Article

Causally Grounded LLM Attribution Agents for High-Dynamic Logistics Systems: Design and Experimental Validation

Sixuan Li [1,*]

[1] McCallum Business School, Bentley University, Waltham, United States
* Correspondence: Sixuan Li, McCallum Business School, Bentley University, Waltham, United States

Abstract: High-dynamic logistics systems frequently generate anomalies due to interacting operational mechanisms like demand surges, driver shortages, and exogenous shocks. While large language models (LLMs) can transform heterogeneous telemetry into natural-language explanations for operator diagnosis, unconstrained language reasoning remains unreliable for root-cause attribution in systems with structured dependencies. To address this, we propose a causally grounded attribution agent architecture integrating a streaming state-preparation layer, a structural causal graph (SCG) to constrain admissible cause-effect paths, a quantitative attribution core, and an LLM reasoning layer. This framework converts grounded evidence into reliable explanations and intervention suggestions. We validate the core components on a controlled synthetic benchmark. The SCG-aligned model achieves a superior macro F1 score of 0.753 on the in-distribution test set and demonstrates robust performance under distribution shifts, outperforming random forest and ungrounded heuristic baselines. Furthermore, a graph misspecification study confirms that the SCG provides critical structural information beyond mere regularization, as removing a single causal edge significantly reduces accuracy. Finally, an LLM evaluation across multiple grounding configurations reveals that full causal grounding improves attribution accuracy by 20 to 35 percentage points, with smaller models benefiting disproportionately. Ultimately, this study contributes a robust, causally grounded agent architecture and a replicable cross-tier evaluation framework for LLM-based causal reasoning, laying the groundwork for future validation on production telemetry and downstream operational impact assessments.

Keywords: causal attribution; causal graphs; logistics analytics; interpretable ai; language models; distribution shift

1. Introduction

Modern logistics and last-mile delivery systems operate in highly dynamic environments where disruptions arise from various interacting sources, including supply-demand imbalances, infrastructure constraints, routing inefficiencies, and external shocks [1]. Identifying the operational mechanisms behind these disruptions in real time is essential for maintaining service reliability and implementing effective interventions.

Recent advancements in large language models (LLMs) have enabled the development of diagnostic systems that synthesize diverse telemetry data and present explanations in natural language. These systems are particularly relevant for logistics organizations with limited analytics resources, where a reasoning agent can reduce the engineering burden of translating raw signals into actionable operational guidance [2]. However, unconstrained LLM reasoning is unreliable in structured environments, as language models may generate plausible but incorrect causal narratives, reverse causal relationships, or recommend actions unsupported by system dynamics.

Classical approaches to causal attribution in operations research and time-series analysis offer complementary strengths [2]. Structural causal models, probabilistic graphical models, and streaming statistical methods provide explicit assumptions,

Received: 13 February 2026
Revised: 05 April 2026
Accepted: 18 April 2026
Published: 23 April 2026

traceability, and uncertainty-aware updates. However, these methods typically produce structured scores or offline analyses rather than operator-facing explanations, creating a gap between causal rigor and practical decision support.

This paper addresses this gap by proposing a causally grounded attribution agent architecture for high-dynamic logistics systems. The design integrates four key components: (i) a streaming state-construction layer that transforms event data into structured causal context, (ii) an SCG-constrained attribution core that evaluates candidate causes under quantified uncertainty, (iii) an LLM reasoning layer that generates natural-language attributions and intervention suggestions based on grounded evidence, and (iv) a closed-loop decision interface that links validated causes to operational actions [3]. In this architecture, the LLM functions as a constrained reasoning layer above a structurally grounded attribution backend, rather than independently inferring causality.

The framework is validated through three experimental layers. First, the attribution backend is benchmarked in a synthetic environment with cross-contaminated causal signatures, demonstrating that SCG alignment provides the strongest stability under distribution shifts [3]. Second, a graph misspecification study systematically removes or adds edges in the SCG, quantifying the sensitivity of attribution performance to graph accuracy. Third, a direct LLM evaluation assesses attribution accuracy, explanation faithfulness, SCG violation rates, and action alignment across four grounding configurations using two models with different capability tiers.

The main contributions are:

1. A causally grounded attribution agent architecture integrating streaming state construction, structural causal constraints, quantitative attribution scoring, LLM-based reasoning, and intervention mapping.
2. Empirical validation on a synthetic benchmark with cross-contaminated signatures, demonstrating that SCG alignment achieves the best stability-accuracy trade-off under distribution shifts.
3. A graph misspecification analysis quantifying the sensitivity of attribution accuracy to edge deletion and spurious edge addition.
4. A cross-tier LLM evaluation comparing two models across four grounding configurations, showing that causal grounding compensates for reduced model capacity.
5. A design specification connecting grounded attribution to intervention suggestions, suitable for future integration into operational decision-support pipelines.

The remainder of this paper is organized as follows. Section 2 reviews related work. Section 3 defines the problem and design requirements. Section 4 presents the proposed method. Section 5 describes the experimental evaluation. Section 6 discusses the results and limitations. Section 7 concludes the paper [1].

2. Related Work

2.1. Causal Inference and Time-Series Causality

The theory of causal inference is built upon two complementary traditions. The structural causal model framework formalizes cause-effect relationships through directed acyclic graphs and the do-calculus, providing a language for interventional reasoning beyond statistical association [4]. The potential-outcomes framework underpins treatment-effect estimation methods widely applied in operations research and experimental design.

In time-series settings, Granger causality evaluates whether lagged values of one series improve the prediction of another [5]. More recent methods address nonlinear temporal data: PCMCI combines condition selection with momentary conditional-independence tests to recover time-lagged causal graphs, while DYNOTEARS formulates temporal structure learning as a continuous optimization problem with acyclicity constraints. In logistics, these time-series causal methods have seen limited adoption due

to operational event streams exhibiting irregular spacing and regime shifts that violate stationarity assumptions.

2.2. LLM-Based Reasoning and Agentic Systems

Large language models have demonstrated strong reasoning capabilities through chain-of-thought prompting, which elicits step-by-step reasoning that improves performance on complex tasks. The ReAct framework interleaves reasoning traces with tool-use actions, enabling these models to operate as autonomous agents that observe, reason, and act in iterative loops [4]. Toolformer demonstrates that such models can learn to invoke external tools to augment their reasoning with structured information.

In domain-specific applications, these agents have been deployed for code generation, scientific discovery, and database querying. However, applications in operational and logistics settings remain scarce. A critical limitation of current agents is their susceptibility to hallucination—generating confident but factually incorrect outputs—which poses severe risks in decision-critical operational environments.

2.3. Grounding LLMs with Structured Knowledge

Several lines of work address the issue of hallucination in large language models (LLMs) through external grounding. Retrieval-augmented generation conditions LLM outputs on retrieved documents, thereby reducing factual errors. Knowledge-graph-grounded generation constrains LLM outputs to align with structured relational knowledge. In the domain of causal reasoning, recent studies have evaluated LLMs' capabilities in causal inference, revealing that while LLMs can identify plausible causal relationships, they often confuse correlation with causation and lack the formal rigor associated with structural causal models [6].

These findings motivate our approach: we provide the LLM with an explicit structural causal graph and uncertainty-quantified state estimates as grounding context, ensuring that reasoning is constrained to causally valid pathways rather than relying solely on the LLM's implicit causal knowledge.

2.4. Bayesian Online Learning and Streaming Inference

Bayesian methods provide a principled framework for sequential belief updating. Streaming variational Bayes updates posterior approximations in a single pass over data. In nonstationary environments, forgetting mechanisms such as exponential discounting of past likelihoods allow the posterior to adapt to distributional shifts. Within fault diagnosis, Bayesian networks have been applied extensively in industrial settings, though typically in offline configurations [6].

In this framework, Bayesian online inference provides the model with uncertainty-qualified numerical evidence, including posterior means and credible intervals for causal effects, grounding the chain-of-thought reasoning in both causal structure and quantified uncertainty [7].

2.5. Agent-Based Systems in Logistics

Agent-based modeling has a long history in transportation and logistics. Multi-agent reinforcement learning has gained traction for dispatching optimization, though existing work typically treats the learning agent and the causal diagnosis component as separate systems. Some studies apply causal machine learning to supply chain risk prediction, while others review neurosymbolic approaches to explainable AI in supply chains. Additionally, combining causal inference with Bayesian networks has been explored for inventory optimization [3].

None of these systems utilize a large language model as the reasoning backbone for causal attribution in a continuously operating agent, nor do they address the grounding problem that arises when large language models are deployed for causal reasoning in operational settings.

2.6. Research Gap

Existing research primarily addresses causal structure learning, reasoning with large language models, Bayesian online inference, and agent-based logistics in isolation. No prior framework integrates a domain-informed structural causal graph, Bayesian state estimation with adaptive forgetting for streaming data, and a reasoning engine based on large language models that performs chain-of-thought causal attribution constrained by the structural causal graph topology and grounded by uncertainty-qualified evidence [8]. Additionally, no study systematically evaluates the sensitivity of such a framework to graph misspecification or compares multiple grounding configurations of large language models across different architectures for causal attribution tasks. This paper introduces such an integrated framework and validates it experimentally across backend attribution, graph robustness, and end-to-end evaluation of large language models.

3. Problem Formulation

3.1. Business Scenarios and Causal Attribution Objective

In high-dynamic logistics systems, order arrivals fluctuate randomly while vehicle loading, route adjustment, and warehouse operations change continuously, producing a minute-level evolving event stream [9]. The outcome indicators during any period, such as timeliness fulfillment rate, are driven jointly by multiple factors, including road-network congestion, warehouse operation rhythm, dispatching policy changes, and unexpected disruptions.

The goal of causal attribution is to identify, along the continuous time axis, the key causes that most significantly affect outcome variables, rather than remaining at the level of correlation statistics. Let y_t denote the outcome indicator at time t and $\boldsymbol{x}_t$ the corresponding operational state vector [10]. Given the historical event sequence $E_{1:t-1}$, the discounted cumulative causal effect of a dispatching sequence $a_{t:t+H}$ of horizon H, relative to a baseline policy $a^0_{t:t+H}$, is:

$$\Delta_t(a_{t:t+H}) = \mathbb{E}[\sum_{\tau=t}^{t+H} \gamma^{\tau-t} y_\tau \mid do(a_{t:t+H}), E_{1:t-1}] - \mathbb{E}[\sum_{\tau=t}^{t+H} \gamma^{\tau-t} y_\tau \mid do(a^0_{t:t+H}), E_{1:t-1}],$$

where $0 < \gamma \leq 1$ is the temporal discount factor [11].

3.2. Design Requirements

The causal attribution agent functions as an online explainer and diagnoser, positioned between the operations monitoring layer and the dispatching decision layer. In addition to meeting the numerical attribution requirements formalized in Eq. (1), the agent must generate interpretable natural-language explanations that dispatching operators can easily understand and act upon. This necessity drives the adoption of a large language model (LLM) as the reasoning backbone [12].

The continuous computing requirement aims to balance attribution quality, interpretability, and resource cost:

$$\min_{\theta} \mathbb{E}[\ell(c_t, y_t)] + \lambda_1 C(\theta) + \lambda_2 \mathcal{H}(c_t, \mathcal{G}),$$

where $\ell(\cdot)$ measures the deviation between attribution results and actual business feedback, $C(\theta)$ represents computation and resource consumption, $\mathcal{H}(c_t, \mathcal{G})$ penalizes attribution conclusions that conflict with the structural causal graph $\mathcal{G}$ (referred to as the causal grounding penalty), and $\lambda_1, \lambda_2 > 0$ are trade-off coefficients [2].

The design principles include: (i) real-time responsiveness, ensuring end-to-end latency does not exceed a scenario-dependent threshold $T_{\max}$; (ii) incremental updating, which avoids full recomputation at each time step; (iii) causal grounding, ensuring all attribution claims are supported by valid causal pathways in $\mathcal{G}$; (iv) interpretability, providing natural-language explanations alongside structured factor scores; and (v) traceability, ensuring each attribution conclusion can be linked to its input evidence.

4. Proposed Method: Causally Grounded LLM Attribution Agent

The proposed framework consists of three integrated layers, as depicted in Figure 1: the streaming state-preparation layer (Section 4.1), the LLM causal reasoning layer (Section 4.2), and the closed-loop dispatching layer (Section 4.4).

Three-layer architecture of the causally grounded LLM attribution agent. The structural causal graph constrains LLM reasoning at two points: the system prompt encodes valid causal pathways, and a post...

4.1. Layer 1: Streaming State Preparation

The streaming state-preparation layer converts raw event data into a structured causal context optimized for large language model consumption [13]. This layer comprises three key components: formal causal modeling, recursive state compression, and Bayesian online inference.

4.1.1. Structural Causal Graph

Let t denote a discrete time step, $X_t = \{x_t^1, \dots, x_t^n\}$ the set of n state variables at time t, and y_t the outcome indicator. The causal structure is represented by a directed acyclic graph $\mathcal{G}$ and structural equations:

$$x_t^i = f_i\big(\mathrm{Pa}(x_t^i), \varepsilon_t^i\big), \quad i = 1, \dots, n,$$

where $\mathrm{Pa}(x_t^i)$ is the parent set of node x_t^i in $\mathcal{G}$, and ε_t^i is an exogenous disturbance term [14]. The graph $\mathcal{G}$ is constructed from domain knowledge: order arrival volume and road-network congestion level serve as exogenous drivers influencing in-warehouse workload and available fleet capacity, which jointly determine the timeliness fulfillment rate. The core state variables are defined in Table 1.

Table 1. State variable definitions in the structural causal graph.

Variable	Description	Type	Layer
Timeliness fulfillment rate (y_t)	Proportion of orders completed on time	Continuous	System output
Order arrival volume (x_t^1)	Number of new orders per unit time	Continuous	Order
Road-network congestion (x_t^2)	Degree of congestion on main routes	Continuous	Road transport
In-warehouse workload (x_t^3)	Intensity of picking and loading tasks	Continuous	Warehouse operation
Available fleet capacity (x_t^4)	Load capacity of dispatchable vehicles	Continuous	Vehicle resource
Dispatching policy (x_t^5)	Route and priority parameters	Mixed	Decision control

Critically, $\mathcal{G}$ serves a dual role: it defines the structural equations used by the Bayesian inference module and constrains the causal pathways that the LLM may invoke during chain-of-thought reasoning [15].

4.1.2. Recursive State Compression

An internal state vector $s_t \in \mathbb{R}^d$ compresses the streaming event history into a fixed-dimensional representation [16]. Let $z_t \in \mathbb{R}^m$ denote the preprocessed feature vector extracted from event e_t:

$$s_t = g(s_{t-1}, z_t; \theta_g),$$

where $g(\cdot)$ is a two-layer gated recurrent unit (GRU) with hidden dimension $d = 128$. The GRU gating mechanism provides learnable forgetting that retains relevant historical signals while discounting stale information. The output s_t is decoded into interpretable state estimates $\hat{x}_t^1, \ldots, \hat{x}_t^n$ via a linear projection, which are included in the LLM prompt context.

4.1.3. Bayesian Online Inference with Adaptive Forgetting

The causal model parameters θ are updated incrementally using a Bayesian recursive form [17].

$$p_t(\theta) \propto p(e_t \mid \theta)^{\rho_t} \, p_{t-1}(\theta)^{1-\rho_t}, \quad 0 < \rho_t \leq 1,$$

Here, ρ_t represents an adaptive forgetting weight derived from operational volatility [18].

$$\rho_t = \sigma\big(\alpha_\rho \cdot \mathrm{vol}_t + \beta_\rho\big),$$

In this context, vol_t denotes the coefficient of variation of order arrival rates and road-network speed changes over the preceding $w = 10$ time steps, while $\sigma(\cdot)$ is the logistic sigmoid function, and α_ρ, β_ρ are learnable parameters.

The posterior predictive distribution under dispatching intervention a_t is estimated through Monte Carlo sampling with $K = 200$ draws.

$$p(y_t \mid do(a_t), D_t) \approx \frac{1}{K}\sum_{k=1}^{K} p\,(y_t \mid do(a_t), \theta^{(k)}), \quad \theta^{(k)} \sim p_t(\theta).$$

The posterior mean and 90% credible interval for each causal effect are extracted and formatted as part of the structured causal context $\mathcal{C}_t$ provided to the LLM [13].

4.1.4. Structured Causal Context Assembly

The output of Layer 1 is a structured causal context $\mathcal{C}_t$ that includes current state estimates $\hat{x}_t^1, \ldots, \hat{x}_t^n$ with 90% credible intervals, estimated causal effects along each edge of $\mathcal{G}$ with posterior uncertainty, a summary of the five most recent anomaly events (time, type, severity), and the current scenario classification (normal, promotion peak, or congestion disturbance) inferred from the state vector. This context is organized into a structured text format for integration into the LLM prompt.

4.2. Layer 2: LLM Causal Reasoning

The LLM operates on the structured causal context $\mathcal{C}_t$ produced by Layer 1, utilizing chain-of-thought attribution reasoning to generate a structured factor-contribution vector and a natural-language explanation.

4.2.1. Prompt Design

The prompt consists of a fixed system component and a time-varying user component [19]. The system prompt encodes (a) the SCG topology $\mathcal{G}$ with all nodes and directed edges, (b) a constraint that attribution may only follow edges in $\mathcal{G}$, (c) the output schema (a JSON object with factor scores, confidence level, and natural-language explanation), and (d) an instruction to reason step by step before producing the final output. The user prompt contains the structured causal context $\mathcal{C}_t$ with labeled sections for state estimates, causal effect estimates with uncertainty intervals, recent anomaly events, and scenario classification. An abbreviated example is shown in Table 2.

Table 2. Abbreviated prompt structure for the LLM causal reasoning layer.

Component	Content (abbreviated)
System	You are a causal attribution agent for a logistics system. The causal structure is: order_volume → warehouse_workload → timeliness; congestion → fleet_capacity → timeliness; dispatching_policy → timeliness. You may ONLY attribute

	causation along these edges. Respond with a JSON object: {"factors": [{"name": ..., "score": ...}], "confidence": ..., "explanation": ...}. Reason step by step before answering.
User	Current state (t=14:35): order_volume = 187 orders/h [CI: 172–201]; congestion = 0.29 [CI: 0.24–0.34]; warehouse_workload = 221 tasks/h [CI: 208–235]; fleet_capacity_util = 0.89 [CI: 0.85–0.93]; timeliness = 0.71 [CI: 0.67–0.75]. Causal effects: congestion → fleet_capacity: − 0.18 [CI: − 0.24 to − 0.12]; order_volume → warehouse_workload: +0.31 [CI: +0.22 to +0.40]; ... Recent anomalies: 14:22 timeliness drop 14%; 14:28 congestion spike on Route A7. Scenario: promotion peak.

4.2.2. Reasoning, Validation, and Latency

The chain-of-thought reasoning trace progresses through five steps: identifying state variables that deviate from normal baselines, tracing causal pathways to the outcome using the SCG topology, assessing pathway strength based on Bayesian effect estimates and uncertainty intervals, ranking contributing factors by effect magnitude, and generating structured attribution output with a natural-language explanation. This approach enhances transparency by allowing operators to inspect the reasoning trace and minimizes hallucinated causal claims by requiring each attribution step to reference SCG edges and numerical evidence.

After generation, a deterministic validator ensures that every causal claim aligns with the SCG topology. If the LLM attributes an effect of x_t^i on y_t, the validator confirms the existence of a directed path from x_t^i to y_t within $\mathcal{G}$. Attributions involving absent edges are rejected, and the LLM is re-prompted with corrective instructions [20].

To address LLM inference latency, the framework employs two strategies [2]. Layer 1 operates at every time step (one-minute intervals), while Layer 2 is activated only when an anomaly is detected or a state deviation exceeds 1.5 standard deviations. During stable periods, the most recent attribution is carried forward with interpolated factor scores. Additionally, LLM inference is conducted asynchronously, allowing Layer 1 to continue processing while awaiting the response.

4.2.3. Agent Orchestration

The system is structured as an orchestrated agent that manages internal event states, activates the attribution backend upon detecting anomalies, provides the language model with validated evidence, and generates a bounded action schema for downstream execution or human review. This architecture distinctly separates numerical attribution, language-based explanation, and intervention selection into independent functions.

During each anomaly window, the agent compiles an event packet containing the state summary, ranked candidate causes, SCG-valid pathways, and effect estimates. The language model generates a response with four components: primary cause, supporting pathway, confidence statement, and recommended intervention class. A deterministic validator verifies the cited pathways and intervention class before presenting the result. If the response conflicts with the SCG or suggests an invalid action, the agent defaults to a backend-only response and logs the case for further review.

4.3. Layer 3: Closed-Loop Dispatching

The attribution output $\hat{c}_t$ feeds into a dispatching decision module that incrementally adjusts the policy parameter vector u_t.

$$u_t = u_{t-1} + \eta\, W \hat{c}_t,$$

Here, $W \in \mathbb{R}^{p \times n}$ maps causal factor contributions to dispatching control variables, and $\eta > 0$ represents the step-size coefficient. To prevent excessive adjustments that could destabilize the system, the policy update is constrained.

$J = \mathbb{E}[L(y_t, \hat{y}_t(u_t))] + \beta \parallel u_t - u_{t-1} \parallel^2$,

This approach balances improvements in the timeliness fulfillment rate with policy smoothness. Execution results are fed back to the data layer, completing the continuous causal-driven dispatching loop. Layer 3 is a design specification, and its empirical validation requires a live operational environment, which is deferred to future work.

The full pipeline is summarized in Algorithm 1.

Initialize GRU state $s_0 \leftarrow \mathbf{0}$; set $\hat{c}_0 \leftarrow \mathbf{0}$.

5. Experimental Evaluation

The empirical study consists of three layers: (i) a backend attribution comparison that evaluates four methods under both in-distribution and shifted environments, (ii) an analysis of graph misspecification to assess sensitivity to structural inaccuracies in the SCG, and (iii) a direct end-to-end evaluation of large language models, comparing four grounding configurations using two models at different capability tiers (Claude Sonnet 4 and Claude Haiku 4.5) on anomaly attribution tasks [21].

5.1. Simulation Environment

We construct a synthetic logistics environment using a structural causal model with nine observed state variables at each time step:

$X_t = \{D_t, S_t, B_t, U_t, L_t, C_t, R_t, W_t, P_t\}$,

where D_t represents demand volume, S_t available supply, B_t backlog, U_t utilization, L_t average delay, C_t completion rate, R_t routing friction, W_t weather severity, and P_t promotion intensity [22].

The simulator evolves through stochastic structural equations, where demand, supply, delay, and completion depend on lagged states, exogenous shocks, and intervention-specific perturbations. Cross-contamination is incorporated, meaning each intervention type partially influences variables outside its primary causal pathway. For instance, demand spikes reduce effective supply, driver dropout increases routing friction, routing disruptions lower effective supply, and weather shocks elevate demand. These overlapping effects complicate the separation of causes from individual features.

Four intervention types are introduced and logged as ground truth: demand spike, driver dropout, routing disruption, and weather shock. The benchmark includes 60 simulated runs, each with 240 time steps, resulting in 14,400 state observations. These runs are divided into 40 for training and 20 for testing, yielding 160 training events and 80 test events with approximately balanced classes. Event-level ground truth is derived from the simulator intervention log rather than subjective annotations.

A distribution-shifted environment is created using 20 additional runs with modified parameters: noise scale is increased from 1.0 to 1.8, intervention magnitudes are reduced to 80% of the baseline (weaker signals), cross-contamination is increased from 0.2 to 0.5, seasonality amplitude is raised by 50%, and structural sensitivity coefficients for routing and supply are reduced to 70% of the baseline. This results in a more challenging environment where trained models must generalize to weaker signals, higher noise levels, and altered structural dynamics.

5.2. Compared Backend Methods

We evaluate the numerical attribution core using four distinct methods:

1. Ungrounded symptom heuristic: This noncausal baseline assigns root causes based on the most extreme z-score among symptom-mapped features, without incorporating structural or supervised information.
2. All-feature logistic regression: A supervised linear baseline that utilizes all available features, including raw state variables, temporal deltas, and interaction terms.

3. SCG-aligned logistic regression: A structure-aware logistic model limited to features that align with SCG-valid causal pathways, incorporating their temporal deltas and SCG-consistent interaction terms.
4. Random forest: A flexible supervised black-box baseline that leverages all features and is capable of capturing nonlinear interactions.

All supervised models are trained on an identical 40-run training set with consistent feature engineering, including temporal deltas and interaction terms [23]. Evaluation is conducted on a held-out 20-run test set, with predictions aggregated from the time-step level to the event level through majority voting within each anomaly window.

5.3. Backend Evaluation Metrics

We report three metrics on the held-out event set: event accuracy, which measures the proportion of anomaly events whose majority-vote prediction matches the ground-truth cause; macro F1, representing the class-balanced F1 score across the four root causes; and average consistency, calculated as the mean within-event agreement of time-step predictions with the event-level majority vote [20].

5.4. In-Distribution Backend Results

Table 3 presents in-distribution results on the held-out test set. The ungrounded heuristic achieves 70.0% event accuracy (macro F1 = 0.661), indicating that symptom-level reasoning without structural grounding is insufficient under cross-contaminated signatures. Both logistic models achieve 78.8% event accuracy; the SCG-aligned variant attains higher macro F1 (0.753 vs [24]. 0.744) and consistency (0.835 vs. 0.830), reflecting improved class-balanced performance from structural feature selection. The random forest achieves 76.2% accuracy with the lowest supervised macro F1 (0.720), consistent with overfitting to noise in the cross-contaminated training data.

Table 3. In-distribution held-out backend results (80 test events, 4 root causes).

Method	Event accuracy	Macro F1	Avg. consistency
Ungrounded symptom heuristic	0.700	0.661	0.767
All-feature logistic regression	0.788	0.744	0.830
SCG-aligned logistic regression	0.788	0.753	0.835
Random forest	0.762	0.720	0.770

5.5. Per-Class Attribution Analysis

Table 4 presents per-class F1 scores on the in-distribution test set. Routing disruption is identified with near-perfect recall by all supervised methods, as routing friction provides a distinctive signal with limited cross-contamination. Demand spikes and weather shocks are more challenging to distinguish because weather shocks interfere with the demand signal. The confusion matrix (Table 5) confirms this: 5 of 22 weather-shock events are misclassified as demand spikes, and 3 of 17 demand-spike events are misclassified as driver dropout.

Table 4. Per-class F1 scores on the in-distribution test set.

Method	Demand spike	Driver dropout	Routing disrupt.	Weather shock
Ungrounded heuristic	0.412	0.621	0.945	0.667
All-feature logistic	0.500	0.714	1.000	0.762
SCG-aligned logistic	0.545	0.769	1.000	0.698

Random forest	0.529	0.640	0.982	0.727

Table 5. Confusion matrix for the SCG-aligned logistic model (in-distribution). Rows are true causes; columns are predicted causes.

	Demand	**Driver**	**Routing**	**Weather**
Demand spike ($n = 17$)	9	3	0	5
Driver dropout ($n = 12$)	1	10	0	1
Routing disruption ($n = 29$)	0	0	29	0
Weather shock ($n = 22$)	6	1	0	15

5.6. Backend Robustness Under Distribution Shift

Table 6 reports results under distribution shift. The SCG-aligned model experiences a degradation from 0.788 to 0.675, representing an 11.2-point drop. This is compared to a 12.5-point drop for the all-feature logistic model, 21.2 points for the random forest model, and 23.8 points for the ungrounded heuristic. Despite the shift, the SCG-aligned model achieves the highest accuracy (0.675) and macro F1 score (0.620) among all evaluated methods [25].

Table 6. Backend event accuracy under in-distribution (ID) and distribution-shifted (Shift) environments. All models trained on the same 40-run ID training set.

Method	**ID accuracy**	**ID F1**	**Shift accuracy**	**Shift F1**
Ungrounded heuristic	0.700	0.661	0.462	0.410
All-feature logistic	0.788	0.744	0.662	0.606
SCG-aligned logistic	0.788	0.753	0.675	0.620
Random forest	0.762	0.720	0.550	0.525

5.7. Graph Misspecification Analysis

A concern with SCG-aligned methods is circularity: if the simulator follows a specific causal graph, a model aligned to that graph may perform well by construction. To address this, the SCG is systematically modified, and performance changes are measured. If the observed benefit were merely a regularization artifact, performance would remain insensitive to which edges are included. Conversely, if the SCG encodes genuine causal structure, removing true edges should degrade performance, while adding spurious edges should not provide any advantage.

Table 7 presents results for six SCG configurations. The correct SCG utilizes the full graph with all valid edges (19 features). The "Drop weather → routing" configuration removes the weather severity variable, eliminating the primary weather-shock signal (16 features). The "Drop supply edge" configuration excludes the supply variable, obscuring driver-dropout signals (16 features). The "Add spurious" configuration introduces promotion intensity and its delta, adding a spurious edge (21 features). The "Drop two edges" configuration removes both weather and supply variables simultaneously (13 features). Finally, the "Minimal" configuration retains only delay and completion variables along with their deltas (4 features).

Table 7. Graph misspecification analysis: attribution accuracy under correct, degraded, and augmented SCG configurations.

SCG configuration	**Features**	**ID acc.**	**ID F1**	**Shift acc.**	**Shift F1**
Correct SCG	19	0.788	0.753	0.675	0.620
Drop weather → routing	16	0.600	0.587	0.400	0.380
Drop supply edge	16	0.750	0.702	0.662	0.595
Add spurious (promotion)	21	0.788	0.744	0.662	0.606
Drop two edges	13	0.562	0.548	0.425	0.405

Minimal (delay + completion)	4	0.362	0.366	0.250	0.239

Three key findings emerge. First, edge deletion results in significant performance degradation: removing the weather → routing edge reduces ID accuracy from 0.788 to 0.600 (−18.8 points), and removing two edges further decreases accuracy to 0.562. The correct SCG (19 features) outperforms both the augmented graph (21 features) and the 16-feature graph with incorrect pathways, eliminating feature-count regularization as a plausible explanation. Second, adding spurious edges provides no in-distribution benefit (accuracy remains at 0.788) but reduces shift robustness (0.662 compared to 0.675), indicating overfitting to noise features. Third, the minimal configuration, which includes only delay and completion variables, performs near chance level, demonstrating that outcome-adjacent features alone are insufficient for accurate root-cause attribution.

6. End-to-End LLM Evaluation

To evaluate the full agent beyond backend classification accuracy, the reasoning layer of the language model is tested on the same anomaly windows used for backend benchmarking. Two models from the Claude family at different capability tiers are utilized: Claude Sonnet 4 and Claude Haiku 4.5. Testing across these tiers assesses whether the grounding framework generalizes across model sizes and whether smaller models benefit disproportionately from causal grounding [26].

6.1. Evaluation Setup

For each anomaly event, the agent receives a structured event packet containing the anomaly-window summary, a feature table with state variable statistics, SCG-valid causal pathways, and backend attribution scores. The LLM generates four output fields: primary cause, causal explanation, confidence statement, and recommended intervention class [27].

We evaluate all 80 anomaly events (20 per root cause) from the held-out test set under four grounding configurations that progressively add information. In the ungrounded configuration, the LLM receives only a symptom-level anomaly summary. The structured configuration includes a feature table with state variable means and standard deviations. The SCG-grounded configuration further incorporates the SCG topology and an explicit constraint prohibiting attribution along absent edges. The SCG + evidence configuration (the proposed agent) integrates backend attribution scores and per-class probabilities from the SCG-aligned model.

All configurations use identical temperature (0), maximum output length (800 tokens), and chain-of-thought instruction ("reason step by step before answering"). Each LLM × configuration × event combination is queried once [5].

6.2. End-to-End Metrics

We report four metrics. Attribution accuracy measures the agreement between the primary cause identified by the LLM and the ground truth. Explanation faithfulness represents the proportion of explanations that reference only SCG-valid pathways, assessed through automated rule matching against the SCG edge list; this metric is reported exclusively for configurations where the SCG is provided. The SCG violation rate indicates the proportion of explanations that reference or fabricate causal links not present in the SCG. Action alignment evaluates whether the recommended intervention class corresponds to the valid action for the ground-truth cause.

6.3. LLM Evaluation Results

Table 8 presents the end-to-end results for both models across all four grounding configurations on the full 80-event held-out test set.

Table 8. End-to-end LLM evaluation results (80 test events, 20 per root cause). Attrib. Acc. = attribution accuracy; Faithful. = explanation faithfulness, scored by automated SCG-edge matching (report...

Model	Configuration	Attrib. Acc.	Faithful.	SCG Viol. (↓)	Action
Claude Sonnet 4	Ungrounded	56.2%	—	18.8%	53.8%
	Structured	51.2%	—	16.2%	48.8%
	SCG-grounded	65.0%	88.8%	7.5%	63.8%
	SCG + evidence	76.2%	93.8%	3.8%	75.0%
Claude Haiku 4.5	Ungrounded	43.8%	—	23.8%	41.2%
	Structured	47.5%	—	20.0%	45.0%
	SCG-grounded	61.2%	83.8%	10.0%	58.8%
	SCG + evidence	78.8%	91.2%	2.5%	77.5%

Three key findings can be observed from Table 8.

First, the SCG + evidence configuration achieves the highest attribution accuracy for both models: 76.2% for Sonnet 4 and 78.8% for Haiku 4.5, representing improvements of 20.0 and 35.0 percentage points over their respective ungrounded baselines.

Second, Haiku 4.5 benefits disproportionately from grounding [28]. Its ungrounded accuracy (43.8%) is 12.4 percentage points lower than Sonnet's (56.2%), but its fully grounded accuracy (78.8%) surpasses Sonnet's (76.2%). This suggests that structural constraints and evidence scores effectively compensate for reduced model capacity.

Third, SCG violation rates decrease consistently with increasing grounding depth. Without the SCG, both models frequently generate invalid causal pathways (18.8% and 23.8%). Adding the SCG constraint reduces violations to 7.5% and 10.0%, while incorporating backend evidence further reduces them to 3.8% and 2.5%.

Explanation faithfulness, reported only for configurations where the SCG is provided, ranges from 83.8% to 93.8%. Sonnet 4 achieves higher faithfulness than Haiku 4.5 at the SCG-grounded level (88.8% vs. 83.8%), but the gap narrows with backend evidence (93.8% vs. 91.2%), indicating that evidence scores help the smaller model adhere to valid pathways. Action alignment closely tracks attribution accuracy, differing by only 1 to 3 percentage points across all configurations.

Scorer limitations are noted [29].

Faithfulness and SCG violation are assessed using an automated rule-based method that identifies invalid causal pathway pairs (e.g., weather and demand) co-occurring near causal-language indicators in the generated explanation. This heuristic has limited sensitivity, as it may overlook violations where the model implies an invalid pathway without explicit causal language. Additionally, it cannot detect fabricated mechanisms outside the predefined invalid-pair list. A more robust approach would involve LLM-as-judge scoring or require explicit pathway chains (e.g., [weather → routing_friction → delay → completion) that can be validated deterministically against the SCG adjacency matrix. The scorer code is provided with the evaluation scripts to enable the integration of stronger measures.

The evaluation code, prompt templates, event packets, and scoring scripts are included as supplementary material.

7. Discussion

Stability under distribution shift. The SCG-aligned model demonstrates the smallest accuracy degradation among supervised methods under distribution shift (11.2 points

compared to 21.2 for the random forest). This finding suggests that structural grounding provides a stable attribution foundation under varying operating conditions [4].

Graph misspecification and causal contribution. The misspecification analysis reveals asymmetric sensitivity to edge changes: removing the weather → routing edge (one variable) results in an 18.8-point accuracy drop, whereas adding a spurious variable does not yield any positive effect. This asymmetry indicates that the SCG encodes domain-relevant causal pathways rather than functioning as a generic regularizer.

Error structure. The confusion matrix (Table 5) shows that demand--weather confusion constitutes the majority of errors, stemming from the weather → demand cross-contamination pathway. During deployment, the agent could flag high-uncertainty demand--weather attributions for human review, or the SCG could integrate an external weather forecast signal to resolve the ambiguity.

Role of the agent architecture. A black-box classifier does not encode cause--effect pathways, provide mechanism-level reasoning, or generate operator-facing explanations. The proposed architecture addresses a distinct requirement: auditable diagnosis that can be mapped to specific interventions. It integrates state construction, attribution, explanation, and intervention mapping into a unified pipeline.

Cross-tier LLM evaluation. Claude Haiku 4.5 achieves 78.8% attribution accuracy with full grounding, surpassing Sonnet 4's 76.2%. Haiku's grounding benefit (+35.0 points) is nearly double Sonnet's (+20.0 points), indicating that structural constraints compensate for reduced model capacity. The reduction in SCG violation rates from 18.8--23.8% (ungrounded) to 2.5--3.8% (fully grounded) is consistent across both model tiers. These results suggest that organizations can deploy smaller, lower-cost models without compromising attribution quality when the causal grounding architecture is implemented [13].

Scope of validation. This paper validates the attribution architecture and its core components---backend classification, graph robustness, and LLM grounding---using a controlled synthetic benchmark. It does not validate the full agent on production telemetry, measure downstream operational impact, or demonstrate closed-loop dispatching improvements. This sequence is deemed appropriate for validating a new agent architecture. Synthetic environments with known ground truth enable controlled comparisons of design alternatives (e.g., SCG-aligned versus ungrounded, grounded versus ungrounded LLM) that would be confounded in live deployments. The graph misspecification study, in particular, requires a data-generating process with an explicit causal graph, which is unavailable in production settings where the true graph is unknown. Several influential agent and causal-inference frameworks have followed a similar progression from synthetic validation to real-world deployment. The advantage of a synthetic-first approach is that design flaws can be identified and corrected before incurring the cost and risk of production integration. The remaining validation steps---real telemetry, operator studies, and closed-loop evaluation---are explicitly scoped as future work rather than left as implicit gaps.

Limitations. First, the evaluation relies on synthetic data. While the simulator incorporates cross-contamination, distribution shift, and realistic noise, production logistics telemetry may present additional complexities, such as partial observability, non-stationary graph structures, and multi-cause events, which are not captured here [21]. Second, the SCG is assumed to be approximately correct. The misspecification analysis indicates that removing two edges reduces accuracy to 0.562, highlighting the need for domain-expert graph validation during deployment. Third, Layer 3 (closed-loop dispatching) is a design specification and has not been validated; assessing its impact on operational outcomes requires a live environment. Fourth, the faithfulness scorer employs automated rule-based matching with limited sensitivity to subtle causal reasoning errors; more rigorous assessment could be achieved through LLM-as-judge or structured-pathway scoring. Fifth, the LLM evaluation measures attribution accuracy and explanation quality but does not assess whether correct attributions lead to improved

operator decisions or logistics outcomes. This downstream evaluation necessitates a user study, which is beyond the current scope.

8. Conclusion

This paper introduced a causally grounded attribution agent architecture tailored for high-dynamic logistics systems and validated its core components using a controlled synthetic benchmark. The architecture incorporates streaming state preparation, an SCG-constrained attribution backend, an LLM reasoning layer, and an intervention-mapping interface, ensuring a comprehensive approach to attribution.

The SCG-aligned backend demonstrated the highest macro F1 score (0.753) in-distribution and exhibited minimal degradation under distribution shift (11.2 points). The removal of a single SCG edge resulted in accuracy degradation of up to 18.8 points, underscoring the graph's role in encoding genuine causal structures. Evaluation of the LLM revealed that full causal grounding enhanced attribution accuracy to 76.2% (Sonnet 4) and 78.8% (Haiku 4.5) compared to ungrounded baselines of 56.2% and 43.8%. Notably, the smaller Haiku model showed a disproportionate improvement (+35.0 points vs. +20.0), suggesting that structural grounding effectively compensates for reduced model capacity.

These findings confirm that the proposed architecture delivers reliable, stable, and interpretable attributions under controlled conditions. Future validation steps include testing on production logistics telemetry with labeled anomaly causes, conducting operator studies to evaluate downstream decision quality, and assessing the closed-loop dispatching layer in a live environment. The evaluation code, prompt templates, and event packets are provided as supplementary material.

References

1. L. G. Neuberg, "Causality: models, reasoning, and inference, by judea pearl, cambridge university press, 2000," Econometric Theory, vol. 19, no. 4, pp. 675–685, 2003.
2. J. Wei et al., "Chain-of-thought prompting elicits reasoning in large language models," Advances in Neural Information Processing Systems, vol. 35, pp. 24824–24837, 2022.
3. Z. Ji et al., "Survey of hallucination in natural language generation," ACM Computing Surveys, vol. 55, no. 12, pp. 1–38, 2023.
4. J. Runge, P. Nowack, M. Kretschmer, S. Flaxman, and D. Sejdinovic, "Detecting and quantifying causal associations in large nonlinear time series datasets," Science Advances, vol. 5, no. 11, eaau4996, 2019.
5. C. W. Granger, "Investigating causal relations by econometric models and cross-spectral methods," Econometrica: Journal of the Econometric Society, pp. 424–438, 1969.
6. K. Cho et al., "Learning phrase representations using RNN encoder–decoder for statistical machine translation," in Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), pp. 1724–1734, Oct. 2014.
7. R. Pamfil et al., "Dynotears: Structure learning from time-series data," in International Conference on Artificial Intelligence and Statistics, pp. 1595–1605, June 2020.
8. T. Schick et al., "Toolformer: Language models can teach themselves to use tools," Advances in Neural Information Processing Systems, vol. 36, pp. 68539–68551, 2023.
9. D. B. Rubin, "Estimating causal effects of treatments in randomized and nonrandomized studies," Journal of Educational Psychology, vol. 66, no. 5, pp. 688, 1974.
10. A. P. Raia, "A study of the educational value of management games," The Journal of Business, vol. 39, no. 3, pp. 339–352, 1966.
11. E. Kiciman, R. Ness, A. Sharma, and C. Tan, "Causal reasoning and large language models: Opening a new frontier for causality," Transactions on Machine Learning Research, 2023.
12. P. Lewis et al., "Retrieval-augmented generation for knowledge-intensive NLP tasks," Advances in Neural Information Processing Systems, vol. 33, pp. 9459–9474, 2020.
13. M. Zečević, M. Willig, D. S. Dhami, and K. Kersting, "Causal parrots: Large language models may talk causality but are not causal," arXiv preprint arXiv:2308.13067, 2023.
14. S. M. Lundberg and S. I. Lee, "A unified approach to interpreting model predictions," Advances in Neural Information Processing Systems, vol. 30, 2017.
15. D. A. Boiko, R. MacKnight, B. Kline, and G. Gomes, "Autonomous chemical research with large language models," Nature, vol. 624, no. 7992, pp. 570–578, 2023.
16. D. Koller and N. Friedman, Probabilistic Graphical Models: Principles and Techniques. MIT Press, 2009.
17. T. Broderick, N. Boyd, A. Wibisono, A. C. Wilson, and M. I. Jordan, "Streaming variational Bayes," Advances in Neural Information Processing Systems, vol. 26, 2013.

18. S. Pan et al., "Unifying large language models and knowledge graphs: A roadmap," IEEE Transactions on Knowledge and Data Engineering, vol. 36, no. 7, pp. 3580–3599, 2024.
19. R. Kulhavý and M. B. Zarrop, "On a general concept of forgetting," International Journal of Control, vol. 58, no. 4, pp. 905–924, 1993.
20. B. Li et al., "Research, application, and challenges of causal inference in industrial fault diagnosis: A survey," Engineering Applications of Artificial Intelligence, vol. 158, 111376, 2025.
21. Z. Xi, W. Guan, and A. Savasan, "Optimizing inventory management: A causal inference-driven Bayesian network with transfer learning adaptation," PeerJ Computer Science, vol. 11, e3262, 2025.
22. T. Ameer and O. F. Valilai, "Cloud-native causal AI for supply chain KPI monitoring: A GCP framework to diagnose out-of-stock events," Machine Learning with Applications, 100765, 2025.
23. Z. Zhang et al., "Casual inference-enabled graph neural networks for generalized fault diagnosis in industrial IoT system," Information Sciences, vol. 694, 121719, 2025.
24. M. Wyrembek, G. Baryannis, and A. Brintrup, "Causal machine learning for supply chain risk prediction and intervention planning," International Journal of Production Research, vol. 63, no. 15, pp. 5629–5648, 2025.
25. F. F. Bastarianto, T. O. Hancock, C. F. Choudhury, and E. Manley, "Agent-based models in urban transportation: review, challenges, and opportunities," European Transport Research Review, vol. 15, no. 1, pp. 19, 2023.
26. M. Wooldridge, An Introduction to Multiagent Systems. John Wiley & Sons, 2009.
27. E. E. Kosasih, E. Papadakis, G. Baryannis, and A. Brintrup, "A review of explainable artificial intelligence in supply chain management using neurosymbolic approaches," International Journal of Production Research, vol. 62, no. 4, pp. 1510–1540, 2024.
28. J. Li, E. Rombaut, and L. Vanhaverbeke, "A systematic review of agent-based models for autonomous vehicles in urban mobility and logistics: Possibilities for integrated simulation models," Computers, Environment and Urban Systems, vol. 89, 101686, 2021.
29. J. Peters, D. Janzing, and B. Scholkopf, Elements of Causal Inference: Foundations and Learning Algorithms. MIT Press, 2017.