Causal AI Scientist: Facilitating Causal Data Science with Large Language Models

Vishal Verma[1*] **Devansh Bhardwaj**[2*] **Sawal Acharya**[3*] **Samuel Simko**[4*]
Anahita Haghighat[5] **Mrinmaya Sachan**[4] **Dominik Janzing**[6†]
Bernhard Schölkopf[4, 7] **Zhijing Jin**[7,8,9]
[1]Carnegie Mellon University [2]IIT Roorkee [3]Stanford University [4]ETH Zürich
[5]Independent [6]Amazon, Tübingen [7]MPI for Intelligent Systems, Tübingen
[8]University of Toronto [9]Vector Institute
vishalv@andrew.cmu.edu zjin@cs.toronto.edu

Abstract

Causal inference aims to quantify the causal effect of one variable on another while accounting for confounders. Existing LLM-powered approaches to causal effect estimation have two main limitations: they require users to specify the methods and variables, and they support only a limited set of causal effect measures. To address these limitations, we present Causal AI Scientist (CAIS), an LLM-augmented causal tool with self-correction capabilities. CAIS interprets natural language queries and selects methods using a decision-tree-based reasoning framework. It then executes the selected method and uses a validation feedback loop to self-correct before answering the input question. By combining causal inference principles with the analytical capabilities of LLMs, CAIS provides interpretable, causality-driven answers to user queries. In experiments with queries drawn from causal inference textbooks, real-world empirical studies, and synthetic datasets, CAIS outperforms baselines in method selection and causal effect estimation.[1]

1 Introduction

Understanding cause-and-effect relationships is central to evidence-based decision-making and empirical research across disciplines such as social science (Imbens, 2024), public health (Glass et al., 2013), and biomedicine (Kleinberg & Hripcsak, 2011). Real-world causal effect estimation is challenging because we do not observe outcomes under both treatment and control for the same unit (Holland, 1986). Therefore, causal inference relies on assumptions that justify comparisons between treated and control groups.

Identifying suitable methods and justifying their applicability typically requires domain expertise. Researchers rely on their understanding of theory, identification strategies, and the data-generating process to select estimation techniques and assess result credibility. This reliance on expert knowledge can limit access to causal analysis for users who may benefit from it but lack methodological training. For example, a policy analyst with employment and wage data may wish to evaluate the impact of minimum wage laws but may not be able to draw reliable conclusions without the appropriate tools.

Recent advances in Large Language Models (LLMs) offer a promising pathway to automate the causal inference process (Kiciman et al., 2024). Existing works use LLMs to generate

[*] Equal contribution

[†] This work is conducted outside Amazon

[1] Code, datasets, and a demo are available at:

- https://github.com/causalNLP/causal-agent (code and datasets)
- https://huggingface.co/spaces/CausalNLP/causal-agent (online demo)

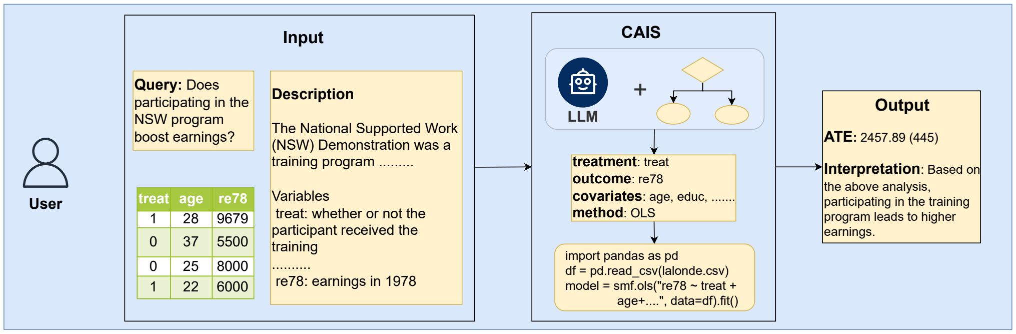
code for user-specified estimation tasks (Liu et al., 2024; Chen et al., 2025). However, users must choose the method and variables, and doing so requires familiarity with a wide range of techniques.

One approach to enabling end-to-end causal analysis is to fine-tune models specifically for causal inference, such as LLM4Causal (Jiang et al., 2024a). However, LLM4Causal supports only a limited set of effect measures, excluding many methods used in applied research. Another direction involves general-purpose data science agents powered by language models. Currently, these agents are mostly targeted toward and tested on machine learning and statistical analysis tasks (Hong et al., 2024; Guo et al., 2024). While there are LLM-powered causal agents, they primarily focus on causal discovery tasks rather than causal effect estimation (Wang et al., 2025; Han et al., 2024).

To address these limitations, we present Causal AI Scientist (CAIS), an end-to-end LLM-powered pipeline for generating causality-driven answers to natural language queries. Given a dataset, its description, and a query, CAIS frames a causal estimation problem, selects an appropriate method, estimates the effect, and interprets the result in the context of the original user query. Inspired by the Tree-of-Thoughts prompting approach (Yao et al., 2023a; Long, 2023), CAIS uses a structured decision tree to break down the method selection process into smaller, focused steps. Each node in the tree prompts the model to evaluate a specific feature of the dataset or query, such as identifying the treatment, outcome, or instrument. This step-by-step approach simplifies the reasoning process and makes it easier to follow. Additionally, CAIS performs diagnostic checks and incorporates a feedback loop to correct potential errors before producing a final answer.

We also create a dataset of natural language causal queries to test our tool. Existing benchmarks for causal inference tasks focus on implementing specified estimation procedures (Liu et al., 2024). Meanwhile, our dataset enables evaluation across the entire causal estimation workflow, including the formulation of the causal estimation problem with the correct treatment and outcome variables, method and model-specific variable selection, and effect estimation. Moreover, our dataset consists of queries drawn from textbook examples, real-world cases, and synthetic datasets with known causal effects, allowing us to cover a wide range of estimation methods and scenarios.

Our experiments on three dataset collections, real-world studies, textbook examples, and synthetic datasets, show that CAIS outperforms baseline methods in selecting the appropriate causal inference method. CAIS also surpasses most baseline methods in accurately estimating causal effect values.



Query: Does participating in the NSW program boost earnings?

treat	age	re78
1	28	9679
0	37	5500
0	25	8000
1	22	6000

Description

The National Supported Work (NSW) Demonstration was a training program ....

Variables
treat: whether or not the participant received the training
....
re78: earnings in 1978

treatment: treat
outcome: re78
covariates: age, educ, ...
method: OLS

```
import pandas as pd
df = pd.read_csv(lalonde.csv)
model = smf.ols("re78 ~ treat +
          age+....", data=df).fit()
```

ATE: 2457.89 (445)

Interpretation: Based on the above analysis, participating in the training program leads to higher earnings.

Figure 1: **CAIS workflow.** The user provides an input dataset (CSV file), its description, and an associated causal query. Guided by a decision tree and a backbone LLM, CAIS selects an appropriate estimation method, executes the code, and returns the estimated causal effect along with a natural language interpretation.

2 Problem Statement

We begin with a motivating example based on the seminal Lalonde dataset (LaLonde, 1986). Suppose a user has a dataset from a job-skill training program containing details such as employees' participation status, their earnings after the program, and background variables like education and work experience. The user may be interested in the question: **Does participating in a training program boost earnings?**

Answering this question requires estimating a measure of causal effect, such as the Average Treatment Effect (ATE).[2] ATE is the expected difference in earnings if everyone participated versus if no one had participated. If the assignment is random, ATE is estimated by directly comparing earnings between employees who participated and those who did not. However, in observational settings, we need to adjust for confounding variables (such as education) that influence both program participation and subsequent earnings.

To formalize this scenario, we consider settings with the following inputs:

- A tabular dataset $\mathcal{D}$, where each row is a unit of analysis (a person or entity) and each column is a variable associated with the unit.
- Metadata describing the columns and the data collection process.
- A natural language query q associated with the dataset.

The goal is to generate a causal answer to the query q. This involves interpreting the query and dataset information to identify the treatment and outcome variables, selecting an appropriate causal effect measure, and computing it using a suitable statistical method.

3 Methodology: CAIS

CAIS consists of four successive methodological stages, each comprising one or more micro-tools. Each stage combines logic grounded in established causal inference principles with Large Language Model (LLM) reasoning, which is selectively applied to sub-tasks that typically require expert-like judgment.

In Stage 1, we analyze the dataset and identify key components, such as treatment and outcome variables, valid instruments, running variables, etc. In Stage 2, a rule-based decision tree uses this information to select a valid method for causal effect estimation. Stage 3 checks the standard assumptions for the chosen method; if any check fails, the system backtracks through the decision tree to select an alternative method, forming a validation loop between Stages 2 and 3. In Stage 4, once all checks pass, the selected method is executed using predefined templates, and the final result is returned with an explanation.

3.1 Stage 1: Dataset Preprocessing and Query Decomposition

CAIS initiates the pipeline with a thorough examination of the dataset, inspecting column names and data types, and computing descriptive statistics. Beyond basic profiling, it investigates potential relationships within the data, such as correlations, and attempts to identify features relevant to causal inference. CAIS then proceeds to conceptualize the user's query and actual data by prompting the LLM to determine which columns correspond to the treatment, outcome, and pre-treatment variables. Additionally, it uses the LLM to identify the presence of instrumental variables, running variables that govern treatment assignment, observed confounders, and time-related variables that indicate the timing of observations.

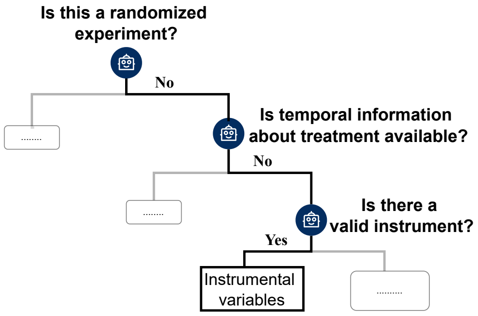

Is this a randomized experiment?
No
....
Is temporal information about treatment available?
No
....
Is there a valid instrument?
Yes
Instrumental variables
....

Figure 2: **Demonstration of CAIS's method selection process**. At each decision node, CAIS prompts an LLM to assess dataset characteristics. Following this example path: the LLM identifies that the data does not come from a randomized trial. It deduces the absence of temporal treatment information (for Difference-in-Differences) and identifies the presence of an instrument, resulting in the selection of an instrumental-variables (IV) approach.

3.2 Stage 2: Method Selection

In this stage, CAIS identifies a suitable estimation method through a structured decision tree (see Appendix Figure 6). Inspired by the Tree-of-Thoughts framework (Yao et al., 2023a; Long, 2023), this tree decomposes method selection into sequential steps, where each node evaluates a specific property of the dataset (e.g., timing of treatment, presence of instruments). Each node is associated with a detailed prompt that specifies the required characteristics of a valid method or variable. The hierarchical structure of the tree enhances interpretability and leverages the LLMs' strengths in performing well on specific, narrowed-down tasks. If multiple branches are viable, CAIS relies on the LLM to rank them by inspecting Stage 1 results. Figure 2 provides a visual illustration of the method selection process.

Compared to machine learning models, causal estimation does not involve hyperparameters to tune, and only a few methods are appropriate for any given task. However, incorrect choices can lead to invalid results. Our goal is to ensure interpretability and accuracy, and the decision tree-based approach supports both objectives. Nonetheless, errors in one step can propagate to subsequent steps. For instance, the model may incorrectly assume an RCT setting for observational data and produce incorrect findings. We implement safeguards to mitigate such cases.

3.3 Stage 3: Validation

After selecting the appropriate method, we first perform statistical checks to assess its reliability, such as the parallel trends test for Difference-in-Differences or the F-statistic for IV. Next, we prompt an LLM to reflect on both the test outcome and the numerical value to evaluate whether the result is sensible. If the LLM perceives the results to be unreliable, we return to the method and variable selection phase. In the next iteration, we incorporate information about validation test results from the previous run. Similarly, if more than one method is identified earlier, we try an alternative one. The feedback mechanism serves as a safeguard against potential errors that may arise in the selection and dataset analysis phases. We provide a detailed example of the method validator in Appendix D.

[2] While we focus here on ATE for illustration, other causal quantities, such as the effect on treated individuals (ATT) or compliers (LATE), may be more appropriate depending on the setting. Estimating these requires different assumptions and techniques. We refer readers to standard causal inference texts for a broader treatment (Imbens & Rubin, 2015; Hernan & Robins, 2025; Cunningham, 2021).

3.4 Stage 4: Method Execution and Interpretation

If all checks corresponding to the selected method pass, we move to the method execution phase. For most selected methods, we rely on predefined code templates with placeholders for key variables determined in Stage 1. This approach differs from previous work that uses LLMs to generate code from scratch (Liu et al., 2024). While LLM-powered code generation can be flexible, implementation errors are common (Chen et al., 2025). One workaround is to use a loop with try/except blocks to refine the code repeatedly until successful execution. However, this requires multiple API calls, making it both time-consuming and expensive. In contrast, using predefined templates with variable placeholders minimizes implementation errors, as the only requirement is to identify and substitute the relevant variables correctly.

Finally, we prompt an LLM to interpret the results of the estimation step in the context of the original query. For example, the model may assess whether the estimated causal effect is strong, weak, or statistically significant. Alongside this interpretation, we include important caveats, such as validation results and notes on assumptions or limitations that could affect the reliability of the estimate.

4 Experimental Setup

4.1 Dataset

Collection	Origin	# Queries	# CSV Files	Median Observations	Median Columns
QRData	Textbook examples	39	35	1209	19
Real-world studies	Research papers	29	14	1720	17.5
Synthetic	Simulated scenarios	45	45	428	7

Table 1: Summary of the dataset collections we use to evaluate CAIS

To comprehensively evaluate CAIS, we use three types of datasets: textbook data from QRData (Liu et al., 2024), real-world studies, and synthetic data.

QRData is a benchmark designed to evaluate the statistical and causal reasoning capabilities of LLMs. The estimation tasks are adapted from causal inference textbooks. Since the datasets are constructed for teaching purposes, they are preprocessed, and the inference process is relatively streamlined.

Real-world studies involve more complex designs, a broader range of variables, and less structured datasets. To evaluate CAIS in these settings, we curate examples directly from published empirical research papers.

Both QRData and real-world examples primarily rely on linear regression, limiting coverage of other estimation methods. Moreover, the true causal effects in real-world settings are not known; only reference values computed by experts are available. To enable evaluation across a wider range of methods with known ground truth values, we construct **synthetic datasets**.

We provide details on the dataset creation procedure in Appendix A.

4.2 Baseline Approaches

Given the lack of LLM-based tools specifically designed for fully automated causal inference, we compare our method against three baseline prompting strategies.

ReAct Prompting Following Liu et al. (Liu et al., 2024), our ReAct-based approach (Yao et al., 2023b) uses iterative reasoning–action cycles. The LLM alternates between causal analysis and program execution. In this process, the LLM examines the dataset and causal question, then identifies variables, explores data characteristics, selects appropriate methods, and computes causal effects through multiple thought–action–observation steps before providing the final results.

Liu et al. (2024) benchmark several prompting strategies for causal inference and find that ReAct achieves the best overall performance.

Program of Thought Prompting Unlike ReAct prompting, which alternates between tool calls and reasoning, Program-of-Thought (PoT)-based approach (Chen et al., 2022) prompts the LLM to generate a complete Python program to perform the causal analysis. The LLM examines the dataset characteristics and the causal question, and outputs code to run the appropriate causal inference method, statistical tests, and result interpretation before storing the relevant outputs. A sample prompt is provided in Appendix Figure 7.

Veridical Data Science Prompting Typical prompting strategies for data science are relatively straightforward and involve limited self-reflection. To address this limitation, we design a prompting structure based on the Veridical Data Science framework (Yu, 2020). After each decision, the LLM is prompted to reflect on its response and reconsider its output, with the goal of improving stability in the reasoning process. A sample veridical science-based prompt is provided in Appendix Figure 9.

4.3 Implementation Details

All models are implemented in Python. For estimating causal effects, we use the `DoWhy` (Sharma & Kiciman, 2020; Blöbaum et al., 2024) and `statsmodels` (Seabold & Perktold, 2010) libraries. The backbone LLMs include GPT-4o, GPT-4o-mini, and o3-mini (OpenAI et al., 2024), llama-3.3-70B-instruct (Grattafiori et al., 2024), and Gemini 2.5 Pro (Team, 2024). All models are accessed through API calls. The temperature parameter is set to 0 for reproducibility.

4.4 Evaluation Metrics

We evaluate our pipeline using the following metrics.

- **Method Selection Accuracy (MSA)**: Percentage of queries where the selected method $\hat{m}_i$ matches the reference method (m_i)

$$\text{MSA} = \frac{1}{N}\sum_{i=1}^{N} \mathbf{1}[\hat{m}_i = m_i] \times 100 \tag{1}$$

- **Mean Relative Error (MRE)**: Average relative error between predicted causal effects $(\hat{\tau}_i)$ and reference values (τ_i):

$$\text{MRE} = \frac{1}{N}\sum_{i=1}^{N} \min\left(\frac{|\hat{\tau}_i - \tau_i|}{|\tau_i|}, 1\right) \times 100\% \tag{2}$$

  To reduce sensitivity to outliers, relative error is capped at 100% per query.

N denotes the total number of queries in the evaluation set.

5 Preliminary Results

In this section, we present and analyze the performance of CAIS across all evaluation datasets. We compare it against baseline prompting strategies in terms of method selection accuracy (MSA) and mean relative error (MRE). Finally, we provide both qualitative and quantitative error analysis of CAIS.

	MSA (↑)				**MRE (↓)**			
LLM	**ReACT**	**PoT**	**Veridical**	**CAIS**	**ReACT**	**PoT**	**Veridical**	**CAIS**
				Textbook Data				
GPT-4o	55	41	60.5	**74.4**	43.2	32.6	40.7	**31.6**
GPT-4o-mini	55.2	54.3	41	**74.3**	33.9	**33.6**	42.2	55.9
o3-mini	21.8	30.7	61.5	**94.1**	44.7	30.7	**27.6**	43.1
Gemini 2.5 Pro	62.2	50	59	**81.2**	43.2	**35.8**	37.8	41.2
Llama 3.3 70B	34.4	53.8	46.1	**81.8**	43.9	**31.5**	55.4	54.19
				Synthetic Data				
GPT-4o	51.2	53.3	**79**	76.9	27.9	19.9	27.7	**17.4**
GPT-4o-mini	41.86	37.7	43.4	**75.9**	21.2	37.7	25.7	**16.2**
o3-mini	46.7	42.2	66.6	**73.3**	21	42.2	**20.2**	20
Gemini 2.5 Pro	48.2	53.2	58.5	**75.6**	20.2	24	26.5	**18.5**
Llama 3.3 70B	55.8	47.6	50	**79.5**	21.3	**21.1**	33.3	50
				Real Data				
GPT-4o	69.5	57.7	48	69.2	**43.1**	54.7	53.6	47.5
GPT-4o-mini	51.8	54.6	28	**65.2**	**52.3**	55.6	54.4	54.55
o3-mini	57.1	33.3	59.2	**76.9**	43.2	46.3	41.2	**39.7**
Gemini 2.5 Pro	55	42.2	53.2	**78.3**	38.1	42	39	**32**
Llama 3.3 70B	44.4	53.8	24	**73**	52.6	53.8	52.8	**37.4**

Table 2: Performance of CAIS and baseline prompting strategies across all datasets and LLMs. Results are reported for both **Method Selection Accuracy (MSA, higher is better)** and **Mean Relative Error (MRE, lower is better)**. Dataset blocks correspond to Textbook, Synthetic, and Real-world settings. Bold entries indicate the best value in each row, underlined entries indicate the second best.

5.1 Overall Performance

Table 2 presents a unified view of method selection accuracy (MSA) and mean relative error (MRE) for all models across the three dataset collections. The results demonstrate that CAIS achieves consistently strong method selection performance while maintaining competitive estimation accuracy.

CAIS shows superior method selection capabilities. Across all three datasets and models, CAIS consistently outperforms the baselines in Method Selection Accuracy (MSA), with significant margins in almost all cases. For example, on the textbook dataset, o3-mini achieves an MSA of 94.1%, which is 32.6% higher than the second-best baseline. On average, CAIS improves MSA over the best-performing baseline per LLM by 22.18 points on the Textbook dataset, 15.58 points on the Synthetic dataset, and 14.10 points on the Real dataset. These results demonstrate the effectiveness of our decision-tree-based method selection.

CAIS achieves competitive estimation accuracy. Performance trends in Mean Relative Error (MRE) are more nuanced, with no consistent pattern across models or baselines. For the Synthetic and Real datasets, CAIS performs as well as or better than the baselines, achieving the lowest or second-lowest MRE for most backbone LLMs. An exception is the textbook-based dataset, where CAIS consistently underperforms. One characteristic of CAIS is its reliance on pre-existing code templates, which means it does not include a retry mechanism for code execution. If execution fails, this can result in a substantial increase in MRE. In contrast, the baselines allow multiple retries, ensuring that they return some causal effect estimate, even if the selected method is incorrect.

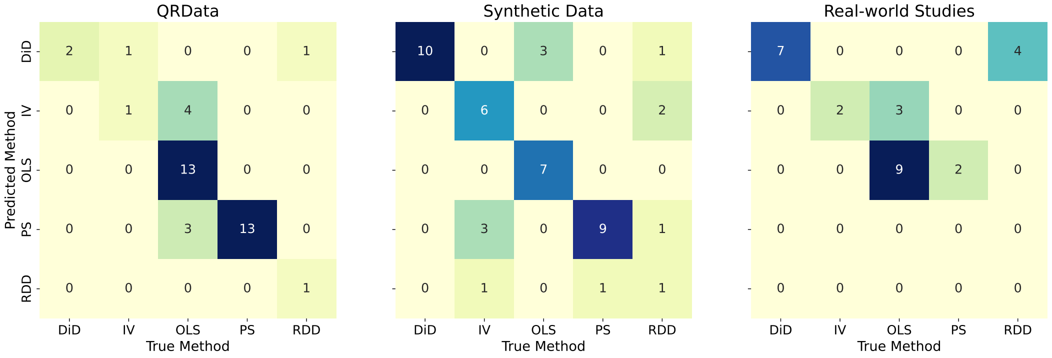


Figure 3: Confusion matrix showing method selection performance of CAIS (GPT-4o).

5.2 Error Analysis

We conduct a systematic qualitative analysis of the errors made by CAIS. The goal is to uncover error patterns, understand their root causes, and identify which stages of the pipeline are most susceptible to failure.

- **Incorrect Variable Selection:** LLMs frequently misinterpret temporal covariates, such as birth year or quarter indicators, as observation time points. This misinterpretation can lead to the selection of Difference-in-Differences as the causal inference method. Additionally, LLMs often misidentify treatment and outcome variables, particularly when column names lack clear descriptive labels or contain ambiguous terminology.
- **Wrong Method Selection:** As demonstrated in Figure 3, LLMs misclassify randomized controlled trials as encouragement designs leading to the selection of Instrumental Variables instead of linear regression. Similarly, for synthetic datasets, the model fails to identify instrumental variables as the optimal method in three instances. This pattern underscores the inherent challenge of selecting valid instruments based solely on data descriptions.
- **Incorrect Data Formats:** Implementation errors also stem from inconsistent data formatting. Specifically, certain variables are encoded as strings when causal inference packages like DoWhy require numerical inputs, creating compatibility issues that compromise execution.

6 Conclusion

In this work, we introduce a pipeline for end-to-end causal inference, CAIS, and evaluate its effectiveness across three diverse dataset collections. CAIS consistently outperforms baseline prompting strategies on method selection while achieving high performance in causal effect estimation. These strong results underscore the value of CAIS 's structured decision-tree-based approach, which decomposes complex reasoning into interpretable steps. This approach not only improves estimation accuracy but also enhances robustness and transparency, which are important for disciplines like public health and policy design. The framework's high performance on well-structured datasets, such as the synthetic dataset, suggests that real-world results can be further improved with better data preprocessing.

As ongoing work, we are improving data preprocessing functionalities. Similarly, we are considering additional metrics for a more comprehensive evaluation, such as how accurately the model selects treatment, outcome, and control covariates, or the choice of model-specific variables, such as instruments. On the dataset end, we are compiling more real-world studies while also scaling up the size of the synthetic dataset.

Acknowledgements

We would like to thank Patrick Blöbaum, Daniela Schkoda, Philipp Faller, Sergio Hernan Garrido Mejia, and Yongjin Yang for helpful comments and suggestions. Likewise, we thank Vivian Yvonne Nastl for pointing to empirical papers using Pearl's approaches for estimating causal effects.

This material is based in part upon work supported by the German Federal Ministry of Education and Research (BMBF): Tübingen AI Center, FKZ: 01IS18039B; by the Machine Learning Cluster of Excellence, EXC number 2064/1 – Project number 390727645; by Schmidt Sciences SAFE-AI Grant; by NSERC Discovery Grant RGPIN-2025-06491; by Cooperative AI Foundation; by the Survival and Flourishing Fund; by a Swiss National Science Foundation award (#201009) and a Responsible AI grant by the Haslerstiftung. The usage of OpenAI credits is largely supported by the Tübingen AI Center.

References

Patrick Blöbaum, Peter Götz, Kailash Budhathoki, Atalanti A. Mastakouri, and Dominik Janzing. Dowhy-gcm: An extension of dowhy for causal inference in graphical causal models. *Journal of Machine Learning Research*, 25(147):1–7, 2024. URL http://jmlr.org/papers/v25/22-1258.html.

Qiang Chen, Tianyang Han, Jin Li, Ye Luo, Yuxiao Wu, Xiaowei Zhang, and Tuo Zhou. Can ai master econometrics? evidence from econometrics ai agent on expert-level tasks, 2025. URL https://arxiv.org/abs/2506.00856.

Wenhu Chen, Xueguang Ma, Xinyi Wang, and William W Cohen. Program of thoughts prompting: Disentangling computation from reasoning for numerical reasoning tasks. *arXiv preprint arXiv:2211.12588*, 2022.

Liying Cheng, Xingxuan Li, and Lidong Bing. Is GPT-4 a good data analyst? In *The 2023 Conference on Empirical Methods in Natural Language Processing*, 2023. URL https://openreview.net/forum?id=PxEhoPiBB0.

Scott Cunningham. *Causal Inference: The Mixtape*. Yale University Press, 2021. ISBN 9780300251685. URL http://www.jstor.org/stable/j.ctv1c29t27.

Nikita Dhawan, Leonardo Cotta, Karen Ullrich, Rahul Krishnan, and Chris J. Maddison. End-to-end causal effect estimation from unstructured natural language data. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. URL https://openreview.net/forum?id=gzQARCgIsI.

Thomas A. Glass, Steven N. Goodman, Miguel A. Hernán, and Jonathan M. Samet. Causal inference in public health. *Annual review of public health*, 34:61–75, March 2013. ISSN 0163-7525. doi: 10.1146/annurev-publhealth-031811-124606.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

Ken Gu, Ruoxi Shang, Ruien Jiang, Keying Kuang, Richard-John Lin, Donghe Lyu, Yue Mao, Youran Pan, Teng Wu, Jiaqian Yu, Yikun Zhang, Tianmai M. Zhang, Lanyi Zhu, Mike A. Merrill, Jeffrey Heer, and Tim Althoff. Blade: Benchmarking language model agents for data-driven science, 2024. URL https://arxiv.org/abs/2408.09667.

Siyuan Guo, Cheng Deng, Ying Wen, Hechang Chen, Yi Chang, and Jun Wang. Ds-agent: Automated data science by empowering large language models with case-based reasoning, 2024. URL https://arxiv.org/abs/2402.17453.

Kairong Han, Kun Kuang, Ziyu Zhao, Junjian Ye, and Fei Wu. Causal agent based on large language model, 2024. URL https://arxiv.org/abs/2408.06849.

M.A. Hernan and J.M. Robins. *Causal Inference: What If*. Chapman & Hall/CRC Monographs on Statistics & Applied Probab. CRC Press, 2025. ISBN 9781420076165. URL https://books.google.com/books?id=_KnHIAAACAAJ.

Paul W. Holland. Statistics and causal inference. *Journal of the American Statistical Association*, 81(396):945–960, 1986. doi: 10.1080/01621459.1986.10478354. URL https://www.tandfonline.com/doi/abs/10.1080/01621459.1986.10478354.

Sirui Hong, Yizhang Lin, Bang Liu, Bangbang Liu, Binhao Wu, Ceyao Zhang, Chenxing Wei, Danyang Li, Jiaqi Chen, Jiayi Zhang, Jinlin Wang, Li Zhang, Lingyao Zhang, Min Yang, Mingchen Zhuge, Taicheng Guo, Tuo Zhou, Wei Tao, Xiangru Tang, Xiangtao Lu, Xiawu Zheng, Xinbing Liang, Yaying Fei, Yuheng Cheng, Zhibin Gou, Zongze Xu, and Chenglin Wu. Data interpreter: An llm agent for data science, 2024. URL https://arxiv.org/abs/2402.18679.

Xueyu Hu, Ziyu Zhao, Shuang Wei, Ziwei Chai, Guoyin Wang, Xuwu Wang, Jing Su, Jingjing Xu, Ming Zhu, Yao Cheng, Jianbo Yuan, Kun Kuang, Yang Yang, Hongxia Yang, and Fei Wu. Infiagent-dabench: Evaluating agents on data analysis tasks. *ArXiv*, abs/2401.05507, 2024. URL https://api.semanticscholar.org/CorpusID:266933185.

Junjie Huang, Chenglong Wang, Jipeng Zhang, Cong Yan, Haotian Cui, Jeevana Priya Inala, Colin Clement, and Nan Duan. Execution-based evaluation for data science code generation models. In Eduard Dragut, Yunyao Li, Lucian Popa, Slobodan Vucetic, and Shashank Srivastava (eds.), *Proceedings of the Fourth Workshop on Data Science with Human-in-the-Loop (Language Advances)*, pp. 28–36, Abu Dhabi, United Arab Emirates (Hybrid), December 2022. Association for Computational Linguistics. URL https://aclanthology.org/2022.dash-1.5/.

Qian Huang, Jian Vora, Percy Liang, and Jure Leskovec. Mlagentbench: Evaluating language agents on machine learning experimentation, 2024. URL https://arxiv.org/abs/2310.03302.

Kosuke Imai and Kentaro Nakamura. Causal representation learning with generative artificial intelligence: Application to texts as treatments, 2024. URL https://arxiv.org/abs/2410.00903.

Guido W. Imbens. Causal inference in the social sciences. *Annual Review of Statistics and Its Application*, qq:1123–152, 2024.

Guido W. Imbens and Donald B. Rubin. *Causal Inference for Statistics, Social, and Biomedical Sciences: An Introduction*. Cambridge University Press, 2015.

Jacqueline A Jansen, Artür Manukyan, Nour Al Khoury, and Altuna Akalin. Leveraging large language models for data analysis automation. *bioRxiv*, 2023. doi: 10.1101/2023.12.11.571140. URL https://www.biorxiv.org/content/early/2023/12/21/2023.12.11.571140.

Haitao Jiang, Lin Ge, Yuhe Gao, Jianian Wang, and Rui Song. Llm4causal: Democratized causal tools for everyone via large language model, 2024a. URL https://arxiv.org/abs/2312.17122.

Haitao Jiang, Lin Ge, Yuhe Gao, Jianian Wang, and Rui Song. LLM4causal: Democratized causal tools for everyone via large language model. In *First Conference on Language Modeling*, 2024b. URL https://openreview.net/forum?id=H1Edd5d2JP.

Emre Kiciman, Robert Ness, Amit Sharma, and Chenhao Tan. Causal reasoning and large language models: Opening a new frontier for causality. *Transactions on Machine Learning Research*, 2024. ISSN 2835-8856. URL https://openreview.net/forum?id=mqoxLkX210. Featured Certification.

Samantha Kleinberg and George Hripcsak. Methodological review: A review of causal inference for biomedical informatics. *J. of Biomedical Informatics*, 44(6):1102–1112, December 2011. ISSN 1532-0464. doi: 10.1016/j.jbi.2011.07.001. URL https://doi.org/10.1016/j.jbi.2011.07.001.

Yuhang Lai, Chengxi Li, Yiming Wang, Tianyi Zhang, Ruiqi Zhong, Luke Zettlemoyer, Wen-Tau Yih, Daniel Fried, Sida Wang, and Tao Yu. DS-1000: A natural and reliable benchmark for data science code generation. In Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett (eds.), *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 18319–18345. PMLR, 23–29 Jul 2023. URL https://proceedings.mlr.press/v202/lai23b.html.

Robert J LaLonde. Evaluating the Econometric Evaluations of Training Programs with Experimental Data. *American Economic Review*, 76(4):604–620, September 1986. URL https://ideas.repec.org/a/aea/aecrev/v76y1986i4p604-20.html.

Victoria Lin, Louis-Philippe Morency, and Eli Ben-Michael. Text-transport: Toward learning causal effects of natural language, 2023. URL https://arxiv.org/abs/2310.20697.

Xiao Liu, Zirui Wu, Xueqing Wu, Pan Lu, Kai-Wei Chang, and Yansong Feng. Are LLMs capable of data-based statistical and causal reasoning? benchmarking advanced quantitative reasoning with data. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar (eds.), *Findings of the Association for Computational Linguistics: ACL 2024*, pp. 9215–9235, Bangkok, Thailand, August 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.findings-acl.548. URL https://aclanthology.org/2024.findings-acl.548.

Jieyi Long. Large language model guided tree-of-thought, 2023. URL https://arxiv.org/abs/2305.08291.

Bodhisattwa Prasad Majumder, Harshit Surana, Dhruv Agarwal, Bhavana Dalvi Mishra, Abhijeetsingh Meena, Aryan Prakhar, Tirth Vora, Tushar Khot, Ashish Sabharwal, and Peter Clark. Discoverybench: Towards data-driven discovery with large language models, 2024. URL https://arxiv.org/abs/2407.01725.

Mohamed Nejjar, Luca Zacharias, Fabian Stiehle, and Ingo Weber. Llms for science: Usage for code generation and data analysis. *J. Softw. Evol. Process*, 37(1), September 2024. ISSN 2047-7473. doi: 10.1002/smr.2723. URL https://doi.org/10.1002/smr.2723.

OpenAI, Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, Red Avila, Igor Babuschkin, Suchir Balaji, Valerie Balcom, Paul Baltescu, Haiming Bao, Mohammad Bavarian, Jeff Belgum, Irwan Bello, Jake Berdine, Gabriel Bernadett-Shapiro, Christopher Berner, Lenny Bogdonoff, Oleg Boiko, Madelaine Boyd, Anna-Luisa Brakman, Greg Brockman, Tim Brooks, Miles Brundage, Kevin Button, Trevor Cai, Rosie Campbell, Andrew Cann, Brittany Carey, Chelsea Carlson, Rory Carmichael, Brooke Chan, Che Chang, Fotis Chantzis, Derek Chen, Sully Chen, Ruby Chen, Jason Chen, Mark Chen, Ben Chess, Chester Cho, Casey Chu, Hyung Won Chung, Dave Cummings, Jeremiah Currier, Yunxing Dai, Cory Decareaux, Thomas Degry, Noah Deutsch, Damien Deville, Arka Dhar, David Dohan, Steve Dowling, Sheila Dunning, Adrien Ecoffet, Atty Eleti, Tyna Eloundou, David Farhi, Liam Fedus, Niko Felix, Simón Posada Fishman, Juston Forte, Isabella Fulford, Leo Gao, Elie Georges, Christian Gibson, Vik Goel, Tarun Gogineni, Gabriel Goh, Rapha Gontijo-Lopes, Jonathan Gordon, Morgan Grafstein, Scott Gray, Ryan Greene, Joshua Gross, Shixiang Shane Gu, Yufei Guo, Chris Hallacy, Jesse Han, Jeff Harris, Yuchen He, Mike Heaton, Johannes Heidecke, Chris Hesse, Alan Hickey, Wade Hickey, Peter Hoeschele, Brandon Houghton, Kenny Hsu, Shengli Hu, Xin Hu, Joost Huizinga, Shantanu Jain, Shawn Jain, Joanne Jang, Angela Jiang, Roger Jiang, Haozhun Jin, Denny Jin, Shino Jomoto, Billie Jonn, Heewoo Jun, Tomer Kaftan, Łukasz Kaiser, Ali Kamali, Ingmar Kanitscheider, Nitish Shirish Keskar, Tabarak Khan, Logan Kilpatrick, Jong Wook Kim, Christina Kim, Yongjik Kim, Jan Hendrik Kirchner, Jamie Kiros, Matt Knight, Daniel Kokotajlo, Łukasz Kondraciuk, Andrew Kondrich, Aris Konstantinidis, Kyle Kosic, Gretchen Krueger, Vishal Kuo, Michael Lampe, Ikai Lan, Teddy Lee, Jan Leike, Jade Leung, Daniel Levy, Chak Ming Li, Rachel Lim, Molly Lin, Stephanie Lin, Mateusz Litwin, Theresa Lopez, Ryan Lowe, Patricia Lue, Anna Makanju, Kim Malfacini, Sam Manning, Todor Markov, Yaniv Markovski, Bianca Martin, Katie Mayer, Andrew Mayne, Bob McGrew, Scott Mayer McKinney, Christine McLeavey, Paul McMillan, Jake McNeil, David Medina, Aalok Mehta, Jacob Menick, Luke Metz, Andrey Mishchenko, Pamela Mishkin, Vinnie Monaco, Evan Morikawa, Daniel Mossing, Tong Mu, Mira Murati, Oleg Murk, David Mély, Ashvin Nair, Reiichiro Nakano, Rajeev Nayak, Arvind Neelakantan, Richard Ngo, Hyeonwoo Noh, Long Ouyang, Cullen O'Keefe, Jakub Pachocki, Alex Paino, Joe Palermo, Ashley Pantuliano, Giambattista Parascandolo, Joel Parish, Emy Parparita, Alex Passos, Mikhail Pavlov, Andrew Peng, Adam Perelman, Filipe de Avila Belbute Peres, Michael Petrov, Henrique Ponde de Oliveira Pinto, Michael, Pokorny, Michelle Pokrass, Vitchyr H. Pong, Tolly Powell, Alethea Power, Boris Power, Elizabeth Proehl, Raul Puri, Alec Radford, Jack Rae, Aditya Ramesh, Cameron Raymond, Francis Real, Kendra Rimbach, Carl Ross, Bob Rotsted, Henri Roussez, Nick Ryder, Mario Saltarelli, Ted Sanders, Shibani Santurkar, Girish Sastry, Heather Schmidt, David Schnurr, John Schulman, Daniel Selsam, Kyla Sheppard, Toki Sherbakov, Jessica Shieh, Sarah Shoker, Pranav Shyam, Szymon Sidor, Eric Sigler, Maddie Simens, Jordan Sitkin, Katarina Slama, Ian Sohl, Benjamin Sokolowsky, Yang Song, Natalie Staudacher, Felipe Petroski Such, Natalie Summers, Ilya Sutskever, Jie Tang, Nikolas Tezak, Madeleine B. Thompson, Phil Tillet, Amin Tootoonchian, Elizabeth Tseng, Preston Tuggle, Nick Turley, Jerry Tworek, Juan Felipe Cerón Uribe, Andrea Vallone, Arun Vijayvergiya, Chelsea Voss, Carroll Wainwright, Justin Jay Wang, Alvin Wang, Ben Wang, Jonathan Ward, Jason Wei, CJ Weinmann, Akila Welihinda, Peter Welinder, Jiayi Weng, Lilian Weng, Matt Wiethoff, Dave Willner, Clemens Winter, Samuel Wolrich, Hannah Wong, Lauren Workman, Sherwin Wu, Jeff Wu, Michael Wu, Kai Xiao, Tao Xu, Sarah Yoo, Kevin Yu, Qiming Yuan, Wojciech Zaremba, Rowan Zellers, Chong Zhang, Marvin Zhang, Shengjia Zhao, Tianhao Zheng, Juntang Zhuang, William Zhuk, and Barret Zoph. Gpt-4 technical report, 2024. URL https://arxiv.org/abs/2303.08774.

Skipper Seabold and Josef Perktold. statsmodels: Econometric and statistical modeling with python. In *9th Python in Science Conference*, 2010.

Amit Sharma and Emre Kiciman. Dowhy: An end-to-end library for causal inference. *arXiv preprint arXiv:2011.04216*, 2020.

Gemini Team. Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context, 2024. URL https://arxiv.org/abs/2403.05530.

Marko Veljanovski and Zach Wood-Doughty. DoubleLingo: Causal estimation with large language models. In Kevin Duh, Helena Gomez, and Steven Bethard (eds.), *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 2: Short Papers)*, pp. 799–807, Mexico City, Mexico, June 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.naacl-short.71. URL https://aclanthology.org/2024.naacl-short.71/.

Xinyue Wang, Kun Zhou, Wenyi Wu, Har Simrat Singh, Fang Nan, Songyao Jin, Aryan Philip, Saloni Patnaik, Hou Zhu, Shivam Singh, Parjanya Prashant, Qian Shen, and Biwei Huang. Causal-copilot: An autonomous causal analysis agent, 2025. URL https://arxiv.org/abs/2504.13263.

Qingyun Wu, Gagan Bansal, Jieyu Zhang, Yiran Wu, Beibin Li, Erkang Zhu, Li Jiang, Xiaoyun Zhang, Shaokun Zhang, Jiale Liu, Ahmed Hassan Awadallah, Ryen W White, Doug Burger, and Chi Wang. Autogen: Enabling next-gen llm applications via multi-agent conversation, 2023. URL https://arxiv.org/abs/2308.08155.

Xueqing Wu, Rui Zheng, Jingzhen Sha, Te-Lin Wu, Hanyu Zhou, Tang Mohan, Kai-Wei Chang, Nanyun Peng, and Haoran Huang. DACO: Towards application-driven and comprehensive data analysis via code generation. In *The Thirty-eight Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2024. URL https://openreview.net/forum?id=NrCPBJSOOc.

Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Thomas L. Griffiths, Yuan Cao, and Karthik Narasimhan. Tree of thoughts: Deliberate problem solving with large language models, 2023a. URL https://arxiv.org/abs/2305.10601.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. React: Synergizing reasoning and acting in language models, 2023b. URL https://arxiv.org/abs/2210.03629.

Pengcheng Yin, Wen-Ding Li, Kefan Xiao, A. Eashaan Rao, Yeming Wen, Kensen Shi, Joshua Howland, Paige Bailey, Michele Catasta, Henryk Michalewski, Oleksandr Polozov, and Charles Sutton. Natural language to code generation in interactive data science notebooks. *ArXiv*, abs/2212.09248, 2022. URL https://api.semanticscholar.org/CorpusID:254854112.

Bin Yu. Veridical data science. In *Proceedings of the 13th International Conference on Web Search and Data Mining*, WSDM '20, pp. 4–5, New York, NY, USA, 2020. Association for Computing Machinery. ISBN 9781450368223. doi: 10.1145/3336191.3372191. URL https://doi.org/10.1145/3336191.3372191.

Lei Zhang, Yuge Zhang, Kan Ren, Dongsheng Li, and Yuqing Yang. Mlcopilot: Unleashing the power of large language models in solving machine learning tasks, 2024. URL https://arxiv.org/abs/2304.14979.

Shujian Zhang, Chengyue Gong, Lemeng Wu, Xingchao Liu, and Mingyuan Zhou. Automl-gpt: Automatic machine learning with gpt, 2023. URL https://arxiv.org/abs/2305.02499.

Yizhang Zhu, Shiyin Du, Boyan Li, Yuyu Luo, and Nan Tang. Are large language models good statisticians?, 2024. URL https://arxiv.org/abs/2406.07815.

A Dataset Generation

In Figure 5 we provide detailed steps for how we preprocess all three datasets. We also provide the distribution of the corresponding estimation methods across the three datasets in Figure 4.

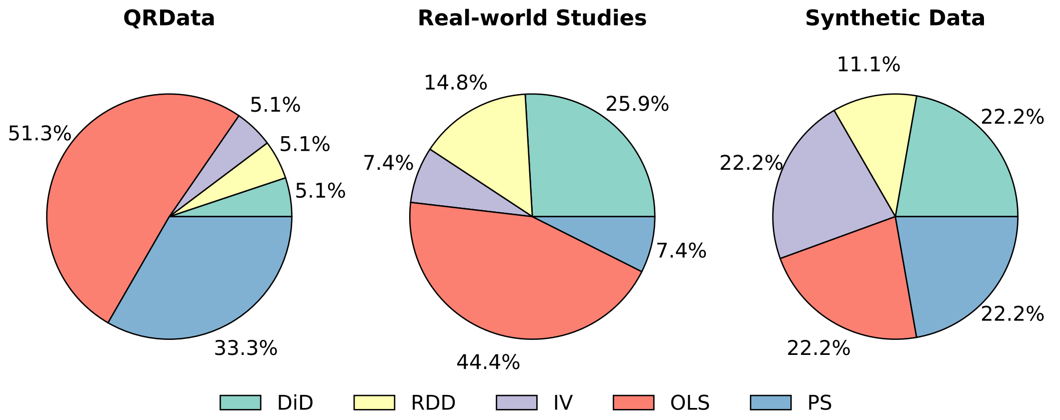
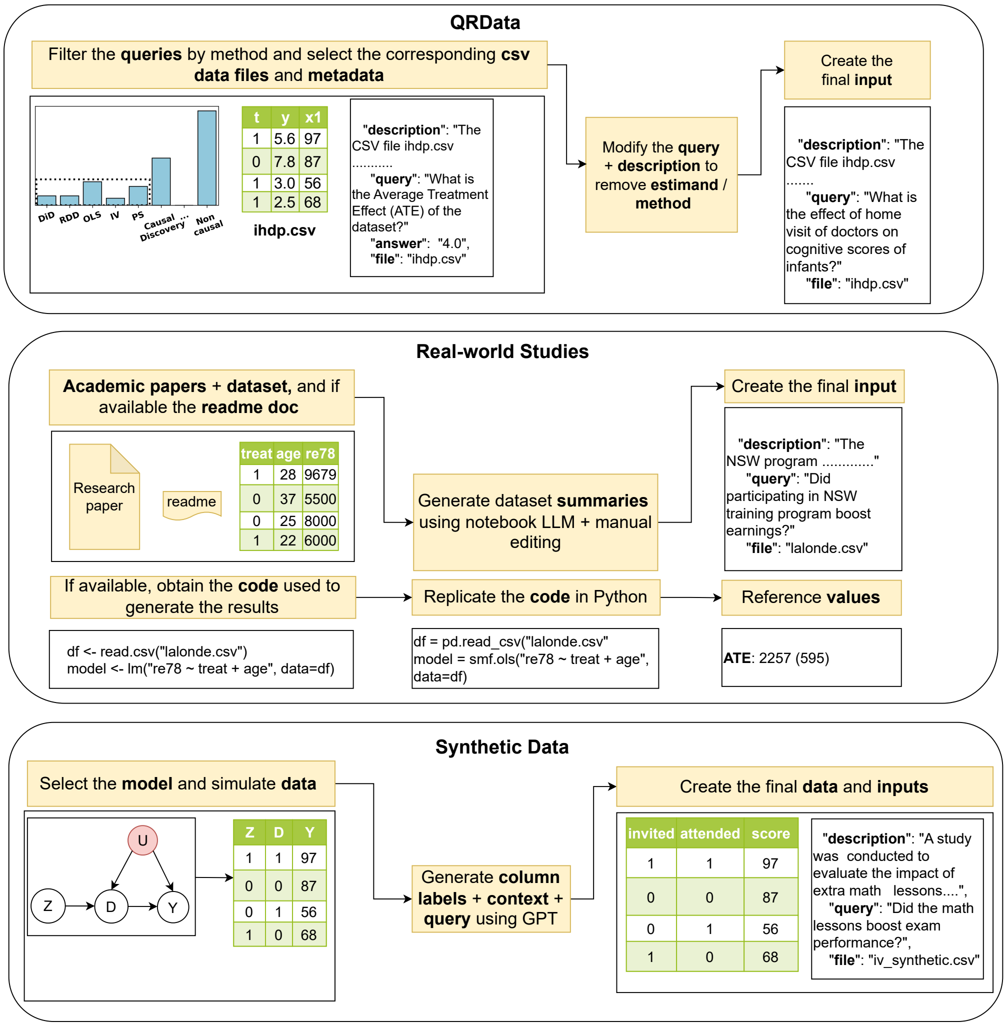


Figure 4: Distribution of estimation methods across the three dataset collections

A.1 Textbook Examples

The causal effect estimation tasks in QRData specify the inference method or estimand. Since our focus is on end-to-end causal inference, including automatic method and variable selection, we modify the queries by removing explicit references to estimation techniques or causal effect measures. For example, the original question, *"What is the Average Treatment Effect (ATE) of the dataset?"* is rephrased as *"What is the effect of home visits by doctors on*


QRData
Filter the queries by method and select the corresponding csv data files and metadata
DiD
RDD
OLS
IV
PS
Causal Discovery
...
Non causal
ihdp.csv
Modify the query + description to remove estimand / method
Create the final input
Real-world Studies
Academic papers + dataset, and if available the readme doc
Research paper
readme
Generate dataset summaries using notebook LLM + manual editing
Create the final input
If available, obtain the code used to generate the results
Replicate the code in Python
Reference values
Synthetic Data
Select the model and simulate data
U
Z
D
Y
Generate column labels + context + query using GPT
Create the final data and inputs

t	y	x1
1	5.6	97
0	7.8	87
1	3.0	56
1	2.5	68

"**description**": "The CSV file ihdp.csv ....
"**query**": "What is the Average Treatment Effect (ATE) of the dataset?"
"**answer**": "4.0",
"**file**": "ihdp.csv"

"**description**": "The CSV file ihdp.csv ....
"**query**": "What is the effect of home visit of doctors on cognitive scores of infants?"
"**file**": "ihdp.csv"

treat	age	re78
1	28	9679
0	37	5500
0	25	8000
1	22	6000

"**description**": "The NSW program ...."
"**query**": "Did participating in NSW training program boost earnings?"
"**file**": "lalonde.csv"

```
df <- read.csv("lalonde.csv")
model <- lm("re78 ~ treat + age", data=df)
```

```
df = pd.read_csv("lalonde.csv"
model = smf.ols("re78 ~ treat + age",
data=df)
```

ATE: 2257 (595)

Z	D	Y
1	1	97
0	0	87
0	1	56
1	0	68

invited	attended	score
1	1	97
0	0	87
0	1	56
1	0	68

"**description**": "A study was conducted to evaluate the impact of extra math lessons....",
"**query**": "Did the math lessons boost exam performance?",
"**file**": "iv_synthetic.csv"

Figure 5: Dataset creation process for **QRData, Real-world Studies, and Synthetic Data**

cognitive scores of infants?". We retain the original dataset descriptions and the associated numerical estimates of the causal effects. Additionally, we restrict our evaluation to queries with numerical answers.

A.2 Real-World Studies

We compile research papers from a range of disciplines, including economics, health policy, and political science. Many of these studies use datasets available in the R package causaldata. For each study, we create a summary that captures key information about the dataset, including variable descriptions, data sources, and collection procedures. We then formulate causal queries by examining the empirical results, the associated statistical models, and how they are interpreted in the original papers.

A.3 Synthetic Data

We randomly select the true causal effect τ in the range $(1, 10)$. Continuous covariates are drawn from a normal distribution, while binary covariates and treatment assignments (for binary treatment settings) are generated from a binomial distribution. The outcome Y is determined by the model specification. For example, for a randomized trial:

$$Y = \alpha + X\vec{\theta} + \tau T + \epsilon, \tag{3}$$

where $\epsilon \sim \mathcal{N}(0, 1)$ is the error term, $\vec{\theta} \sim \mathcal{N}(u, kI)$, and α is the intercept. Here, X denotes the covariates and T is the treatment variable.

We also use LLMs, specifically GPT-4o, to generate hypothetical contexts for each synthetic dataset. Specifically, we prompt the model to create plausible scenarios explaining how and why the data might have been collected. As part of the process, the LLM also produces dataset metadata, including headings and descriptions for covariates, treatment variables, and outcomes. This approach adds context to synthetic datasets and allows us to test CAIS's ability to handle real-world-like scenarios.

B Related Work

LLMs and causal inference The ability of LLMs to perform causal effect estimation in tabular datasets has been explored in Liu et al. (2024) and Chen et al. (2025). However, both approaches require users to specify the estimation method/variables, unlike CAIS where method and variable selection is automated. Jiang et al. (2024b) introduces a fine-tuned foundation model for both causal discovery and effect estimation. For the latter, the model does not support methods such as Instrumental Variables and Difference-in-Differences, which are widely used in the social sciences. Causal Co-pilot (Wang et al., 2025) expands the range of supported methods using general foundation models as the backbone, but has been evaluated primarily on causal discovery tasks. Another approach to causal inference involves building causal graphs from variable information using an LLM and applying techniques such as frontdoor and backdoor estimation to obtain the causal effects (Kiciman et al., 2024; Han et al., 2024). However, this approach limits us to graph-based methods only.

Beyond tabular data, LLMs have been applied to causal estimation with text data (Dhawan et al., 2024; Lin et al., 2023; Imai & Nakamura, 2024; Veljanovski & Wood-Doughty, 2024).

LLMs-powered data analysis Several works have studied the code generation capabilities of LLMs for data science tasks, including machine learning, statistical analysis, data manipulation, and visualization (Huang et al., 2022; Lai et al., 2023; Cheng et al., 2023; Nejjar et al., 2024; Jansen et al., 2023). However, these approaches require users to provide specific instructions. Wu et al. (2024) extends this line of work by enabling LLM-powered tools to perform statistical reasoning and generate solutions to natural language questions. However, these don't involve causal methods. A promising direction for end-to-end analysis is the development of LLM-powered agents. Most of these tools are geared towards machine learning tasks (Zhang et al., 2023; 2024; Huang et al., 2024) or data science tasks involving both machine learning and statistical methods (Guo et al., 2024; Hong et al., 2024). The capabilities of these tools have been enhanced through case-based reasoning (Guo et al., 2024), hierarchical decomposition (Hong et al., 2024), and interactive tools (Wu et al., 2023). However, these agents do not focus on causality-based analysis, which requires different methodological considerations.

Benchmark Datasets Earlier benchmarks focused on LLMs' ability to generate data science code. These include DS-1000 (Lai et al., 2023) and ARCADE (Yin et al., 2022). StatQA (Zhu et al., 2024), DACO (Wu et al., 2024), and (Hu et al., 2024) evaluated LLMs' ability to frame and implement quantitative methods to answer natural language queries related to input datasets. However, these benchmarks primarily focus on tasks that involve non-causal statistical methods. (Liu et al., 2024) is a benchmark dataset meant to test causal capabilities of LLMs. However, this focuses on implementing and interpreting specified methods. Our benchmark dataset is closely related to Blade (Gu et al., 2024) and DiscoveryBench (Majumder et al., 2024) in spirit. These two benchmarks, curated from real-world studies, seek to evaluate the ability of LLMs to perform data-driven scientific analysis to answer questions of interest. However, these datasets are more open-ended in terms of analysis tools while our dataset is geared specifically for causality-based analysis.

C Decision Tree for Model Selection

Figure 6 presents the sequence of diagnostic questions that CAIS asks the language model to link any dataset–query pair to the most suitable causal estimator. The root node checks whether the study is a randomized controlled trial; if it is, the tree first looks for an encouragement design, where assignment is random but compliance is imperfect, and directs such cases to instrumental-variable (IV) estimation. Fully compliant RCTs are divided by the presence of valid pre-treatment covariates, choosing covariate-adjusted OLS when those covariates are available and a simple difference-in-means when they are not. If the data are not randomized, the tree distinguishes between binary and non-binary treatments. For binary treatments, temporal structure leads to Difference-in-Differences, and a running variable triggers Regression Discontinuity Design. In the absence of both, the model checks for a valid backdoor adjustment set, frontdoor variables, or an instrument. When a valid backdoor adjustment set is found, the pipeline evaluates covariate overlap. Adequate overlap selects IPW, whereas limited overlap calls for matching. In general, the preference order is IV > frontdoor regression > backdoor adjustment. For non-binary treatments, the algorithm first seeks a valid instrument and applies IV if one exists. Without an instrument, it evaluates the frontdoor criterion. As a last resort, it falls back on backdoor adjustment with a regression model. This hierarchy yields a transparent mapping from contextual cues to estimator choice while also providing clear stopping rules when assumptions are unmet (Figure 6).

D Detailed Study: Method Validation Loop

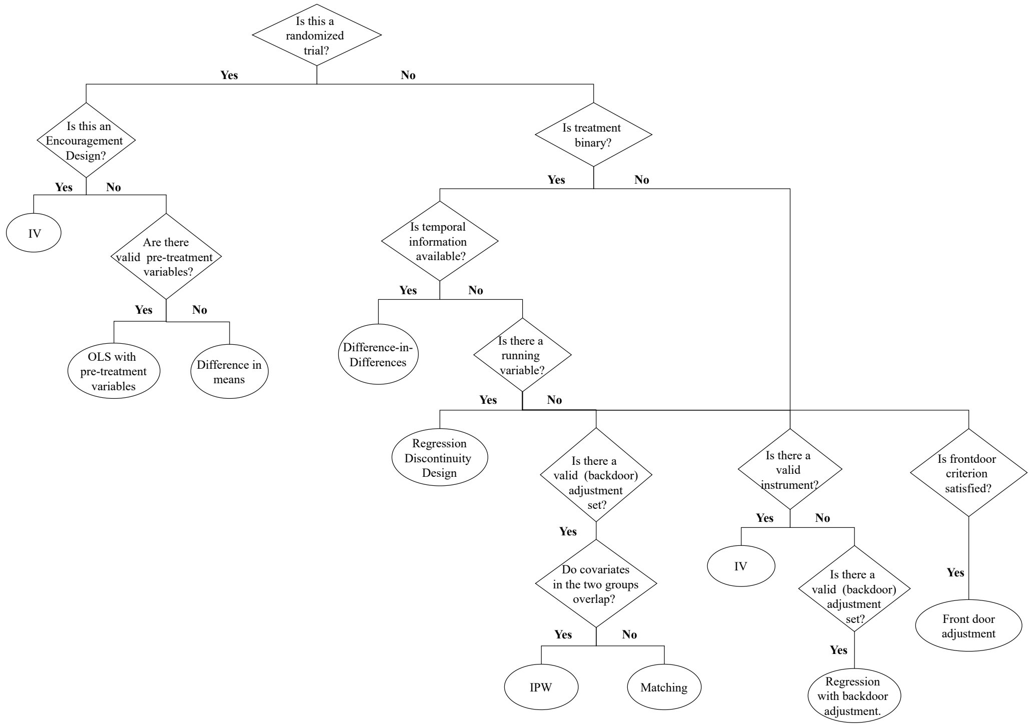

Is this a randomized trial?
Yes
No
Is this an Encouragement Design?
Is treatment binary?
Yes
No
Yes
No
IV
Are there valid pre-treatment variables?
Is temporal information available?
Yes
No
Yes
No
OLS with pre-treatment variables
Difference in means
Difference-in-Differences
Is there a running variable?
Yes
No
Regression Discontinuity Design
Is there a valid (backdoor) adjustment set?
Is there a valid instrument?
Is frontdoor criterion satisfied?
Yes
Yes
No
IV
Do covariates in the two groups overlap?
Is there a valid (backdoor) adjustment set?
Yes
Front door adjustment
Yes
No
Yes
IPW
Matching
Regression with backdoor adjustment.

Figure 6: Decision-tree that guides method selection in CAIS. We prompt an LLM to generate responses to queries corresponding to the decision nodes, and traverse the tree accordingly before reaching a leaf node, which corresponds to a method

Worked Example: Method Validation

Query: Does having access to electricity increase kerosene expenditures?

Dataset: `electrification_data.csv`

Database: All_Data Collection (Rural Electrification Survey)

Description: This household survey covers 686 households in 120 habitations across Uttar Pradesh, India. Using a geographic eligibility rule (households within 20–35 m vs. 45–60 m of a power pole), it records monthly expenditures on food, education, kerosene, total expenditure, appliance ownership, lighting usage, and satisfaction measures to assess the impact of electrification.

Method Validation: During validation, the pipeline fits local regressions on kerosene expenditure immediately below and above the 40 m cutoff to test for a sharp discontinuity. When using the lightweight gpt-4o-mini model, the agent misidentified the “distance” variable effectively widening the window around 40 m and consequently observed no statistically significant jump in outcomes at the threshold ($p > 0.05$). Because a pronounced, localized shift at the cutoff is the cornerstone of RDD, this absence of any detectable discontinuity constituted a direct violation of the RDD assumptions and led to its rejection. The system then automatically backtracked down the decision tree, removed RDD from consideration, and evaluated the next class of methods. Given the observational nature of the data and the rich set of covariates, it advanced to propensity-score-matching as the alternative method to create balanced treatment and control groups before estimating the effect.

E Baseline Prompts

Program of thought based Prompt

Prompt: You are a data analyst with strong quantitative reasoning skills. Your task is to answer a data-driven causal question using the provided dataset. The dataset description and query are given below.
You should analyze the **first 10 rows** of the dataset and then write Python code to generalize the analysis to the full table. You may use any Python libraries.

The returned value of the program should be the final answer. Please follow this format:

```
def solution():
    # import libraries if needed
    # load data from {self.dataset_path}
    # write code to get the answer
    # return answer

print(solution())
```

Dataset Description: {`self.dataset_description`} **Dataset Path:** {`self.dataset_path`}

First 10 rows of data: {`df.head(10)`}

Question: {`self.query`}

Example Methods (choose one if applicable):

- `propensity_score_weighting`: output the ATE
- `propensity_score_matching_treatment_to_control`: output the ATT
- `linear_regression`: output coefficient of variable of interest
- `instrumental_variable`: output coefficient
- `matching`: output the ATE
- `difference_in_differences`: output coefficient
- `regression_discontinuity_design`: output coefficient
- `linear_regression / difference_in_means`: output coefficient / DiM

Response: The final answer should include a structured summary with the following fields (use `"NA"` where not applicable):

- Method
- Causal Effect
- Standard Deviation
- Treatment Variable
- Outcome Variable
- Covariates
- Instrument / Running Variable / Temporal Variable
- Results of Statistical Test
- Explanation for Model Choice
- Regression Equation

Figure 7: Quantitative reasoning prompt for baseline evaluation.

ReACT Prompt Example

Prompt: You are working with a pandas DataFrame in Python. The name of the DataFrame is `df`.
You should use the tools below to answer the question posed to you:
`python_repl_ast`: A Python shell. Use this to execute Python commands. Input should be a valid Python command. When using this tool, sometimes output is abbreviated—make sure it does not look abbreviated before using it in your answer.
Use the following format:

- **Question:** the input question you must answer
- **Thought:** what you should do next
- **Action:** the action to take (e.g., `python_repl_ast`)
- **Action Input:** the input to the action (code to execute)
- **Observation:** the result of the action

(This Thought/Action/Action Input/Observation can repeat N times.)
Final Answer: The final answer to the original input question. Please provide a structured response including the following:

- Method
- Causal Effect
- Standard Deviation
- Treatment Variable
- Outcome Variable
- Covariates
- Instrument / Running Variable / Temporal Variable
- Results of Statistical Test
- Explanation for Model Choice
- Regression Equation

Instructions:

- Import libraries as needed.
- Do **not** create any plots.
- Use the `print()` function for all code outputs.
- If you output an Action step, stop after generating the Action Input and await execution.
- If you output the Final Answer, do not include an Action step.

Example Usage of `python_repl_ast`:
Action: `python_repl_ast`

Figure 8: Example of a ReACT-style prompt used in baseline prompting.

Veridical Prompt

You are an expert in statistics and causal reasoning. You will use a rigorous scientific framework to answer a causal question using a structured, step-by-step process with checklists.
Problem Statement: self.query
Step 1: Domain Understanding - What is the real-world question? Why is it important? - Could alternate formulations impact the final result?
Step 2: Dataset Overview - Dataset Path: dataset_path - Description: dataset_description - Dataset Summary, Types, Missing Values, Preview Rows
Checklist: - How was data collected? Design principles? - What are the variables, types, and units? - Are there errors or pre-processing artifacts?
Step 3: Exploratory Analysis - Identify confounders, mediators, biases - Suspect endogeneity? What instruments might be relevant? - Are strong correlations present?
Step 4: Modeling Strategy - Choose 3 candidate methods (e.g., matching, regression, IV) - State assumptions and reasons for each method - Discuss software libraries to be used and potential pitfalls - Outline key outputs and steps in analysis
Step 5: Post Hoc Analysis - Are relationships or outcomes unexpected? - Assess result stability and robustness
Step 6: Interpretation and Reporting Final Answer: Report the following fields: - Method, Causal Effect, Standard Deviation - Treatment and Outcome Variables - Covariates, Instruments or Temporal Elements - Results of any statistical tests - Justification of model choice - Equation or summary used

Figure 9: Veridical Style Prompting.