Causal AI Scientist: Towards End-to-End Causal Inference with Large Language Models

Sawal Acharya [1 2] **Rahul Babu Shrestha** [1] **Jacob T. Emmerson** [1] **Vishal Verma** [1] **Devansh Bhardwaj** [1]
Samuel Simko [1 3] **Punya Syon Pandey** [1 3] **Anahita Haghighat** [1] **Yongjin Yang** [1] **Dominik Janzing** [4]
Mrinmaya Sachan [5] **Bernhard Schölkopf** [6 7] **Zhijing Jin** [1 3 6]

Abstract

Causal inference is a cornerstone of empirical research, yet conducting rigorous analysis on tabular datasets requires substantial methodological expertise to navigate complex identification assumptions. While current agentic frameworks aim to lower this barrier, they often require users to manually specify methods or variables, or they support only a narrow set of techniques with limited validation on real-world data. To address these limitations, we present Causal AI Scientist (CAIS), an autonomous agent designed for end-to-end causal inference. Given a dataset and its description, CAIS leverages Large Language Models (LLMs) to interpret metadata and navigate a structured decision tree guided by causal inference principles to select methods. To ensure reliability, CAIS implements a validation feedback loop that performs post-selection empirical diagnostics, allowing the agent to verify assumptions and self-correct method choices before implementation. We evaluate CAIS on CauSciBench, a rigorous benchmark comprising synthetic and real-world empirical studies, designed to evaluate the real-world causal estimation capabilities of LLMs. Our results demonstrate that while specific frontier models can excel with alternative prompting, CAIS achieves the best average performance across model families, including GPT-4o and GPT-4o mini. More broadly, CAIS showcases how agentic frameworks can democratize expert-level causal analysis by automating the path from natural language queries to rigorous causal estimates.

[1]Jinesis Lab, University of Toronto & Vector Institute [2]Stanford University [3]EuroSafeAI [4]Amazon, Tübingen [5]ETH Zürich [6]Max Planck Institute for Intelligent Systems, Tübingen, Germany [7]ELLIS Institute, Tübingen. Correspondence to: Zhijing Jin <zjin@cs.toronto.edu>.

Proceedings of the 43rd International Conference on Machine Learning, Seoul, South Korea. PMLR 306, 2026. Copyright 2026 by the author(s).

1. Introduction

Causal inference aims to quantify the effect of a treatment or intervention on outcomes of interest (Pearl, 2009a; Imbens, 2024). Understanding cause-and-effect relationships is central to empirical research and evidence-based decision-making across disciplines, including the social sciences (Imbens, 2024), public health (Glass et al., 2013), and biomedicine (Kleinberg & Hripcsak, 2011). Conducting causal analysis on tabular datasets requires substantial methodological expertise. Practitioners apply domain knowledge to select appropriate estimands, justify identification assumptions, choose suitable inference methods, and conduct sensitivity analyses. This need for methodological expertise limits accessibility for users who could otherwise benefit from causal inference.

Large language models (LLMs) (Vaswani et al., 2017; Brown et al., 2020; Zhao et al., 2023) have the potential to overcome this expertise barrier. Given the amount of information the models learn during training, one can use them as domain experts to set up and implement causal analyses (Kiciman et al., 2024; Ma, 2025). Towards this goal, several agentic frameworks have been proposed, including Causal-Copilot (Wang et al., 2025), Causal Agent (Han et al., 2024), Econometrics AI Agent (Chen et al., 2025), and Causal Modeling Agents (CMA) (Abdulaal et al., 2024).

However, the aforementioned approaches face limitations. Causal Copilot, CMA, and Causal Agent focus mainly on causal discovery tasks, where the goal is to learn a causal graph from observational data. While one can use backdoor and frontdoor adjustments (Pearl, 2009b) for causal effect estimation, these graph-based techniques do not support methods based on the potential outcomes framework, such as difference-in-differences (DiD) and regression discontinuity design (RDD). Econometric AI Agent supports potential outcomes-based methods, but requires specification of the exact inference method and/or variables and therefore does not support end-to-end estimation. LLM4Causal enables

end-to-end estimation, but it focuses on only two causal effect measures and is evaluated on synthetic examples only. CausalFM enables instrumental-variable-based estimation but does not support quasi-experimental methods such as DiD and RDD. Moreover, these are not general-purpose tools for automating end-to-end causal analysis from natural language queries.

To address these limitations, we present Causal AI Scientist (CAIS), a general-purpose tool for end-to-end causal analysis starting from natural language inputs. Users provide three inputs: a description of the data, the dataset itself, and a causal query in natural language. CAIS then formulates the causal estimation problem by selecting appropriate variables and identifying suitable estimation methods. We employ frontier foundation models to assess key characteristics of the data variables, along with their descriptions. Method selection follows a structured decision tree that evaluates characteristics, such as whether the study is a randomized trial and whether candidate instruments are available. This approach decomposes method selection into interpretable steps, providing a clear rationale for the choice of a particular method. Additionally, CAIS performs diagnostic checks to validate the appropriateness of the selected method and incorporates a feedback loop to reassess method choice when issues are detected.

We evaluate CAIS on CauSciBench (Acharya et al., 2025), a benchmark designed to assess the ability of LLMs to perform causal estimation on tabular datasets. Experiments on its three subsets show that CAIS achieves the best performance in method selection and effect accuracy when results are aggregated across the foundation models.

In summary, our contributions are:

- We propose Causal AI Scientist (CAIS), an end-to-end tool for performing causality-driven data analysis from natural language queries using the potential outcomes framework.
- CAIS uses frontier language models as domain experts to assess data characteristics and identify plausible methods, then applies a novel decision tree based on causal inference principles to guide method selection.
- We evaluate CAIS on examples curated from real-world studies and perform ablation studies to demonstrate the effectiveness of our approach.

2. Problem Formulation

We consider scenarios where a system must perform causal effect estimation in the process of answering (causal) natural language queries[1]. The inputs are:

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

The objective is to interpret q and the metadata to select an appropriate causal inference method, identify relevant variables (treatment, outcome, and covariates or instruments), and estimate the causal effect with uncertainty measures.

3. Methodology

CAIS consists of four methodological stages, each composed of one or more micro-tools. Every stage combines logic from established causal inference principles with LLM reasoning. The overall framework is depicted in Figure 1. Detailed descriptions of the specific tools used in each stage are provided in Appendix C. The metrics used for evaluation are provided in Appendix A.4.

Outline Our method consists of the following stages:

- **Stage 1 (Data Preprocessing & Query Decomposition)** The process begins by analyzing the dataset to identify key components, such as treatment and outcome 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 standard assumptions for the selected method are then assessed. If any assumption check fails, the system backtracks to Stage 2 to find an alternative method, creating a validation loop.
- **Stage 4 (Method Execution & Interpretation)** Finally, once all checks pass, the chosen method is executed using predefined templates, and the final result is returned with an interpretation.

Dataset To evaluate CAIS, we use CauSciBench (Acharya et al., 2025), a comprehensive benchmark for assessing LLMs on causal estimation tasks. CauSciBench consists of causal queries drawn from three sources: (1) QRData (Liu et al., 2024), a benchmark dataset primarily based on causal inference textbooks; (2) published papers across multiple

[1]Example query phrasings include: "Does X increase Y?", "What is the effect of X on Y?", and "How does X affect Y?", where X and Y denote the quantities of interest.

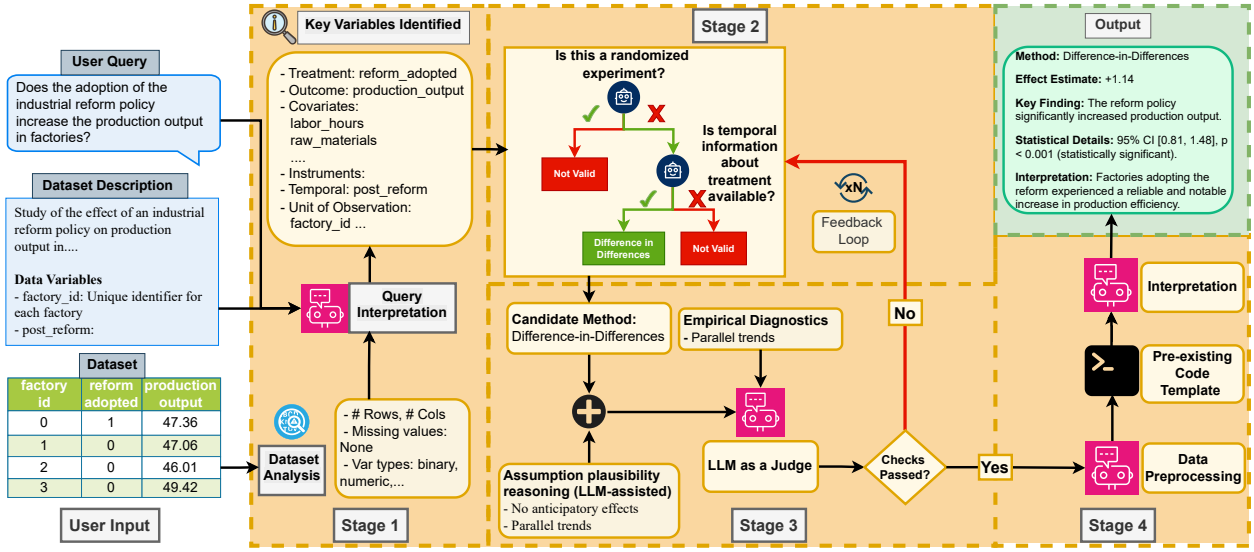

Figure 1. **CAIS workflow.** The user provides an input dataset (CSV file), its description, and an associated causal query. Guided by a decision tree and a backbone LLM, CAIS selects an appropriate estimation method, executes the code, and returns the estimated causal effect along with a natural language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

disciplines; and (3) synthetic datasets with known ground-truth causal effects. This collection covers a wide range of evaluation scenarios and methods used in practice.

4. Results

We evaluate the performance of CAIS (Section 4.1) against the 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). We then analyze the effectiveness of its key components, the decision tree and the validation feedback loop, through ablation studies (Section 4.2), followed by a detailed error analysis (Appendix A.5).

4.1. Overall Performance

Table 1 presents the main results, comparing CAIS with the baselines on three subsets of CauSciBench.

Performance on real-world studies On real-world data, CAIS achieves the best method selection performance across two models: GPT-4o (51.61%) and GPT-4o-mini (34.84%), outperforming baseline prompting strategies. However, the overall best method accuracy on real data is achieved by Veridical prompting with o3-mini (60.00%). CAIS comes in second with a score of 53.55%. We attribute the performance of Veridical prompting to its reasoning approach, which requires models to reflect thoroughly on both the inputs and decision choices. This emphasis on self-reflection, combined with o3-mini's strong reasoning capabilities, appears particularly effective for real-world method selection where models must reason about untestable assumptions and confounding. Moreover, the effect accuracy metric depends on the method. Different model specifications from the reference are likely to yield higher errors. As such, Veridical achieves the best score in the effect accuracy metric. However, if we consider the average performance across all models, CAIS achieves the best results for both method selection and effect accuracy.

Performance on synthetic and textbook data As with real data, CAIS achieves the best aggregate performance in terms of method and effect accuracy for synthetic data. Similarly, Veridical prompting with o3-mini achieves the best score in terms of method selection, along with ReAct + GPT-4o. Both achieve a score of 62.2%.

4.2. Ablation Studies

We conduct two ablation studies to assess the role of two key components of our method selection process: the decision tree and the validation feedback loop.

For the decision tree ablation, we replace the decision-tree-based method selection with zero-shot prompting. We prompt an LLM to select a suitable method to answer the query based on the input dataset and its description. For the validation loop ablation, we remove the validation-based feedback loop and proceed with the method selected by the tree-based scheme.

Decision tree significantly improves method-selection accuracy. The results in Table 2 show that, in general, incor-

	Method Accuracy (↑)				Effect Accuracy (↑)				CI Coverage 95 (↑)			
Method	**Overall**	**GPT-4o**	**GPT-4o-mini**	**o3-mini**	**Overall**	**GPT-4o**	**GPT-4o-mini**	**o3-mini**	**Overall**	**GPT-4o**	**GPT-4o-mini**	**o3-mini**
						Textbook Data						
ReAct	**94.02**	**94.87**	**92.31**	94.87	58.11	59.66	53.58	61.11	58.97	43.59	**64.10**	69.23
PoT	90.60	89.74	87.18	94.87	**63.24**	**61.21**	**54.37**	**74.15**	**74.36**	**69.23**	**64.10**	**89.74**
Veridical	92.31	92.31	84.62	**100.00**	54.47	53.46	42.21	67.73	12.82	7.69	0.00	30.77
CAIS	65.81	51.28	71.79	74.36	29.18	13.68	41.13	32.74	20.51	10.26	28.21	23.08
						Synthetic Data						
ReAct	46.62	**62.24**	25.87	51.75	35.29	42.00	19.56	44.31	**31.70**	**39.86**	15.38	**39.86**
PoT	32.87	49.65	22.38	26.57	27.34	35.92	20.73	25.39	24.48	30.07	19.58	23.78
Veridical	46.62	58.74	18.88	**62.24**	37.21	41.98	11.73	**57.92**	15.85	26.57	3.50	17.48
CAIS	**47.32**	51.75	**50.35**	39.86	**45.44**	**49.59**	**48.26**	38.48	29.60	32.17	**30.77**	25.87
						Real Data						
ReAct	37.63	47.74	25.81	39.35	9.44	14.28	2.96	11.07	11.18	**17.42**	5.16	10.97
PoT	38.28	43.87	32.26	38.71	13.12	7.08	**13.22**	19.07	**18.49**	15.48	**18.71**	**21.29**
Veridical	41.08	42.58	20.65	**60.00**	15.19	12.50	5.55	**27.51**	8.39	9.03	2.58	13.55
CAIS	**46.67**	**51.61**	**34.84**	53.55	**18.18**	**20.03**	12.39	22.11	16.77	16.77	13.55	20.00

Table 1 Performance of CAIS and baseline prompting strategies across datasets and LLMs. Metrics reported are **Method Accuracy**, **Effect Accuracy**, and **95% CI Coverage**. Bold values indicate the best-performing *method* for each metric within each dataset. The **Overall** column represents aggregate performance across all models.

porating a structured decision tree for method selection leads to 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 synthetic data, GPT-4o-mini improves from 37.8% to 50.3% and o3-mini from 36.4% to 39.9%.

The validation loop helps reduce the performance gap between weaker LLMs and stronger LLMs. As shown in Table 2, 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 23.1% to 71.8% (+48.7 points) on QRData and from 19.4% to 34.8% (+15.4 points) on the Real dataset. The loop provides this significant boost by allowing weaker models to recover from incorrect initial method selections, which they are more prone to making.

Dataset Collection	Model	Method Selection		Validation Loop	
		CAIS	Δ	CAIS	Δ
QR	GPT-4o	84.62	(-7.69)	51.30	(-30.80)
	GPT-4o-mini	74.36	(+2.56)	71.80	(+48.70)
	o3-mini	74.40	(+2.60)	74.40	(+5.10)
Real	GPT-4o	57.58	(-18.18)	51.60	(+15.50)
	GPT-4o-mini	69.70	(-6.06)	34.80	(+15.50)
	o3-mini	53.50	(+29.00)	53.50	(+18.10)
Synth	GPT-4o	57.78	(+2.22)	51.70	(-9.80)
	GPT-4o-mini	42.22	(-4.44)	50.30	(+0.00)
	o3-mini	39.90	(+3.50)	39.90	(+21.70)

Table 2 Ablation study evaluating the decision tree (method acc.) and validation loop (effect acc.). Δ = CAIS − Ablated.

5. Conclusion

In this work, we introduce Causal AI Scientist (CAIS), an end-to-end tool for automating the causal inference process. CAIS takes a dataset, its description, and a corresponding natural language query as input. Using frontier foundation models as the backbone LLM, it analyzes the relevant characteristics and employs a decision tree based on these analyzed characteristics to select an appropriate inference method before implementing it. While alternative methods may perform better with specific models, CAIS achieves the best performance in method selection and effect accuracy when results are aggregated across models, highlighting its ability to work with a range of models. Moreover, the decision tree approach enhances interpretability as users are provided with clear reasoning for why a certain method was selected. This transparency is important in the context of causal inference, where ground truths are unknown, and there are no definitive metrics to optimize for.

Impact Statement

This paper presents work aimed at advancing the field of AI-enhanced causal inference, which facilitates empirical, data-driven research across many disciplines. Currently, we don't see any potentially harmful social implications from this work.

References

Abdulaal, A., adamos hadjivasiliou, Montana-Brown, N., He, T., Ijishakin, A., Drobnjak, I., Castro, D. C., and Alexander, D. C. Causal modelling agents: Causal graph discovery through synergising metadata- and data-driven reasoning. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=pAoqRlTBtY.

Acharya, S., Zhang, T. J., Kim, A., Haghighat, A., Sun, X., Mordig, M., Shrestha, R. B., Danisman, F., Jose, C., Qi, Y., Cobben, P., Schölkopf, B., Sachan, M., and Jin, Z. Causcibench: Assessing llm causal reasoning for scientific research. 2025. URL https://zhijing-jin.com/files/papers/2025_CauSciBench.pdf.

Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33: 1877–1901, 2020.

Card, D. 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.

Card, D. and Krueger, A. B. 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.

Carpenter, C. and Dobkin, C. 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.

Chen, Q., Han, T., Li, J., Luo, Y., Wu, Y., Zhang, X., and Zhou, T. Can ai master econometrics? evidence from econometrics ai agent on expert-level tasks, 2025. URL https://arxiv.org/abs/2506.00856.

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

Glass, T., Goodman, S., Hernán, M., and Samet, J. 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.

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

Han, K., Kuang, K., Zhao, Z., Ye, J., and Wu, F. Causal agent based on large language model, 2024. URL https://arxiv.org/abs/2408.06849.

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

Holland, P. W. 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.

Imbens, G. W. Instrumental variables: An econometrician's perspective. *Statistical Science*, 29(3):323–358, 2014. ISSN 08834237, 21688745. URL http://www.jstor.org/stable/43288511.

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

Kiciman, E., Ness, R., Sharma, A., and Tan, C. 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.

Kleinberg, S. and Hripcsak, G. 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.

Liu, X., Wu, Z., Wu, X., Lu, P., Chang, K.-W., and Feng, Y. Are LLMs capable of data-based statistical and causal reasoning? benchmarking advanced quantitative reasoning with data. In Ku, L.-W., Martins, A., and Srikumar, V. (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.

Ma, J. Causal inference with large language model: A survey. In Chiruzzo, L., Ritter, A., and Wang, L. (eds.), *Findings of the Association for Computational Linguistics: NAACL 2025*, pp. 5886–5898, Albuquerque, New Mexico, April 2025. Association for Computational Linguistics. ISBN 979-8-89176-195-7. doi: 10.18653/v1/2025.

findings-naacl.327. URL https://aclanthology.org/2025.findings-naacl.327/.

Miguel, E. and Kremer, M. 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.

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

Pearl, J. *Causality: Models, Reasoning and Inference*. Cambridge University Press, USA, 2nd edition, 2009a. ISBN 052189560X.

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

Rosenbaum, P. R. and Rubin, D. B. 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.

Rubin, D. B. 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.

Stuart, E. A. 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.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. Attention is all you need. In *Advances in neural information processing systems*, pp. 5998–6008, 2017.

Wang, X., Zhou, K., Wu, W., Singh, H. S., Nan, F., Jin, S., Philip, A., Patnaik, S., Zhu, H., Singh, S., Prashant, P., Shen, Q., and Huang, B. Causal-copilot: An autonomous causal analysis agent, 2025. URL https://arxiv.org/abs/2504.13263.

Yao, S., Zhao, J., Yu, D., Du, N., Shafran, I., Narasimhan, K., and Cao, Y. React: Synergizing reasoning and acting in language models, 2023. URL https://arxiv.org/abs/2210.03629.

Yu, B. 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.

Zhao, W. X., Zhou, K., Li, J., Tang, T., Wang, X., Hou, Y., Min, Y., Zhang, B., Zhang, J., Dong, Z., et al. A survey of large language models. *arXiv preprint arXiv:2303.18223*, 2023.

A. Additional Analysis

A.1. Identification Strategy

We adopt the potential outcomes framework (Rubin, 2005). The system parses query q and metadata to identify the treatment (T) and outcome (Y). For the Average Treatment Effect (ATE):

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

For identification, we make two key assumptions:

1. **Conditional Ignorability** states that if all the confounders, X, are observed then,

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

   If the data is from a randomized trial, we have strong ignorability: $(Y(1), Y(0)) \perp\!\!\!\perp T$

2. **SUTVA** (Stable Unit Treatment Value Assumption) states that (1) the observed outcome corresponds to the potential outcome under the assigned treatment, and (2) the potential outcome of unit i is unaffected by the treatment status of other units

In randomized settings with ignorability and SUTVA, the ATE is given by the difference in means across treated and control groups, i.e.,

$$\hat{\tau}_{ATE} = \frac{1}{n_1} \sum_{i:T_i=1} Y_i - \frac{1}{n_0} \sum_{i:T_i=0} Y_i \quad n_1 = \sum_i \mathbf{1}[T_i = 1] \text{ and } n_0 = \sum_i \mathbf{1}[T_i = 0] \tag{3}$$

For observational data, under conditional ignorability and SUTVA, the ATE can be identified as,

$$\tau_{ATE} = \mathbb{E}_X[\mathbb{E}[Y \mid T = 1, X] - \mathbb{E}[Y \mid T = 0, X]] \tag{4}$$

This can be estimated via regression adjustment or other methods, such as matching or Inverse Probability Weight (IPW).

A.2. The Role of LLMs

The conditional ignorability assumption is untestable. To justify its use, we must argue that the dataset contains all relevant confounders, and this is typically done using domain knowledge. While statistical tests can provide supporting evidence (covariate balance checks, sensitivity analysis), the foundational justification relies on domain knowledge. LLMs can serve as proxies for domain experts in this step.

For exposition, we have focused on estimating ATEs from observational data. However, domain knowledge justifications are employed for other causal inference methods as well. For example, when using instrumental variables, one relies on domain expertise to argue why a candidate variable reasonably satisfies the key assumptions: exclusion restriction, relevance, independence, and monotonicity. These assumptions are largely untestable, and their justification involves domain-based arguments supplemented by statistical tests, such as the F-test for relevance.

A.3. Baselines

Given the lack of LLM-based tools designed for fully automated causal inference, we compare our method against three strong baseline prompting strategies. We provide concrete examples of the prompts in Appendix H.

ReAct Prompting The ReAct framework (Yao et al., 2023) prompts the LLM to work through a problem in iterative thought–action–observation cycles. For causal analysis, the LLM first reasons about the problem, then generates and executes a relevant code snippet (the action), and finally observes the output to inform its next thought. We include this strong baseline because prior work found that ReAct achieves the best overall performance among several prompting strategies for causal inference (Liu et al., 2024).

PoT Prompting In contrast to ReAct's iterative approach, Program of Thought (PoT) prompting (Chen et al., 2022) instructs the LLM to generate a single, complete program to conduct the entire causal analysis. This program includes all steps for data loading, variable identification, executing the chosen causal method, and reporting the final results.

Veridical Data Science Prompting This baseline is inspired by the Veridical Data Science framework (Yu, 2020), which emphasizes stability and predictability in analysis. We designed a prompt that requires the LLM to engage in self-reflection. After making a key decision (e.g., selecting a method), the LLM is prompted to critique its own choice and reconsider the output, with the goal of improving the stability of its reasoning.

A.4. Evaluation Metrics

Notations $\hat{m}_i$ is the preferred causal inference method and $\hat{m}$ is the method selected by CAIS and baseline models. Likewise, τ is the reference causal effect and $\hat{\tau}$ is the estimated effect. Similarly, se_i denotes the reference standard error.

Metrics We evaluate our pipeline using the following metrics:

1. **Method Accuracy (MA)** rewards accurate effect estimates when the correct method is chosen, and assigns zero credit to cases with incorrect method selection.

$$\text{MA} = \mathbb{E}\left[\mathbf{1}[\hat{m}_i = m_i]\right], \qquad r_i = \min\left(\frac{|\hat{\tau}_i - \tau_i|}{|\tau_i|}, 1\right).$$

   where $\hat{m}_i$ is the preferred method and $\hat{m}$ is the predicted method.

2. **Effect Accuracy (EffAcc)** rewards accurate effect estimates when the correct method is chosen, and assigns zero credit to incorrect method selection.

$$\text{EffAcc} = \mathbb{E}[\mathbf{1}[\hat{m}_i = m_i] \cdot (1 - r_i)], \qquad r_i = \min\left(\frac{|\hat{\tau}_i - \tau_i|}{|\tau_i|}, 1\right).$$

3. **Confidence Interval Coverage (Cov)** rewards estimates if they fall within the 95% confidence interval of the reference values.

$$\text{Cov} = \frac{1}{N}\sum_{i=1}^{N} \mathbf{1}\left[\hat{\tau}_i \in [\tau_i - 1.96 \cdot se_i, \tau_i + 1.96 \cdot se_i]\right]$$

A.5. 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. A breakdown of baseline error rates is provided in Appendix F.

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.

Wrong Method Selection: LLMs misclassify Randomized Controlled Trials as Encouragement Designs, leading to the selection of Instrumental Variables instead of linear regression with control variables or difference-in-means. Similarly, for synthetic datasets, the model fails to identify Instrumental Variables as the optimal method. 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.

Model	Success (%)	Runtime Err. (%)	Validation Err. (%)	Missing Var. (%)	Key/Data Err. (%)
GPT-4o-mini	79.5	20.5	0.0	0.0	17.9
GPT-4o	56.4	43.6	2.6	2.6	41.0
o3-mini	74.4	25.6	2.6	5.1	25.6
All Models	70.1	29.9	1.7	2.6	28.2

Table 3 Error analysis of CAIS across QRData queries. A query is considered successful if it completes without runtime exceptions. Runtime errors include uncaught exceptions and tracebacks. Key/Data errors correspond to missing fields (e.g., `KeyError`, missing results), while missing-variable errors arise from absent treatment or outcome identification. Percentages are computed over total queries per model.

B. Explanation of decision tree

B.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)

B.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.

B.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 & 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.

B.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.

B.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.

B.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 & 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 & 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.

B.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 & 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.

B.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 ($\text{Cov}(Z, T) \neq 0$).
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.

B.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.

B.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 & Imbens, 2004). The scores are computed using the backdoor adjustment set.

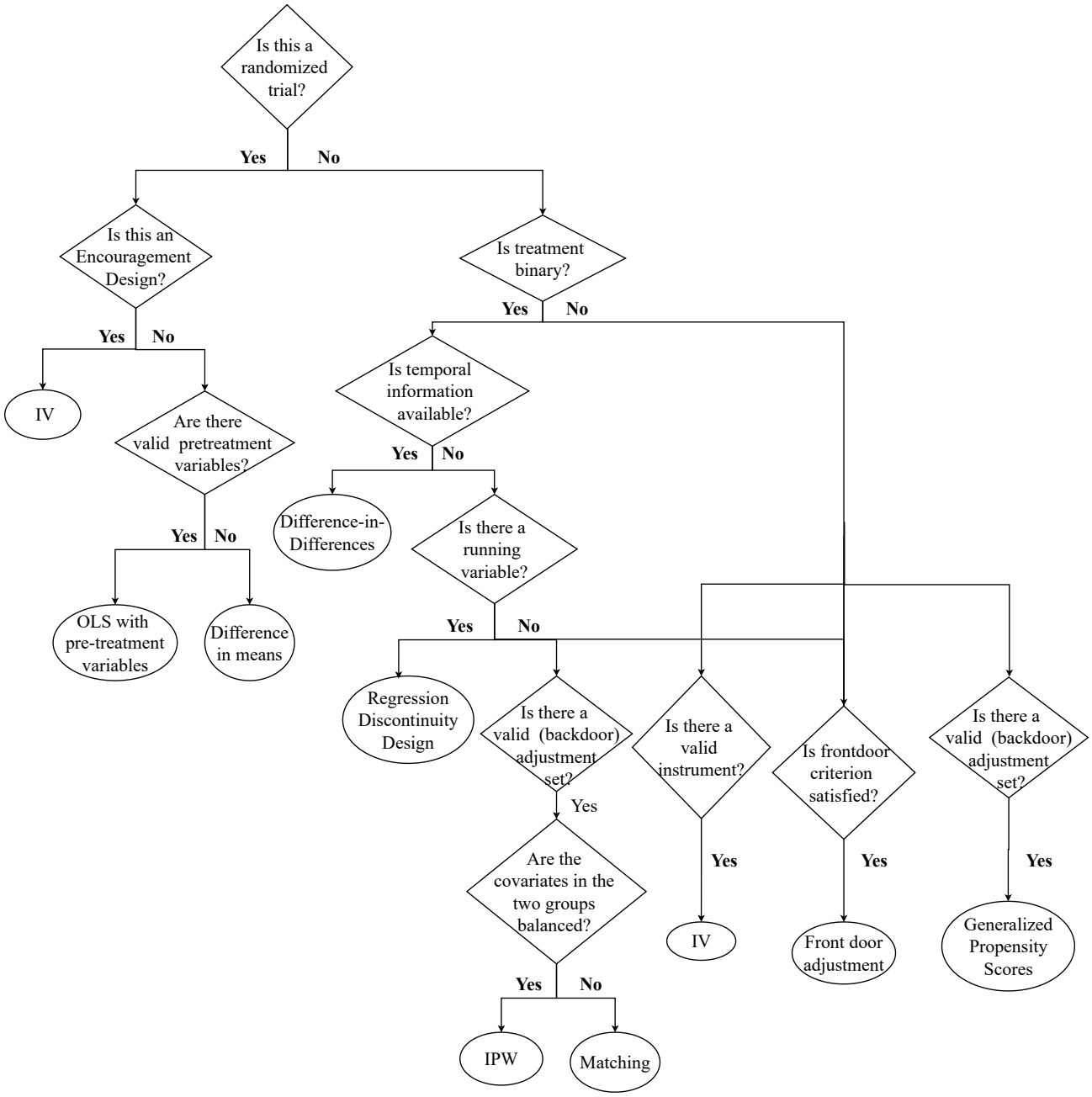

Figure 2. Decision-tree that guides method selection in CAIS. We prompt an LLM to generate responses to queries corresponding to the decision nodes, and traverse the tree accordingly before reaching a leaf node, which corresponds a method.

C. Detailed Methodology

In this section we extend the Methodology of CAIS discussed in Section 3 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).

C.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 E.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 E.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 E.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.

C.2. Stage 2: Method Selection

Given the analysis plan, the **method_selector** chooses a candidate estimator via a 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 completely deterministic and requires no involvement of LLMs directly. However, it uses the information extracted by the LLM in Stage 1. The design choice behind the nodes of the decision tree is already discussed in great detail in Appendix B.

C.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.

C.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). Ex-

ecution 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.

D. Additional Figures

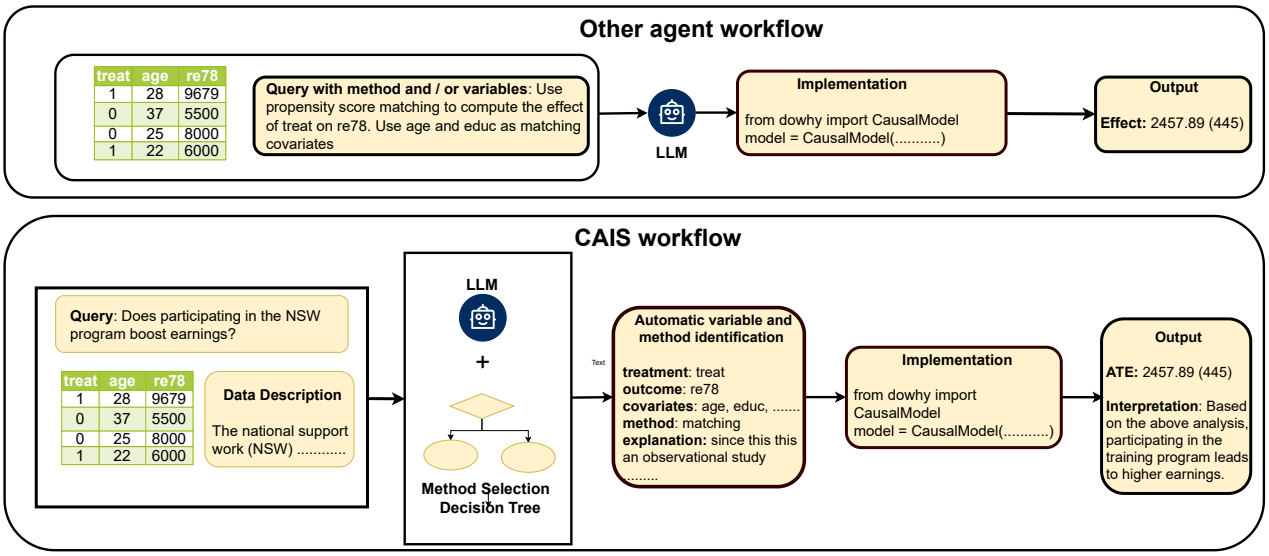

Figure 3. Comparison of the workflows on CAIS with other agents for causal effect estimation, including Econometrics AI Agent and Causal Agent

E. Methodology Prompts

E.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.

E.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"]
}
```

E.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`}

Method Family	Total Queries	Error Rate (%)	Missing Method (%)	Missing Effect (%)
ReAct	546	14.29	6.23	8.06
Veridical	546	4.76	1.83	2.93
PoT-M	546	2.20	0.18	2.01

Table 4 Global baseline error rates aggregated across all models and dataset splits. Error rates reflect the fraction of queries that failed to produce a valid scalar causal estimate. Errors are decomposed into missing method selection and missing effect estimation.

- Column Types: {`column_types`}

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

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

F. Baseline Error Analysis

Table 4 shows that among baseline method families, Program-of-Thoughts achieves the lowest overall error rate (2.20%), followed by Veridical (4.76%), whereas ReAct exhibits substantially higher errors (14.29%), primarily due to failures in method selection (6.23%) and effect estimation (8.06%).

G. 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.

H. 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 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

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