TRACE: Traceable Root-cause Analysis with Calibrated Evidence-grounded agents

Aniket Joshi **Cyrus Dsouza** **Sejal Jain** **Manjunath Channappagoudar**
Amazon
{anikjosh, dsocyrus, sejaljn, mcgoudar}@amazon.com

Abstract

Root Cause Analysis (RCA) for defects in large e-commerce enterprises is manual and slow: a single defect spanning thousands of microservices and SOPs takes experts one to two weeks to diagnose. Generic agentic RCA frameworks fail on this workload because they stop at proximate causes, cannot verify numeric claims, ignore past reviewer decisions, and return enumerated hypotheses rather than the quantified narratives business teams act on. We present TRACE, an end-to-end agentic RCA system that combines supervised hypothesis-driven investigation with recursive why-drilling, an evidence-traceable multi-tool suite with stable artifact identifiers, a feedback retrieval and ingestion pipeline, and a two-layer confidence framework combining six mechanistic checks with an LLM judge - including a novel numerical-traceability score. Across several thousand cohorts spanning eight use cases in three metric categories (cost, quality, and inventory), TRACE achieves 45.5% expert acceptance; on a 240-cohort head-to-head benchmark, 47.1% vs. 28.8% for a matched ReAct baseline ($+18.3$pp, $p<0.001$) with calibrated confidence (AUC 0.819, ECE 0.041).

1 Introduction

A large e-commerce enterprise's operational surface spans thousands of microservices, hundreds of teams, and tens of thousands of Standard Operating Procedures (SOPs). Defects on this surface - anomalous quality, cost, or inventory metrics, compliance misses - rarely originate where they surface: an anomaly may trace back to a catalog attribution issue, a warehouse packaging SOP, or a compliance regulation several hops upstream. Domain experts in enterprise quality-improvement programs historically spend one to two weeks per defect reconstructing this causal chain by hand - reading wiki pages, hunting for datasets, and tracking configuration changes; at the scale of such programs, manual RCA is the binding constraint.

LLM agents have made automated RCA credible: ReAct (Yao et al., 2023)-style loops extend to DevOps incidents (Wang, 2026; Pei et al., 2025; Xu et al., 2025), firmware security (Zhang et al., 2025), and text-to-SQL over enterprise tables (Cao et al., 2026). Yet they fall short for enterprise experts on four counts. *First*, they terminate at proximate causes - "driver X explains the anomaly" is not actionable until the chain is drilled to an atomic level (an SOP line, a catalog attribute, a concrete operational decision). *Second*, they cannot verify numeric claims; an LLM-generated aggregate untraceable to a tool-call artifact cannot drive financial decisions (Zheng et al., 2023). *Third*, they ignore past expert decisions, re-litigating rejected hypotheses across cohorts. *Fourth*, they produce the wrong output shape: experts and business teams consume quantified *defect narratives* in the enterprise's operational vocabulary, not enumerated hypothesis lists.

We present TRACE, an agentic RCA system that closes all four gaps. Our contributions are (i) an architecture combining hypothesis-driven investigation with automatic recursive why-drilling to atomic actionable causes; (ii) an evidence-traceable tool suite (distributed query, page-evaluation, retrieval, and deterministic-impact tools) in which every tool call emits a stable artifact identifier; (iii) a three-tier feedback retrieval pipeline with four stage-specific injection points, delivering same-day continual learning without model weight updates; (iv) a two-layer confidence framework pairing six mechanistic checks with five LLM-judge fidelity checks, including a novel *numerical-traceability* score that verifies every numeric token against its cited artifacts, calibrated with a feedback-derived Bayesian prior; and (v) a layered output design with model-separated grounding (LLM reasons qualitatively, deterministic pipelines compute aggregates) that eliminates a large class of numeri-

cal hallucinations by construction. In deployment across eight use cases and several thousand cohorts, TRACE achieves 45.5% expert acceptance; on a 240-cohort paired head-to-head benchmark it reaches 47.1% versus 28.8% for a matched ReAct baseline ($+18.3$ pp), with confidence predicting acceptance at AUC 0.819. Beyond the specific system, we argue five design principles generalize to any enterprise agentic investigation where numerical claims must be audited and human feedback is expensive.

2 Related Work

ReAct (Yao et al., 2023) established the tool-use reasoning loop behind modern RCA agents, extended to incident response by AgentTrace (Wang, 2026), Flow-of-Action (Pei et al., 2025), OpenRCA (Xu et al., 2025), and TimeRAG (Yang et al., 2024); APEX-SQL (Cao et al., 2026) adds Hypothesis-Verification loops for enterprise text-to-SQL, and GraphRAG (Mountzouris et al., 2026) and data-to-paper (Ifargan et al., 2025) emphasize traceable evidence lineage. Recursive architectures (FirmHive (Zhang et al., 2025), ReDel (Zhu et al., 2024), DEPART (Hsu et al., 2025), ThReaD (Schroeder et al., 2025)) spawn sub-agents for deeper analysis. TRACE differs by targeting enterprise *business* defects rather than infrastructure incidents, triggering recursive recursive why-drilling specifically on PROVEN hypotheses, and treating numerical grounding as a first-class, mechanically verified requirement rather than a post-hoc check. We do not adapt these systems as baselines: each presupposes inputs that do not exist for enterprise business defects (Flow-of-Action needs SOP-graphs keyed to incident types, AgentTrace execution logs, OpenRCA telemetry with ground-truth failure records), and constructing them ourselves would let our own choices shape the baseline's results. §4 instead isolates TRACE's design via subtractive ablation and an additive structured-loop variant.

For evaluation, G-Eval (Liu et al., 2023), RAGAS (Es et al., 2025), LLM-as-a-Judge (Zheng et al., 2023), multi-agent debate (Du et al., 2023), and the Aegean consensus protocol (Ruan et al., 2025) judge LLM outputs with other LLMs, and standard RAG (Ram et al., 2023) injects knowledge at a single generation point. We instead pair an independent *mechanistic* audit (six deterministic checks, including the novel fuzzy-match numerical-traceability score) with the LLM judge, calibrate via a feedback-derived Bayesian prior over hypothesis categories, and split feedback retrieval across *four* stage-specific injection points - using retrieval rather than fine-tuning as the vehicle for expert feedback.

3 Methodology

3.1 Problem Formulation

We formalize enterprise RCA as an agentic investigation task with human actionability as the success criterion. The input is a *defect statement* -a natural-language anomaly description (e.g., "the cost-per-unit metric is elevated by $3.20/unit"), metadata of all the existing tables related to the anomaly, an applicable use case $u \in \mathcal{U}$ (one of eight use cases, with per-use-case configurations in Appendix 6), and a knowledge context $\mathcal{K}_u$ of domain documentation and historical reviewer feedback. The output is a triple $\mathcal{O} = \langle \mathcal{H}^\star, \mathcal{A}, c \rangle$: validated root-cause hypotheses organized into a causal chain, quantified *defect narratives* in the enterprise's operational vocabulary, and a composite confidence score $c \in [0, 100]$. The primary success signal is *expert acceptance*: a reviewer presented with $\mathcal{O}$ marks it actionable (sufficient for a business unit to initiate corrective action) or not. A critical requirement is *numerical grounding*: for every numeric token τ in the final report, an evidence artifact must exist with value $\phi(\tau)$ such that $|\tau - \phi(\tau)| \leq \epsilon$, since LLM numerical hallucination is catastrophic when outputs drive financial decisions. The problem is to maximize expected expert acceptance under this constraint while emitting c whose distribution aligns with empirical acceptance, so that c functions as a useful filter for stakeholder attention.

3.2 System Overview

Figure 1 shows the end-to-end architecture, organized as four bounded subsystems—the *Agentic Investigation Loop*, the *Two-Layer Confidence* block, the *Expert-Appropriate Output* block, and the *Human-in-the-Loop Feedback* block - fed by a left-hand input column and joined by a closed feedback path that does not require model fine-tuning.

A *Context Pre-Loader* retrieves four streams in parallel: the defect statement and transaction-data references, domain knowledge from a managed vector KB (SOPs, audits, historical RCAs, glos-

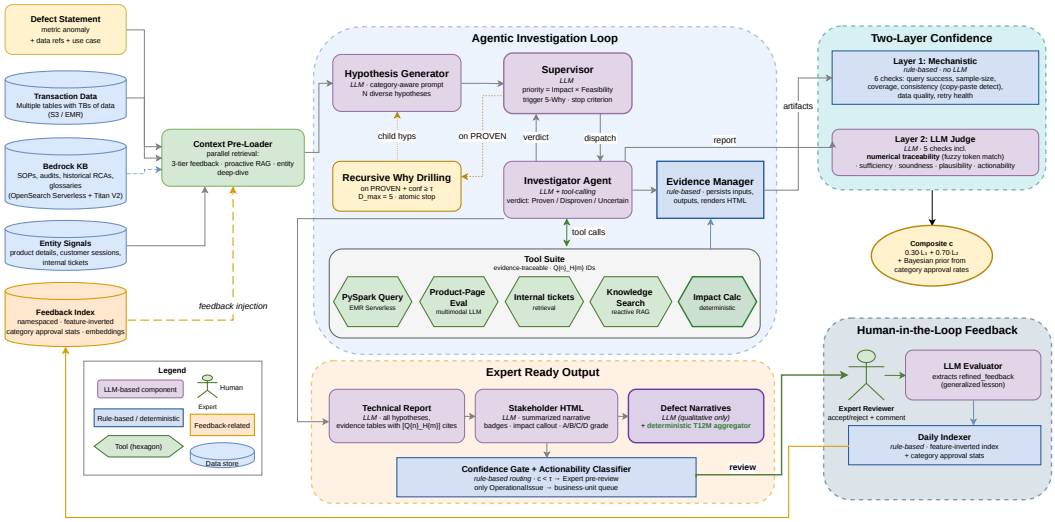

Figure 1: End-to-end architecture of the system. Structural feedback retrieval (orange) and RAG knowledge retrieval (blue) enter the pipeline at four and two points respectively; the investigation loop is supervised and supports recursive recursive why-drilling.

saries), relevant reviewer feedback from the Feedback Index via three-tier retrieval, and per-entity deep-dive signals (catalog, customer comments, tickets). Inside the *Agentic Investigation Loop*, a *Hypothesis Generator* (LLM) produces N categorized hypotheses; a lightweight *Supervisor* scores them on Impact $\times$ Feasibility, dispatches the top candidate to an *Investigator Agent* (LLM with tool-calling), and on a PROVEN verdict triggers *Recursive Why-Drilling* of child hypotheses toward an atomic cause. Every tool call is persisted by a rule-based *Evidence Manager* under stable `Q{n}_H{m}` identifiers. The *Two-Layer Confidence* subsystem then scores the investigation with Layer 1 (six rule-based mechanistic checks) and Layer 2 (an LLM judge with five fidelity checks including the fuzzy-match numerical-traceability score), combined as $c = 0.30\,L_1 + 0.70\,L_2$ and calibrated by a category-level Bayesian prior. The *Expert-Appropriate Output* subsystem emits a technical report, a stakeholder HTML, and defect narratives, with a rule-based gate routing sub-threshold or non-OPERATIONALISSUE outputs to expert pre-review. Finally, the *Human-in-the-Loop Feedback* subsystem closes the loop: an *LLM Evaluator* extracts generalized lessons from accept/reject decisions and a *Daily Indexer* refreshes the Feedback Index, which flows back into the Context Pre-Loader - yielding same-day continual learning without any model weight update.

3.3 Hypothesis-Driven Investigation with Recursive Why-Drilling

Generic agentic RCA terminates at proximate causes: "carrier X is driving cost" is not actionable until we explain why X is being selected for this pattern. We formalize a **supervised hypothesis-driven loop** augmented with automatic **recursive why-drilling**. A category-aware prompt (Appendix 6) elicits N structured hypotheses, each with a category, statement, causal mechanism, features to investigate, expected pattern, and estimated impact; category diversity is enforced to avoid LLM mode collapse. The investigator agent tests each selected hypothesis via a bounded sequence of tool calls and emits a structured verdict with a conclusion in {PROVEN, DISPROVEN, UNCERTAIN}, a confidence, and supporting/contradicting evidence items with sample sizes. When a hypothesis h is concluded PROVEN with confidence above τ_{drill}, the supervisor automatically generates a child set by prompting an LLM to enumerate: "given that h is the proximate cause, why does the antecedent condition hold?" Recursion is bounded by $D_{\max} = 5$ and terminates early at an atomic cause (a policy rule, data-quality issue, or concrete operational decision) that is directly actionable, when sample sizes drop below threshold, or when child confidence falls below τ_{drill}. The resulting chain is surfaced as a "Root Cause Chain" section, empirically the single most-valued output in expert reviews. The **Supervisor** is a lightweight LLM

controller (Appendix 6) that scores uninvestigated hypotheses on Impact × Feasibility with a category-diversity bonus, dispatches the top-scored hypothesis, triggers why-recursion when warranted, and terminates based on the marginal value of the next-best hypothesis, also re-dispatching recent low-quality investigations with instructions to explore different features.

3.4 Evidence-Traceable Tool Use

A central architectural commitment is that every claim surfaced to a stakeholder must be traceable to a specific tool-call artifact, a requirement flowing from our grounding constraint (§3.1). The investigator has access to a registry of domain-specific tools activated per use case: a **distributed query tool** (natural-language → PySpark, executed on a serverless Spark backend); a **product-page evaluation tool** for catalog defects using a multimodal LLM; a **ticket retrieval tool** for operational incident tickets; a **knowledge/wiki search tool** (reactive RAG over the managed vector KB) etc. All tool calls publish through a unified Evidence Manager that assigns each call a stable identifier `Q{n}_H{m}` and persists inputs, outputs, and a rendered view. These identifiers propagate through the pipeline-investigator verdicts citing them as [`Q0_H1`], preserved through report generation and stakeholder HTML-making numerical traceability (§3.6) verifiable at the token level.

3.5 Context Augmentation: RAG and Feedback Retrieval

Two non-data knowledge sources are essential: *structured reviewer feedback* (accept/reject decisions with refined lessons) and *unstructured domain knowledge* (SOPs, audits, historical RCAs, glossaries). These have different retrieval characteristics and we design complementary strategies.

Domain knowledge via RAG - Documents are indexed in a single managed vector KB with a `namespace` metadata filter, embedded with a production text-embedding model. The KB is consumed proactively at init (2–3 targeted queries concatenated into the hypothesis prompt) and reactively during investigation (an agent tool call for context data analysis alone cannot supply). **Three-tier feedback retrieval and ingestion** - Because reviewer feedback is structured JSON with explicit metadata, we do a **Tier 1 (structural)** check by applying namespace filter, feature-overlap scoring (exact feature-value match w=1.0, same-feature-different-value w=0.3), and a time-window filter, narrowing thousands to ∼50–200 candidates via a feature-inverted index. **Tier 2 (semantic)** embeds anomaly description plus candidate hypotheses with the same embedding model and ranks by $s_{\text{combined}} = 0.4\, s_{\text{struct}} + 0.6\, s_{\text{sem}}$; feedback embeddings are pre-computed, so retrieval is inference-free. **Tier 3 (context fit)** deduplicates, balances REJECTED/APPROVED feedbacks at $60/40$ (rejected patterns prevent repeated mistakes), and truncates to the injection token budget. We inject feedback at distinct granularities: hypothesis generation (rejected+validated patterns at namespace level), per-hypothesis investigation (feature-matched feedbacks), report generation (APPROVED-only at cohort similarity), and confidence calibration (aggregate category-level statistics). The system does not fine-tune weights; feedback flows purely through retrieval and confidence calibration, yielding same-day continual learning where an expert rejection today influences tomorrow's investigations.

3.6 Two-Layer Confidence Framework

Agentic systems producing numerical business conclusions cannot be used on raw output alone. We introduce a two-layer framework combining mechanistic validation over raw artifacts with LLM-as-Judge evaluation of report fidelity including a novel *numerical traceability* check, composed with a feedback-derived Bayesian prior. **Layer 1: Mechanistic evidence quality** - Six deterministic checks run over the artifact stream: query success rate; sample-size adequacy (fraction of claims with $n \geq 30$ and median observed n); investigation coverage; evidence–verdict consistency (flagging e.g. PROVEN with zero supporting evidence and, uniquely, duplicate `final_verdict` strings across hypotheses as a reliable copy-paste-hallucination signal); data-quality signals; and retry health. L_1 is a weighted sum with weights tuned on a development set, and emits structured flags (e.g., `UNSUPPORTED_PROVEN: H3`) surfaced to experts as targeted diagnostics. **Layer 2: Report fidelity with numerical traceability** - An LLM judge evaluates the report on five dimensions: numerical traceability, evidence sufficiency, reasoning soundness, root-cause plausibility, and actionability. The novel check is mechanistic: let $\mathcal{T}$ be numeric tokens in the report and $\mathcal{V}$ the union of numeric values in referenced artifacts; for each $\tau \in \mathcal{T}$ we attempt a fuzzy match $\phi(\tau) \in \mathcal{V}$ with unit normalization and tol-

erance $|\tau - \phi(\tau)|/\max(|\tau|, \epsilon_0) \leq 0.005$, yielding $s_{\text{num}} = 100 \cdot |\{\tau : \phi(\tau) \text{ exists}\}|/|\mathcal{T}|$. Untraceable tokens are almost always hallucinations, plausible-looking invented figures not grounded in evidence; s_{num} receives the largest Layer 2 weight and orphan tokens are reported verbatim for expert review.

Composite and Bayesian calibration - $c_{\text{raw}} = 0.30\,L_1 + 0.70\,L_2$; the tilt reflects that report fidelity correlates more strongly with expert acceptance than raw evidence quality. We calibrate using category-level priors: for a hypothesis of category κ with historical approval rate ρ_κ, $c = \sigma\big(\text{logit}(c_{\text{raw}}/100) + \lambda \log \frac{\rho_\kappa}{1-\rho_\kappa}\big) \cdot 100$. This down-weights PROVEN conclusions in historically low-acceptance categories. Both layers are necessary: the mechanistic layer catches artifact-level inconsistencies and copy-paste verdicts that LLM judges overlook, while the judge catches semantic implausibility the mechanistic layer cannot reason about. In our evaluation, c is rendered as a stakeholder badge with grades and routes sub-threshold cohorts to expert pre-review.

3.7 Expert-Appropriate Output Generation

Experts and business stakeholders have distinct information needs; generic enumerated-hypothesis reports are rejected as too verbose and missing the narrative unit used in enterprise business communication - the *defect narrative*. We therefore emit three outputs: a **technical report** with all hypotheses, evidence tables with `Q{n}_H{m}` citations, and diagnostic flags (for expert review); a **stakeholder HTML** (Appendix 6) with prioritized hypothesis cards, an impact callout, and a confidence badge; and **defect narratives**, short quantified narratives for operational teams. Visibility is confidence-gated: sub-threshold outputs are routed to expert pre-review rather than delivered to business teams.

Beyond the Layer 2 traceability check, we structurally preclude a class of numerical hallucinations by separating qualitative LLM reasoning from quantitative computation: no aggregate dollar totals may appear in narrative bodies (these are computed by the downstream deterministic pipeline). An actionability classifier labels each report as OPERATIONALISSUE, DATAQUALITY, FALSEPOSITIVE, or NORMALBEHAVIOR; only OPERATIONALISSUE reports are routed to business-unit queues.

Metric	ReAct	TRACE
Expert acceptance	28.8%	**47.1%**
Num. trace. s_{num}	62.3%	**94.6%**
Orphan tokens/report	7.8	**1.2**
Atomic-cause (≥ 3)	4.2%	**58.3%**
Review time (med.)	8 m 42 s	**3 m 18 s**
Tool calls/cohort	6	14

Table 1: End-to-end comparison on 240 matched cohorts (double-blind expert review).

4 Deployment and Evaluation

4.1 Dataset and Deployment Setting

TRACE is evaluated across eight use cases spanning three metric categories (cost, quality, inventory); each anomalous cohort becomes an RCA input per §3.1. The system queries a large-scale transactional data substrate, a domain knowledge base, and a feedback index of past reviewed cases. Per-use-case expert acceptance ranges from 41.5% to 49.5% (overall 45.5%) at 10–14 min median latency, with several hundred reviewed cohorts per use case.

Cohort selection. Cohorts are real production defects: a cohort is a feature slice whose outcome metric significantly deviates from its parent slice's baseline (e.g., a category$\times$brand slice with a 10% return rate against the category's 5%). Candidates are ranked by *estimated recovery* - the metric improvement expected if the defect is fixed - and high-recovery cohorts enter RCA; evaluation cohorts reflect the live production distribution.

Review protocol and agreement. Each report is independently reviewed by three experts trained in the enterprise quality-improvement program on a shared actionability rubric ("is the evidence sufficient for a business unit to initiate corrective action"), marking it ACTIONABLE/NOTACTIONABLE with the final label by majority vote. The pool comprises 46 distinct experts; inter-rater agreement is Fleiss' $\kappa = 0.62$ (substantial). In the E1 head-to-head (§4.2), the same expert reviewed both arms of each pair, blind to system identity, so the paired comparison controls for per-rater leniency. Experts never see confidence scores during review; labels are used only post hoc to tune confidence weights, so calibration results measure alignment with unanchored expert judgment.

4.2 Experimental Setup

All the below experiments use Claude Sonnet 4.5 as backbone LLM and a standard text-embedding model for embeddings.

E1 (End-to-end) - We compare TRACE against a VANILLA-REACT baseline (Yao et al., 2023) with the same backbone LLM and tools but (no Evidence Manager, why-drilling, RAG/feedback, confidence layer, or narrative pipeline); experts review outputs double-blind on 240 matched cohorts stratified across use cases. An *additive* variant (REACT + hypothesis generation + supervisor only) is evaluated on the same 240 cohorts to isolate a generic structured loop from TRACE-specific components.
E2 (Ablations). On 480 labeled cohorts, we remove one component at a time: *why-drilling*, *Feedback*, *RAG*, *Confidence* (deliver all outputs unfiltered), *Narratives* (emit only the technical report); plus a joint removal of *why-drilling* + *Feedback* to measure interaction effects.
E3 (Confidence Calibration). On the full labeled set, we measure AUC, Brier score, and ECE (10 bins) of c against expert acceptance, comparing Layer 1 only, Layer 2 only, composite, composite+prior, and an LLM self-critique baseline matching G-Eval (Liu et al., 2023)/RAGAS (Es et al., 2025).
E4 (Feedback injection schedule). On a 200-cohort held-out set, we vary feedback active injection points $\{0, 1, 2, 3\}$ and include a control (same token budget concatenated at a single point) to isolate *where* feedback is injected from and *how much*. Confidence weights were tuned on the earliest 200 labeled cohorts; evaluation uses the remaining labeled cohorts plus the 240 paired E1 cohorts. Significance uses McNemar's test (paired acceptance) and DeLong's test (AUC differences).

4.3 End-to-end comparison and component ablation

Table 1 reports the head-to-head. TRACE reaches 47.1% expert acceptance versus 28.8% for VANILLA-REACT on 240 paired cohorts ($+18.3$ pp, $p < 0.001$, McNemar).[1] Numerical traceability rises from 62.3% to 94.6%, and the atomic-cause rate (chain drilled ≥ 3 levels) from 4.2% to 58.3% - confirming why-drilling reaches substantively deeper findings than a single-pass agent. Median expert review time drops from 8 m 42 s to 3 m 18 s, driven by the narrative-centric stakeholder HTML.

Component ablation (Table 2) shows the confidence gate is the largest single contributor (-10.7 pp), because low-fidelity outputs that would have been pre-reviewed instead reach the expert; narrative output (-9.0) confirms that the technical report alone is too verbose; why-drilling (-6.6) and feedback (-5.7) are substantial; RAG (-2.6) is complementary. All drops are significant at $p < 0.01$. Removing the gate drags s_{num} from 94.8% to 84.2% - not because reports contain more hallucinations, but because the gate had been withholding the lowest-fidelity reports.

Joint ablation and additive baseline. Removing why-drilling and feedback *together* yields 36.9% (-9.9 pp; Table 2, bottom row) - less than the -12.3 pp sum of the individual drops, a sub-additive interaction consistent with partial overlap: feedback frequently steers the *direction* of recursive why-drilling. Conversely, an *additive* variant - REACT plus hypothesis generation plus supervisor only - evaluated on the same 240 E1 cohorts reaches 35.4% (85/240) acceptance, versus 28.8% for the bare agent and 47.1% for TRACE: a generic structured loop recovers roughly a third of the gap; the remaining 11.7 pp is attributable to TRACE-specific components.

4.4 Confidence calibration and injection schedule

Tables 3 and 4 present complementary results. On calibration (Table 3), our composite predicts expert acceptance at AUC 0.819 with ECE 0.041, substantially better than LLM self-critique (AUC 0.671, ECE 0.138). Layer 1 (0.742) and Layer 2 (0.776) contribute complementary signal, confirming the mechanistic-plus-judge thesis; the Bayesian prior adds $+0.015$ AUC and cuts ECE from 0.051 to 0.041 by correcting over-confidence in low-approval-rate categories. On the injection schedule (Table 4), a single point at hypothesis generation yields $+2.8$ pp; progressively activating points 2–4 adds $+3.9$ pp more to 47.1%. The key row is the control: concatenating the same $\sim$3,000 tokens at a single point yields only 44.1% (-3.0 pp vs. the distributed schedule) - indicating *where* feedback is injected matters as much as *how much* and what is injected.

[1] Table 1's 47.1% (240 paired E1 cohorts) and Table 2's Full-row 46.8% (separate 480-cohort ablation set) are measured under the identical labeling protocol; the difference reflects the different cohort sets.

Variant	Acc.	Δ	s_{num}
Full	**46.8%**	—	**94.8%**
–Why-drill	40.2%	−6.6	93.1%
–Feedback	41.1%	−5.7	92.4%
–RAG	44.2%	−2.6	94.0%
–Confidence	36.1%	−10.7	84.2%
–Narratives	37.8%	−9.0	94.7%
–Drill & Fdbk.	36.9%	−9.9	93.5%

Table 2: Component ablation (480 cohorts).

Conf. signal	AUC	Brier	ECE
Self-critique	0.671	0.211	0.138
L_1 only	0.742	0.188	0.092
L_2 only	0.776	0.173	0.068
Composite c_{raw}	0.804	0.164	0.051
+prior c	**0.819**	**0.159**	**0.041**

Table 3: Confidence calibration (full labeled set).

Inj. config	Acc.	Δ	Tok.
None	40.4%	—	0
Pt. 1 only	43.2%	+2.8	1,487
Pt. 1+2	45.6%	+5.2	2,284
Full (1–4)	**47.1%**	+6.7	2,978
Same tok., Pt. 1	44.1%	+3.7	2,971

Table 4: Injection-schedule ablation (200 held-out).

Across the evaluation, the confidence gate routes 14.7% of cohorts to expert pre-review at precision 0.71.

Cost and latency. Median end-to-end latency is 10–14 min per cohort at ∼$1.4 inference cost, versus one to two expert-weeks manually. VANILLA-REACT costs ∼$0.7 per cohort (6 vs. 14 tool calls) but rarely reaches atomic causes; TRACE's ∼2× inference cost purchases the +18.3 pp gain and the 8 m 42 s → 3 m 18 s review-time reduction. Only the investigation loop and Layer-2 judge are LLM-bound; all other subsystems are rule-based with negligible cost.

4.5 Failure analysis

Roughly 51–58% of TRACE outputs are rejected per use case. Rejections show two dominant modes, neither of which is numerical error. *(a) Contextual-knowledge gaps*: mathematically correct but unactionable findings - e.g., an intra-city delivery-cost anomaly flagged with numerically flawless evidence, rejected because a localized weather disruption was the known, temporary cause, knowledge absent from every onboarded source. *(b) Duplicate findings*: re-surfacing already-known defects, directly mitigated by the feedback loop (part of its −5.7 pp in Table 2). Investigation quality, numerical grounding, and formatting were rarely the cause: failures concentrate in *knowledge coverage*, motivating the interactive knowledge-injection direction in §5.

5 Future Work

Three directions extend this work: (i) *interactive knowledge injection* - rejections concentrate in contextual-knowledge gaps (§4.5), motivating conversational expert corrections that enter the feedback index; (ii) *automated remediation* (vendor engagement, SOP-change proposals, automated case creation); and (iii) *causal validation* via RCT-style experiments on an enterprise A/B platform. All three build on the evidence-traceability and feedback-calibration substrates introduced here.

6 Conclusion

We presented TRACE, an agentic RCA system that closes four gaps in generic agentic RCA - proximate-cause termination, unverified numeric claims, absent feedback consumption, and wrong output shape - via supervised why-drilling investigation, evidence-traceable tooling with stable artifact identifiers, multi-granular feedback retrieval, and a two-layer mechanistic-plus-judge confidence framework with a novel numerical-traceability score. Across eight use cases and several thousand cohorts, TRACE reaches 45.5% expert acceptance in deployment; on a 240-cohort paired benchmark it reaches 47.1% versus 28.8% for a matched ReAct baseline (+18.3 pp) with calibrated confidence (AUC 0.819). Its design principles generalize to any enterprise agentic investigation where numerical claims must be audited and human feedback is expensive.

Limitations

TRACE has a few limitations that also point to ongoing work. The associations it surfaces are derived from observational transactional data, and the system relies on LLM reasoning - guided by domain knowledge and prior feedback - to infer their causal nature; a chain can thus be internally consistent and numerically grounded yet still reflect a confounded correlation, since distinguishing correlation from causation from observational data alone is fundamentally hard, and we are actively extending TRACE with explicit causal models and experiment-based validation to promote high-impact findings from correlational to causal evidence. Root-cause analysis is also inherently

subjective - for a given defect there is rarely a single verifiable "correct" cause and reasonable experts can disagree on the most actionable explanation - so we evaluate against expert judgments rather than an objective oracle, and our acceptance and calibration figures (AUC, ECE) should be read as alignment with expert assessment rather than provable correctness. Reaching a specific, actionable root cause is further complicated in an enterprise setting, where a single defect can span many systems owned by different teams, each with its own data stores and tooling; TRACE can only investigate what its tool suite can reach, so broad and deep coverage requires aligning with these teams to onboard the data sources and tools they already use, and gaps in this integration bound how deeply the causal chain can be drilled. Finally, enterprise root-causing remains heavily human-dependent: output quality depends on experts supplying good feedback and steering documents - corrections, domain rules, and accept/reject decisions - that guide the LLM toward the right hypotheses and away from previously rejected ones, which is precisely why feedback consumption is a first-class component of our design but also means new use cases face a cold-start period until enough feedback accumulates.

References

Bowen Cao, Weibin Liao, Yushi Sun, Dong Fang, Haitao Li, and Wai Lam. 2026. Apex-sql: Talking to the data via agentic exploration for text-to-sql. *Preprint*, arXiv:2602.16720.

Yilun Du, Shuang Li, Antonio Torralba, Joshua B. Tenenbaum, and Igor Mordatch. 2023. Improving factuality and reasoning in language models through multiagent debate. *Preprint*, arXiv:2305.14325.

Shahul Es, Jithin James, Luis Espinosa-Anke, and Steven Schockaert. 2025. Ragas: Automated evaluation of retrieval augmented generation. *Preprint*, arXiv:2309.15217.

Hao-Lun Hsu, Jing Xu, Nikhil Vichare, Francesco Carbone, Miroslav Pajic, and Giuseppe Carenini. 2025. Depart: A hierarchical multi-agent system for multi-turn interaction.

Tal Ifargan, Lukas Hafner, Maor Kern, Ori Alcalay, and Roy Kishony. 2025. Autonomous llm-driven research — from data to human-verifiable research papers. *NEJM AI*, 2(1):AIoa2400555.

Yang Liu, Dan Iter, Yichong Xu, Shuohang Wang, Ruochen Xu, and Chenguang Zhu. 2023. G-eval: Nlg evaluation using gpt-4 with better human alignment. *Preprint*, arXiv:2303.16634.

Christos Mountzouris, Grigorios Protopsaltis, and John Gialelis. 2026. A graphrag-based question-answering system for explainable and advanced reasoning over air quality insights. *Air*, 4(1).

Changhua Pei, Zexin Wang, Fengrui Liu, Zeyan Li, Yang Liu, Xiao He, Rong Kang, Tieying Zhang, Jianjun Chen, Jianhui Li, Gaogang Xie, and Dan Pei. 2025. Flow-of-action: Sop enhanced llm-based multi-agent system for root cause analysis. In *Companion Proceedings of the ACM on Web Conference 2025*, WWW '25, page 422–431, New York, NY, USA. Association for Computing Machinery.

Ori Ram, Yoav Levine, Itay Dalmedigos, Dor Muhlgay, Amnon Shashua, Kevin Leyton-Brown, and Yoav Shoham. 2023. In-context retrieval-augmented language models. *Transactions of the Association for Computational Linguistics*, 11:1316–1331.

Chaoyi Ruan, Yiliang Wang, Ziji Shi, and Jialin Li. 2025. Reaching agreement among reasoning llm agents. *Preprint*, arXiv:2512.20184.

Philip Schroeder, Nathaniel Morgan, Hongyin Luo, and James Glass. 2025. Thread: Thinking deeper with recursive spawning. *Preprint*, arXiv:2405.17402.

Zhaohui Geoffrey Wang. 2026. Agenttrace: Causal graph tracing for root cause analysis in deployed multi-agent systems. *Preprint*, arXiv:2603.14688.

Junjielong Xu, Qinan Zhang, Zhiqing Zhong, Shilin He, Chaoyun Zhang, Qingwei Lin, Dan Pei, Pinjia He, Dongmei Zhang, and Qi Zhang. 2025. Openrca: Can large language models locate the root cause of software failures? In *The Thirteenth International Conference on Learning Representations*.

Silin Yang, Dong Wang, Haoqi Zheng, and Ruochun Jin. 2024. Timerag: Boosting llm time series forecasting via retrieval-augmented generation. *Preprint*, arXiv:2412.16643.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. 2023. React: Synergizing reasoning and acting in language models. *Preprint*, arXiv:2210.03629.

Xiangrui Zhang, Zeyu Chen, Haining Wang, and Qiang Li. 2025. Llms as firmware experts: A runtime-grown tree-of-agents framework. *Preprint*, arXiv:2511.18438.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric P. Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. 2023. Judging llm-as-a-judge with mt-bench and chatbot arena. *Preprint*, arXiv:2306.05685.

Andrew Zhu, Liam Dugan, and Chris Callison-Burch. 2024. Redel: A toolkit for llm-powered recursive multi-agent systems. *Preprint*, arXiv:2408.02248.

Appendix

A. Hypothesis Generator Prompt

Below is an abridged version of the prompt used by the Hypothesis Generator (§3.3). Use-case-specific fields (in {braces}) are substituted from the configurations in §6; verbose instructions, examples, and JSON-schema detail are elided for space.

```
You are a hypothesis generation specialist for {
      business_context}
analysis. Generate {hypothesis_count} distinct, testable
      hypotheses
that could explain why the given cohort has {
      analysis_objective}.
You have a PySpark query tool; budget {MAX_STEPS} tool calls
      maximum.

INPUT DATA
Granularity (the analysis is scoped to this group; never
      filter on
it): {group}
Cohort Information (definition + stats): {cohort_info}
Column Definitions: {COLUMN_DEFINITIONS}
Additional Defect Context: {context}
Wiki Knowledge Context: {wiki_context}
Domain Guidance: {domain_guidance}

RULES
Before generating, briefly reason about: (1) what makes this
      cohort
unusual based on the stats, (2) which feature combinations are
       most
suspicious, (3) what causal mechanisms could explain the
      anomaly.

{hypothesis_prioritization}

Then apply:
1. Diversity: span the categories listed below (mode collapse
      is the
   dominant failure mode of naive hypothesis generation):
   {hypothesis_categories}
2. Specificity: each hypothesis must name specific features to
   investigate, state expected contrastive behavior, and be
   falsifiable with data.
3. No label leakage (do NOT use post-outcome features):
   {excluded_features}
4. Contrastive framing: each hypothesis must explain why
   {metric_name} is {comparison_phrase} the counterfactual.

{additional_instructions}
{column_warnings}

OUTPUT FORMAT
Return ONLY a JSON array. Each element:
{
  "id": "H1",
  "category": "policy|data_quality|logistics|product|customer|
      pricing",
  "statement": "One-sentence hypothesis",
  "reasoning": "2-3 sentences covering (1) what signal
      triggered
    this hypothesis, (2) the causal mechanism, (3) why
      plausible
    given cohort characteristics.",
  "features_to_investigate": ["feature1", "feature2", ...],
  "expected_pattern": "What data should show if hypothesis is
      true",
  "counterfactual": "What cohort to compare against",
  "estimated_impact": "$ amount or percentage if applicable"
}
```

B. Supervisor Prompt

Below is an abridged version of the supervisor controller's prompt (§3.3). The supervisor runs once per iteration, reads cumulative investigation state, and emits a single-step decision.

```
You are the investigation supervisor coordinating a multi-
      hypothesis
root cause analysis. Decide which hypothesis to investigate
      next
and when to stop.

Analysis Type: {business_context} - investigating {
      analysis_objective}

INVESTIGATION STATE
Iteration: {iteration}/{max_iterations}
Cohort: {cohort_info}
Generated Hypotheses: {hypotheses_summary}
Investigation Results So Far: {investigation_results}

DECISION TASK
(1) Which hypothesis to investigate next?
    Priority scoring: Expected Impact (High=$100K+, Medium=
      $10K-100K,
    Low=<$10K) x Feasibility (are features available in the
      dataset?)
    x Category Diversity bonus (new category preferred).
    Tiebreaker toward unexplored categories; do NOT
      investigate the
    same category twice in a row unless critical; skip
      hypotheses
    subsumed by proven ones.
    If the most recent investigation has confidence 0.4-0.6
      AND
    queries_executed < 3, you may re-select the SAME
      hypothesis
    with a note to explore different features.

(2) Continue or conclude?
    Continue if: high-impact hypotheses remain uninvestigated
      (unless budget is nearly exhausted), OR the top
      hypothesis
      is still UNCERTAIN (confidence 0.4-0.7).
    Conclude if: top 2-3 hypotheses are PROVEN/DISPROVEN with
      confidence > 0.85 AND no remaining uninvestigated
      hypotheses
      have High estimated_impact; OR iteration budget
      exhausted.

OUTPUT FORMAT (JSON only)
{
  "action": "INVESTIGATE_NEXT|CONCLUDE",
  "selected_hypothesis_id": "H1|H2|...|null",
  "reasoning": "1-2 sentence explanation including priority
      rationale",
  "investigation_complete": true|false,
  "remaining_high_priority_hypotheses": 0-5
}
If action is CONCLUDE, set selected_hypothesis_id to null and
investigation_complete to true.
```

C. Use Case Configurations

Each use case is parameterized by a configuration dictionary consumed by the Hypothesis Generator, Investigator, and Narrative Generator prompts. Below are two representative configurations (key fields only; verbose multi-line instructions and category lists are summarized in prose). The remaining six use cases follow the same schema.

C.1 A quality-type use case.

```
use_case_name     : QUALITY_METRIC
tool_domains      : [generic, catalog_product]
group_column      : group_label
metric_name       : anomaly rate
metric_description: elevated anomaly rate
comparison_phrase : higher than
business_context  : <quality-type use case>
analysis_objective: drivers of an elevated rate metric

hypothesis_categories:
  - Policy/Program    (program-driven effects)
  - Data Quality      (miscoded flags, tracking errors)
  - Operations        (handling, routing, facility)
  - Product Issues    (attribute/listing mismatch)
```

```
  - Customer Behavior (anomalous usage patterns)
  - Pricing/Promotion (incentive-driven effects)

excluded_features:
  - outcome_reason
  - resolution_status
  - free_text_comments
  - post-outcome attributes

customer_feedback_column: feedback_topic

column_warnings:
  'item_category' is NOT a column in the dataset; data is
  pre-filtered for granularity.

additional_instructions:
  Inspect feedback_topic for signals on affected items; this
  column is populated only on the outcome event, so do NOT
  compute rates over feedback_topic values.

hypothesis_prioritization:
  If item-level deep-dive context is present, prioritize
  cohort-specific hypotheses first, then broaden to generic
      drivers.

narrative_config:
  mode               : context_only
  template           : QUALITY_METRIC
  recovery_formula: null
  max_agent_steps    : 0
```

C.2 A cost-type use case.

```
use_case_name     : COST_METRIC
tool_domains      : [generic]
group_column      : group_label
metric_name       : cost per unit
metric_description: higher cost per unit
comparison_phrase : higher than
business_context  : <cost-type use case>
analysis_objective: identify drivers of higher per-unit cost
metric_unit       : cost per unit

focus_areas:
  per-unit cost above a reference baseline; high-cost cohorts;
  unexpected premium charges.

hypothesis_categories:
  - Allocation/Placement  (resource placement, routing choices
      )
  - Method Selection      (alternative-method premiums)
  - Measurement/Data Qty  (attribute miscalculation, misclass
      .)
  - Eligibility/Threshold (threshold and routing effects)
  - Service-Level         (premium when standard suffices)
  - Distance/Topology     (path-length and locality effects)
  - Multi-Stage Handoff   (inter-stage transport and mix)

excluded_features:
  - downstream_outcome_date (unless compared to a target)
  - post-outcome complaint signals
  - outcome indicators caused by the cost issue

cost_benchmarks:
  - Method-A vs Method-B : moderate per-unit advantage
  - Alt. providers       : larger per-unit advantage
  - Surcharge effects    : notable aggregate impact
  - Attribute inaccuracy : substantial per-unit effect
  - Allocation optimize  : large total opportunity

narrative_config:
  mode               : context_only
  template           : COST_METRIC
  recovery_formula: null
  max_agent_steps    : 0
```

D. Sample Stakeholder Report

The system's primary expert-facing artifact is a stakeholder report rendered as HTML. To avoid exposing use-case-specific or operational detail, we describe its structure and give a short, fully synthetic and illustrative excerpt rather than reproducing a real report. A report contains: (a) a header with a cohort card, an executive summary, and a top PROVEN hypothesis card with inline `[Q#_H#]` evidence citations; (b) DISPROVEN hypothesis cards with quantitative refutation and confidence scores (rather than silently dropped claims); (c) a conclusion block and an evidence-appendix index listing every query executed; (d) expandable evidence entries with the expert-facing query text and a per-section analytical summary; (e) the underlying PySpark code and raw query-result JSON that the Layer 2 traceability check verifies against; and (f) a business-narrative supplementary block whose aggregates (trailing-twelve-month units, recovery estimates, trends, top entities) are computed by the deterministic downstream pipeline (§3.7), never by the LLM.

Synthetic, illustrative report excerpt

Group: `all_data` **Cohort:** (illustrative cohort)
Projected Impact: (relative, see below) **Confidence:** 91/100 (Grade A)

Executive Summary. Investigation of the cohort revealed a routing/allocation issue in which a premium handling path is applied to items that do not require it `[H1]`. The premium path shows roughly a $5\times$ per-unit cost versus the standard path for items that genuinely need it; more importantly, the majority of premium-path volume consists of lightweight items eligible for the standard path, incurring an order-of-magnitude per-unit premium `[H1]`. The near-zero error rate confirms this is a systematic routing-policy issue, not operational error `[H1]`. A second hypothesis attributing cost to a facility tier was DISPROVEN after the initial claim's figures failed verification `[H2]`.

H1 — Premium-Path Misrouting **Proven** (Confidence: 0.95)
Why this hypothesis. The premium handling path was the highest cost-deviation driver in the cohort; the magnitude suggested possible misclassification of standard items into premium infrastructure.
Finding. The premium path exhibits a dual cost structure: legitimate handling for genuinely heavy items, and systematic misrouting of the majority of its volume (lightweight, standard-eligible items) at a roughly $14\times$ per-unit premium over the standard path `[Q2_H7][Q3_H7]`. The near-zero error rate confirms routing policy rather than operational error `[Q3_H7]`.
Key findings.

- Premium vs. standard path: $\sim 5\times$ per-unit cost for legitimately heavy items `[Q2_H7][Q4_H7]`.
- Majority of premium-path volume is lightweight, standard-eligible items at an $\sim 14\times$ per-unit premium `[Q3_H7]`.
- A scheduled-handling option adds a further per-unit premium on a minority of volume `[Q2_H7]`.

Business impact. A quantified per-unit saving is available by redirecting standard-eligible volume off the premium path (aggregate impact reported only in the deterministic supplementary block, not asserted by the LLM).

H2 — Facility-Tier Inefficiency **Disproven** (Confidence: 0.85)
Why this hypothesis. The initial claim attributed a large premium to a lower-automation facility tier.
Finding. The hypothesis is disproven because the initial claim's figures (volume share and per-unit cost) failed verification against the artifacts `[Q4_H3]`. A smaller, factually corrected premium is retained as a DATAQUALITY-flagged supplementary finding rather than dropped.

Conclusion. The investigation identified one critical, actionable routing-policy issue (premium-path misrouting of standard-eligible volume) and one disproven-but-corrected facility-tier finding. The most actionable result is redirecting standard-eligible volume off the premium path, an immediate per-unit optimization opportunity.

Actionability classifier output.

```
{
  "is_actionable": true,
  "confidence": 0.95,
  "category": "OPERATIONAL_ISSUE",
```

```
  "reasoning": "Routing-policy issue identified with
     high
   confidence: the majority of premium-path volume is
   standard-eligible and incurs a large per-unit
     premium. The
   near-zero error rate confirms a systematic routing
     policy
   rather than operational error. Recommended action:
     audit the
   premium-path routing rules and redirect standard-
     eligible
   volume to the standard path."
}
```

Evidence Appendix (summary). Each Q#_H# citation in the report body links to the full PySpark code, execution status, and per-section summary. A representative entry is a multi-section deep-dive whose sections each return SUCCESS with adequate sample sizes and no data-quality flags. The Layer 2 judge (§3.6) verifies each inline number in the Executive Summary against these Q#_H# payloads before the report is finalized.