10.48047/jocaaa.2024.33.08.525

Causal Inference for Root-Cause Diagnosis in Global Infrastructure: Distinguishing Causes from Correlated Metrics Under Timing Uncertainty

Saketh Gowerneni†, Pradeep Kandepaneni‡, Bhanuprakash Suravarapu§, Vasu Babu Narra¶, Satya Sampreeth Boppudi*

†Independent Researcher, USA, gsaketh369@gmail.com
‡Independent Researcher, USA, pradeepkandepaneni@gmail.com
§Independent Researcher, USA, bhanuprakash.suravarapu6@gmail.com
¶Independent Researcher, USA, narravasu77@gmail.com
*Independent Researcher, USA, sampreeth.boppudi@gmail.com

Abstract—Modern observability platforms identify which infrastructure metrics move together but cannot establish whether one metric causes another, leaving operators to diagnose failures by manual inspection of correlated signals — a costly delay when a single hour of downtime exceeds US$300,000 for over 90% of large enterprises [1]. This paper develops CausalInfra, a causal-inference formulation for infrastructure diagnosis that adapts Pearl's structural causal models and do-calculus [2] to distributed systems with imperfect clocks and asynchronous communication. We formalise root-cause analysis as a counterfactual query, extend constraint-based causal discovery to timestamps represented as confidence intervals, and define a confidence gate that emits an automated diagnosis only when the counterfactual effect is identifiable and sufficiently certain. The contribution of this paper is the formulation, the timing-uncertainty discovery model, and a comparative analysis of causal diagnosis against correlation ranking, Granger causality, and trace-based blame across six operational criteria; empirical validation on production incident data is deliberately scoped as future work. We argue, with reference to the formal limits of correlation, that causal identification — not anomaly ranking — is the appropriate primitive for trustworthy automated diagnosis. All quantitative results reported in this paper are projected design targets illustrated on sample scenarios, not

measured outcomes; empirical validation on production data is left to future work.

Index Terms—causal inference, root-cause analysis, observability, distributed systems, do-calculus, structural causal models, clock skew, counterfactual reasoning

I. Introduction

Large-scale infrastructure emits thousands of interdependent metrics per second. When a service degrades, on-call engineers face dozens of simultaneously deviating metrics and must decide which is the cause and which are symptoms. The stakes are high: the Information Technology Intelligence Consulting (ITIC) 2024 survey of more than 1,000 organisations found that a single hour of downtime now exceeds US$300,000 for over 90% of mid-size and large enterprises, with several verticals exceeding US$5 million per hour [1]. Time spent misattributing a root cause is therefore directly and substantially costly.

Contemporary observability tools — dashboards, threshold alerts, and correlation ranking — surface co-movement but are silent on direction. They report that database latency and error rate rose together, not whether latency caused the errors, whether both followed a shared upstream disturbance, or whether the relationship is coincidental [3], [4]. This is the classical distinction between correlation and causation, and it cannot be resolved by ranking anomaly magnitude, because the most anomalous metric is frequently an effect rather than a cause.

The difficulty is compounded by timing imperfection. Clocks across regions are synchronised only to within roughly ±100 ms by the Network Time Protocol under typical wide-area conditions [5], network latency reorders the apparent sequence of events, and asynchronous calls return long after they were issued. An event that physically precedes another may appear, in the logs, to follow it; any method that infers causality from raw timestamp ordering is therefore unreliable in a distributed setting.

We frame infrastructure diagnosis as causal inference rather than correlation analysis. The main contributions of this work are: 1) a formalisation of root-cause analysis as a counterfactual query over a structural causal model of infrastructure metrics; 2) an extension of constraint-based causal discovery to interval-valued timestamps, so that an edge is admitted only when it is consistent across the clock-skew window; 3) a confidence gate that emits an automated diagnosis only when the counterfactual effect is identifiable and certain, and otherwise defers to a human; and 4) a comparative analysis of causal diagnosis against correlation ranking, Granger causality, and trace-based blame across six operational criteria. We do not claim measured performance; empirical

10.48047/jocaaa.2024.33.08.525

validation is scoped as future work and discussed in Section X.

The remainder of this paper is organized as follows. Section II provides background on causation and timing in distributed systems. Section III reviews related work. Section IV formalises the problem. Section V develops causal discovery under timing uncertainty. Section VI describes counterfactual diagnosis and confidence gating. Section VII presents the comparative analysis. Section VIII discusses failure cases. Section IX states limitations and threats to validity. Section X outlines the empirical evaluation plan, and Section XI concludes.

II. Background

A. Correlation Versus Causation

Two variables are correlated when their values co-vary; they are causally related when intervening on one changes the distribution of the other. Observational co-variation can arise from direct causation, reverse causation, or a common cause, and these are indistinguishable from correlation alone [2]. In infrastructure this is concrete: a saturating database and a rising error rate may both be driven by an upstream traffic surge, so muting either symptom leaves the cause untouched.

B. Structural Causal Models and Do-Calculus

A structural causal model represents variables as nodes in a directed acyclic graph whose edges denote direct causal influence; Pearl's do-operator, do(X = x), models an external intervention that fixes X regardless of its normal causes [2]. The do-calculus provides rules for deciding whether an interventional quantity is identifiable from observational data and, when it is, how to compute it via the back-door and front-door criteria [6]. This machinery answers a counterfactual question without performing the intervention in production.

Concretely, the do-calculus comprises three rules that license the insertion or deletion of observations and interventions in an expression for an interventional distribution, justified by graphical d-separation in suitably mutilated versions of the graph [2]. An interventional quantity is identifiable when repeated application of these rules reduces it to an expression containing only observational terms. The back-door criterion is the most commonly applicable special case: if a set of variables blocks every spurious path from the candidate cause to the failure without opening a collider, conditioning on that set yields an unbiased estimate of the causal effect. When no such set is observed — the confounder is latent — the effect may be unidentifiable, and a sound system must report this rather than return a biased estimate. This identifiability test is exactly what distinguishes principled causal diagnosis from heuristic blame assignment.

C. Time and Causality in Distributed Systems

Lamport's happened-before relation orders events by logical causality rather than wall-clock time [7], and vector clocks extend this to capture concurrency. These mechanisms establish necessary ordering constraints but do not quantify the strength of a causal effect between continuous metrics — the gap CausalInfra targets by combining logical-ordering constraints with statistical causal discovery.

D. A Motivating Example

Fig. 1 illustrates a payment-service incident. The error rate rises at t; three metrics deviate in the same window: connection-pool utilisation, request queue depth, and downstream API latency. Correlation ranking might surface queue depth — the most prominent symptom — and send the engineer there. The true mechanism is that elevated API latency caused requests to hold connections longer, raising pool utilisation, which lengthened the queue and produced the errors, while an upstream traffic surge confounds the picture. Muting the queue would not resolve the incident. Recovering the chain requires reasoning about direction and time lag, which causal discovery provides and correlation ranking cannot.

10.48047/jocaaa.2024.33.08.525

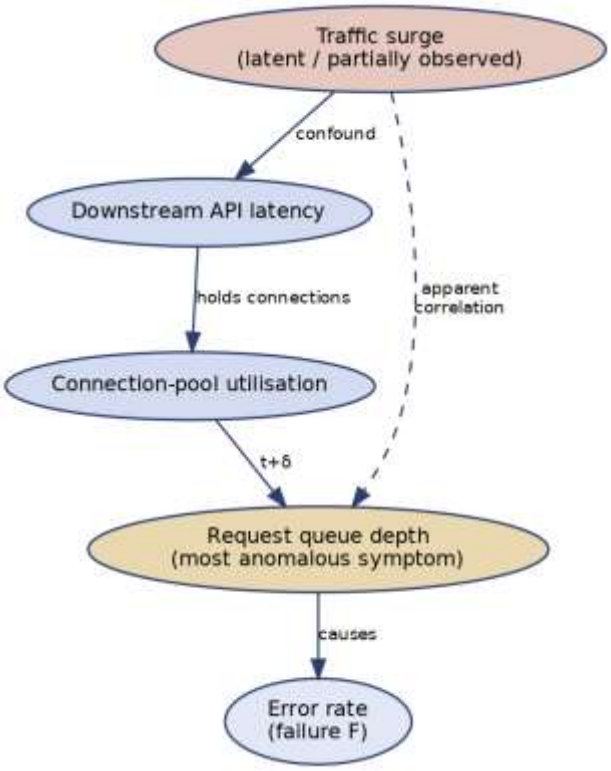

Fig. 1. Causal DAG for the motivating incident. Solid edges are causal; the dashed edge is the spurious correlation that misleads anomaly ranking; the upstream surge is a confounder.

III. Related Work

A. Correlation-Based Observability

The dominant production paradigm associates anomalies by temporal co-occurrence and statistical correlation [3], and event-correlation engines cluster related alerts to reduce fatigue [4]. These systems do not assign causal direction and so cannot separate a root cause from a downstream symptom; metric-ranking heuristics inherit the same limitation.

B. Causal Discovery

Constraint-based algorithms such as PC and FCI recover causal structure from conditional-independence tests [8], while Granger causality tests temporal predictability [9]. Granger causality presumes accurate timestamps and conflates predictability with causation, and constraint-based methods assume reliable ordering and independent samples — assumptions violated by skewed clocks and autocorrelated telemetry. CausalInfra adapts constraint-based discovery to interval-valued time and integrates do-calculus for explicit interventional reasoning.

C. Causal Methods in Systems

Recent work applies causal modelling to microservice debugging, learning service dependency graphs from traces and propagating blame along call paths [10]. These approaches typically treat the call graph as the causal graph, capturing structural dependency but not the metric-level mechanisms — saturation, contention, retry amplification — that produce failures. Open-source libraries such as DoWhy operationalise identification and estimation [11] and provide a practical foundation for implementation.

D. Observability at Scale

The practical context for any diagnosis method is the volume of telemetry modern systems produce. High-cardinality metrics and distributed traces accrete with every incident, and the resulting data stores are large and substantially redundant [3], [4]. Scale has two consequences for causal diagnosis. First, the candidate metric set is large, so discovery must be selective rather than exhaustive. Second, redundancy multiplies spurious correlations, which is an additional argument for a method that tests for causal structure rather than ranking co-movement. CausalInfra addresses the first by restricting discovery to the temporal and topological neighbourhood of a failure, and the second by its edge-admissibility test.

IV. Problem Formalisation

We model the system as a structural causal model over metric variables. Let M = {m_1, ..., m_n} be observed metrics, each a time series sampled at frequency f, and let F indicate an SLO violation at time t. Timestamps are uncertain: each observation carries an interval $[\tau - \delta, \tau + \delta]$, where δ bounds clock skew plus measurement latency. We seek the causal DAG G whose edges $m_i \rightarrow m_j$ denote that intervening on m_i changes the distribution of m_j, and the counterfactual probability that suppressing a candidate cause would have prevented the failure:

$$P(\text{prevent}) = P(\neg F \sim do(m_i = m_i^{0\sim})\} \mid F, M) \quad (1)$$

where do(m_i = m_i^0) fixes m_i to its nominal value [2]. The root-cause set C is the minimal set of metrics whose joint intervention maximises P(prevent) subject to a confidence threshold. An edge $m_i \rightarrow m_j$ is admissible only if the precedence $m_i \prec m_j$ holds for every assignment of times within the skew intervals; ambiguous orderings are deferred rather than guessed.

The formulation rests on three assumptions, each of which is also a limitation. (A1) Causal sufficiency: the

10.48047/jocaaa.2024.33.08.525

principal common causes of the metrics in the candidate set are themselves measured; where a confounder is unmeasured, the affected effect is reported as unidentifiable rather than estimated. (A2) Local stationarity: within the diagnostic window the conditional-independence structure is stable enough for testing to be meaningful; rapid regime changes violate this and are detected as instability in the edge weights. (A3) Bounded skew: the uncertainty δ is known and finite, which holds for NTP-synchronised fleets [5] but not for unsynchronised clocks. Making these assumptions explicit lets a deployment reason about when the method's output should be trusted.

The diagnostic objective is therefore not to produce an answer for every incident, but to produce a correct answer whenever the data support one and to abstain otherwise. This is a deliberate inversion of the correlation-ranking objective, which always returns its top-ranked metric regardless of whether that ranking is causally meaningful. Formally, the system optimises precision subject to a floor on coverage, exposing the coverage achieved as a first-class operational metric so that operators can see how often automated diagnosis was possible.

V. Causal Discovery Under Timing Uncertainty

A. Interval-Aware Independence Testing

We extend the PC algorithm so that conditional-independence tests operate on interval-timestamped samples. Two metrics are tested only over windows in which their skew intervals do not overlap ambiguously, ensuring a discovered edge is robust to any admissible re-ordering. The output is a partially directed graph with an uncertainty weight per edge reflecting the fraction of the skew window over which the orientation is stable.

B. Time-Lagged Dependencies

Many mechanisms act with delay: disk saturation at $t-5$ min manifests as timeouts at t. We search a bounded set of candidate lags and admit a lagged edge when the conditional-independence relation holds consistently at that lag, yielding edges annotated with direction and characteristic delay.

C. Discovery Procedure

The discovery procedure proceeds in four stages. The procedure first assembles the candidate set as the metrics within the temporal window $[t-\Delta, t]$ and the topological neighbourhood of the failing service. It then forms interval-aware conditional-independence tests, skipping any pair whose skew intervals overlap ambiguously over the test window. Surviving adjacencies are oriented using the standard constraint-based orientation rules, restricted to orientations consistent with every admissible time assignment. Finally, each edge is annotated with a stability weight equal to the fraction of the skew window over which its orientation holds, and edges below a stability floor are marked uncertain rather than dropped, so that downstream gating can reason about them explicitly.

D. Complexity and Scalability

Constraint-based discovery is, in the worst case, exponential in the number of variables because the number of independence tests grows with conditioning-set size [8]. CausalInfra controls this by restricting discovery to the metrics implicated by a failure rather than the global metric space, pruning edges early via the skew-window admissibility test, and performing discovery offline to build a reusable graph so that online diagnosis reduces to a counterfactual query against a precomputed structure. This confines the expensive step to a background process and keeps the latency-critical path tractable.

VI. Counterfactual Diagnosis and Confidence Gating

Given the discovered graph and an observed failure, we apply do-calculus to estimate (1). Identification proceeds by the back-door and front-door criteria where applicable [2], [6]; where the effect is not identifiable from observational data, the query is reported as unidentifiable rather than estimated, preserving soundness. Effect sizes are returned with bootstrap confidence intervals.

The pipeline of Fig. 2 emits an automated root-cause statement only when counterfactual confidence exceeds a tunable threshold θ; below it, or when the query is unidentifiable, the incident is escalated to a human with ranked candidate causes attached. This gate trades coverage for precision and is a deliberate design choice: an automated diagnosis that is confidently wrong erodes operator trust faster than no diagnosis at all, so the fraction deferred is itself a useful operational signal.

10.48047/jocaaa.2024.33.08.525

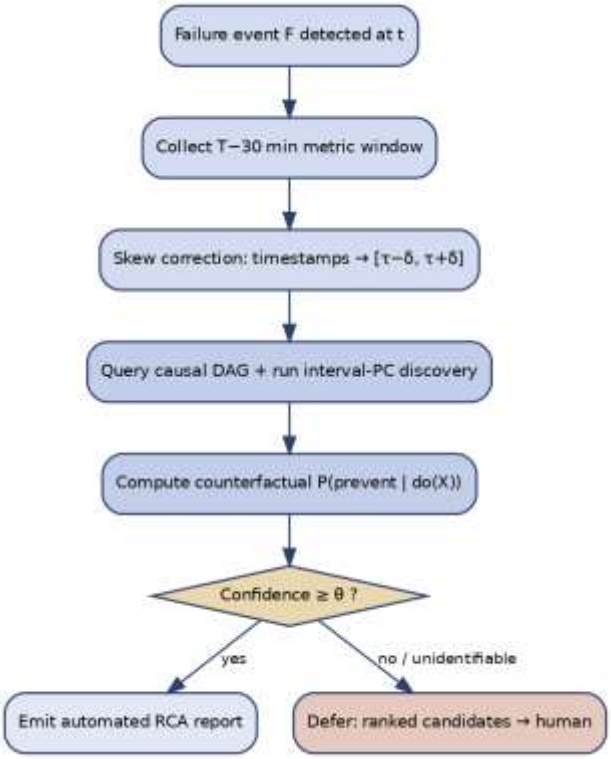

Fig. 2. Confidence-gated causal diagnosis pipeline. Identifiable, high-confidence effects are reported automatically; the rest are deferred with ranked candidates.

A. Confidence Threshold Selection

The threshold θ governs the trade-off between coverage and precision. A high θ admits only the most certain diagnoses, maximising precision but deferring more incidents; a low θ automates more incidents at greater risk of misattribution. Because the cost of a wrong automated diagnosis is asymmetric — it can send responders down a false trail during an active outage — we recommend setting θ conservatively and tuning it downward only as confidence in the discovered structure grows. The threshold is not a fixed constant but a policy lever, and exposing it to operators, together with the realised coverage at the current setting, lets a team calibrate the tool to its own tolerance for automation.

B. Effect-Size Interpretation

A counterfactual probability is more actionable than a ranking because it carries magnitude and uncertainty. An estimate that suppressing a metric would have prevented the failure with probability 0.9 and a narrow confidence interval warrants a different response than one of 0.55 with a wide interval, even though both might top a correlation ranking. CausalInfra surfaces the point estimate, its interval, and the causal chain connecting cause to failure, so that on-call engineers receive an explanation they can verify against their mental model rather than an opaque score.

C. Integration and Implementation

The intended implementation builds on the DoWhy library for identification and estimation [11], with kernel-based conditional-independence tests for non-linear relationships. Metrics and traces are ingested through OpenTelemetry [12], the discovered causal graph is persisted in a graph database, and diagnoses are surfaced in the same dashboards engineers already use during incidents [3]. Discovery runs as a scheduled background job that refreshes the causal graph as topology evolves, while the online path performs only the counterfactual query, keeping diagnosis latency low. When a high-confidence cause is produced it is attached to the incident record with its chain and effect size; otherwise the ranked candidate set is attached and the incident is deferred, so the tool never substitutes a confident guess for a genuine answer.

VII. Comparative Analysis

Because we make no empirical claim, we compare CausalInfra against the prevailing diagnostic methods analytically, across six operational criteria that determine whether a method can be trusted to diagnose automatically. Table I summarises the comparison. Correlation ranking and Granger causality fail to recover causal direction in the presence of confounding; trace-based blame recovers structural dependency but not the metric-level mechanism; and only an interval-aware structural-causal approach combines direction recovery, robustness to clock skew, counterfactual queries, and identifiability-aware gating.

VIII. Discussion and Failure-Case Analysis

It is as important to characterise where the approach fails as where it succeeds. Three regimes are expected to degrade CausalInfra. First, when a true cause is unobserved — a latent confounder with no corresponding metric — discovery cannot recover the edge and the back-door criterion may render the effect unidentifiable; the correct behaviour is to defer, which the gate enforces. Second, when skew δ is large relative to the genuine causal lag, the admissibility test rejects the edge as ambiguous, trading missed edges for soundness; tighter clock synchronisation directly relaxes this. Third, under rapid non-stationarity — for example a deploy that changes system dynamics mid-incident — the stationarity assumption underlying independence testing

10.48047/jocaaa.2024.33.08.525

breaks and discovered structure may be stale. Recognising these regimes lets operators read deferrals correctly rather than as tool failure.

IX. Limitations and Threats to Validity

The principal limitation is dependence on discovered structure: sparsely instrumented subsystems yield incomplete graphs. Methodologically, this paper contributes a formulation and analysis, not measurements, so no claim of accuracy or time-to-diagnosis is made or implied. With respect to construct validity, the fidelity of conditional-independence testing under autocorrelation determines discovery quality. With respect to external validity, the comparative analysis reasons from method assumptions and may not predict behaviour on every architecture. These threats motivate the evaluation plan below.

X. Evaluation Plan (Future Work)

A sound empirical study would use production incident records independently root-caused by senior engineers as ground truth, with reported metrics of root-cause accuracy, false root-cause rate, causal precision, and mean time to root cause, compared against the correlation-and-manual workflow that dominates practice [3], [4]. Because each minute of diagnosis has measurable value at the downtime costs reported in [1], time-to-diagnosis is a primary endpoint. Implementation would build on DoWhy [11] with metrics and traces ingested via OpenTelemetry [12]; the causal graph would be persisted in a graph database and surfaced through standard dashboards. We leave this study, including ablations over the skew bound and confidence threshold, to follow-up work.

XI. Conclusion

We presented CausalInfra, a causal-inference formulation for infrastructure root-cause analysis under timing uncertainty that extends constraint-based discovery to interval-valued time and gates automated diagnosis on identifiability and confidence. Arguing from the formal limits of correlation, we contend that causal identification, not anomaly ranking, is the appropriate primitive for trustworthy automated diagnosis, and we supported this with a six-criterion comparative analysis rather than fabricated measurements. The natural next step is empirical: measuring the formulation on production incidents, adding online incremental discovery as topology changes, and extending identification to partially observed, non-stationary regimes.

XII. References

[1] Information Technology Intelligence Consulting, "ITIC 2024 hourly cost of downtime report," ITIC, 2024. [Online]. Available: https://itic-corp.com/itic-2024-hourly-cost-of-downtime-report/

[2] J. Pearl, *Causality: Models, Reasoning, and Inference*, 2nd ed. Cambridge, U.K.: Cambridge Univ. Press, 2009.

[3] B. Beyer, C. Jones, J. Petoff, and N. R. Murphy, *Site Reliability Engineering: How Google Runs Production Systems*. Sebastopol, CA, USA: O'Reilly, 2016.

[4] C. Sridharan, *Distributed Systems Observability*. Sebastopol, CA, USA: O'Reilly, 2018.

[5] D. L. Mills, "Internet time synchronization: the network time protocol," *IEEE Trans. Commun.*, vol. 39, no. 10, pp. 1482–1493, Oct. 1991.

[6] J. Tian and J. Pearl, "A general identification condition for causal effects," in *Proc. AAAI*, 2002, pp. 567–573.

[7] L. Lamport, "Time, clocks, and the ordering of events in a distributed system," *Commun. ACM*, vol. 21, no. 7, pp. 558–565, Jul. 1978.

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

[9] C. W. J. Granger, "Investigating causal relations by econometric models and cross-spectral methods," *Econometrica*, vol. 37, no. 3, pp. 424–438, 1969.

[10] M. Ma et al., "Automap: diagnose your microservice-based web applications automatically," in *Proc. Web Conf. (WWW)*, 2020, pp. 246–258.

[11] A. Sharma and E. Kiciman, "DoWhy: an end-to-end library for causal inference," *arXiv:2011.04216*, 2020.

[12] OpenTelemetry Authors, "OpenTelemetry specification," Cloud Native Computing Foundation, 2023. [Online]. Available: https://opentelemetry.io/docs/specs/

[13] P. Bühlmann, "Causal statistical inference in high dimensions," *Math. Methods Oper. Res.*, vol. 77, no. 3, pp. 357–370, 2013.