OrcheCause Agent: From Textual Knowledge to End-to-End Causal Inference

Jinseok Yang*, Jung-Hee Kim*, Juhyun Lyu*, Soonyoung Lee, Woohyung Lim

LG AI Research

{jinseok.yang, junghee.kim, enriquelyu, soonyoung.lee, w.lim}@lgresearch.ai

Abstract

Causal agents have emerged as promising tools for automating causal analysis based on user queries. However, existing causal agent systems are often limited to a single causal task, limiting their ability to handle complex queries. In addition, they accept only numerical data as input, preventing the integration of domain knowledge expressed in natural language. To overcome these limitations, we propose the OrcheCause agent, a causal agent leveraging textual knowledge for end-to-end causal inference. Specifically, OrcheCause is designed to orchestrate a sequence of interrelated causal tasks in response to user queries. Furthermore, OrcheCause supports diverse data types—numerical as well as textual data—by extracting cause-effect pairs from the relevant sources and incorporating them into causal discovery (CD), thereby improving the performance of CD. OrcheCause also introduces a metric-based hyperparameter optimization framework for CD when ground-truth graphs are not available.

Introduction

Causal inference seeks to answer questions—such as "What features are most important for predicting the outcome?"—from observational data (Imbens 2024). In practice, it requires a multi-step pipeline that includes data preparation, algorithm selection, and causal tasks such as causal discovery (CD) and causal effect estimation (CEE). However, because the multi-step pipeline involves diverse algorithms with different underlying assumptions, it is vulnerable to assumption violations, thereby hindering its practical application. This highlights the need for an agent framework to automate and orchestrate causal analysis (Wang et al. 2025).

Recent advances in causal agents have addressed some challenges. Causal-Copilot provides an end-to-end causal agent framework for tabular and time-series data, supporting causal tasks such as CD and CEE (Wang et al. 2025). In addition, multi-agent frameworks using LLMs have been introduced to enhance the performance of traditional statistical CD (Shen et al. 2025; Le, Xia, and Zhang 2024). However, these systems are limited to fixed tasks—mainly CD or CEE—and cannot orchestrate task sequences in response to user queries. They also rely solely on numerical data, restricting the integration of domain knowledge expressed in natural language.

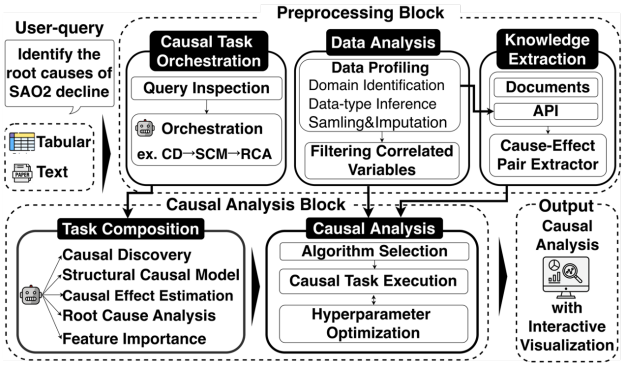

Figure 1: Overall workflow of OrcheCause. The causal analysis executes the task flow planned by orchestration and integrates prior knowledge from textual and external sources.

To address these limitations, we propose a new causal analysis agent, called OrcheCause. The main contributions of OrcheCause are summarized as follows:

- **Causal task orchestration** – We introduce, for the first time, an LLM-based module that dynamically adapts to user queries, ensuring that the most relevant causal analysis steps are executed.
- **LLM-based textual knowledge extraction** – We extract prior knowledge from relevant textual and external sources using LLMs, enhancing the accuracy of CD.
- **Hyperparameter optimization (HPO) without ground-truth (GT) via metric guidance** – OrcheCause provides a metric-based HPO module for CD, enabling robust hyperparameter selection in the absence of GT by leveraging metrics such as Bayesian Information Criterion (BIC) (Chickering 2020).

System Design and Implementation

As shown in Figure 1, OrcheCause comprises four core modules—**causal task orchestration**, **data analysis**, **knowledge extraction**, and **causal analysis**—linked by an interactive chat-based UX layer.

*These authors contributed equally.

Copyright © 2026, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

Causal task orchestration This module enables flexible, multi-task causal analysis, in contrast to prior systems restricted to single-task execution. It consists of two LLM-based submodules: query inspection and task classification.

- **Query inspection**: Identifies user query's intent and classifies it into four categories: (1) causal modeling (e.g., "Discover causal relations among variables."), (2) explanation and attribution (e.g., "Identify root cause of SAO2."), (3) explanation of causal tools (e.g., "What is root cause analysis (RCA)?";answered via search), and (4) non-causal tasks (prompt user re-request).
- **Task classification**: Applies only to queries classified as (1) or (2), and determines the sequence of tasks to execute. For example, when a user query is identified as an RCA task, it triggers the pipeline CD $\rightarrow$ SCM $\rightarrow$ RCA.

Data Analysis This module validates dataset suitability: (i) domain identification to search relevant external databases, (ii) data type detection, (iii) check sample size and impute missing values, (iv) correlation checks to remove the highly correlated features for robust CD. The dataset analysis results and availability of prior knowledge integration guide the selection of CD algorithms.

Knowledge integration This module extracts cause-effect (CE) pairs from text and external databases, which are then applied as constraints in CD algorithms. This enables OrcheCause to combine numerical and textual data. It leverages two complementary sources of knowledge:

- **User-provided text**: To identify causal relationships from the text, an LLM infers the directionality between each CE pair. To extract CE pairs more precisely, we adopt the following principles (Halpern 2016): (i) causes must precede effects, (ii) entities and scenarios must align, (iii) causes should be direct and plausible, and (iv) avoid restated, post-effect, overly abstract, or implausible causes. This procedure is also applied to knowledge from external sources. The CE pairs from the user-provided text are applied as hard constraints, strengthening the reliability of causal analysis.
- **External knowledge database**: Based on the domain of the dataset, OrcheCause queries relevant databases (e.g., PubMed, Pathway Commons), extracts CE pairs, and applies them as soft constraints, thereby complementing data-driven inference with domain knowledge.

Causal analysis This module executes a sequence of tasks planned by the causal task orchestration module.

- **Causal inference algorithms** – OrcheCause integrates algorithms for CD, CEE, structural causal model (SCM), feature importance (FI), and RCA, drawing from open-source libraries such as causal-learn (Zheng et al. 2024), DoWhy (Sharma and Kiciman 2020), pgmpy (Ankan and Textor 2024), and causalml (Chen et al. 2020). As noted, data analysis guides the selection of CD algorithms, ensuring the application of the most suitable methods.
- **HPO without GT via metric guidance** – In the absence of a GT causal graph, this module employs GT-free evaluation metrics such as BIC (Chickering 2020) to assess candidate causal graphs. It explores combinations of knowledge sources and hyperparameters to determine the best-performing configuration.

Alarm	Baseline	HPO	DK+HPO	DK+EK+HPO
SHD ↓	13	13	10	10
Precision ↑	0.974	0.974	0.975	0.975
Recall ↑	0.884	0.884	0.906	0.906
F1 ↑	0.927	0.927	0.940	0.940
Sachs	**Baseline**	**HPO**	**DK+HPO**	**DK+EK+HPO**
SHD ↓	21	23	14	10
Precision ↑	0.417	0.375	0.647	0.846
Recall ↑	0.263	0.316	0.578	0.578
F1 ↑	0.323	0.343	0.611	0.688

Table 1: Ablation study on Alarm and Sachs datasets.

Interactive UX A chat-based interface lets users request causal tasks in natural language and receive results in real-time results for interactive analysis.

Demonstration

The effectiveness of OrcheCause was validated through experiments on the Alarm dataset (Beinlich et al. 1989) and Sachs dataset (Sachs et al. 2005). Domain knowledge (DK) was taken from the original dataset papers (Beinlich et al. 1989; Sachs et al. 2005), while external knowledge (EK) was obtained from PubMed for Alarm (US National Library of Medicine 2025) and from Pathway Commons for Sachs (Rodchenkov et al. 2020).

As an illustrative example, consider a user query requesting an RCA task, OrcheCause plans CD $\rightarrow$ SCM $\rightarrow$ RCA within the orchestration module. After data analysis, CE pairs are extracted from the relevant sources and integrated into CD algorithms. The causal tasks are then executed sequentially, and HPO is applied to optimize the CD performance. Finally, the downstream RCA task is performed via SCM fitting. Follow-up queries can trigger additional tasks (e.g., CEE). For instance, users can analyze the causal effect of a treatment on a target based on the RCA results.

Table 1 shows the ablation study of OrcheCause's CD algorithm on the Sachs and Alarm datasets. We report results using the PC algorithm as a representative case. The baseline uses default CD parameters from the library; HPO tunes CD with the BIC score; DK+HPO adds DK before HPO; and DK+EK+HPO incorporates both DK and EK prior to HPO. The results indicate that CD performance improves progressively as each component is added.

Conclusions & Future Works

In this demonstration, we introduced OrcheCause, the first causal AI agent that performs composite causal task orchestration and integrates textual knowledge. We also presented the metric-based HPO in the absence of GT. For future work, we aim to extend the framework to ensure robust CD in high-dimensional, low-sample settings, and enable the analysis of time-series data. We also plan to advance the knowledge extraction module to better handle real-world documents.

References

Ankan, A.; and Textor, J. 2024. pgmpy: A Python Toolkit for Bayesian Networks. *Journal of Machine Learning Research*, 25(265): 1–8.

Beinlich, I. A.; Suermondt, H. J.; Chavez, R. M.; and Cooper, G. F. 1989. The ALARM Monitoring System: A Case Study with Two Probabilistic Inference Techniques for Belief Networks. In *Proceedings of the 2nd European Conference on Artificial Intelligence in Medicine*, 247–256.

Chen, H.; Harinen, T.; Lee, J.-Y.; Yung, M.; and Zhao, Z. 2020. CausalML: Python Package for Causal Machine Learning. arXiv:2002.11631.

Chickering, M. 2020. Statistically Efficient Greedy Equivalence Search. In *Proceedings of the 36th Conference on Uncertainty in Artificial Intelligence (UAI)*, volume 124 of *Proceedings of Machine Learning Research*, 241–249. PMLR.

Halpern, J. Y. 2016. *Actual Causality*. The MIT Press. ISBN 0262035022.

Imbens, G. W. 2024. Causal inference in the social sciences. *Annual Review of Statistics and Its Application*, 11(1): 123–152.

Le, H. D.; Xia, X.; and Zhang, C. 2024. Multi-Agent Causal Discovery Using Large Language Models. arXiv:2407.15073.

Rodchenkov, I.; Babur, O.; Luna, A.; Aksoy, B. A.; Wong, J. V.; Fong, D.; Franz, M.; Siper, M. C.; Cheung, M.; Wrana, M.; Mistry, H.; Mosier, L.; Dlin, J.; Wen, Q.; O'Callaghan, C.; Li, W.; Elder, G.; Smith, P. T.; Dallago, C.; Cerami, E.; Gross, B.; Dogrusoz, U.; Demir, E.; Bader, G. D.; and Sander, C. 2020. Pathway Commons 2019 Update: integration, analysis and exploration of pathway data. *Nucleic Acids Research*, 48(D1): D489–D497.

Sachs, K.; Perez, O.; Pe'er, D.; Lauffenburger, D. A.; Nolan; and P, G. 2005. Causal Protein-Signaling Networks Derived from Multiparameter Single-Cell Data. *Science*, 308(22): 523–529.

Sharma, A.; and Kiciman, E. 2020. DoWhy: An End-to-End Library for Causal Inference. *arXiv preprint arXiv:2011.04216*.

Shen, C.; Chen, Z.; Luo, D.; Xu, D.; Chen, H.; and Ni, J. 2025. Exploring Multi-Modal Data with Tool-Augmented LLM Agents for Precise Causal Discovery. In *Findings of the Association for Computational Linguistics*, 636–660.

US National Library of Medicine. 2025. PubMed. https://pubmed.ncbi.nlm.nih.gov/. Accessed: 2025-09-16.

Wang, X.; Zhou, K.; Wu, W.; Singh, H. S.; Nan, F.; Jin, S.; Philip, A.; Patnaik, S.; Zhu, H.; Singh, S.; Prashant, P.; Shen, Q.; and Huang, B. 2025. Causal-Copilot: An Autonomous Causal Analysis Agent. Technical report, University of California San Diego.

Zheng, Y.; Huang, B.; Chen, W.; Ramsey, J.; Gong, M.; Cai, R.; Shimizu, S.; Spirtes, P.; and Zhang, K. 2024. Causal-learn: Causal discovery in python. *Journal of Machine Learning Research*, 25(60): 1–8.