think I'll go with Option D.

Figure 5: Representative case of over-specificity by o3 in CHARTSEER.

All models remain far below human performance. As shown in Table 2, all evaluated MLLMs struggle with inductive reasoning over charts. Among proprietary models, o3 achieves the best rule induction performance at 57.5, while Gemini 3.1 Pro achieves the best example inference performance at 38.1. In comparison, humans achieve 70.0 on rule induction and 77.3 on example inference. Among open-source models, Qwen3-VL-8B performs best, reaching 48.9 on rule induction and 33.6 on example inference, but still falls far short of human performance.

Models perform better on simple regularities. Models generally perform better on relatively simple regularities, such as trend, while showing lower performance on more complex regularities, particularly periodicity. Averaged over both task formats, proprietary and open-source models achieve 54.3 and 38.0 on trend, compared with only 23.7 and 22.9 on periodicity, respectively. Distribution and association show intermediate performance, with average accuracies of 39.1 and 44.6 for proprietary models and 38.1 and 32.3 for open-source models. This may be because association requires jointly tracking changes across multiple variables, while distribution requires aggregating evidence across many data points or bins, making their underlying regularities less visually direct than simple trends.

Example inference is more challenging for current models. Overall, example inference is more challenging than rule induction, with average accuracies dropping from 50.0 to 30.8 for proprietary models and from 39.2 to 26.4 for open-source models. This performance gap is especially pronounced for trend, periodicity, and association, where nearly all models perform worse on example inference. This may be because example inference requires models to first infer the underlying regularity and then apply it to an unseen case, introducing an additional reasoning step. Distribution is the main exception, where some models even perform better on example inference. We attribute this to the fact that distribution-based inference often involves situating a new instance within a known distribution, rather than extrapolating an observed pattern to predict an unknown future state. A controlled paired analysis in Appendix H further shows that models often fail on example inference even after correctly identifying the corresponding rule.

4.3 ERROR ANALYSIS

Representative error. Figure 5 shows a representative case of over-specificity by o3 on association rule induction. The model correctly identifies several intervals in which the two series move in the same direction and even considers the broad positive association in Option C plausible. However,

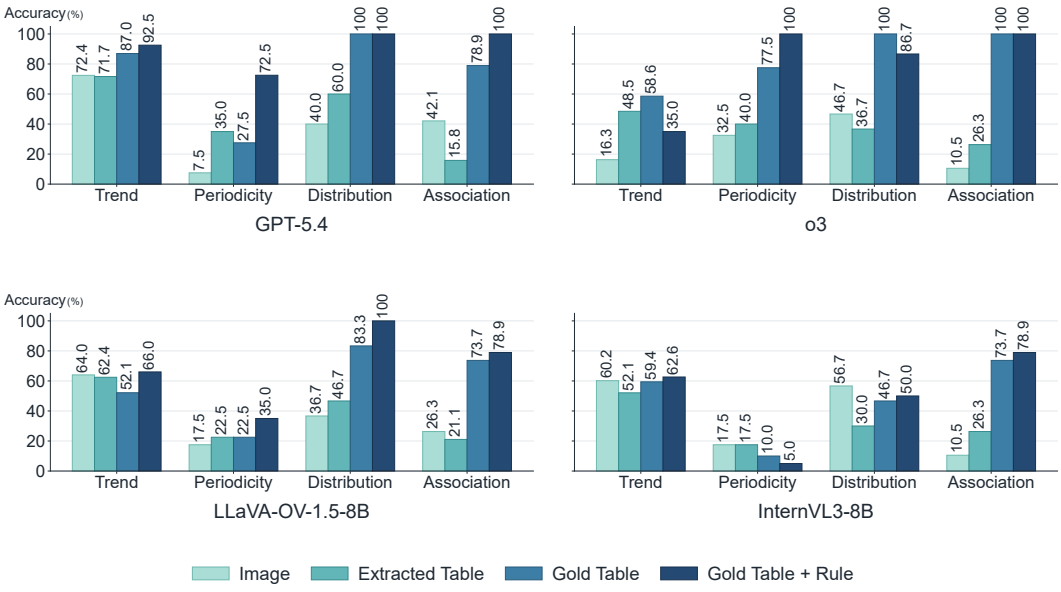


Figure 6: Bottleneck analysis of inductive chart reasoning across four representative models.

it ultimately selects Option D by placing greater weight on local fluctuations and short intervals of weak co-movement. This reflects over-specificity in inductive reasoning: the model incorporates local variations into the inferred rule instead of abstracting the dominant pattern. Thus, although o3 successfully extracts much of the relevant evidence, it fails to generalize it into the appropriate chart-level regularity. Inspired by prior taxonomies of inductive reasoning errors (Hua et al., 2025), we categorize this failure as over-specificity and further group 20 selected failure cases into four recurring mechanisms in Appendix K.

Bottleneck analysis. To identify the sources of difficulty in inductive chart reasoning, we compare four controlled input conditions while keeping the question, answer format, and evaluation criterion fixed. *Image* uses the original chart as input. *Extracted Table* uses a table extracted from the original chart by the model, *Gold Table* replaces the extracted table with the ground-truth table, and *Gold Table + Rule* further provides the reference rule together with the ground-truth table. Detailed prompts are provided in Appendix J.

Figure 6 shows that providing the gold table substantially improves performance for most representative models, especially GPT-5.4 and o3, suggesting that chart perception and reliable information extraction are major sources of difficulty. However, performance remains imperfect with the ground-truth table, pointing to additional challenges beyond perception. Providing the reference rule further improves GPT-5.4 and LLaVA-OV-1.5-8B, indicating that inducing the underlying regularity is another important bottleneck, although the effect is model-dependent as o3 slightly decreases. Overall, these results suggest that inductive chart reasoning is constrained by chart perception and induction.

5 Conclusion

In this paper, we introduced ChartSeer, a benchmark for evaluating inductive reasoning over charts. Unlike existing chart-understanding benchmarks that largely focus on deductive reasoning over observed values, ChartSeer evaluates whether MLLMs can infer latent regularities from chart data and apply them to new cases through two complementary tasks, rule induction and example inference. Experiments on 14 state-of-the-art MLLMs show that current models remain substantially behind human performance on both tasks. Our controlled analysis further reveals that visual perception and rule induction are major bottlenecks in inductive chart reasoning. These results demonstrate that discovering and applying latent regularities from charts remains a significant challenge for current MLLMs, and we hope ChartSeer can support future progress toward more capable chart reasoning and automated data analysis.

AI USE STATEMENT

In this work, we used generative AI tools to assist in designing and providing feedback on the benchmark methodology and experiments, implementing methods, and cleaning and reformatting datasets. We have not used generative AI tools for translation or result interpretation, and the remaining required-disclosure tasks are not applicable to this work. Additionally, we used generative AI tools to edit software code and improve the readability of the manuscript. We have reviewed all AI-assisted work. AI-assisted code was executed and tested by the authors, and benchmark QAs were subject to automatic checks and manual quality control. We take responsibility for the final content of this work, including text, claims, or artifacts produced with the aid of generative AI.

ETHICS STATEMENT

CHARTSEER is constructed from publicly available datasets and chart repositories, which are used in accordance with their respective licenses and terms of use. Our human evaluation involved 10 participants with expertise in deep learning and visualization, who voluntarily completed chart question-answering tasks for evaluating human performance on the benchmark. Participants were informed of the purpose of the study, and their responses were used only for this evaluation and analyzed in aggregate. No sensitive personal information was collected or released.

REPRODUCIBILITY STATEMENT

We provide the benchmark data, construction code, and evaluation code through the anonymous repository linked in the paper. The appendix documents the data sources, benchmark construction and filtering procedures, QA generation and verification, model checkpoints and inference configurations, evaluation prompts, metrics, and human-evaluation protocol. These materials are provided to support reproduction of both the benchmark construction process and the reported experimental results.

REFERENCES

Anthropic. Claude Opus 4.6 System Card, February 2026a. URL https://www.anthropic.com/claude-opus-4-6-system-card. System card.

Anthropic. Claude Sonnet 4.6 System Card, February 2026b. URL https://www.anthropic.com/claude-sonnet-4-6-system-card. System card.

Anthony Bagnall, Hoang Anh Dau, Jason Lines, Michael Flynn, James Large, Aaron Bostrom, Paul Southam, and Eamonn Keogh. The uea multivariate time series classification archive, 2018. *arXiv preprint arXiv:1811.00075*, 2018.

Shuai Bai, Yuxuan Cai, Ruizhe Chen, Keqin Chen, Xionghui Chen, Zesen Cheng, Lianghao Deng, Wei Ding, Chang Gao, Chunjiang Ge, et al. Qwen3-vl technical report. *arXiv preprint arXiv:2511.21631*, 2025.

Barry Becker and Ronny Kohavi. Adult. UCI Machine Learning Repository, 1996. URL https://archive.ics.uci.edu/dataset/2/adult.

R. Bock. Magic gamma telescope. UCI Machine Learning Repository, 2004. URL https://archive.ics.uci.edu/dataset/159/magic+gamma+telescope.

Zhe Chen, Jiannan Wu, Wenhai Wang, Weijie Su, Guo Chen, Sen Xing, Muyan Zhong, Qinglong Zhang, Xizhou Zhu, Lewei Lu, et al. Internvl: Scaling up vision foundation models and aligning for generic visual-linguistic tasks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 24185–24198, 2024.

Yew Ken Chia, Vernon Toh, Deepanway Ghosal, Lidong Bing, and Soujanya Poria. Puzzlevqa: Diagnosing multimodal reasoning challenges of language models with abstract visual patterns. In *Findings of the Association for Computational Linguistics: ACL 2024*, pp. 16259–16273, 2024.

Hoang Anh Dau, Anthony Bagnall, Kaveh Kamgar, Chin-Chia Michael Yeh, Yan Zhu, Shaghayegh Gharghabi, Chotirat Ann Ratanamahatana, and Eamonn Keogh. The ucr time series archive. *IEEE/CAA Journal of Automatica Sinica*, 6(6):1293–1305, 2019.

Çağatay Demiralp, Peter J Haas, Srinivasan Parthasarathy, and Tejaswini Pedapati. Foresight: Rapid data exploration through guideposts. *arXiv preprint arXiv:1709.10513*, 2017.

Chunyuan Deng, Yilun Zhao, Xiangru Tang, Mark Gerstein, and Arman Cohan. Investigating data contamination in modern benchmarks for large language models. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pp. 8706–8719, 2024.

Gaël Gendron, Qiming Bao, Michael Witbrock, and Gillian Dobbie. Large language models are not strong abstract reasoners. *arXiv preprint arXiv:2305.19555*, 2023.

Google DeepMind. Gemini 3.1 flash-lite - model card. `https://deepmind.google/models/model-cards/gemini-3-1-flash-lite/`, 2026a.

Google DeepMind. Gemini 3.1 pro - model card. `https://deepmind.google/models/model-cards/gemini-3-1-pro/`, 2026b.

Wenyue Hua, Tyler Wong, Fei Sun, Liangming Pan, Adam Jardine, and William Yang Wang. Inductionbench: Llms fail in the simplest complexity class. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 26526–26546, 2025.

Kushal Kafle, Brian Price, Scott Cohen, and Christopher Kanan. Dvqa: Understanding data visualizations via question answering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 5648–5656, 2018.

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

Po-Ming Law, Alex Endert, and John Stasko. Characterizing automated data insights. In *2020 IEEE Visualization Conference (VIS)*, pp. 171–175. IEEE, 2020.

Bo Li, Yuanhan Zhang, Dong Guo, Renrui Zhang, Feng Li, Hao Zhang, Kaichen Zhang, Peiyuan Zhang, Yanwei Li, Ziwei Liu, et al. Llava-onevision: Easy visual task transfer. *arXiv preprint arXiv:2408.03326*, 2024.

Jiachun Li, Pengfei Cao, Zhuoran Jin, Yubo Chen, Kang Liu, and Jun Zhao. Mirage: Evaluating and explaining inductive reasoning process in language models. In *International Conference on Learning Representations*, volume 2025, pp. 73944–73969, 2025.

Pan Lu, Hritik Bansal, Tony Xia, Jiacheng Liu, Chunyuan Li, Hannaneh Hajishirzi, Hao Cheng, Kai-Wei Chang, Michel Galley, and Jianfeng Gao. Mathvista: Evaluating mathematical reasoning of foundation models in visual contexts. In *International Conference on Learning Representations*, volume 2024, pp. 23439–23554, 2024.

JR Maat, A Malali, and P Protopapas. Timesynth: A multipurpose library for synthetic time series in python. *URL: http://github. com/TimeSynth/TimeSynth*, 2017.

Ahmed Masry, Xuan Long Do, Jia Qing Tan, Shafiq Joty, and Enamul Hoque. Chartqa: A benchmark for question answering about charts with visual and logical reasoning. In *Findings of the association for computational linguistics: ACL 2022*, pp. 2263–2279, 2022.

Ahmed Masry, Megh Thakkar, Aayush Bajaj, Aaryaman Kartha, Enamul Hoque, and Shafiq Joty. Chartgemma: Visual instruction-tuning for chart reasoning in the wild. In *Proceedings of the 31st International Conference on Computational Linguistics: Industry Track*, pp. 625–643, 2025.

Minesh Mathew, Viraj Bagal, Rubèn Tito, Dimosthenis Karatzas, Ernest Valveny, and CV Jawahar. Infographicvqa. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pp. 1697–1706, 2022.

Nitesh Methani, Pritha Ganguly, Mitesh M Khapra, and Pratyush Kumar. Plotqa: Reasoning over scientific plots. In *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, pp. 1527–1536, 2020.

Suvir Mirchandani, Fei Xia, Pete Florence, Brian Ichter, Danny Driess, Montserrat Gonzalez Arenas, Kanishka Rao, Dorsa Sadigh, and Andy Zeng. Large language models as general pattern machines. *arXiv preprint arXiv:2307.04721*, 2023.

National Institute of Standards and Technology. NIST Digital Library of Mathematical Functions. https://dlmf.nist.gov/, n.d.

OpenAI. GPT-4o system card. https://openai.com/index/gpt-4o-system-card/, 2024.

OpenAI. OpenAI o3 and o4-mini system card. https://openai.com/index/o3-o4-mini-system-card/, 2025.

OpenAI. Introducing GPT-5.4. https://openai.com/index/introducing-gpt-5-4/, 2026.

Our World in Data. Our world in data. https://ourworldindata.org, 2026.

Jaden Park, Mu Cai, Feng Yao, Jingbo Shang, Soochahn Lee, and Yong Jae Lee. Contamination detection for vlms using multi-modal semantic perturbations. In *International Conference on Learning Representations*, volume 2026, pp. 95733–95754, 2026.

Joanna Purich, Arjun Srinivasan, Michael Correll, Leilani Battle, Vidya Setlur, and Anamaria Crisan. Toward a scalable census of dashboard designs in the wild: A case study with tableau public, 2023.

Tableau. Tableau Public. https://public.tableau.com/, 2026. URL https://public.tableau.com/.

Liyan Tang, Grace Kim, Xinyu Zhao, Thom Lake, Wenxuan Ding, Fangcong Yin, Prasann Singhal, Manya Wadhwa, Zeyu Liu, Zayne Sprague, Ramya Namuduri, Bodun Hu, Juan Diego Rodriguez, Puyuan Peng, and Greg Durrett. Chartmuseum: Testing visual reasoning capabilities of large vision-language models. In *Advances in Neural Information Processing Systems*, volume 38, 2025.

UC Irvine. Uci machine learning repository, 2026. URL https://archive.ics.uci.edu/. Source catalogue accessed September 23, 2026.

Zirui Wang, Mengzhou Xia, Luxi He, Howard Chen, Yitao Liu, Richard Zhu, Kaiqu Liang, Xindi Wu, Haotian Liu, Sadhika Malladi, et al. Charxiv: Charting gaps in realistic chart understanding in multimodal llms. *Advances in Neural Information Processing Systems*, 37:113569–113697, 2024.

Renqiu Xia, Hancheng Ye, Xiangchao Yan, Qi Liu, Hongbin Zhou, Zijun Chen, Botian Shi, Junchi Yan, and Bo Zhang. Chartx & chartvlm: A versatile benchmark and foundation model for complicated chart reasoning. *IEEE Transactions on Image Processing*, 2025.

Tianchi Xie, Minzhi Lin, Mengchen Liu, Yilin Ye, Changjian Chen, and Shixia Liu. InfochartQA: A benchmark for multimodal question answering on infographic charts. In *Advances in Neural Information Processing Systems*, volume 38, 2025.

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

Cheng Yang, Chufan Shi, Yaxin Liu, Bo Shui, Junjie Wang, Mohan Jing, Linran Xu, Xinyu Zhu, Siheng Li, Yuxiang Zhang, Gongye Liu, Xiaomei Nie, Deng Cai, and Yujiu Yang. ChartMimic: Evaluating LMM's cross-modal reasoning capability via chart-to-code generation. In *International Conference on Learning Representations*, volume 2025, pp. 26590–26646, 2025.

I-Cheng Yeh. Default of credit card clients. UCI Machine Learning Repository, 2009. URL https://archive.ics.uci.edu/dataset/350/default+of+credit+card+clients.

Tianyu Yu, Zefan Wang, Chongyi Wang, Fuwei Huang, Wenshuo Ma, Zhihui He, Tianchi Cai, Weize Chen, Yuxiang Huang, Yuanqian Zhao, et al. Minicpm-v 4.5: Cooking efficient mllms via architecture, data, and training recipe. *arXiv preprint arXiv:2509.18154*, 2025.

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

Liang Zhang, Anwen Hu, Haiyang Xu, Ming Yan, Yichen Xu, Qin Jin, Ji Zhang, and Fei Huang. Tinychart: Efficient chart understanding with visual token merging and program-of-thoughts learning. *arXiv preprint arXiv:2404.16635*, 2024.

Patrick Zippenfenig. Open-meteo. com weather api. *Zenodo*, 2023.

Part I
Appendix

Table of Contents

A Why Do We Study Inductive Reasoning on Charts? 15

B Motivating Study 15

C Benchmark Construction Details 17
C.1 Candidate Data Preparation 17
C.2 QA Construction 22

D Quality Control 28

E Detailed Benchmark Statistics and Data Split 29
E.1 Chart and QA Composition 30
E.2 Comparison with Existing Chart Benchmarks 31
E.3 Data Split Protocol 32

F Bottleneck Analysis Details 32
F.1 Intervention Design 32
F.2 More Experimental Results 33

G Robustness and Shortcut Analyses 35
G.1 Cross-Group Robustness 35
G.2 Shortcut Controls 36

H From Rule Induction to Example Inference 37

I Evaluation Details 37

J Evaluation Prompts 38

K Failure Cases 43
K.1 Perception Error 43
K.2 Over-generalization 48
K.3 Over-specificity 52
K.4 Incorrect Rule Identification 58

L More Detailed Information for Human Evaluation 63
L.1 Human-Study and Model-Prediction Alignment 63

M Data-Annotation Interface Examples 63

A WHY DO WE STUDY INDUCTIVE REASONING ON CHARTS?

Practical significance in real-world data analysis. Existing chart understanding benchmarks mainly focus on numerical extraction and deductive reasoning tasks, such as retrieving values, comparing explicit quantities, or answering fact-based questions grounded in charts. While these tasks evaluate whether models can accurately interpret visualized data, they do not assess whether models can infer the latent regularities underlying the data. In real-world scenarios, however, charts often exhibit meaningful patterns such as trends, periodicity, distributions, and associations. Recognizing such regularities is essential for higher-level analytical reasoning, including forecasting future stock trends, summarizing patterns from scientific observations, and anticipating potential anomalies.

Figure 7 illustrates rule identification followed by application to a query. In the released benchmark, Rule Induction (RI) evaluates chart-conditioned rule verification or selection: the question can name the relation family, variables, or candidate rule, and the model must judge which formulation the chart supports.

Example Inference (EI) evaluates whether a model can apply an inferred regularity to a queried target case. Depending on the template, the target can be a future point, a supplied instance, a transformed case, or a candidate outcome whose consistency with the inferred regularity must be determined. These forms require reasoning from the chart-supported regularity rather than directly reading an isolated value.

B MOTIVATING STUDY

Overview. We conduct a controlled diagnostic study to examine whether the performance gap between direct chart reading and rule induction persists across different data regularities. The study follows the same collect-then-filter order as the full benchmark construction: we first assemble candidate data from source families considered in CHARTSEER, then inspect the underlying numerical values and metadata to retain eligible charts, and only afterward sample the diagnostic set. We perform stratified sampling over four common regularity categories—*trend*, *periodicity*, *distribution*, and *association*—selecting 25 charts from each category for a total of 100 charts. For every selected chart, we construct matched Value Extraction, Comparison, and rule induction questions on the same visual input. This design holds chart content and visual complexity fixed. Answer formats, answer spaces, and scoring rules differ across tasks, so the accuracy gap does not isolate the effect of the reasoning operation.

Data sampling. We consider four common data regularities—*trend*, *periodicity*, *distribution*, and *association*—that are widely used in prior visualization and exploratory data analysis studies (Law et al., 2020; Demiralp et al., 2017). We first assemble candidate data from source families also used in the full benchmark construction, spanning DLMF-referenced analytic functions (National Institute of Standards and Technology, n.d.), TimeSynth-referenced synthetic series (Maat et al., 2017), public tabular datasets, and Tableau Public visualizations (Tableau, 2026). We then inspect the underlying values and metadata and apply predefined data-quality and regularity criteria, retaining only candidates with sufficient data points, limited missing data, unambiguous semantics, a verifiable target regularity, and metadata sufficient to establish chart–data alignment and task semantics. All eligibility decisions are made before model evaluation.

After filtering, we perform stratified sampling by regularity category and select 25 charts for each category, yielding 100 charts in total. Table 3 summarizes the source composition and sampling units of this diagnostic set.

Chart preparation. Not all charts are synthesized. When a source provides both a usable original chart and reliable underlying data or metadata, we retain the original chart image; otherwise, we render a chart from the retained numerical data using a standard representation appropriate for its structure. In both cases, access to the underlying values and metadata enables deterministic question construction and answer verification. For rendered charts, we preserve the original numerical relationships and derive titles, axis labels, legends, and category labels from the available metadata when applicable.

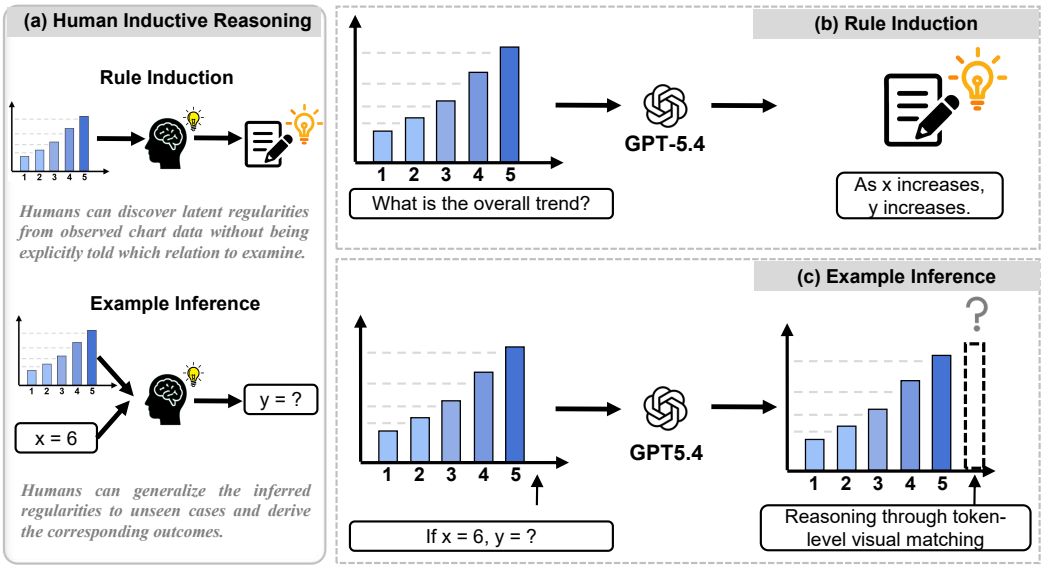


Figure 7: *(a) Human inductive reasoning* over charts involves first inferring a latent visual regularity from repeated data points and then applying the inferred rule to unseen or future cases. Panels *(b)* and *(c)* illustrate representative forms of rule induction and example inference in CHARTSEER, where the inferred regularity is used to determine the outcome for a queried target case.

The same chart is subsequently used for all question types associated with that instance. Therefore, differences among the three tasks cannot be attributed to changes in chart content, visualization style, or visual quality.

Matched question construction. For each chart, we construct three question types using the same visual evidence.

Value extraction asks the model to retrieve an explicitly observed value from the chart. The answer can be obtained by locating a specified visual element and reading its corresponding numerical value.

Comparison asks the model to compare two or more observed values, categories, or series. Unlike value extraction, this task requires a simple operation over observed values but does not require discovering a latent data-generating pattern.

Rule induction asks the model to infer the regularity exhibited by the observations as a whole, such as the dominant trend, recurring pattern, distributional structure, or relationship between multiple series. Answering these questions requires integrating evidence across multiple observations rather than directly retrieving individual values.

All questions and their ground-truth answers are generated from machine-readable numerical data or verified structured chart metadata. We manually inspect the generated questions to remove malformed or semantically ambiguous cases, without reference to model predictions. This yields matched question triplets in which value extraction, comparison, and rule induction are evaluated on exactly the same chart.

Evaluation protocol. We evaluate six representative MLLMs, namely Gemini 3.1 Pro (Google DeepMind, 2026b), GPT-5.4 (OpenAI, 2026), Qwen3-VL-8B-Instruct (Bai et al., 2025), LLaVA-OneVision-1.5-8B (Li et al., 2024), InternVL3-8B (Chen et al., 2024), and MiniCPM-V 4.5 (Yu et al., 2025), as reported in Figure 1(c), under the same prompting and inference settings. Each model receives only the chart and its corresponding question as input. Model predictions are evaluated using accuracy.

To obtain the aggregate results reported in Figure 1(c), we average model accuracy over the 100 charts and then report the mean performance across the six evaluated MLLMs for each question type. Value Extraction and Comparison achieve average accuracies of 61.7% and 74.6%, respectively,

Category	Data source	Domain	Charts	Chart form	Eligibility / sampling unit
Trend	Analytic functions	Controlled	7	Line	Clear monotonic or shape-based trend.
	Our World in Data (Our World in Data, 2026)	Health, economy, population, environment	9	Line, bar, scatter	$\geq$5 valid observations; clear dominant trend.
	ChartQA (Masry et al., 2022)	Social/economic statistics	9	Bar, line	Ordered numeric series with recoverable metadata.
Periodicity	Analytic periodic functions	Controlled	7	Line	Multiple visible cycles with known parameters.
	Open-Meteo	Weather/climate	10	Line, heatmap	Complete timestamps; verifiable temporal cycle.
	Tableau	Energy, transport, environment, society	8	Infographic, heatmap	Verifiable periodicity and temporal metadata.
Distribution	Analytic functions	Controlled	5	Histogram, density	Known modality, symmetry, skewness, or tails.
	Adult	Demographics, income	5	Histogram, density	Sufficient group support; limited missingness.
	Default of Credit Card Clients	Finance/credit risk	5	Histogram, box	Recoverable distributional difference.
	MAGIC Gamma Telescope	Physical science	5	Histogram, density	Adequate support; clear concentration pattern.
	ChartQA (Masry et al., 2022)	Public/economic statistics	5	Bar, stacked bar	Recoverable category frequencies or proportions.
Association	Controlled paired functions	Controlled	5	Scatter, paired line	Known positive, negative, nonlinear, or null association.
	UCR Time Series Archive (Dau et al., 2019)	Health, motion, sensors, industry	6	Paired line	Equal-length pair passing association and quality criteria.
	UEA Multivariate Time Series Archive (Bagnall et al., 2018)	Health, activity, sensors	6	Line, scatter	Aligned pair within a predefined association interval.
	UCI	Mixed domains	4	Dual line, scatter	Aligned trends with a verifiable relationship.
	Tableau	Finance, housing, environment, society	4	Infographic, scatter	Sufficient metadata to verify the relationship.
		Total	**100**	**25 charts per regularity category.**	

Table 3: Data sources and sampling protocol used in the motivating study.

whereas rule induction achieves only 33.8%. These are descriptive differences between question types evaluated on the same charts.

C Benchmark Construction Details

This section describes the benchmark construction process, including source and dataset selection, candidate preparation, regularity filtering, QA construction, benchmark composition, and quality control.

C.1 Candidate Data Preparation

C.1.1 Stage Definition and Data Sources

Candidate filtering pipeline. We apply four sequential selection stages: candidate preparation, statistical-criteria filtering, coverage-criteria filtering, and data quality control. Table 4 reports the number of candidate charts retained from each source family after each stage. Starting from an initial pool of 5,196 candidates, 4,082 remain after candidate preparation, 3,201 after statistical-criteria filtering, and 2,321 after coverage-criteria filtering. The final data quality control stage yields 1,438 charts included in ChartSeer.

The source pool covers complementary types of chart data. OWID contributes life-expectancy series from 16 countries across three time windows, while ChartQA provides tables from its training split. Open-Meteo contributes weather records across multiple cities, years, variables, and temporal resolutions. Tableau Public contributes multi-line, dual-axis-line, and heatmap sheets. UCI provides tabular datasets for distribution and association views, while UCR/UEA contributes univariate and multivariate time-series examples, including windowed Epilepsy series. NIST DLMF provides

Table 4: Number of candidate charts retained at each filtering stage of CHARTSEER. For reporting, the Adult, Default, and MAGIC datasets are grouped under the UCI source family.

Source family	Pool	Candidate Prep.	Statistical Criteria	Coverage Criteria	Data Quality Control
Our World in Data (Our World in Data, 2026)	708	576	443	310	177
ChartQA (Masry et al., 2022)	788	675	562	449	336
Open-Meteo (Zippenfenig, 2023)	1,248	1,014	780	546	312
Tableau Public (Tableau, 2026; Purich et al., 2023)	340	100	95	90	85
UCI (UC Irvine, 2026)	1,008	819	630	441	252
UCR/UEA (Dau et al., 2019; Bagnall et al., 2018)	956	777	598	419	239
NIST DLMF (National Institute of Standards and Technology, n.d.)	72	59	45	32	18
TimeSynth (Maat et al., 2017)	76	62	48	34	19
Total	**5,196**	**4,082**	**3,201**	**2,321**	**1,438**

parameterized mathematical functions over multiple domains, and TimeSynth contributes synthetic signal families and paired-series examples.

C.1.2 CANDIDATE VALIDATION, ALIGNMENT, AND DEDUPLICATION

Data readiness and alignment. Candidate preparation retains records that are suitable for subsequent analysis. Each record must be readable, contain finite numerical values, and, where applicable, provide a reliable alignment between the chart and its underlying data. Source-specific identifiers are used to ensure consistent matching, such as Grapher slugs for OWID, image–table pairs for ChartQA, spatiotemporal metadata for Open-Meteo, instance and feature identifiers for tabular and time-series datasets, and visualization metadata for Tableau Public.

Preparation criteria. Preparation rejects unsupported schemas, insufficient finite observations, non-unique alignment, and duplicate numerical views. Numerical views require at least 95% finite observations and at least 30 observations, or 20 for paired-series association views; categorical views require two groups and 30 observations. Periodicity preparation also requires nonzero variance after detrending.

Additional source details. We collect additional candidate examples from ChartQA, OWID, Tableau Public, Open-Meteo, UCI, and UCR/UEA. For ChartQA, we use the official training split and select candidate tables after removing nine overlaps with the existing inventory. For OWID, we use life-expectancy series from 16 countries over three time windows. For Tableau Public, we query multi-line, dual-axis-line, and heatmap visualizations, retain publicly accessible and downloadable sheets matching the queried chart type, remove duplicate workbook–sheet records. Open-Meteo provides six weather variables at hourly and daily resolutions from 48 city–year combinations; UCI provides individual-feature and feature-pair views for distribution and association; and UCR/UEA provides individual-series and multivariate dimension-pair views from 12 time-series datasets, supplemented with windowed examples from Epilepsy.

C.1.3 SOURCE-SPECIFIC PREPARATION DETAILS

Existing chart-table sources. For ChartQA (Masry et al., 2022), the paired table is parsed in row-major or column-major orientation and matched to the original chart image. OWID (Our World in Data, 2026) records are keyed by Grapher slug and preserve the original visualization and series metadata.

Tabular and time-series sources. Open-Meteo (Zippenfenig, 2023) records preserve timestamps, locations, variables, and the aggregates needed for later rendering and analysis. Adult (Becker & Kohavi, 1996), Default (Yeh, 2009), and MAGIC (Bock, 2004) retain the feature and class fields needed to construct class-conditional views. UCR/UEA (Dau et al., 2019; Bagnall et al., 2018) records retain archive names, instance identifiers, dimension indices, and aligned series values.

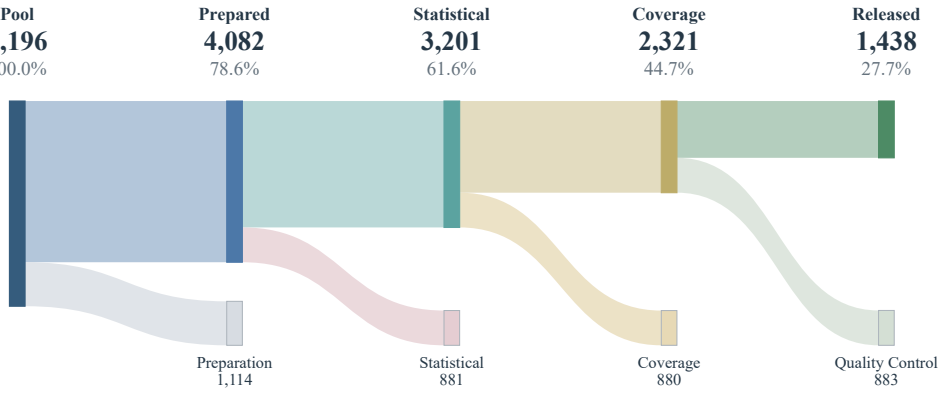


Figure 8: Candidate selection across filtering stages: 5,196 → 4,082 → 3,201 → 2,321 → 1,438.

Table 5: Source card for CHARTSEER benchmark.

Source	Snapshot	Original license	Release	Charts	Form	Underlying data
NIST DLMF	Release 1.2.7	DLMF terms (reference only)	Benchmark-generated assets only	18	rendered	generator parameters and tables
Our World in Data	snapshot recorded per Grapher slug	Upstream OWID/data-provider terms	Metadata + URL only	177	original	Grapher table
ChartQA	source split and file recorded	ChartQA and source-chart terms	Metadata + URL only	336	original	paired source table
Open-Meteo	retrieval date recorded in metadata	Upstream weather-provider terms	Metadata + URL only	312	rendered	hourly weather table
Tableau Public	per-view access date	Workbook and source terms	Metadata + URL only	85	original	paired source table
Adult	UCI snapshot	CC BY 4.0	With attribution	52	rendered	UCI table
Default of Credit Card Clients	UCI snapshot	CC BY 4.0	With attribution	138	rendered	UCI table
MAGIC Gamma Telescope	UCI snapshot	CC BY 4.0	With attribution	62	rendered	UCI table
UCR/UEA archives	archive and dataset name recorded	Dataset-specific upstream terms	Metadata + URL only	239	rendered	multivariate/time-series records
TimeSynth	2017 reference	MIT (reference software)	Benchmark-generated assets only	19	rendered	generator relation, parameters, seeds, and arrays
Total				**1,438**		

Benchmark-generated controlled sources. The DLMF-referenced component (National Institute of Standards and Technology, n.d.) is generated by our benchmark builder from standard analytic families documented in DLMF, including constant, linear, polynomial, absolute-value, exponential, logarithmic, sigmoid/tanh/ReLU, Gaussian-like, and mixed polynomial–sinusoidal curves. Each generated record stores its function name, parameters, sampled domain, and derived numerical attributes. The TimeSynth-referenced component (Maat et al., 2017) is generated from fixed primitive-shape and signal mappings with recorded parameters and random seeds.

C.1.4 LICENSE AND REDISTRIBUTION

Table 5 summarizes the provenance, licensing, and redistribution status of all chart sources in CHARTSEER. For each source, we record the corresponding snapshot or access information, upstream license or usage terms, release form, number of retained charts, visualization form, and available underlying data. For sources with redistribution constraints, such as OWID, ChartQA, Tableau Public, and UCR/UEA, the benchmark release provides metadata and source URLs rather than

Table 6: Statistical and coverage criteria used in CHARTSEER.

Regularity	Statistical criteria	Coverage criteria
Trend	Monotonic: $\lvert\tau\rvert \geq 0.5$, $p < 0.05$. Curved: quadratic $R^2 \geq 0.60$ and gain ≥ 0.15 over linear. Piecewise: two-segment R^2 gain ≥ 0.20, with opposite standardized slopes or contrast ≥ 0.5.	At least 30 points. Monotonic: 70% direction agreement overall and 60% in each of three consecutive blocks. For an in-range quadratic turning point, at least 20% of points lie on each side. Piecewise segments each contain at least 20% of observations.
Periodicity	Cycles: two of ACF ≥ 0.5, seasonal strength ≥ 0.6, or adjacent-cycle correlation ≥ 0.6. Peaks/troughs: interval CV ≤ 0.20.	At least two complete cycles; at least 60% of cycles correlate ≥ 0.6 with the estimated seasonal phase. Extrema require three occurrences and mean spacing within 20% of the specified period.
Distribution	Absolute skewness ≥ 1; excess kurtosis ≥ 2; dip-test $p < 0.05$ with two modes; positive-data CV ≤ 0.1 or ≥ 0.5; or a top-two category gap ≥ 0.1.	Numeric shape/moment views require 50 observations. Categorical views require 30 observations in total and five in each represented group. Each retained mode needs 5% mass in its neighboring three-bin region. Primary normalized-effect margin is at least 10%.
Association	Pearson or Spearman magnitude ≥ 0.6, $p < 0.05$; or dCor ≥ 0.4, permutation $p < 0.05$. No-detectable association: both correlation magnitudes ≤ 0.2 and dCor $p \geq 0.1$. Segment-specific effects use magnitude ≥ 0.6 with a sign change or strength contrast ≥ 0.5.	At least 40 aligned observations and ten distinct values per dimension. A global linear/monotonic sign must be supported in both halves with correlation magnitude ≥ 0.3. Segment-specific targets also retain the 20% segment minimum. Other targets retain the sample and distinct-value requirements.

redistributing the original assets. Benchmark-generated visualizations and data are released directly where permitted, with attribution or source-specific terms preserved as applicable.

C.1.5 CHART RENDERING

Table-only sources are rendered with Matplotlib using chart types appropriate to their data schema, while original images from OWID, ChartQA, and Tableau Public are retained when they can be uniquely matched to the corresponding data. Rendering and sampling are deterministic to ensure reproducibility. Chart-type balancing is applied only after regularity filtering.

Figures 9–11 illustrate the chart templates used for rendering. Figure 9 shows variants of bar, line, value-based map, and heatmap charts; Figure 10 covers scatter, histogram, density, box, and violin charts; and Figure 11 presents grouped-line, dual-axis-line, and multi-series Cartesian variants. These templates vary common visual encodings and layouts within each chart family.

C.1.6 STATISTICAL AND STRUCTURAL CRITERIA

Table 6 gives the operational definitions used to filter the candidate pool. For each regularity, the statistical criteria identify the target pattern, while the coverage criteria ensure that it is supported by sufficient data points and is not driven by a small or localized subset of the chart.

C.1.7 THRESHOLD SETTINGS AND GATE EXECUTION

The thresholds in Table 6 are operational criteria for candidate filtering. Statistical and coverage criteria are evaluated separately, with coverage requiring sufficient within-chart support for the detected regularity.

C.1.8 REGULARITY-SPECIFIC FILTERING DETAILS

Trend. Trend filtering is applied to DLMF-referenced functional candidates (National Institute of Standards and Technology, n.d.), OWID series (Our World in Data, 2026), and eligible ChartQA records (Masry et al., 2022). Generated functions retain their stored direction and monotonicity attributes for verification. ChartQA and OWID candidates are evaluated on their aligned numerical series for monotonic, piecewise, and curved trends.

Periodicity. Periodicity filtering is applied to cycle-aware weather views and curated visualizations (Zippenfenig, 2023; Tableau, 2026). Candidate definitions specify the location or source chart, variable, time window, aggregation, and tested period. Statistical support and visible cycle coverage are checked for each candidate before primary assignment.



Figure 9: Rendering templates for bar, line, value-based map, and heatmap charts.

Distribution. Distribution filtering uses class-conditional views of the UCI datasets, Adult, Default, and MAGIC (Becker & Kohavi, 1996; Yeh, 2009; Bock, 2004). Applicable subtype criteria in Table 6 are tested against the underlying observations, with sufficient group support and a determinate primary target required for retention.

Association. Association filtering is applied to TimeSynth-referenced synthetic multi-series mappings (Maat et al., 2017), UCR/UEA time-series pairs (Dau et al., 2019; Bagnall et al., 2018), eligible Tableau Public visualizations (Tableau, 2026), and ChartQA monotone-break candidates (Masry et al., 2022). For UCR/UEA, candidates require sufficient aligned observations for the requested global

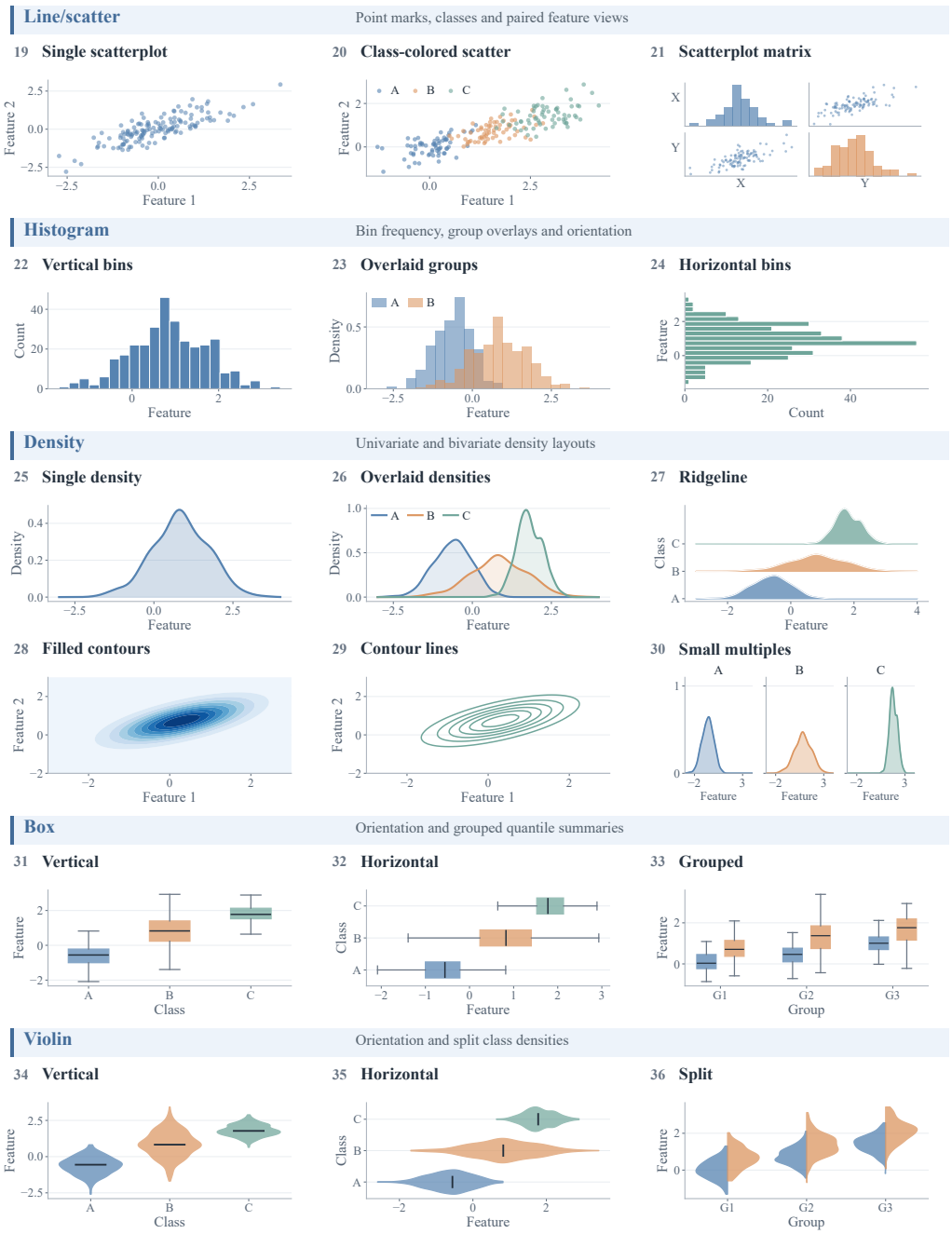


Figure 10: Rendering templates for scatter, histogram, density, box, and violin charts.

or segment-wise calculation; strong positive and negative pairs use $r \geq 0.6$ and $r \leq -0.6$, while pairs assigned the weak-correlation label use $|r| \leq 0.2$. The ChartQA predicate requires an otherwise directionally consistent relation with one uniquely identifiable violating point, which is treated as a local relation violation. The final Tableau Public subset contains 39 association visualizations.

C.2 QA Construction

For the original benchmark, the input to QA construction was the set of qualified chart–table–regularity triples. For each triple, we first define one verified target regularity and instantiate every

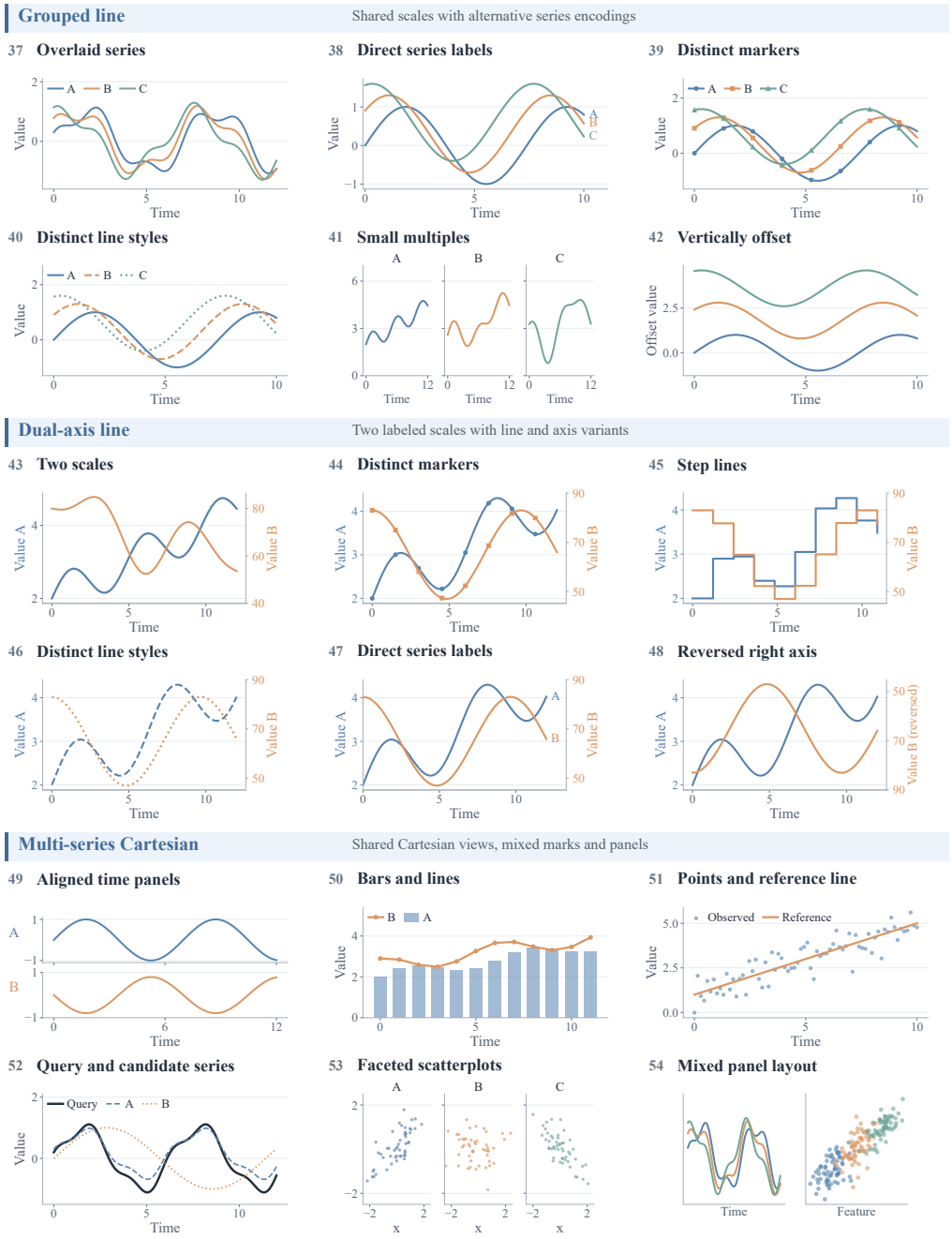


Figure 11: Rendering templates for grouped-line, dual-axis-line, and multi-series Cartesian charts.

applicable question template from that same target. Rule induction asks the model to identify or verify a chart-supported rule, whereas example inference applies the inferred regularity to a queried target case or candidate outcome.

C.2.1 Question and Evaluation Mapping

Each reported regularity–task cell is formed from questions carrying the corresponding regularity label and task type. All non-numerical records are multiple-choice and use exact normalized option-label accuracy, while numerical records use relaxed accuracy with a 5% relative-error tolerance.

Invalid parses receive zero accuracy credit. Cell accuracies are computed over questions rather than charts, so a chart with multiple QAs contributes once per applicable QA. The complete question templates and their regularity assignments are provided in Table 7.

C.2.2 Question Templates and Instantiation

Table 7 lists the released template families. The data fact names the targeted property; the question type specifies the answer operation. Braces denote chart-specific variables. RI prompts can name a relation or present candidate rules, so the task is to verify or select a chart-supported formulation. EI templates operationalize rule application by asking the model to determine an expected value, category, state, segment, or revised outcome for a queried target. Continuation and classification instantiate new-case inference directly, while transformation and structural templates test whether the same inferred regularity is applied consistently to the specified target.

For real-world continuation questions, the reference answer is the consequence of carrying the verified empirical regularity into the queried case. It is conditional on that persistence assumption, rather than an observed future outcome or a causal effect. For generated functions, the reference follows the stored generator. Deterministic answer generation establishes consistency with these benchmark rules.

Table 7: Question templates for all question types.

Task format	Data fact	Question type	Question template	Instructions
Rule Induction	Trend	Rule Verification	Which candidate option correctly states whether {property} is true for {series} over {range}?	• Inspect the complete specified range rather than a single point. • Treat absolute terms such as "always" and "never" literally. • Return only the requested option label.
Rule Induction	Trend	Rule Verification	Which candidate option correctly describes whether {series} exhibits {global pattern} over the displayed range?	• Judge the dominant end-to-end pattern rather than local fluctuations. • Possible patterns may include increasing, decreasing, stable, fluctuating, U-shaped, or inverted-U. • Return only the requested option label.
Rule Induction	Distribution	Rule Verification	Which candidate option correctly states whether {group} is more concentrated or dominant in {region}?	• Compare all groups over the same visual region and scale. • Distinguish a dense concentration from isolated extreme points. • Return only the requested option label.
Rule Induction	Recurrence / Association	Rule Verification	Which candidate option correctly states whether {series/association} follows {dependency or recurrence rule}?	• Check repeated intervals, phase alignment, or co-movement across the displayed data. • Do not verify the rule from only one timestamp or cycle. • Return only the requested option label.

Continued on next page

Table 7 – continued from previous page

Task format	Data fact	Question type	Question template	Instructions
Rule Induction	Trend	Rule Identification	Which candidate rule best describes the overall direction or shape of {series}?	• Compare every candidate with the complete plotted range. • Prefer the simplest rule that explains the dominant pattern without contradicting major regions. • Return only the requested option label.
Rule Induction	Recurrence	Rule Identification	Which candidate rule best describes the period, peak, trough, or phase of {series}?	• Estimate the repeated interval before comparing the candidate rules. • Distinguish period, phase, amplitude change, and one-time noise. • Select the option supported by the majority of visible cycles.
Rule Induction	Distribution	Rule Identification	Which candidate rule best describes {variable/group pattern} over {range/region}?	• Infer the dominant distributional property, such as which side or group dominates, where values concentrate, or how groups differ over the displayed support. • Compare all candidate rules using relative density or frequency over the complete displayed support. • Return only the requested option label.
Rule Induction	Association	Rule Identification	Which candidate rule best describes the association between {variable A} and {variable B} under {condition}?	• Identify the association direction and any visible regime boundary. • Reject options that confuse association direction, conditional behavior, or coincidence. • Return only the best-supported option label.
Rule Induction	Trend	Rule Identification	Which candidate rule best describes {target trend relation}?	• The target relation may describe how one quantity changes with another (e.g., as {x} increases) or how one series changes relative to another over {range}. • Infer the dominant direction or ordered relation across the complete displayed range, including a supported crossover such as "initially below and later above." • Return only the requested option label.

Continued on next page

Table 7 – continued from previous page

Task format	Data fact	Question type	Question template	Instructions
Rule Induction	Trend	Rule Identification	Which candidate qualitative shape rule best describes {series} over {range}?	• Inspect turning points and compare the starting and ending levels. • Distinguish stable-with-noise, fluctuation, oscillation, U-shape, and inverted-U. • Return only the requested option label.
Rule Induction	Recurrence / Association	Rule Identification	Which candidate rule best describes {recurring/relational target}?	• For recurrence, identify the repeated unit and the relative behavior of relevant phases (e.g., weekdays versus weekends or early versus late months within each cycle). • For association, identify the dependency direction or co-movement between the variables (e.g., positive or negative association). • Return only the requested option label.
Example Inference	Value	Numerical Prediction	Following the observed rule, what is the expected value of {y} at {new x position}?	• Infer the trend or function before extrapolating beyond the observed range. • Preserve the requested order and chart units. • Return only the requested value or ordered value list, without unsupported precision.
Example Inference	Outlier	Numerical Prediction	For the queried point, which {coordinate/value} is expected under the inferred pattern, or which candidate violates that expectation?	• Infer the local baseline from neighboring points or repeated structure. • Apply the inferred baseline consistently to the queried point or candidate value. • Return exactly the requested coordinate or numerical value.
Example Inference	Derived Value	Numerical Prediction	After applying {shift/transformation}, what {time, phase, index, or quantity} follows?	• Convert the specified transformation into the chart's time or index units. • Apply the transformation exactly once. • Return only the requested time, phase, index, or quantity, including units when required.
Example Inference	Categori-zation	Categorical Prediction	Given {feature} = {value}, which candidate category is the new instance most likely to belong to?	• Locate the supplied value in the relevant class-conditioned distribution. • Select the category with the strongest visual support at that value. • Return only the requested option label.

Continued on next page

Table 7 – continued from previous page

Task format	Data fact	Question type	Question template	Instructions
Example Inference	Condition / Comparison	Categorical Prediction	Under the induced rule, which candidate option correctly states whether {new case} satisfies {property or comparison}?	• Apply the same retained regularity to the specified new time, cycle, or case. This includes relative comparisons such as whether one series is expected to be above another, or whether one phase is expected to exceed another in a new cycle. • Respect the stated horizon and strict versus inclusive comparison wording. • Return only the requested option label.
Example Inference	State / Relation	Categorical Prediction	For {new case}, which candidate {category/state} is expected under the induced rule? Or, if {variable A} {change}, which candidate correctly describes how {variable B} is expected to change?	• Apply the same retained distributional or relational rule to the new case. Valid states may include labels such as "common/extreme" or directional outcomes such as "increase/decrease/stable." • For association questions, preserve the induced dependency direction when mapping a change in one variable to the other. • Return only the requested option label.
Example Inference	Structural Location	Multi-Target Prediction	Under the inferred structural rule, which candidate set of target segments or locations satisfies {target structural pattern}?	• Inspect every labeled segment using the same structural criterion. • Include all matches and exclude near-matches. • Return only the requested option label.
Example Inference	Structural Location	Multi-Target Prediction	Under the inferred structural rule, which candidate target segment or location is consistent with {target structural feature}?	• Compare every candidate segment under a consistent definition of the target feature. • Select the strongest valid match. • Return only the requested option label.
Example Inference	Structural Location	Multi-Target Prediction	For the supplied statement about {target structural case}, which candidate revision is consistent with the inferred structural rule?	• Identify the smallest inconsistency between the supplied statement and the inferred structural rule. • Apply the same rule to compare the candidate revisions. • Return only the requested option label.

Continued on next page

Table 7 – continued from previous page

Task format	Data fact	Question type	Question template	Instructions
Example Inference	Ordering / Phase	Multi-Target Prediction	Following {`ordering, recurrence, or phase rule`}, which candidate {`segment, position, or phase`} should appear next?	• Infer the visual ordering or phase relation before applying the requested continuation or shift. • Preserve every dimension that is not explicitly changed. • Return only the requested option label.

C.2.3 Regularity-specific QA Instantiation

For Trend, rule induction questions use stored generator attributes or regularities recovered from ChartQA/OWID series, while example inference uses a generator-produced continuation target or an equivalent new-case query. For Periodicity, rule induction asks for a peak or cycle rule computed from the retained aggregate, whereas example inference applies the stored phase shift, such as four hours later, three months later, or two hours earlier. For Distribution, rule induction asks which class exhibits the target concentration or distributional rule, and example inference maps a new feature value or feature tuple to the class supported by the stored profile. For Association, rule induction covers cross-series relationships and the direction tasks listed in the release mapping; example inference applies the stored mapping, segment condition, or candidate-selection rule to the queried target, including local relation-violation cases.

C.2.4 Counterfactual Construction Details

Scope and selection. The final subset contains 252 of the 695 eligible chart configurations from these sources. Selection used masked-label recoverability. For each selected chart, we intervene on the target label in the underlying table before rendering, while keeping the feature values, row order, sample count, and visualization specification fixed. Because all three sources use binary targets, the intervention is a deterministic global label complement: `>50K` and `<=50K` are exchanged for Adult, `default` and `no_default` for Default, and `gamma` and `hadron` for MAGIC. We then recompute the class-conditioned statistics and re-render the chart from the transformed table. Figure 12 shows two representative label interventions.

D Quality Control

All 3,929 released QAs are programmatically verified, all 1,438 charts are manually inspected, and a stratified subset of 400 QAs is independently audited by three expert annotators.

Programmatic QA verification. At QA-generation time, every answer is recomputed by executing the analytic rule associated with its question template on the same table, generating function, class profile, correlation record, or peak statistic used to render the chart. An instance is retained only when all template variables are bound, its image exists, the target regularity satisfies the template-specific constraints, and the answer is finite, representable in the declared answer space, and unique. For every non-numerical question, exactly one option must match the recomputed answer. Distractors are generated programmatically from the ground-truth regularity and underlying data using rule-based transformations that instantiate incorrect regularities, including overly specific and overly general interpretations. Records with zero or multiple valid options are rejected as invalid or ambiguous.

Chart-level inspection. We manually inspect all 1,438 retained charts before release. For each chart, the reviewer checks (i) rendering integrity, including missing or corrupted marks, labels, legends, and axes; (ii) readability at the resolution distributed to models; and (iii) consistency between the visible pattern and the intended regularity. Charts that are unreadable, incorrectly rendered, or do not provide sufficient visual evidence for the target regularity are corrected when the source data permit an unambiguous repair and otherwise removed together with their associated QA pairs.

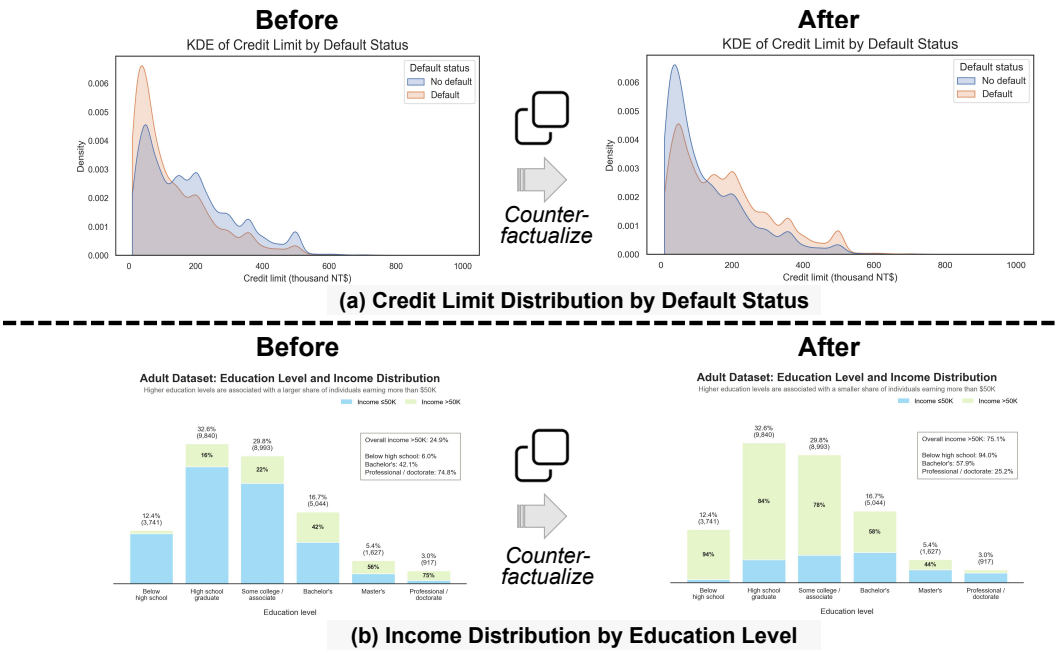


Figure 12: Representative counterfactual construction. The left column shows the original charts, and the right column shows their counterfactual versions after deterministic label interventions. *(a)* Default status is flipped. *(b)* The binary income label is flipped. Other variables and the visualization design remain unchanged.

Stratified semantic audit. We further audit 400 QA pairs using a balanced 4×2 design: 50 examples are sampled from each combination of the four regularity categories (trend, periodicity, distribution, and association) and the two task families (rule induction and example inference). Three annotators with expertise in statistics and data visualization independently answer every sampled question using only the chart, question, and answer format shown to evaluated models. The benchmark answer is hidden during annotation. In addition to recording an answer, each annotator flags questions for which the chart supports another plausible interpretation, the wording underspecifies the intended operation, or no unique answer can be determined.

Adjudication and corrective action. All answer disagreements and ambiguity flags are reviewed jointly. Adjudication checks the displayed chart against its underlying data and the executable rule used to create the answer. A local error is corrected or removed. When a case reveals a systematic problem, we discard all instances generated from the affected template, distractor rule, or construction constraint rather than revising and regenerating them. The final counts therefore describe the retained inventory after adjudication. Figures 50–52 show representative annotation-interface examples.

Merge-time consistency checks. The merged benchmark is checked to ensure that every QA references a retained chart and that task labels belong to rule verification, rule identification, numerical prediction, categorical prediction, or multi-target prediction. As a final merge-time check, every option key is resolved against its option text, exactly one key matches the stored answer, every QA points to a retained chart, and all QAs attached to a chart share its split. For reproducibility, the benchmark metadata records the old-to-new option-key map for every permuted question. Applying the same map to a stored letter prediction preserves its recorded correctness.

E Detailed Benchmark Statistics and Data Split

Figure 13 summarizes four marginal distributions of the released benchmark. Capability families and question types are counted over all 3,929 QAs; chart regularities and source families are counted over 1,438 unique charts. Each chart contributes once to each chart-level distribution, irrespective of its

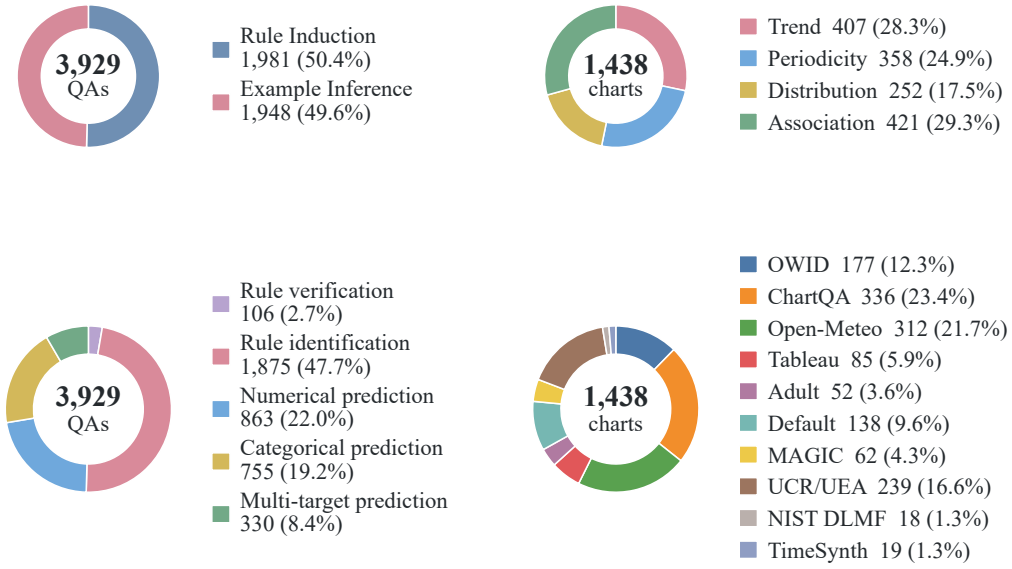


Figure 13: Composition of the released benchmark.

number of questions. Question-type percentages therefore use the full QA set, rather than the number of QAs within a capability family.

E.1 Chart and QA Composition

The final source composition is reported in Table 4, while Figure 14 summarizes the visualization-family composition. For NIST DLMF and TimeSynth, the source name denotes the external reference used to define the function or signal family; the benchmark values themselves are generated by the released programs with recorded parameters and random seeds.

Figure 14 reports the frequency and associated regularity of each retained visualization family. The five question-type totals in this section match Figure 4: 106 Rule Verification, 1,875 Rule Identification, 863 Numerical Prediction, 755 Categorical Prediction, and 330 Multi-Target Prediction questions. The chart-type cross-tab is given in Figure 14, the per-source counts in Table 4, and the answer-space checks are described above. The inventory in Table 7 contains 21 authored template families: 4 Rule Verification, 7 Rule Identification, 3 Numerical Prediction, 3 Categorical Prediction, and 4 Multi-Target Prediction families. Across the 3,929 instantiated questions, whitespace tokenization gives a mean length of 16.2 words and median of 11 (range 6–102). Across all answer options, the mean is 3.8 words and the median is 2 (range 1–28).

Figure 15 visualizes the regularity–question-type cross-tab while preserving exact small-cell counts in the text below. For completeness, the exact (RV, RI, NP, CP, MTP) count vectors are Trend $(23, 1{,}088, 345, 68, 31)$, Periodicity $(16, 449, 120, 134, 18)$, Distribution $(6, 155, 138, 345, 31)$, and Association $(61, 183, 260, 208, 250)$. Their row totals are 1,555, 737, 675, and 962; the corresponding column totals are 106, 1,875, 863, 755, and 330, for 3,929 QAs.

RI and EI have different category and output-format mixtures. Trend contributes 1,111 of 1,981 RI items (56.1%) and 444 of 1,948 EI items (22.8%); EI also contains numerical and multi-target outputs. Aggregate RI–EI differences combine these composition and scoring differences with task demands. The higher accuracy on Trend than Periodicity may reflect differences in the visual evidence required by their component tasks.

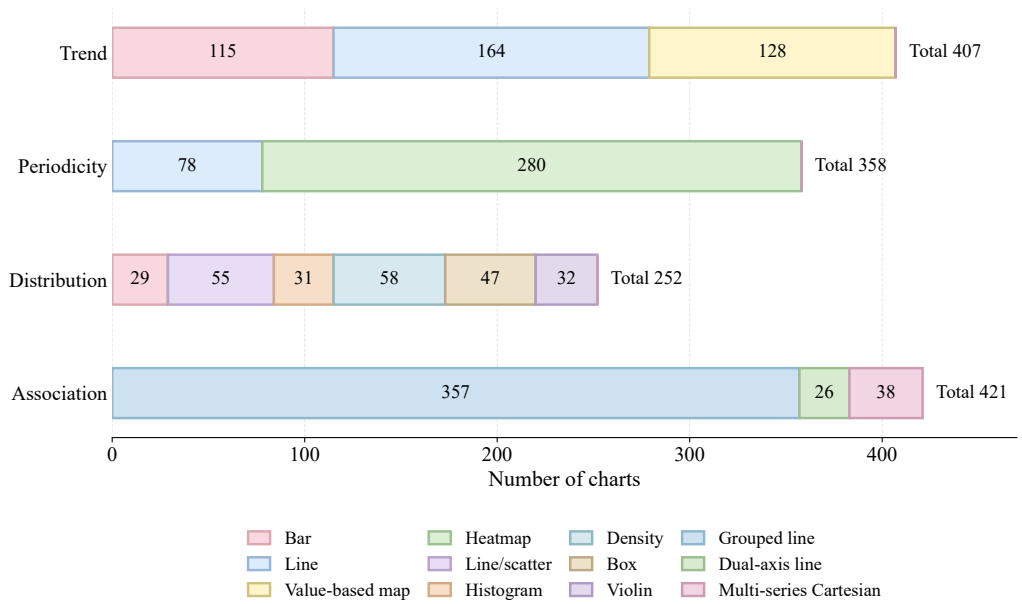


Figure 14: Chart-family composition of the 1,438 retained charts in CHARTSEER. Every nonzero cell is labeled with its exact chart count; rare variants are merged into their parent visualization families.

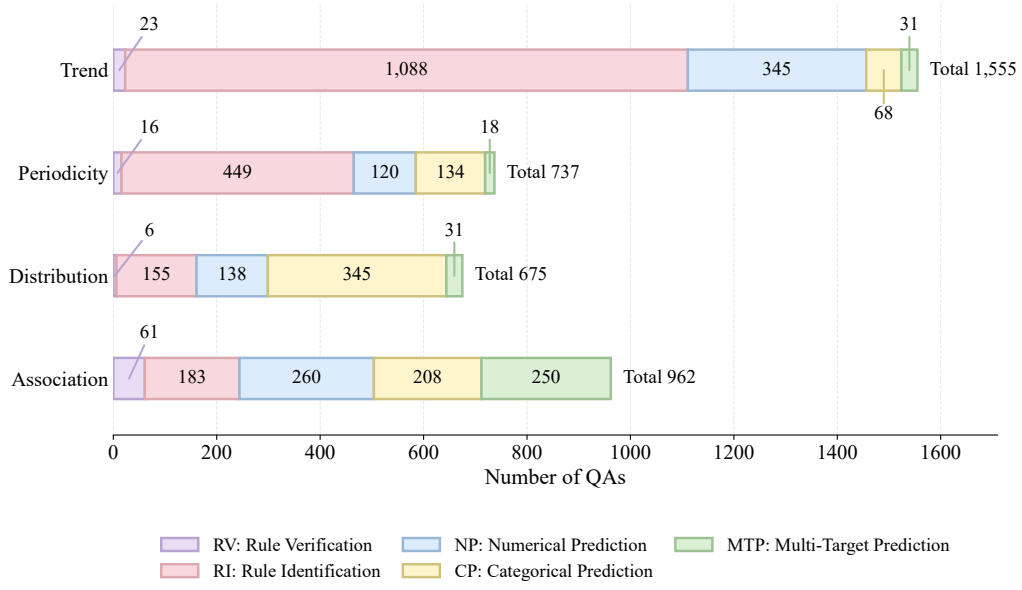


Figure 15: Question-type composition by regularity across 3,929 QAs.

E.2 COMPARISON WITH EXISTING CHART BENCHMARKS

Figure 16 summarizes available chart-level counts for CHARTSEER and five existing benchmarks: DVQA (Kafle et al., 2018), ChartQA (Masry et al., 2022), CharXiv (Wang et al., 2024), InfoChartQA (Xie et al., 2025), and ChartMuseum (Tang et al., 2025). Counts for DVQA and InfoChartQA come from released annotations; ChartQA is estimated from non-header rows in a deterministic 1-in-25 sample of released tables (836 charts); and CHARTSEER uses associated data for 1,353 charts. For CharXiv and ChartMuseum, where underlying tabular data are not available for the sampled charts, we instead count visible marks or cells in deterministic random samples of 100 charts. Mean and median summarize these source-specific counts. CHARTSEER has a mean of 9,189

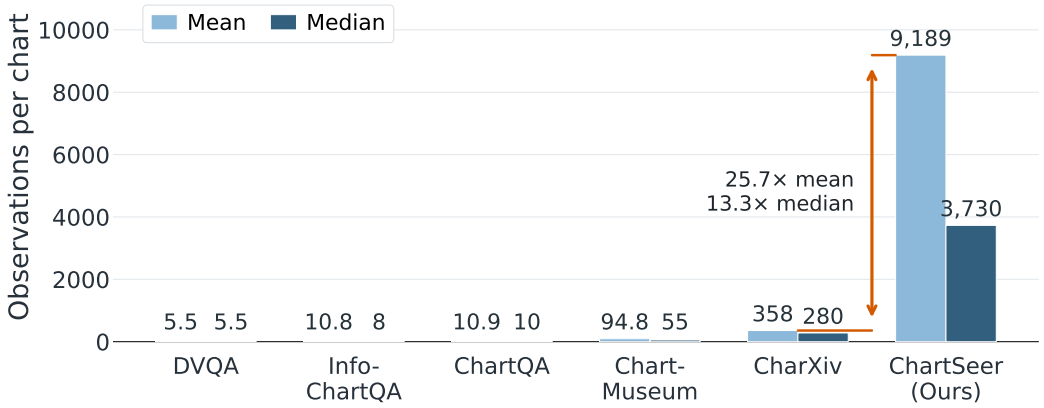


Figure 16: Chart-level data-point counts. Light- and dark-blue bars show means and medians, respectively.

Table 8: Development/test split.

Split	Charts	QAs	RV	RI	NP	CP / MTP
Development	288	780	16	367	168	152 / 77
Test	1,150	3,149	90	1,508	695	603 / 253
Total	1,438	3,929	106	1,875	863	755 / 330

and a median of 3,730 underlying observations per chart. Relative to the CharXiv pilot counts (mean 358; median 280), the numerical ratios are $25.7\times$ and $13.3\times$, respectively.

E.3 Data Split Protocol

We create an 80/20 test/development partition at chart level. Charts are stratified jointly by regularity and source family; a largest-remainder allocation preserves the 288-chart development target across strata. All questions attached to a chart inherit the same split. Chart identifiers are SHA-1 hashes of the source component and normalized image path, and the intersection of development and test chart hashes is empty.

The chart-level regularity counts are 81/72/50/85 for Trend, Periodicity, Distribution, and Association in development, and 326/286/202/336 in test. No training split is distributed because ChartSeer is an evaluation benchmark. Table 2 reports the test-only score.

F Bottleneck Analysis Details

We compare four input conditions on 1,090 example inference instances with source-derived structured tables for six open-source models, with the subset size constrained by API limits: Qwen3-VL-8B-Instruct, LLaVA-OneVision-1.5-8B, MiniCPM-V 4.5, InternVL3-8B, TinyChart, and Chart-Gemma. Figures 17 reports the results of this analysis.

F.1 Intervention Design

We compare the original image with a model-extracted table, a source-derived table, and a source-derived table plus rule guidance. The question, answer format, and scoring criterion remain fixed within each instance.

Original Image. The model receives the original benchmark chart together with the example inference question and directly predicts the answer. This setting therefore reflects the complete

example inference pipeline, in which the model must read the chart, infer the relevant regularity, and apply it to the queried case.

Model-Extracted Table. The model first converts the chart into a question-relevant Markdown table. The extracted table, rather than the original chart, is then provided together with the same example inference question for answer prediction. Because the table is produced by the model itself, this setting retains any information loss or numerical errors introduced during chart reading.

Gold Table. The original chart is replaced by a structured table derived from the source data, including numerical series or symbolic segment mappings. The model answers the same question using this representation.

Gold Table + Rule. The model is provided with both the gold underlying table and the gold rule associated with the chart. The rule is represented as a source-derived fitting or peak-selection procedure or as an instantiated rule statement.

For table-based settings, the original chart is withheld during answer prediction. When a model implementation requires an image input, we provide a neutral 16×16 compatibility image that contains no task-relevant information. Multiple-choice questions are evaluated using exact-match accuracy. Numerical-prediction questions use the same relaxed accuracy criterion as the main benchmark, with a 5% relative-error tolerance.

F.2 More Experimental Results

Figure 17 reports the input-condition analysis for six open-source models on example inference tasks. The comparisons describe the effect of changing the available representation and rule information.

How does model performance change with gold-table input? Accuracy changes substantially when the chart image is replaced by its source-derived gold table. Qwen3-VL-8B-Instruct improves from 61.5% to 75.2%, LLaVA-OneVision-1.5-8B from 60.5% to 67.0%, and ChartGemma from 37.0% to 48.3%. The effect is particularly large for association: averaged over Qwen3-VL-8B-Instruct, LLaVA-OneVision-1.5-8B, MiniCPM-V 4.5, and InternVL3-8B, accuracy rises from 14.5% to 68.4%. However, the intervention is not uniformly beneficial. For example, the average trend accuracy of these four models drops from 68.0% to 61.9%. We hypothesize that this drop arises because converting charts into tables removes global visual cues, such as the overall trajectory of a series, that can make trends easier to recognize directly from the chart.

☞ **Finding 1:** *Providing gold tables substantially improves overall MLLM accuracy, suggesting that visual perception is a major bottleneck.*

How does model performance change with reference-rule input? Accuracy further improves when reference rule information is provided together with the gold table. For example, LLaVA-OneVision-1.5-8B improves from 67.0% to 91.3%. The effect is particularly large for periodicity: averaged over Qwen3-VL-8B-Instruct, LLaVA-OneVision-1.5-8B, MiniCPM-V 4.5, and InternVL3-8B, accuracy rises from 13.6% to 40.8%. Distribution accuracy also increases substantially, from 78.5% to 97.1%. We hypothesize that these gains arise because providing the reference rule removes the need to infer the latent regularity from the table, allowing models to focus on applying the rule to the target case.

☞ **Finding 2:** *Providing reference rules further improves table-based MLLM accuracy, suggesting that inferring latent regularities remains an additional bottleneck after visual perception is removed.*

What errors remain with reference-rule input? Errors remain even when both the gold table and reference rule are provided. For example, MiniCPM-V 4.5 achieves only 55.7% on trend and 63.2% on association, while InternVL3-8B reaches 68.5% and 68.4%, respectively. In contrast, performance on distribution is much higher: Qwen3-VL-8B-Instruct, LLaVA-OneVision-1.5-8B, and MiniCPM-V 4.5 reach 100%, while InternVL3-8B reaches 88.5%. These remaining errors suggest

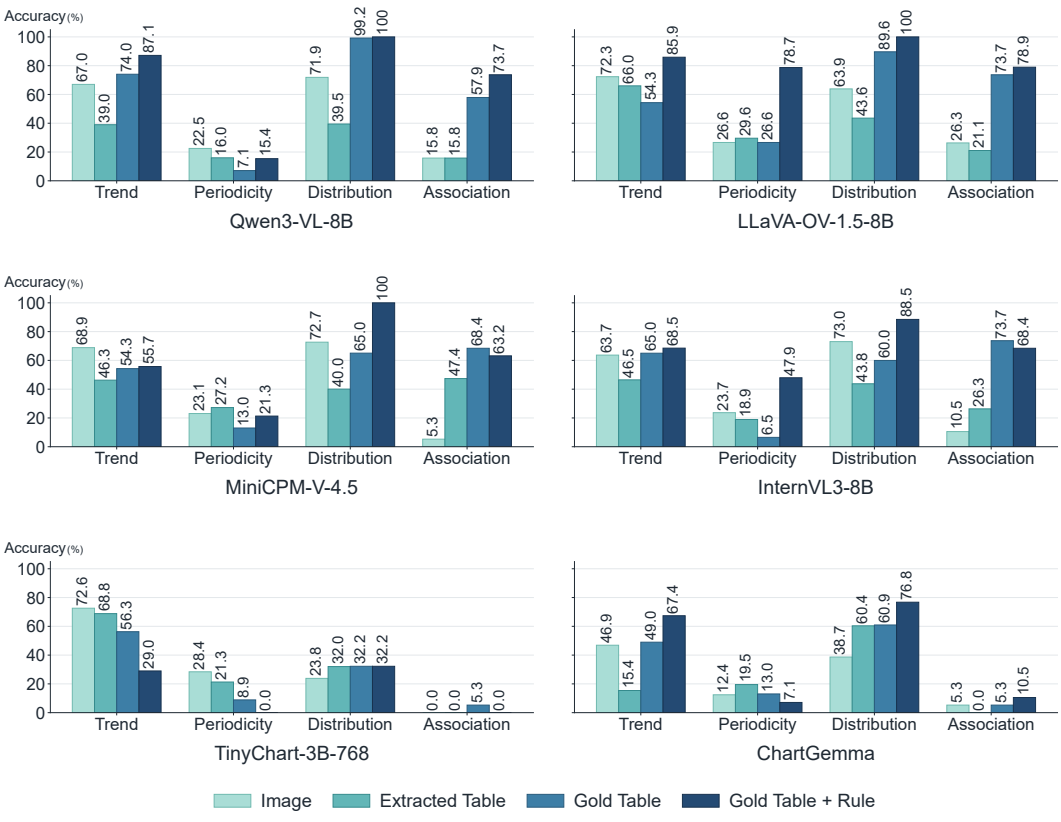


Figure 17: Bottleneck analysis for six open-source MLLMs.

that, even after visual perception and rule induction are largely bypassed, some models still struggle to apply the supplied regularity to the target case.

☞ **Finding 3:** *Errors persist even with gold tables and reference rules, indicating remaining difficulties in applying the supplied regularity to new cases.*

How does model performance change with extracted-table input? Performance is generally lower with model-extracted tables than with gold tables, indicating that errors can already arise during chart-to-table conversion. This issue is particularly severe for specialized chart models: TinyChart and ChartGemma produce outputs satisfying the required table format on only 1.7% and 0.9% of instances, respectively, compared with nearly 99% for LLaVA-OneVision-1.5-8B, MiniCPM-V 4.5, and InternVL3-8B and 69.4% for Qwen3-VL-8B-Instruct. These results show that the extracted-table setting evaluates the full extraction-and-reasoning pipeline, and its performance can therefore be substantially affected by failures in constructing the intermediate table representation.

☞ **Finding 4:** *Model-extracted tables introduce an additional extraction bottleneck, with some models frequently failing to construct a valid intermediate table representation from the chart.*

Do proprietary models exhibit the same pattern? Proprietary models show a similar pattern across the four input conditions. For example, Gemini 3.1 Pro improves from 37.5% to 95.0% on periodicity when moving from Gold Table to Gold Table + Rule. With both the gold table and reference rule provided, GPT-5.4 reaches 92.5% on trend, 72.5% on periodicity, and 100% on both distribution and association. However, performance is not uniformly monotonic across all categories, as replacing the chart with a textual table changes the input representation in addition to removing visual extraction requirements.

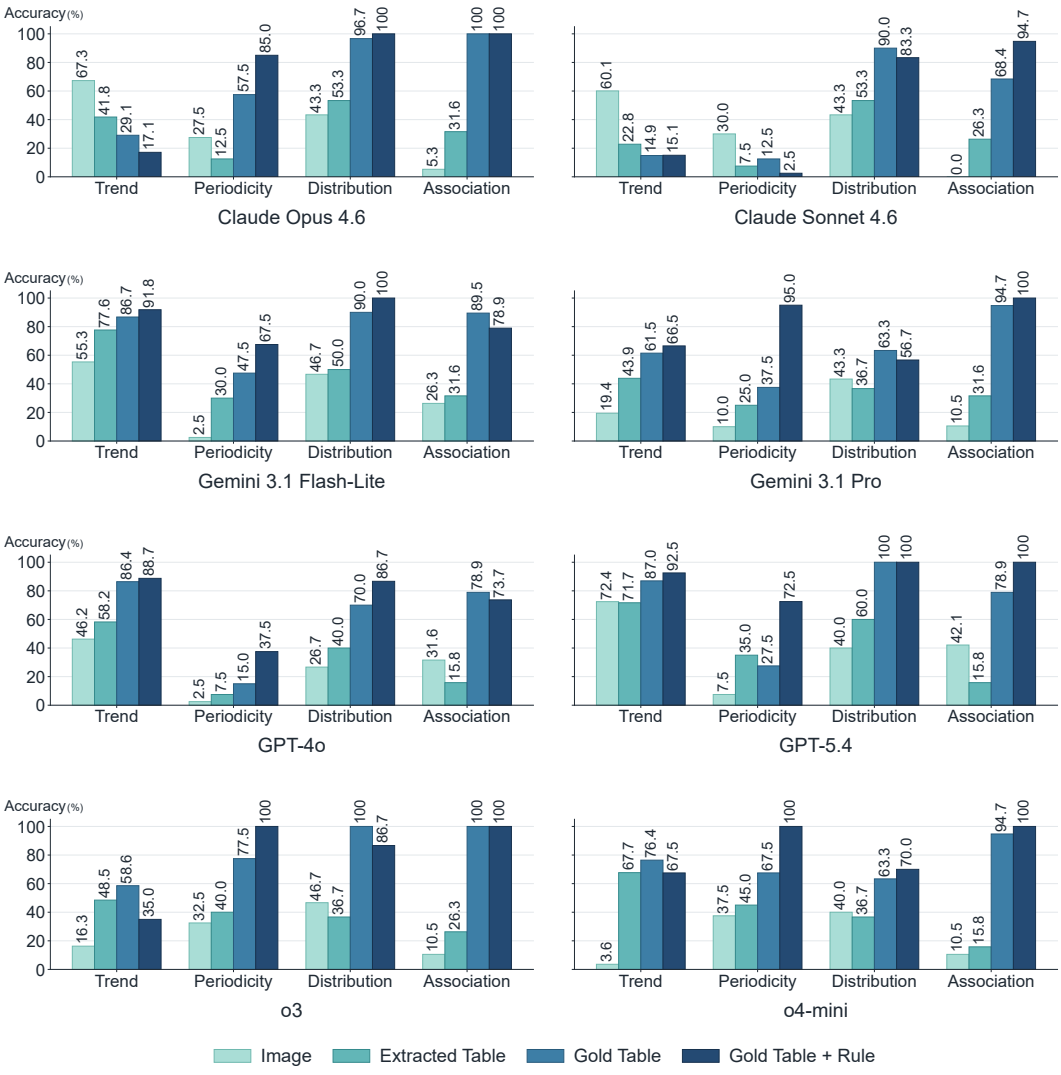


Figure 18: Bottleneck analysis for eight proprietary MLLMs.

☞ **Finding 5:** *Proprietary MLLMs exhibit the same overall pattern: gold tables and reference rules generally improve performance, although the magnitude of the gains varies across regularities.*

G Robustness and Shortcut Analyses

To examine whether model performance is overly dependent on particular question forms, rule realizations, data sources, or superficial cues, we conduct a set of complementary robustness analyses on a deterministic $\sim$10% pilot of the benchmark. The pilot is sampled independently of model predictions and is evaluated with two representative open-source models, Qwen3-VL-8B-Instruct and InternVL3-8B. We consider two types of analyses: cross-group robustness across benchmark characteristics and controlled tests for potential shortcuts.

G.1 Cross-Group Robustness

Rule-realization split. To examine robustness across different manifestations of the same high-level regularity, we group charts by their concrete rule realization within Trend, Periodicity, Distribution, and Association. Examples include linear versus piecewise trends, different periodic structures, and

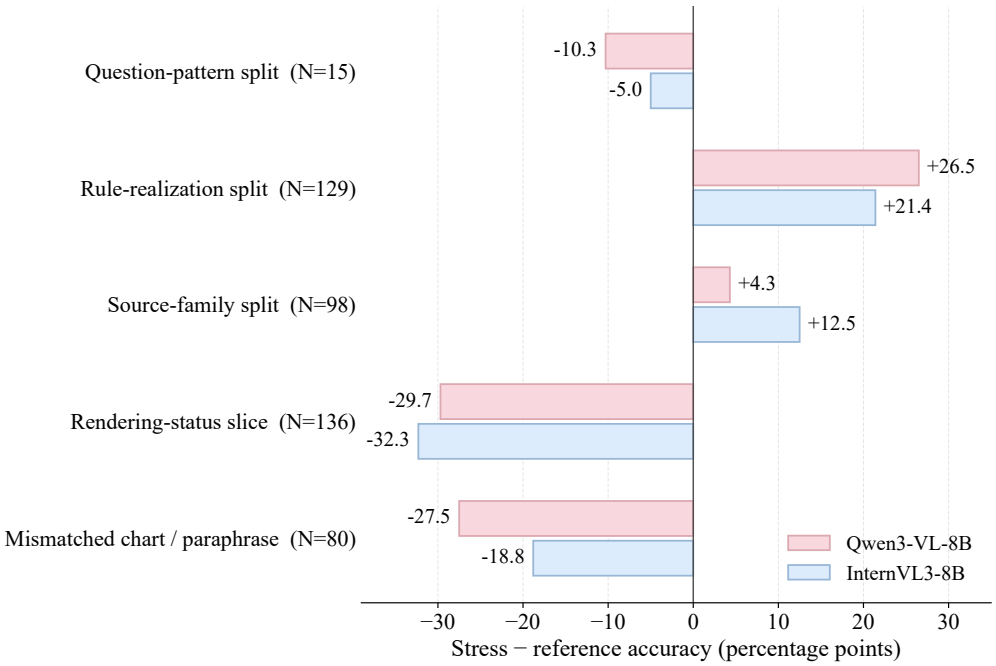


Figure 19: Robustness and shortcut analyses on the sampled open-source-model pilot. Δ denotes stress minus reference accuracy; every plotted shift and stress-slice sample size is labeled directly.

linear versus nonlinear associations. Complete realization groups are assigned to the evaluation slice while the corresponding high-level regularity remains represented in the reference set.

Source-family split. To measure sensitivity to data provenance, we partition examples by complete source families and compare performance between the selected source-family slice and the remaining examples.

Rendering-status slice. We additionally compare performance across original public charts, charts rendered from public tables, and controlled generated charts. This analysis captures performance variation across different chart-production settings. Because rendering status is correlated with source, regularity, and chart type, we treat the result as a subgroup comparison rather than an isolated visual style effect.

G.2 Shortcut Controls

We further construct two controlled interventions to test whether models rely on chart evidence and whether predictions are sensitive to superficial wording.

Mismatched-chart control. Each question is paired with another chart that has the same regularity category and answer format but a different gold answer. If a model relies on the visual evidence rather than question priors alone, its accuracy should decrease after the chart is replaced.

Question-paraphrase control. We keep the chart, target rule, and answer fixed while paraphrasing the question without changing its semantics. A model that is robust to superficial wording changes should produce consistent predictions before and after paraphrasing.

Figure 19 summarizes subgroup differences and the paired shortcut controls. Only chart replacement and paraphrasing modify the same question instances; the cross-group slices compare different items.

The exact reference-to-stress accuracies for Qwen3-VL-8B-Instruct are 37.0 to 26.7 (question pattern, $N = 15$), 27.8 to 54.3 (rule realization, $N = 129$), 35.5 to 39.8 (source family, $N = 98$), 48.1 to 18.4 (rendering status, $N = 136$), and 55.0 to 27.5 (mismatched chart, $N = 80$). The corresponding

InternVL3-8B values are 31.6 to 26.7, 24.3 to 45.7, 28.3 to 40.8, 43.4 to 11.0, and 47.5 to 28.8. Question-paraphrase consistency is 97.5% for Qwen3-VL-8B-Instruct and 98.8% for InternVL3-8B.

The pilot initially contains 393 clean QAs. Five TimeSynth-referenced paired-series examples use an unsupported list-valued option format and are excluded for both models, leaving 388 valid examples. Overall accuracy on these examples is 36.6% for Qwen3-VL-8B-Instruct and 31.4% for InternVL3-8B.

The cross-group analyses reveal different forms of performance variation. Accuracy is lower by 10.3 and 5.0 points on the question-pattern slice. Because this slice contains different items, the difference does not isolate sensitivity to wording. In contrast, the selected rule-realization and source-family slices obtain higher accuracy than their reference groups for both models. This indicates that these groups are not uniformly more difficult and that the observed differences primarily reflect heterogeneous difficulty across benchmark subgroups. The largest difference occurs across rendering status, where performance decreases by 29.7 and 32.3 points on the selected rendered-chart slice.

The shortcut controls provide a clearer test of model behavior. Replacing the original chart with a mismatched chart reduces accuracy from 55.0% to 27.5% for Qwen3-VL-8B-Instruct and from 47.5% to 28.8% for InternVL3-8B, indicating that both models substantially rely on chart-specific evidence. In contrast, paraphrasing the question preserves model predictions on 97.5% and 98.8% of examples, respectively, showing stability on the tested paraphrases. These controls do not eliminate all textual shortcuts. In Table 2, text-only GPT-5.4 reaches 44.1% on Distribution EI, compared with 3.2% on Trend and 2.3% on Periodicity. Thus some questions can be answered without the image. Without random-choice, majority-label, and option-frequency controls, the contribution of particular textual cues is unresolved.

H From Rule Induction to Example Inference

When a rule is correctly identified, can it be successfully applied? To examine whether models can successfully apply a correctly identified regularity, we sample 160 matched RI–EI pairs, balanced across the four regularity categories to account for the uneven distribution of the full benchmark. Each pair shares the same chart and verified target regularity. When multiple RI or EI questions are available for the same chart–regularity pair, we deterministically select one question from each task without reference to model predictions. We report the conditional EI accuracy given correct or incorrect RI, together with the four possible RI–EI outcome combinations. Figure 20 shows the joint RI–EI outcome distribution for each model.

The conditional EI accuracies given correct versus incorrect RI are 44.9% versus 30.6% for GPT-5.4, 38.5% versus 43.2% for o3, 39.1% versus 28.1% for Qwen3-VL-8B-Instruct, and 33.9% versus 22.8% for InternVL3-8B. EI accuracy is higher when RI is correct for three of the four models, with o3 being the exception. We speculate that o3 may be better able to generalize directly from chart evidence, even when it fails to explicitly identify the corresponding rule in RI.

Nevertheless, all four models frequently fail on EI even when RI is correct. Specifically, EI fails on 27 of 49 RI-correct pairs (55.1%) for GPT-5.4, 40 of 65 (61.5%) for o3, 39 of 64 (60.9%) for Qwen3-VL-8B-Instruct, and 39 of 59 (66.1%) for InternVL3-8B. These results show that correctly identifying a regularity does not guarantee success on its corresponding application question.

☞ **Finding 6:** *Correctly identifying a regularity does not guarantee its successful application, with all four models failing on more than half of their RI-correct pairs.*

I Evaluation Details

Model checkpoints. Table 9 lists the exact API model identifiers and open-source checkpoints used in our experiments. We evaluate proprietary models via their APIs and run open-source models using their publicly released Hugging Face checkpoints. Throughout the appendix, LLaVA-OneVision-1.5-8B denotes the Instruct checkpoint, and TinyChart denotes the 3B model with 768-pixel input resolution; their exact checkpoint identifiers are given in Table 9.

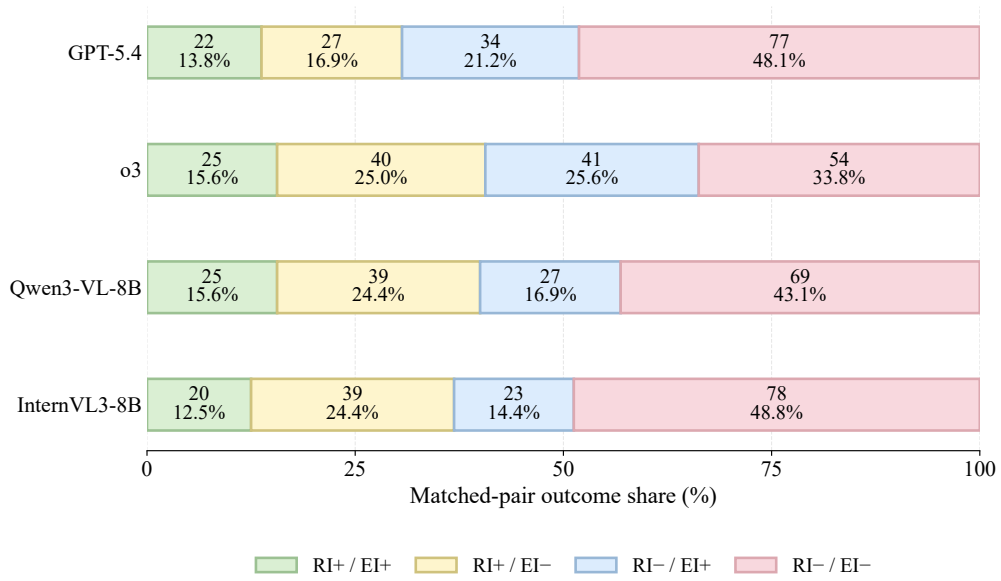


Figure 20: Joint Rule Induction (RI) and Example Inference (EI) outcomes on 160 matched chart–regularity pairs per model. Each segment reports its exact pair count and share; + and − denote correct and incorrect predictions, respectively.

Model	Checkpoint / API Model ID	Decoding	Max Output Tokens
Proprietary MLLMs			
Claude Opus 4.6	`anthropic/claude-opus-4.6`	Temperature 0	1024
Claude Sonnet 4.6	`anthropic/claude-sonnet-4.6`	Temperature 0	1024
Gemini 3.1 Flash-Lite	`google/gemini-3.1-flash-lite-preview`	Temperature 0	1024
Gemini 3.1 Pro	`google/gemini-3.1-pro-preview`	Temperature 0	1024
GPT-4o	`openai/gpt-4o`	Temperature 0	1024
GPT-5.4	`openai/gpt-5.4`	Temperature 0	1024
o3	`openai/o3`	Native reasoning	1024
o4-mini	`openai/o4-mini`	Native reasoning	1024
Open-Source MLLMs			
InternVL3-8B	`OpenGVLab/InternVL3-8B`	Greedy	1024
LLaVA-OneVision-1.5-8B	`Deep-VLM/LLaVA-OneVision-1.5-8B-Instruct-hf`	Greedy	1024
ChartGemma	`ahmed-masry/chartgemma`	Greedy	512
MiniCPM-V 4.5	`openbmb/MiniCPM-V-4_5`	Greedy	1024
TinyChart	`mPLUG/TinyChart-3B-768`	Greedy	512
Qwen3-VL-8B-Instruct	`Qwen/Qwen3-VL-8B-Instruct`	Greedy	1024

Table 9: Model checkpoints and inference configurations used in our experiments.

Inference settings. We use deterministic decoding whenever supported. For open-source models, we use greedy decoding (`do_sample=False`); for proprietary models, we set the temperature to 0 when temperature control is supported. Reasoning models use their native reasoning configuration. The maximum output length is set to 1,024 tokens for general-purpose models and 512 tokens for ChartGemma and TinyChart.

Prompting and scoring. In the main model evaluation, each instance contains the chart, question, and its declared answer format. For multiple-choice questions, predictions are scored by exact match on the normalized option label. Numerical predictions are considered correct when all requested values are parseable and fall within a 5% relative-error tolerance of the reference answer.

J Evaluation Prompts

Figures 21–23 document prompt assembly in the released evaluation runners and the four-condition diagnostic experiments. Placeholders in braces are substituted at runtime. Bracketed image markers

indicate attached images and are not literal prompt text. Bold headings and explanatory descriptions organize this appendix; they are not added to the model input. Task-specific question wording is described in Table 7. For the Text-only (GPT-5.4) baseline, the chart image is removed while all other inputs and evaluation settings remain unchanged. We additionally provide the following instruction: *"You are given a chart question, but the chart image is not available. Make the most reasonable guess using only the question text and the provided answer format. Return only the final answer."*

Historical image baseline. For proprietary models, the Image condition reuses historical original-image predictions for the paired evaluation. The direct-answering template in Figure 22 documents the diagnostic runner's image-call interface and does not establish the exact wording of the earlier calls.

Information supplied by the reference rule. The {gold_rule} field provides additional reference information whose specificity depends on the regularity type. For trend, the guidance can provide fitted coefficients; for periodicity, the canonical cycle peak and temporal wrap-around information; for distribution, the predicted local-majority class; and for association, a verified cross-chart symbol mapping. Some classification instances therefore explicitly receive the target class in the supplied guidance. Accordingly, Gold Table + Rule should be interpreted as an oracle-style diagnostic of performance under supplied reference rule information, rather than as an independent test of rule induction.

For example, the distribution guidance uses:

> The fully instantiated latent local decision rule for the stated neighborhood predicts class '{class}' because it is the unique local majority. Do not recompute or replace this rule; apply it to the Example Inference query and return exactly the required class label.

Evaluation Prompts for the Five Task Types

Input Contract:

- **Contains:** chart image, original question text, and optional answer choices.
- **Question text:** preserves task-specific instructions and any requested answer format.
- **Options:** one option per line, formatted as `A. {option_text}`, in the supplied order.

Task Types:

1. **Rule Verification:** verify a proposed chart regularity.
2. **Rule Identification:** identify or describe the regularity supported by the chart.
3. **Numerical Prediction:** predict the numerical target specified in the question.
4. **Categorical Prediction:** predict the requested category or state.
5. **Multi-Target Prediction:** predict the requested collection or structured target.

Shared Input Structure:

[Chart Image]
{question}
{choice_suffix}

Choice Suffix — Open-Source Runner:

For questions with explicit choices, append a blank line followed by:

Options:
{options}
Answer with only the option key.

Choice Suffix — Proprietary-Model Runner:

For questions with explicit choices, append a blank line followed by:

{options}
Answer only with the option key.

Output Specification:

When explicit choices are absent, {choice_suffix} is empty. The runner retains the original question and its answer requirements without adding a universal output instruction. Task type alone does not determine answer format: a numerical target, for example, may be queried through either multiple choice or free response.

Figure 21: Prompt assembly for the five task types in the released evaluation runners. The two choice suffixes reproduce the wording of their respective implementations.

Diagnostic Prompts: Image and Table Extraction

Input Contract:

Both stages shown here receive the chart image. The image-answering call predicts the answer directly; the extraction call produces a table for a subsequent, separate answer call.

Shared Question Block:

Question: {question}
Options:
{options}

The `Options:` line and option list are omitted when explicit choices are absent. This block is denoted by {question_block} below.

Image — Direct Answering:

[Chart Image]

Infer the latent regularity from the chart, then apply it to answer the Example Inference question. Do not merely transcribe the chart.

{question_block}

{answer_instruction}

Extracted Table — Stage 1:

[Chart Image]

Extract the chart's numerical data that are relevant to the question below. Preserve labels, x-values, y-values, series names, and ordering. For heatmaps give the relevant row/column values; for distributions give exact-looking bin or group counts/summary values visible in the chart. Output only a compact Markdown table or CSV-like table. Do not answer the question and do not state the latent rule.

{question_block}

Implementation Details:

- The extraction prompt above reproduces the open-source wording. The proprietary runner omits "the" before "relevant row/column values".
- The extracted text becomes {extracted_table} in the next call. The chart is withheld from that answer call.

Figure 22: Image-answering and chart-to-table extraction prompts in the diagnostic runners. Table-based answering templates are shown in Figure 23.

Diagnostic Prompts: Table-Based Answering

Input Contract:

The answer call receives a textual table, the question, optional choices, and an answer instruction. Gold Table + Rule additionally supplies reference guidance. The original chart is withheld; a fixed blank 16×16 white compatibility image containing no task-relevant information is used when required by an open-source implementation.

Extracted Table — Stage 2:

The following is a table extracted from the chart by the same model. Use it instead of a chart image. Infer the latent regularity and apply it.

TABLE
{extracted_table}

{question_block}

{answer_instruction}

Gold Table:

The following is a verified gold table. Use it instead of a chart image. Infer the latent regularity and apply it.

TABLE
{gold_table}

{question_block}

{answer_instruction}

Gold Table + Rule:

The following is a verified gold table. Use it instead of a chart image. Apply the supplied fully instantiated latent rule.

TABLE
{gold_table}

GOLD LATENT RULE
{gold_rule}

Apply this supplied rule; do not re-induce a different rule.

{question_block}

{answer_instruction}

Answer Instruction — First Applicable Case:

1. **Explicit choices:** Return only the option key (for example A, B, C, or D), with no explanation.
2. **Regression format:** Return only one numeric value, with no units or explanation.
3. **Boolean reference answer:** Return only true or false.
4. **Otherwise:** Return only the concise answer in the format requested by the question.

Figure 23: Prompt templates for the three table-based answering conditions, with runtime fields represented by placeholders.

K FAILURE CASES

We present 20 model–reference disagreements organized into four descriptive categories: **Perception Error**, **Over-generalization**, **Over-specificity**, and **Incorrect Rule Identification**.

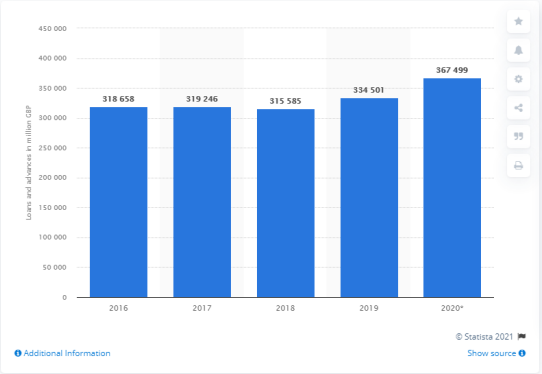
K.1 PERCEPTION ERROR

Perception errors occur when the visual evidence itself is read incorrectly, such as a value, coordinate, relative magnitude, color intensity, or trend direction. These descriptions concern observable responses rather than the internal order of reasoning.

K.1.1 EXAMPLE 1: LOCATING THE BREAK IN AN INCREASING SERIES

Chart (Trend)

Question. In 'Loans and advances in million GBP', which x-value corresponds to the point that breaks the otherwise monotonic trend?



A) 2017

B) 2018

C) 2019

D) 2020

GPT-5.4's Reasoning Process

The totals rise from 2015 through 2018, dip only slightly in 2019, and then drop much more sharply in 2020. Relative to the generally increasing pattern before it, 2020 breaks the trend most clearly. That makes 2020 the standout non-conforming year.

So D) 2020 is the best fit.

Figure 24: Failure case 1.

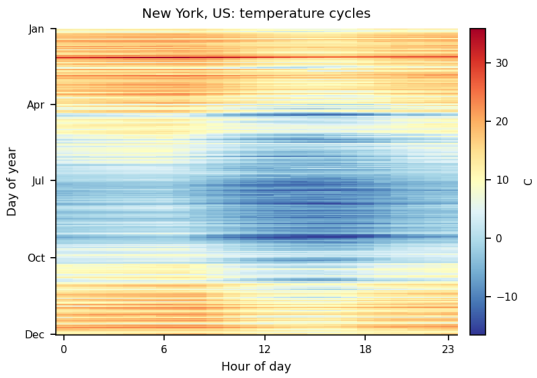
Model response. The totals rise from 2015 through 2018, dip only slightly in 2019, and then drop much more sharply in 2020. Relative to the generally increasing pattern before it, 2020 breaks the trend most clearly. That makes 2020 the standout non-conforming year. So D) 2020 is the best fit.

Failure analysis. GPT-5.4 selects D (2020), describing a sharp decline at the end of the series. The labeled bars are 318,658, 319,246, 315,585, 334,501, and 367,499 for 2016–2020. The only decrease occurs from 2017 to 2018, so B (2018) is the reference answer; 2020 is the highest bar. Misperceiving the relative bar heights leads directly to a *perception error* in the response.

K.1.2 Example 2: Reading the phase of a temperature heatmap

Chart (Periodicity)

Question. Which rule best describes the summer daily temperature cycle in New York?



A) The highest temperature occurs around 06:00.

B) The highest temperature occurs around 12:00.

C) The highest temperature occurs around 15:00.

D) The highest temperature occurs around 18:00.

GPT-5.4's Reasoning Process

The New York heatmap is warmest in the afternoon columns during summer, with the hottest band centered later than noon and earlier than evening. The **strongest warm colors cluster around about 15:00.** That rules out 06:00, 12:00, and 18:00.

So **C) The highest temperature occurs around 15:00** is the best fit.

Figure 25: Failure case 2.

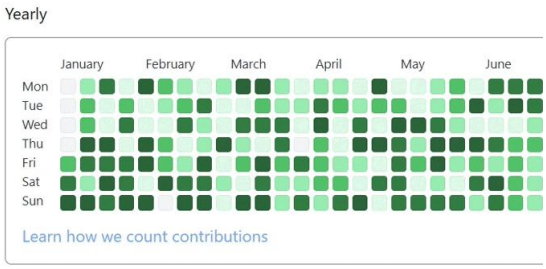
Model response. The New York heatmap is warmest in the afternoon columns during summer, with the hottest band centered later than noon and earlier than evening. The strongest warm colors cluster around about 15:00. That rules out 06:00, 12:00, and 18:00. So C) The highest temperature occurs around 15:00 is the best fit.

Failure analysis. GPT-5.4 selects C (15:00), whereas the reference answer A corresponds to the warmest band near 06:00. The afternoon columns are visibly cooler in the displayed heatmap. Assigning the peak to the wrong horizontal position is a *perception error* in phase localization, rather than a failure to understand periodicity.

K.1.3 Example 3: Comparing contribution intensity across calendar rows

Chart (Periodicity)

Question. Which option best describes the contribution-activity heatmap?



A) January is the least busy month among all months.

B) Thursday is busier than Friday.

C) Sunday is busier than Friday because it has more light-colored cells.

D) March is the least busy month among all months.

GPT-5.4's Reasoning Process

I compared the month blocks shown in the heatmap and March contains more pale cells overall than January. **Friday is not lighter than Thursday** often enough to support the busier-Thursday claim, and Sunday is not clearly busier than Friday across the whole span. Among the listed month statements, **March looks least busy**.

So **D) March is the least busy month among all months** is the best fit.

Figure 26: Failure case 3.

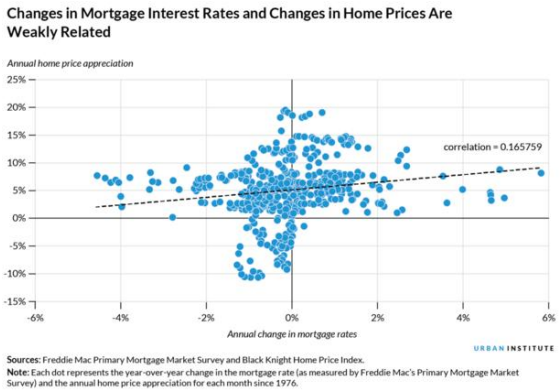
Model response. I compared the month blocks shown in the heatmap and March contains more pale cells overall than January. Friday is not lighter than Thursday often enough to support the busier-Thursday claim, and Sunday is not clearly busier than Friday across the whole span. Among the listed month statements, March looks least busy. So D) March is the least busy month among all months is the best fit.

Failure analysis. GPT-5.4 selects D, claiming that March is the least busy month, whereas the reference answer is B: Thursday is busier than Friday. The response relies on a visually salient pale region in March and dismisses the weekday comparison without consistently aligning corresponding weeks. Misreading the color intensities and row alignment therefore produces a *perception error*.

K.1.4 EXAMPLE 4: DISTINGUISHING A SPARSE TAIL FROM THE DENSE CENTER

Chart (Distribution)

Question. Which option best describes where observations are relatively sparse or dense in the scatter plot?



A) Annual home-price-appreciation observations are sparse only when the annual change in mortgage rates is near 6%.

B) Annual home-price-appreciation observations are dense only when the annual change in mortgage rates is near 0%.

C) Annual home-price-appreciation observations are dense only when the annual change in mortgage rates is near -2%.

D) Annual home-price-appreciation observations are sparse only when the annual change in mortgage rates is near -4%.

GPT-5.4's Reasoning Process

The densest concentration of points is around an annual mortgage-rate change **near 0%**, with many observations packed vertically there. The far left near -4% and far right near 6% are both relatively sparse, so the sparse-only claims are too narrow. The best description is that density is concentrated near 0%.

So **B) Annual home-price-appreciation observations are dense only when the annual change in mortgage rates is near 0%** is the best fit.

Figure 27: Failure case 4.

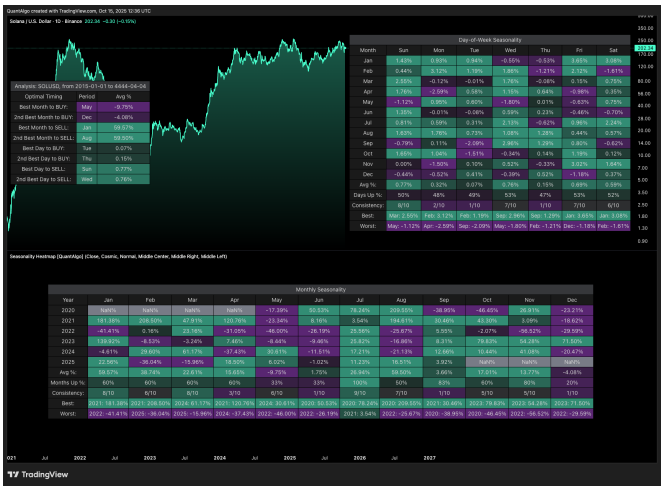
Model response. The densest concentration of points is around an annual mortgage-rate change near 0%, with many observations packed vertically there. The far left near -4% and far right near 6% are both relatively sparse, so the sparse-only claims are too narrow. The best description is that density is concentrated near 0%. So B) Annual home-price-appreciation observations are dense only when the annual change in mortgage rates is near 0% is the best fit.

Case discussion. GPT-5.4 selects B by focusing on the dense center near 0%, whereas the reference answer D corresponds to the sparse region near -4%. This reflects a *perception error*, where the model identifies a salient visual pattern but fails to locate the chart region relevant to the correct answer.

K.1.5 EXAMPLE 5: READING THE CORRECT ROW IN SEASONALITY TABLES

Chart (Periodicity)

Question. Which option best describes the lowest average return in the day-of-week or monthly seasonality tables?



A) June has the lowest average return among the twelve months.

B) Tuesday has the lowest average return among the days of the week.

C) Thursday has the lowest average return among the days of the week.

D) December has the lowest average return among the twelve months.

GPT-5.4's Reasoning Process

I compared the average rows in both tables. In the monthly seasonality table, **December has the most negative average return**, while June is positive; in the day-of-week table, neither Tuesday nor Thursday is lower than December. Therefore the lowest average return shown is December among the months.

So **D) December has the lowest average return among the twelve months** is the best fit.

Figure 28: Failure case 5.

Model response. I compared the average rows in both tables. In the monthly seasonality table, December has the most negative average return, while June is positive; in the day-of-week table, neither Tuesday nor Thursday is lower than December. Therefore the lowest average return shown is December among the months. So D) December has the lowest average return among the twelve months is the best fit.

Failure analysis. GPT-5.4 selects D and states that December has the lowest monthly average. In the monthly Avg % row, however, May is -9.75%, below December at -4.08%, so D is false. In the weekday table, Tuesday has the lowest average at 0.07%, below Thursday at 0.15%, making B the supported option. Misreading the extrema and mixing the two temporal groupings produces a *perception error*.

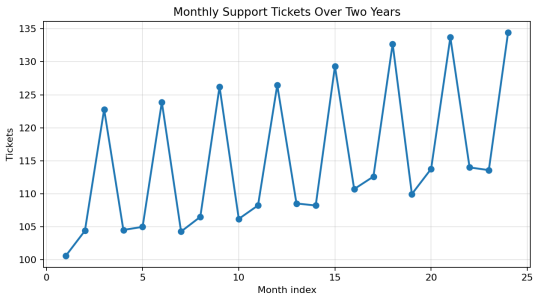
K.2 Over-generalization

Over-generalization occurs when a pattern supported only within a particular interval, subset, or region is extended to a broader domain. The local observation may be valid, but its scope is not preserved.

K.2.1 Example 6: Extending a short cycle into an annual rule

Chart (Periodicity)

Question. What is the main repeating cycle in this support-tickets chart?



A) 3 months

B) 7 days

C) 12 months

D) 24 hours

GPT-5.4's Reasoning Process

The chart shows a clear seasonal pattern where ticket counts rise and fall in a similar way across the first 12 months and then **repeat across the next 12 months,** with peaks appearing at regular yearly intervals. That repeating shape over two years indicates the main cycle is annual rather than daily, weekly, or every 3 months.

So **C) 12 months** is the best fit.

Figure 29: Failure case 6.

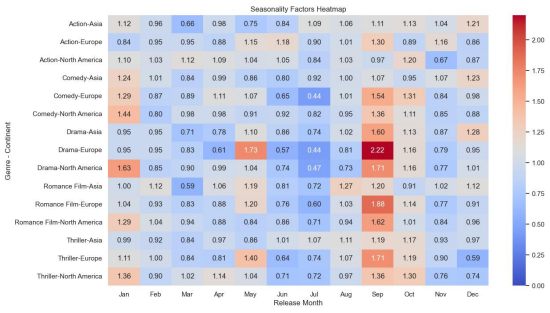
Model response. The chart shows a clear seasonal pattern where ticket counts rise and fall in a similar way across the first 12 months and then repeat across the next 12 months, with peaks appearing at regular yearly intervals. That repeating shape over two years indicates the main cycle is annual rather than daily, weekly, or every 3 months. So C) 12 months is the best fit.

Failure analysis. GPT-5.4 selects C (12 months), interpreting the two displayed years as repeated annual profiles. However, peaks recur at approximately months 3, 6, 9, 12, 15, 18, 21, and 24, supporting A (3 months) as the dominant cycle. A 12-month interval spans four repetitions of this shorter cycle. Elevating a coarse similarity between annual blocks over the repeated local cycle leads to *over-generalization*.

K.2.2 Example 7: Generalizing a month comparison across categories

Chart (Periodicity)

Question. Which statement best captures the dominant month-to-month relationship across the 15 genre–continent categories?



A) May tends to have a higher seasonality factor than April.

B) February tends to have a higher seasonality factor than April.

C) December tends to have a higher seasonality factor than February.

D) November tends to have a higher seasonality factor than March.

GPT-5.4's Reasoning Process

Across the 15 genre–continent combinations, November is slightly more often higher than March than the other listed comparisons are true. **The heatmap shows many rows where March is relatively lower while November is modestly higher.**

So **A) May tends to have a higher seasonality factor than April.**

is the best fit.

Figure 30: Failure case 7.

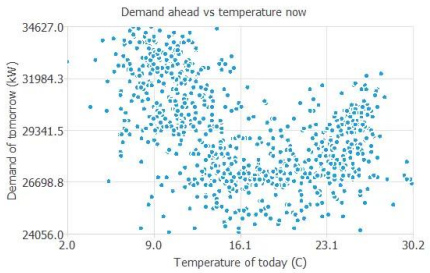
Model response. Across the 15 genre–continent combinations, November is slightly more often higher than March than the other listed comparisons are true. The heatmap shows many rows where March is relatively lower while November is modestly higher. So A) May tends to have a higher seasonality factor than April. is the best fit.

Case discussion. The selected answer A generalizes the May–April ordering into a recurring pattern, although May exceeds April in only 6 of the 15 displayed rows. The accompanying rationale similarly draws on comparisons involving other months rather than establishing that this ordering holds broadly across the observations. This reflects *over-generalization*: the model elevates patterns supported by only a subset of the data into a chart-level regularity.

K.2.3 Example 8: Extending one arm of a U-shaped relationship

Chart (Distribution)

Question. Which option best describes the relationship between today's temperature and tomorrow's demand in the scatter plot?



A) The higher today's temperature is, the higher tomorrow's demand tends to be.

B) The lower today's temperature is, the higher tomorrow's demand tends to be.

C) When today's temperature is around 30°C, tomorrow's demand is relatively low.

D) When today's temperature is around 20°C, tomorrow's demand is relatively low.

GPT-5.4's Reasoning Process

The scatter has a high-demand cluster at lower temperatures around 7 to 12 C, while many points from roughly 15 to 25 C sit lower on the y-axis. I compared the **low-temperature and mid-temperature regions** and the overall pattern slopes downward more than upward. That rules out the positive-Association options, and the low-demand claim near 30 C is weaker than the broader low band near about 20 C.

So **B) The lower today's temperature is, the higher tomorrow's demand tends to be** is the best fit.

Figure 31: Failure case 8.

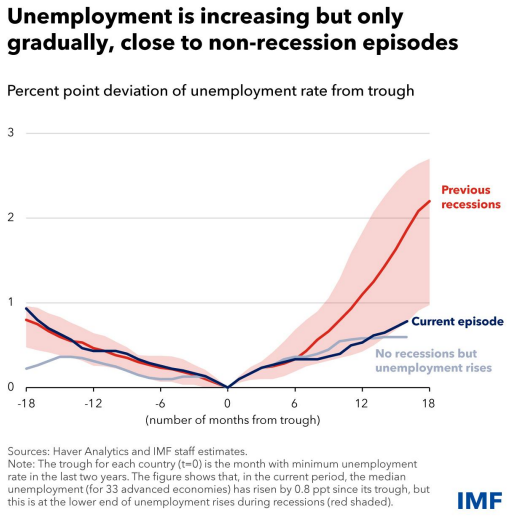
Model response. The scatter has a high-demand cluster at lower temperatures around 7 to 12 C, while many points from roughly 15 to 25 C sit lower on the y-axis. I compared the low-temperature and mid-temperature regions and the overall pattern slopes downward more than upward. That rules out the positive-correlation options, and the low-demand claim near 30 C is weaker than the broader low band near about 20 C. So B) The lower today's temperature is, the higher tomorrow's demand tends to be is the best fit.

Failure analysis. GPT-5.4 selects B, interpreting the scatter as a globally decreasing relationship between temperature and next-day demand. This description is supported only on the left side of the plot: demand decreases from the lower-temperature region toward the middle, but rises again at higher temperatures. The overall pattern is U-shaped, with a trough around 20°C, which supports D. The error is an *over-generalization* from one arm of a nonlinear relationship to the full temperature range.

K.2.4 EXAMPLE 9: EXTENDING THE POST-TROUGH COMPARISON TO THE WHOLE EPISODE

Chart (Association)

Question. Which option best describes how the current unemployment episode compares with previous recession episodes in the chart?



A) At 18 months from the trough, the current episode is closer to recession episodes.

B) The current episode does not resemble a recession episode.

C) At -18 months from the trough, the current episode is closer to recession episodes.

D) The current episode resembles a recession episode.

GPT-5.4's Reasoning Process

The blue current-episode line stays much closer to the gray non-recession line than to the red recession line, especially after the trough. **By 12 to 18 months**, the recession path rises far above the current episode while the non-recession path remains nearby. So the chart does not support the idea that the current episode resembles recessions.

So **B) The current episode does not resemble a recession episode** is the best fit.

Figure 32: Failure case 9.

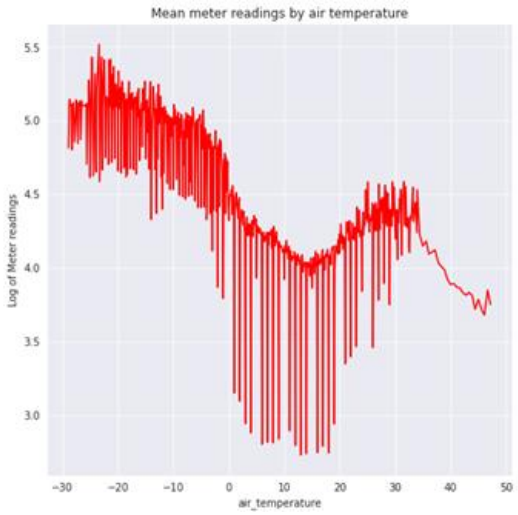
Model response. The blue current-episode line stays much closer to the gray non-recession line than to the red recession line, especially after the trough. By 12 to 18 months, the recession path rises far above the current episode while the non-recession path remains nearby. So the chart does not support the idea that the current episode resembles recessions. So B) The current episode does not resemble a recession episode is the best fit.

Case discussion. GPT-5.4 selects the broad non-recession description B, while reference C concerns proximity to the recession path at -18 months. The later trajectory and the chart title support the model's broader comparison. This illustrates a difference in temporal scope rather than a uniquely incorrect global description.

K.2.5 Example 10: Replacing a local trough with a global negative rule

Chart (Association)

Question. Which option best describes the relationship between air temperature and the logarithm of meter readings?



A) Air temperature and the logarithm of meter readings are negatively correlated.

B) The logarithm of meter readings is lowest when air temperature is between 0 and 20.

C) The higher the air temperature, the higher the logarithm of meter readings.

D) The lower the air temperature, the higher the logarithm of meter readings.

GPT-5.4's Reasoning Process

I traced the red line from cold to warm temperatures: it starts higher at low temperatures, declines toward the middle, only partly rebounds, and then falls again at the warmest end. **The overall left-to-right tendency is downward rather than upward.** That makes the negative-Association description the best fit.

So **A) Air temperature and the logarithm of meter readings are negatively correlated** is the best fit.

Figure 33: Failure case 10.

Model response. I traced the red line from cold to warm temperatures: it starts higher at low temperatures, declines toward the middle, only partly rebounds, and then falls again at the warmest end. The overall left-to-right tendency is downward rather than upward. That makes the negative-correlation description the best fit. So A) Air temperature and the logarithm of meter readings are negatively correlated is the best fit.

Case discussion. GPT-5.4 selects the global negative-association description A, while reference B identifies low values over the 0–20 temperature range. The overall downward envelope and the local trough describe different aspects of the same curve. Their coexistence makes the disagreement sensitive to the scope intended by the question.

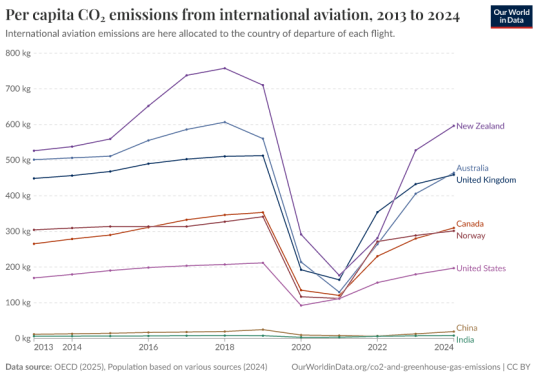
K.3 Over-specificity

Over-specificity occurs when a response adds unnecessary local structure, segment boundaries, conditions, or rankings instead of retaining the broader regularity requested by the question.

K.3.1 Example 11: Over-segmenting an overall emissions trend

Chart (Trend)

Question. Which pattern best summarizes the overall trend in Australia's per-capita aviation emissions from 2013 to 2024?



A) Emissions increased from 2013 to 2018, decreased from 2018 to 2021, and fluctuated from 2021 to 2024.

B) Emissions decreased overall from 2013 to 2024, with fluctuations along the way.

C) Emissions increased from 2013 to 2018, then decreased from 2018 to 2024.

D) Emissions decreased from 2013 to 2021, then increased from 2021 to 2024.

GPT-5.4's Reasoning Process

Australia's line rises from about 500 kg in 2013 to just over 600 kg around 2018, then drops sharply by 2020–2021 and only partially recovers to around 460 kg by 2024, still below the 2018 peak.

So C is the best choice.

Figure 34: Failure case 11.

Model response. Australia's line rises from about 500 kg in 2013 to just over 600 kg around 2018, then drops sharply by 2020–2021 and only partially recovers to around 460 kg by 2024, still below the 2018 peak. So C is the best choice.

Case discussion. Reference B summarizes the net decrease over 2013–2024, whereas GPT-5.4 selects C and describes the rise to 2018 followed by a decline and partial recovery. The disagreement concerns the granularity of the summary: local segmentation alone does not invalidate a full-range description.

K.3.2 Example 12: Adding an unnecessary ordering within the weekend

Chart (Periodicity)

Question. Which rule best captures the common pattern across the observations?



A) Daily RevPAR is higher on Fridays and Saturdays than on Sundays through Thursdays.

B) Daily RevPAR is highest on Saturdays, followed by Fridays, and lower on Sundays through Thursdays.

C) Daily RevPAR is highest on Fridays and Saturdays, followed by Sundays, and lowest on Mondays through Thursdays.

D) Daily RevPAR is highest on Fridays and Saturdays, followed by Sundays, then Thursdays, and lowest on Mondays through Wednesdays.

GPT-5.4's Reasoning Process

Across most months, **the Saturday column is most often the darkest teal/green (highest RevPAR), Friday is usually the next strongest, and Sunday–Thursday are generally lighter or redder** by comparison. The weekend pattern is visible repeatedly even as overall seasonality changes month to month.

So **B) Daily RevPAR is highest on Saturdays, followed by Fridays, and lower on Sundays through Thursdays** is correct.

Figure 35: Failure case 12.

Model response. Across most months, the Saturday column is most often the darkest teal/green (highest RevPAR), Friday is usually the next strongest, and Sunday–Thursday are generally lighter or redder by comparison. The weekend pattern is visible repeatedly even as overall seasonality changes month to month. So B) Daily RevPAR is highest on Saturdays, followed by Fridays, and lower on Sundays through Thursdays is correct.

Case discussion. Reference A groups Fridays and Saturdays above the other weekdays. GPT-5.4 selects B, which adds a Saturday-before-Friday ordering. The stronger ordering requires support across the relevant rows; it is not contradicted merely by the reference's broader grouping.

K.3.3 Example 13: Adding secondary peaks to a common seasonal rule

Chart (Periodicity)

Question. Which rule best captures the common pattern across the observations?

Fig. 3 More than One Peak Season for Sales

	Jan	Feb	Mar	Apr	May	Jun	Jul	Aug	Sep	Oct	Nov	Dec
Hobby, toy, and games	100	102	121	110	110	105	106	108	113	125	192	276
Family clothing stores	100	111	142	136	145	138	145	152	132	145	172	224
Gifts and Novelties	100	110	121	126	148	145	152	146	142	193	150	221
Department stores	100	106	124	118	127	122	119	130	114	122	155	215
Clothing stores	100	109	138	133	140	132	135	142	127	136	159	205
Sporting goods stores	100	102	134	128	132	145	139	148	121	117	134	198
Shoe stores	100	125	149	139	140	130	141	179	128	130	144	193
Electronics stores	100	98	103	91	97	99	100	105	99	99	131	168
Online Shopping	100	94	107	104	108	105	105	110	106	112	132	165
Beer, wine, and liquor	100	101	113	112	123	122	127	123	116	120	126	164
Total ex Autos, Fuel and Food	100	98	111	109	116	112	112	115	108	113	119	137
Household appliance stores	100	101	113	107	118	118	119	118	114	113	128	127
Hardware stores	100	95	117	127	139	131	123	122	114	123	119	122
Building Materials	100	98	125	138	150	144	138	135	129	133	127	118
Health and personal care	100	96	105	101	103	101	101	103	100	104	102	116
Furniture stores	100	102	116	104	111	108	111	115	113	108	115	115
Food and beverage stores	100	94	103	100	106	103	106	105	101	104	105	114
Pharmacies and drug stores	100	94	103	100	102	99	99	101	98	103	100	112
Paint and wallpaper stores	100	101	129	133	143	150	148	152	141	139	118	104
Office Supplies	100	87	91	84	82	77	83	118	94	83	81	93
Book stores	100	49	49	48	55	52	51	108	74	51	52	92

Reading from left to right, green indicates higher values and red indicates lower values
Source: Flexport calculations based on U.S. Census Bureau data.

A) Sales in December are higher than in November, May, and August, and sales in those three months are higher than in all remaining months.

B) Sales in December are higher than in every month from January through November.

C) Sales in December are higher than in November, and sales in November are higher than in every month from January through October.

D) Sales in each month from May through December are higher than in the remaining months.

GPT-5.4's Reasoning Process

The chart shows December as the strongest month overall, and across many categories there are also secondary peaks around May, August, and November. The aggregate row Total ex Autos, Fuel and Food reflects this clearly: Dec 137 > Nov 119 > May 116 > Aug 115, with those months above most others.

So A best captures the chart's overall more than one peak season pattern, whereas B and C focus too much on just the holiday ramp, and D is too broad.

Figure 36: Failure case 13.

Model response. The chart shows December as the strongest month overall, and across many categories there are also secondary peaks around May, August, and November. The aggregate row Total ex Autos, Fuel and Food reflects this clearly: Dec 137 > Nov 119 > May 116 > Aug 115, with those months above most others. So A best captures the chart's overall more than one peak season pattern, whereas B and C focus too much on just the holiday ramp, and D is too broad.

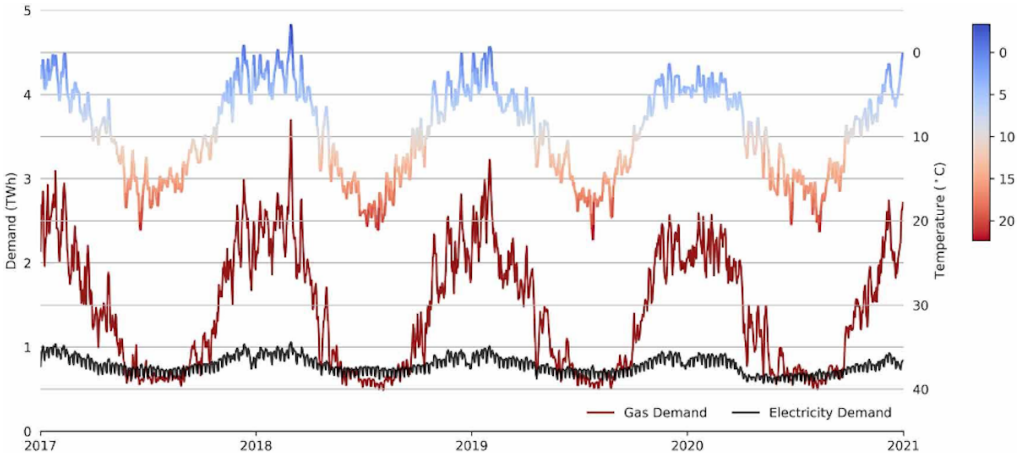
Case discussion. The reference captures the dominant seasonal regularity that December is the main peak, whereas GPT-5.4 further incorporates secondary peaks around May, August, and November

into the inferred rule. These additional details are supported only by more localized patterns and are not necessary for characterizing the dominant seasonal structure. This reflects *over-specificity*, where the model encodes secondary fluctuations into the inferred rule instead of abstracting the simpler chart-level regularity.

K.3.4 Example 14: Restricting an overall inverse association to annual cycles

Chart (Association)

Question. Which statement best characterizes the overall relationship between UK daily temperature and gas demand?



A) UK daily temperature and gas demand are negatively correlated from 2017 to 2018.

B) UK daily temperature and gas demand are negatively correlated from 2018 to 2020.

C) UK daily temperature and gas demand are negatively correlated overall from 2017 to 2021.

D) UK daily temperature and gas demand show periods of negative correlation within each annual cycle from 2017 through 2020.

GPT-5.4's Reasoning Process

1. The red gas-demand line peaks repeatedly in winter-like periods and falls in summer-like periods **across 2017, 2018, 2019, and 2020.** 2. The temperature-colored line shows the opposite seasonal movement within each year: when temperatures are lower, gas demand is higher; when temperatures are higher, gas demand is lower. 3. This inverse pattern is visible as recurring annual cycles rather than a single uniform straight-line relationship over the whole span. 4. Therefore the chart **best supports periods of negative correlation within each annual cycle from 2017 through 2020.**

Figure 37: Failure case 14.

Model response. The gas-demand line peaks repeatedly in winter-like periods and falls in summer-like periods across 2017–2020. The temperature-colored line shows the opposite seasonal movement within each year: lower temperatures coincide with higher gas demand, while higher temperatures coincide with lower gas demand. I

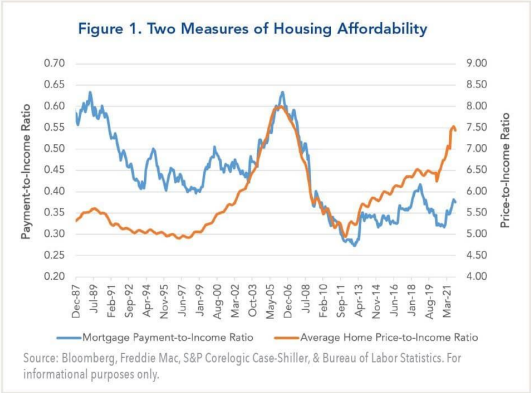
therefore treated the inverse relationship as recurring periods within each annual cycle. So D) periods of negative correlation within each annual cycle from 2017 through 2020 is the best fit.

Case discussion. Reference C gives an overall inverse association over 2017–2021, while GPT-5.4 selects recurring negative-association periods within 2017–2020. Both descriptions use the same association sign, but differ in time coverage and granularity.

K.3.5 Example 15: Fragmenting the overall housing-affordability association

Chart (Association)

Question. Which option best describes the relationship between the two housing-affordability ratios?



A) The two ratios show no clear Association from Dec 1987 to Oct 2003, but are positively correlated from Oct 2003 to May 2021.

B) The two ratios are positively correlated from Dec 1987 to Dec 2006, negatively correlated from Dec 2006 to Sep 2011, and positively correlated again from Sep 2011 to May 2021.

C) The mortgage-payment-to-income ratio and the average home-price-to-income ratio are broadly positively correlated overall.

D) The two ratios show no clear correlation from Dec 1987 to Oct 2003, are positively correlated from Oct 2003 to Sep 2011, and then show no clear correlation from Sep 2011 to May 2021.

GPT-5.4's Reasoning Process

D is the best choice. From **Dec 1987 to Oct 2003**, the two lines do not move together consistently: the mortgage-payment-to-income ratio generally trends down and fluctuates, while the home-price-to-income ratio first falls slightly and then rises, so there's no clear Association. **From Oct 2003 to Sep 2011**, they move much more similarly: both rise sharply into the mid-2000s peak and then fall steeply through 2011, indicating a positive Association. **From Sep 2011 to May 2021**, the home-price-to-income ratio rises steadily, but the mortgage-payment-to-income ratio is much flatter and more uneven, so again there is no clear Association.

Figure 38: Failure case 15.

Model response. D is the best choice. From Dec 1987 to Oct 2003, the two lines do not move together consistently: the mortgage-payment-to-income ratio generally trends down and fluctuates, while the home-price-to-income ratio first falls slightly and then rises, so there's no clear Association. From Oct 2003 to Sep 2011, they move much more similarly: both rise sharply into the mid-2000s peak and then fall steeply through 2011, indicating a positive Association. From Sep 2011 to May 2021, the home-price-to-income ratio rises steadily, but the mortgage-payment-to-income ratio is much flatter and more uneven, so again there is no clear Association.

Case discussion. Reference C describes broad positive association across the full span. GPT-5.4 selects a segmented account, with the clearest positive co-movement in the middle housing cycle. Global and segment-specific associations address different scopes, so choosing between these descriptions depends on the requested level of summary.

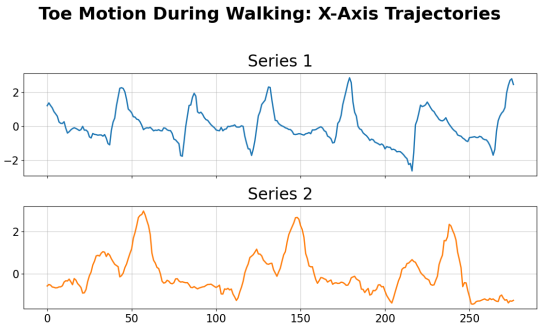
K.4 Incorrect Rule Identification

Incorrect rule identification occurs when a response applies a relationship or decision criterion that differs from the one supported by the chart. This category is distinct from visual misreading and from errors in the scope of an otherwise plausible rule.

K.4.1 Example 16: Inferring Association from Loosely Aligned Peaks

Chart (Association)

Question. What type of Association exists between the two series?



A) Positive

B) Negative

C) No clear relationship

D) Changes over time

GPT-5.4's Reasoning Process

The major peaks in the two trajectories occur at broadly **similar positions**, especially through the middle and later sections. I treated those repeated peak alignments as the dominant pattern and read the relationship as positive despite several local mismatches.

So A) Positive is the best fit.

Figure 39: Failure case 16.

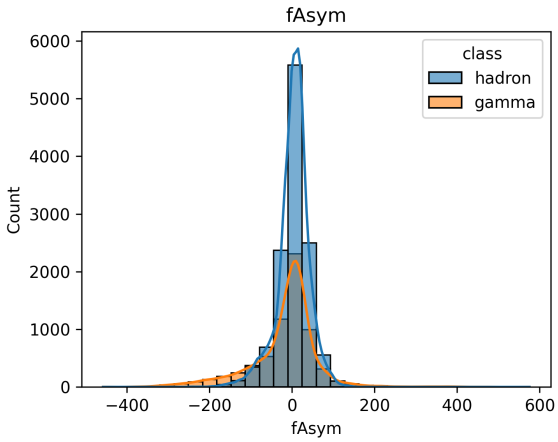
Model response. The major peaks in the two trajectories occur at broadly similar positions, especially through the middle and later sections. I treated those repeated peak alignments as the dominant pattern and read the relationship as positive despite several local mismatches. So A) Positive is the best fit.

Failure analysis. GPT-5.4 selects A (positive) because several prominent peaks occur at broadly similar horizontal positions. The reference answer is C (no clear relationship). Across the full trajectories, many rises, falls, and extrema do not align consistently, so a few approximate peak coincidences are insufficient to establish positive association. Using sparse landmark coincidence instead of consistent paired co-movement is *incorrect rule identification*.

K.4.2 Example 17: Using the central peak for a far-tail query

Chart (Distribution)

Question. If fAsym=-313.7, is the particle more likely gamma or hadron?



A) gamma | **B) hadron**

GPT-5.4's Reasoning Process

The displayed density panel has a much taller and narrower hadron peak around the center, while the gamma curve is lower and broader. Treating the queried value as belonging to that central concentration, **I assigned it to the class with the stronger peak.**

So B) hadron is the best fit.

Figure 40: Failure case 17.

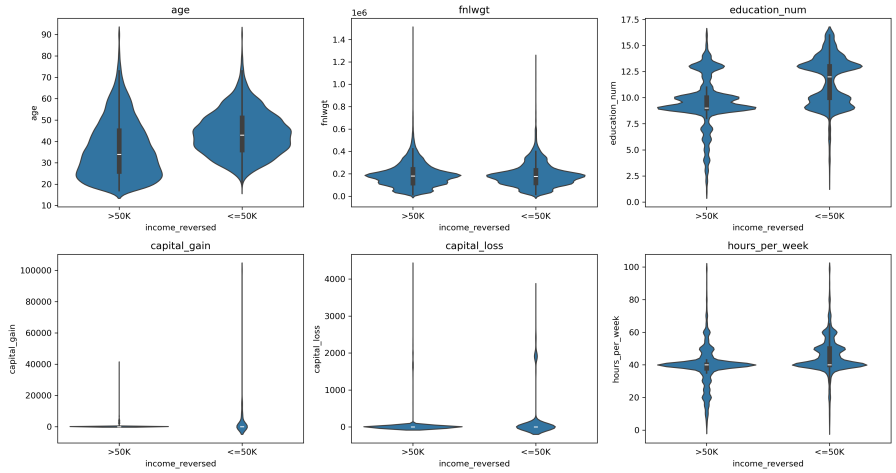
Model response. The displayed density panel has a much taller and narrower hadron peak around the center, while the gamma curve is lower and broader. Treating the queried value as belonging to that central concentration, I assigned it to the class with the stronger peak. So B) hadron is the best fit.

Failure analysis. The chart and question both concern `fAsym`, and the queried value is -313.7, far into the left tail. The reference answer is A (gamma), whereas GPT-5.4 selects B (hadron) from the taller hadron peak near the center. Classification should instead compare the two distributions at the queried x-location, where the broader gamma distribution supplies the relevant tail support. Applying global peak height rather than local density at the query value is *incorrect rule identification*.

K.4.3 EXAMPLE 18: REVERSING THE VISIBLE SPREAD COMPARISON

Chart (Distribution)

Question. which option best describes the relationship between capital_gain and income in terms of overall distribution spread?



A) >50K shows the larger spread on `capital_gain`

B) <=50K shows the larger spread on `capital_gain`

C) Their spreads are similar

D) No clear spread difference is visible

GPT-5.4's Reasoning Process

In the capital_gain subplot, the <=50K violin occupies a taller overall range and a visibly thicker central body. That makes the <=50K class look more broadly spread on capital_gain.

So B) <=50K shows the larger spread on `capital_gain` is the best fit.

Figure 41: Failure case 18.

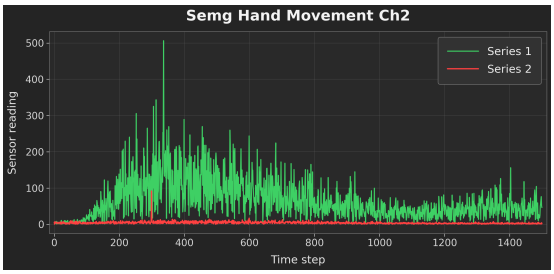
Model response. In the capital_gain subplot, the <=50K violin occupies a taller overall range and a visibly thicker central body. That makes the <=50K class look more broadly spread on capital_gain. So B) <=50K shows the larger spread on `capital_gain` is the best fit.

Failure analysis. The reference answer A identifies the $>$ 50K group as having the larger overall spread on `capital_gain`. In the relevant violin subplot, that distribution extends across a visibly larger vertical range. GPT-5.4 selects B by treating the central thickness of the $\leq$ 50K violin as evidence of greater spread. Because violin width represents local density rather than total range, using thickness as the comparison criterion is *incorrect rule identification*.

K.4.4 Example 19: Inferring association from a shared visual envelope

Chart (Association)

Question. What type of Association exists between the two series?



A) Positive

B) Negative

C) No clear relationship

D) Changes over time

GPT-5.4's Reasoning Process

Both series are most elevated through the broad middle portion and lower toward the beginning and end. I focused on that shared envelope rather than the very different amplitudes, so I interpreted the two series as moving together overall.

So A) Positive is the best fit.

Figure 42: Failure case 19.

Model response. Both series are most elevated through the broad middle portion and lower toward the beginning and end. I focused on that shared envelope rather than the very different amplitudes, so I interpreted the two series as moving together overall. So A) Positive is the best fit.

Failure analysis. GPT-5.4 selects A (positive), whereas the reference answer is C (no clear relationship). The blue series contains large, rapidly varying bursts, while the orange series remains much smaller and does not consistently track those changes. Treating a similar broad envelope as evidence of paired co-movement substitutes silhouette similarity for association, resulting in *incorrect rule identification*.

K.4.5 EXAMPLE 20: ASSIGNING ASSOCIATION SIGNS FROM SEPARATED TREND REGIMES

Chart (Association)

Question. Which option best describes the relationship between the 30-year fixed mortgage rate and the house-price-to-income ratio from 1971 to 2023?



A) The two series show no clear relationship because their curves do not rise and fall together.

B) The two series are positively correlated from 1971 to 1991 and negatively correlated from 1991 to 2023.

C) The two series are positively correlated from 1971 to 1991 but show no clear relationship from 1991 to 2023.

D) The 30-year fixed mortgage rate and the house-price-to-income ratio are positively correlated because their curves broadly rise and fall together.

GPT-5.4's Reasoning Process

From roughly 1971 to 1991, both lines broadly move together, first rising and then falling, which suggests a positive relationship in that span. **After about 1991**, the mortgage rate trends downward overall while the house-price-to-income ratio trends upward overall. That makes the **later relationship negative** rather than positive or absent.

So **B) The two series are positively correlated from 1971 to 1991 and negatively correlated from 1991 to 2023** is the best fit.

Figure 43: Failure case 20.

Model response. From roughly 1971 to 1991, both lines broadly move together, first rising and then falling, which suggests a positive relationship in that span. After about 1991, the mortgage rate trends downward overall while the house-price-to-income ratio trends upward overall. That makes the later relationship negative rather than positive or absent. So B) The two series are positively correlated from 1971 to 1991 and negatively correlated from 1991 to 2023 is the best fit.

Case discussion. GPT-5.4 selects positive association before 1991 and negative association afterward, whereas reference D specifies overall positive association. Segment-level signs and an overall sign need not agree. This case illustrates dependence on the selected interval rather than establishing that segmentation itself is an incorrect rule.

Table 10: Human-baseline sample sizes and weighting sensitivity.

Task	Regularity	Unique QAs	Responses	Response-weighted	QA-weighted
Rule Induction	Trend	19	38	65.8	65.8
Rule Induction	Periodicity	19	38	73.7	73.7
Rule Induction	Distribution	19	38	76.3	76.3
Rule Induction	Association	18	36	63.9	63.9
Rule Induction	All	75	150	70.0	70.0
Example Inference	Trend	19	38	86.8	86.8
Example Inference	Periodicity	19	38	73.7	73.7
Example Inference	Distribution	18	36	88.9	88.9
Example Inference	Association	19	38	60.5	60.5
Example Inference	All	75	150	77.3	77.3

L More Detailed Information for Human Evaluation

We recruited 10 participants for the human-performance evaluation. The participants were graduate students (aged 20–30) from universities in China, were native Chinese speakers, and had expertise in deep learning and visualization. None is an author of this paper.

The evaluation uses a stratified sample of 150 unique QAs, corresponding to 3.8% of the 3,929-QA benchmark, with 75 rule induction and 75 example inference items. Each QA is independently answered by two different participants, yielding 300 human QA responses in total. The assignment audit confirms that all 10 participants completed exactly 30 questions and that each of the 150 QAs has exactly two responses.

Human evaluation is substantially more time-consuming than ordinary answer-label verification because participants must inspect each chart, infer the latent regularity supported by the observations, and then answer the corresponding reasoning question without access to correctness feedback. Each 30-question assignment requires approximately 60 minutes to complete. Across 10 participants, the full evaluation therefore corresponds to approximately 600 participant-minutes, or 10 participant-hours, of human reasoning time. We use a stratified subset rather than the full benchmark to keep this annotation burden tractable while covering both task families and all four regularity categories.

L.1 Human-Study and Model-Prediction Alignment

Table 2 summarizes the full benchmark, while Table 10 reports the human study on 150 sampled QAs, with 75 RI and 75 EI items and two responses per item. Figure 44 further provides a fair comparison between human and model performance across RI, EI, and overall accuracy.

Participants use a web interface with a fixed chart, question, answer options, and answer format for each assigned item. The interface does not reveal correctness feedback or ground-truth answers. It records progress within the assigned subset and allows participants to export their responses after completing the evaluation. Figures 45– 49 illustrate the complete participant workflow: entering the assigned task, navigating the 30-question desktop interface, monitoring progress, exporting the completed responses, and reaching the final thank-you page.

M Data-Annotation Interface Examples

Figures 50–52 show data-annotation pages aligned with the representative trend, periodicity, and distribution examples in the main paper.

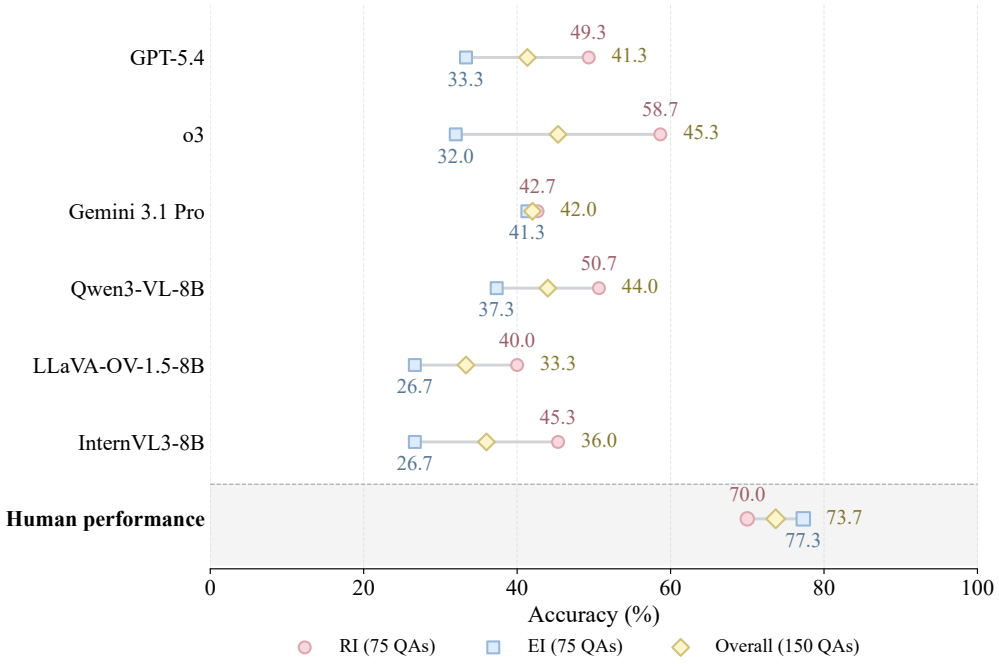


Figure 44: Comparison of human and model performance.

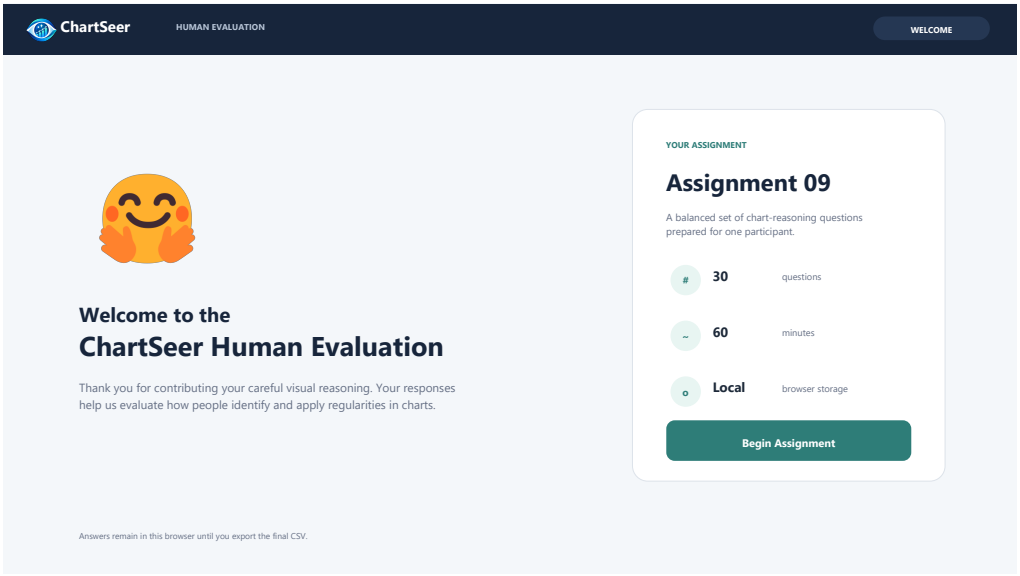


Figure 45: Welcome page of the CHARTSEER human-evaluation website. The desktop interface identifies the assigned 30-question set, states the expected completion time, and informs participants that responses remain in local browser storage until export.

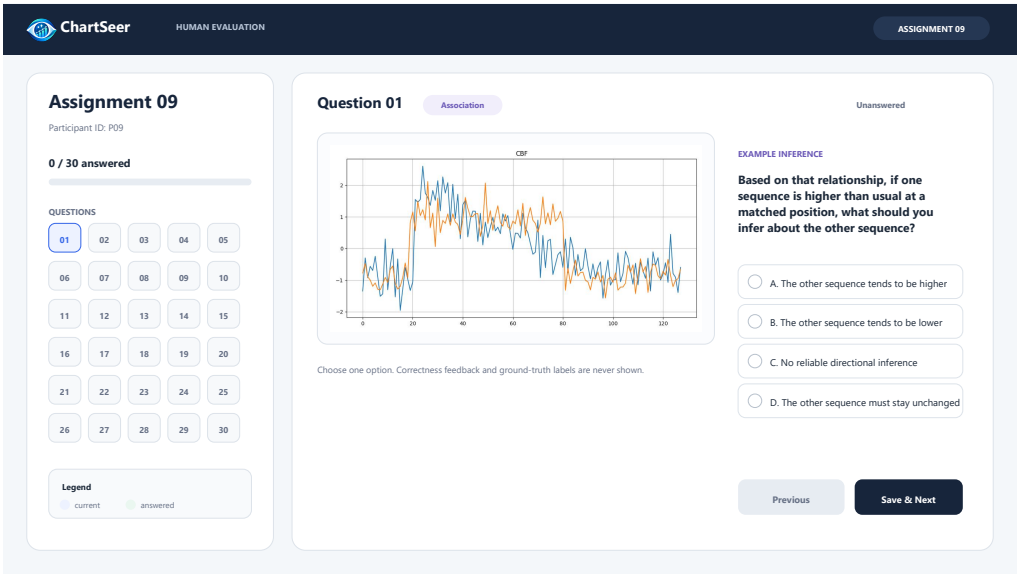


Figure 46: Desktop question-navigation view for Assignment 09. The compact numbered grid displays all 30 assigned questions at once, while the main panel presents the current chart, question, and answer choices. The interface reports completion state without exposing correctness feedback or ground-truth answers.

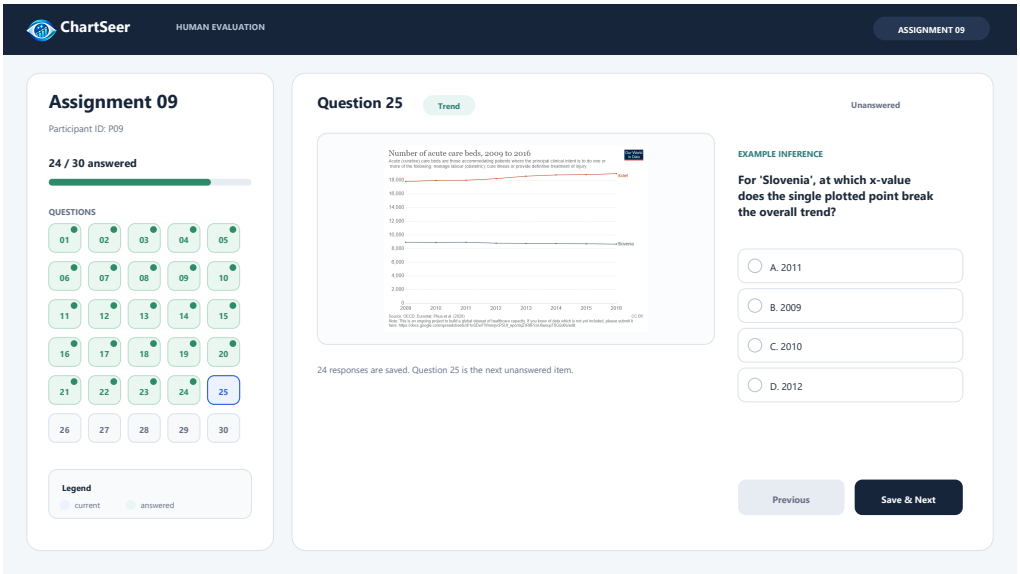


Figure 47: In-progress view reconstructed from participant 09's recorded responses after 24 of 30 questions. Completed items are marked in green, the progress bar reports the current completion state, and the next unanswered question is shown in the main panel.

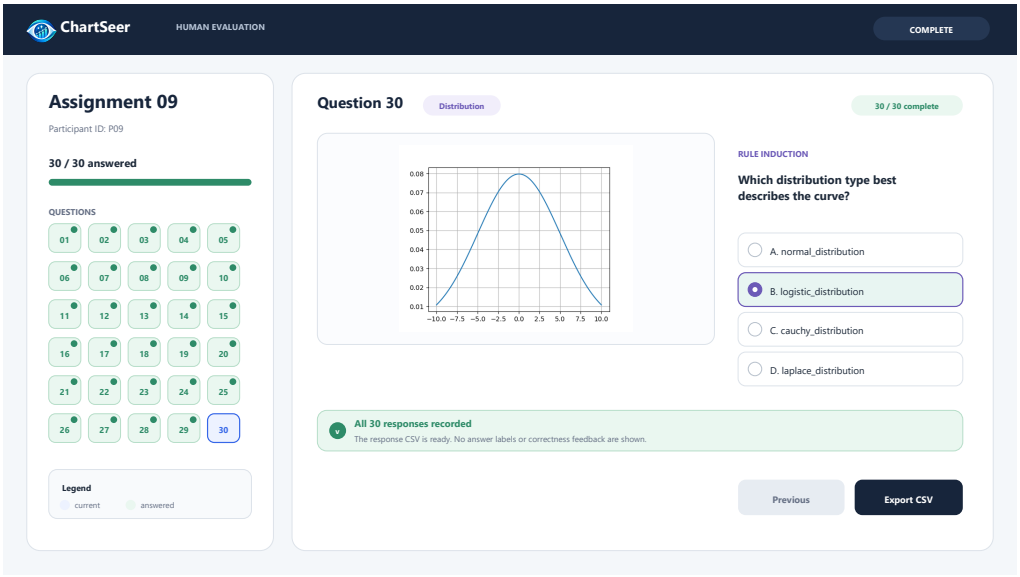


Figure 48: Completion and export view after all 30 responses have been recorded. At 30/30 completion, the CSV export control becomes active; the interface still reveals neither correctness feedback nor ground-truth answers.

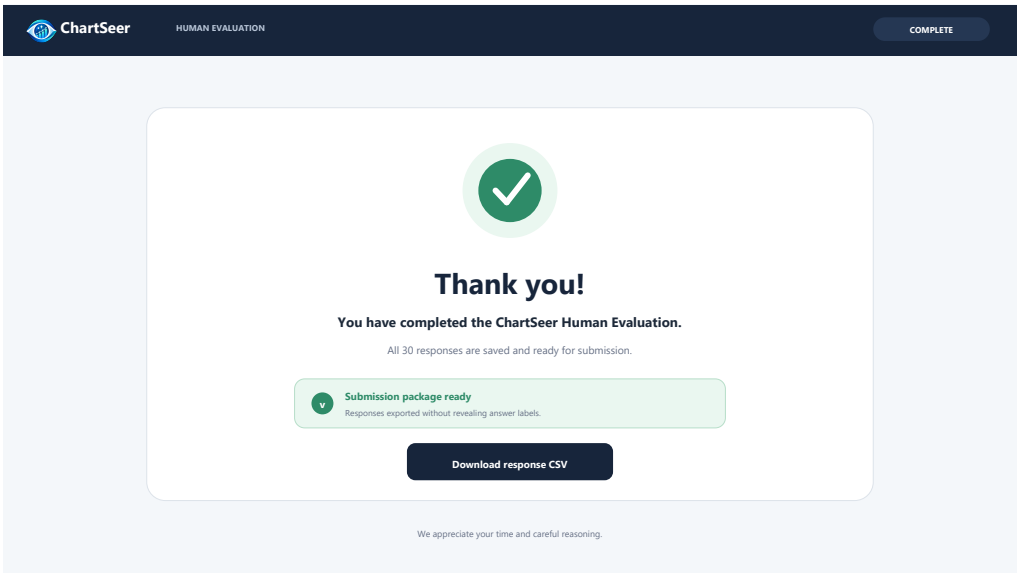


Figure 49: Completion page shown after all 30 responses have been recorded. The page thanks the participant, confirms that the submission package is ready, and provides the final CSV download control without revealing answer labels or correctness feedback.

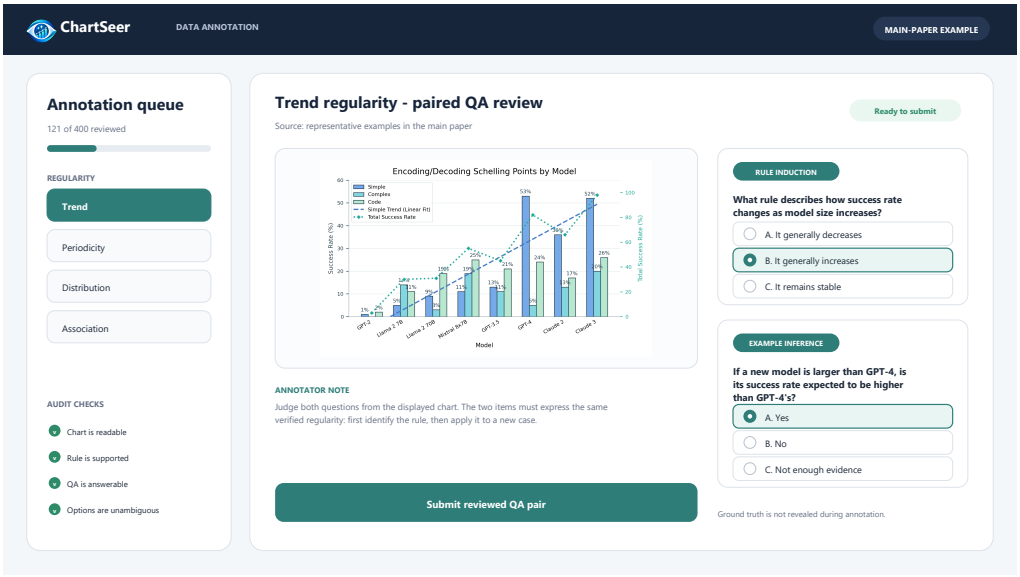


Figure 50: Data-annotation interface for the trend-regularity QA pair used in the main paper. The paired items verify the increasing success-rate regularity and its application to a hypothetical larger model.

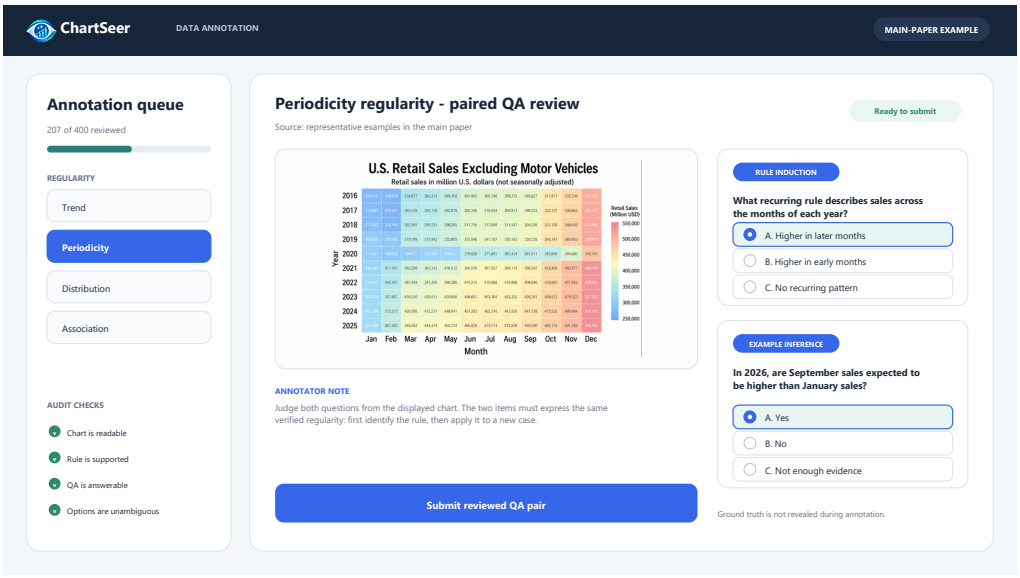


Figure 51: Data-annotation interface for the periodicity-regularity QA pair used in the main paper. The page uses the U.S. retail-sales heatmap and records both the recurring within-year pattern and its application to a new year.

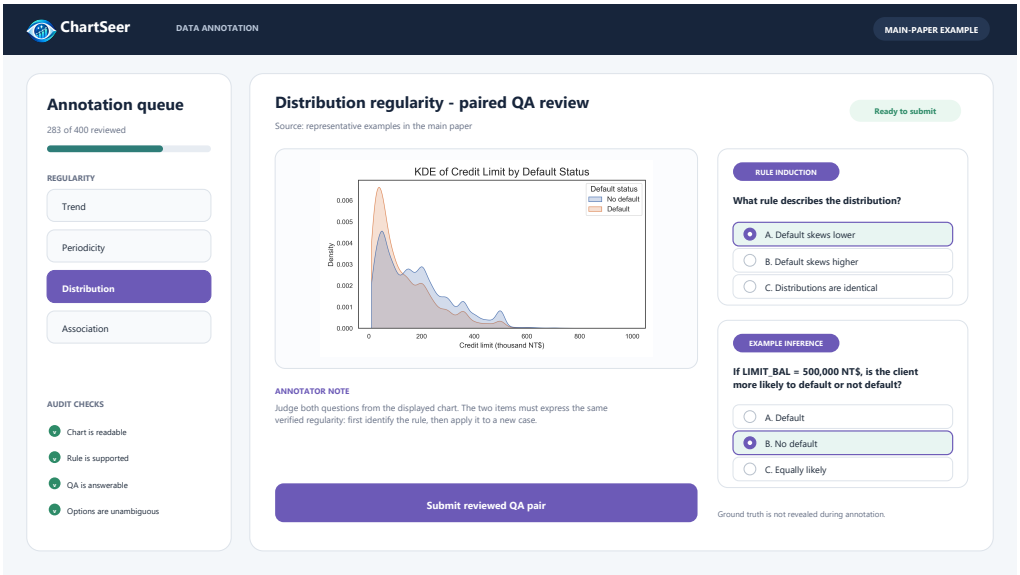


Figure 52: Data-annotation interface for the distribution-regularity QA pair used in the main paper. The paired items verify the concentration of defaulting customers at lower credit limits and apply that regularity to a new client.