Diagnostic Knowledge Graphs

Automated Benchmark Construction and Deterministic Evaluation for Multi-Step Reasoning Agents

Chuci Chen · Zhenyu Zhang · Xinyang Shen · Amazon FBA

KDD 2026
Agentic AI Evaluation Workshop
August 9 · Jeju, Korea

WHAT WE DO We build **Diagnostic Knowledge Graphs (DKGs)** — hierarchical cause trees with frequency priors and per-node SQL detectors — **automatically from noisy resolved tickets**. The same DKG (i) **normalizes gold chains** to a closed vocabulary for reproducible per-step evaluation, and (ii) **constrains an LLM diagnostic agent** to historically-observed causes at inference time.

Applied to Amazon FBA logistics customer support (**14,953 tickets, 284 symptom families**): **σ = 0 inter-run variance**, **92% expert-agreement normalization**, and a **+32pp action-accuracy lift** over the unstructured LLM baseline — **from the same artifact**.

THE EVALUATION PROBLEM

Extreme vocabulary explosion: 5,076 unique cause strings from 2,196 tickets on one symptom, **92% appearing exactly once**. Fuzzy LLM-as-judge scoring is variance-prone (**±2–3pp**) and only moderately faithful to experts (κ = 0.60).

ONE STRUCTURE, TWO ROLES

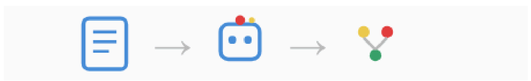
Benchmark construction and agent design are **dual problems solved by the same artifact**. **Benchmark side —** canonical vocabulary → **deterministic exact-match**, σ = 0. **Agent side —** constrained hypothesis space → **+32pp action lift**.

Five-stage pipeline: from noisy tickets to a data-grounded diagnostic tree

Precision-first at every stage: frequency filter → SQL verification → one-sided detectors

① Chain Extraction

One LLM call per resolved ticket extracts a structured *symptom → multi-level causes → action* chain at temperature 0.



Example stuck in receiving → warehouse backlog → peak demand → manual verification → *advise monitor*

Constraints: causes must describe *mechanisms*, not outcomes; case-specific IDs stripped; actions kept *reusable*.
Failure mode: 8% of tickets discarded when no valid chain is extractable — a hard noise floor for downstream evaluation.

1,292 build tickets → 1,193 chains (92% success) · depth 2–4

② DKG Construction & Normalization

Cluster symptoms via embeddings (τ_s = 0.85), then two-pass BERTopic + LLM merge compresses raw cause strings into canonical clusters.

Stuck in receiving (n=988)
Warehouse processing backlog
Peak season demand → Manual verification
High concurrent shipments · Multi-step placement
Compliance hold → Labeling · Restricted product

LLM merge criterion: two clusters merge *only if* they describe the same cause *and* resolve to the same action — catches semantic duplicates with near-zero token overlap.
Scale: 5,076 raw strings → 73 canonical clusters (70× compression, 97% coverage); pipeline extends to **284 families from 14,953 tickets**.

③ SQL Grounding

For each node, an iterative loop (K = 7 max) synthesizes and tests a parameterized SQL template against 30 stratified ground-truth tickets.


```
-- Node: "warehouse receiving backlog"
closed_ts IS NULL OR closed_ts > {ref_date}
```

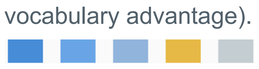
Validation: 70–83% on 30 tickets/node

Two thresholds: break at 0.8 (stops overfitting), verify at 0.7 (label-noise δ ≈ 0.08 → true acc ≥ 0.62).
Ticket as test case: resolved text is both the label *and* the operational record — closed-loop validation without extra labeling.

35 / 55 nodes verified (≥ 70%) · unverified retained, deferred to children

④ Agent Evaluation

Precision-first traversal: execute SQL at every node top-down; keep TRUE matches; a single LLM synthesizes the derived chain from verified evidence. All five conditions emit top-3 candidates, scored by the same semantic judge against *raw* gold chains (no vocabulary advantage).



M1 diagnostic (%): +SQL **74**, +AdHoc 70, +Tree 70, RAG 60, LLM 63
M2 action (%): +SQL **73**, +AdHoc 69, +Tree 66, RAG 43, LLM 41
Qty discrepancy M2: +SQL **55**, +Tree 49, RAG 26, LLM 11

Gold receiving → backlog → peak season → dock congestion
Derived receiving → backlog (29d) → dock congestion (29d) · 3/4 levels
Action monitor warehouse queue · **correct**
Why it works siblings evaluated independently — a single false negative can't derail routing.

n = 100 · tree structure + priors drive 78–86% of the +32–44pp lift over LLM-only

⑤ Augmentation

New tickets with unknown causes are collected; an LLM triage decides genuinely-new vs. paraphrase (≤ 3 new nodes per family per round).

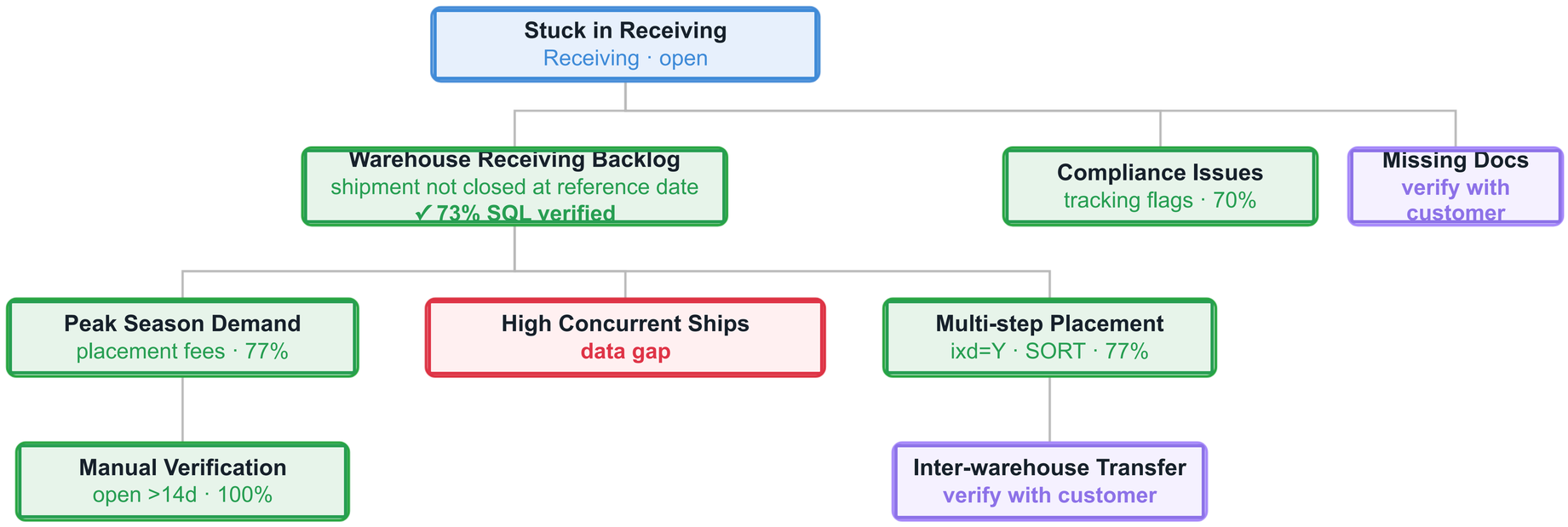
+NEW dock unloading delay
+NEW carrier/warehouse sync lag
+NEW transfer blocks receiving
SKIP "warehouse queue backlog" · "warehouse delay"

Coverage convergence: $C_{t+1} = C_t + (1 - C_t) \cdot \alpha_t$. Empirically $C_0 = 0.97$, $\alpha_1 \approx 1.0$ (all 3 novel causes captured in one round).

3 added · 12 skipped (80% filter rate) · loops back to Stages ② + ③

What the agent sees: an executable cause hierarchy for "stuck in receiving"

SQL verified (≥ 70%) · Data gap · Ask customer



Nine-box receiving-family view: 1 symptom root + 8 cause nodes across 3 levels. Full v2 taxonomy has 14 nodes; 8 SQL-verified at 70–83% on 30 tickets each.

Precision-first design

Every node is a **one-sided detector** (TRUE = active; FALSE discarded, never routes). All siblings are evaluated independently — **a single false negative cannot derail the diagnosis**.
Cost: broader candidate sets. Benefit: **no unrecoverable errors**. Verification thresholds break at 0.8 to prevent overfitting; verify at 0.7 to absorb 8% extraction noise.

Error decomposition (n = 16 failures, receiving family)

12 / 16	2 / 16	2 / 16
SQL misdirection — parent fires with true child (recoverable)	*Tree gap* — cause absent from DKG (Stage 5 fix)	*Ticket misclassification* — noise floor from 8% extraction error

Loss decomposes as $1 - \text{DiagAcc}(x) \geq \varepsilon_{\text{struct}} + \varepsilon_{\text{sql}} + \varepsilon_{\text{eval}}$ — each targetable by a specific gate.

Boundary condition

Effectiveness depends on **frequency concentration**. Skewed distributions (receiving) yield **+32pp**; near-uniform distributions (transit, n = 65) only **+5pp** — **predictable** from the canonical vocabulary's rank-frequency curve.

Evidence: reproducible benchmarks improve agents — from the same artifact

Normalize the long tail

5,076 → 73 canonical clusters on the largest family (70× compression)

97% coverage · **92% expert-validated** normalization accuracy (n = 50)

Scales to **284 symptom families from 14,953 tickets** — no manual curation.

Make evaluation trustworthy

92% vs. 80% expert agreement: normalization beats the LLM judge it replaces

σ = 0 inter-run variance under deterministic exact-match (vs. ±2–3pp for the judge)

First protocol that makes **sub-3pp longitudinal drift** reliably detectable.

Improve the agent

73% vs. 41% M2 action accuracy: DKG+SQL vs. LLM-only (receiving, n = 100)

+44pp on quantity discrepancy (55% vs. 11%)

- **Tree + priors alone** account for **78–86%** of the observed action-accuracy lift — no SQL required.
- **SQL adds auditable evidence** and deployment-time reliability; the *structure* drives the diagnostic-accuracy gain.
- **Generalizes** to any domain with resolved cases + operational data: IT incidents, medical triage, fraud, defect analysis.

chuciche@amazon.com · zhenyuzh@amazon.com · xinyans@amazon.com