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

Anonymous ACL submission

Abstract

Causal effect estimation is a core task in empirical research and evidence-based decision-making. Successfully performing this task typically requires familiarity with a range of inference methods, their statistical assumptions, and domain-specific considerations. Recent advances in large language models (LLMs) offer the potential to automate the end-to-end causal inference pipeline and broaden access to causality-driven analysis. However, existing LLM-based approaches often require users to specify the estimation method and relevant variables, which needs prior knowledge of causal inference on users end. Similarly, the end-to-end tools support a limited set of causal effect measures, omitting many methods commonly used in applied research. To address these limitations, we introduce Causal AI Scientist (CAIS), an end-to-end causal estimation tool that takes a natural language query, maps it to a formal causal estimation problem, selects and implements a suitable method, and interprets the result to answer the original query. CAIS supports a broad range of causal inference methods, enabling estimation across diverse scenarios. We evaluate CAIS using examples drawn popular benchmark dataset, academic publications, and synthetic datasets, covering a wide spectrum of causal effect measures and estimation tasks.

1 Introduction

Causal effect estimation aims to quantify the impact of a treatment or intervention on an outcome of interest. Understanding cause-and-effect relationships is central to evidence-based decision-making and empirical research across disciplines such as social science, public health, and biomedicine (Imbens, 2024; Glass et al., 2013; Kleinberg and Hripcsak, 2011). One reason why real-world causal effect estimation is hard is because we do not observe outcomes under both treatment and control for the same unit (Holland, 1986). Hence, causal inference relies on assumptions that justify comparisons between treated and control groups.

Identifying suitable methods and justifying their applicability to the task at hand typically requires domain expertise. Researchers rely on their understanding of theory, identification strategies, and the data-generating process to select estimation techniques and assess result credibility. This reliance on expert knowledge can limit access to causal analysis for users who may benefit from it but may lack methodological training. For example, a policy analyst with employment and wage data may wish to evaluate the impact of minimum wage laws but struggle to draw reliable conclusions without the appropriate tools.

Recent advances in Large Language Models (LLMs) offer a promising pathway to automate the causal inference process (Kiciman et al., 2024). Existing works use LLMs to generate code for user-specified estimation tasks (Liu et al., 2024; Chen et al., 2025). However, users need to choose the method and variables, and doing so requires familiarity with a wide range of techniques.

One approach to enable end-to-end causal analysis is fine-tuning models specifically for causal inference, such as LLM4Causal (Jiang et al., 2024). However, LLM4Causal support only a limited set of effect measures, excluding many methods used in applied research. Another direction involves general-purpose agents powered by general purpose language models. Some agents focus on statistical (Wu et al., 2024) or machine learning tasks (Hong et al., 2024; Guo et al., 2024), while others include causal inference capabilities but emphasize discovery rather than estimation (Wang et al., 2025; Han et al., 2024).

To address these limitations, we present the Causal AI Scientist (CAIS), an end-to-end LLM-powered pipeline for generating causality-driven answers to natural language queries. Given a

dataset, its description, and a query, CAIS frames the task as a causal estimation problem, selects an appropriate method, estimates the effect, and interprets the result in context. Inspired by the Tree-of-Thoughts prompting approach (Yao et al., 2023a; Long, 2023), CAIS uses a structured decision tree to break down the method selection process into smaller and focused steps. Each node in the tree prompts the model to evaluate a specific feature of the dataset or query, such as identifying the treatment, outcome, or instrument. This step-by-step approach helps simplify the reasoning process and makes it easier to follow. In addition, CAIS performs diagnostic checks and incorporates a feedback loop to correct potential errors before producing a final answer.

Additionally, we introduce a dataset of natural language causal queries. While existing benchmarks focus on implementing specified estimation procedures (Liu et al., 2024), our dataset evaluates whether LLMs can correctly frame causal estimation problems. This approach builds on related work in data-driven reasoning (Majumder et al., 2024; Wu et al., 2024; Gu et al., 2024) and extends it to causal inference. Finally, we evaluate CAIS on three types of datasets: textbook-based (QRData (Liu et al., 2024)), synthetic, and real-world examples, thus covering a wide range of causal estimation methods and scenarios.

2 Problem Statement

We are provided with:

1. A dataset $\mathcal{D} = \{X_i, Y_i, T_i\}_{i=1}^{n}$, where i denotes the participating units, $X_i \in R^d$ denotes the covariates, T_i is the treatment (discrete or continuous), and Y_i is the observed outcome for unit i.

2. A brief description of the dataset and how it was collected

3. A natural language (causal) question associated with the dataset

Given the above inputs, the goal is to generate a causality-driven answer to the query q. Consider the example query: **Does participating in the training program lead to higher earnings?** To answer this, we need to estimate the causal effect of participating in the program on earnings. A common measure of causal effect i.e. the estimand is **Average Treatment Effect (ATE)**, which the expected difference in earnings if everyone were to participate versus if no one did.

If the dataset comes from a randomized experiment, we can estimate the ATE by directly comparing average outcomes between treated and control groups. On the other hand, if the data is observational, we must account for confounding variables, like education or prior income that affect both treatment and outcome. An appropriate **estimation method** computes the estimand. This value then informs the answer to the natural language query.

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

3 Dataset

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

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

Real-world studies involve more complex designs, a broader range of variables, and less structured datasets. To evaluate CAIS in these settings, we curate examples directly from published empirical research. ‘ Both QRData and real-world examples primarily rely on linear regression, which limits coverage of other estimation methods. To address this, we construct **synthetic datasets** where the ground-truth causal effect is known and variables are generated according to specific model specifications. This enable evaluation across a wider range of methods and scenarios.

3.1 Textbook Examples

The causal effect estimation tasks in QRData specify the inference method or estimand. Since our focus is on end-to-end causal inference, including automatic method and variable selection, we modify the queries by removing explicit references to esti-

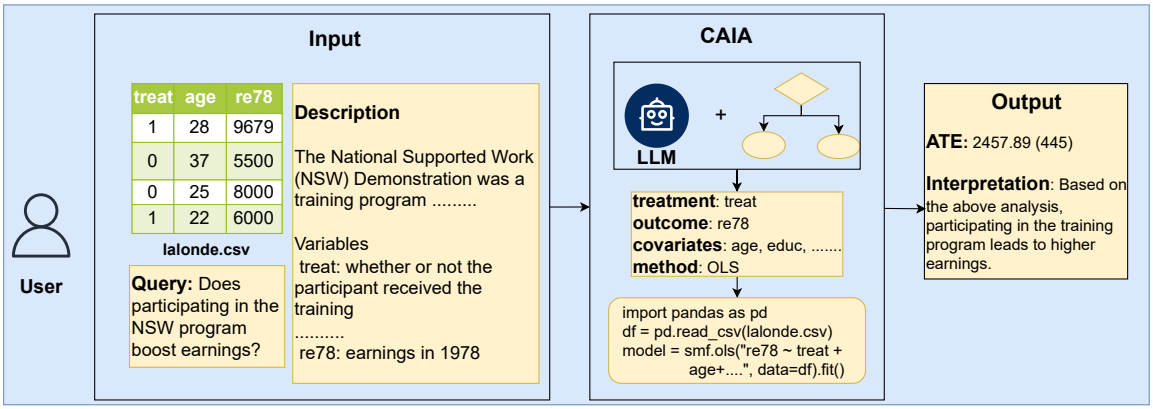

Figure 1: **CAIS workflow.** The user provides an input dataset (CSV file), a description, and a causal query. Guided by a decision tree and a backbone LLM, CAIS selects an appropriate estimation method, executes the code, and returns the estimated causal effect along with a natural language interpretation.

Collection	Origin	#Queries	#CSV	Median Obs.	Median Cols.
QRData	Textbook examples	39	35	1209	19
Real-World	Empirical research papers	29	14	1720	17.5
Synthetic	Simulated scenarios	45	45	428	7

Table 1: Properties of dataset collections used for evaluating CAIS.

mation techniques or causal effect measures. For example, the original question, **"What is the Average Treatment Effect (ATE) of the dataset?"** is rephrased as **"What is the effect of home visits by doctors on cognitive scores of infants?"** We retain the original dataset descriptions and the associated numerical estimates of the causal effects. Additionally, we restrict our evaluation to queries with numerical answers.

3.2 Real-world Studies

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

3.3 Synthetic Data

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

$$Y = \alpha + X\vec{\theta} + \tau T + \epsilon, \quad (1)$$

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

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

4 CAIS

Here, we describe CAIS, an end-to-end LLM-powered tool for generating causality-driven answers to natural language (causal) queries over tabular datasets. At a high level, CAIS operates in four stages:

Stage 1: Dataset Preprocessing and Query decomposition CAIS begins by conducting a preliminary analysis of both the input dataset and the user query. This involves prompting a large language model (LLM) to extract key variables and dataset attributes, including treatment and outcome

variables, the presence of valid instrumental variables, timing of observations, and other relevant characteristics.

Stage 2: Method Selection and Execution Based on the variables identified in the previous stage such as whether data comes from a randomized controlled trial (RCT), CAIS selects one or more suitable causal inference methods. These methods are then applied to estimate the causal effect of interest.

Stage 3: Validation To assess the reliability of the estimated causal effects, CAIS performs a series of diagnostic evaluations, including statistical tests and robustness checks. If these diagnostics reveal potential issues, CAIS either revisits the method selection process or switches to an alternative modeling approach.

Stage 4: Interpretation Finally, the outcomes of the estimation and validation stages are synthesized to generate an interpretable response that directly addresses the original user query.

4.1 Stage 1: Dataset Preprocessing and Qeury decompostion

CAIS initiates the pipeline with a thorough examination of the dataset, inspecting column names, data types, quantifying missing values, and computing descriptive statistics. Beyond basic profiling, it investigates potential relationships within the data, such as correlations, and attempts to identify features relevant to causal inference. CAIS then proceed to conceptualize the user's query and actual data by prompting the LLM to determine which columns correspond to the treatment, outcome, and control variables. In addition, it guides the 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

4.2 Stage 2: Method Selection and Execution

In this stage CAIS identifies a suitable estimation method through a structured decision tree (refer Appendix figure B). Inspired by the Tree of Thoughts framework (Yao et al., 2023a; Long, 2023), this tree decomposes method selection into sequential steps, where each node evaluates a specific property of the dataset (e.g., timing of treatment, presence of instruments). Additionally, each node is associated with a detailed prompt that specifies the required characteristics of a valid method or variable. The hierarchical structure of the tree enhances interpretability and leverages the LLM's strength to perform well on a specific narrowed-down task. If multiple branches remain viable, CAIS relies on LLM to rank them by inspecting stage 1 results

Compared to machine learning models, causal estimation does involve hyperparameters to tune, and a small number of methods are appropriate for the task at hand. However, incorrect choices can lead to invalid results. Thus, our goal is to ensure interpretability and accuracy, and the decision tree supports that. Nonetheless, errors in one step can propagate to subsequent steps. For instance, the model may incorrectly assume an RCT setting for observational data, and produce incorrect findings. We have implemented safeguards to mitigate such cases.

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

4.3 Stage 3: Validation

After computing the numerical result, we first perform statistical checks to assess its reliability. Next, we prompt an LLM to reflect on both the test outcome and the numerical value to evaluate whether the result is sensible. If the LLM perceives the results to be unreliable, we go back to the method and variable selection phase. In the next iteration, we incorporate information about estimation and validation test results from the previous run. This allows an LLM to factor in previous results. Likewise, if more than one method was identified earlier, we try an alternative one.

The feedback mechanism serves as a safeguard against potential errors that may arise in the selection and estimation phases. We provide a detailed example of the method validator in appendix C

4.4 Stage 4: Interpretation

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

5 Experimental Setup

5.1 Baseline

We consider the following baseline models:

ReAct Prompting Following Liu et al. (2024), we adopt a ReAct-based prompting strategy (Yao et al., 2023b). Each prompt includes (i) a description of the dataset, and (ii) a natural language causal query, along with a list of candidate estimation methods. The LLM is then asked to identify relevant variables, select a suitable method, and compute the causal effect accordingly. A sample prompt is provided in the Appendix 7.

Liu et al. (2024) benchmark several prompting strategies for causal inference and find that ReAct achieves the best overall performance. Hence, we select ReAct-based approach as the baseline.

Veridical Data Science Prompting Typical prompting strategies for data science are relatively straightforward and involve limited self-reflection. To address this, we design a prompting structure based on the Veridical Data Science framework (Yu, 2020). After each decision, the LLM is prompted to reflect on its response and reconsider its output, with the goal of improving stability in the reasoning process, refer Appendix 8.

Program of Thought Prompting : Unlike ReACT prompting, which interleaves reasoning and tool invocation through iterative steps, Program of Thought (PoT) (Chen et al., 2022) follows a structured template where the model outputs a complete executable Python function. This approach aims to improve execution reliability and interpretability, reduces retry loops, and facilitates code reuse, making it particularly effective for scientific domains requiring traceable and reproducible causal analysis, refer Appendix 6.

5.2 Implementation Details

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

5.3 Evaluation Metrics

We evaluate our pipeline using the following metrics.

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

$$\text{MSA} = \frac{1}{N}\sum_{i=1}^{N}\mathbf{1}[\hat{m}_i = m_i] \times 100 \quad (2)$$

- **Variable Selection Accuracy (VSA)**: Average overlap between predicted model variables $(\hat{V}_i)$ and reference model variables (V_i):

$$\text{VSA} = \frac{1}{N}\sum_{i=1}^{N}\frac{|\hat{V}_i \cap V_i|}{|V_i|} \times 100 \quad (3)$$

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

$$\text{MRE} = \frac{1}{N}\sum_{i=1}^{N}\min\left(\frac{|\hat{\tau}_i - \tau_i|}{|\tau_i|}, 1\right) \times 100\% \quad (4)$$

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

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

5.4 Ablation Studies

We conduct four ablation studies to assess the role of the core components in our pipeline.

Removing Feedback from Validation To assess the role of internal feedback, we remove the connection between the validation and method selection. Without this feedback loop, CAIS loses the ability to self-correct when validation detects an

Study	Feedback Loop	GPT-4o	GPT-4o-mini	o3-mini
QR	Without	48.4	**54**	43.3
	With	**31.6**	55.9	**43.1**
Real	Without	61.55	60.8	60.6
	With	**47.5**	**54.5**	**39.7**
Synth	Without	18.84	43.32	16
	With	**17.4**	**16.2**	**20**

Table 2: Mean Relative Error (%) across dataset for With and Without validation feedback loop for causal effect estimation (lower is better)

error or inconsistency. Then, the pipeline proceeds with the initially selected method. Results (Table 2) clearly suggests having feedback from validation loop improves overall efficiency of CAIS

Removing Decision Tree Guidance This ablation examines the benefit of structured decision-tree reasoning in method selection. Instead of relying on the tree, we prompt the LLM directly to select a suitable method based on the dataset and query. The chosen method is then executed using CAIS's implementation module. The results (Table

Study	Method Selection	GPT-4o	GPT-4o-mini	o3-mini
QR	LLM Based	48.4	50.6	50
	Decision Tree	**74.4**	**74.3**	**94.1**
Real	LLM Based	48	60.8	45.5
	Decision Tree	**69.2**	**65.2**	**76.9**
Synth	LLM Based	57.5	**79.4**	57.1
	Decision Tree	**76.9**	78.9	**73.3**

Table 3: Method Selection Accuracy(%) across dataset for LLM Based vs Decision Tree (CAIS's) approach (higher is better)

3) reveal a decision tree-based structured method approach significantly outperforms the LLM-based approach for method selection.

6 Results

6.1 Method Selection

The results (Table 4) show that CAIS consistently surpasses baseline models (ReACT, PoTM, and Veridical) across all dataset types and LLMs, with substantial margins. This performance gain is primarily attributed to CAIS 's structured, decision-tree-based approach to method selection. In contrast, baseline models often fail to capture complex causal relationships and tend to default to simplistic estimation techniques, such as linear regression, regardless of the data context.

6.2 Causal Effect Estimation

Table 5 presents the mean relative errors of causal effect estimates across different models. For larger

LLM	Datatype	ReACT	PoT	Veridical	CAIS
GPT-4o	QR Data	55	41	60.5	**74.4**
GPT-4o-mini		55.2	54.3	41	**74.3**
o3-mini		21.8	30.7	61.5	**94.1**
Gemini 2.5 Pro		62.2	50	59	**81.2**
Llama 3.3 70B		34.4	53.8	46.1	**81.8**
GPT-4o	Synthetic Data	51.2	53.3	**79**	76.9
GPT-4o-mini		41.86	37.7	43.4	**75.9**
o3-mini		46.7	42.2	66.6	**73.3**
Gemini 2.5 Pro		48.2	53.2	58.5	**75.6**
Llama 3.3 70B		55.8	47.6	50	**79.5**
GPT-4o	Real Data	69.5	57.7	48	**69.2**
GPT-4o-mini		51.8	54.6	28	**65.2**
o3-mini		57.1	33.3	59.2	**76.9**
Gemini 2.5 Pro		55	42.2	53.2	**78.3**
Llama 3.3 70B		44.4	53.8	24	**73**

Table 4: Method Selection Accuracy (%) across datasets comparing CAIS with baselines (higher is better)

LLM	Datatype	ReACT	PoT	Veridical	CAIS
GPT-4o	QR Data	43.2	32.6	40.7	**31.6**
GPT-4o-mini		33.9	**33.6**	42.2	55.9
O3-mini		44.7	30.7	27.6	**43.1**
Gemini 2.5 Pro		43.2	35.8	**37.8**	41.2
Llama 3.3 70B		43.9	**31.5**	55.4	54.19
GPT-4o	Synthetic Data	27.9	19.9	27.7	**17.4**
GPT-4o-mini		21.2	37.7	25.7	**16.2**
O3-mini		21	42.2	20.2	**20**
Gemini 2.5 Pro		20.2	24	26.5	**18.5**
Llama 3.3 70B		21.3	**21.1**	33.3	50
GPT-4o	Real Data	**43.1**	54.7	53.6	47.5
GPT-4o-mini		**52.3**	55.6	54.4	54.55
O3-mini		43.2	46.3	41.2	**39.7**
Gemini 2.5 Pro		38.1	42	39	**32**
Llama 3.3 70B		52.6	53.8	52.8	**37.4**

Table 5: Mean Relative Error (%) across datasets comparing CAIS with baselines (lower is better)

models such as GPT-4o and o3-mini, CAIS produces smaller errors in causal effect estimation compared to the overall baseline approach. However, the baseline model demonstrates comparable performance QR Data and Baseline. This discrepancy primarily stems from incorrect variable selection during model implementation. Real-world datasets tend to be more complex and extensive, often containing multiple interrelated variables. Consequently, despite correct method selection, the choice of inappropriate model variables leads to higher estimation errors. Another notable observation is that the magnitude of differences is smaller for synthetic data. This occurs because synthetic data is generated from well-defined distributions, ensuring that even when model misspecification occurs, the overall estimates do not deviate substantially from true values.

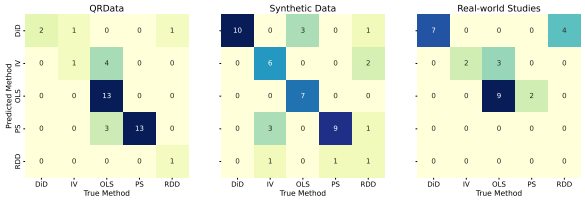

Figure 2: Confusion matrix showing method selection performance of CAIS (GPT-4o).

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

Table 6: Comparison of performance and error types between baseline and CAIS (CAIS). Arrow indicates direction of change from baseline to CAIS.

7 Error Analysis

7.1 Fine Grained Error Analysis

A case-by-case analysis of the outputs reveals several key sources of error:

- **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:** , As demonstrated in Figure 2, LLMs misclassify Randomized Control Trials as Encouragement Designs, leading to the selection of Instrumental Variables instead of linear regression. Similarly, for synthetic datasets, the model failed to identify Instrumental Variables as the optimal method in three instances. This pattern underscores the inherent challenge of selecting valid instruments based solely on data descriptions.

- **Incorrect Data Formats:** Implementation errors also stem from inconsistent data formatting. Specifically, certain variables are encoded as strings when causal inference packages like DoWhy require numerical inputs, creating compatibility issues that compromise execution.

7.2 Overall Error Analysis

We compared overall performance of baseline approaches, which are based on prompting base LLM code generation vs. CAIS's structured approach.

- **Higher Method Selection Accuracy:** CAIS achieves a 46.3% higher method match rate than the baseline (76.2% vs. 52.08%), indicating more accurate identification of appropriate causal methods.

- **Substantial Reduction in Retries:** CAIS reduces total retries by 54.6% per query (in case of CAIS retry refers to feedback via validation loop), suggesting more robust and executable outputs due to structured prompt generation and template-based code execution.

- **Improved Execution Stability:** Execution and runtime errors are reduced by 33.4%, and method mismatches decrease by 28.8%, reflecting enhanced reliability in model reasoning and implementation.

- **No Data Loading Failures:** CAIS handles datasets more reliably with 0% data loading failures compared to 3.1% in the baseline.

- **Trade-offs in Estimation Quality:** While CAIS increases mean error slightly (from 35.38% to 37.66%), this may stem from using more advanced methods rather than defaulting to simple linear regression.

8 Related Work

LLMs and causal effect estimation The use of large language models (LLMs) for causal effect estimation in tabular datasets has been explored in Liu et al. (2024) and Chen et al. (2025), though both approaches require users to specify the estimation method and relevant variables. Jiang et al. (2024) introduce a fine-tuned model enabling end-to-end estimation but focus on only two causal effect measures. Causal Copilot (Wang et al., 2025) expands the range of supported methods using general foundation models but has primarily

been evaluated on causal discovery tasks. Another approach leverages LLMs' existing knowledge to build causal graphs from variable names (Kiciman et al., 2024; Han et al., 2024) and apply front-door or back-door criteria (Pearl, 2009). Beyond tabular data, LLMs have been applied to causal estimation with text data (Dhawan et al., 2024; Lin et al., 2023; Imai and Nakamura, 2024; Veljanovski and Wood-Doughty, 2024).

LLMs and causal structure learning Several approaches have explored the applicability of large language models (LLMs) in causal discovery. These include methods for identifying the causal ordering (i.e., the topological order of variables in a causal graph) (Vashishtha et al., 2023), integrating prompting with data-driven techniques (Ban et al., 2023), leveraging LLMs as experts to infer edges in a causal graph (Long et al., 2022), and designing efficient full-graph discovery algorithms (Jiralerspong et al., 2024). Other works construct domain-specific probabilistic models to evaluate causal relations (Choi et al., 2022) or use variable names to build causal graphs (Kiciman et al., 2024). Beyond discovery, several studies investigate causal reasoning with LLMs. These include testing whether models can identify cause-effect relations from textual descriptions (Gao et al., 2023; Han et al., 2024), or distinguish causation from correlation in statements (Jin et al., 2024). Jin et al. (2023) develop a framework for formal causal reasoning grounded in Pearl's structural causal model framework (Pearl, 2009; Peters et al., 2017). Comprehensive evaluations of LLMs' capabilities in causal inference across a range of prompting strategies and tasks are presented in Zhou et al. (2024) and Chen et al. (2024).

LLMs-powered data analysis Several works study LLM code generation capabilities in data science, including machine learning, statistical analysis, and data manipulation (Huang et al., 2022; Lai et al., 2023; Cheng et al., 2023; Nejjar et al., 2024; Jansen et al., 2023). Wu et al. (2024) extend this approach by enabling LLM-powered tools to perform statistical reasoning and generate data-driven solutions to natural language queries. Another promising direction is development if LLM-powered agents that perform end-to-end analysis, including data preprocessing, model selection, and evaluation. Some tools focus on machine learning tasks (Zhang et al., 2023, 2024; Huang et al., 2024), while others emphasize broader data analysis (Guo et al., 2024; Hong et al., 2024). Recent enhancements include Case-Based Reasoning (Guo et al., 2024), Hierarchical Decomposition (Hong et al., 2024), and interactive tools (Wu et al., 2023). However, these agents do not focus on causality-based analysis, which requires fundamentally different methodological considerations. Our work addresses this gap by developing an agent specifically designed for causal effect estimation.

Evaluation benchmarks include DS-1000 (Lai et al., 2023) and ARCADE (Yin et al., 2022) for code generation, and StatQA (Zhu et al., 2024) and DACO (Wu et al., 2024) for answering data-driven natural language queries. More comprehensive benchmarks evaluate general data analysis tasks (Hu et al., 2024) and machine learning tasks (Huang et al., 2024). Blade (Gu et al., 2024) and DiscoveryBench (Majumder et al., 2024) focus on scientific hypothesis evaluation in a more open-ended fashion.

9 Conclusion

In this work, we introduce Causal AI Scientist (CAIS), an end-to-end framework that maps natural language queries and datasets to formal causal estimation tasks, automatically selecting appropriate methods and interpreting results. We evaluate CAIS across diverse causal inference tasks using three dataset types: QRData, synthetic, and real-world studies. CAIS consistently outperforms baseline prompting strategies on method selection and achieves competitive performance on causal effect estimation, particularly on structured datasets like QRData and synthetic examples.

These strong results underscore the value of CAIS's structured decision-tree-based approach, which decomposes complex reasoning into interpretable steps. This guided approach not only improves estimation accuracy but also enhances robustness and transparency critical qualities for researchers and practitioners in social science, healthcare, economics, and related fields. The framework's reliable performance on well-structured datasets suggests that real-world results can be further improved with better data preprocessing, reinforcing the broader utility of CAIS as a trustworthy tool for domain experts seeking accessible and interpretable causal analysis.

10 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 CAIS 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

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.

Qiang Chen, Tianyang Han, Jin Li, Ye Luo, Yuxiao Wu, Xiaowei Zhang, and Tuo Zhou. 2025. Can ai master econometrics? evidence from econometrics ai agent on expert-level tasks. *Preprint*, arXiv:2506.00856.

Sirui Chen, Bo Peng, Meiqi Chen, Ruiqi Wang, Mengying Xu, Xingyu Zeng, Rui Zhao, Shengjie Zhao, Yu Qiao, and Chaochao Lu. 2024. Causal evaluation of language models. *Preprint*, arXiv:2405.00622.

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

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*.

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*.

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

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*.

Jinglong Gao, Xiao Ding, Bing Qin, and Ting Liu. 2023. Is chatGPT a good causal reasoner? a comprehensive evaluation. In *The 2023 Conference on Empirical Methods in Natural Language Processing*.

Thomas A. Glass, Steven N. Goodman, Miguel A. Hernán, and Jonathan M. Samet. 2013. Causal inference in public health. *Annual review of public health*, 34:61–75.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, and 1 others. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

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.

Siyuan Guo, Cheng Deng, Ying Wen, Hechang Chen, Yi Chang, and Jun Wang. 2024. Ds-agent: Automated data science by empowering large language models with case-based reasoning. *Preprint*, arXiv:2402.17453.

Kairong Han, Kun Kuang, Ziyu Zhao, Junjian Ye, and Fei Wu. 2024. Causal agent based on large language model. *Preprint*, arXiv:2408.06849.

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

Paul W. Holland. 1986. Statistics and causal inference. *Journal of the American Statistical Association*, 81(396):945–960.

Sirui Hong, Yizhang Lin, Bang Liu, Bangbang Liu, Binhao Wu, Ceyao Zhang, Chenxing Wei, Danyang Li, Jiaqi Chen, Jiayi Zhang, Jinlin Wang, Li Zhang, Lingyao Zhang, Min Yang, Mingchen Zhuge, Taicheng Guo, Tuo Zhou, Wei Tao, Xiangru Tang, and 8 others. 2024. Data interpreter: An llm agent for data science. *Preprint*, arXiv:2402.18679.

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

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.

Qian Huang, Jian Vora, Percy Liang, and Jure Leskovec. 2024. Mlagentbench: Evaluating language agents on machine learning experimentation. *Preprint*, arXiv:2310.03302.

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

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

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

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

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.

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*.

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*.

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

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.

Samantha Kleinberg and George Hripcsak. 2011. Methodological review: A review of causal inference for biomedical informatics. *J. of Biomedical Informatics*, 44(6):1102–1112.

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.

Victoria Lin, Louis-Philippe Morency, and Eli Ben-Michael. 2023. Text-transport: Toward learning causal effects of natural language. *Preprint*, arXiv:2310.20697.

Xiao Liu, Zirui Wu, Xueqing Wu, Pan Lu, Kai-Wei Chang, and Yansong Feng. 2024. 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.

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

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*.

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.

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).

OpenAI, Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, Red Avila, Igor Babuschkin, Suchir Balaji, Valerie Balcom, Paul Baltescu, Haiming Bao, Mohammad Bavarian, Jeff Belgum, and 262 others. 2024. Gpt-4 technical report. *Preprint*, arXiv:2303.08774.

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

J. Peters, D. Janzing, and B. Schölkopf. 2017. *Elements of Causal Inference: Foundations and Learning Algorithms*. MIT Press, Cambridge, MA, USA.

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

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

Gemini Team. 2024. Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context. *Preprint*, arXiv:2403.05530.

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.

Marko Veljanovski and Zach Wood-Doughty. 2024. DoubleLingo: Causal estimation with large language models. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 2: Short Papers)*, pages 799–807, Mexico City, Mexico. Association for Computational Linguistics.

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.

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

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*.

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

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. 2023b. React: Synergizing reasoning and acting in language models. *Preprint*, arXiv:2210.03629.

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

Bin Yu. 2020. Veridical data science. In *Proceedings of the 13th International Conference on Web Search and Data Mining*, WSDM '20, page 4–5, New York, NY, USA. Association for Computing Machinery.

Lei Zhang, Yuge Zhang, Kan Ren, Dongsheng Li, and Yuqing Yang. 2024. Mlcopilot: Unleashing the power of large language models in solving machine learning tasks. *Preprint*, arXiv:2304.14979.

Shujian Zhang, Chengyue Gong, Lemeng Wu, Xingchao Liu, and Mingyuan Zhou. 2023. Automl-gpt: Automatic machine learning with gpt. *Preprint*, arXiv:2305.02499.

Yu Zhou, Xingyu Wu, Beicheng Huang, Jibin Wu, Liang Feng, and Kay Chen Tan. 2024. Causalbench: A comprehensive benchmark for causal learning capability of llms. *Preprint*, arXiv:2404.06349.

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

A Appendix

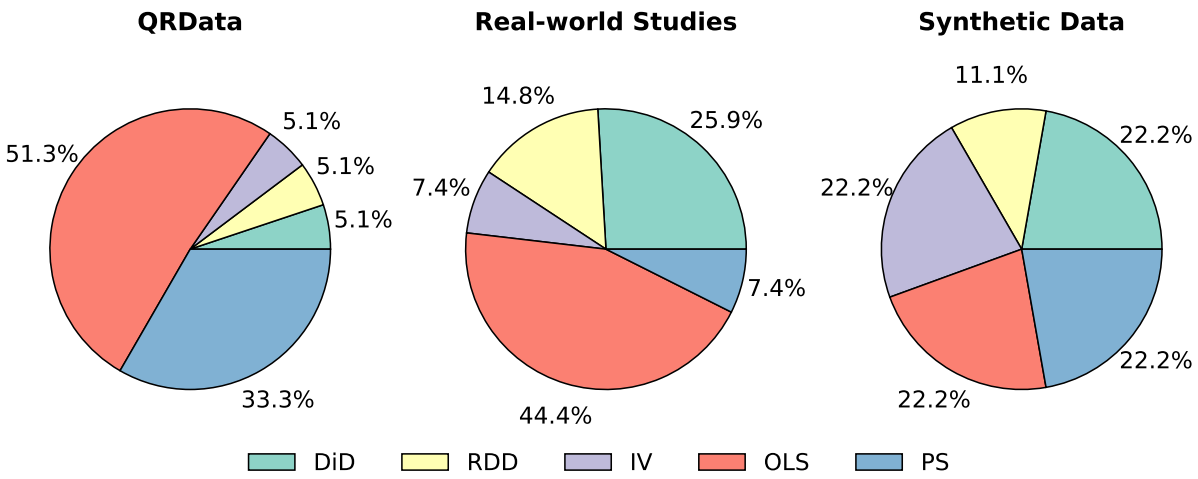

Figure 3: Distribution of estimation methods across the three dataset collections

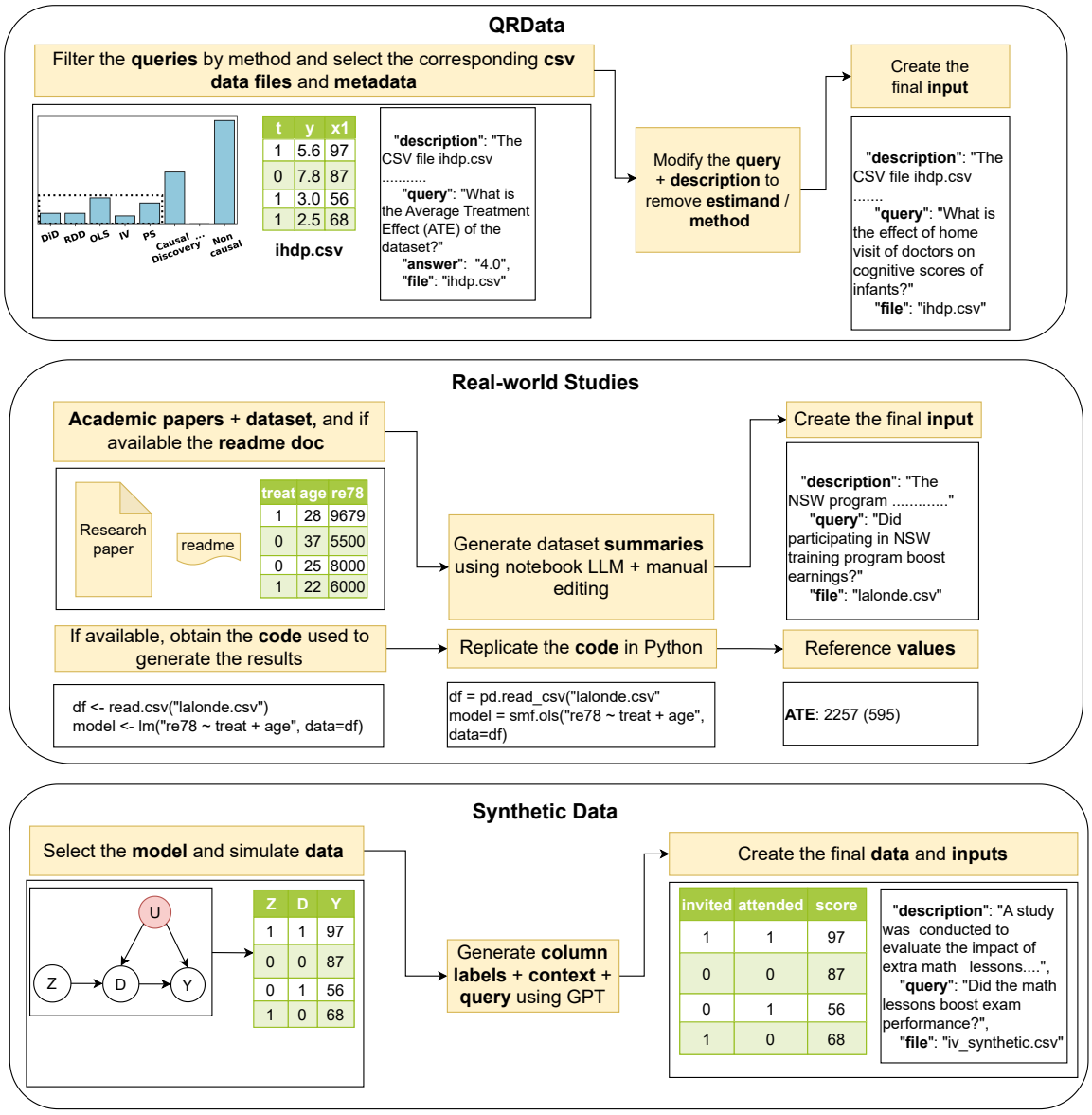

Figure 4: Dataset creation process for **QRData, Real-world Studies, and Synthetic Data**

B Decision Tree for Model Selection

C Detailed Study: Method Validation Loop

D Baseline Prompts

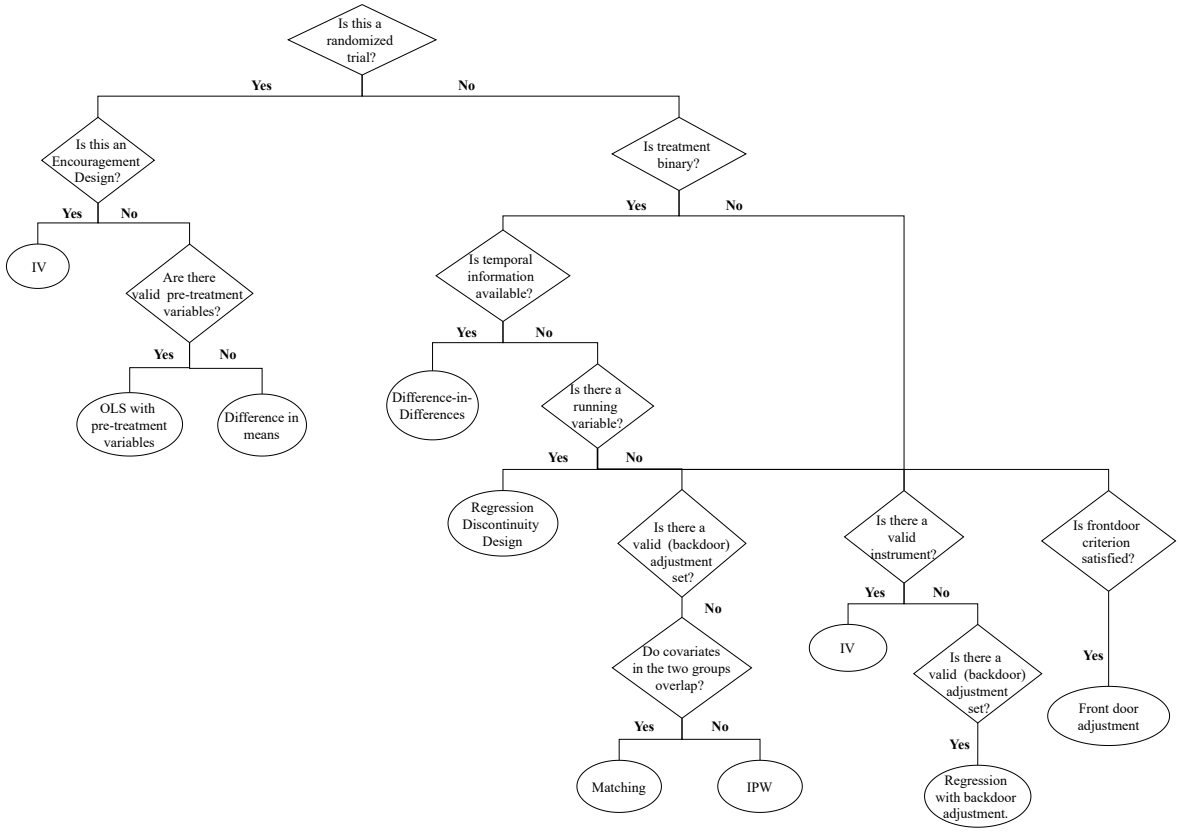

Figure 5: Decision-tree that guides method selection in CAIS. We prompt an LLM to generate responses to queries corresponding to the decision nodes, and traverse the tree accordingly before reaching a leaf node, which corresponds a method

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.

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

ReACT Prompt Example

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

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

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

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

Instructions:

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

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

Figure 7: Example of a ReACT-style prompt used in baseline prompting.

Veridical Prompt

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

Figure 8: Veridical Style Prompting .