AGENTIC RCA UNDER PARTIAL OBSERVABILITY: FROM ROOT OBSERVATION TO DIAGNOSIS

Anonymous authors
Paper under double-blind review

ABSTRACT

LLM-based root-cause analysis (RCA) agents iteratively inspect metrics, logs, and traces to identify the source of an incident, but fault propagation and concurrent activity can make non-root components plausible competing explanations. Final-answer accuracy alone cannot reveal whether telemetry from the root-cause component was retrieved, whether the root was considered or favored, or whether it reached the final diagnosis. We analyze 335 OpenRCA queries by aligning incident-side signal conditions with retrieved telemetry, expressed diagnostic states, and final diagnoses. Later root signals and especially stronger non-root competition accompany poorer diagnosis, whereas broader anomalous activity does not show the same pattern. Root retrieval shows no consistent association with non-root competition. After observing the root, however, agents facing stronger competition are less likely to favor it or select it in the final diagnosis. We further evaluate three targeted agent designs on 22 queries. Although these designs improve component accuracy, residual failures show that additional structure can leave competition unresolved. These findings motivate *cross-hypothesis discrimination*—using evidence to distinguish among causal hypotheses—as a design target for agentic RCA.

1 INTRODUCTION

In modern cloud and microservice systems, a fault in one component can propagate through dependencies across services and infrastructure layers (Wang et al., 2018; Yu et al., 2021; Li et al., 2022; Xu et al., 2025). Monitoring can consequently reveal anomalous signals at affected non-root components earlier or more prominently than at the root-cause component. RCA agents must therefore distinguish the causal source from plausible non-root alternatives rather than merely identify anomalous components. This challenge is especially relevant for LLM-based RCA agents, which iteratively inspect telemetry, choose what to query next, update candidate root causes, and eventually produce a final diagnosis (Xu et al., 2025; Jha et al., 2026). A prominent non-root signal can thus shape the investigation before telemetry from the root-cause component is retrieved.

Recent LLM-based RCA systems support such investigation through topology and operational knowledge, tool use, procedural guidance, hypothesis management, and collaboration among agents (Chen et al., 2024; Wang et al., 2024; Zhang et al., 2024; Pei et al., 2025; Gao et al., 2026; Sanwouo et al., 2026; Jha et al., 2026). Evaluation has also expanded beyond final accuracy to examine retrieved telemetry, tool-use trajectories, causal structure, and step-level process quality (Fang et al., 2026a; Wang et al., 2026b; Gong et al., 2026; Fan et al., 2026), while failure analyses characterize why agents fail despite access to multimodal telemetry (Kim et al., 2026). We ask a complementary question: *how do competing non-root signals relate to diagnosis after a concrete root observation, and can targeted agent designs help agents distinguish the root from plausible rivals?*

We study 335 OpenRCA queries (Xu et al., 2025). Independently of agent behavior, we characterize three root-relative properties of incident telemetry: *relative peak timing*, whether the root peaks later than surrounding components; *rival prominence*, the prevalence of non-root components with stronger anomalies than the root; and *concurrent anomaly breadth*, the extent of anomalous activity across the system. We align these incident-side signal conditions with telemetry retrieved during agent execution, expressed diagnostic states in the trace, and the final diagnosis. This distinguishes whether root telemetry is retrieved from whether the root enters the candidate set, becomes the leading diagnosis, and is included in the final diagnosis. We then evaluate three targeted agent designs to

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

examine whether different forms of diagnostic support can mitigate this post-observation gap and where it remains unresolved.

Our agent-execution analysis reveals a gap between encountering the root and favoring it as the cause. The three incident-side signal conditions show different associations with diagnosis: later relative peak timing and especially greater rival prominence accompany poorer diagnostic outcomes, whereas concurrent anomaly breadth does not show the same consistent pattern. Root telemetry retrieval shows no consistent association with rival prominence. Among histories with a concrete root observation, however, greater rival prominence is associated with a lower likelihood that the root becomes the leading diagnosis or is included in the final diagnosis. Most additional root omissions under stronger competition arise in histories where the root never becomes the leading diagnosis. Thus, an agent can encounter root telemetry while continuing to favor another component.

The targeted designs examine different ways to support diagnosis. On 22 Telecom incidents, all three achieve higher component accuracy than baseline. Some native failures in which root telemetry had already been observed are also recovered. Their residual failures nevertheless include cases in which evidence enters intermediate judgments, belief updates, or review without resolving competition with a plausible alternative hypothesis. These cases highlight a distinction between accumulating support for a candidate and establishing why that candidate should be preferred over alternatives. Together, the gains and residual failures motivate *cross-hypothesis discrimination*—using available evidence to distinguish a causal source from plausible rivals—as a design target beyond evidence acquisition and process structure alone. Our contributions are threefold:

- **Separating root retrieval from diagnostic recognition.** We align incident-side signal conditions, retrieved telemetry, expressed diagnostic states, and final diagnoses to distinguish root retrieval, candidate inclusion, diagnostic leadership, and final root selection.
- **Revealing a post-observation diagnostic gap under competing signals.** We show that greater rival prominence is associated with a lower likelihood that the observed root becomes the leading diagnosis or reaches the final answer, revealing a post-observation gap that final accuracy alone cannot locate.
- **Examining the limits of targeted diagnostic support.** We develop and evaluate three targeted agent designs on matched incidents. Their accuracy gains and residual failures motivate cross-hypothesis discrimination as a design target beyond evidence acquisition and process structure alone.

2 Background and Problem Formulation

2.1 Agentic RCA under Partial Observability

Agentic RCA proceeds through sequential investigation: an agent selects which part of the accessible telemetry to inspect and updates its diagnosis from the returned observations (Xu et al., 2025; Wang et al., 2024; Pei et al., 2025). Observation scope may omit relevant measurements or expose fault-related information only indirectly (Xie et al., 2024). Even within that scope, the telemetry accessible to the agent is distinct from the information that its selected investigations actually return.

Consider one execution on an incident z. Let K denote its observation scope and let h_0 contain the initial query and instructions. An agent with policy π proceeds as

$$a_t \sim \pi(\cdot \mid h_{t-1}), \qquad o_t = O_K(z, a_t), \qquad h_t = (h_{t-1}, a_t, o_t), \qquad \hat{d} = g(h_T). \tag{1}$$

Here, a_t is the selected investigation, o_t is its returned content, and T is the stopping time. Observation scope constrains what can be inspected, while the investigation policy determines what enters the history. The final diagnosis therefore depends on an investigation-dependent view of the incident rather than its complete state.

2.2 From Observation to Diagnostic Commitment

Partial observability describes the boundary on what an agent can inspect. Separately, even under the same observation scope, incidents can differ in how the root-cause component appears relative to non-root signals in the available telemetry. We refer to these root-relative properties of the signal

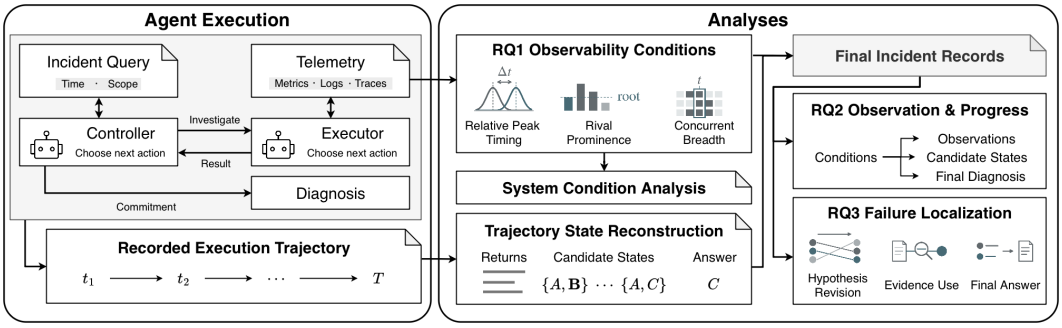


Figure 1: **Analysis Pipeline.** We run Agent-RCA from OpenRCA to get execution trajectory of agents. We align incident-side signal conditions, returned observations, candidate states, and final diagnosis to analyze relationships between them.

environment as *incident-side signal conditions*. They characterize the relative timing, prominence, and distribution of incident signals.

Observation and diagnosis are also distinct. A returned observation about a component does not ensure that the agent considers it a possible cause, and consideration does not ensure that the component becomes the favored or final diagnosis. We call a component represented as a possible cause a *candidate*, and the candidate currently favored by the agent the *leading diagnosis*. We use *expressed diagnostic state* for the candidate configuration and leading diagnosis represented in the visible trace after returned observations. These states are observable properties of the recorded trajectory rather than claims about the agent's latent belief. For incident i and execution r, we distinguish

$$\mathcal{M}_{ir} = \left(E_i,\ H^{\pi}_{ir},\ \mathbf{S}_{ir},\ \hat{d}_{ir}\right), \qquad \mathbf{S}_{ir} = (S_{ir,1}, \ldots, S_{ir,L_{ir}})\,. \tag{2}$$

Here, E_i denotes incident-side signal conditions measured from incident telemetry independently of the agent trace; H^{π}_{ir} is the realized investigation history; $\mathbf{S}_{ir}$ is the ordered sequence of expressed diagnostic states reconstructed from the visible trace; and $\hat{d}_{ir}$ is the final diagnosis. The sequence contains L_{ir} measured states for that execution. We link E_i to each execution by incident identity and preserve the within-execution ordering of returned observations, expressed states, and diagnosis. This alignment organizes the empirical analysis without assuming a causal or monotonic progression.

3 Related Work

RCA methods and agentic systems. Data-driven RCA uses metric, trace, and multimodal telemetry to rank candidate causes, while work on limited observability additionally considers root causes that are not directly monitored (Wang et al., 2018; Yu et al., 2021; Li et al., 2022; Pham et al., 2024; Xie et al., 2024). LLM-based systems support active diagnosis through operational retrieval, tool execution, workflow guidance, specialist collaboration, adaptive agent composition, and explicit hypothesis management (Chen et al., 2024; Wang et al., 2024; Zhang et al., 2024; Pei et al., 2025; Gao et al., 2026; Sanwouo et al., 2026; Wei & Fink, 2026; Bi et al., 2026). Graph-guided exploration and belief revision further connect sequential observation to causal discovery under partial observability (Jha et al., 2026). These methods improve how agents acquire and organize diagnostic information. Our work is complementary: rather than introducing a new RCA architecture, we analyze agent executions to examine how incident-side signal conditions relate to the telemetry retrieved during investigation and how the expressed diagnostic state subsequently evolves.

RCA benchmarks and process evaluation. RCAEval consolidates localization datasets and reproducible baselines, while OpenRCA evaluates LLM agents that investigate enterprise telemetry with tools (Pham et al., 2025; Xu et al., 2025). Other benchmarks emphasize realistic fault propagation, grounded explanatory paths, tool-use trajectories, or production-style diagnosis (Fang et al., 2026b;a; Wang et al., 2026b; Gong et al., 2026). These benchmarks provide complementary views of terminal correctness and realized investigation quality. Our goal is not to introduce another generic trajectory-quality measure. We instead align incident-side signal conditions measured from incident telemetry with the retrieved telemetry from native investigation, the subsequent position of the root-cause component in the expressed diagnostic state, and the final diagnosis within matched incidents.

Failure analysis, process attribution, and intervention. Prior RCA analyses characterize recurring reasoning and communication failures and distinguish unavailable supporting evidence from evidence that was available but unused (Kim et al., 2026; Riddell et al., 2026; Gopal & Krishnan, 2026). AgentProcessBench (Fan et al., 2026) evaluates step-level process quality, while intervention-based methods alter intermediate steps or information to attribute failures and test repairs (Bonagiri et al., 2026; Wang et al., 2026a; Luo & Peng, 2026). These approaches classify, evaluate, or intervene on realized executions. We ask a different but complementary question: which diagnostic paths become more frequent as incident-side competition increases, including after a concrete root observation.

4 Empirical Motivation

Recent RCA agents have advanced through tool use, topology, hypothesis management, and multi-agent coordination (Wang et al., 2024; Pei et al., 2025; Wei & Fink, 2026; Bi et al., 2026). Table 1 compares full and partial-credit performance across three agent configurations on the same 335 OpenRCA queries. Although the stronger configurations substantially improve over GPT-4o, roughly two thirds of queries still fail to receive full credit. Final accuracy, however, does not reveal what makes an incident difficult or where the agent's diagnosis begins to diverge from the labeled root-cause component.

Table 1: **Accuracy on 335 OpenRCA queries**. All requested fields must be correct to be Correct.

Configuration	Correct	Partial
GPT-4o	6.6%	24.5%
Claude	31.6%	53.7%
GPT-5.5	33.1%	54.9%

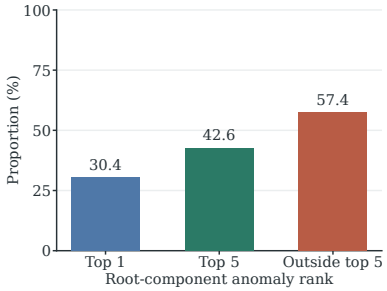


Figure 2: **Root component outside the strongest initial signals.** Top 1 is included in Top 5; Outside top 5 is its complement.

Consider a simple case in which component A is the root-cause component, but fault propagation produces an earlier or stronger anomaly at component B. An agent may reasonably investigate B first. Even if telemetry from A is later retrieved, several outcomes remain possible: A may never enter the candidate set, it may enter the candidate set but remain secondary to B, or it may eventually become the leading diagnosis. Thus, the same final error can arise through different diagnostic trajectories as the agent responds to the incident-side signal conditions.

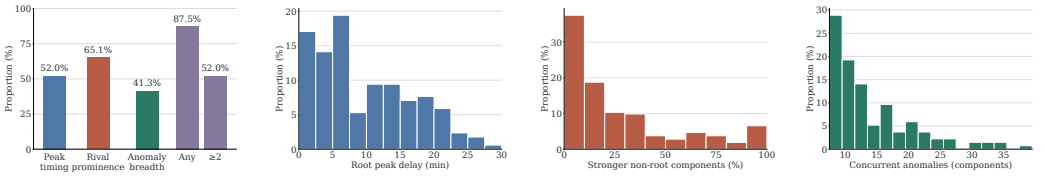
This setting is not rare in OpenRCA. In a five-minute onset snapshot of 148 Market incidents, the labeled root-cause component is the strongest anomalous component in only 30.4% of cases and appears among the top five in 42.6%; its median rank is 12 (Figure 2). The root-cause component is therefore often not the most salient location at the start of diagnosis, requiring the agent to distinguish it from stronger-looking alternatives rather than simply follow the largest anomaly. To understand how agents respond to such alternatives, we first need to characterize the incident-side signal conditions independently of what agents choose to investigate. This motivates examining how incident-side signal conditions are reflected in the subsequent diagnostic trajectory.

5 Analysis

5.1 Research Questions

Under partial observability, RCA agents must infer an incident's cause from the observations returned by their investigations. The observation scope constrains what they can inspect, while the agent's investigation determines which telemetry is actually retrieved. A concrete root observation does not by itself establish the root-cause component as the cause. We therefore examine incident-side signal conditions, their associations with retrieved telemetry and diagnostic progression, and test whether targeted agent designs can mitigate the gap between observation and discrimination of the root.

- **RQ1—Incident-Side Signal Conditions:** What incident-side signal conditions characterize the telemetry available for root-cause diagnosis?



(a) Prevalence (b) Relative peak timing (c) Rival prominence (d) Concurrent anomaly breadth

Figure 3: **Prevalence and variation of incident-side signal conditions.** (a) Shares of 327 supported queries with peak timing > 0, rival prominence > 0, or anomaly breadth ≥ 8. Any and ≥ 2 indicate at least one and at least two conditions. (b)–(d) Distributions within the corresponding subsets ($n = 170$, 213, and 135), each normalized to 100%. Appendix B defines the measures.

- **RQ2—Retrieved Telemetry and Diagnostic Progression:** How are these conditions associated with telemetry retrieved during agent investigation and with diagnostic progression after a concrete root observation?
- **RQ3—Improving Diagnosis after Root Observation:** Can targeted RCA-agent designs improve diagnosis after a concrete root observation, and what limitations remain when competing hypotheses persist?

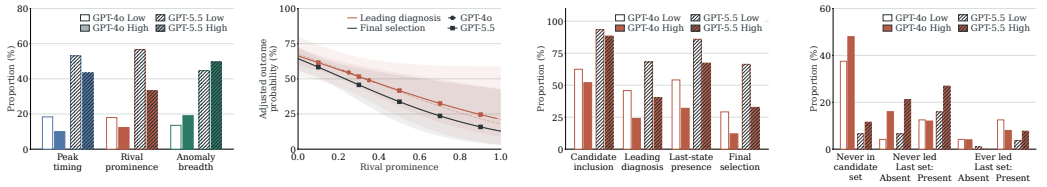
5.2 Analysis Setting

We analyze 335 OpenRCA queries from Bank, Telecom, and Market (Xu et al., 2025). In the Agent-RCA workflow from OpenRCA, a Controller directs an Executor to write and run code over metrics, logs, and traces. The Controller reads each return and decides whether to investigate further or submit a final diagnosis (Figure 1). For each native agent execution, our analysis links incident-side signal conditions, retrieved telemetry, and the expressed diagnostic state, and aligns them with the final diagnosis. We first compute incident-side signal conditions from benchmark telemetry independently of the agent trace. We then identify telemetry retrieved during investigation and reconstruct the expressed diagnostic state from the visible Controller trace using an LLM-based state reader. Reader prompts and measurement details are provided in Appendix A.2. For RQ1–RQ2, we use three repeated executions for each of GPT-4o and GPT-5.5, with one Claude Sonnet 4.6 execution used for supplementary analysis of concrete root observation. For RQ3, we additionally evaluate three targeted RCA-agent design variants on a matched set of Telecom component-diagnosis incidents and inspect their recorded trajectories to understand both successful recoveries and remaining failures. Full execution settings and statistical specifications are provided in Appendix A.

5.3 RQ1: Incident-side Signal Conditions

We characterize three incident-side signal conditions that capture different ways the root-cause component appears relative to non-root activity. Relative peak timing captures whether the root reaches its strongest anomaly later than eligible non-root components, rival prominence captures the prevalence of non-root components with stronger anomalies than the root, and concurrent anomaly breadth captures how widely anomalous activity spans components. We compute all three independently of agent trajectories and outcomes. Among 327 supported queries, the root peaks later than the median eligible non-root component in 52.0%, at least one stronger non-root component appears in 65.1%, and breadth reaches eight or more components in 41.3%. Figure 3 summarizes their prevalence and variation; the examples below illustrate how each condition can appear in an agent investigation, with detailed definitions in Appendix B.

Relative peak timing. Temporal separation can complicate causal interpretation when the root reaches its strongest anomaly after activity elsewhere in the system. For example, the true root-cause component is `paymentservice`, with relative peak timing of 20.5 minutes in a Market incident. Telemetry shows `frontend` pod CPU anomalies before the later payment-related anomalies, and the agent explicitly uses this ordering when assessing the causal location. The agent favors `frontend` alternatives over `paymentservice`, investigates those alternatives, and ultimately selects `frontend`. Likewise, relative timing can shape the agents' diagnostic process.



(a) Final root selection (b) Adjusted associations (c) Diagnostic progression (d) Paths to root omission

Figure 4: **Diagnostic progression under incident-side signal conditions.** (a) Final root selection across conditions. (b) Adjusted outcome probabilities across rival prominence. (c) Post-observation progression by rival prominence. (d) Mutually exclusive paths to final root omission. Bands and error bars show pointwise 95% confidence intervals. Details appear in Appendices D and E.

Rival prominence. Strong non-root signals can make alternative causal locations appear more compelling. In one incident in Market system, the labeled root-cause component is `node-6`, but several `cartservice` pods have larger standardized deviations. The agent likewise favors a pod-level explanation based on these larger deviations and ultimately selects `cartservice-2`, citing its extreme memory page-fault activity. The same page-fault feature also makes `cartservice-2` one of the non-root components whose anomaly exceeds the root in our rival-prominence measurement. The case illustrates how relative anomaly magnitude can shape which component is favored as the cause and where subsequent investigation is directed.

Concurrent anomaly breadth. Broad anomalous activity can leave multiple components as plausible candidates within the same investigation. In another incident of Market system, concurrent anomaly breadth is 10, indicating anomalous activity across ten components within the event window, including five nodes. After identifying anomalies across multiple nodes, the agent selects `node-6` because it has the largest number of detected faults. The investigation then focuses on that node's resource metrics, and the final diagnosis attributes the incident to `node-6`, although the true root-cause component is `node-3`. The case illustrates the localization challenge associated with breadth: distinguishing the root-cause component from several concurrently anomalous alternatives.

RQ1 Answer. Telemetry from the root-cause component does not appear in isolation. Across OpenRCA incidents, the root can reach its strongest anomaly later than non-root components, appear with stronger non-root signals, or lie within broadly distributed anomalous activity. These signal conditions present distinct diagnostic challenges involving temporal order, relative anomaly strength, and concurrent activity across components. RQ2 examines how these conditions are associated with retrieved telemetry and the subsequent diagnostic progression of the root-cause component.

5.4 RQ2: Retrieved Telemetry and Diagnostic Progression

We next examine how the three incident-side signal conditions in RQ1 relate to diagnosis, with particular attention to what happens after a concrete root observation. A *concrete root observation* occurs when returned content contains the root-cause component together with a concrete metric value, log event, or trace datum. Among histories with such an observation, we track whether the root-component enters the candidate set, becomes the leading diagnosis, remains present in the last valid pre-final state, and is included in the final diagnosis.

Final Root-Component Selection. We first compare low and high values of each incident-side signal condition using fixed 25th- and 75th-percentile cutoffs within each dataset in Figure 4(a). Under later relative peak timing, final root selection decreases from 18.4% to 10.0% for GPT-4o and from 53.1% to 43.3% for GPT-5.5. Under greater rival prominence, it decreases from 18.0% to 12.3% and from 56.6% to 33.3%, respectively. Concurrent anomaly breadth shows no corresponding decrease (13.6% to 19.0% for GPT-4o; 44.6% to 49.7% for GPT-5.5). Thus, both later relative peak timing and greater rival prominence accompany lower final root selection, whereas concurrent anomaly breadth does not show the same pattern. Among the three conditions, rival prominence shows the clearest descriptive relationship with diagnostic progression after a concrete root observation, so we use it to examine how stronger non-root signals relate to the subsequent standing of the root-cause component.

Rival Prominence and Post-Observation Diagnosis. Across the continuous rival-prominence range, the probability that the root becomes the leading diagnosis or is included in the final diagnosis decreases as stronger non-root components become more prevalent (Figure 4(b)). This relationship is clearest for GPT-5.5, while GPT-4o shows the same directional pattern with greater uncertainty. Importantly, rival prominence shows no consistent relationship with whether a concrete root observation occurs across agent executions (Appendix Figure 7(b)). Figure 4(c) instead shows that the separation becomes more pronounced after observation. Candidate inclusion changes comparatively little between low- and high-prominence histories, whereas the gap is substantially larger in whether the root becomes the leading diagnosis and remains visible at final root selection. For GPT-5.5, the low–high difference grows from 5.0 percentage points for candidate inclusion to 27.8 points for becoming the leading diagnosis. A further decomposition shows that much of this difference is already present when the root first enters the candidate set, rather than arising only from poorer subsequent promotion after a non-leading entry (Appendix E.2). Thus, greater rival prominence is more clearly associated with whether it becomes the leading diagnosis.

Additional Failures. Among the same histories with a concrete root observation, the final diagnosis omits the root 17.2 percentage points more often under high rival prominence for GPT-4o and 33.7 points more often for GPT-5.5 (Figure 4(d)). In both models, most of this additional failure is concentrated in paths where the root never becomes the leading diagnosis. These paths include failure to enter the candidate set, disappearance after candidate inclusion, and continued candidate presence without ever overtaking a competing root-cause hypothesis. Disappearance after candidate inclusion increases under high prominence in both models, showing that the gap is not confined to initial candidate inclusion. Continued presence without becoming the leading diagnosis increases substantially for GPT-5.5 but remains nearly unchanged for GPT-4o. Across both models, most of the additional omissions under greater rival prominence therefore arise before the root becomes the leading diagnosis, although the specific paths differ.

RQ2 Answer. Later relative peak timing and especially greater rival prominence are associated with poorer diagnostic progression, whereas concurrent anomaly breadth does not show the same pattern. Rival prominence is not consistently associated with whether telemetry from the root-cause component is retrieved. Its clearest relationship appears afterward: under greater rival prominence, the root-cause component is less likely to become the leading diagnosis or be included in the final diagnosis, and most additional omissions occur before it ever becomes the leading diagnosis. In other words, agents can encounter the root without recognizing it as the cause.

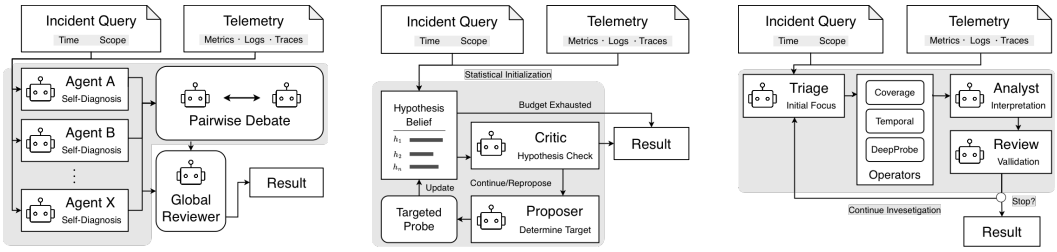
5.5 RQ3: Improving Diagnosis after Root Observation

RQ2 shows that retrieving telemetry from the root-cause component does not ensure that the root becomes the favored diagnosis, particularly when stronger non-root signals compete with it. We therefore ask whether agent designs that support comparison, hypothesis revision, and diagnostic review can improve diagnosis after a concrete root observation, and what limitations remain.

Agent Designs and Overall Outcomes. We evaluate three targeted agent designs: Local-to-Global Diagnosis (LGD), Hypothesis-Guided Investigation (HGI), and Structured Diagnostic Control (SDC). Figure 5 summarizes their organization, and design details are provided in Appendix A.4. Each targets a different part of the post-observation problem identified in RQ2: LGD is designed to preserve and integrate component-level judgments, HGI uses explicit hypotheses to guide evidence-driven revision, and SDC structures candidate tracking, investigation, evidence updates, and review. All three produce higher component accuracy than Agent-RCA on the 22 matched Telecom incidents (Table 2). Among the four native failures in which root telemetry had already been observed, HGI recovers two incidents, LGD one, and SDC none. We examine the trajectories to identify what enabled these recoveries and where competition among plausible root-cause hypotheses remained unresolved despite these mechanisms.

Table 2: **Component accuracy on 22 Telecom incidents.** Rec. and Reg. denote recoveries among 18 failures and regressions among four successes from Agent-RCA.

Model	Agent	Comp. acc.	Rec.	Reg.
GPT-4o	Agent-RCA	18.2%	–	–
	LGD	31.8%	5/18	2/4
	HGI	27.3%	4/18	2/4
	SDC	22.7%	4/18	3/4



(a) LGD (b) HGI (c) SDC

Figure 5: **Alternative agent architectures.** (a) **LGD** combines component-level self-diagnosis, pairwise debate, and global review. (b) **HGI** maintains statistically initialized hypothesis beliefs and uses a Critic–Proposer loop for targeted probing and belief updates. (c) **SDC** coordinates coverage, temporal, and deep-probe operators with evidence interpretation and iterative diagnostic review.

Local-to-Global Diagnosis (LGD). When multiple components exhibit prominent anomalies, global investigation can prioritize some alternatives before others receive substantial consideration. LGD addresses this difficulty by first distributing diagnosis across component-specific agents. Each agent assesses its local evidence and compares causal roles with neighboring components, after which a global Reviewer integrates these assessments to identify the most plausible cause. This organization gives individual components an opportunity to establish diagnostic support before the final global comparison. Successful histories included local judgments distinguishing the root's own-resource anomaly from propagated symptoms; error-free local support for the root appeared in six of seven successful histories. However, the same local-support pattern also appeared in two of fifteen failures, where the root was favored over its local comparison partner but the Reviewer selected a different component with stronger own-resource evidence. For example, support for `db_003` over `docker_007` did not resolve its competition with `os_021`, whose large CPU anomaly drove the final choice. Three failures interpreted the root's symptoms as propagated effects from another component, and three lost an initially confirmed root claim at the Reviewer's candidate limit. Thus, LGD can surface useful local evidence, but support against one comparison partner does not necessarily establish why the root should be preferred over other candidates in the global comparison.

Hypothesis-Guided Investigation (HGI). HGI maintains a distribution over component–reason hypotheses and uses it to guide targeted probes, evidence updates, and critic judgments. This revision mechanism can correct an initial preference: in one successful incident, uninformative network probes were followed by a positive CPU observation for the root, shifting the final diagnosis to the correct component. However, its effectiveness depends on which candidates and signals are investigated. In twelve of sixteen failures, positive measurements for the eventually selected component were repeatedly incorporated into belief updates, and four of these histories ended through probability-and-margin-based early commitment. Repeated updates also occurred in three of six successes, so repetition alone does not establish a failure. The key limitation is that stronger support for an anomalous component does not necessarily establish that it is the incident's origin. These residual failures illustrate how explicit belief updates can increase confidence in a favored hypothesis within a limited comparison without resolving whether it should prevail over the remaining alternatives.

Structured Diagnostic Control (SDC). SDC organizes candidate tracking, investigation, evidence updates, and review into an explicit diagnostic loop. In all five successful histories, root-directed coverage returned evidence and the root subsequently attained numerical leadership, including one incident with broad anomalous activity and relatively late root timing. Residual failures expose two distinct limits. In six of seventeen failures, root-directed coverage returned evidence but the root never attained numerical leadership; in the remaining eleven, the root appeared in the initial focus set but was never directly targeted. The former cases retain a gap after root-directed coverage has returned evidence, whereas the latter show that focus-set membership does not ensure direct investigation. In one `db_007` failure, further probing still left `os_001` in the lead, and the final review accepted its posterior advantage and anomaly support. Across all seventeen failures, the final alternative was a component with higher signal in fourteen histories. Thus, structured investigation can support coverage and reconsideration, but these steps do not by themselves ensure that returned evidence resolves the root's competition with the leading diagnosis.

RQ3 Answer. The targeted agent designs improve component accuracy and recover some incidents in which the native agent had already observed the root but selected another component, showing that the post-observation diagnostic gap can be mitigated in some incidents. Their remaining failures nevertheless include cases in which local support for the root does not settle the global comparison, repeated evidence updates strengthen a favored hypothesis without resolving its competition with alternatives, or structured investigation returns root evidence while another component remains the leading diagnosis. These cases highlight a distinction between accumulating support for a candidate and establishing why it should be preferred over plausible alternatives. Candidate exclusion and a lack of direct investigation also remain limitations. We therefore identify *cross-hypothesis discrimination*—using available evidence to determine whether and why one root-cause hypothesis should be preferred over plausible alternatives—as an explicit design and evaluation target beyond evidence acquisition and process structure alone.

6 Discussion

6.1 From Competing Signals to Diagnostic Revision

Taken together, our analyses suggest that diagnostic difficulty under partial observability concerns not only whether root telemetry is retrieved, but also how it is interpreted relative to plausible competing hypotheses. Later relative peak timing and especially greater rival prominence accompany poorer diagnosis, whereas concurrent anomaly breadth does not show the same pattern. This contrast, together with the gap between concrete root observation and leading diagnosis, suggests that the challenge cannot be understood solely in terms of the extent of anomalous activity or the absence of root telemetry. The targeted agent designs improve component accuracy, yet their residual failures show that additional investigation, belief updates, or review can leave competition unresolved. The central distinction is therefore between accumulating support for a candidate and establishing why it should be preferred over plausible alternatives. We view cross-hypothesis discrimination as a design and evaluation target that complements evidence acquisition, organization, and review. Evaluation should distinguish concrete root observation, candidate inclusion, leading diagnosis, and final root selection across incident-side signal conditions. Since a concrete root observation does not establish that the evidence suffices to identify the cause, the key question is whether returned evidence warrants maintaining or revising a preference, rather than whether the expressed diagnostic state changes.

6.2 Robustness across Agent Executions

Repeated agent executions help distinguish population-level regularities from properties of a particular stochastic trajectory. Investigations of the same incident can differ in retrieved telemetry, expressed diagnostic states, and diagnostic trajectories leading to the final diagnosis. The adjusted analyses estimate a common within-model association across executions while allowing execution-specific baseline differences (Figure 4(b)). The aggregate progression and path analyses associate greater rival prominence with poorer post-observation progression, with most additional omissions occurring before the root becomes the leading diagnosis (Figures 4(c) and 4(d)). In the execution-specific comparisons, lower rates of the root becoming the leading diagnosis and of final root selection under high rival prominence recur across all three GPT-5.5 executions, whereas the corresponding GPT-4o contrasts vary across executions (Appendix D.3). These findings characterize aggregate associations whose consistency across agent executions depends on the model and stage of diagnostic progression.

7 Conclusion

RCA agents can retrieve telemetry from the root-cause component yet continue to favor a plausible non-root alternative. Across OpenRCA incidents, later relative peak timing and especially greater rival prominence accompany poorer diagnosis. After a concrete root observation, greater rival prominence is associated with a lower likelihood that the root becomes the leading diagnosis or is included in the final diagnosis. Targeted agent designs improve component accuracy on matched Telecom incidents, yet residual failures show that additional diagnostic structure can leave competition among causal hypotheses unresolved. Together, these findings motivate cross-hypothesis discrimination as a design and evaluation target for agentic RCA: accumulating support for a candidate does not by itself establish why it should be preferred over plausible alternatives.

AI USE STATEMENT

We used generative AI tools in two analysis-time roles. The LLM-based diagnostic-state reader (GPT-4o) described in the paper was used to reconstruct expressed diagnostic states from recorded agent traces. We used GPT-5.6-Sol for LLM-assisted annotation of recorded agent trajectories using predefined, design-specific codebooks. The model assigned criterion- or failure-pattern labels with source-linked rationales and excerpts; all retained annotations were reviewed by the authors. We also used generative AI tools for manuscript polishing, including wording, organization, clarity, and table presentation. All AI-assisted outputs incorporated into the reported analyses or manuscript were reviewed by the authors using the validation and source-checking procedures described in the paper. The authors made the final decisions regarding the analyses, interpretations, and manuscript content and take responsibility for the final submission.

ETHICS STATEMENT

This work analyzes existing root-cause analysis benchmarks and recorded LLM agent executions and does not involve human subjects or the collection of personal data. The study concerns the evaluation and analysis of diagnostic agents rather than autonomous operational deployment. We are not aware of additional ethical concerns specific to this work.

REPRODUCIBILITY STATEMENT

Appendix A describes the execution settings, analysis populations, measurement definitions, diagnostic-state reader specification, main design components of the alternative agents, trajectory annotation procedures, and statistical procedures used in our analyses. Appendices B–E document the construction and measurement of the incident-side signal conditions, concrete root observation, expressed diagnostic state, and failure paths. Appendix F.1 separately describes the review and annotation procedure used for the failure analysis. We provide the implementation and usage details for the LLM-based diagnostic-state reader, together with the extraction prompt and a worked example at `https://anonymous.4open.science/r/RCA_observation-826B`. These materials document the measurement, annotation, and analysis procedures of the reported results.

REFERENCES

Tingzhu Bi, Xinrui Jiang, Xun Zhang, Pengcheng Su, Congjie He, Jinglin Li, Ping Wang, and Meng Ma. Justdiag!: A diagnostic justification engine for accountable root cause analysis, 2026. URL https://arxiv.org/abs/2606.19407.

Akash Bonagiri, Devang Borkar, Gerard Janno Anderias, Setareh Rafatirad, and Houman Homayoun. Causalflow: Causal attribution and counterfactual repair for llm agent failures, 2026. URL https://arxiv.org/abs/2605.25338.

Yinfang Chen, Huaibing Xie, Minghua Ma, Yu Kang, Xin Gao, Liu Shi, Yunjie Cao, Xuedong Gao, Hao Fan, Ming Wen, Jun Zeng, Supriyo Ghosh, Xuchao Zhang, Chaoyun Zhang, Qingwei Lin, Saravan Rajmohan, Dongmei Zhang, and Tianyin Xu. Automatic root cause analysis via large language models for cloud incidents. In *Proceedings of the Nineteenth European Conference on Computer Systems*, EuroSys '24, pp. 674–688, New York, NY, USA, 2024. Association for Computing Machinery. ISBN 9798400704376. doi: 10.1145/3627703.3629553. URL https://doi.org/10.1145/3627703.3629553.

Shengda Fan, Xuyan Ye, Yupeng Huo, Zhi-Yuan Chen, Yiju Guo, Shenzhi Yang, Wenkai Yang, Shuqi Ye, Jingwen Chen, Haotian Chen, Xin Cong, and Yankai Lin. Agentprocessbench: Diagnosing step-level process quality in tool-using agents, 2026. URL https://arxiv.org/abs/2603.14465.

Aoyang Fang, Yifan Yang, Jin'ao Shang, Qisheng Lu, Junjielung Xu, Rui Wang, Songhan Zhang, Yuzhong Zhang, Boxi Yu, and Pinjia He. Openrca 2.0: From outcome labels to causal process supervision, 2026a. URL https://arxiv.org/abs/2606.27154.

Aoyang Fang, Songhan Zhang, Yifan Yang, Haotong Wu, Junjielong Xu, Xuyang Wang, Rui Wang, Manyi Wang, Qisheng Lu, and Pinjia He. Rethinking the evaluation of microservice rca with a fault propagation-aware benchmark. *Proc. ACM Softw. Eng.*, 3(FSE), June 2026b. doi: 10.1145/3797100. URL https://doi.org/10.1145/3797100.

Yufei Gao, Zhengong Cai, and Bowei Yang. Rcaflow: A workflow-informed hierarchical planning multi-agent system for root cause analysis. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 300–308, 2026.

Albert Gong, Kyuseong Choi, Abhineet Agarwal, Jason Schechner, Ryan Huang, Raj Agrawal, Anish Agarwal, and Raaz Dwivedi. Orca-bench: How ready are language model agents for oncall?, 2026. URL https://arxiv.org/abs/2607.28545.

Athira Gopal and Ashwanth Krishnan. How far can root cause analysis go on real-world telemetry data?, 2026. URL https://arxiv.org/abs/2607.13548.

Saurabh Jha, Rohan Arora, Bhavya, Noah Zheutlin, Paulina Toro Isaza, Laura Shwartz, Yu Deng, Daby Sow, Ruchi Mahindru, and Ruchir Puri. Think locally, explain globally: Graph-guided llm investigations via local reasoning and belief propagation, 2026. URL https://arxiv.org/abs/2601.17915.

Taeyoon Kim, Woohyeok Park, Hoyeong Yun, and Kyungyong Lee. Why do ai agents systematically fail at cloud root cause analysis? *arXiv preprint arXiv:2602.09937*, 2026.

Mingjie Li, Zeyan Li, Kanglin Yin, Xiaohui Nie, Wenchi Zhang, Kaixin Sui, and Dan Pei. Causal inference-based root cause analysis for online service systems with intervention recognition. In *Proceedings of the 28th ACM SIGKDD conference on knowledge discovery and data mining*, pp. 3230–3240, 2022.

Jingkun Luo and Da-Tian Peng. Success is not self-explanatory: Auditing success provenance in agent evaluation, 2026. URL https://arxiv.org/abs/2607.24054.

Changhua Pei, Zexin Wang, Fengrui Liu, Zeyan Li, Yang Liu, Xiao He, Rong Kang, Tieying Zhang, Jianjun Chen, Jianhui Li, et al. Flow-of-action: Sop enhanced llm-based multi-agent system for root cause analysis. In *Companion Proceedings of the ACM on Web Conference 2025*, pp. 422–431, 2025.

Luan Pham, Huong Ha, and Hongyu Zhang. Baro: Robust root cause analysis for microservices via multivariate bayesian online change point detection. volume 1, pp. 2214–2237, 2024.

Luan Pham, Hongyu Zhang, Huong Ha, Flora Salim, and Xiuzhen Zhang. Rcaeval: A benchmark for root cause analysis of microservice systems with telemetry data. In *Companion Proceedings of the ACM on Web Conference 2025*, pp. 777–780, 2025.

Evelien Riddell, James Riddell, Gengyi Sun, Michał Antkiewicz, and Krzysztof Czarnecki. Stalled, biased, and confused: Uncovering reasoning failures in llms for cloud-based root cause analysis. In *Proceedings of the 2026 IEEE/ACM Third International Conference on AI Foundation Models and Software Engineering*, FORGE '26, pp. 172–183, New York, NY, USA, 2026. Association for Computing Machinery. ISBN 9798400724770. doi: 10.1145/3793655.3793732. URL https://doi.org/10.1145/3793655.3793732.

Brell Sanwouo, Clément Quinton, and Paul Temple. Dynamic agent generation for self-adaptive root cause analysis. In *SEAMS'26-21st International Conference on Software Engineering for Adaptive and Self-Managing Systems*, 2026.

Junle Wang, Xingchuang Liao, and Wenjun Wu. Star: A stage-attributed triage and repair framework for rca agents in microservices, 2026a. URL https://arxiv.org/abs/2605.15581.

Ping Wang, Jingmin Xu, Meng Ma, Weilan Lin, Disheng Pan, Yuan Wang, and Pengfei Chen. Cloudranger: Root cause identification for cloud native systems. In *2018 18th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (CCGRID)*, pp. 492–502. IEEE, 2018.

Yilun Wang, Guangba Yu, Haiyu Huang, Zirui Wang, Yujie Huang, Pengfei Chen, and Michael R. Lyu. Cloud-opsbench: A reproducible benchmark for agentic root cause analysis in cloud systems, 2026b. URL https://arxiv.org/abs/2603.00468.

Zefan Wang, Zichuan Liu, Yingying Zhang, Aoxiao Zhong, Jihong Wang, Fengbin Yin, Lunting Fan, Lingfei Wu, and Qingsong Wen. Rcagent: Cloud root cause analysis by autonomous agents with tool-augmented large language models. In *Proceedings of the 33rd ACM international conference on information and knowledge management*, pp. 4966–4974, 2024.

Amaury Wei and Olga Fink. Agentic root cause analysis through evidence-grounded reasoning, 2026. URL https://arxiv.org/abs/2607.22385.

Zhe Xie, Shenglin Zhang, Yitong Geng, Yao Zhang, Minghua Ma, Xiaohui Nie, Zhenhe Yao, Longlong Xu, Yongqian Sun, Wentao Li, and Dan Pei. Microservice root cause analysis with limited observability through intervention recognition in the latent space. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, KDD '24, pp. 6049–6060, New York, NY, USA, 2024. Association for Computing Machinery. ISBN 9798400704901. doi: 10.1145/3637528.3671530. URL https://doi.org/10.1145/3637528.3671530.

Junjielong Xu, Qinan Zhang, Zhiqing Zhong, Shilin He, Chaoyun Zhang, Qingwei Lin, Dan Pei, Pinjia He, Dongmei Zhang, and Qi Zhang. Openrca: Can large language models locate the root cause of software failures? In *The thirteenth international conference on learning representations*, 2025.

Guangba Yu, Pengfei Chen, Hongyang Chen, Zijie Guan, Zicheng Huang, Linxiao Jing, Tianjun Weng, Xinmeng Sun, and Xiaoyun Li. Microrank: End-to-end latency issue localization with extended spectrum analysis in microservice environments. In *Proceedings of the Web Conference 2021*, pp. 3087–3098, 2021.

Wei Zhang, Hongcheng Guo, Jian Yang, Zhoujin Tian, Yi Zhang, Yan Chaoran, Zhoujun Li, Tongliang Li, Xu Shi, Liangfan Zheng, et al. mabc: multi-agent blockchain-inspired collaboration for root cause analysis in micro-services architecture. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pp. 4017–4033, 2024.

A Study Setup and Statistical Methods

A.1 OpenRCA Executions and Analysis Populations

OpenRCA comprises 335 queries across Bank, Telecom, and two Market/cloudbed systems, spanning 298 dataset–window groups. Each query requests the root component, failure reason, occurrence time, or a combination of these fields. A Controller directs an Executor to inspect metrics, logs, and traces, reads the returned summary and execution output, and chooses whether to continue the investigation or submit a final diagnosis. Benchmark root labels are not provided to the agent. One agent execution per query for each of GPT-4o, Claude Sonnet 4.6, and GPT-5.5 provides the benchmark outcomes and the analysis of concrete root observations. For GPT-4o and GPT-5.5, two more repeated executions per query provide the post-observation diagnostic progression, failure-path, and repeated-execution analyses. These executions share query identities but have distinct investigation histories.

Table 3 summarizes the populations for the incident-side signal analyses and the native-agent analyses. Incident-side signal conditions require metric support; the eight unsupported queries receive no fallback values. Concrete-root-observation and diagnostic-progression analyses additionally require a single labeled root, and progression requires an eligible expressed diagnostic state that consumes the first concrete root observation. Intermediate expressed diagnostic states can be analyzed across task types, whereas final root selection is evaluated only for queries requesting a component, checking all parsed component slots in the final diagnosis. This outcome differs from benchmark *Correct*, which requires all requested fields to be correct, and *Partial*, which includes any positive credit.

Table 3: **Analysis populations and outcome-specific eligibility.** Queries are unique across models and executions; histories count individual agent executions. Indented rows denote subsets. Rows may overlap and should not be summed. Parentheses give Low/High rival-prominence history counts. Here, observation refers to the first concrete root observation.

		Investigation histories	
Analysis / eligibility	Unique queries	GPT-4o	GPT-5.5
Incident-side signal conditions (RQ1; Appendix B)	327	—	—
Concrete root observation (Appendix C): assessment population of supported single-root queries, one execution each	253	253	253
Three-execution analyses			
Final root selection (RQ2): supported component-answer queries; concrete root observation not required	178	$178 \times 3 = 534$	$178 \times 3 = 534$
Post-observation diagnostic progression, all task types	**229**	**127**	**446**
Candidate inclusion (Appendix D): no root candidate in any valid expressed diagnostic state before observation	191	109	274
Leading diagnosis (Appendix D): root never the leading diagnosis before observation; candidate inclusion after observation	187	64	281
Root rank at first post-observation candidate inclusion (Appendix E): Low/High subset eligible for the leading-diagnosis analysis	131	39 (21/18)	192 (127/65)
Initially non-leading root inclusion (Appendix E): subsequent diagnostic progression and competitor strength	89	23 (11/12)	106 (61/45)
Last-state presence (Appendix D): candidate inclusion after observation	209	71	380
Final root selection (Appendix D): queries requesting a component in the final diagnosis	125	84	253
Low/High component-answer subset: diagnostic progression (Appendix D) and paths to final root omission (Appendix E)	78	49 (24/25)	159 (107/52)
Failure review (Appendix F): root omitted from the final diagnosis	56	39 (17/22)	71 (36/35)
Root-directed follow-up (Appendix F): leading-diagnosis eligibility; valid follow-up measurement	45	17 (8/9)	54 (27/27)

The concrete-root-observation assessment covers 253 supported single-root queries, regardless of whether a concrete root observation occurs. The supported component-answer population does not require a concrete root observation and includes both single- and multi-root queries; its 178 queries contribute 534 investigation histories per model across three executions. For queries with multiple labeled root components, we use the component in the query's corresponding benchmark record as the focal root for signal-condition measurement and count final root selection when that component appears in any parsed component slot of the final diagnosis. The four italicized outcome rows give overlapping regression eligibility sets. Descriptive diagnostic progression uses the same Low/High component-answer histories for all four outcomes. The analyses of initial root rank and subsequent diagnostic progression in Appendix E include all task types without conditioning on final error; the native-agent failure review is restricted to final root omissions. All prominence groups are included unless Low/High is specified.

A.2 Trace Measurements and Diagnostic-State Reconstruction

Our analysis separates the telemetry available for investigation, the returned content that actually enters the Controller-visible history, the expressed diagnostic state subsequently recorded in the Controller trace, and the final diagnosis. These boundaries need not coincide: a concrete root observation can occur without the root-cause component entering the candidate set, and candidate inclusion does not imply that the root becomes the leading diagnosis or is selected in the final diagnosis. Incident-side signal conditions are computed independently of agent traces, and root-cause labels are used only for retrospective evaluation. The operational definitions underlying these boundaries are provided with the corresponding analyses: construction of the incident-side signal conditions in Appendix B, the concrete-root-observation rule in Appendix C, and diagnostic-progression outcomes, outcome-specific eligibility, and state-reader repeatability in Appendix D. Diagnostic-path definitions and the native-agent failure-review procedure appear in Appendices E and F, respectively. To reconstruct the expressed diagnostic state without altering the original investigation, we apply a separate GPT-4o state reader after each recorded Controller response. The reader receives only the conversation available to that response, the response itself, the allowed component vocabulary for the task, and the observation identifiers accumulated up to that point. The same reader model is used for GPT-4o and GPT-5.5 agent histories. No later turns or retrospective root-cause labels are provided, and the resulting snapshot is stored separately rather than fed back to the investigating agent. The reconstructed state therefore represents what is expressed in the visible trace at that point, not the agent's latent belief or an independent judgment of the correct diagnosis.

Prompt for State Reader The exact extraction request is reproduced below, with line wrapping and JSON indentation for readability. At runtime, it was followed by `Allowed candidates:` and `Allowed observation IDs:`, each with its corresponding JSON array.

```
Return ONLY one JSON object that audits the diagnostic state
implied by the conversation and the planner response. Do not
solve again and do not request new evidence. Schema:
{
  "candidate_state": [
    {
      "component": "candidate or null",
      "rank": 1,
      "support": "high|medium|low|unknown",
      "status": "supported|possible|weakened|ruled_out|unknown",
      "supporting_observation_ids": [],
      "refuting_observation_ids": []
    }
  ],
  "leader": "candidate or null",
  "next_target": "candidate or null",
  "stop_intent": false,
  "state_basis": "explicit|inferred_from_planner_text"
}.
Candidate names must come from the supplied vocabulary.
Observation IDs must come from the supplied list.
Empty/unknown state is allowed.
```

A.3 Statistical Analysis

We use two complementary analyses to relate incident-side signal conditions to diagnostic outcomes. Adjusted logistic regressions estimate associations over the continuous condition measures, while fixed low/high contrasts provide descriptive comparisons in the original analysis populations. Both analyses preserve repeated agent executions and account for shared incident windows; neither is interpreted as estimating a causal effect.

For the regression analyses, each continuous condition is standardized within dataset using the mean and population standard deviation of supported unique query rows, with zero assigned when the scale is zero. The same reference values are shared across models and executions. For outcome k, condition j, and eligible history set $\mathcal{R}_k$, the association specification is

$$\operatorname{logit} \Pr(Y_{ir}^{(k)} = 1 \mid (i,r) \in \mathcal{R}_k) = \alpha_k + \beta_{jk} E_{ij}^{\mathrm{std}} + \gamma_{k,d(i)} + \eta_{k,\mathrm{task}(i)} + \xi_{k,r}. \tag{3}$$

Each binomial generalized linear model includes one incident-side signal condition together with fixed effects for dataset, requested task, and execution. The model-specific estimates reported in the main text fit this specification separately to GPT-4o and GPT-5.5. Outcome-specific analysis sets are given in Appendix D.1. Model-specific estimates describe within-model associations and are not tests of differences between models.

A dataset together with its parsed query window defines the clustering unit, so query rows and repeated executions sharing the same underlying window contribute to a common cluster. Sandwich covariance estimates provide pointwise 95% confidence intervals. Odds ratios correspond to a one-standard-deviation increase in the condition within dataset, and reported marginal probability differences average the associated change in predicted probability over eligible histories with the adjustment variables held fixed. Benjamini–Hochberg correction is applied across the 24 exploratory model-specific tests. Unsupported or failed fits remain in this correction family with $p = 1$. Support requirements and complete estimates are reported in Appendix D.

The low/high comparisons are descriptive and are kept separate from the continuous-condition regressions. Low and high groups use fixed dataset-specific 25th- and 75th-percentile cutoffs computed from supported query rows and shared across models and executions. Their 95% percentile bootstrap intervals are obtained by resampling complete dataset–window clusters with replacement within model and dataset while retaining all histories in each sampled cluster. Execution comparisons and fixed-history reader rereads are reported in Appendix D.3.

A.4 Analysis of Alternative Agent Designs

Local-to-Global Diagnosis (LGD). LGD organizes diagnosis around component-specific agents. Each agent examines evidence associated with its component and produces a self-assessment, including whether the observed behavior suggests an originating fault or a reaction to another component. Selected pairs then exchange evidence through local debates, assessing the causal relationship between their components. These local judgments are consolidated for a Global Reviewer alongside component self-assessments and evidence summaries. The Reviewer compares candidates across the incident, considering their own-resource behavior and local debate outcomes to distinguish a potential fault origin from propagated effects. In the audited implementation, the reviewer input combines debate candidates with confirmed self-originating claims and is capped at 12 candidates. The Global Reviewer compares the candidates presented to it using their self-assessments, pairwise debate outcomes, and evidence summaries.

Hypothesis-Guided Investigation (HGI). HGI maintains numerical beliefs over candidate component–reason hypotheses, initialized by a Labeled Latent Dirichlet Allocation model using telemetry-derived token counts that encode anomaly categories, component identities, and screening clusters. The initializer is estimated separately within each dataset from labeled examples, excluding the evaluated query through leave-one-out estimation. The Critic examines the current leading hypotheses and investigation history to determine whether to commit, continue investigating, or request a revised proposal. When further investigation is required, the Proposer selects a target and a probe, such as a category-specific KPI check, a component comparison, or a timing inquiry. The probe retrieves observations, and applicable results are converted into numerical updates to the hypothesis

Table 4: **Operational criteria used to analyze the 22 Telecom incidents.** The reference component is the root-cause component.

Criterion	Definition and identification rule
LGD: support in a local debate	At least one pairwise debate favors the reference component over its opponent, with no recorded tool error in that debate. This local decision is distinguished from the Global Reviewer's choice and the final pipeline prediction.
LGD: exclusion from global review	The reference component receives a confirmed or almost-certain self-assessment as a fault origin, but falls outside the first 12 entries of the reconstructed candidate pool. Those 12 entries must match the recorded Reviewer input, confirming its exclusion from global comparison.
LGD: interpretation as a reaction	The summary presented to the Reviewer characterizes the reference component as showing inherited or reactive behavior. The Reviewer selects another component with an own-resource anomaly, and that selection remains the final component prediction.
HGI: repeated positive updates	The same measurement is used in at least two positive belief updates supporting the finally selected component. Measurements match in component, category, peak magnitude, and peak timestamp; reuse may occur across different tools. Positive update direction must be verified from the recorded actions, observations, and update rules.
HGI: automatic early commitment	The trace records termination through the automatic rule: the leading hypothesis has probability at least 0.85 and a margin over the second-ranked hypothesis of at least 0.50. This path bypasses the LLM Critic.
SDC: evidence return without numerical leadership	A Coverage action targeting only the reference component returns at least one evidence item, but that component never appears as the highest-ranked hypothesis in the recorded rankings. Evidence return alone does not establish that the returned evidence was sufficient to identify the root.
SDC: focus without direct investigation	The reference component appears in the initial investigation focus but is absent from the target lists of all subsequently executed investigation actions. Focus membership alone therefore does not count as direct investigation.

beliefs. The updated beliefs inform the next Critic assessment and subsequent probe selection. The loop returns the current leading hypothesis when commitment occurs or after at most eight investigation rounds. Commitment also includes an automatic path based on the leading hypothesis's probability and its margin over alternatives.

Structured Diagnostic Control (SDC). SDC organizes diagnosis as a repeated cycle of investigation, evidence interpretation, and review. Triage identifies an initial set of components to investigate. A control policy then selects investigation actions using the current hypothesis scores, accumulated evidence, and investigation coverage. Coverage operators examine insufficiently investigated components; Temporal operators examine time-local anomaly patterns; and DeepProbe operators conduct targeted investigation of leading hypotheses. When invoked, the Analyst interprets the returned evidence, while code incorporates that evidence into the numerical diagnostic belief. Review examines the evolving diagnosis and provides guidance for subsequent investigation. The maintained state connects these stages by retaining candidate beliefs, evidence, and previously investigated targets. Investigation repeats until the implemented stopping conditions are met or a budget of 25 investigation actions is exhausted, after which the current diagnostic state is used to produce the final result.

Reconstruction and Annotation. We retrospectively analyzed executions of Local-to-Global Diagnosis (LGD), Hypothesis-Guided Investigation (HGI), and Structured Diagnostic Control (SDC) on 22 of 23 Telecom single-root component-answer queries; one LGD query lacked a matching query window and was excluded from all methods for execution alignment rather than diagnostic correctness. The three designs are diagnostic probes, not controlled architecture comparisons, and differ in initialization and investigation budget. For each history, we traced candidate comparison, root-related returns, priority updates, reconsideration, and final selection through investigation targets, observations, judgments, state changes, and termination. An exploratory, design-specific codebook combined deterministic extraction with semantic annotation by one LLM assistant, with retained annotations reviewed by the authors; criteria were coded as confirmed, not meeting the definition (including non-occurrence), or undetermined when required records were insufficient. Table 4 defines the key criteria, which differ from the state-reader definitions of root observation and expressed leadership used elsewhere.

A.5 MODEL USAGE.

Table 5 reports recorded usage for RCA executions, diagnostic-state reconstruction, and retained analysis probes. Both state-reader rows use GPT-4o on the corresponding baseline histories. Candidate-disappearance probes count the shared control once per checkpoint. Telemetry processing, benchmark scoring, and statistical analysis require no additional LLM calls.

Table 5: **Recorded model usage.** Counts include Controller and Executor calls for one full 335-query run per model, corresponding state-reader calls, and alternative design probes. Repeated full runs are omitted. M denotes 10^6 tokens; values are rounded independently.

Experiment	Reported scope	Input (M)	Output (M)	Total (M)
RCA agent (GPT-4o)	335 queries	93.540	1.643	95.182
RCA agent (GPT-5.5)	335 queries	178.496	9.497	187.994
RCA agent (Claude Sonnet 4.6)	335 queries	358.237	34.198	392.435
State reader (GPT-4o histories)	335 queries	43.214	0.489	43.703
State reader (GPT-5.5 histories)	335 queries	72.828	0.646	73.474
LGD (GPT-4o)	22 queries	3.035	1.193	4.228
HGI (GPT-4o)	22 queries	0.507	0.043	0.550
SDC (GPT-4o)	22 queries	1.995	0.186	2.180

B INCIDENT TELEMETRY MEASURES

B.1 INITIAL-SNAPSHOT MOTIVATION

Figure 2 summarizes 148 Market incidents using the first five minutes after the annotated onset and the preceding 30-minute baseline. Node and service metrics, trace features, and log features give each displayed entity its maximum absolute anomaly score. Entities are ranked by this score; the first normalized name matching the labeled root-cause component supplies its rank, with an unmatched root-cause component placed below all scored entities. The root-cause component is Top-1 in 30.4%, Top-5 in 42.6%, and has median rank 12. This snapshot provides illustrative motivation; the incident-side signal conditions below use metric telemetry over a fixed component pool and supply the main RQ1/RQ2 analyses.

B.2 CONSTRUCTION OF THE INCIDENT-SIDE SIGNAL CONDITIONS

The three incident-side signal conditions characterize different ways the root-cause component appears relative to non-root activity in benchmark telemetry: when the root reaches its strongest observed deviation relative to eligible non-root components, how frequently non-root components exhibit stronger anomalies than the root, and how broadly anomalous activity is distributed around the incident. All three are computed from benchmark telemetry independently of agent trajectories and final outcomes. Raw entities are mapped to the fixed component registry using source-specific name normalization, with unmatched identifiers excluded and pod and base-service levels kept distinct.

For relative peak timing and rival prominence, each source–metric series is normalized against its recent pre-query baseline, and each component is represented by the largest absolute standardized deviation observed during the query window. For query window W_i,

$$z_{cf}(t) = \frac{x_{cf}(t) - \mu_{cf}}{\sigma_{cf}}, \qquad M_c = \max_{f,t \in W_i} |z_{cf}(t)|. \tag{4}$$

Relative peak timing. This condition captures whether the root reaches its strongest observed deviation later than the eligible non-root components in the same incident. Let $\mathcal{C}_i^-$ denote the eligible non-root comparison pool and τ_c the selected peak time for component c. Relative peak timing is

$$D_i = \tau - \text{median}_{c \in \mathcal{C}_i^-} \tau_c. \tag{5}$$

A positive D_i therefore means that the root's selected metric peak occurs later than the median eligible non-root peak. The measure concerns the relative timing of observed metric peaks rather than fault onset.

Rival prominence. This condition captures the share of eligible non-root components whose peak anomaly magnitude exceeds the root's. Using the same eligible comparison pool, rival prominence is

$$R_i = \frac{\sum_{c \in \mathcal{C}_i^-} \mathbf{1}\{M_c > M\}}{|\mathcal{C}_i^-|}. \tag{6}$$

Larger R_i therefore means that a larger fraction of eligible non-root components exhibit stronger anomalies than the root-cause component. The measure is defined independently of which components the agent later mentions, investigates, or selects, and equal magnitudes are not counted as stronger.

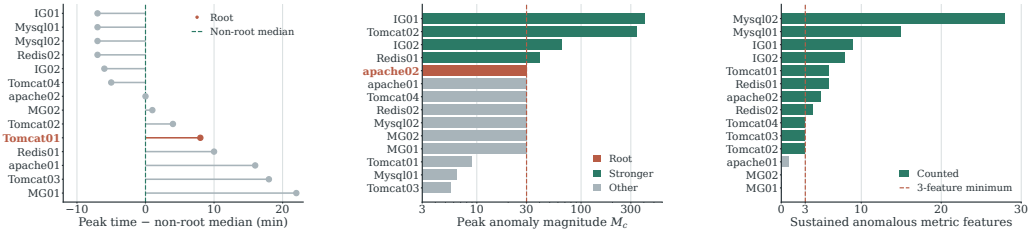
Concurrent anomaly breadth. This condition captures how widely anomalous activity is distributed around the labeled event. Unlike relative peak timing and rival prominence, which compare component-level peak behavior with the root, concurrent anomaly breadth summarizes sustained activity across multiple metrics and components near the event. Metric series are normalized against a robust pre-event baseline and summarized over a short event neighborhood; a component contributes to breadth only when anomalous activity is supported across multiple metric features. Let $\widetilde{z}_{cf}(t)$ denote the robustly standardized series, $\mathcal{N}_i$ the event neighborhood, θ_B the feature-level anomaly threshold, and m_B the required number of qualifying features:

$$B_i = \sum_c \mathbf{1}\left\{ \sum_f \mathbf{1}\left\{ \text{median}_{t \in \mathcal{N}_i} |\widetilde{z}_{cf}(t)| > \theta_B \right\} \geq m_B \right\}. \tag{7}$$

Larger B_i therefore indicates that anomalous activity spans more components, regardless of whether those components individually exceed the root in peak magnitude.

Table 6: **Summary of the three incident-side signal conditions.** Counts use the 327-query population. Descriptive thresholds summarize prevalence; association analyses use the continuous measures.

Condition	Continuous measure	Descriptive threshold (queries)
Relative peak timing	D_i (minutes)	$D_i > 0$ (170/327)
Rival prominence	R_i (fraction)	$R_i > 0$ (213/327)
Concurrent anomaly breadth	B_i (components)	$B_i \geq 8$ (135/327)



(a) Relative peak timing (b) Rival prominence (c) Concurrent anomaly breadth

Figure 6: **Examples of the three OpenRCA signal conditions.** (a) The root peaks 15 minutes after query start, versus a non-root median of 7 minutes ($D_i = 8$ minutes). (b) 4 of 13 non-root components exceed the root's magnitude ($R_i = 30.8\%$). (c) 11 components meet the three-feature criterion, exceeding the descriptive threshold of eight ($B_i = 11$).

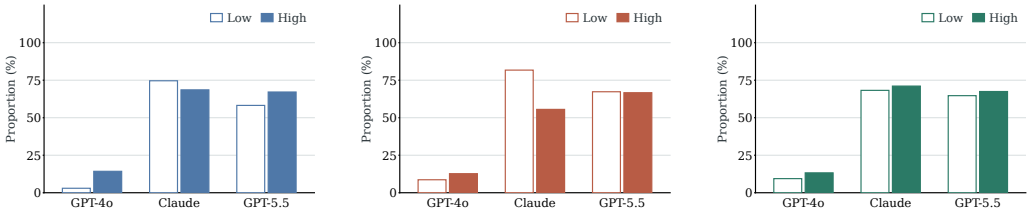
Implementation details. Peak scores use the 30 minutes immediately preceding the query as baseline, and a source–metric feature requires at least 20 baseline samples and a nonempty observation window. Components with $M_c \geq 3$ form the comparison pool for relative peak timing and rival

prominence; the root is retained below this threshold when it has an eligible series, and unsupported queries receive no imputed values (Appendix A.1). For each component, τ_c is the earliest selected peak among feature series whose maximum absolute standardized deviation reaches 3; a sub-threshold root uses the peak of its maximum-deviation series. Concurrent anomaly breadth uses the baseline from 60 to 15 minutes before the labeled event and its ± 5-minute neighborhood $\mathcal{N}_i$. Its scale is the first positive value among $1.4826\,\mathrm{MAD}$, $\mathrm{IQR}/1.349$, and the population standard deviation, with $\theta_B = 4$ and $m_B = 3$. For either construction, a zero-scale baseline assigns zero to unchanged values and signed magnitude 30 to changed values. Table 6 summarizes the resulting continuous measures and the separate thresholds used to describe their prevalence in RQ1. Figure 6 illustrates construction within individual incidents.

C Concrete Root Observation

A concrete root observation requires the root-cause component's identifier and a telemetry datum in the same sentence or table row of a Controller-visible return. Component-boundary matching requires a numerical metric payload, a quoted or timestamped log event, or a numerical trace datum with a relation or duration cue. Bare names in code, schemas, component/KPI inventories, or requests do not qualify. Identifier digits, query dates and times, percentile labels, explicitly labeled epochs, and row-count-only outputs do not establish a payload; negative no-data statements are excluded. Numerical summaries and baseline thresholds qualify when they display a telemetry value from the root-cause component, which need not be anomalous. The earliest qualifying return is the first concrete root observation; alignment with the expressed diagnostic state that consumes this return is specified in Appendix D. This endpoint is measured directly from returned content, independently of the reconstruction of expressed diagnostic state. Detection rules and regression examples were fixed before fitting the post-observation models. The operational same-unit rule records displayed telemetry; it does not establish causal support for the root-cause component, discrimination among competing root-cause hypotheses, or diagnostic sufficiency.

Figure 7 compares concrete root observation frequencies across low and high tails of the three incident-side signal conditions, using the concrete-root-observation population in Appendix A.1. Rival prominence shows no consistent relationship with whether a concrete root observation occurs: the high group has a higher observation rate for GPT-4o, a lower rate for Claude, and a nearly unchanged rate for GPT-5.5 (Figure 7(b)). These contrasts across agent executions do not support a uniform explanation in which poorer downstream diagnostic progression is associated with lower rates of concrete root observation. Appendix D examines the distinct question of whether, after a concrete root observation, the root enters the candidate set, becomes the leading diagnosis, is present in the last valid pre-final state, and is selected in the final diagnosis.



(a) Peak timing (b) Rival prominence (c) Anomaly breadth

Figure 7: **Concrete root observation across agent executions.** Open/filled bars denote dataset-specific low/high condition tails for GPT-4o, Claude Sonnet 4.6, and GPT-5.5. The endpoint is measured directly from Controller-visible returned content, independently of the reconstruction of expressed diagnostic state.

D Post-Observation Diagnostic Progression

D.1 Diagnostic-State Reconstruction and Analysis Boundaries

The first concrete root observation (Appendix C) is aligned to the exact valid, noncompleted Controller state that consumes that return; a later return is not substituted when this state is ineligible. The reader

in Appendix A.2 reconstructs the expressed diagnostic state in the visible trace rather than the agent's internal belief. Structural checks enforce valid component names, ranks, field values, observation references, and stopping status, without establishing semantic correctness.

Candidate inclusion records root membership in any valid noncompleted state at or after the first concrete root observation; *leading diagnosis* records whether the root becomes the leading diagnosis in any such state. *Last-state presence* records root membership in the last valid noncompleted candidate state, and *final root selection* records the root in any parsed component slot of the final diagnosis for component tasks. These outcomes need not form a monotonic sequence: the root can disappear and re-enter, and last-state presence does not imply final root selection. Adjusted post-observation associations use the outcome-specific eligibility sets summarized in Table 3 in Appendix A.1.

D.2 Post-Observation Diagnostic Progression

Relative peak timing and rival-prominence estimates are below one wherever estimable, whereas concurrent anomaly breadth has mixed directions (Table 7). The rival-prominence association with the GPT-5.5 leading diagnosis outcome is the only one that passes correction across the 24 model-specific tests. GPT-4o estimates have weaker statistical support; its final root selection fits fail the minimum-event requirement.

Table 7: **Standardized adjusted post-observation associations.** Estimates are fitted separately for each model and correspond to Figure 4(b). Odds ratios and pointwise 95% intervals follow Appendix A.3; q adjusts all 24 model-specific tests together. Unsupported fits retain $q = 1$.

		GPT-4o		GPT-5.5	
Condition	Outcome	OR [95% CI]	q	OR [95% CI]	q
Relative peak timing	Candidate inclusion	0.90 [0.58, 1.42]	0.833	0.82 [0.58, 1.18]	0.531
	Leading diagnosis	0.29 [0.11, 0.78]	0.090	0.77 [0.57, 1.04]	0.254
	Last-state presence	0.56 [0.27, 1.17]	0.329	0.74 [0.49, 1.12]	0.339
	Final root selection	Unsupported	1.000	0.67 [0.47, 0.98]	0.152
Rival prominence	Candidate inclusion	0.95 [0.63, 1.43]	0.920	0.69 [0.51, 0.93]	0.090
	Leading diagnosis	0.49 [0.25, 0.99]	0.160	0.59 [0.43, 0.81]	0.024
	Last-state presence	0.59 [0.29, 1.21]	0.339	0.67 [0.47, 0.97]	0.152
	Final root selection	Unsupported	1.000	0.51 [0.31, 0.82]	0.067
Concurrent anomaly breadth	Candidate inclusion	0.82 [0.50, 1.34]	0.582	0.95 [0.64, 1.41]	0.920
	Leading diagnosis	1.59 [0.67, 3.80]	0.531	1.13 [0.86, 1.48]	0.580
	Last-state presence	1.46 [0.68, 3.10]	0.531	1.25 [0.80, 1.94]	0.531
	Final root selection	Unsupported	1.000	1.14 [0.83, 1.56]	0.582

Table 8: **Diagnostic progression under low and high rival prominence.** Rates are percentages; high–low differences and 95% intervals are in percentage points, using 5,000 dataset-window cluster bootstrap resamples within model and dataset. The low/high groups contain 24/25 eligible investigation histories for GPT-4o and 107/52 for GPT-5.5.

	GPT-4o			GPT-5.5		
Outcome	Low $n = 24$	High $n = 25$	High–low [95% CI]	Low $n = 107$	High $n = 52$	High–low [95% CI]
Candidate inclusion	62.5	52.0	-10.5 [-38.9, 20.5]	93.5	88.5	-5.0 [-14.4, 4.2]
Leading diagnosis	45.8	24.0	-21.8 [-46.0, 6.7]	68.2	40.4	-27.8 [-46.8, -8.1]
Last-state presence	54.2	32.0	-22.2 [-46.3, 5.2]	86.0	67.3	-18.7 [-36.8, -0.3]
Final root selection	29.2	12.0	-17.2 [-40.0, 5.3]	66.4	32.7	-33.7 [-53.8, -12.5]

The common-population contrasts in Table 8 locate substantial gaps in whether the root becomes the leading diagnosis and in final root selection, especially for GPT-5.5. For these two outcomes, the intervals exclude zero for GPT-5.5 and include zero for GPT-4o. These aggregate outcomes do not define a monotonic sequence for individual histories; Appendix E decomposes the detailed paths to final root omission.

D.3 Execution and Reader Comparisons

Diagnostic progression across repeated executions. We summarize post-observation progression separately for three repeated agent executions (Table 9). Within each execution, we apply the same component-answer eligibility rules and fixed dataset-specific rival-prominence cutoffs. Each low/high group uses a common set of eligible histories for all four outcomes. Since whether a concrete root observation occurs and whether valid post-observation expressed diagnostic states are available depend on the realized investigation, the eligible histories can differ across executions. For GPT-5.5, candidate inclusion remains frequent in both prominence groups across all three executions. The root becomes the leading diagnosis, remains present in the last valid pre-final state, and is selected in the final diagnosis less frequently under high prominence in every execution, although the size of these differences varies. The low–high ordering for candidate inclusion changes across executions. Thus, the repeated results consistently show a prominence-related difference in whether the root becomes the leading diagnosis and in final root selection, while candidate inclusion shows a less consistent contrast. GPT-4o exhibits greater execution-to-execution variation. The high-prominence group has lower rates across all four outcomes in the baseline and first repeat, but equal or higher rates in the second repeat. Its available groups contain only 5–12 histories, making the observed proportions sensitive to individual histories. These results support recurrence of the later-stage GPT-5.5 pattern, while indicating that the GPT-4o contrasts are less consistent across executions.

Table 9: **Diagnostic progression across agent executions.** Entries are positive/eligible histories (n/N) by rival-prominence group.

Model	Execution	Prominence	Candidate inclusion	Leading diagnosis	Last-state presence	Final root selection
GPT-4o	Baseline	Low	3/7	3/7	3/7	1/7
		High	1/5	0/5	0/5	0/5
	Repeat 1	Low	6/8	5/8	6/8	4/8
		High	3/8	2/8	1/8	0/8
	Repeat 2	Low	6/9	3/9	4/9	2/9
		High	9/12	4/12	7/12	3/12
GPT-5.5	Baseline	Low	31/35	21/35	27/35	21/35
		High	16/18	6/18	13/18	4/18
	Repeat 1	Low	33/35	25/35	33/35	22/35
		High	16/16	7/16	12/16	6/16
	Repeat 2	Low	36/37	27/37	32/37	28/37
		High	14/18	8/18	10/18	7/18

Table 10: **Diagnostic-state reader repeatability.** Pairwise agreement in root-state summaries, diagnostic-path categories, and competing non-root candidates across the original reading and two rereads of identical investigation histories. Entries show agreeing pairs/total pairs (percentage).

Measurement	GPT-4o	GPT-5.5
Candidate inclusion	77/81 (95.1%)	421/435 (96.8%)
Leading diagnosis	75/81 (92.6%)	417/435 (95.9%)
Last-state presence	79/81 (97.5%)	421/435 (96.8%)
Coarse diagnostic path	69/81 (85.2%)	395/435 (90.8%)
Competing non-root candidates (exact match)	45/81 (55.6%)	369/435 (84.8%)

Diagnostic-state reader repeatability. We assessed reader repeatability using the original reading and two archived rereads of identical baseline investigation histories. The same reader was used with fixed observations, Controller responses, and contexts. We included all 172 histories with complete valid readings: 27 from GPT-4o executions and 145 from GPT-5.5 executions. Each history contributes three pairwise comparisons among the three readings, yielding 81 and 435 pairs, respectively. Table 10 reports agreement on history-level summaries. The root-state summaries record whether the root-cause component ever enters the candidate set, ever becomes the leading diagnosis, and is present in the last valid pre-final state. The coarse diagnostic path combines these indicators. Agreement is high

for the three root-state indicators, ranging from 92.6–97.5% for GPT-4o histories and 95.9–96.8% for GPT-5.5 histories, supporting the repeatability of the core diagnostic-progression measurements. Exact-match agreement for competing non-root candidates applies a stricter criterion to the set of non-root components identified as dynamic competitors across the analyzed post-observation expressed diagnostic states. Any added or omitted component counts as disagreement, including cases where the two sets partially overlap. The lower exact-set agreement indicates greater variability in detailed identification of competing non-root candidates, particularly for GPT-4o histories.

E Post-Observation Failure Paths

This appendix examines where diagnostic progression breaks down after a concrete root observation. We first decompose final root omissions according to the root's history of candidate inclusion and becoming the leading diagnosis. We then examine progression from a shared starting point: the root is in the candidate set but is not the leading diagnosis. Finally, we describe the investigation activity and competing candidates at that point, providing context for the decision-level review in Appendix F.

E.1 Paths to Final Root Omission

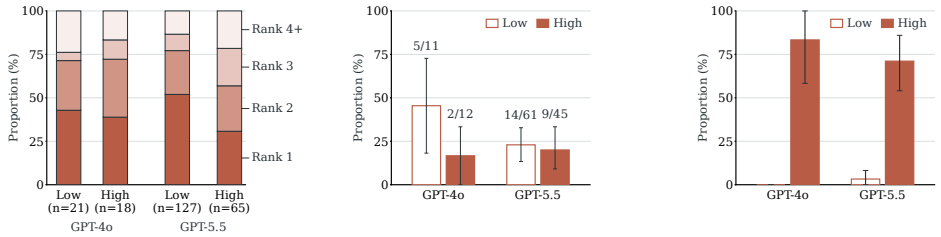
The progression rates show how often the root enters the candidate set, becomes the leading diagnosis, remains present in the last valid pre-final state, and is selected in the final diagnosis. They do not show which combinations of these outcomes characterize final omissions. Figure 4(d) and Table 11 connect these measurements by partitioning omissions into five mutually exclusive paths.

Table 11: **Paths to final root omission by model.** Low/high rates are percentages; high–low differences and marginal 95% intervals are in percentage points. Intervals use 1,000 dataset-window bootstrap replicates. Differences are calculated before rounding.

	GPT-4o			GPT-5.5		
Path	Low $n = 24$	High $n = 25$	High–low [95% CI]	Low $n = 107$	High $n = 52$	High–low [95% CI]
Never included	37.5	48.0	+10.5 [-19.3, 39.6]	6.5	11.5	+5.0 [-4.2, 13.7]
Never led; absent last	4.2	16.0	+11.8 [-5.3, 30.8]	6.5	21.2	+14.6 [0.9, 30.2]
Never led; present last	12.5	12.0	-0.5 [-19.0, 17.3]	15.9	26.9	+11.0 [-3.8, 27.5]
Previously led; absent last	4.2	4.0	-0.2 [-11.1, 11.5]	0.9	0.0	-0.9 [-3.1, 0.0]
Previously led; present last	12.5	8.0	-4.5 [-18.5, 12.1]	3.7	7.7	+4.0 [-3.4, 13.2]
All final omissions	70.8	88.0	+17.2 [-5.0, 39.8]	33.6	67.3	+33.7 [14.2, 55.2]

We use component-answer histories with a concrete root observation in the fixed low/high rival-prominence groups. The paths distinguish roots that never enter the candidate set, roots that enter but never become the leading diagnosis, and roots that become the leading diagnosis at least once. The latter two categories are further divided by last-state presence. Every path rate uses the full eligible low/high group as its denominator, including histories that select the root. Consequently, the five rates sum to the total omission rate, and their high–low differences sum to the overall omission gap.

In both models, most of the observed high–low omission gap is accounted for by paths in which the root never becomes the leading diagnosis. This includes histories where the root enters the candidate set but is absent from the last valid pre-final state; for GPT-5.5, it also includes more histories where the root remains a candidate without ever becoming the leading diagnosis. Candidate presence therefore does not by itself imply that the root is favored: the root can remain under consideration while another component is ultimately selected. The decomposition locates the observed gap mainly in histories where the root never becomes the leading diagnosis, while preserving differences in the specific paths across models. Last-state absence alone does not establish explicit rejection or irreversible disappearance. The decomposition therefore identifies failure to become the leading diagnosis, including after candidate inclusion, as the main contributor to the observed high–low omission gap.



(a) Recorded root rank at candidate inclusion (b) Later diagnostic leadership after non-leading inclusion (c) Current competitor strength

Figure 8: **Initial root state under rival prominence.** (a) Root rank at first post-observation candidate entry. (b) Later leadership among initially non-leading entries. (c) Stronger current competitors among the same histories. Error bars in (b,c) show 95% dataset-window bootstrap intervals.

Table 12: **Paths after non-leading root entry.** Cells are histories n/N within each model and prominence group. The three paths are mutually exclusive; later leadership includes recovery after disappearance. These are diagnostic-state paths, without conditioning on final component error.

Model	Group	Later leadership	Disappeared; never led	Completed; never led
GPT-4o	Low	5/11	1/11	5/11
GPT-4o	High	2/12	6/12	4/12
GPT-5.5	Low	14/61	9/61	38/61
GPT-5.5	High	9/45	14/45	22/45

E.2 Progression after Non-Leading Root Inclusion

The omission paths in RQ2 locate much of the rival-prominence-associated gap in histories where the root never becomes the leading diagnosis. We therefore examine when this disadvantage becomes visible: when the root first enters the candidate set, or later after it has already entered as a non-leading candidate. The analysis includes all requested task types, requires a concrete root observation followed by candidate inclusion, and excludes histories in which the root had already become the leading diagnosis. First inclusion denotes the first post-observation expressed diagnostic state in which the root appears in the candidate set. Low/high groups contain 21/18 GPT-4o and 127/65 GPT-5.5 histories in Figure 8(a); the initially non-leading subsets contain 11/12 and 61/45 histories in Figure 8(b). These populations are not conditioned on final error and therefore differ from RQ2's component-answer population with a concrete root observation.

Initial State of Root Component. Candidate inclusion does not imply that the root is already treated as the most plausible cause. For GPT-5.5, the root enters at rank 1 in 66/127 low-prominence histories (52.0%) but only 20/65 high-prominence histories (30.8%; Figure 8(a)). Conversely, rank 3 or worse increases from 29/127 (22.8%) to 28/65 (43.1%), while rank 2 remains nearly unchanged (25.2% versus 26.2%). Thus, the difference is not only whether the root enters the candidate set, but where it stands when it enters: under greater rival prominence, GPT-5.5 more often represents the root behind several competing root-cause hypotheses from the outset. GPT-4o shows a smaller rank-1 difference (9/21 versus 7/18) without the same clear shift toward lower ranks, so this initial-standing pattern is most evident for GPT-5.5.

Later Recovery. We next restrict attention to histories in which the root enters as a non-leading candidate. For GPT-5.5, the root later becomes the leading diagnosis in 14/61 low-prominence histories (23.0%) and 9/45 high-prominence histories (20.0%; Figure 8(b)). Most initially secondary roots therefore never become the leading diagnosis in either group, and the similar point estimates do not support a prominence-related later-promotion deficit for GPT-5.5. GPT-4o shows a larger descriptive contrast, 5/11 (45.5%) versus 2/12 (16.7%), but the clustered intervals for both models' high–low contrasts include zero. Taken together with the entry-rank results, the clearer GPT-5.5

separation is already visible when the root first enters the candidate set rather than emerging only from poorer later recovery. Thus, a concrete root observation and candidate inclusion can still leave the root diagnostically secondary from the outset. Table 12 retains the full conditional paths, including disappearance and continued candidate presence without the root becoming the leading diagnosis.

Competitor Strength. Separately, we examine whether the component represented ahead of a non-leading root is itself one of the non-root components whose anomaly exceeds the root's in the rival-prominence measurement. The current competitor is such a component in 0/11 low-prominence versus 10/12 high-prominence GPT-4o histories, and in 2/61 versus 32/45 GPT-5.5 histories (Figure 8(c)). The current competitor is the non-root leading diagnosis or, when no leader is recorded, the uniquely highest-ranked non-root candidate. These results show that high rival prominence is not only a property of the surrounding telemetry: in many such histories, a non-root component with a stronger anomaly is also the competing candidate represented ahead of the root. Because rival prominence itself measures the prevalence of stronger non-root components, this comparison is descriptive rather than an independent estimate of their effect. Its role is to connect the incident-side signal condition to the specific competing candidate occupying that position in the expressed diagnostic state.

F Candidate Revision, Evidence Use, and Commitment

F.1 Failure-Review Procedure

Table 13: **Patterns and counts in reviewed failures.** Entries are positive/eligible histories in the low/high rival-prominence groups. Root-directed follow-up uses histories in which the root had not become the leading diagnosis before the first concrete root observation and was eventually included in the candidate set after observation (under the RQ2 eligibility rules); missing measurements are excluded. Other rows use all reviewed failures.

Pattern	What is recorded in the history	GPT-4o		GPT-5.5	
		Low	High	Low	High
Root-directed follow-up	Root named in an investigation instruction or executed code during a valid, noncompleted post-observation root-candidate state before first becoming the leading diagnosis.	2/8	2/9	16/27	17/27
Competing candidate changes	Defined non-root competitor changes between consecutive analyzed post-observation expressed diagnostic states.	7/17	15/22	10/36	10/35
Attribution from relative symptoms, timing, or position	Relative anomaly magnitude/count, timing, or downstream position favors an alternative or discounts the root without a recorded discriminating check.	15/17	22/22	33/36	30/35
Unresolved verification	An explicitly posed diagnostic distinction remains unresolved before the final diagnosis.	13/17	15/22	13/36	4/35
Unsupported interpretation	A factual or causal claim exceeding or contradicting the cited return supports component selection.	5/17	5/22	6/36	4/35
Information loss or distortion	Decision-relevant content is demonstrably lost or altered in a derived representation or final account, and the changed content supports selection.	0/17	1/22	1/36	0/35
Final-diagnosis continuity	Any parsed, normalized component in the final diagnosis matches the non-root leading diagnosis in the last expressed diagnostic state.	6/17	14/22	33/36	27/35

We reviewed all 110 final root omissions in RQ2's fixed low/high rival-prominence groups that contained a concrete root observation: 39 GPT-4o histories and 71 GPT-5.5 histories. GPT-5.6-

Sol was used as an analysis assistant in an interactive, tool-assisted review workflow. For each history, it examined the recorded Controller sequence and saved final diagnosis, consulting the corresponding Executor returns and execution code when an assignment depended on whether a requested check was completed, whether an interpretation was supported by the returned content, or whether decision-relevant information was lost or transformed. The criteria in Table 13 were developed before full-population application and organized around source-linked diagnostic episodes: a diagnostic claim or question, the relevant returned content, the subsequent interpretation or action, and its connection to component selection. Each positive assignment recorded a category, rationale, and source excerpt with step references. Categories could overlap but were counted at most once per history, and the primary component-related analysis excluded errors confined to the final resource-reason label. Group labels were withheld during annotation, and group-specific frequencies were not computed until all histories had been reviewed. The same assistant subsequently re-examined the assignments and their scope, with revisions and withdrawals retained in the review ledger. The authors reviewed the source excerpts.

These frequencies describe recorded reasoning within failures, not failure risk or causal effects of rival prominence; repeated executions share incidents. Attribution records the properties used to support a competing root-cause hypothesis, not an error in using symptom cues, and unsupported interpretation is judged against the cited return. Resource/reason-only errors are excluded. For final-diagnosis continuity, empty or unmatched final diagnoses and last states in which the root is the leading diagnosis or no leading diagnosis is expressed remain in the denominator but do not qualify.

F.2 Recorded Grounds and Diagnostic Continuity

Relative Signal Properties Shape Competing Diagnoses. In most reviewed failures, agents use relative properties of the observed signals to favor a non-root component or discount the root (Figure 9). At least one such cue—anomaly magnitude, timing, frequency or persistence, or system position—appears in 37/39 GPT-4o and 63/71 GPT-5.5 failures, with relative anomaly magnitude being the most common. These frequencies do not imply that using such cues is itself erroneous; rather, they show how competing diagnoses acquire support after root telemetry has been observed. We next examine what subsequent investigation does once such an alternative has gained diagnostic priority.

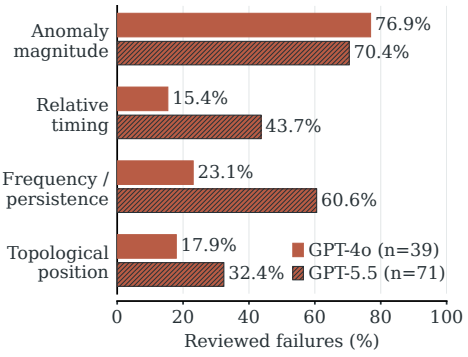


Figure 9: **Recorded reasons for alternative preference.** Recorded reasons used to favor a non-root component or discount the observed root across reviewed final-root omissions (39 GPT-4o and 71 GPT-5.5 histories); categories can overlap and are descriptive rather than comparisons of failure risk.

Final Selection Preserves or Extends Intermediate Diagnostic Priority. The final answer frequently carries forward diagnostic priority established during investigation: it selects the last non-root leader in 20/39 GPT-4o failures and 60/71 GPT-5.5 failures. This continuity is especially pronounced for GPT-5.5 and shows that the competing diagnosis expressed before commitment often remains the component ultimately selected. In Figure 10, the final answer preserves `MG01` even as subsequent checks revise the proposed resource explanation. Other failures extend an earlier attribution through an additional unsupported inference at completion (Section F.3). Thus, final errors can preserve an alternative's established diagnostic priority even when later investigation changes its supporting explanation or leaves the root–alternative distinction unresolved.

F.3 Failure Case Study

These cases illustrate how returned content supports, or fails to support, the diagnosis selected during an investigation. They explain the patterns in Table 13; they are not additional prevalence samples or matched comparisons. Figure cards paraphrase the recorded sequences.

Subsequent Investigation Can Refine an Alternative Without Revisiting Its Basis. Figure 10 illustrates how later investigation can deepen an alternative diagnosis without directly revisiting the judgment that moved the diagnosis away from the observed root. After observing the root `Tomcat04`, the agent initially favors `Tomcat03` because of its larger disk-read anomaly, then uses the reconstructed call chain to relocate the suspected cause to the downstream `MG01`/`MG02` group. Subsequent checks distinguish between the MG components and revise the proposed resource explanation for `MG01`; even a negative disk-read result changes the explanation within this alternative causal location rather than reopening the earlier root–alternative distinction. The final answer therefore selects `MG01`.

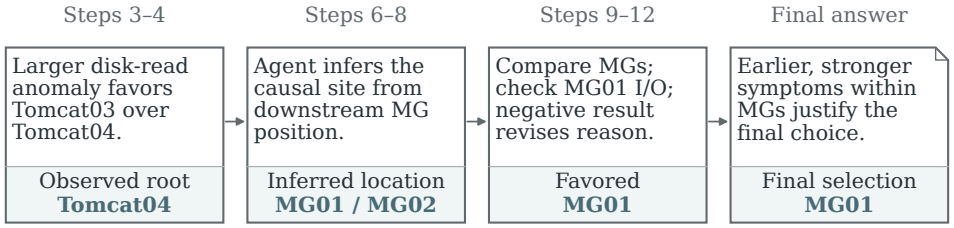


Figure 10: **From downstream localization to component selection.** An alternative causal location is refined through subsequent investigation, ending in `MG01`.

Some Intended Comparisons Remain Unresolved Before Commitment. A different pattern occurs when the investigation explicitly recognizes a diagnostic distinction but does not complete the comparison. An explicitly posed verification remains unresolved at commitment in 28/39 GPT-4o and 17/71 GPT-5.5 reviewed failures. In Figure 11, the agent requests a direct comparison between `Mysql02` and the observed root `Tomcat04`; when the request fails, the retry checks only `Mysql02`'s CPU I/O wait, confirming that `Mysql02` is anomalous without resolving the intended comparison before the agent selects it. Thus, further investigation can add support for a favored alternative without establishing the distinction that motivated the check.

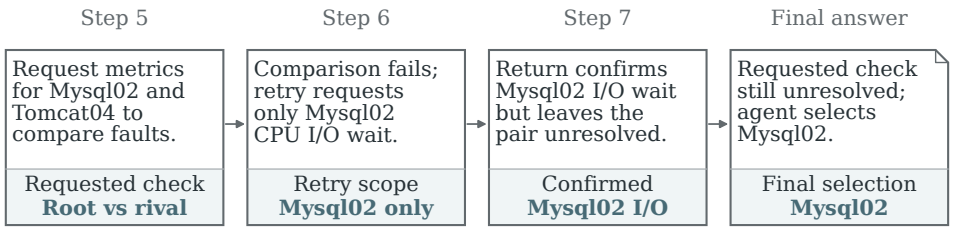


Figure 11: **From unresolved comparison to final selection.** A failed Mysql02–Tomcat04 comparison is followed by a Mysql02-only retry and final selection without resolving the requested distinction.

Unsupported Interpretation. Figure 12 shows an unsupported interpretation in a high-prominence Bank history with labeled root-cause component `MG01`. The return reports three GC "Allocation Failure" events for `Tomcat04`, compared with one or two for the other Tomcat components, alongside a lower reported peak memory value. It nevertheless describes `Tomcat04` as having the most severe memory-related issues based on GC frequency. The Controller uses this frequency-based severity assessment to select `Tomcat04` and submits heap out-of-memory as the reason. The returned comparison establishes neither heap exhaustion nor an out-of-memory event. The case connects a relative symptom count to component selection and an unsupported claim about the failure mechanism.

Information Loss or Distortion. Figure 13 shows a rare component-identity error. The initial Telecom metric rows retain component and KPI identifiers, including the labeled root-cause component `db_007`. The Executor aggregates anomaly segments across these series but drops the component and KPI keys. The remaining `fault_id` identifies an anomaly segment within a series, not a system component. The Executor nevertheless reports the maximum-deviation segment, `fault_id=0`, as "component 0," and the Controller accepts that attribution.

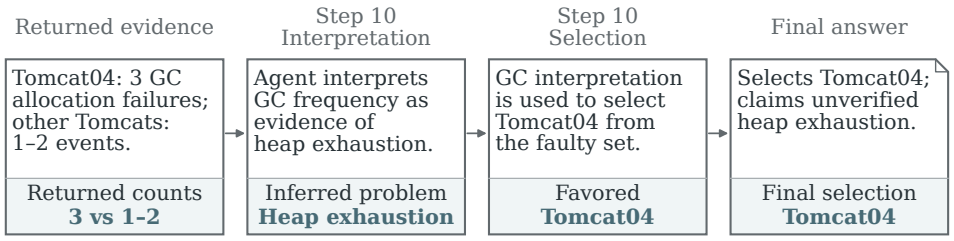


Figure 12: **GC frequency becomes a diagnosis of heap exhaustion.** Labeled root-cause component `MG01`. Interpretation and selection occur in the same Controller response.

The final diagnosis selects `docker_001` through a container-failure interpretation without a verified mapping from the segment identifier. This sequence connects loss of component identity in a derived table to an unsupported final attribution. It does not establish deletion of the original returned content or loss of a correctly selected component during answer transmission.

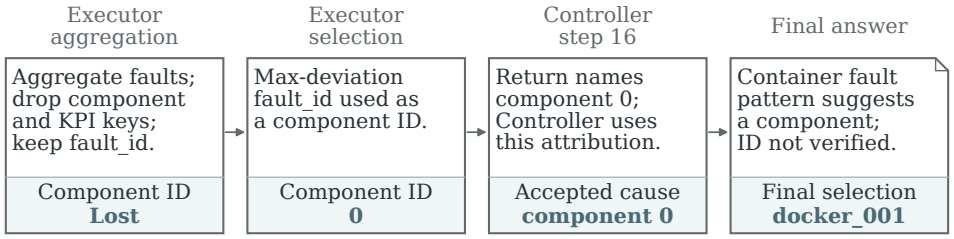


Figure 13: **An anomaly-segment label becomes a component identity.** High rival prominence; labeled root-cause component `db_007`. Aggregation removes the identifiers needed to connect the selected segment to a component.

G Limitations and Scope

Our analysis is retrospective and observational. The reported associations do not identify causal effects of incident-side signal conditions, investigation actions, or intermediate diagnostic updates. Post-observation results are conditional on a concrete root observation and should not be directly generalized to all agent executions. Candidate states are reconstructed from visible trace content and should be interpreted as expressed diagnostic states rather than latent beliefs. Likewise, a concrete root observation establishes that telemetry from the root-cause component was displayed, not that the evidence was sufficient to identify the cause or distinguish it from competing root-cause hypotheses. The primary empirical evidence comes from OpenRCA under the evaluated model and tool configurations, and the targeted-design results concern the evaluated implementations on matched Telecom incidents. Broader conclusions about other RCA architectures, benchmarks, or operational environments require separate validation.