An end-to-end Causal Modeling Framework for Advanced Attribution in Supply Chain Operations

Pavan Nithin Mullapudi
pavmul@amazon.com
Amazon
Seattle, WA, USA

Sreyoshi Bhaduri
drsre@amazon.com
Amazon
New York, NY, USA

Alessandro Casadei
acasadei@amazon.com
Amazon
Seattle, WA, USA

Lu Guo
luguo@amazon.com
Amazon
Seattle, WA, USA

Hungjen Wang
hungjen@amazon.com
Amazon
New York, NY, USA

Vykunth Ashok
vykunth@amazon.com
Amazon
San Fransisco, CA, USA

Rohit Malshe
malshe@amazon.com
Amazon
Seattle, WA, USA

Abhilasha Katariya
abhkata@amazon.com
Amazon
Seattle, WA, USA

Ankush Pole
ankupole@amazon.com
Amazon
Seattle, WA, USA

Arkajit Rakshit
rakshit@amazon.com
Amazon
Seattle, WA, USA

Abstract

Effective attribution of causes to outcomes is crucial for optimizing complex supply chain operations. Traditional methods, often relying on waterfall logic or correlational analysis, frequently fall short in identifying the true drivers of performance issues. This paper proposes a comprehensive framework leveraging data-driven causal discovery to construct and validate Structural Causal Models (SCMs). We contrast this approach with baseline models derived from existing business definitions or metric-guided Large Language Models (LLMs). The core methodology involves (1) discovering a Directed Acyclic Graph (DAG) from observational data using the PC (Peter-Clark) algorithm, (2) comparing it to a baseline DAG, (3) building SCMs from these DAGs using DoWhy's GCM module, (4) rigorously validating both DAGs (via falsification tests) and SCMs (via mechanism and model fit evaluations), and (5) utilizing the validated SCM to perform advanced causal queries—including root cause attribution, intervention analysis, and counterfactual reasoning. We illustrate the framework's superiority over traditional methods through its application to a supply chain KPI, demonstrating how it provides deeper, actionable insights. Results suggest that data-driven SCMs, when properly validated, offer more robust and nuanced attribution than simpler rule-based or purely qualitative models. Our results maintain analytical accuracy while utilizing representative metrics instead of proprietary organizational data.

Keywords

Causal Discovery, Structural Causal Models, Supply Chain Attribution, DoWhy, GCM, Falsification Tests, LLM

ACM Reference Format:
Pavan Nithin Mullapudi, Sreyoshi Bhaduri, Alessandro Casadei, Lu Guo, Hungjen Wang, Vykunth Ashok, Rohit Malshe, Abhilasha Katariya, Ankush Pole, and Arkajit Rakshit. 2025. An end-to-end Causal Modeling Framework for Advanced Attribution in Supply Chain Operations. In *Proceedings of the 1st Workshop on "AI for Supply Chain: Today and Future" @ 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V.2 (KDD '25), August 3, 2025, Toronto, ON, Canada.* KDD, Toronto, CA, 7 pages.

1 Introduction

In the intricate web of modern supply chains, pinpointing the precise causes of operational inefficiencies, delays, or cost overruns is a formidable challenge. Traditional business intelligence often relies on descriptive analytics, correlational studies, or pre-defined "waterfall" logic for attribution. Waterfall attribution typically involves a hierarchical, rule-based assignment of responsibility based on a sequence of checks, which may not capture complex interactions, feedback loops (when unrolled over time), or confounding factors inherent in dynamic systems [1]. This involves assumption of independence among causes, reliance on binary causal attribution (cause applies: Yes, No) and dependence on heuristic thresholds to determine if a cause applies. Moreover, waterfall logic cannot simulate what-if interventions—questions.

These limitations motivate a shift to Structural Causal Models (SCMs), which represent causal relationships as a system of structural equations guided by a causal Directed Acyclic Graph (DAG) [1]. Unlike waterfall rules, SCMs capture joint and conditional dependencies among all variables, assign continuous causal attributions, and support counterfactual and interventional queries.

KDD '25, Toronto, ON, Canada.

This paper introduces an end-to-end framework for building and utilizing validated SCMs for causal attribution in supply chains. Our primary approach focuses on data-driven causal discovery using algorithms like the PC algorithm followed by the use of LLMs to validate edges and break cyclicality [2] to learn the DAG structure from observational data. We also consider a comparative baseline approach where a DAG might be derived from existing business rules, domain expertise, or guided Large Language Models (LLMs) using predefined definitions of metrics to articulate current understanding of the relationships between metrics [3, 6].

A cornerstone of our framework is rigorous validation at multiple stages: the discovered DAGs are subjected to falsification tests against data [4], and the subsequently constructed SCMs are evaluated for their mechanistic plausibility and fit [5]. By focusing on a well-validated, data-driven SCM, the framework aims to provide more reliable causal insights than traditional methods or less rigorously validated models. We illustrate how this framework can be applied to understand the drivers of a critical supply chain KPI and demonstrate a variety of causal analyses (e.g., root cause attribution, impact of interventions, counterfactual reasoning) that are often intractable with simpler attribution models. This work highlights the practical utility of modern causal inference tools like DoWhy and its GCM module [7, 8] in moving beyond correlation to causation for tangible operational improvements.

The contributions of this paper are:

(1) A comprehensive, multi-stage framework for data-driven causal discovery, SCM construction, and robust validation for attribution in supply chains.
(2) A comparative discussion highlighting the advantages of SCM-based attribution over traditional waterfall logic and simpler LLM-derived baselines.
(3) An illustration of the framework's application to a complex supply chain KPI, showcasing its ability to answer diverse causal questions and provide actionable insights.

2 Background and Related Work

2.1 Limitations of Traditional Attribution

Traditional attribution in business settings, particularly "waterfall logic," assigns causality based on a predefined, often linear, sequence of checks or rules. For example, if an order is late, a waterfall might first check for supplier delays; if none, then check for warehouse processing time; if normal, then check for transport issues. While simple to implement, this approach struggles with:

- **Confounding:** It often fails to account for common causes affecting multiple stages.
- **Interactions:** It may not identify situations where multiple factors jointly cause an outcome.
- **Feedback:** It typically cannot model cyclical relationships (though SCMs represent DAGs, which are acyclic, they can model systems that have feedback when unrolled over time or by representing equilibrium states).
- **Quantification of Effects:** It usually provides a binary "responsible/not responsible" rather than quantifying the magnitude of a cause's impact.

These limitations motivate the need for more sophisticated causal modeling.

2.2 Causal Discovery from Data

The PC algorithm [2] is a well-known constraint-based method for learning the structure of a DAG from observational data. It operates by systematically performing conditional independence tests to identify the "skeleton" of the graph and then orienting edges to respect the identified independencies and avoid cycles. Key assumptions include causal sufficiency (no unobserved confounders) and faithfulness (observed independencies reflect the true causal graph). Libraries like 'causal-learn' offer practical implementations [10].

2.3 LLMs in Causal Discovery

LLMs are being explored for their potential to contribute to causal discovery by leveraging their embedded knowledge [3, 12]. Approaches include direct prompting for causal links [13] or using LLMs to generate priors that can be combined with data-driven methods [6]. The idea of using "metric definitions" suggests guiding LLM prompts with specific criteria (e.g., related to known mechanisms, impact types, or temporal precedence) to elicit more structured or reliable causal hypotheses. This can serve as a way to formalize existing business understanding into a baseline causal graph.

2.4 Structural Causal Models (SCMs)

An SCM defines a causal system through a DAG and a set of structural equations $X_i = f_i(PA_i, N_i)$ for each variable X_i, where PA_i are its direct causes (parents in the DAG) and N_i are exogenous noise terms [1]. SCMs explicitly model the mechanisms by which causes generate effects. DoWhy's GCM module [8] allows these functions f_i to be diverse, including linear models, non-linear machine learning models (e.g., within Additive Noise Models - ANMs), or custom functions, and facilitates fitting these SCMs to data.

2.5 Validation of Causal Models

2.5.1 DAG Falsification. A hypothesized DAG implies specific conditional independencies. Falsification tests (e.g., `dowhy.gcm.falsify_graph` [4]) compare these model-implied independencies with those observed in the data. Significant deviations can lead to the rejection of the DAG.

2.5.2 SCM Evaluation. Once an SCM is fitted (i.e., functional forms f_i are chosen and parameters estimated), its validity needs assessment. This includes:

- **Mechanism Fit:** How well does each equation $X_i = f_i(PA_i, N_i)$ predict X_i given its parents? For ANMs, this involves checking if the residuals N_i are independent of PA_i.
- **Overall Model Fit:** Does the SCM adequately reproduce the observed joint distribution of the data? This involves a graph falsification test to assess the probability the given DAG is equivalent to a randomly generated one (see *Figure* 3)
- **Sensitivity Analysis:** How robust are the SCM's conclusions to changes in assumptions or parameters [11]?

The `dowhy.gcm.evaluate_causal_model` function provides tools for assessing some of these aspects, such as the performance of

individual mechanisms and the overall consistency of the GCM with the data [5].

3 Proposed Causal Attribution Framework

Our framework (conceptually depicted in Figure 1) aims to provide a robust methodology for causal attribution, particularly in complex domains like supply chains.

Figure 1: Conceptual End-to-End Causal Attribution Framework

(1) Observational Data Input
(2) Parallel DAG Discovery:
  - Path A: Data-Driven (e.g., PC Algorithm) → DAG_{Data}
  - Path B: LLM-Metric/Business Rule Based → DAG_{LLM} (Baseline)
(3) SCM Construction (for DAG_{Data} & DAG_{LLM}) using DoWhy GCM
(4) Validation Phase:
  - DAG Falsification (on DAG_{Data} & DAG_{LLM})
  - SCM Evaluation (on SCM_{Data} & SCM_{LLM})
(5) Selection of Validated SCM (primarily SCM_{Data} if superior)
(6) Advanced Causal Queries (Attribution, Intervention, Counterfactuals, etc.)
(7) Actionable Insights & Decisions

Figure 1: Overview of the proposed end-to-end causal attribution framework, highlighting parallel DAG discovery, SCM construction, rigorous validation, and advanced causal querying for deriving actionable insights.

3.1 Step 1: Data Acquisition and Preparation

Collect relevant time-series and cross-sectional data pertaining to the supply chain processes and outcomes of interest. This includes identifying key variables, handling missing data, and appropriate transformations.

3.2 Step 2: Parallel DAG Discovery

3.2.1 Path A: Data-Driven DAG Discovery. Employ a data-driven causal discovery algorithm, such as the PC algorithm (e.g., via 'causal-learn' [10]), on the prepared observational dataset. This yields DAG_{Data}, representing statistically inferred causal relationships. Assumptions like causal sufficiency and faithfulness are critical here.

3.2.2 Path B: LLM-Metric/Business-Rule Derived DAG (Baseline). This path aims to formalize existing domain knowledge or current business understanding into a causal graph, DAG_{LLM}.

- **Metric Definition:** Define clear qualitative or quantitative metrics that characterize expected causal links (e.g., "direct physical impact," "information flow dependency," "regulatory constraint").
- **LLM Elicitation:** Use an LLM, prompted with descriptions of system variables and the defined metrics, to suggest or score potential causal relationships. This could involve structured querying for pairwise relationships or more complex path elicitation.
- **Business Rule Codification:** Alternatively, existing documented business process flows or rule-based logic can be translated into a DAG structure.

DAG_{LLM} serves as a baseline representing current hypotheses or a qualitative understanding.

3.3 Step 3: Structural Causal Model (SCM) Construction

For both DAG_{Data} and DAG_{LLM}, construct SCMs using 'dowhy.gcm' [8].

(1) **Mechanism Specification:** For each node X_i in a DAG, assign a causal mechanism $X_i = f_i(PA_i, N_i)$. This can be achieved via `dowhy.gcm.auto.assign_causal_mechanisms`, which selects appropriate models (e.g., Additive Noise Models with linear regression, GPs, or classifiers) based on data types, or by manual specification using domain knowledge.
(2) **Model Fitting:** Fit the specified mechanisms to the observational data using 'dowhy.gcm.fit()' to learn the parameters of the functions f_i and distributions of N_i.

This results in SCM_{Data} and SCM_{LLM}.

3.4 Step 4: Validation

3.4.1 DAG Validation. Both DAG_{Data} and DAG_{LLM} are critically assessed using `dowhy.gcm.falsify_graph()` [4]. This function tests the conditional independencies implied by each DAG against the observational data. Results (e.g., p-values for LMC violations) indicate how consistent each graph structure is with the observed data.

3.4.2 SCM Evaluation. The fitted SCMs (SCM_{Data} and SCM_{LLM}) are then evaluated using `dowhy.gcm.evaluate_causal_model()` [5]. This step assesses the goodness-of-fit of individual causal mechanisms (e.g., residual analysis for ANMs), the validity of modeling assumptions (e.g., noise independence), and the overall ability of the SCM to capture the joint data distribution.

3.5 Step 5: Advanced Causal Querying

The SCM that demonstrates better validation results (SCM_{Data} in this paper's narrative) is then used for in-depth causal analysis and attribution. If DAG_{LLM} captures distinct, domain-critical aspects not statistically evident but deemed important, SCM_{LLM} can be used for qualitative what-if scenarios or to guide further data collection. Key queries include:

- **Root Cause Attribution:** E.g., `dowhy.gcm.attribute_anomalies()` for specific outcomes.
- **Intervention Analysis:** E.g., `dowhy.gcm.interventional_samples()` to predict effects of changes.
- **Counterfactual Reasoning:** E.g., `dowhy.gcm.counterfactual_samples()` for "what-if" on past events.

- **Mediation Analysis:** Decomposing effects into direct and indirect pathways. This is especially valuable in scenarios involving non-controllable factors, such as UpstreamDelays. By identifying mediating variables, we can trace how these non-actionable causes propagate through the system and determine which downstream, controllable factors (e.g., acquiring additional equipment to reduce EquipmentUtilization saturation) can be adjusted to offset their impact.
- **Causal Influence Quantification:** E.g., `dowhy.gcm.arrow_strength()` to measure direct link strengths.

4 Illustrative Application: Analyzing "Capped Out Hours"

We now illustrate the framework by applying it (conceptually) to understand the causes of "Capped Out Hours" in a supply chain logistics hub. "Capped Out Hours" (COH) is defined as the sum of time a station (e.g., a delivery hub) was operating at or above its designated capacity threshold prior to its daily order cutoff time. High COH is undesirable as it often leads to diversion of order volume to more expensive third-party carriers, potential service delays for customers, increased operational costs (overtime, expediting), and an overall negative impact on network efficiency and customer satisfaction. Typical suspected drivers include issues with demand forecasting, inadequacies in tactical capacity cappings, problems with asset/labor utilization and network disruptions.

4.1 Conceptual Experimental Setup

The conceptual experiment aims to identify which factors most significantly contribute to a delivery station operating beyond its planned capacity. By analyzing the relationships between these variables and the outcome, we can gain insights into the key drivers of operational strain and develop strategies to mitigate risks of capacity overruns. Below we introduce such factors.

- **Outcome Variable:** CappedOutHours (continuous, non-negative). This represents the number of hours a delivery station operates beyond its planned capacity, indicating operational strain.
- **Potential Causal Factors (Nodes):**
  - DemandForecastError: Accuracy of predicting daily package volume, measured as Mean Absolute Percentage Error (MAPE). Higher errors may lead to resource misalignment.
  - ActualInboundVolume: The actual number of packages received by the station daily. Unexpected spikes can overwhelm capacity.
  - StationProcessingCapacity: The station's ability to sort and prepare packages for delivery, measured in units per hour. This depends on staffing levels and available equipment.
  - UpstreamDelays: Delays in package arrivals from upstream warehouses, measured in hours. Late arrivals can compress processing time.
  - StaffAvailability: The ratio of actual to scheduled staff hours. Actual hours can be lower due to absenteeism.
  - TacticalCapSetting: Manually set daily limits on volume that can be processed in a warehouse, used to manage workload. May influence the likelihood of exceeding capacity.
  - EquipmentUtilization: The percentage of time equipment is operational. Equipment downtime can significantly impact processing capacity.
  - ExternalEvents: Unexpected occurrences like severe weather or local disruptions that may affect operations. Recorded as binary (occurred/did not occur) or categorical (type of event).
- **Data:** Daily operational data collected from multiple package delivery stations over a one-year period. This includes all variables mentioned above, providing a comprehensive view of factors potentially influencing operational capacity exceedance.

4.2 DAG Generation and Validation Results

4.2.1 DAG Discovery.

- **DAG_{Data}:** The PC algorithm is run on the historical data. (Illustrative DAG shown in 2).
- **DAG_{LLM}:** An LLM is prompted with variable definitions and causal "metric definitions" (e.g., "Metric 1: Is variable X a direct input to the calculation or operational definition of Y?" "Metric 2: Does standard business logic/SOP explicitly state that X must be managed to control Y?"). This results in a baseline DAG reflecting current business understanding or easily articulated hypotheses.

Figure 2: Illustrative Data-Driven DAG for Capped Out Hours Example Structure:

- 'DemandForecastError' → 'TacticalCapSetting'
- 'ActualInboundVolume' → 'CappedOutHours'
- 'StationProcessingCapacity' → 'CappedOutHours'
- 'TacticalCapSetting' → 'CappedOutHours'
- 'StaffAvailability' → 'StationProcessingCapacity'
- 'EquipmentUtilization' → 'StationProcessingCapacity'
- 'UpstreamDelays' → 'ActualInboundVolume' (affecting timing/bunching)
- 'ExternalEvents' → 'TrafficLevel' (unshown intermediate) → 'CappedOutHours' (indirectly)

(An illustrative example)

Figure 2: Illustrative data-driven DAG for factors influencing 'CappedOutHours'. Arrows indicate hypothesized direct causal influences learned via the PC algorithm.

4.2.2 DAG Validation . Figure 3 summarizes the falsification test outcomes. DAG_{Data} shows significantly fewer conditional independence violations (e.g., higher p-value from `falsify_graph`) compared to DAG_{LLM} when tested against the observational data. This suggests DAG_{Data} is more consistent with the statistical patterns in the specific dataset. DAG_{LLM}, while reflecting business logic, might miss some statistical nuances or include links not strongly supported by this particular dataset.

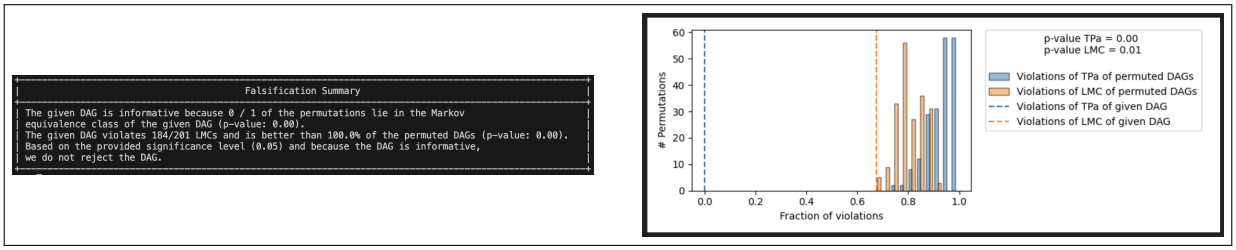

Figure 3: Results of DAG falsification tests.

4.2.3 SCM Evaluation. SCM_{Data} (built from DAG_{Data}) undergoes evaluation using `evaluate_causal_model`. Hypothetically, the mechanisms for key drivers like 'ActualInboundVolume' and 'StationProcessingCapacity' show good fit (e.g., residuals appear random, predictive performance within mechanism is high). SCM_{LLM} show poorer fit for some mechanisms since the underlying links in DAG_{LLM} are not statistically strong.

4.3 SCM-Powered Causal Analysis for COH

Based on stronger validation, SCM_{Data} is chosen for detailed causal querying. Table 1 outlines the types of analyses performed.

4.4 Contrast with Waterfall Attribution

A traditional waterfall approach for COH might sequentially check: 1. Was 'ActualInboundVolume' > 'TacticalCapSetting'? If yes, attribute to volume/caps. 2. If no, was 'StaffAvailability' low? If yes, attribute to staffing. This simplistic logic would miss:

- How 'DemandForecastError' influences 'TacticalCapSetting' and subsequently COH.
- The quantitative impact of a 10% improvement in 'EquipmentUtilization' versus a 10% increase in 'StaffAvailability'.
- The counterfactual scenario of what COH would have been if multiple factors had been different simultaneously.
- Confounding effects, e.g., if 'ExternalEvents' affect both 'UpstreamDelays' and 'StaffAvailability'.

The SCM framework, by modeling the mechanisms and allowing diverse queries (Table 1), provides a far richer and more reliable attribution, leading to more informed interventions. For example, SCM analysis might reveal that while 'ActualInboundVolume' is a direct cause, improving 'DemandForecastError' has a larger total (direct + indirect via 'TacticalCapSetting') effect on reducing COH than previously assumed by simpler models.

5 Discussion

The proposed hybrid framework, culminating in a validated SCM, offers significant advantages over traditional attribution methods like waterfall logic, especially for complex systems such as supply chains. By grounding causal claims in data-driven graph structures and explicit mechanisms, it enables nuanced insights like quantifying interventional impacts and attributing specific anomalies. The data-driven DAG, when rigorously validated, provides a more robust foundation than relying solely on pre-defined business rules or purely qualitative LLM outputs, which served as useful baselines in our conceptual application.

The inclusion of an LLM-driven path, guided by "metric definitions," allows for the incorporation of domain knowledge or existing hypotheses. Even if DAG_{LLM} is found to be less statistically robust than DAG_{Data} (as per our results), it can still highlight areas where business intuition diverges from data patterns, prompting further investigation or targeted data collection. The tactful comparison is key: the LLM/business-defined graph isn't necessarily "wrong" but may be incomplete or less precise than what can be learned from and validated against specific operational data.

Key strengths include:

- **Principled Attribution:** Moves beyond correlation to model causal mechanisms.
- **Quantitative Insights:** Enables prediction of interventional outcomes and quantification of causal strengths.
- **Comprehensive Validation:** Incorporates both DAG falsification and SCM evaluation.
- **Flexibility:** DoWhy GCM supports diverse functional forms for causal mechanisms.

Limitations persist:

- **Data Requirements:** Data-driven discovery requires sufficient, high-quality observational data.
- **Assumptions:** PC algorithm and SCMs rely on assumptions (e.g., causal sufficiency, faithfulness, correct functional forms) that must be carefully considered.
- **LLM Reliability:** LLM outputs can be sensitive to prompting and may reflect biases; "metric definitions" aim to mitigate but not eliminate this.
- **Scalability:** Constructing and validating complex SCMs with many variables can be computationally intensive.

Future work should focus on enhancing the synergy between data-driven and LLM-based approaches, potentially through iterative refinement loops where LLMs help critique or improve data-driven graphs, or where LLM-priors are more formally integrated into data-driven search. Developing more sophisticated SCM validation metrics within accessible tools would also be beneficial.

6 Conclusion

This paper presented a hybrid framework for causal discovery and attribution, emphasizing data-driven DAG learning and SCM construction, validated against observational data and contrasted with baseline models. Applied to a critical supply chain KPI like "Capped Out Hours," the framework demonstrates its capability to move beyond simplistic waterfall attribution, offering a richer, more quantitative, and mechanistically interpretable understanding of

Table 1: SCM Use Cases for Analyzing "Capped Out Hours" (COH) using Validated SCM_{Data}

Causal Query Type	Question for COH	DoWhy GCM Function	Potential Insight
Root Cause Attribution	For days with high COH, which upstream factors (e.g., `DemandForecastError`, `StaffAvailability`) contributed most to COH exceeding a threshold?	`attribute_anomalies(scm, 'CappedOutHours', anomalous_samples)`	Identifies key drivers for specific high COH instances; e.g., "70% of COH anomaly on Day X attributed to low `StaffAvailability`."
Intervention Analysis	What would be the expected change in average COH if 'DemandForecastError' is reduced by 10% through a new system?	'interventional_samples(scm, 'DemandForecastError': lambda x: x*0.9)'	Quantifies impact of planned changes; e.g., "A 10% reduction in forecast error is predicted to reduce average COH by 1.2 hours."
	What is the impact on COH if 'StationProcessingCapacity' is increased by 500 units/hr at a specific station (subgroup)?	Filter data for station, fit SCM (or use conditional SCM if 'Station' is a parent), then 'interventional_samples'.	Station-specific intervention impact; allows targeted capacity adjustments.
Counterfactual Reasoning	On a specific past day with high COH and a known low 'EquipmentUtilization', what would COH have been if utilization was at its 90th percentile target?	'counterfactual_samples(scm, 'EquipmentUtilization': target_value, observed_data=specific_day_data)'	Understands impact of specific past deviations; e.g., "If utilization had been at target, COH on Day Y would likely have been 2.5 hours lower."
Mediation Analysis	How much of the effect of 'UpstreamDelays' on COH is direct, versus mediated through its impact on 'ActualInboundVolume' (timing/bunching)?	(Requires careful setup, potentially using 'dowhy.api.mediation' with SCM outputs or specific GCM mediation if available)	Decomposes total effect into pathways, clarifying mechanisms.
Causal Influence Quantification	What is the direct causal strength of 'TacticalCapSetting' on 'CappedOutHours' compared to other direct parents?	'arrow_strength(scm, 'CappedOutHours')'	Ranks direct drivers by impact magnitude; e.g., "A unit change in 'TacticalCapSetting' has X times the impact on COH variance compared to a unit change in 'StationProcessingCapacity'."

causal drivers. By leveraging tools like DoWhy GCM, this approach allows for a range of advanced causal queries crucial for effective operational decision-making and continuous improvement in complex systems. While the data-driven path, when well-validated, is posited as the more robust foundation for quantitative SCMs, the integration of LLM-based insights (guided by metrics) can provide valuable qualitative context and hypothesis generation.

7 Future Research

Future research should explore hybrid approaches that treat data-driven and LLM-based methods as complementary rather than competing. This need arises from the limitations of the current plethora of validation tests available for evaluating causal models. While each test provides useful insights, they tend to answer very specific questions and fail to assess the overall practical utility of a structural causal model (SCM) in complex real-world settings.

For instance, placebo tests are designed to check whether observed causal effects could be due to random chance—but with large sample sizes (as is typical at Amazon), p-values tend to be near zero, making the test trivially passed regardless of model quality. Conditional independence (CI) tests, on the other hand, evaluate whether a graph fits the data better than random DAGs. However, a DAG can pass these tests even while containing many conditional independence violations that materially impact causal estimates. Additionally, CI test tends to favor PC algorithm generated graphs, as they are built using conditional independence rules. Ultimately, even when several tests are passed, their combined result is unlikely to reflect the true practical effectiveness of the SCM—especially in large-scale, high-dimensional systems.

A promising direction is to use LLMs to generate an initial DAG based on domain knowledge—offering a representation of expert-informed assumptions. This graph can then be refined via conditional independence–based pruning, discarding edges not supported by the data. In initial experiments, however, we observed that pruning can both improve or worsen validation results across iterations. This happens because while pruning edges may reduce the number of CI violations, it also generates new, sparser graphs that introduce more assumed independencies—which themselves must

be validated. This tradeoff suggests that pruning is a non-monotonic process. A more robust path forward could involve combinatorial search algorithms that evaluate falsification metrics across multiple pruning configurations to identify the optimal structure.

Finally, model validation should take a more holistic approach—blending quantitative diagnostics (e.g., placebo tests, CI checks, accuracy measures of causal mechanisms) with qualitative feedback from domain experts. This would allow for critical evaluation of both the plausibility of graph assumptions and the practical usefulness of resulting causal insights.

Acknowledgments

The authors acknowledge the open-source communities behind 'causal-learn' and 'DoWhy' for providing the tools that make such frameworks increasingly accessible.

References

[1] J. Pearl, *Causality: Models, Reasoning and Inference*, 2nd ed. Cambridge, UK: Cambridge University Press, 2009.

[2] P. Spirtes, C. Glymour, and R. Scheines, *Causation, Prediction, and Search*, 2nd ed. Cambridge, MA, USA: MIT Press, 2000.

[3] E. Kiciman, R. Ness, A. Sharma, and C. Tan, "Causal reasoning and large language models: A survey," *arXiv preprint arXiv:2305.00050*, 2023.

[4] PyWhy Authors, "Falsification of User-Given Directed Acyclic Graphs," *DoWhy Documentation*. Accessed: May 13, 2025. [Online]. Available: https://www.pywhy.org/dowhy/v0.10/example_notebooks/gcm_falsify_dag.html

[5] PyWhy Authors, "Evaluate a GCM," *DoWhy Documentation*. Accessed: May 13, 2025. [Online]. Available: https://www.pywhy.org/dowhy/v0.11/user_guide/modeling_gcm/model_evaluation.html

[6] D. Zhang, Y. Liu, J. Li, L. Chen, and P. S. Yu, "LLM-Driven Causal Discovery via Harmonized Prior," *IEEE Transactions on Knowledge and Data Engineering*, pp. 1-15, 2024, doi: 10.1109/TKDE.2024.3386695. (Example, use actual if found)

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

[8] P. Blöbaum, P. Götz, K. Budhathoki, A. A. Mastakouri, and D. Janzing, "DoWhy-GCM: An extension of DoWhy for causal inference in graphical causal models," *Journal of Machine Learning Research*, vol. 25, no. 147, pp. 1–7, 2024.

[9] K. Budhathoki, L. Minorics, P. Blöbaum, and D. Janzing, "Causal structure-based root cause analysis of outliers," in *Proc. International Conference on Machine Learning (ICML)*, 2022, vol. 162, pp. 2357–2369, PMLR.

[10] Y. Zheng, et al., "causal-learn: Causal discovery in Python," 2023. [Online]. Available: https://github.com/cmu-phil/causal-learn (Version 0.1.3.6 used as example)

[11] C. Cinelli and J. Pearl, "Sensitivity analysis of linear structural causal models," *arXiv preprint arXiv:1902.08202*, 2019.

[12] S. Zhang, Y. Liu, L. Li, and J. Zhang, "A Survey on Large Language Models for Causal Inference: Advances and Opportunities," *arXiv preprint arXiv:2402.01106*, 2024. (Example, use most relevant survey found)

[13] W. Jin, R. He, X. L. Li, H. Liu, and J. Han, "Can Large Language Models Build Causal Graphs?" *arXiv preprint arXiv:2306.05180*, 2023. (Example, use most relevant paper found)