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

Vishal Verma[1] **Sawal Acharya**[2] **Samuel Simko**[3] **Devansh Bhardwaj**[4] **Anahita Haghighat**[5]
Mrinmaya Sachan[3] **Dominik Janzing**[6] *****Bernhard Schölkopf**[7] **Zhijing Jin**[7,8,9]
[1]Carnegie Mellon University [2]Stanford University [3]ETH Zürich [4]IIT Roorkee
[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

Large Language Models (LLMs) have recently been used to automate the causal inference process by overcoming the expertise barrier. However, existing LLM-powered approaches for causal effect estimation often require human users to manually specify variables and methods, and those that do not require manual specification 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. Specifically, given a natural language query and a dataset along with its description, CAIS uses LLMs to understand the user query and dataset, and then selects a method based on a decision tree approach. It then executes the selected method, applies a validation feedback loop for self-correction, and uses the results to answer the input question, enabling fully autonomous causal analysis. Extensive experiments across diverse queries curated from textbooks, synthetic data, and real-world datasets demonstrate CAIS's ability to produce precise causal effect estimates through improved method selection and self-correction, while reducing runtime errors. We believe CAIS will serve as a strong foundation for enabling fully automated causal inference with LLMs.

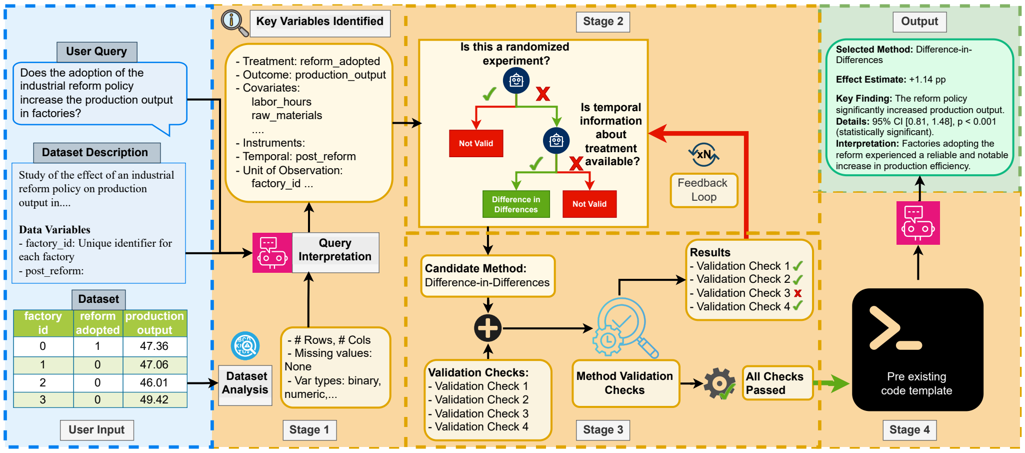
1 Introduction

Causal inference [Pearl, 2009, Imbens and Rubin, 2015] aims to quantify the effect of a treatment or intervention on an outcome of interest. Understanding cause-effect relationships is central to evidence-based decision-making in fields such as social science [Imbens, 2024], public health [Glass et al., 2013], and biomedicine [Kleinberg and Hripcsak, 2011]. Conducting rigorous causal analysis typically requires methodological expertise, from selecting valid causal effect measures (i.e., estimands) to choosing appropriate statistical methods, which limits accessibility for non-experts and poses significant challenges to fully automating the causal inference pipeline. For instance, a policy analyst with education and wage data may wish to estimate the effect of a job training program on earnings but, without causal identification knowledge, could reach invalid conclusions.

Recently, Large Language Models (LLMs) have emerged as a solution to overcoming the expertise barrier, as they can automate parts of the causal inference process using their extensive knowledge across various domains [Kiciman et al., 2024]. Jiang et al. [2024a] have developed a specialized

[*] This work is done outside Amazon.

Accepted to the NeurIPS 2025 Workshop on CauScien: Uncovering Causality in Science.



Does the adoption of the industrial reform policy increase the production output in factories?

Study of the effect of an industrial reform policy on production output in....

Data Variables
- factory_id: Unique identifier for each factory
- post_reform:

factory id	reform adopted	production output
0	1	47.36
1	0	47.06
2	0	46.01
3	0	49.42

- Treatment: reform_adopted
- Outcome: production_output
- Covariates:
  labor_hours
  raw_materials
  ....
- Instruments:
- Temporal: post_reform
- Unit of Observation:
  factory_id ...

- # Rows, # Cols
- Missing values: None
- Var types: binary, numeric,...

Selected Method: Difference-in-Differences

Effect Estimate: +1.14 pp

Key Finding: The reform policy significantly increased production output.
Details: 95% CI [0.81, 1.48], p < 0.001 (statistically significant).
Interpretation: Factories adopting the reform experienced a reliable and notable increase in production efficiency.

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.

foundational model, LLM4Causal, for causal inference and causal graph learning. Similarly, more recent approaches have developed causal agents that leverage general-purpose foundational models like GPT [OpenAI et al., 2024] to enable end-to-end performance of causal inference and learning tasks [Wang et al., 2025, Han et al., 2024]. However, existing approaches face three main limitations: ***(i)*** fine-tuned models are often narrow in scope, supporting only a limited set of causal effect measures and excluding widely used methods in applied research; ***(ii)*** general-purpose tools are primarily evaluated on causal discovery tasks, leaving their causal estimation capabilities untested; and ***(iii)*** those that are tested for causal effect estimation are evaluated on examples where users specifically mention the causal effect to be computed, which assumes prior knowledge of causal inference. These limitations are insufficient for enabling the automation of causal inference that is both comprehensive and accessible to others.

To address these limitations, we present Causal AI Scientist (CAIS), an end-to-end LLM-augmented causal tool for generating causality-driven answers to natural language queries. Given a dataset, its description, and a query, CAIS frames the task as a causal inference problem, automatically selects an appropriate method, estimates the causal effect, and interprets the result in context, as outlined in Figure 1. To select the right method, CAIS uses a structured decision tree that breaks down the selection process into focused steps. At each node, it prompts LLMs to evaluate specific features of the dataset or query, such as identifying the treatment, outcome, or instrument. This step-by-step approach reduces errors that commonly arise when relying entirely on LLMs for model selection. Additionally, CAIS performs diagnostic checks and incorporates a feedback loop to self-correct potential errors before producing a final answer. We evaluate CAIS on CauSciBench [Acharya et al., 2025], a real-world benchmark designed to evaluate the ability of LLMs to perform causal analysis on tabular datasets.

2 Problem Formulation and Methodology: CAIS

We begin by formalizing the task of automated causal inference. We are tasked with developing a system that can automatically perform causal effect estimation. The system receives three inputs:

- A tabular dataset, $\mathcal{D}$, containing observations for multiple units.
- Metadata describing the variables and the data collection process.
- A natural-language query, q, posing a causal question about the relationships within the data.

The objective is to interpret the query q in the context of the dataset $\mathcal{D}$ and its metadata to produce an estimate of the causal effect of interest. We provide a detailed formulation of the task in Appendix B.

Methodological Framework To address this task, we propose CAIS, a four-stage framework that integrates causal inference principles with LLM reasoning. Each stage employs one or more micro-tools, selectively applying human-like reasoning to sub-tasks requiring interpretation while relying on formal rules for structured decision-making. The overall pipeline is illustrated in Figure 1, with detailed descriptions of the specific micro-tools and stages provided in Appendix F.

Stages of CAIS Specifically, the system proceeds through the following stages:

- **Stage 1 (Data Preprocessing & Query Decomposition):** The process begins by analyzing the dataset to identify key components, such as control and target variables.
- **Stage 2 (Method Selection):** A rule-based decision tree is used to select a valid method for causal effect estimation based on the identified components.
- **Stage 3 (Validation):** The assumptions required by the chosen method are validated. If any assumption check fails, the system backtracks to Stage 2 to select an alternative method, forming a validation loop.
- **Stage 4 (Method Execution & Interpretation):** Once a valid method passes all checks, it is executed using predefined templates, and the final result is returned along with an interpretation.

3 Results

We evaluate the performance of CAIS on CauSciBench against three baseline prompting strategies: ReAct prompting [Yao et al., 2023], Program of Thoughts (PoT) prompting [Chen et al., 2022], and a Veridical Data Science-inspired prompt [Yu, 2020]. Our experiments employ several LLMs as the reasoning backbone, including GPT-4o, Llama-3.3-70B-Instruct, and Gemini 2.5 Pro, with the temperature set to 0 to ensure reproducibility. To assess effectiveness, we use two metrics: Method Selection Accuracy (MSA) and Mean Relative Error (MRE). The remaining details of the experimental setup are provided in Appendix G.

In addition, we conduct ablation studies (Section 3.1) to analyze the contributions of two key components of CAIS: the decision tree and the validation feedback loop. A detailed comparative performance analysis is provided in Appendix C, along with a thorough error analysis in Appendix D. Table 1 presents the main results, comparing CAIS with the baselines across three subsets of CauSciBench.

	Method Selection Accuracy (↑)					Mean Relative Error (↓)				
Method	**GPT-4o**	**GPT-4o-mini**	**o3-mini**	**Gemini 2.5 Pro**	**Llama 3.3 70B**	**GPT-4o**	**GPT-4o-mini**	**o3-mini**	**Gemini 2.5 Pro**	**Llama 3.3 70B**
					Textbook Data					
ReAct	55.0	55.2	21.8	62.2	34.4	43.2	33.9	44.7	43.2	43.9
PoT	41.0	54.3	30.7	50.0	53.8	32.6	**33.6**	30.7	**35.8**	**31.5**
Veridical	60.5	41.0	61.5	59.0	46.1	40.7	42.2	**27.6**	37.8	55.4
CAIS	**74.4**	**74.3**	**94.1**	**81.2**	**81.8**	**31.6**	55.9	43.1	41.2	54.2
					Synthetic Data					
ReAct	51.2	41.9	46.7	48.2	55.8	27.9	21.2	21.0	20.2	21.3
PoT	53.3	37.7	42.2	53.2	47.6	19.9	37.7	42.2	24.0	**21.1**
Veridical	**79.0**	43.4	66.6	58.5	50.0	27.7	25.7	20.2	26.5	33.3
CAIS	76.9	**75.9**	**73.3**	**75.6**	**79.5**	**17.4**	**16.2**	**20.0**	**18.5**	50.0
					Real Data					
ReAct	**69.5**	51.8	57.1	55.0	44.4	**43.1**	**52.3**	43.2	38.1	52.6
PoT	57.7	54.6	33.3	42.2	53.8	54.7	55.6	46.3	42.0	53.8
Veridical	48.0	28.0	59.2	53.2	24.0	53.6	54.4	41.2	39.0	52.8
CAIS	69.2	**65.2**	**76.9**	**78.3**	**73.0**	47.5	54.6	**39.7**	**32.0**	**37.4**

Table 1: 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. CAIS consistently outperforms baselines on MSA while maintaining competitive MRE.

CAIS shows superior method selection capabilities. Across all three datasets and models, we observe that CAIS outperforms the baselines in MSA, with significant margins in nearly all cases. For

Study	Method	GPT-4o	GPT-4o-mini	o3-mini
QR	LLM	48.4	50.6	50.0
	Tree	**74.4**	**74.3**	**94.1**
Real	LLM	48.0	60.8	45.5
	Tree	**69.2**	**65.2**	**76.9**
Synth	LLM	57.5	**79.4**	57.1
	Tree	**76.9**	78.9	**73.3**

(a) Decision tree vs. LLM-based method selection.

Study	Loop	GPT-4o	GPT-4o-mini	o3-mini
QR	No	**80.0**	41.2	97.1
	Yes	74.4	**74.3**	**94.1**
Real	No	56.0	33.3	75.0
	Yes	**69.2**	**65.2**	**76.9**
Synth	No	71.4	60.0	77.3
	Yes	**76.9**	**75.9**	**73.3**

(b) Impact of the validation feedback loop.

Table 2: Ablation results: (a) decision tree vs. LLM-based method selection, and (b) validation feedback loop.

example, on the textbook dataset, o3-mini achieves an MSA of 94.1%, which is 32.6 percentage points 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 data, +15.58 points on the synthetic data, and +14.10 points on the real data. These results demonstrate the effectiveness of our decision-tree-based method selection and validation feedback loop compared to relying solely on LLM reasoning.

CAIS achieves competitive causal estimation accuracy. Performance trends in MRE are more nuanced. On the synthetic and real datasets, our model performs competitively, achieving either the lowest or second-lowest MRE for most LLMs. However, a notable exception is the Textbook dataset, where our model consistently underperforms.

3.1 Ablation Studies

We conduct two ablation studies to assess the role of the core components in our pipeline: the *decision tree* and the *validation loop*. For the decision tree ablation, we compare CAIS with a variant that removes the decision tree and prompts an LLM to directly select the method.

Decision tree significantly improves method-selection accuracy. The results in Table 2a clearly show that incorporating a structured decision tree for method selection leads to substantially higher MSA compared to relying solely on the LLM's direct judgment. The decision tree decomposes the selection process into a sequence of targeted diagnostic questions, each focused on a specific dataset property (e.g., treatment timing, presence of instruments, covariate balance). This step-by-step approach constrains the LLM's reasoning to smaller, well-defined decisions, reducing the likelihood of overgeneralization or bias toward familiar methods. For example, on QRData, GPT-4o improves from 48.4% to 74.4% and o3-mini from 50% to 94.1%. Overall, the decision-tree-based approach is superior in performance compared to solely relying on LLMs.

The validation loop helps reduce the performance gap between weaker LLMs and stronger LLMs. As shown in Table 2b, the validation feedback loop is a critical component, especially for correcting errors from less capable models. This effect is most evident with GPT-4o-mini, whose accuracy improves dramatically from 41.2% to 74.3% (+33.1 points) on QRData and from 33.3% to 65.2% (+31.9 points) on the Real dataset. The loop provides this significant boost by allowing weaker models to recover from incorrect initial method selections, to which they are more prone. Conversely, the benefits are less pronounced for stronger models like GPT-4o since these models are more likely to select the correct method on the first attempt.

4 Conclusion

In this work, we introduce Causal AI Scientist (CAIS), an end-to-end tool that maps natural language queries and datasets to formal causal inference tasks by automatically selecting appropriate methods and interpreting results. When evaluated across diverse causal inference tasks using CauSciBench, CAIS consistently outperforms baseline prompting strategies in method selection and achieves competitive performance in causal effect estimation, particularly on structured datasets such as QRData and synthetic examples. These results highlight the value of CAIS's decision-tree-based approach, which decomposes complex reasoning into interpretable steps. This not only improves estimation accuracy but also enhances robustness and transparency—qualities critical for researchers and practitioners in social science, healthcare, and related fields.

References

Sawal Acharya, Terry Jingchen Zhang, Andrew Kim, Anahita Haghighat, Xianlin Sun, Maximilian Mordig, Rahul Babu Shrestha, Furkan Danisman, Clijo Jose, Yahang Qi, Pepijn Cobben, Bernhard Schölkopf, Mrinmaya Sachan, and Zhijing Jin. Causcibench: Assessing llm causal reasoning for scientific research. 2025. URL `https://zhijing-jin.com/files/papers/2025_CauSciBench.pdf`.

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`.

David Card. Using geographic variation in college proximity to estimate the return to schooling. Working Paper 4483, National Bureau of Economic Research, October 1993. URL `http://www.nber.org/papers/w4483`.

David Card and Alan B. Krueger. Minimum wages and employment: A case study of the fast-food industry in new jersey and pennsylvania. *The American Economic Review*, 84(4):772–793, 1994. ISSN 00028282. URL `http://www.jstor.org/stable/2118030`.

Christopher Carpenter and Carlos Dobkin. The effect of alcohol consumption on mortality: Regression discontinuity evidence from the minimum drinking age. *Am Econ J Appl Econ*, 1(1):164–182, Jan 2009. ISSN 1945-7782 (Print); 1945-7790 (Electronic); 1945-7790 (Linking). doi: 10.1257/app.1.1.164.

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`.

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.

Noah Greifer. *Assessing Balance*, May 2025. URL `https://cran.r-project.org/web/packages/MatchIt/vignettes/assessing-balance.html`. R package `MatchIt` vignette.

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`.

Keisuke Hirano and Guido W. Imbens. The propensity score with continuous treatments. *Applied Bayesian Modeling and Causal Inference from Incomplete-Data Perspectives*, pages 73–84, 2004.

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.

Paul W. Holland. Causal inference, path analysis, and recursive structural equations models. *Sociological Methodology*, 18:449–484, 1988. ISSN 00811750, 14679531. URL http://www.jstor.org/stable/271055.

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.

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, editors, *Proceedings of the Fourth Workshop on Data Science with Human-in-the-Loop (Language Advances)*, pages 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. Instrumental variables: An econometrician's perspective. *Statistical Science*, 29(3): 323–358, 2014. ISSN 08834237, 21688745. URL http://www.jstor.org/stable/43288511.

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, editors, *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 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, editors, *Findings of the Association for Computational Linguistics: ACL 2024*, pages 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`.

Edward Miguel and Michael Kremer. Worms: Identifying impacts on education and health in the presence of treatment externalities. *Econometrica*, 72(1):159–217, 2004. ISSN 00129682, 14680262. URL `http://www.jstor.org/stable/3598853`.

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.

Judea Pearl. *Causality: Models, Reasoning and Inference*. Cambridge University Press, New York, 2000.

Judea Pearl. Causal inference in statistics: An overview. *Statistics Surveys*, 3(none):96 – 146, 2009. doi: 10.1214/09-SS057. URL https://doi.org/10.1214/09-SS057.

Paul R. Rosenbaum and Donald B. Rubin. The central role of the propensity score in observational studies for causal effects. *Biometrika*, 70(1):41–55, 1983. ISSN 00063444, 14643510. URL http://www.jstor.org/stable/2335942.

Donald B Rubin. Causal inference using potential outcomes. *Journal of the American Statistical Association*, 100(469):322–331, 2005. doi: 10.1198/016214504000001880. URL https://doi.org/10.1198/016214504000001880.

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.

Elizabeth A. Stuart. Matching Methods for Causal Inference: A Review and a Look Forward. *Statistical Science*, 25(1):1 – 21, 2010. doi: 10.1214/09-STS313. URL https://doi.org/10.1214/09-STS313.

Marko Veljanovski and Zach Wood-Doughty. DoubleLingo: Causal estimation with large language models. In Kevin Duh, Helena Gomez, and Steven Bethard, editors, *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 2: Short Papers)*, pages 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, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. React: Synergizing reasoning and acting in language models, 2023. URL https://arxiv.org/abs/2210.03629.

Bin Yu. Veridical data science. In *Proceedings of the 13th International Conference on Web Search and Data Mining*, WSDM '20, page 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.

Acknowledgements

We would like to thank Patrick Blöbaum for helpful discussions on adapting PyWhy into CAIS, as well as for ideas on automating causal inference. We also thank Vivian Yvonne Nastl for directing us to empirical papers that apply Pearl's approach to causal effect estimation. We are further grateful to Daniela Schkoda, Philipp Faller, and Sergio Hernan Garrido Mejiar for their valuable suggestions and discussions on causal learning, causal inference, and their intersection with machine learning.

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.

A Related Work

LLMs and Causal Inference LLMs have been applied to causal inference with text data [Dhawan et al., 2024, Lin et al., 2023, Imai and Nakamura, 2024, Veljanovski and Wood-Doughty, 2024]. Recent research has also explored their use for causal effect estimation in tabular datasets [Liu et al., 2024, Chen et al., 2025]. However, these approaches often require users to specify the estimation method or variables. Jiang et al. [2024b] introduced a fine-tuned model for causal discovery and effect estimation, but it does not support methods like Instrumental Variables and Difference-in-Differences. Causal-Copilot [Wang et al., 2025] expands the range of methods but was primarily evaluated on causal discovery, not causal inference. Another approach builds causal graphs with LLMs [Kiciman et al., 2024, Han et al., 2024], but this is limited to graph-based estimation techniques. In contrast, our model automates variable and method selection using a decision tree and adds self-correction, creating a fully automatic causal inference pipeline for tabular datasets.

LLM-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] extend this line of work by enabling LLM-powered tools to perform statistical reasoning and generate solutions to natural language questions. However, these do not involve causal methods. A promising direction for end-to-end analysis is the development of LLM-powered agents. Most of these tools are geared toward 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.

B Task Formulation

B.1 Causal Model and Estimand

To formalize this task, we adopt the potential outcomes framework [Rubin, 2005]. The system must first parse the query q and the metadata to identify the key variables: the **treatment** (T), the **outcome** (Y), and a set of covariates (X). Each unit i has two potential outcomes: $Y_i(1)$, the outcome if the unit received the treatment, and $Y_i(0)$, the outcome if it did not. Similarly, T_i denotes whether unit i belongs to the treatment ($T_i = 1$) or the control ($T_i = 0$) group.

One of the causal estimands of interest is the Average Treatment Effect (**ATE**), defined as the expected difference between these potential outcomes across the population:

$$\tau_{ATE} = \mathbb{E}[Y(1) - Y(0)]$$

However, the fundamental problem of causal inference is that for any given unit, we can only observe one of its potential outcomes [Holland, 1986]. The observed outcome is $Y_i = Y_i(T_i)$. This means the ATE cannot be calculated directly.

Example: Job Training Program To make this concrete, consider a dataset from a job-training program (e.g., the Lalonde dataset [LaLonde, 1986]) and the query, "*Does the training program boost earnings?*". Here, the treatment T is program participation, the outcome Y is earnings, and covariates X could include education, age, and prior income. The ATE would represent the average boost in earnings if everyone in the population participated in the program versus if no one did.

B.2 Identification: From Randomized to Observational Data

The process of connecting the unobservable ATE to our observed data is called **identification**. This requires making assumptions.

The simplest case is a Randomized Controlled Trial (RCT), where individuals are randomly assigned to treatment ($T = 1$) or control ($T = 0$). This randomization ensures that, on average, the two groups are identical before treatment. This satisfies the **ignorability** assumption ($(Y(1), Y(0)) \perp\!\!\!\perp T$), meaning the treatment assignment is independent of the potential outcomes. In an RCT, the ATE can be simply estimated by the difference in the average outcomes of the two groups:

$$\hat{\tau}_{ATE} = \frac{1}{N_1} \sum_{i:T_i=1} Y_i - \frac{1}{N_0} \sum_{i:T_i=0} Y_i. \tag{1}$$

However, most data is **observational**, not from an RCT. In observational data, ignorability is often violated. For instance, more motivated individuals (who might have higher earnings potential anyway) may be more likely to sign up for a job training program. This motivation is a **confounder**, a variable that affects both treatment assignment and the outcome.

To handle confounders, we rely on a stronger, untestable assumption: **conditional ignorability**. This states that the treatment assignment is independent of the potential outcomes when conditioned on all the common causes (X) of T and Y. Mathematically, it can be formulated as follows:

$$(Y(1), Y(0)) \perp\!\!\!\perp T \mid X \tag{2}$$

Along with the **Stable Unit Treatment Value Assumption (SUTVA)** (no interference between units and consistency of treatment), conditional ignorability allows us to identify the ATE by adjusting for the covariates X:

$$\tau_{ATE} = \mathbb{E}_X[\mathbb{E}[Y \mid T = 1, X] - \mathbb{E}[Y \mid T = 0, X]]$$

Estimation via Statistical Models To compute the quantity above, we use structural causal models [Pearl, 2009] to describe the relationships between Y, T, and X. A common approach is a linear structural model, which can be expressed as below:

$$Y = \alpha + X^T \beta + \tau T + \epsilon, \tag{3}$$

where α is the intercept, β is a vector of coefficients for the covariates X, τ is the treatment effect parameter, and ϵ is the unobserved error term. In this model, the coefficient τ directly corresponds to the ATE, as it represents the change in Y for a one-unit change in T after adjusting for X. We can then use an estimation method, such as linear regression, to find a sample-based estimate, $\hat{\tau}$, from the data.

B.3 The Role of the LLMs: Supplying Domain Expertise

The entire causal inference process hinges on the validity of the conditional ignorability assumption, which cannot be verified from data alone. Justifying this assumption requires **domain knowledge** to argue that the set of measured covariates X is sufficient to account for all major confounders. This is precisely where the LLMs come in. The LLMs are designed to act as a proxy for a human domain expert. By leveraging their vast knowledge base, the LLMs can analyze the dataset's metadata and the user's query to:

Metric	Baseline	CAIS	Change (%)
General Statistics			
Total Queries	1551	585	–
Successful Queries	1476	512	–
Total Retries	930	159	–
Retries per Query (%)	59.96	**27.18**	↓ 54.69
Method Match Rate (%)	52.08	**76.20**	↑ 46.3
Mean Error (%)	**35.38**	37.66	↑ 6.4
Error Breakdown (%)			
Execution & Runtime Error	34.39	**22.91**	↓ 33.4
Method Mismatch	29.77	**21.20**	↓ 28.8
Data Loading Failure	3.10	**0.00**	↓ 100.0
Missing Result	0.76	6.84	↑ 800.0

Table 3: Comparison of performance and error types between baseline and CAIS. Arrow indicates direction of change from baseline to CAIS.

- Propose a plausible causal relationship between the variables.
- Identify the most likely confounders that must be included in the set X.
- Justify the conditional ignorability assumption, thereby enabling a principled estimation of the causal effect.

C Comparative Performance Analysis

Here, we compare the overall performance of CAIS, which uses a structured approach, with baselines that rely on prompting-based LLM code generation, focusing on execution stability, error types, and the frequency of retries.

- **Higher Method Selection Accuracy:** CAIS achieves a 46.3% higher method match rate than the baseline (76.2% vs. 52.08%), indicating more accurate identification of appropriate causal methods.
- **Substantial Reduction in Retries:** CAIS reduces total retries by 54.6% per query (In CAIS, a retry refers to feedback via validation loop), suggesting more robust and executable outputs due to structured prompt generation and template-based code execution.
- **Improved Execution Stability:** Execution and runtime errors are reduced by 33.4%, and method mismatches decrease by 28.8%, reflecting enhanced reliability in model reasoning and implementation.
- **No Data Loading Failures:** CAIS handles datasets more reliably with 0% data loading failures compared to 3.1% in the baseline.
- **Trade-offs in Estimation Quality:** While CAIS increases mean error slightly (from 35.38% to 37.66%), this may stem from using more advanced methods rather than defaulting to simple linear regression.

D Error Analysis

This section provides a qualitative breakdown of the model's primary failure modes. We analyze common error patterns to identify their root causes and determine which stages of the pipeline are most vulnerable.

- **Incorrect Variable Selection:** LLMs frequently misinterpret temporal covariates, such as birth year or quarter indicators, as observation time points. This misinterpretation can erroneously 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.

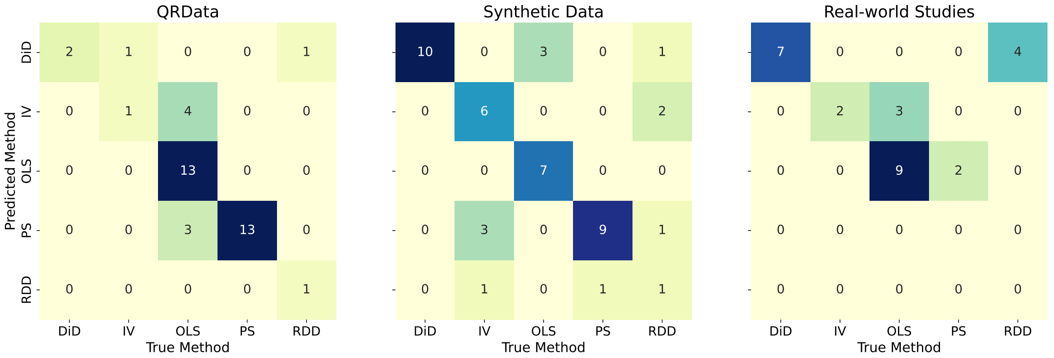


Figure 2: Confusion matrix showing method selection performance of CAIS (GPT-4o).

- **Wrong Method Selection:** As demonstrated in Figure 2, 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.

E Explanation of decision tree

E.1 Notation

We first define key notation used throughout this guide:

- **Y**: Outcome (the variable we want to understand or predict)
- **T**: Treatment assignment (whether a unit is assigned to treatment; $T = 0$ indicates the control group, $T = 1$ indicates the treatment group)
- **D**: Treatment uptake (actual receipt of treatment, used in encouragement designs where assignment is not the same as uptake)
- **Z**: Instrumental variable (a variable used to identify causal effects in the presence of unobserved confounding)
- **M**: Mediator (a variable that lies on the causal pathway between treatment and outcome)
- **U**: Unobserved confounder (a variable that affects both treatment and outcome but is not measured)
- **X**: Covariates (observed variables that may influence treatment or outcome)
- **i**: Individual units of analysis (for example people, organizations, or countries)

E.2 Randomized Controlled Trials (RCTs)

We begin by determining if the data comes from a Randomized Controlled Trial (RCT). RCTs are the gold standard for causal inference because random assignment eliminates confounding by making treatment and control groups comparable on average.

E.2.1 Encouragement Designs

Within RCTs, we first check if the data comes from an encouragement design. An encouragement design refers to scenarios where treatment assignment is random, but not all assignees accept or take up their assigned treatment, i.e., $T_i \neq D_i$ [Holland, 1988].

A classic example is the deworming experiment by Miguel and Kremer [Miguel and Kremer, 2004]. Students were randomly assigned to receive deworming drugs, but not all students who were assigned actually took the drugs. In such cases, the corresponding relation would be:

$$T \text{ (randomly assigned to get dewormed)} \rightarrow D \text{ (actually took the drug)} \rightarrow Y \text{ (school enrollment).}$$

To estimate the effect of actual treatment uptake (rather than just assignment), one uses instrumental variable analysis, with the random assignment T serving as an instrument for actual uptake D.

E.2.2 Standard RCT Analysis

If the data is not from an encouragement design, meaning everyone accepts their assigned treatment, this is the classic RCT analysis. One can simply compute the difference in means between the treatment and control groups.

In many cases, the dataset might have pretreatment variables (baseline characteristics measured before treatment). While one can still compute a simple difference in means, the trend in the literature is to include these pretreatment variables as controls in an Ordinary Least Squares regression model. This is primarily done to improve the precision of the estimates (reduce the standard errors) rather than to address bias, since randomization already handles confounding.

E.3 Observational Studies

If the data is not from an RCT, we move to observational methods. Here, we cannot rely on randomization to eliminate confounding. Techniques to address confounding varies according to the characteristics of the data.

E.3.1 Binary Treatments with Temporal Variation

Difference in Differences (DiD) For binary treatments, we first consider Difference in Differences when time based confounding is a concern. DiD requires information about treatment timing, which could be either binary (pre or post treatment) or staggered (different units adopting treatment at different time periods).

A classic example is the minimum wage study by Card and Krueger [Card and Krueger, 1994]. Pennsylvania and New Jersey are two states with similar characteristics. New Jersey increased the minimum wage, but Pennsylvania did not. We want to know the effect of the minimum wage policy on employment. However, over time many things change that could affect employment outcomes. DiD uses the control group (Pennsylvania) as a counterfactual. Since the two states are similar, they would have evolved similarly absent treatment. By subtracting the changes in the control state from the changes in the treated state, DiD removes time varying confounders that affect both states equally. If no temporal information is available, DiD cannot be used.

Regression Discontinuity Design (RDD) If DiD is not applicable, we check for Regression Discontinuity Design. RDD exploits situations where treatment assignment is determined by a threshold or cutoff rule, creating a sharp change in treatment status at a specific value of a running variable.

For example, to study the impact of alcohol consumption on road accidents [Carpenter and Dobkin, 2009], we can use the fact that alcohol consumption is legally allowed after age 21 in the United States. By comparing accident rates for people just above and below age 21, we can identify the causal effect of legal drinking. Here, age is the running variable, and treatment assignment is: treatment $= 1$ if age ≥ 21, treatment $= 0$ if age < 21. If there is no clear running variable with a threshold that determines treatment assignment, RDD cannot be used.

E.3.2 General Methods for Binary Treatments

DiD and RDD can potentially work when the dataset meets certain characteristics. This would be the presence of treatment and outcome information over time for DiD and the presence of a running variable for RDD. If the dataset does not meet the respective criteria, we can rule those methods out. Next, we describe a more general group of methods.

Backdoor adjustment Observational studies often rely on the conditional ignorability assumption: treatment assignment is independent of potential outcomes conditional on all observed confounders. This requires that all relevant confounders are observed. We can satisfy this assumption using Pearl's backdoor adjustment criterion [Pearl, 2000]. For causal effect estimation, we use inverse probability weighting (IPW) using propensity scores [Rosenbaum and Rubin, 1983] and matching [Stuart, 2010]. Propensity score measures the probability that a unit receives the treatment given the observed covariates. Mathematically, $e(X) = P(T = 1|X)$. One can estimate propensity scores by fitting a logit or a probit model with X as the dependent variable and T as the independent variable. To choose between IPW and matching, we assess covariate balance using the standardized mean difference (SMD) [Greifer, 2025]:

$$\text{SMD} = \frac{\bar{X}_{\text{treated}} - \bar{X}_{\text{control}}}{\sqrt{\frac{s^2_{\text{treated}} + s^2_{\text{control}}}{2}}}.$$

where $\bar{X}_{\text{treated}}$ denotes the mean of the covariates in the treated group, and s^2_{treated} their variance. Analogously, $\bar{X}_{\text{control}}$ and s^2_{control} refer to the mean and variance of the covariates in the control group.

If covariates are well balanced (typically SMD < 0.1), we use IPW methods directly. If covariates are poorly balanced, we use matching techniques to improve balance before estimating causal effects.

E.3.3 Methods for Unmeasured Confounding

Backdoor adjustment based methods fail if important confounders are unobserved. The presence of unobserved confounders decision is mostly made by combining domain knowledge and the data generating process. In cases where unobserved confounders are suspected, we can use Instrumental Variable or frontdoor estimation.

Instrumental Variables (IV) The first approach is IV analysis [Imbens, 2014], which requires a valid instrument satisfying two conditions:

1. **Relevance**: The instrument must be correlated with treatment (Cov$(Z, T) \neq 0$).
2. **Exclusion restriction**: The instrument should affect the outcome only through treatment, not directly.

The relevance assumption is testable, while the exclusion restriction must be justified through domain knowledge. A classic example of IV analysis is Card's study [Card, 1993] estimating returns to years of education, using geographic proximity to college as an instrument. Many unobserved factors (parental income, personal motivation) affect earnings and education, making it difficult to isolate education's causal effect. However, proximity to college affects educational attainment (relevance) but has no direct effect on earnings except through education (exclusion restriction). Finding good instruments is challenging, and quite often the dataset does not contain good candidates for an instrument.

Frontdoor Criterion Another option that may work in the presence of unobserved confounders is frontdoor estimation based on the frontdoor criterion [Pearl, 2000]. This works when there exists a mediator that completely captures the treatment's effect on the outcome and satisfies the frontdoor criterion:

- The mediator must completely intercept the path from the treatment to the outcome.
- The relationship between the treatment and the mediator must not be confounded.
- The treatment must block all confounding paths between the mediator and the outcome.

E.3.4 Method Hierarchy

When multiple methods are applicable, there is a clear preference hierarchy:

$$\text{Instrumental Variables} > \text{Frontdoor Criterion} > \text{Backdoor Adjustment}.$$

This hierarchy exists because IV and frontdoor methods can handle unobserved confounders, while backdoor adjustment requires that all confounders are observed. However, IV and frontdoor methods are less generally applicable, since finding valid instruments or suitable mediating variables is often challenging. Backdoor adjustment methods are more widely applicable but require stronger assumptions about confounding.

E.3.5 Nonbinary treatments

For nonbinary treatments in observational studies, we consider three methods: instrumental variables (IV), frontdoor adjustment, and generalized propensity scores using the backdoor adjustment set. IV and frontdoor approaches apply almost exactly as in the case of binary treatments described above. Backdoor estimation also works similarly, except that we use a different estimation method to compute causal effects. In this case, we use generalized propensity scores, which extend the idea of propensity scores to continuous treatments [Hirano and Imbens, 2004]. The scores are computed using the backdoor adjustment set.

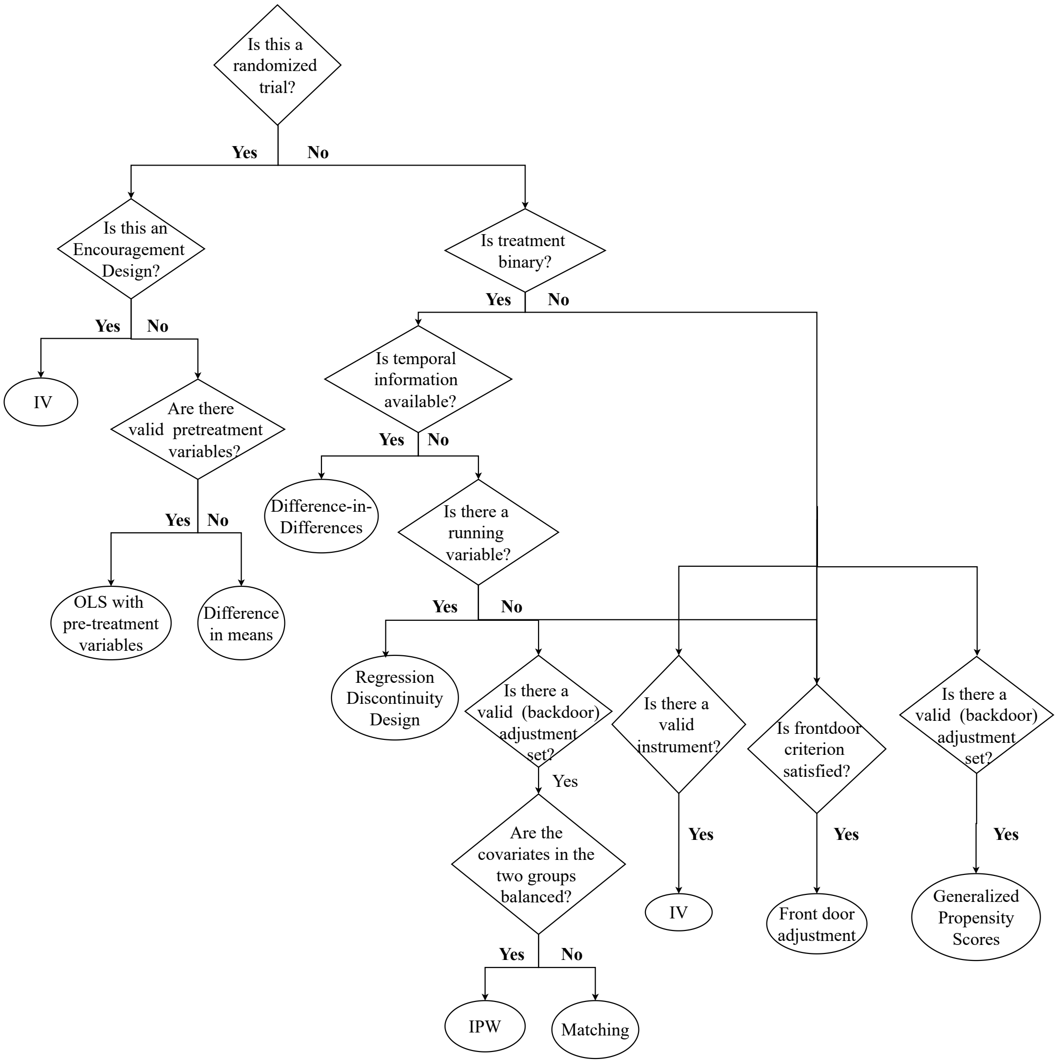

Is this a randomized trial?
Yes
No
Is this an Encouragement Design?
Is treatment binary?
Yes
No
Yes
No
IV
Are there valid pretreatment variables?
Is temporal information available?
Yes
No
Yes
No
OLS with pre-treatment variables
Difference in means
Difference-in-Differences
Is there a running variable?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

Figure 3: 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.

F Detailed Methodology

In this section we extend the Methodology of CAIS discussed in Section **??** in detail. CAIS operates in four sequential stages. Each stage comprises one or more micro-tools that pass a typed artifact to the next stage. A validation loop connects Stage 3 back to Stage 2 when assumptions fail. In total, CAIS uses eight micro-tools: (1) input_parser, (2) dataset_analyzer, (3) query_interpreter, (4) method_selector, (5) method_validator, (6) method_executor, (7) explanation_generator, and (8) output_formatter. All tools read/write a shared typed state (dataset profile, normalized query, analysis plan, selected method with assumptions, diagnostics, estimates, and final report).

F.1 Stage 1: Data Preprocessing & Query Decomposition

We begin by using **`input_parser`** tool to analyze user-specified causal queries and extract three key components: the *query type*, the *relevant variables*, and any *explicit constraints*. It adopts a hybrid strategy combining regex-based heuristics with LLM-driven structured parsing. A tailored prompt H.1, enriched with dataset context when available, guides the LLM to classify the query (e.g., `EFFECT_ESTIMATION`, `COUNTERFACTUAL`, `CORRELATION`, `DESCRIPTIVE`, or `OTHER`) and to output a structured JSON conforming to a predefined schema. This schema enforces explicit roles for variables (treatment, outcome, covariates, grouping variables, instruments) and records constraints or dataset paths if mentioned. The LLM output is validated to ensure logical consistency (e.g., requiring both treatment and outcome for effect estimation queries) and alignment with dataset columns when available. Regex-based matching serves as a complementary mechanism, particularly for dataset path extraction, thereby improving robustness. The final output is a standardized dictionary encapsulating the original query, its type, extracted variables, constraints, and dataset path.

Following this, the **`dataset_analyzer`** tool profiles the dataset to identify characteristics relevant for causal inference. It begins by extracting basic metadata such as the number of rows, columns, file name, and data types. A detailed analysis is then conducted to detect *temporal structures*, *panel data patterns*, and *potential discontinuities*, alongside computing correlations among numeric variables. The module further identifies *potential treatments* and *outcomes*, either heuristically or with LLM assistance, and assesses candidate *instrumental variables*. The prompts used in this stage are relatively light weight and mainly framed as direct questions to the LLM, an example prompt for identifying potential treatment and outcome variables can be found in Appendix H.2. For binary treatment candidates, it computes per-group summary statistics (e.g., group sizes, means, and standard deviations of covariates) to facilitate balance checks. Additionally, it records missing values, unique value counts, and column categorizations to provide a structured overview of the dataset. The final output is a comprehensive dictionary that consolidates dataset information, candidate causal variables, detected structures, and diagnostic statistics.

Finally, the **`query_interpreter`** reconciles the normalized query with the dataset profile to materialize the treatment variable T, outcome Y, admissible covariates X, and any design-specific fields (instruments Z, running variable R with cutoff c, time and group indicators for DiD, etc.). For each of these fields, the system issues targeted prompts to an LLM, these structured prompts can be found in Appendix H.3. The responses are aggregated and validated to ensure coherence with both the dataset schema and causal design assumptions. The final output is a structured *analysis plan* comprising named columns, encoding details (e.g., reference levels, interaction terms), and a canonical estimand (such as ATE, ATT, LATE, sharp/fuzzy RDD, or DiD horizon), which then serves as the input for subsequent method selection phase.

F.2 Stage 2: Method Selection

Given the analysis plan, the **`method_selector`** chooses a candidate estimator via a deterministic decision tree that encodes standard causal design logic. The selector also emits an assumption checklist and an ordered fallback list to support backtracking. This stage is completelty deterministic and requires no involvement of LLMs. The design choice behind the nodes of the decision tree is already discussed in great detail in Appendix E.

F.3 Stage 3: Validation

Once a candidate estimator is selected, the `method_validator` module verifies whether the assumptions required for its validity hold in the given dataset. This includes statistical diagnostics (e.g., overlap and positivity checks for treatment assignment, relevance and exogeneity tests for instruments, bandwidth support for RDD, and parallel trends diagnostics for DiD). The validator also cross-checks variable roles (treatment, outcome, covariates, instruments, etc.) against the dataset schema to ensure consistency. Failures in any diagnostic trigger a backtracking mechanism: the system reverts to the Method Selector's fallback list to propose an alternative estimator whose assumptions are more compatible with the data. This creates a validation loop that guarantees every executed method is grounded in both causal theory and empirical feasibility, while preserving determinism in the diagnostic procedures. In this stage as well no LLMs are involved.

F.4 Stage 4: Method Execution & Interpretation

On a validated design, the Method Executor runs a pre-verified estimator template (statsmodels/2SLS, DiD with appropriate fixed effects, local polynomial RDD, matching/weighting for PSM/PSW, GPS, diff-in-means). Execution favors templates over code-generation for stability and speed. One cross-cutting prompt, STATSMODELS_PARAMS_IDENTIFICATION_PROMPT_TEMPLATE, helps select the correct coefficient(s) to report when formulas include encodings or interactions (used by linear/GLM-style estimators). Several methods expose small, function-local LLM assists for parameter suggestions or narrative—e.g., IV, RDD, DiD, GPS/backdoor/diff-in-means/linear regression helpers, each constrained to strict JSON and backed by deterministic fallbacks. Finally Explanation Generator then converts structured artifacts (chosen design, diagnostics, estimates) into a concise, dataset-specific justification and interpretation, while the Output Formatter assembles the final response object with numeric results, uncertainty, method card (assumptions & checks), and a short plain-language answer for downstream rendering.

G Experimental Setup

Dataset To evaluate CAIS, we use CauSciBench [Acharya et al., 2025], a comprehensive benchmark for assessing LLMs on causal-estimation tasks. CauSciBench consists of 113 causal queries drawn from three sources: (1) QRData [Liu et al., 2024]; (2) published papers across multiple disciplines; and (3) synthetic datasets with known ground-truth causal effects. Summary statistics of CauSciBench are presented in Table **??**.

Baselines Given the lack of LLM-based tools for fully automated causal inference, we compare our method against three strong prompting strategies. We utilise ReAct prompting [Yao et al., 2023], which prior work shows to be highly effective for causal inference [Liu et al., 2024], Program of Thoughts (PoT) prompting [Chen et al., 2022] and a Veridical Data Science–inspired prompt [Yu, 2020]. Examples of these prompts are provided in Appendix J.

Implementation Details For estimating causal effects, we use the DoWhy [Sharma and Kiciman, 2020, Blöbaum et al., 2024] and statsmodels [Seabold and Perktold, 2010] Python libraries. Our experiments use several LLMs as the reasoning backbone, including GPT-4o, Llama-3.3-70B-Instruct, and Gemini 2.5 Pro. All models were accessed via their respective APIs. To ensure reproducibility for all experiments, we used greedy decoding by setting the temperature parameter to 0.

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).
- **Mean Relative Error (MRE)**: Average relative error between predicted causal effects $(\hat{\tau}_i)$ and reference values (τ_i). Relative error is capped at 100% per query.

H Methodology Prompts

H.1 Causal Query Parsing Prompt

Analyze the following causal query **strictly in the context of the provided dataset information (if available)**. Identify the query type, key variables (mapping query terms to actual column names when possible), constraints, and any explicitly mentioned dataset path.
User Query: `"{query}`
`{dataset_context}`
Guidance for Identifying Query Type:

- **EFFECT_ESTIMATION**: Look for keywords like "effect", "impact", "influence", "cause", "affect", "consequence". Also consider questions asking "how does X affect Y?" or comparing outcomes between groups based on an intervention.
- **COUNTERFACTUAL**: Look for hypothetical scenarios, often using phrases like "what if", "if X had been", "would Y have changed", "imagine if", "counterfactual".
- **CORRELATION**: Look for keywords like "correlation", "association", "relationship", "linked to", "related to". These queries ask about statistical relationships without necessarily implying causality.
- **DESCRIPTIVE**: Queries that ask for summaries, descriptions, trends, or statistics about the data without investigating causal links (e.g., "Show sales over time", "What is the average age?").
- **OTHER**: Use if the query does not fit any of the above categories.

Choose the most appropriate type from: `EFFECT_ESTIMATION`, `COUNTERFACTUAL`, `CORRELATION`, `DESCRIPTIVE`, `OTHER`.
Variable Roles to Identify:

- treatment: The intervention or variable whose effect is being studied.
- outcome: The result or variable being measured.
- covariates_mentioned: Variables explicitly mentioned to control for or adjust for.
- grouping_vars: Variables identifying specific subgroups for analysis (e.g., "for men", "in the sales department").
- instruments_mentioned: Variables explicitly mentioned as potential instruments.

Constraints: Conditions applied to the analysis (e.g., filters on columns, specific time periods).
Dataset Path Mentioned: Extract the file path or URL if explicitly stated in the query.

Output: ONLY a valid JSON object matching this schema (no explanations or surrounding text):

```
{
  "query_type": "<Identified Query Type>",
  "variables": {
    "treatment": ["<Treatment Variable(s) Mentioned>"],
    "outcome": ["<Outcome Variable(s) Mentioned>"],
    "covariates_mentioned": ["<Covariate(s) Mentioned>"],
    "grouping_vars": ["<Grouping Variable(s) Mentioned>"],
    "instruments_mentioned": ["<Instrument(s) Mentioned>"]
  },
  "constraints": ["<Constraint 1>", "<Constraint 2>"],
  "dataset_path_mentioned": "<Path Mentioned or null>"
}
```

If Dataset Context is provided, ensure variable names in the output JSON correspond to actual column names where possible. If no context is provided, or if a mentioned variable doesn't map directly, use the phrasing from the query. Respond with only the JSON object.

H.2 Light Weight Inline Prompts for Dataset Analysis

Prompt:
You are an expert causal inference data scientist. Identify potential treatment and outcome variables from this dataset.
Dataset Description: `{description_text}`
Dataset Columns: `{columns_list}`
Column Types: `{column_types}`
Binary Columns (good treatment candidates): `{binary_cols}`

Instructions:

1. Identify **TREATMENT** variables: interventions, treatments, programs, policies, or binary state changes. Look for binary variables or names with "treatment", "intervention", "program", "policy", etc.
2. Identify **OUTCOME** variables: results, effects, or responses to treatments. Look for numeric variables (especially non-binary) or names with "outcome", "result", "effect", "score", etc.

Output: ONLY a valid JSON object with two lists:

```
{
  "potential_treatments": ["treatment_a", "program_b"],
  "potential_outcomes": ["result_score", "outcome_measure"]
}
```

H.3 Query Interpretation Prompts

Instrumental Variable Identification Prompt

Prompt:
You are a causal inference assistant tasked with assessing whether a valid Instrumental Variable (IV) exists in the dataset. A valid IV must satisfy **all** of the following conditions:

1. **Relevance**: It must causally influence the Treatment.
2. **Exclusion Restriction**: It must affect the Outcome only through the Treatment — not directly or indirectly via other paths.
3. **Independence**: It must be as good as randomly assigned with respect to any unobserved confounders affecting the Outcome.
4. **Compliance (for RCTs)**: If the dataset comes from a randomized controlled trial or experiment, IVs are only valid if compliance data is available i.e., if some units did not follow their assigned treatment. In this case, the random assignment may be a valid IV, and compliance is the actual treatment variable. If compliance related variable is not available, do not select IV.
5. The instrument must be one of the listed dataset columns (not the treatment itself), and must not be assumed or invented.

You should **only suggest an IV if you are confident that all the conditions are satisfied**. Otherwise, return `"NULL"`.

Information Provided:

- User Query: `{query}`
- Dataset Description: `{description}`
- Treatment: `{treatment}`
- Outcome: `{outcome}`

- Available Columns: {column_info}

Output: Return a JSON object with the following structure:

```
{ "instrument_variable": "COLUMN_NAME_OR_NULL" }
```

RDD Identification Prompt

Prompt:
You are an expert causal inference assistant helping to determine if Regression Discontinuity Design (RDD) is applicable for quasi-experimental analysis.

Information Provided:

- User Query: {query}
- Dataset Description: {description}
- Identified Treatment (tentative): {treatment}
- Identified Outcome (tentative): {outcome}
- Available Columns: {column_info}

Task: Your goal is to check if there is a **Running Variable**, i.e., a variable that determines treatment/control.

- If the variable is above a certain cutoff, the unit is categorized as treated; if below, it is control.
- The running variable must be numeric and continuous. Do not use categorical or low-cardinality variables.
- The treatment variable must be binary. If not, RDD is not valid.

Output: Respond ONLY with a valid JSON object. If RDD is not suggested by the context, return null for both fields.

```
{ "running_variable": "COLUMN_NAME_OR_NULL", "cutoff_value": NUMERIC_VALUE_OR_NULL }
```

Examples:

```
{ "running_variable": "test_score", "cutoff_value": 70 }
{ "running_variable": null, "cutoff_value": null }
```

RCT Identification Prompt

Prompt:
You are an expert causal inference assistant helping to determine if the data comes from a Randomized Controlled Trial (RCT). Your goal is to assess if the treatment assignment mechanism described or implied was random.

Information Provided:

- User Query: {query}
- Dataset Description: {description}
- Identified Treatment (tentative): {treatment}
- Identified Outcome (tentative): {outcome}
- Available Columns: {column_info}

Output: Respond ONLY with a valid JSON object matching the required schema.

```
{ "is_rct": BOOLEAN_OR_NULL }
```

Examples:

```
{ "is_rct": true }   # RCT likely
{ "is_rct": false }  # Observational likely
{ "is_rct": null }   # Unsure
```

Treatment Reference Identification Prompt

Prompt:
You are a causal inference assistant.

Dataset Information:

- Dataset Description: `{description}`
- Identified Treatment Variable: `"{treatment_variable}"`
- Unique Values in Treatment Variable (sample): `{treatment_variable_values}`
- User Query: `{query}`

Task: Based on the user query, determine if it specifies a particular category of the treatment variable `{treatment_variable}` that should be considered the control, baseline, or reference group for comparison.
Examples:

- Query: "Effect of DrugA vs Placebo" → Reference for treatment "Drug" = "Placebo"
- Query: "Compare ActiveLearning and StandardMethod against NoIntervention" → Reference for "TeachingMethod" = "NoIntervention"

If a reference level is clearly specified or strongly implied **and** it is one of the unique values provided, identify it. Otherwise, state `null`. If multiple values seem like controls (e.g., "compare A and B vs C and D"), return `null` for now.

Output: Return ONLY a JSON object adhering to this schema:

```
{
    "reference_level": "string_representing_the_level_or_null",
    "reasoning": "string_or_null_brief_explanation"
}
```

Interaction Term Identification Prompt

Prompt:
You are a causal inference assistant.
Your task is to determine whether the user query suggests the inclusion of an interaction term between the treatment and one covariate, specifically to assess heterogeneous treatment effects (HTE).

Information Provided:

- User Query: `{query}`
- Dataset Description: `{description}`
- Identified Treatment Variable: `{treatment_variable}`
- Available Covariates (name: type): `{covariates_list_with_types}`

Instructions:

- ONLY suggest an interaction if the query explicitly mentions treatment across a subgroup.

- DO NOT suggest an interaction if the query asks for an overall average effect or does not mention subgroup analysis.
- If unsure, default to no interaction.

Output Schema:

```
{
    "interaction_needed": boolean,
    "interaction_variable": string_or_null,
    "reasoning": string
}
```

Examples:

```
{
  "interaction_needed": true,
  "interaction_variable": "gender",
  "reasoning": "Query asks if the treatment effect is for men."
}
```

```
{
  "interaction_needed": false,
  "interaction_variable": null,
  "reasoning": "Query asks for the overall average treatment effect,
               no specific subgroups mentioned for effect heterogeneity."
}
```

Treatment Variable Identification Prompt

Prompt:
You are an expert in causal inference. Your task is to identify the **treatment variable** in a dataset in order to perform a causal analysis that answers the user's query.

Information Provided:

- User Query: `{query}`
- Dataset Description: `{description}`
- List of Available Variables: `{column_info}`

Task: Based on the query, dataset description, and available variables, determine which variable is most likely to serve as the treatment variable.
If a clear treatment variable cannot be determined, return `null`.

Output Schema:

```
{ "treatment": "COLUMN_NAME_OR_NULL" }
```

Outcome Variable Identification Prompt

Prompt:
You are an expert in causal inference. Your task is to identify the **outcome variable** in a dataset in order to perform a causal analysis that answers the user's query.

Information Provided:

- User Query: `{query}`
- Dataset Description: `{description}`
- Available Variables: `{column_info}`

Task: Based on the query, dataset description, and available variables, determine which variable is most likely to serve as the outcome variable in the causal analysis.
Do not speculate. If a clear outcome variable cannot be identified, return `null`.

Output Schema:

```
{ "outcome": "COLUMN_NAME_OR_NULL" }
```

Covariates Identification Prompt

Prompt:
You are an expert in causal inference. Your task is to identify the **pre-treatment variables** in a dataset that can be used as controls in a causal estimation model to answer the user's query.

Information Provided:

- User Query: `{query}`
- Dataset Description: `{description}`
- Available Variables: `{column_info}`
- Treatment Variable: `{treatment}`
- Outcome Variable: `{outcome}`

Task: Pre-treatment variables are those that are measured **before** the treatment is applied and are **not affected** by the treatment. These variables can be used as controls in the causal model.
For example, in an RCT with outcome Y, treatment T, and pre-treatment variables $X1$, $X2$, $X3$, we can perform a regression of the form:

$$Y \sim T + X1 + X2 + X3$$

Based on the information above, return a list of variables that qualify as pre-treatment variables from the available columns. If no suitable pre-treatment variables can be identified, return an empty list.

Output Schema:

```
{ "covariates": ["LIST_OF_COLUMN_NAMES_OR_EMPTY_LIST"] }
```

Estimand Identification Prompt

Prompt:
You are an expert in causal inference. Your task is to determine the appropriate **estimand** to answer a given query.

Information Provided:

- User Query: `{query}`
- Dataset Description: `{dataset_description}`
- Variables in Dataset: `{dataset_columns}`
- Treatment Variable: `{treatment}`
- Outcome Variable: `{outcome}`

Task: Given this information, decide whether the **Average Treatment Effect (ATE)** or the **Average Treatment Effect on the Treated (ATT)** is more appropriate for answering the query.

Output: Only return the estimand name:

```
"att" or "ate"
```

Confounder Identification Prompt

Prompt:
You are an expert in causal inference. Your task is to identify potential **confounders** in a dataset that should be adjusted for when estimating the causal effect described in the user query.

Information Provided:

- User Query: `{query}`
- Dataset Description: `{description}`
- Available Variables: `{column_info}`
- Treatment Variable: `{treatment}`
- Outcome Variable: `{outcome}`

Definition of Confounder: A **confounder** is a variable that:

1. Affects the treatment (influences who receives the treatment),
2. Affects the outcome,
3. Is not caused by the treatment (must be a pre-treatment variable),
4. Is not a mediator between treatment and outcome.

These variables can create spurious associations between treatment and outcome if not adjusted for.

Task: Based on the user query and the dataset description, identify which variables are likely to be confounders. Only include variables that you believe causally affect both treatment and outcome. If uncertain, only include variables where the justification is clear from the query or description.

Output Schema:

```
{ "confounders": ["LIST_OF_COLUMN_NAMES_OR_EMPTY_LIST"] }
```

DiD Term Identification Prompt

Prompt:
You are a causal inference assistant tasked with determining whether a valid Difference-in-Differences (DiD) **interaction term** already exists in the dataset.
This DiD term should be a **binary variable** indicating whether a unit belongs to the **treatment group after treatment was applied**.
For example, if a policy was enacted in 2020 for a particular state, then the DiD term would equal 1 for units from that state in years after 2020, and 0 otherwise.

Information Provided:

- User Query: `{query}`
- Time Variable: `{time_variable}`
- Group Variable: `{group_variable}`
- Dataset Description: `{description}`
- Available Columns: `{column_info}`
- Column Types: `{column_types}`

Output: Return your answer as a valid JSON object with the following format:

```
{ "did_term": "COLUMN_NAME_OR_NULL" }
```

I Detailed Study: Method Validation Loop

This example presents the complete prompt employed in the validation feedback loop of CAIS.

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.

J Baselines Prompts

Here, we display all the prompts used for the baselines: ReAct, PoT, and Veridical Prompts for causal inference.

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`

Program of Thoughts 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

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