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

Anonymous ACL submission

Abstract

The end-to-end process of running a causal analysis requires knowledge about a wide range of estimation methods, statistical assumptions, and a technical understanding of the phenomena of interest. Recent advances in large language models (LLMs) can circumvent the need for expert knowledge by automating the inference pipeline, thereby widening the accessibility to causal inference tools. In this work, we present Causal AI Assistant (CAIA), an end-to-end pipeline for performing causal analysis. By implementing a method selection pipeline using a tree-of-thoughts-inspired approach, our pipeline leverages LLM's reasoning capabilities to select and execute appropriate inference methods to generate data-driven answers to natural language causal queries. Furthermore, we test our pipeline on preexisting datasets in addition to synthetic examples and datasets drawn from published social science studies. We show through extensive evaluation that our pipeline approach outperforms existing work in automated causal inference.

1 Introduction

Recent advances in large language models (LLMs) offer a promising avenue for enhancing causal inference, including automating the estimation of causal effects. LLMs facilitate automation in several ways. First, their knowledge can be used to construct causal graphs relevant to a given phenomenon (Kiciman et al., 2024; Vashishtha et al., 2023). These graphs can help estimate causal effects between variables of interest (Pearl, 2009). Second and more relevant to our work, LLMs facilitate causal data analysis by assisting in implementing econometric and statistical methods to datasets of interest (Liu et al., 2024c; Ji et al., 2025).

Current works leveraging LLMs for data-driven causal analysis focus on settings where the users specify the estimand/method, and the LLM handles the implementation (Liu et al., 2024a). However, choosing the appropriate estimand/method is often the most challenging step in the causal inference pipeline, where experts draw upon their knowledge about a wide range of techniques, the data-generation process, and the underlying phenomena. To address this bottleneck, Jiang et al. (2024) proposed LLM4Causal, a foundation model fine-tuned to perform end-to-end causal inference. While promising, for causal effect estimation, LLM4Causal has mainly been evaluated on tasks involving the estimation of the Average Treatment Effect (ATE) and Heterogeneous Treatment Effect (HTE), leaving out a wide range of other estimands and methods unexplored. Furthermore, its performance is largely tested on synthetic datasets, which may not capture the complexity of real-world scenarios.

More recently, Wang et al. (2025) introduced Causal-Copilot, a system designed to automate the causal inference pipeline. However, their framework does not support widely-used econometric methods such as Difference-in-Differences (DiD) and Regression Discontinuity Design (RDD), which are central to empirical research in the social sciences. Additionally, their evaluation has focused primarily on causal discovery tasks, rather than causal effect estimation.

To evaluate the ability of LLMs to estimate causal effects from real-world datasets, it is crucial to evaluate their performance across a broader set of methods and scenarios. Toward this goal, we introduce **CausalAI Assistant** (**CAIA**), an end-to-end pipeline that supports causal analysis for a diverse range of social science contexts. Given a dataset, its description, and a natural language query, CausalAI Assistant uses LLMs to automatically identify and execute the most appropriate inference method, then uses the resulting estimates to address the user's query.

At the core of our pipeline is the **Tree of Thoughts (ToT)** prompting framework (Yao et al., 2023; Long, 2023). At each node, the LLM is prompted to assess specific features of the data, and this structured reasoning guides method selection. This approach not only simplifies method selection but also enhances the interpretability of the process. To assess the practical validity of our pipeline, we test it on existing benchmarks such as QRDATA (Liu et al., 2024a). QRDATA primarily consists of examples from textbooks, where the inference process is relatively more streamlined and structured. Real-world causal inference, however, is often less structured and complex. Hence, we extend our evaluation to case studies from academic papers as well as synthetic datasets mimicking real-world settings.

In sum, our key contributions are:

1. We introduce an LLM-powered end-to-end tool, **CAIA**, for estimating causal effects on a given dataset to generate data-driven ansters to use queries. CAIA automatically selects and implements the appropriate method and interprets the final numerical results in the context of the user query.
2. Our pipeline leverages the **Tree of Thoughts (ToT)** prompting approach to break down the selection of the method and the appropriate variables. This decomposition simplifies the causal analysis steps, thereby making the results more interpretable.
3. We evaluate our pipeline on existing benchmark datasets, causal queries based on real-world studies, and synthetic datasets. CAIA outperforms the baseline models in terms of method selection across all three datasets. Similarly, it achieves lower error rates on queries associated with QRData and synthetic dataset.

2 Problem Formulation

We are provided with:

1. A dataset $\mathcal{D} = \{X_i, Y_i, T_i\}_{i=1}^n$, where $X_i \in \mathbb{R}^d$ denotes the covariates, T_i is the treatment (binary or continuous), and Y_i is the observed outcome for unit i.
2. A description D, detailing the variables and the data collection mechanism.
3. A natural language causal query q. For example: **Does participating in the training program lead to higher earnings?**

The goal is to then generate a causality-driven answer to the query q.

2.1 Causal Estimand

The key numerical quantity of interest are causal estimands, which gives a measure of the causal effect. The primary causal estimands we consider are:

- **Average Treatment Effect (ATE)**: $\mathbb{E}[Y(1) - Y(0)]$
- **Average Treatment Effect on the Treated (ATT)**: $\mathbb{E}[Y(1) - Y(0) \mid T = 1]$
- **Local Average Treatment Effect (LATE)**: $\mathbb{E}[Y(1) - Y(0) \mid \text{Compliers}]$

The reported causal estimand directly informs the answer to the causal query by quantifying the causal effect. Additionally, one can compute the confidence interval associated with the estimates to gauge its statistical reliability.

2.2 Inference Method Selection

To estimate the estimand, we must first identify the appropriate inference method. This choice depends largely on the characteristics of the dataset. For example, if the data originates from a randomized controlled trial (RCT), the treatment effect can be estimated using the simple difference in means:

$$\hat{\tau} = \frac{1}{n_1} \sum_{i:T_i=1} Y_i - \frac{1}{n_0} \sum_{i:T_i=0} Y_i \qquad (1)$$

where n_1 and n_0 are the number of treated and control units, respectively.

Each method relies on specific assumptions for identification. Thus, to assess the suitability of a method, it is essential to test the assumptions. The assumptions underlying each inference method can be found in most standard causal inference textbooks (Imbens and Rubin, 2015; Cunningham, 2021; Huntington-Klein, 2021; Hernan and Robins, 2025).

3 Methodology

OurModel is implemented as a *modular, agent-based pipeline* that decomposes the overall causal inference process into a sequence of well-defined

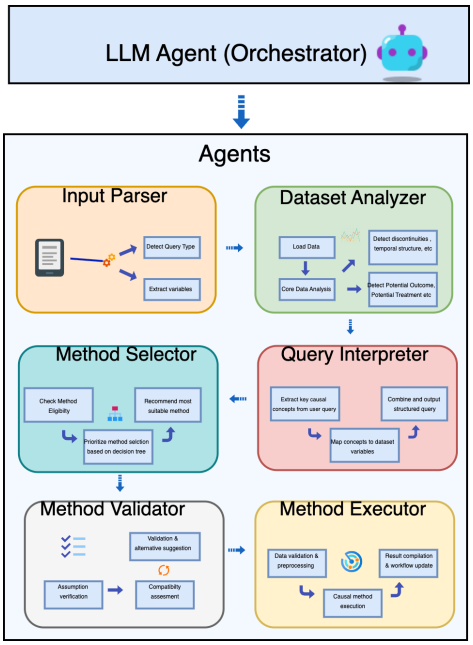

Figure 1: The overall architecture of our Causal-AI Assistant.

tasks (Figure 1). It consists of three stages: preprocessing, causal inference, and final interpretation. Additionally, each stage consists of distict components that perform a specific function. This decomposition enhances interpretability by allowing users to trace how a causal estimate is derived.
.

3.1 Overview of the Three-Stage Agentic Workflow

3.2 Stage 1: Preprocessing

The preprocessing stage performs preliminary analysis of the input dataset and the user query. It identifies key variables and characteristics of the datasets, such as treatment and outcome variables, presence of valid instrument variables, presence of observation timings, etc. The tasks in this stage are performed using three agents.

3.2.1 Agent 1a: Input Parser

This agent parses the user's natural language query and the description of the dataset. Additionally, it checks the query for any references to treatment and outcome variables. Likewise, the parser also obtains the data from the user specified path.

3.2.2 Agent 1b: Dataset Analyzer

This agent conducts a comprehensive examination of the provided data set. Identifies column names, infers data types, quantifies missing values, and computes summary statistics. Beyond basic profiling, the agent explores potential relationships within the data, such as correlations, and attempts to identify features pertinent to causal inference, including candidate treatment and outcome variables, temporal structures, and possible instrumental variables. The heuristics act as a fallback mechanism in cases where the LLM fails to identify the variables.

3.2.3 Agent 1c: Query Interpreter

This agent bridges the gap between the user's conceptualization of the causal query and the actual data. Building on the output of the parser and analyzer, it prompts the LLM to determine which columns correspond to the treatment, outcome, and control variables. In addition, it guides the LLM to identify the presence of instrumental variables, running variables that govern treatment assignment, observed confounders, and time-related variables that indicate the timing of observations. It also prompts the LLM to infer the nature of the data - whether it is observational or experimental - based on the dataset and query context.

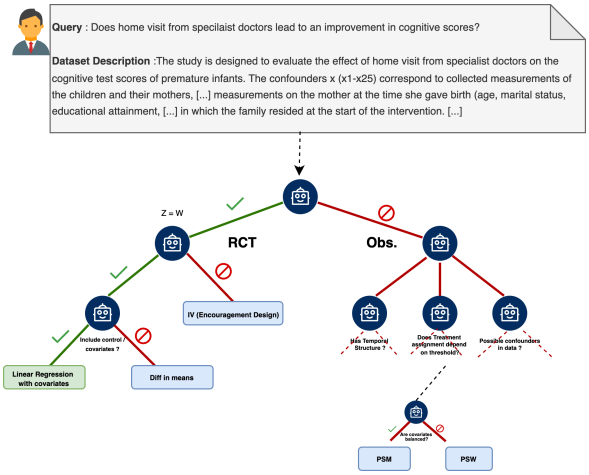

Figure 2: Illustration of the decision tree used in method selection.

3.3 Stage 2: Causal Inference

In this stage, we leverage the results of preprocessing stage to select the appropriate causal inference method, validate its assumptions, and execute the estimation

3.3.1 Agent 2a: Method Selector

Method Selector agent is responsible for selecting the most appropriate causal inference method to answer the user's query. It uses a decision tree structure (Figure 2), where each node checks for a key characteristic of the data or query, such as whether the data are observational or experimental,

the presence of discontinuities or the availability of instrumental variables. Based on these checks, the tree guides the selection process toward a suitable method, which corresponds to the leaf nodes (Full decision tree method selection process can be refereed in Appendix C).

3.3.2 Agent 2b: Method Validator

In this phase, we perform checks to gauge the reliability of the method. Method-specific diagnostic checks are conducted to ensure the validity of the underlying assumptions. Violation of the assumptions compromise the result of the estimates. The nature of these checks varies by method — for example, the parallel trends assumption is tested for Difference-in-Differences, while instrument strength is assessed in Instrumental Variables analysis. The output of this component is a diagnostic report indicating whether the necessary assumptions hold or not.

3.3.3 Agent 2c: Method Executor

Finally, this agent implements the selected causal inference method. For most methods, pre-defined code templates are used with placeholders for key variables, which are filled using the output of the LLMs. However, for certain methods, such as propensity score matching, LLMs are more involved. For instance, we prompt the LLM to select the variables used for computing propensity score. The output of this component is the estimated causal effect along with standard error, p-values, and confidence intervals.

It is crucial to emphasize that the reported confidence intervals primarily quantify statistical uncertainty due to finite sampling, under the specific model and its assumptions. These statistical measures do not, in isolation, confirm the overarching causal claim, as the validity of such claims also hinges critically on the appropriateness of the chosen method, the untestable identifying assumptions (e.g., absence of unobserved confounders, selection bias) and data quality.

3.4 Stage 3: Final Interpretation

The final component interprets the results of the causal analysis within the context of the original query and the data set. Produces a comprehensive answer to the causal question by presenting the estimated effect alongside any limitations and caveats identified through diagnostic and validity assessments. This approach guarantees that the response is both insightful and appropriately qualified.

4 Experimental Setup

4.1 Baseline Models

To evaluate the performance of our Tree-of-Thoughts based approach, we compare it against a baseline that uses Chain-of-Thought(Wei et al., 2023) prompting (Appendix section B) for end-to-end causal data analysis (Liu et al., 2024a). The process involves three steps: (i) providing the LLM with a description and summary of the dataset, (ii) supplying the causal query along with a set of candidate methods, and (iii) prompting the LLM to select an appropriate method and then write a code to implement the method using a selection of Python libraries. The prompt also includes instructions to return the selected method, the causal effect estimate, and the standard errors and confidence intervals associated with the estimate. The key outputs of interest are the estimated causal effects and the inference method.

4.2 Implementation Details

Both the baseline and CAIA utilize GPT models (4o and 4o-mini) as the core LLM to interpret natural language queries, dataset descriptions, and summaries. For causal effect estimation, we rely on the DoWhy and statsmodels libraries, using scikit-learn's logistic regression for propensity score estimation. The causal inference methods included are Difference-in-Differences (DiD), Regression Discontinuity Design (RDD), Ordinary Least Squares (OLS), Instrumental Variables (IV), Propensity Score Matching (PSM), and Inverse Propensity Score Weighting (IPW). Data preprocessing is performed with pandas and numpy.

All experiments are implemented in Python and interface with GPT via the OpenAI API, with the temperature parameter fixed at 0 to ensure reproducibility

4.3 Evaluation Metrics

We evaluate our pipeline on two metrics.

- **Method Accuracy (MA)**: The proportion of cases where CAIA selects the same causal inference method as specified in the reference datasets or studies. Numerically,

$$\text{MA} = \frac{1}{N}\sum_{i=1}^{N} \mathbf{1}[\hat{m}_i = m_i] \qquad (2)$$

where $\hat{m}_i$ is the method predicted by CAIA, and m_i is the method used in the references.

- **Mean Relative Error (MRE)**: The average relative error between the estimated causal effects and the reference values:

$$\text{MRE} = \frac{1}{N}\sum_{i=1}^{N} \min\left(\frac{|\hat{\tau}_i - \tau_i|}{|\tau_i|}, 1\right) \times 100\% \tag{3}$$

where $\hat{\tau}_i$ is the estimated causal effect and τ_i is the reference causal effect. The relative error is sensitive to outliers. To avoid the effect of the outliers, we cap the relative error for each estimate to 100%.

4.4 Prompt Setup

Prompt structuring is a core element of OurModel, enabling rigorous and reliable guidance of the LLM throughout the causal inference workflow. Each prompt is carefully constructed as a dynamic template that embeds relevant dataset metadata, variable information, and the user's causal query, thereby providing clear and context-sensitive instructions. We categorize these prompts into four principal groups: (i) Method Identification, (ii) Dataset Analysis, (iii) Result Interpretation, and (iv) Regression Analysis.

Key elements of our prompt structure include:

- **Explicit task definition**: Precise specification of the objective, such as method selection or instrument identification.
- **Comprehensive contextual input**: Inclusion of dataset summaries, variable descriptions, and other metadata to anchor LLM reasoning.
- **Structured output requirements**: Mandating responses in standardized, machine-readable formats (e.g., JSON) for seamless downstream integration.
- **Illustrative guidance**: Providing examples and expected formats to facilitate consistent and accurate model outputs.

5 Results and Analysis

5.1 Performance on Textbook Data

5.1.1 Benchmark Dataset

QRData (Liu et al., 2024b) is a benchmark dataset that primarily draws examples from causal inference textbooks. The queries specify the method/estimand of interest and instructs LLM to implement them. Since our focus is on performing end-to-end causal analysis, including method and variable selection, we modify the queries to remove mention of method and categorization of variables as treatment and outcomes. We omit 3 that are out of scope for our pipeline. Likewise, we use each variant of the 10 IHDP datasets to create one query. This brings the total number of queries with numerical answers to 39.

5.1.2 Result

Model	Prompt	MA	MRE
gpt-4o	Baseline	45	46.2
	CAIA	**76.9**	**31.6**
gpt-4o-mini	Baseline	36	40
	CAIA	**60**	**67**

Table 1: Performance on causal queries in QRData reported in terms of Method Accuracy (**MA**) and Mean Relative Error (**MRE**) of the causal effect estimates

As shown in Table 3, CAIA outperforms the baseline in method selection accuracy for both GPT-4o and GPT-4o-mini across all 39 queries. For causal effect estimation, CAIA with GPT-4o also achieves lower mean relative error than the baseline based on the results for 34 out of 39 queries. For 5 of the queries, we ran into implementation errors. (described in See 5.3)

5.2 Performance on Synthetic Data

5.2.1 Synthetic Data Creation

One of the challenges in testing our approach is the limited availability of open-source datasets with known causal effects. To address this limitation, we create synthetic datasets for each causal inference method in our pipeline. We randomly select the true causal effect τ in the range (1, 10). Continuous covariates are generated from a normal distribution, while binary covariates and treatment assignments (in binary treatment settings) are generated from a binomial distribution. The outcome Y is generated based on the model specification. For instance, for a randomized trial,

$$Y = \alpha + X\vec{\theta} + \tau T + \epsilon \tag{4}$$

where $\epsilon \sim \mathcal{N}(0, 1)$ is the error term, $\theta \sim \mathcal{N}(u, kI)$, and α is the intercept. Similarly, X represents the concatenation of binary and continuous covariates, and T is the treatment variable.

After generating the numerical values, we employ GPT-4o to create hypothetical contexts for each

dataset. Specifically, we prompt GPT-4o to invent realistic scenarios from which the data could have arisen. Simultaneously, we ask GPT to generate headings and descriptions for the covariates, outcomes, and treatment variables. This process provides meaningful backgrounds to the dataset for testing our pipeline's ability to handle diverse real-world situations.

5.2.2 Results

As with the QRData benchmark, CAIA outperforms the baseline in both method selection and causal effect estimation. We excluded four queries from the relative error evaluation due to implementation failures. While the gap in method selection accuracy is substantial, the difference in relative error between the baseline and CAIA is comparatively smaller. This is largely because the true causal effects across most queries are relatively small—typically in the range of 1 to 10, which reduces the magnitude of differences in estimation error.

Model	Prompt	MA	MRE
gpt-4o	Baseline	30	28
	CAIA	**73.3**	**22.24**
gpt-4o-mini	CAIA	48.65	51.9

Table 2: Performance on causal queries for synthetic dataset, reported in terms of Method Accuracy **(MA)** and Mean Relative Error **(MRE)** of the causal effect estimates.

5.3 Fine-Grained Analysis on QR+Syn Combined

Method Selection Accuracy For both QRData and synthetic data, we observe a relatively high accuracy for method selection and a low mean relative error. One possible reason for this is the simplified nature of the dataset. QRData uses examples from causal inference textbooks. Thus, the data is heavily preprocessed to enable the implementation of the inference methods. Similarly, given the data-generating process, all columns of synthetic datasets are numerical. Likewise, the column names are unambiguous and distinct, which makes it easier for the LLM to select the correct set of variables.

Common Errors Here we briefly describe the common types of errors.

- **Incorrect Variable Selection:** LLMs often misinterpret time-related covariates, such as year of birth or quarter, as observation times. This can erroneously lead to the selection of Difference-in-Differences as the causal inference method. Likewise, wrong columns get selected as treatment and outcomes, especially if the column names are ambiguous.
- **Wrong Method Selection** LLMs perceive Randomized Control Trials as Encouragement Designs leading to the selection of IVs instead of OLS as shown in figure 3. Similarly, for synthetic data, the model failed to recognize IV as the preferred method in 3 cases. This highlights the general difficulty of choosing valid instruments based on data descriptions.
- **Incorrect Data Formats** Errors also arise due to inconsistent formatting of the data. For instance, certain columns are formatted in strings, and packages like DoWhy needs inputs in numerical formats.

5.4 Ablation Study

5.4.1 Impact of Dataset Descriptions on LLM-Guided Causal Inference

We conduct an ablation study to assess the impact of explicit dataset descriptions in the Causal AI Assistant pipeline. When prompts include detailed descriptions of the semantics of the variable and the study context in natural language, the LLM more accurately identifies treatments, outcomes, and covariates, resulting in better selection of methods and estimation of effects across QRData and synthetic datasets. In contrast, omitting these descriptions leads to more frequent errors, especially with ambiguous or domain-specific column names. These findings underscore the importance of curated dataset descriptions for robust and reliable LLM-driven causal analysis.

5.4.2 Decision Tree vs. LLM-Only Method Selection

We specifically compared the method selection logic implemented within the method selector agent, evaluating two distinct strategies: an explicit decision tree-based approach (Figure 2) versus a purely LLM-driven approach. While both strategies performed similarly on simple queries, the decision tree's structured, rule-based logic consistently delivered higher accuracy and interpretability for complex cases involving multiple treatments or interaction effects. Conversely, the LLM-only approach often failed to capture critical dataset nu-

ances and methodological requirements, leading to less reliable method selection in challenging scenarios. This analysis underscores the importance of robust, expert-encoded logic in guiding causal method selection.

Model	Prompt	MA
gpt-4o	LLM-Only	60
	Decision-Tree	**76.9**

Table 3: Performance comparison on Decision-Tree vs LLM-Only approach for method selection, reported in terms of Method Accuracy **(MA)**

5.5 Future Work Direction: Real-World Data

5.5.1 Real-World Data Collection

To evaluate our model in complex and real-world scenarios, we create test cases using published social science studies. For each study, we use a summary that captures key details about the dataset, including the main variables and the experimental procedure involved in data creation. A substantial portion of the studies are associated with datasets found in the R package causaldata. We develop the causal questions associated with the curated studies by considering the empirical answer, the corresponding statistical model, and its relation to the original study.

5.5.2 Results

Model	Prompt	MA	MRE
gpt-4o	Baseline	45	46.2
	CAIA	**73.3**	62.97
gpt-4o-mini	Baseline	36	40
	CAIA	**60**	67

Table 4: Performance on causal queries curated from real-world studies, reported in terms of Method Accuracy **(MA)** and Mean Relative Error **(MRE)** of the causal effect estimates.

CAIA achieves an accuracy of 73.3% on real-world studies for method selection and outperforms the baseline model when using GPT-4o. However, the error in the causal effect estimate is very high relative to the baseline model. A big reason for this is incorrect selection of variables when implementing the model. The chance of incorrect variable selection is higher for real studies because the raw dataset contains large number of columns.

Error Type	Dataset	Percentage %
Variable Identification	QRData	20%
	Synthetic	25%
Method Selection	QRData	18%
	Synthetic	15%
Formulation	QRData	20%
	Synthetic	20%
Others	QRData	42%
	Synthetic	40%

Table 5: Error Analysis of Causal AI assistant with gpt-4o on QRdata and synthetic dataset

5.5.3 Challenges and Future Work

One of the key challenges in working with real-world studies is the structure of the raw datasets. These datasets are often direct transcriptions of surveys and include a large number of variables. We currently use the data in its raw form, which leads to the inclusion of many control covariates in the model. This can adversely the accuracy of the estimates in regression models. Additionally, this may introduce instability. For example, in one case, the selected model included complementary covariates: an indicator for being born in the given country was used as the treatment, while a dummy variable for immigrant status was included as a control covariate. This led to multicollinearity issues. The presence of large number of variables also increases the likelihood of incorrect variable selection, especially when column names are similar or ambiguous. Such errors in variable identification can also result in the selection of inappropriate causal inference methods.

Currently, for each specific task, we prompt the LLM only once and do not apply filters to verify the correctness of its output. If the initial response is incorrect, the entire downstream pipeline can be compromised. To address this limitation, we are exploring techniques such as Chain of Verification (Dhuliawala et al., 2023), which prompts the LLM to re-evaluate and validate its own outputs. Another promising direction involves integrating pre-processing tools that can reformat raw datasets into formats compatible with causal inference libraries. This could improve both variable selection and model robustness.

6 Related Work

LLMs and causality The applications of LLMs in causal inference are an active area of research. LLMs enable the estimation of treatment effects from high-dimensional textual data (Dhawan et al.,

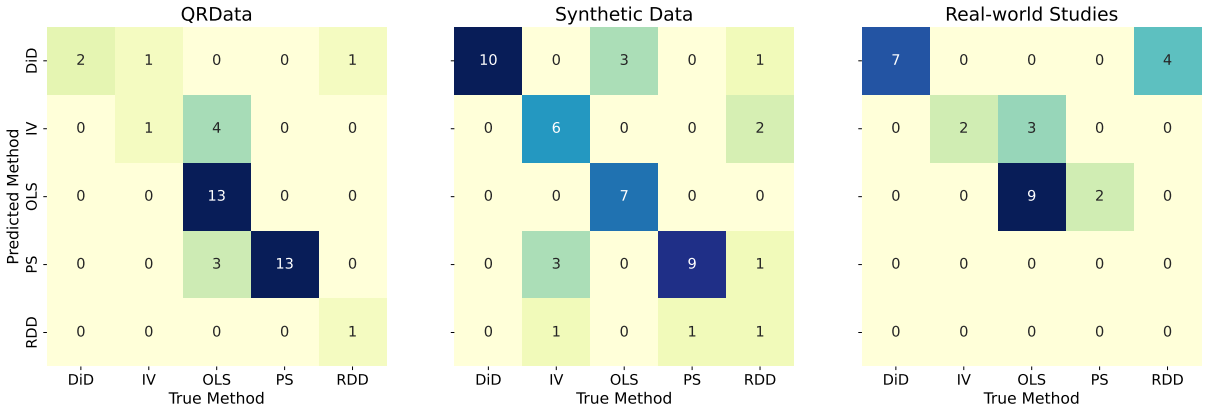

Figure 3: CAIA (GPT-4o): Method Selection Performance via Confusion Matrixo

2024; Imai and Nakamura, 2024). Another line of work uses LLMs for generating high-quality labels (Egami et al., 2024; Durvasula et al., 2025). Recent econometrics papers (Ludwig et al., 2025; Battaglia et al., 2025) provide a statistical analysis of the properties of estimators computed using LLM-generated data. LLMs have also been used for causal discovery. Most papers take the approach where they view LLM as domain experts and utilize their knowledge to build expressive causal graphs (Kiciman et al., 2024; Choi et al., 2022; Long et al., 2022). Ban et al. (2023) takes a hybrid approach, combining LLM's knowledge with classical methods to enhance causal structural learning. (Zečević et al., 2023) provides a more critical perspective on the causal graphs generating capabilities of LLMs, conjecturing that LLMs may behave more like a causal parrot that memorizes causal relationships rather than inherently understanding them. The ability of LLMs to reason causally when provided with complex queries and models has also been studied by Jin et al. (2023) and Jin et al. (2024). More recent works, such as Jiralerspong et al. (2024), have focused on minimizing the computational costs associated with querying.

LLMs for data analysis and code generation Various studies have proposed frameworks and benchmark datasets to evaluate the code generation abilities of LLMs (Huang et al., 2022; Lai et al., 2023; Wu et al., 2024). A thorough analysis of the code generation capabilities of GPT in the context of data analysis is presented in Cheng et al. (2023). Liu et al. (2024a) extend this line of work by analyzing LLMs' code generation and analysis skills for answering queries that involve implementing statistical and causal inference models. Wu et al. (2024) presents fine-tuning methods to enhance the analytical capabilities of LLMs for more difficult tasks. Nejjar et al. (2024) and Jansen et al. (2023) analyze the code generation and data analysis capabilities of language models in the context of scientific research. More recently, LLMs have been applied in automating the end-to-end causal inference process via LLM Co-pilots(Alaa et al., 2024; Wang et al., 2025) have been proposed. On the benchmarking end, datasets like DISCOVERYBENCH (Majumder et al., 2024), BLADE (Gu et al., 2024), and StatQA (Zhu et al., 2024) have been proposed to assess the ability of LLMs to perform data-driven analysis.

7 Conclusion

In this work, we propose **CausalAI Assistant (CAIA)**, an end-to-end framework for generating causality-driven answers to user queries based on an input dataset. Currently, CAIA supports methods primarily used in the social sciences domain (Imbens, 2024). We evaluate CAIA across a range of causal inference methods using three types of datasets: QRData, synthetic data, and real-world studies. CAIA outperforms the baseline model on both QRData and synthetic datasets in terms of method selection and causal effect estimation. While its performance on real-world studies is comparatively lower for causal effect estimation, the strong results on QRData and synthetic datasets, which are more structured and cleaned, suggest that CAIA's performance can be improved on real-world data through better preprocessing.

Limitations

Our work has several limitations. Some of these have been discussed in earlier sections, including the need for improved data preprocessing and additional verification steps to validate LLM outputs. Moreover, the results reported in this study are based on a single run per dataset. Given the variability in LLM outputs, a robust evaluation would require running multiple trials on the input datasets. While CAIA supports a diverse set of causal inference methods applicable to a broad range of datasets, our current focus has been primarily on queries and datasets from the social sciences. Causal inference is a vast field, and this work concentrates on a selected subset of tools and techniques.

References

Ahmed Alaa, Rachael V. Phillips, Emre Kıcıman, Laura B. Balzer, Mark van der Laan, and Maya Petersen. 2024. Large language models as co-pilots for causal inference in medical studies. *Preprint*, arXiv:2407.19118. 8

Taiyu Ban, Lyvzhou Chen, Xiangyu Wang, and Huanhuan Chen. 2023. From query tools to causal architects: Harnessing large language models for advanced causal discovery from data. *Preprint*, arXiv:2306.16902. 8

Laura Battaglia, Timothy Christensen, Stephen Hansen, and Szymon Sacher. 2025. Inference for regression with variables generated by ai or machine learning. *Preprint*, arXiv:2402.15585. 8

Liying Cheng, Xingxuan Li, and Lidong Bing. 2023. Is GPT-4 a good data analyst? In *The 2023 Conference on Empirical Methods in Natural Language Processing*. 8

Kristy Choi, Chris Cundy, Sanjari Srivastava, and Stefano Ermon. 2022. LMPriors: Pre-trained language models as task-specific priors. In *NeurIPS 2022 Foundation Models for Decision Making Workshop*. 8

Scott Cunningham. 2021. *Causal Inference: The Mixtape*. Yale University Press. 2

Nikita Dhawan, Leonardo Cotta, Karen Ullrich, Rahul Krishnan, and Chris J. Maddison. 2024. End-to-end causal effect estimation from unstructured natural language data. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*. 7

Shehzaad Dhuliawala, Mojtaba Komeili, Jing Xu, Roberta Raileanu, Xian Li, Asli Celikyilmaz, and Jason Weston. 2023. Chain-of-verification reduces hallucination in large language models. *Preprint*, arXiv:2309.11495. 7

Maya M. Durvasula, Sabri Eyuboglu, and David M. Ritzwoller. 2025. Counting clinical trials: New evidence on pharmaceutical sector productivity. *Preprint*, arXiv:2405.08030. 8

Naoki Egami, Musashi Hinck, Brandon M. Stewart, and Hanying Wei. 2024. Using imperfect surrogates for downstream inference: Design-based supervised learning for social science applications of large language models. *Preprint*, arXiv:2306.04746. 8

Ken Gu, Ruoxi Shang, Ruien Jiang, Keying Kuang, Richard-John Lin, Donghe Lyu, Yue Mao, Youran Pan, Teng Wu, Jiaqian Yu, Yikun Zhang, Tianmai M. Zhang, Lanyi Zhu, Mike A. Merrill, Jeffrey Heer, and Tim Althoff. 2024. Blade: Benchmarking language model agents for data-driven science. *Preprint*, arXiv:2408.09667. 8

M.A. Hernan and J.M. Robins. 2025. *Causal Inference: What If*. Chapman & Hall/CRC Monographs on Statistics & Applied Probab. CRC Press. 2

Junjie Huang, Chenglong Wang, Jipeng Zhang, Cong Yan, Haotian Cui, Jeevana Priya Inala, Colin Clement, and Nan Duan. 2022. Execution-based evaluation for data science code generation models. In *Proceedings of the Fourth Workshop on Data Science with Human-in-the-Loop (Language Advances)*, pages 28–36, Abu Dhabi, United Arab Emirates (Hybrid). Association for Computational Linguistics. 8

N. Huntington-Klein. 2021. *The Effect: An Introduction to Research Design and Causality*. CRC Press. 2

Kosuke Imai and Kentaro Nakamura. 2024. Causal representation learning with generative artificial intelligence: Application to texts as treatments. *Preprint*, arXiv:2410.00903. 8

Guido W. Imbens. 2024. Causal inference in the social sciences. *Annual Review of Statistics and Its Application*, qq:1123–152. 8

Guido W. Imbens and Donald B. Rubin. 2015. *Causal Inference for Statistics, Social, and Biomedical Sciences: An Introduction*. Cambridge University Press. 2

Jacqueline A Jansen, Artür Manukyan, Nour Al Khoury, and Altuna Akalin. 2023. Leveraging large language models for data analysis automation. *bioRxiv*. 8

Wenlong Ji, Weizhe Yuan, Emily Getzen, Kyunghyun Cho, Michael I. Jordan, Song Mei, Jason E Weston, Weijie J. Su, Jing Xu, and Linjun Zhang. 2025. An overview of large language models for statisticians. *Preprint*, arXiv:2502.17814. 1

Haitao Jiang, Lin Ge, Yuhe Gao, Jianian Wang, and Rui Song. 2024. Llm4causal: Democratized causal tools for everyone via large language model. *Preprint*, arXiv:2312.17122. 1

Zhijing Jin, Yuen Chen, Felix Leeb, Luigi Gresele, Ojasv Kamal, Zhiheng LYU, Kevin Blin, Fernando Gonzalez Adauto, Max Kleiman-Weiner, Mrinmaya Sachan, and Bernhard Schölkopf. 2023. CLadder: A benchmark to assess causal reasoning capabilities of language models. In *Thirty-seventh Conference on Neural Information Processing Systems*. 8

Zhijing Jin, Jiarui Liu, Zhiheng LYU, Spencer Poff, Mrinmaya Sachan, Rada Mihalcea, Mona T. Diab, and Bernhard Schölkopf. 2024. Can large language models infer causation from correlation? In *The Twelfth International Conference on Learning Representations*. 8

Thomas Jiralerspong, Xiaoyin Chen, Yash More, Vedant Shah, and Yoshua Bengio. 2024. Efficient causal graph discovery using large language models. *Preprint*, arXiv:2402.01207. 8

Emre Kiciman, Robert Ness, Amit Sharma, and Chenhao Tan. 2024. Causal reasoning and large language models: Opening a new frontier for causality. *Transactions on Machine Learning Research*. Featured Certification. 1, 8

Yuhang Lai, Chengxi Li, Yiming Wang, Tianyi Zhang, Ruiqi Zhong, Luke Zettlemoyer, Wen-Tau Yih, Daniel Fried, Sida Wang, and Tao Yu. 2023. DS-1000: A natural and reliable benchmark for data science code generation. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 18319–18345. PMLR. 8

Xiao Liu, Zirui Wu, Xueqing Wu, Pan Lu, Kai-Wei Chang, and Yansong Feng. 2024a. Are LLMs capable of data-based statistical and causal reasoning? benchmarking advanced quantitative reasoning with data. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 9215–9235, Bangkok, Thailand. Association for Computational Linguistics. 1, 2, 4, 8

Xiao Liu, Zirui Wu, Xueqing Wu, Pan Lu, Kai-Wei Chang, and Yansong Feng. 2024b. Are LLMs capable of data-based statistical and causal reasoning? benchmarking advanced quantitative reasoning with data. In *Findings of the Association for Computational Linguistics ACL 2024*, pages 9215–9235, Bangkok, Thailand and virtual meeting. Association for Computational Linguistics. 5

Xiaoyu Liu, Paiheng Xu, Junda Wu, Jiaxin Yuan, Yifan Yang, Yuhang Zhou, Fuxiao Liu, Tianrui Guan, Haoliang Wang, Tong Yu, Julian McAuley, Wei Ai, and Furong Huang. 2024c. Large language models and causal inference in collaboration: A comprehensive survey. *Preprint*, arXiv:2403.09606. 1

Jieyi Long. 2023. Large language model guided tree-of-thought. *Preprint*, arXiv:2305.08291. 2

Stephanie Long, Tibor Schuster, and Alexandre Piché. 2022. Can large language models build causal graphs? In *NeurIPS 2022 Workshop on Causality for Real-world Impact*. 8

Jens Ludwig, Sendhil Mullainathan, and Ashesh Rambachan. 2025. Large language models: An applied econometric framework. *Preprint*, arXiv:2412.07031. 8

Bodhisattwa Prasad Majumder, Harshit Surana, Dhruv Agarwal, Bhavana Dalvi Mishra, Abhijeetsingh Meena, Aryan Prakhar, Tirth Vora, Tushar Khot, Ashish Sabharwal, and Peter Clark. 2024. Discoverybench: Towards data-driven discovery with large language models. *Preprint*, arXiv:2407.01725. 8

Mohamed Nejjar, Luca Zacharias, Fabian Stiehle, and Ingo Weber. 2024. Llms for science: Usage for code generation and data analysis. *J. Softw. Evol. Process*, 37(1). 8

Judea Pearl. 2009. *Causality: Models, Reasoning and Inference*, 2nd edition. Cambridge University Press, USA. 1

Aniket Vashishtha, Abbavaram Gowtham Reddy, Abhinav Kumar, Saketh Bachu, Vineeth N Balasubramanian, and Amit Sharma. 2023. Causal inference using llm-guided discovery. *Preprint*, arXiv:2310.15117. 1

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. 2025. Causal-copilot: An autonomous causal analysis agent. *Preprint*, arXiv:2504.13263. 1, 8

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc Le, and Denny Zhou. 2023. Chain-of-thought prompting elicits reasoning in large language models. *Preprint*, arXiv:2201.11903. 4

Xueqing Wu, Rui Zheng, Jingzhen Sha, Te-Lin Wu, Hanyu Zhou, Tang Mohan, Kai-Wei Chang, Nanyun Peng, and Haoran Huang. 2024. 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*. 8

Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Thomas L. Griffiths, Yuan Cao, and Karthik Narasimhan. 2023. Tree of thoughts: Deliberate problem solving with large language models. *Preprint*, arXiv:2305.10601. 2

Matej Zečević, Moritz Willig, Devendra Singh Dhami, and Kristian Kersting. 2023. Causal parrots: Large language models may talk causality but are not causal. *Transactions on Machine Learning Research*. 8

Yizhang Zhu, Shiyin Du, Boyan Li, Yuyu Luo, and Nan Tang. 2024. Are large language models good statisticians? *Preprint*, arXiv:2406.07815. 8

A Dataset Information

Method	QRData	Synthetic	Real-Studies
DiD	2	10	7
RDD	2	5	4
IV	2	10	2
OLS	20	10	12
Propensity Score	13	10	2
Total	39	45	27

Table 6: Number of queries by method type across QRData, synthetic, and real-world studies

A.1 Sample of Synthetic Data Description Created by GPT

Descriptions and variable names generated by GPT-4o for Synthetic Data associated with Difference-in-Differences

Dataset Name: did_canonical_data0.csv

Variable labels
D: policy_change,
post: post_policy_year
Y: health_score
X1: age
X2: BMI
X3: smoker
X4: sports_participation
X5: fast_food_consumption
unit_id: student_id

Description: The dataset comprises information from a two-year study conducted in public schools, investigating the effects of a new policy requiring biannual health check-ups for students. The data were collected from student health records, surveys about lifestyle habits, and school administrative databases. The student's health score was calculated by licensed physicians based on various medical parameters. The student's age and Body Mass Index (BMI) were recorded as continuous variables. The student's lifestyle habits, namely whether they smoke, participate in sports, and consume fast food, were noted as binary variables (1 for yes, 0 for no). Each unique student is represented by a unique student ID.
The policy_change variable indicates whether a student belonged to a school where the new policy was implemented. The post_policy_year indicates whether the data point belongs to the year after the policy was introduced. The health_score is a numerical score representing the student's overall health condition. The 'age' represents the age of the student. The BMI represents the Body Mass Index of the student. The smoker variable indicates whether the student smokes. The sports_participation variable indicates whether the student participates in sports. The fast_food_consumption variable indicates whether the student frequently consumes fast food. The student_id is a unique identifier for each student.

Query: Did the introduction of the biannual health examination policy improve the overall health of students?

B Pipeline prompts

B.1 Baseline prompt

Baseline prompt

You are an expert in statistics and causal reasoning. You will answer a causal question on a tabular dataset.
The dataset is located at self.dataset_path.
The dataset has the following description:
“‘ self.dataset_description “‘
To help you understand it, here is the result of df.describe():
“‘ df_info “‘
Here are the columns and their types:
“‘ columns_and_types “‘
Here are the first 5 rows of the dataset:
“‘ df.head() “‘
If there are less than 10 columns, here is the result of df.cov():
“‘ (df.cov(numeric_only=True) if len(df.columns) < 10 else "Too many columns to compute covariance")
“‘
Finally, here is the output of df.isnull().sum(axis = 0):
“‘ nan_per_column “‘
The causal question I would like you to answer is:
“‘ self.query “‘
Here are some examples methods, you can choose one from them: [
'propensity_score_weighting' # output the ATE
'propensity_score_matching_treatment_to_control' # output the ATT
'linear_regression' # output the coefficient of the variable of interest
'instrumental_variable' # output the coefficient of the variable of interest
'matching' # output the ATE
'difference_in_differences' # Output the coefficient
'regression_discontinuity_design' # output the coefficient
'matching_treatment_to_control' # output the ATT
'linear_regression / difference_in_means' # output the coefficient / DiM
].
Using the descriptions and information from the dataset, implement a python code to answer the causal question. Remember the dataset is located at self.dataset_path. In the case you need to preprocess the data, please do so in the code. The following libraries are available to you: dowhy, pandas, numpy, scipy, scikit-learn, statsmodels. Use the methods from the libraries as best as you can. Don't code yourself that which is already implemented in the libraries. Do not create random data. Make sure it outputs the quantitative value in the comments of the example method. The code you output will be executed and you will receive the output. Please make sure to output only one block of code, and make sure the code prints the result you are looking for at the end. Everything between your first codeblock: '“‘python' and '“‘' will be executed. If there is an error, you will have several attempts to correct the code.

C Decision Tree Method Selection Process

The method selection process in our Causal AI Assistant is governed by a decision tree designed to recommend the most

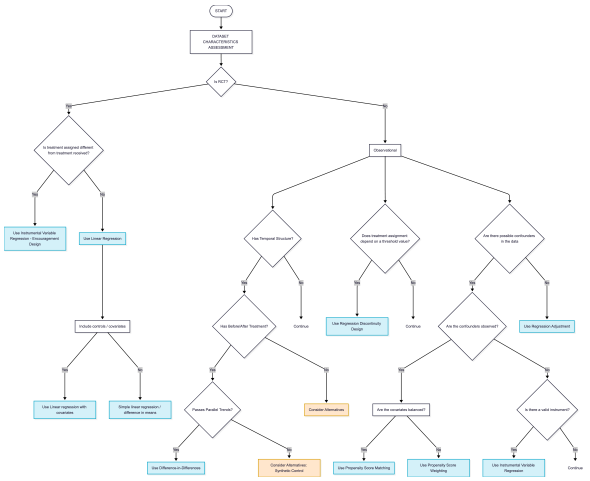

Figure 4: The overall architecture of our Causal-AI Assistant.

appropriate causal inference method based on dataset characteristics and the causal query. This process is implemented programmatically to ensure transparency, reproducibility, and alignment with best practices in causal inference (Figure 4

1. Input Requirements

The decision tree requires the following inputs:

- **Dataset**: Provided as a structured DataFrame.
- **Identified Variables**: Including the treatment variable, outcome variable, covariates, and optionally, time, group, instrument, and running variables
- **Dataset Analysis Results**: Metadata and diagnostics about the dataset, such as the presence of temporal structure, variable types, and potential instruments.
- **Study Design Indicator**: Whether the data originates from a randomized controlled trial (RCT) or an observational study.
- **Language Model Assistance** :For nuanced recommendations between similar methods (e.g., matching vs. weighting)

2. Stepwise Selection Logic

The decision tree operates through a series of prioritized, mutually exclusive checks, each corresponding to a class of causal inference methods. The process is as follows:

1. Randomized Controlled Trials (RCT)

A. **Encouragement Design (Instrumental Variable in RCTs)**: If an instrument variable (e.g., an encouragement or assignment indicator) is present and distinct from the treatment variable, the tree selects the Instrumental Variable (IV) method. This is appropriate for "encouragement designs," where randomization affects an instrument rather than the treatment directly.

B. **With Covariates**:If covariates are available, the tree recommends Linear Regression, leveraging covariate adjustment to increase the precision of the treatment effect estimate.

C. **No Covariates**: If no covariates are present, the tree defaults to a simple Difference in Means estimator, comparing outcomes between treatment and control groups.

2. Observational Data

A. **Difference-in-Differences (DiD)**: If the dataset exhibits temporal structure (e.g., contains a time variable) and the research question involves pre/post or treated/control comparisons over time, the tree selects the Difference-in-Differences method. If an instrument is also present, IV is suggested as an alternative.

B. **Regression Discontinuity Design (RDD)**: If a running variable and a cutoff value are identified (indicating treatment assignment based on a threshold), the tree selects Regression Discontinuity Design

C. **Instrumental Variable (IV)**: If an instrument variable is present (and not already handled above), the tree selects IV regression, which is appropriate when a valid instrument affects treatment but not the outcome directly. If temporal structure is also present, DiD may be suggested as an alternative.

D. **Propensity Score Methods**: If covariates are available but no special design features (e.g., time, instrument, running variable) are present, the tree considers propensity score methods. The choice between Propensity Score Matching (PSM) and Propensity Score Weighting (PSW) is informed by dataset characteristics such as group sizes, covariate balance, and sample size. Optionally, a language model may be used to recommend the most suitable approach based on summary statistics.

E. **Fallback – Correlation Analysis**: If none of the above conditions are met (e.g., no covariates, no design structure, no instrument), the tree defaults to correlation analysis. This is accompanied by a warning that such analysis does not support causal claims and is subject to confounding bias.

3. Extensibility

The decision tree is implemented as a modular, extensible system. New methods or decision rules can be incorporated as the field evolves or as new requirements emerge. The logic is encoded in a transparent, auditable manner, ensuring that the rationale for each method selection is clear and reproducible.

D Causal AI Assistant Run Output

Example of Run using Causal AI assistant

Dataset Description: The CSV file ihdp_4.csv contains data obtained from the Infant Health and Development Program (IHDP). The study is designed to evaluate the effect of home visit from specialist doctors on the cognitive test scores of premature infants. The confounders x (x1-x25) correspond to collected measurements of the children and their mothers, including measurements on the child (birth weight, head circumference, weeks born preterm, birth order, first born, neonatal health index, sex, twin status), as well as behaviors engaged in during the pregnancy (smoked cigarettes, drank alcohol, took drugs) and measurements on the mother at the time she gave birth (age, marital status, educational attainment, whether she worked during pregnancy, whether she received prenatal care) and the site (8 total) in which the family resided at the start of the intervention. There are 6 continuous covariates and 19 binary covariates.
Dataset Name: ihdp_4.csv

Query: What is the effect of home visits on the cognitive test scores of children who actually received the intervention?

Run Logs
Invoking: ‘input_parser_tool‘ with ‘’input_text’: ’My question is: What is the effect of home visits on the cognitive test scores of children who actually received the intervention? The dataset is located at: ihdp_4.csv Dataset Description: "The CSV file ihdp_4.csv contains data obtained from the Infant Health and Development Program (IHDP). The study is designed to evaluate the effect of home visit from specialist doctors on the cognitive test scores of premature infants. [...], whether she worked during pregnancy, whether she received prenatal care) and the site (8 total) in which the family resided at the start of the intervention. There are 6 continuous covariates and 19 binary covariates. Please perform the causal analysis following the workflow.’‘

Invoking: ‘dataset_analyzer_tool‘ with ‘’dataset_path’: ’ihdp_4.csv’, ’dataset_description’ The CSV file ihdp_4.csv contains data obtained from [...]‘

Invoking: ‘query_interpreter_tool‘ with ‘’query_info’: ’query_text’: ’What is the effect of home visits on the cognitive test scores of children who actually received the intervention?’, ’potential_treatments’: [’home visits’], ’potential_outcomes’: [’cognitive test scores’], ’covariates_hints’: [], ’instrument_hints’: [], ’running_variable_hints’: [], ’cutoff_value_hint’: None, ’dataset_analysis’: ’dataset_info’: ’num_rows’: 747, ’num_columns’: 27, ’file_path’: ’benchmark/all_data_1/ihdp_4.csv’, ’file_name’: ’ihdp_5.csv’, ’columns’: [’treatment’, ’y’, ’x1’, ’x2’, ’[...]], ’potential_treatments’: [’treatment’], ’potential_outcomes’: [’y’], ’temporal_structure_detected’: False, ’panel_data_detected’: False, ’potential_instruments_detected’: True, ’discontinuities_detected’: True, ’temporal_structure’: ’has_temporal_structure’: False, ’temporal_columns’: [], ’is_panel_data’: False, ’id_column’: None, ’time_column’: None, ’time_periods’: None, ’units’: None[...]’‘

Output Log continue

Invoking: ‘method_selector_tool‘ with ‘’variables’: ’treatment_variable’: ’treatment’, ’treatment_variable_type’: ’binary’, ’outcome_variable’: ’y’, ’instrument_variable’: None, ’covariates’: [’x1’, ’x2’, ’x3’, ’x4’, ’x5’, ’x6’, ’x7’, ’x8’, ’x9’, ’x10’], ’time_variable’: None, ’group_variable’: None, ’running_variable’: None, ’cutoff_value’: None, ’is_rct’: True, ’treatment_reference_level’: ’0’, ’interaction_term_suggested’: True, ’interaction_variable_candidate’: ’x3’, ’dataset_analysis’: ’dataset_info’: ’num_rows’: 747, ’num_columns’: 27, ’file_path’: ’benchmark/all_data_1/ihdp_5.csv’, ’file_name’: ’ihdp_5.csv’, ’columns’: [’treatment’, ’y’, ’x1’, ’x2’, ’x3’, ’x4’, ’x5’, ’x6’, ’x7’, ’x8’, ’x9’,[....]’‘

’method_info’: ’selected_method’: ’linear_regression’, ’method_name’: ’Linear Regression’, ’method_justification’: ’Data is from an RCT and covariates are provided. Linear regression with covariates is used to potentially increase precision.’, ’method_assumptions’: [’Linear relationship between treatment, covariates, and outcome’, ’No unmeasured confounders (if observational)’, ’Correct model specification’, ’Homoscedasticity of errors’, ’Normally distributed errors (for inference)’]

Invoking: ‘method_validator_tool‘

Invoking: method_executor_tool

Executing method: linear_regression
Initializing LLM client: Provider=’openai’, Model=’gpt-4o’
Treating ’treatment’ as numeric for Patsy formula.
Adding interaction term to formula: treatment:x3
Using formula for Linear Regression: y treatment + x1 + x2 + x3 + x4 + x5 + x6 + x7 + x8 + x9 + x10 + treatment:x3
OLS model fitted successfully.

Results ’effect_estimate’: 4.041754652507938, ’confidence_interval’: [3.827091973654759, 4.256417331361118], ’standard_error’: 0.10934318809454593, ’p_value’: 9.833881203649175e-170, ’method_used’: ’Linear Regression (OLS)’, ’llm_assumption_check’: None, ’raw_results’: None, [...] ’current_step’: ’method_execution’, ’method_execution_completed’: ’method_executed’, ’next_tool’: ’explainer_tool’, ’next_step_reason’: ’Now we need to explain the results and their implications’

Output
The analysis of the dataset from the Infant Health and Development Program (IHDP) using Linear Regression has provided the following insights:
Estimated Causal Effect: The effect of home visits on the cognitive test scores of children who actually received the intervention is estimated to be **4.0418**. **Confidence Interval**: The 95% confidence interval for this estimate is [3.8271, 4.2564].
Statistical Significance: The p-value is extremely small (p=0.0000), indicating that the effect is statistically significant.
This suggests that home visits have a substantial and statistically significant positive effect on the cognitive test scores of the children in the study. The analysis assumes a linear relationship between the treatment, covariates, and outcome, and that the data is from a randomized controlled trial (RCT), which supports the validity of these findings.