

Knsight: Self-Evolving Agentic System for Production Cloud Root Cause Analysis

Bo Lyu^{*†¶} Hongge He^{*} Xueqi Wang

Kuaishou Technology, Beijing, China

boironic@gmail.com

Abstract

Cloud incident response increasingly relies on LLM agents that translate heterogeneous telemetry and natural-language alerts into operator-facing diagnoses. However, in production RCA workflows, LLM agents commonly serialize these structured signals into text, losing the temporal and topological structure that is critical for diagnosis. We present Knsight, an industry-deployed RCA agent that converts operational telemetry into chart-image canvases and uses a frozen multimodal LLM to reason directly over visual evidence. Knsight introduces Vision Recursive-of-Thought (V-RoT), an iterative diagnose-review-refine loop in which a Reviewer module inspects each canvas and decides whether to refine, zoom in, or stop. To adapt across recurring production failures, Knsight further converts Reviewer feedback into a shared, alias-substituted skill library used at inference time. Offline, on the OpenRCA benchmark, Knsight improves aggregate Correct by 7.0 points over the strongest text-only baseline and reduces timeouts from 117 to 11. Online, in a 90-day deployment across 1,284 production incidents, Knsight increases top-1 acceptance from 38.9% to 71.4% and reduces time-to-acknowledge and time-to-resolve by 59% and 32%, respectively.

1 Introduction

Modern cloud services emit incidents across heterogeneous telemetry—KPI time series, dependency traces, service logs, and a natural-language alert—and a diagnoser must infer when a fault began, which component is responsible, and why the symptoms support that conclusion. Recent LLM-based RCA agents (Yao et al., 2023b; Xu et al., 2025; Luo et al., 2025) retrieve telemetry on demand, but the dominant design is single-agent and text-native: tables, topology, and onset patterns are

repeatedly textified, forcing one controller to reconstruct multimodal evidence from prose (Kong et al., 2024). This textification drives three recurring failure modes on OpenRCA (Xu et al., 2025)—evidence compression loss, temporal anchoring, and prior collapse (§2)—that prompting does not remove, because the evidence is already gone before the model reasons over it.

We present Knsight, a multimodal RCA system deployed in production on a major cloud provider’s SRE platform (Figure 1). At its core, Knsight contributes a multimodal LLM reasoning interface for structured operational telemetry: it lets a frozen multimodal LLM reason directly over metric, trace, and log streams rendered as chart-image canvases, recovering the temporal and topological structure that textification and linear analysis strip away before an LLM agent reasons. Vision Recursive-of-Thought (V-RoT) renders each step’s telemetry into a canvas and recurses over it under a separate Reviewer that issues STOP/REFINE/ZOOM, replacing linear text-native analysis with revisitable visual reasoning. Skill self-evolution retrieves confirmed-diagnosis trajectories from a shared library to condition new incidents in context, with no weight update.

Our contributions are:

- We introduce the chart-image canvas as the unit of intermediate state for LLM reasoning over structured operational telemetry, with deterministic hash-pinned rendering that makes every diagnosis replayable for post-incident audit.
- We propose V-RoT, a Reviewer-gated recursion over canvases whose monotone-shrinkage action space the model provably cannot violate.
- We propose skill self-evolution, a multi-user-shared canvas/skill library mined from on-call-confirmed traces and retrieved at inference, with drift-based retirement and no weight update.

^{*}Equal contribution.

[†]Corresponding author.

[¶]Project lead.

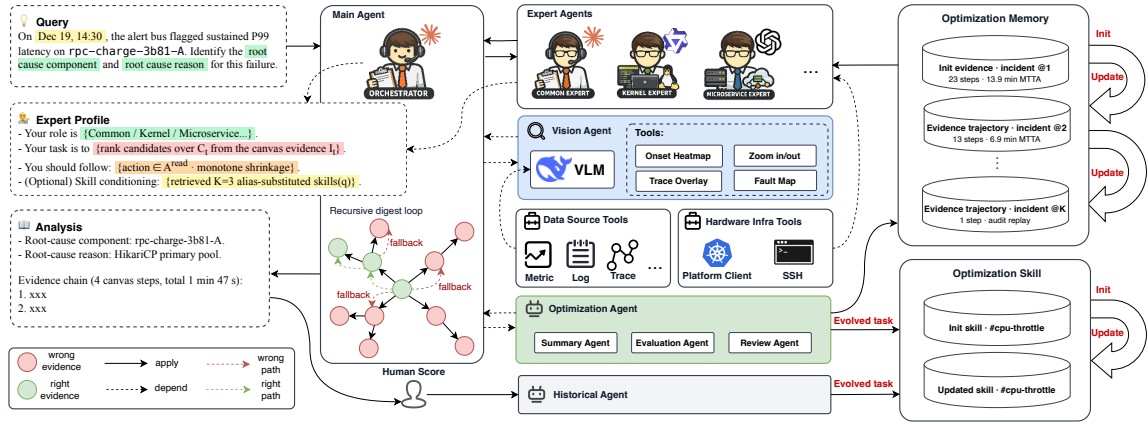


Figure 1: Knsight pipeline. The Supervisor runs V-RoT over chart-image canvases produced by a deterministic Visioner, while a separate Reviewer issues REFINE/ZOOM/STOP (orange) and dispatches domain Experts. Confirmed diagnoses feed a shared skill library that conditions future incidents without any weight update (§3.4).

- We evaluate Knsight on a public RCA benchmark and in a 90-day production deployment, where on-call engineers accept its diagnoses as auditable decision support.

On a public RCA benchmark and in a 90-day production deployment, Knsight reliably outperforms strong text-only agents and is accepted by on-call engineers as auditable decision support, showing that the right visual interface lets a frozen multi-modal LLM reason over operational telemetry that defeats text-native agents.

2 Related Work

Agentic systems for SRE. Recent generative-AI approaches to site reliability engineering automate detection and diagnosis: OpenRCA (Xu et al., 2025) benchmarks LLMs on multimodal RCA, OpsAgent (Luo et al., 2025) adds PPO + reflection-based self-evolution, and STRATUS (Chen et al., 2025) orchestrates detection / diagnosis / mitigation agents in a state machine with a Transactional No-Regression safety property. These systems remain text-native at each reasoning step, repeatedly summarizing telemetry into prose and losing temporal shape, distribution structure, and cross-series alignment. Knsight instead makes the chart-image canvas the shared unit of intermediate state, and its deterministic, hash-pinned canvases give bit-identical post-incident replay for audit (§3.3).

Multi-agent orchestration and RL. Orchestrator-worker LLM collectives—MetaGPT (Hong et al., 2024), AutoGen (Wu et al., 2024), and GRPO-family extensions (Dang et al., 2025; Li et al., 2026; Hong et al., 2025; Feng et al., 2026)—learn agent topology or train

under task-agnostic reward without crediting the shape of the recursion trace, and Tran and Kiela (2026) further show single-agent LLMs can match multi-agent systems under equal token budgets. Complementary efficiency work compresses intermediate reasoning through deterministic visual supervision (Lv et al., 2026) or prunes KV-cache channels to reduce inference memory and raise throughput (Lv et al., 2025). We position Knsight not as a model-compression or multi-hop-reasoning optimizer but as a production-readiness solution: the multi-agent decomposition is the price of deterministic audit, bounded-token Reviewer decisions, and shared-skill amortization across an SRE organization. Knsight fixes a small action space over deterministic canvases that the heuristic Supervisor operates today, leaving canvas-shape-aware offline policy optimization to future work (Appendix A).

Self-evolution via memory and skills. Lifelong-learning agents accumulate experience into external structures—executable skill libraries (Voyager (Wang et al., 2023)), episodic reflections (Reflexion (Shinn et al., 2023)), mined insights (ExpeL (Zhao et al., 2024)), induced workflows (AWM (Wang et al., 2025)), and RL-trained memory (MemAgent (Yu et al., 2026))—but these are single-user and either co-evolve memory with model weights or mine skills offline without an audit guarantee. Knsight occupies a different point: a multi-user-shared canvas store with alias-substituted skills mined from on-call-confirmed traces, retrieved at inference without any weight update and drift-retired by rolling success rate. The submitted system is frozen and retrieval-only; the

same store would also seed offline policy optimization (§A), but no reported result depends on it.

Why text-native RCA fails. The gap these systems leave is concrete. On OpenRCA (Xu et al., 2025), text-native ReAct agents under default prompting exhibit three recurring pathologies that motivate Knsight’s design. (P1) Evidence compression loss: tool outputs are summarized to a median 12% of their token count, so downstream reasoning operates on a coarsened abstraction. (P2) Temporal anchoring: onsets are over-predicted in the first window quintile by $2.8\times$ relative to the gold distribution. (P3) Prior collapse: predictions concentrate on a few component types while the gold spans far more. Knsight counters each by rendering structured canvases that preserve evidence (P1), expose the full-window onset distribution before commitment (P2), and force explicit hypothesis enumeration through its Reviewer and Experts (P3).

3 Methodology

3.1 Problem Setting

We define multimodal RCA as an automated task on the following environment. A target cloud system is characterized by an incident query q , three telemetry streams $\mathcal{D} = (\mathcal{D}_m, \mathcal{D}_t, \mathcal{D}_l)$ (metric / trace / log), a set of candidate components C^{all} , and a time window $[l^{\text{all}}, r^{\text{all}}]$ surrounding the alert. The goal is to emit one or more root-cause tuples $\hat{y} = (\hat{t}, \hat{c}, \hat{r})$ specifying onset time, responsible component, and root-cause reason. All evaluation uses OpenRCA’s official harness (Xu et al., 2025).

Knsight organizes five reasoning roles and a Visioner into a Reviewer-gated agentic workflow that mirrors production SRE diagnosis. We use $\alpha_S, \alpha_{V_{is}}, \alpha_{R_{ev}}, \alpha_{X_c}, \alpha_{X_k}, \alpha_{X_m}$ for the roles defined below. Following the SRE-task decomposition of Chen et al. (2025), the V-RoT cycle realizes detection, localization, and reason verification, while explicit failure mitigation is delegated to the on-call operator (autonomous mitigation is out of scope; cf. §4).

Action space. Each step the Supervisor emits a structured action $a_t = (s_t, \kappa_t, M_t, C_t, [l_t, r_t], \gamma_t, b_t)$ over stage, canvas type ($\kappa_t \in \{\text{HEATMAP}, \text{FAULT-MAP}, \text{TRACE-OVERLAY}, \text{EVIDENCE-PANEL}, \text{COMPARISON}\}$), data streams, candidate set, time window, zoom target, and stop flag. Two monotone invariants

hold: $C_{t+1} \subseteq C_t$ and $[l_{t+1}, r_{t+1}] \subseteq [l_t, r_t]$. We partition by mutability into A^{read} (telemetry fetches, canvas renders—used by all LLM roles) and A^{write} (mitigations); Knsight restricts itself to A^{read} at inference, with A^{write} enacted only by the on-call operator. This read-only restriction by construction rules out the regression risk that motivates Transactional No-Regression (Chen et al., 2025).

3.2 Multi-Agent Architecture

The five roles communicate by strict-JSON messages and share a diagnosis state $h_t = (\text{stage}, C_t, [l_t, r_t], \text{hypotheses})$ plus the multi-user canvas store Store.

Supervisor (α_S). The only agent the user talks to. It runs the V-RoT loop (§3.3), advances h_t across five stages, emits a_t , and is the only role that may call sub-agents. In the reported system the Supervisor is a frozen multimodal LLM (Claude-3.5-Sonnet via vision API) conditioned on retrieved skills (§3.4); the action space admits a learned policy with the same I/O contract, which we leave to future work (§A).

Visioner ($\alpha_{V_{is}}$). Given the Supervisor action a_t and fetched evidence $\mathcal{E}_t$ (metrics, logs, traces, and topology), the Visioner emits a paired canvas $I_t = (\text{PNG}_t, \text{JSON}_t)$ under fixed axes and aggregation rules. Each canvas type targets one of the three text-native pathologies (§2): HEATMAP exposes full-window onset distribution before commitment (P2 anchoring); FAULT_MAP preserves topology and propagation direction (P1 compression); COMPARISON forces hypothesis enumeration (P3 collapse). The action log stores a_t and the hash-pinned $\mathcal{E}_t$, allowing the same evidence view to be reconstructed for post-incident audit.

Reviewer ($\alpha_{R_{ev}}$). A separate vision-API call that ingests I_t and emits decision $d_t \in \{\text{STOP}, \text{REFINE}, \text{ZOOM}\}$. Decoupling Reviewer from Supervisor avoids context pollution that mixes planning chain-of-thought with evidence-grounded verification: the verifier’s vote cannot be silently absorbed into Supervisor planning. Decision rules: STOP when $|C_t| \leq 1$ or budget exhausted; REFINE when top-1 vs. median anomaly density $\geq 3:1$; ZOOM otherwise.

Experts ($\alpha_{X_c}, \alpha_{X_k}, \alpha_{X_m}$). When the diagnosis narrows to a candidate component, the Supervisor

dispatches to one of three domain Experts. Common $\mathcal{X}_c$ is the safe default. Kernel $\mathcal{X}_k$ specializes in OS / hardware signals (ECC, CFS-throttling, NUMA, dmesg)—representative task: GPU ECC pre-flight (Appendix I.3). Microservice $\mathcal{X}_m$ specializes in distributed-trace signals (P99 wavefronts, connection-pool saturation, 5xx ratios)—representative task: HikariCP pool exhaustion (Appendix I.2). Dispatch is by a strict-JSON router rule on the query and Stage-1 onset heatmap; a wrong route is recoverable via Expert-emitted `defer`.

Recursive agent loop. Every LLM-driven role is itself a recursive loop: given a sub-problem x' , role ρ executes $h_{k+1}^\rho = \mathcal{F}_\rho(h_k^\rho, \text{retrieve}(x'), \text{call}(\rho'))$ and returns a structured summary to its caller. Crucially, the loop is depth-first rather than linear. When the Reviewer rejects a leaf, the Supervisor does not abandon the diagnosis; it backtracks to the nearest unexplored sibling hypothesis and re-enters the loop on fresh canvas evidence—the orange fall-back edges in Figure 2. This is what separates V-RoT from a single ReAct chain, which commits to one path and fails outright when that path dead-ends (Figure 2a), and from fixed-depth ToT (Yao et al., 2023a), which cannot return to a branch once it has been pruned. Recursion also nests across roles: an Expert dispatched on REFINES may open its own V-RoT cycle inside the Supervisor’s, so a microservice hypothesis can be pursued to trace-level depth without expanding the Supervisor’s frontier. Hard budgets bound the nested control flow ($T_{\max}^\rho=8$ per Expert, $T_{\max}^S=12$ for the Supervisor), and only the Supervisor may open one Expert loop per iteration. Thus the recursion depth is at most two and the worst case is $12 + 12 \times 8 = 108$ role iterations; equivalently, $\text{Cost}(x) \leq 12c_S + 96c_X$, where c_S aggregates one outer Supervisor–Reviewer iteration and c_X one Expert iteration. The monotone projection of §3.1 prevents candidate-set or time-window expansion, while the hard budgets guarantee termination even if a model repeatedly requests refinement. Algorithm 1 states the full Supervisor loop.

3.3 Vision Recursive-of-Thought

The Supervisor’s loop operates on canvas states. Let $\text{skills}(q)$ be the top- K retrieved skills (§3.4).

Algorithm 1 Vision Recursive-of-Thought

Require: q , telemetry $\mathcal{D}$, skills S , budget $T_{\max}$

- 1: init $h_0, C_0 \leftarrow C^{\text{all}}$, window $[l_0, r_0]$
- 2: **for** $t = 0$ **to** $T_{\max} - 1$ **do**
- 3: $a_t \leftarrow \alpha_S(q, I_{<t}, S)$ ▷ plan + act
- 4: project a_t onto monotone space
- 5: $\mathcal{E}_t \leftarrow \mathcal{Q}(a_t, \mathcal{D})$ ▷ fetch evidence
- 6: $I_t \leftarrow \alpha_{V_{is}}(a_t, \mathcal{E}_t)$ ▷ render canvas
- 7: $d_t \leftarrow \alpha_{R_{ev}}(h_t, I_t)$ ▷ Reviewer
- 8: **if** $d_t = \text{STOP}$ **then**
- 9: **break**
- 10: **else if** $d_t = \text{REFINE}$ **then**
- 11: $h_{t+1} \leftarrow \text{UPDATE}(h_t, \alpha_X)$ ▷ Expert
- 12: **else**
- 13: $h_{t+1} \leftarrow \text{SHRINK}(h_t)$ ▷ zoom
- 14: **end if**
- 15: **end for**
- 16: **return** $\hat{y}$ from h_t

One V-RoT step is:

$$\begin{aligned}
 h_t &= \alpha_{S, \text{plan}}(q, I_{<t}, \mathcal{E}_{<t}, \text{skills}(q)), \\
 a_t &= \alpha_{S, \text{act}}(q, h_t, I_{<t}), \quad \mathcal{E}_t = \mathcal{Q}(a_t, \mathcal{D}), \\
 I_t &= \alpha_{V_{is}}(a_t, \mathcal{E}_t), \quad d_t = \alpha_{R_{ev}}(h_t, I_t), \\
 h_{t+1} &= \text{UPDATE}(h_t, d_t, \text{call}(\alpha_{X_*})).
 \end{aligned}$$

$\text{call}(\alpha_{X_*})$ is the optional Expert dispatch on REFINES, and Algorithm 1 unrolls the loop step by step. Action validity (the two monotone invariants of §3.1) is enforced post-hoc by deterministic projection, so no LLM emission can violate the structured action space. Unlike ToT (Yao et al., 2023a)—which expands pre-defined branches but cannot backtrack—V-RoT’s transitions are driven by canvas content, allowing the policy to revisit windows and components without re-querying telemetry; Figure 2 contrasts the resulting reasoning shape against linear ReAct on a representative incident.

The heuristic Supervisor uses robust-z anomaly scores $A(c, t) = \max_k |z_{c,k}(t)|$ with median and MAD over a 24h baseline; PRUNE retains the top- $k = 10$ at the prune stage and top-3 at verify; SHRINK contracts $[l_t, r_t]$ to $[t^* - 5\text{min}, t^* + 5\text{min}]$.

3.4 Skill Self-Evolution

A single-agent RCA pipeline discards each diagnosis after it ends; Knsight instead treats the canvas store as a multi-user-shared skill library so that confirmed diagnoses improve future ones, with the model frozen (Figure 1, right). After the Supervisor emits STOP and the on-call confirms the answer, the trajectory is written to memory and, if it survives an alias-substituted replay on its own incident, distilled into a reusable skill: a symptom condition, the canvas action/decision sequence $[a_1, \dots, a_T]$ / $[d_1, \dots, d_T]$, an alias map, and rolling statistics.

Table 1: RCA accuracy on OpenRCA (Xu et al., 2025) and our internal CloudRCA deployment set (1,284 incidents); the skill library is empty on OpenRCA but active on CloudRCA (§3.4). Correct = exact-match, Partial $\geq$ one subtask correct, TO = timeout count (— = n/a); Correct cells are mean $\pm$ std over 3 runs, and Aggregate is their OpenRCA sample-weighted mean.

Method	Telecom		Bank		Mkt-1		Mkt-2		CloudRCA		Aggregate	
	Corr	Part	Corr	Part	Corr	Part	Corr	Part	Corr	Part	Corr	TO
CoT	10.0 $\pm$ 1.8	15.0	14.7 $\pm$ 1.1	19.1	11.4 $\pm$ 2.0	16.3	10.3 $\pm$ 1.9	15.4	12.0 $\pm$ 0.7	17.0	12.2 $\pm$ 0.4	—
Few-shot	11.8 $\pm$ 1.9	18.0	18.4 $\pm$ 1.2	24.3	14.3 $\pm$ 2.1	21.4	12.8 $\pm$ 1.9	19.2	14.5 $\pm$ 0.8	21.0	15.0 $\pm$ 0.5	—
ART	5.9 $\pm$ 1.2	19.6	3.7 $\pm$ 1.0	13.2	2.9 $\pm$ 0.9	22.9	1.3 $\pm$ 0.6	19.2	1.5 $\pm$ 0.4	16.0	3.3 $\pm$ 0.4	—
Reflexion	15.7 $\pm$ 2.0	28.4	18.4 $\pm$ 1.3	30.1	14.3 $\pm$ 2.2	25.7	11.5 $\pm$ 1.8	22.4	13.5 $\pm$ 0.8	26.0	14.7 $\pm$ 0.6	93
RCA-Agent	19.6 $\pm$ 2.2	35.3	16.9 $\pm$ 1.3	35.3	5.7 $\pm$ 1.5	30.0	3.8 $\pm$ 1.0	27.6	8.0 $\pm$ 0.7	30.0	11.5 $\pm$ 0.7	85
Codex CLI (GPT-4o)	13.7 $\pm$ 1.9	28.4	18.4 $\pm$ 1.3	32.4	14.3 $\pm$ 2.1	27.1	10.3 $\pm$ 1.8	22.4	13.0 $\pm$ 0.8	26.0	14.2 $\pm$ 0.6	88
Claude Code	17.6 $\pm$ 2.1	31.4	22.1 $\pm$ 1.4	37.5	17.1 $\pm$ 2.3	30.0	12.8 $\pm$ 1.9	24.4	16.5 $\pm$ 0.9	31.0	17.4 $\pm$ 0.5	71
ReAct + Qwen2.5	5.9 $\pm$ 1.2	13.7	8.1 $\pm$ 1.0	18.4	2.9 $\pm$ 0.9	18.6	1.3 $\pm$ 0.6	17.9	2.5 $\pm$ 0.5	16.0	4.6 $\pm$ 0.4	6
ReAct + Claude-3.5	21.6 $\pm$ 2.2	36.3	25.7 $\pm$ 1.5	40.4	21.4 $\pm$ 2.4	34.3	15.4 $\pm$ 2.0	29.5	19.5 $\pm$ 0.9	33.0	21.5 $\pm$ 0.7	117
OpsAgent	30.0 $\pm$ 2.4	45.0	18.5 $\pm$ 1.3	40.7	10.2 $\pm$ 1.9	40.7	10.2 $\pm$ 1.8	40.7	17.0 $\pm$ 0.9	38.0	16.5 $\pm$ 0.5	—
Knsight	32.5 $\pm$2.5	50.0	31.6 $\pm$2.3	46.3	27.1 $\pm$3.5	40.0	23.1 $\pm$3.2	35.9	60.0 $\pm$2.8	74.0	28.5 $\pm$0.6	11

Algorithm 2 Skill self-evolution

Require: incident x , query q ; store Store; budget K, N
1: $S \leftarrow \text{RETRIEVE}(\text{key}(q), \text{Store}, K)$ $\triangleright$ top- K
2: $\hat{y} \leftarrow \text{V-RoT}(x \mid S)$ $\triangleright$ no weight update
3: **if** on-call confirms $\hat{y}$ **then**
4: $\tau \leftarrow$ alias-substitute trajectory of x
5: **if** $\text{REPLAY}(\tau)$ solves x **then**
6: commit $\langle \text{cond}, \tau, \text{stats} \rangle$ to Store
7: **end if**
8: **end if**
9: update $s \in S$ success over N ; retire if $< 50\%$

Concrete identifiers—service names, hosts, request IDs—are replaced by stable hash-derived aliases, so a skill mined from one tenant transfers to others without leaking data (Algorithm 2).

Retrieve, condition, retire. Before V-RoT enters Stage 1 the Supervisor retrieves the top- $K=3$ skills whose symptom key matches the incident—Jaccard ≥ 0.6 over canvas-hash bags or cosine ≥ 0.85 over symptom-key embeddings—and conditions its prompt on their action sequences and success rates; the online path performs no weight update. A rolling-success drift detector over an $N=500$ window auto-retires any skill below 50% (Wang et al., 2023; Zhao et al., 2024), so the library is self-pruning and its drift is observable. For OpenRCA evaluation the library is empty, so no skill is shared between training and test; over a 90-day pilot it settled at 482 active and 47 retired skills, with a 22.7% retrieval hit-rate and 81.9% on-hit top-1 acceptance.

4 Experiments

Setup and baselines. We use the full OpenRCA benchmark (Xu et al., 2025): Telecom ($n=51$),

Bank ($n=136$), Mkt-1 ($n=70$), Mkt-2 ($n=78$), plus our internal CloudRCA dataset ($n=1,284$). Correct and Partial follow Xu et al. (2025); significance is McNemar’s χ^2 vs. the strongest text-only baseline, with 95% bootstrap CIs (10,000 paired resamples) computed per cell. All LLM-agent baselines share the same tool set and budget; ReAct + Claude-3.5 hits the same vision endpoint as Knsight with image input disabled. Reported numbers come from the heuristic Supervisor over the frozen backbone; $\text{skills}(q) = \emptyset$ on OpenRCA (§3.4). Baselines span prompting (CoT, few-shot), RCA-specialized (ART (Sun et al., 2024), ReAct (Yao et al., 2023b) + {Claude-3.5, Qwen2.5-72B}, Reflexion (Shinn et al., 2023), RCA-Agent (Xu et al., 2025), OpsAgent (Luo et al., 2025)), and off-the-shelf-agent stress baselines (Codex CLI (OpenAI, 2025), Claude Code (Anthropic, 2025)). Tool signatures, prompts, and exact agent configurations: Appendix J.

Main results. Knsight improves over every baseline on every split (Table 1): +7.0 aggregate Correct over the strongest text-only baseline, ReAct + Claude-3.5, ranging from +5.7 on Mkt-1 to +10.9 on Telecom. Two effects separate it from the text-native agents. First, rendering telemetry as a canvas preserves the temporal and topological structure that prose discards (P1, §2); the gain therefore holds against the strongest text agent on the same backbone rather than shrinking into noise, and the off-the-shelf coding agents (Codex CLI, Claude Code), which never see the signal as an image, trail by 11–14 points despite strong general tool use. Second, the Reviewer-gated recursion collapses

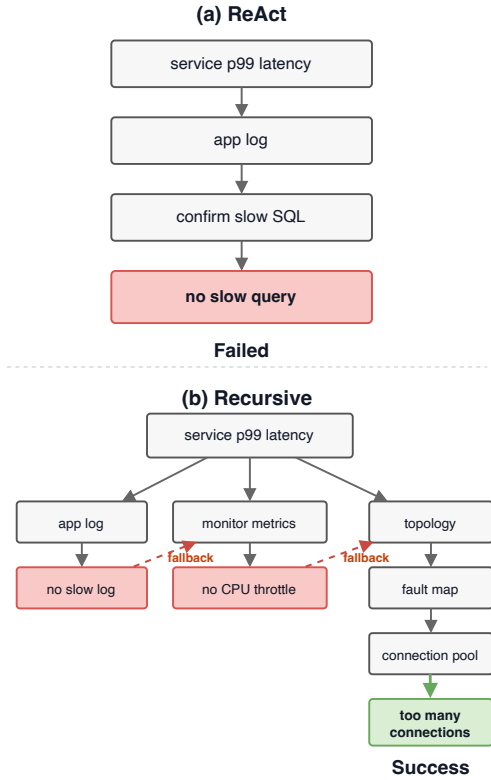


Figure 2: Reasoning-shape comparison on a latency-spike incident. (a) Linear ReAct commits to one chain and cannot recover once it dead-ends at no slow query. (b) Knsight’s V-RoT branches the alert across logs, metrics, and topology, falling back on rejected hypotheses until it reaches the gold root cause (connection-pool saturation).

timeouts from 117 to 11: it halts exploration once the candidate set is settled instead of letting the controller wander until the step budget runs out, and the mean candidate set shrinks from 18.3 to 1.2 before a hypothesis is emitted—which is what makes the output reviewable by an on-call engineer. On CloudRCA, where the shared skill library is active (it is empty on OpenRCA), the lead widens to ~ 40 points, indicating that in-context retrieval of confirmed trajectories compounds the canvas gains once a library has accumulated. The matched-backbone and skill-store controls in Tables 2 and 3 isolate these two effects; Appendix H gives the controlled backbone-replacement protocol.

Ablation. Canvas-as-image is the largest single contributor: PNG-only beats JSON-only by 3.7 / 4.5 Correct on Bank / Mkt-2, with the residual ~ 1.4 to the full system coming from JSON fea-

Table 2: Matched-backbone Correct accuracy (%).

Dataset	Backbone	ReAct	Knsight	Δ
OpenRCA	Qwen2.5-VL-32B	19.5	25.8	+6.3
	Claude-3.5-Sonnet	21.5	28.5	+7.0
	GPT-4o	19.6	27.7	+8.1
CloudRCA	Qwen2.5-VL-32B	17.7	54.3	+36.6
	Claude-3.5-Sonnet	19.5	60.0	+40.5
	GPT-4o	18.2	58.4	+40.2

Table 3: Shared-skill ablation on CloudRCA (%).

Method	Correct	Partial
Knsight	60.0	74.0
w/o shared skills	55.9	67.3

tures that disambiguate near-tied candidates. The rendered channel jointly exposes shared-axis onset alignment, cross-series shapes (spikes, steps, drift, and periodicity), spatial anomaly density, and fault-map propagation direction; the matched PNG-only / JSON-only comparison isolates the combined diagnostic value of these visual cues while keeping the underlying telemetry fixed. This ranking holds on both evaluated splits. Their combination yields the highest Correct score in each. This ordering is the central evidence for our claim—the model gains more from seeing the telemetry than from reading the identical signal as structured text, even though the JSON channel carries the same numbers—and it is what distinguishes Knsight from a structured-tool-use agent. Removing the Reviewer raises Bank timeouts $4 \rightarrow 21$ (-4.5 Correct), confirming that the recursion needs an external gate: left to itself the Supervisor cannot reliably decide when to stop. A rule-based Reviewer recovers most of this on Mkt-2 (22.2 vs. 23.1) but lags on Bank (29.8 vs. 31.6), where the stop/refine call hinges on visual evidence a fixed density threshold cannot read. Expert dispatch helps the microservice-heavy Mkt-2 (-2.1 on removal) more than Bank (-0.7), and removing shared skills reduces CloudRCA Correct from 60.0 to 55.9 and Partial from 74.0 to 67.3 (Table 3); the full component ablation is in Appendix F.

Failure modes on OpenRCA. A 100-case audit of wrong/partial Bank predictions decomposes into canvas-perception (28%), Reviewer wrong-STOP (24%), Expert mis-dispatch (17%), and strict exact-match mismatch (31%, predicted component is direct upstream/downstream of gold—operationally close but wrong under EM); full taxonomy and per-bucket fixes in Appendix G.

Industrial pilot. CloudRCA is our 90-day production deployment of Knsight on a major cloud SRE platform ($\sim 10^5$ machines, $\sim 10^9$ events/day) in step-confirm mode under HITL boundaries (§5): the same 1,284 stratified incidents are scored offline by Correct/Partial in Table 1 and, in production, by the operational metrics below—since what matters in deployment is whether the on-call accepts a diagnosis and how fast incidents are handled, not exact-match alone. The platform’s ReAct-based diagnoser was invoked concurrently on every incident; postmortems were written by a separate team (paired operational measurement, not a blind A/B). This holds incident mix fixed across systems. Separate postmortems anchor wrong-root-cause labels. Metric definitions and full breakdowns: Appx. C, B. With 95% bootstrap CIs: top-1 acceptance 71.4% [69.0, 73.7] vs. 38.9% (Wilcoxon paired $p < 10^{-6}$), top-3 88.2%, MTTA 4.6 vs. 11.3 min (−59%), MTTR −32%, and wrong-root-cause 7.0% [5.6, 8.5]. Reviewers used the displayed canvases to scrutinize each hypothesis during step confirmation. Their 83% catch rate quantifies the operational value of this evidence trail.

5 Conclusion

Knsight reframes root-cause analysis as multimodal LLM reasoning: rather than textify operational telemetry into prose, it renders metric, trace, and log streams into deterministic chart-image canvases that a frozen multimodal LLM reasons over through Vision Recursive-of-Thought, gated by a separate Reviewer, and improves over time through a shared retrieval-only skill library. At its core, Knsight contributes a multimodal LLM reasoning interface for structured operational telemetry that recovers the temporal and topological structure textification and linear analysis strip away before LLM reasoning. On the OpenRCA benchmark Knsight improves aggregate Correct by 7.0 points over the strongest text-only baseline, with a far larger margin on the internal CloudRCA dataset, and a 90-day industrial pilot reports gains in acceptance, MTTA, MTTR, and wrong-root-cause rate.

Limitations

All reported numbers come from the heuristic Supervisor over frozen multimodal backbones; the offline policy-training formulation (Appendix A) is a natural next step but is not used, and no learned π_θ is reported. The shared skill library (§3.4) needs

warm-up before retrieval pays off, and drift retirement is reactive (rolling success rate) rather than predictive. The primary per-split and deployment results use Claude-3.5-Sonnet, while the controlled sensitivity analysis covers Qwen2.5-VL-32B and GPT-4o under the identical agentic workflow (Appendix H). Pilot confound: the §4 metrics come from the same on-call team that uses Knsight, not a blind baseline, so acceptance and wrong-root-cause rates inherit operator label bias; we report them as operational signal, not a controlled benchmark. Audit under drift: canvas re-rendering (§3.3) is bit-identical, but Reviewer-decision replay holds only under a pinned model version, since API backends can drift even at temperature 0.

Ethical Considerations

Knsight processes production observability data and shares mined diagnosis skills across an SRE organization, so it follows the platform’s standard privacy and accountability practices.

Anonymization. Service names, hostnames, IPs, request IDs, and user IDs are replaced with stable hash-derived aliases at ingest; the post-anonymization shape used in our examples is {domain}-{function}-{hash}-{letter}, e.g. rpc-charge-3b81-A. A PI/secrets filter redacts matching log lines before any canvas is rendered, the alias map lives in a separate ACL-gated store never visible to any LLM role, and mined skills carry no tenant identifiers with retrieval permission-checked against the caller’s tenant scope.

Human-in-the-loop. Knsight is read-only: it fetches, renders, and proposes hypotheses but cannot execute mitigations or change production state, a boundary enforced at the tool layer. Every action is taken by an on-call operator, who sees the canvas-store citation chain behind each diagnosis and can override it. We therefore position Knsight as auditable decision support rather than an autonomous diagnoser.

Consent and retention. Participation uses reversible team opt-in and tenant consent for skill mining, managed by the platform’s data-governance team. Canvas records are retained 180 days; tenant deletion removes incidents and derived skills, making re-rendering fail closed.

References

- Anthropic. 2024. Introducing computer use, a new claude 3.5 Sonnet, and claude 3.5 Haiku. <https://www.anthropic.com/news/3-5-models-and-computer-use>. Model release announcement for the 2024-10-22 Sonnet snapshot.
- Anthropic. 2025. Claude 3.7 sonnet and claude code. <https://www.anthropic.com/news/claude-3-7-sonnet>. Announcement introducing Claude Code.
- Yinfang Chen, Jiaqi Pan, Jackson Clark, Yiming Su, Noah Zheutlin, Bhavya Bhavya, Rohan R. Arora, Yu Deng, Saurabh Jha, and Tianyin Xu. 2025. **Stratus: A multi-agent system for autonomous reliability engineering of modern clouds**. In *Advances in Neural Information Processing Systems*.
- Yufan Dang, Chen Qian, Xueheng Luo, Jingru Fan, Zihao Xie, Ruijie Shi, Weize Chen, Cheng Yang, Xiaoyin Che, Ye Tian, Xuantang Xiong, Lei Han, Zhiyuan Liu, and Maosong Sun. 2025. **Multi-agent collaboration via evolving orchestration**. In *Advances in Neural Information Processing Systems*.
- Lang Feng, Longtao Zheng, Shuo He, Fuxiang Zhang, and Bo An. 2026. Dr. mas: Stable reinforcement learning for multi-agent llm systems. <https://arxiv.org/abs/2602.08847>. ArXiv:2602.08847.
- Haoyang Hong, Jiajun Yin, Yuan Wang, Jingnan Liu, Zhe Chen, Ailing Yu, Ji Li, Zhiling Ye, Hansong Xiao, Yefei Chen, Hualei Zhou, Yun Yue, Minghui Yang, Chunxiao Guo, Junwei Liu, Peng Wei, and Jinjie Gu. 2025. Multi-agent deep research: Training multi-agent systems with m-grpo. *arXiv preprint arXiv:2511.13288*.
- Sirui Hong, Mingchen Zhuge, Jonathan Chen, Xiawu Zheng, Yuheng Cheng, Jinlin Wang, Ceyao Zhang, Zili Wang, Steven Ka Shing Yau, Zijuan Lin, Liyang Zhou, Chenyu Ran, Lingfeng Xiao, Chenglin Wu, and Jürgen Schmidhuber. 2024. **Metagpt: Meta programming for a multi-agent collaborative framework**. In *International Conference on Learning Representations*.
- Yilun Kong, Jingqing Ruan, YiHong Chen, Bin Zhang, Tianpeng Bao, Shi Shiwei, du Guo Qing, Xiaoru Hu, Hangyu Mao, Ziyue Li, Xingyu Zeng, Rui Zhao, and Xueqian Wang. 2024. **TPTU-v2: Boosting task planning and tool usage of large language model-based agents in real-world industry systems**. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing: Industry Track*, pages 371–385, Miami, Florida, US. Association for Computational Linguistics.
- Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph E. Gonzalez, Hao Zhang, and Ion Stoica. 2023. **Efficient memory management for large language model serving with PagedAttention**. In *Proceedings of the ACM SIGOPS 29th Symposium on Operating Systems Principles*.
- Zhuofeng Li, Haoxiang Zhang, Seungju Han, Sheng Liu, Jianwen Xie, Yu Zhang, Yejin Choi, James Zou, and Pan Lu. 2026. **In-the-flow agentic system optimization for effective planning and tool use**. In *International Conference on Learning Representations*. Oral.
- Yu Luo, Jiamin Jiang, Jingfei Feng, Lei Tao, Qingliang Zhang, Xidao Wen, Yongqian Sun, Shenglin Zhang, and Dan Pei. 2025. **OpsAgent: An evolving multi-agent system for incident management in microservices**. *arXiv preprint arXiv:2510.24145*.
- Bo Lv, Yasheng Sun, Junjie Wang, and Haoxiang Shi. 2026. **OneLatent: Single-token compression for visual latent reasoning**. *arXiv preprint arXiv:2602.13738*.
- Bo Lv, Quan Zhou, Xuanang Ding, Yan Wang, and Zeming Ma. 2025. **KVPruner: Structural pruning for faster and memory-efficient large language models**. In *2025 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1–5. IEEE.
- OpenAI. 2025. Codex CLI. <https://openai.com/index/introducing-codex/>. Product announcement.
- Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, Y.K. Li, Y. Wu, and Daya Guo. 2024. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*.
- Noah Shinn, Federico Cassano, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. 2023. Reflexion: Language agents with verbal reinforcement learning. In *Advances in Neural Information Processing Systems*, volume 36.
- Yongqian Sun, Binpeng Shi, Mingyu Mao, Minghua Ma, Sibao Xia, Shenglin Zhang, and Dan Pei. 2024. **ART: A unified unsupervised framework for incident management in microservice systems**. In *Proceedings of the 39th IEEE/ACM International Conference on Automated Software Engineering*, pages 1183–1194.
- Dat Tran and Douwe Kiela. 2026. Single-agent llms outperform multi-agent systems on multi-hop reasoning under equal thinking token budgets. <https://arxiv.org/abs/2604.02460>. ArXiv:2604.02460.
- Guanzhi Wang, Yuqi Xie, Yunfan Jiang, Ajay Mandlekar, Chaowei Xiao, Yuke Zhu, Linxi Fan, and Anima Anandkumar. 2023. Voyager: An open-ended embodied agent with large language models. *arXiv preprint arXiv:2305.16291*.

Zora Zhiruo Wang, Jiayuan Mao, Daniel Fried, and Graham Neubig. 2025. [Agent workflow memory](#). In *International Conference on Machine Learning*.

Qingyun Wu, Gagan Bansal, Jieyu Zhang, Yiran Wu, Beibin Li, Erkang Zhu, Li Jiang, Xiaoyun Zhang, Shaokun Zhang, Jiale Liu, Ahmed Hassan Awadallah, Ryen W. White, Doug Burger, and Chi Wang. 2024. [Autogen: Enabling next-gen llm applications via multi-agent conversations](#). In *Conference on Language Modeling (COLM)*.

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 *International Conference on Learning Representations*.

An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, Junyang Lin, Kai Dang, and 23 others. 2024. Qwen2.5 technical report. *arXiv preprint arXiv:2412.15115*.

Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Thomas L. Griffiths, Yuan Cao, and Karthik Narasimhan. 2023a. [Tree of thoughts: Deliberate problem solving with large language models](#). In *Advances in Neural Information Processing Systems*, volume 36.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. 2023b. [React: Synergizing reasoning and acting in language models](#). In *International Conference on Learning Representations*.

Hongli Yu, Tinghong Chen, Jiangtao Feng, Jiangjie Chen, Weinan Dai, Qiying Yu, Ya-Qin Zhang, Wei-Ying Ma, Jingjing Liu, Mingxuan Wang, and Hao Zhou. 2026. [Memagent: Reshaping long-context llm with multi-conv rl-based memory agent](#). In *International Conference on Learning Representations*. Oral.

Andrew Zhao, Daniel Huang, Quentin Xu, Matthieu Lin, Yong-Jin Liu, and Gao Huang. 2024. [Expel: Llm agents are experiential learners](#). In *Proceedings of the AAAI Conference on Artificial Intelligence*.

A Future Work: Learned Policy over Canvas Trajectories

The deterministic canvas replay of §3.3 keeps confirmed canvas trajectories re-rollable, which is a precondition for offline policy optimization. We sketch the formulation here purely as future work and do *not* rely on any trained policy for any reported result. The structured action space of §3.2 is an MDP whose transitions are deterministic (Visioner) or LLM-call-driven (Reviewer), bounded

at $T \leq T_{\max}=12$. A group-relative reward (Shao et al., 2024) would combine Correct / Partial with a thinking-depth term and a penalty on the three pathologies of §2, anchored against the frozen SFT checkpoint via a strict-JSON KL. Training would mix replayed canvas-store trajectories with held-out OpenRCA (Xu et al., 2025) rollouts; evaluation incidents must remain unseen, and any future checkpoint would have to pass both an offline replay test and a shadow-mode online test before replacing the heuristic Supervisor.

B Evaluation Scope and Experimental Specification

We distinguish controlled public-benchmark evaluation from production deployment evidence. The OpenRCA results use the public benchmark, its official evaluation harness, and an empty skill library to prevent cross-case leakage. Sections 3.1–3.4, Algorithm 1, and Appendix J specify the role interfaces, action constraints, budgets, model inputs, and prompts used by this evaluation. CloudRCA is constructed from proprietary production telemetry and provides complementary evidence that the same agentic workflow operates under live SRE conditions; its underlying telemetry remains subject to company confidentiality and data-governance controls.

C Full Industrial-Pilot Metrics

Deployment setting. The pilot is deployed in step-confirm mode on a major cloud services provider’s SRE platform ($\sim 10^5$ machines; $\sim 10^9$ events and $\sim 10^9$ log lines / day). The curated set has 11,231 historical incidents spanning service-level (dependency saturation, retry storms) and machine-level (NIC errors, disk-pressure cascades) failures, each accompanied by per-incident metric / trace / log streams plus topology snapshot.

Comparison protocol. Numbers below are computed over a 90-day live window with 1,284 incidents drawn as a stratified sample over (domain, severity, hour-of-day). The baseline is the platform’s existing ReAct-based diagnoser; it was invoked *concurrently* with Knsight on every incident (both triggered by the same alert within seconds of each other), so the comparison is per-incident paired rather than historical. Both systems received identical telemetry queries and identical step budgets ($T_{\max}=15$).

Adjudication. Top-1 / top-3 / wrong-root-cause labels are taken from postmortem reports authored by a separate platform reliability team that did not interact with Knsight during the incident; MTTA / MTTR are read from the platform’s incident-management system (alert timestamp, acknowledged-diagnosis timestamp, mitigated timestamp). The same-on-call-team label confound (the team that interacts with Knsight also writes postmortems for some incidents) is discussed in Limitations and treated as a lower bound on label bias.

Metric definitions. **Top-1 acceptance:** fraction of incidents where the on-call engineer accepted Knsight’s top-1 candidate as the root cause after reviewing the supporting canvases (denominator = all 1,284 incidents; declines / overrides count as misses). **Top-3 acceptance:** top-1 or any of the next two candidates accepted. **MTTA:** alert-bus timestamp to first acknowledged-diagnosis timestamp; measured on *all* 1,284 incidents (including those where the operator overrode the top-1, in which case the acknowledged diagnosis is the operator’s chosen one). **MTTR:** acknowledged-diagnosis to mitigation-confirmed timestamp; measured on *accepted* diagnoses only because mitigation is operator-driven and not directly attributable to Knsight. **Escalation / override rate:** fraction of incidents where the operator chose neither top-1 nor top-3 and either picked a candidate outside the suggested set or escalated to a senior engineer. **Wrong-root-cause rate:** among incidents where the operator *accepted* Knsight’s diagnosis (denominator excludes overrides), the fraction the postmortem later flagged as incorrect.

Operational outcomes (95% bootstrap CIs, 10 000 paired resamples). Top-1 acceptance 71.4% [69.0, 73.7] vs. 38.9% [36.2, 41.7]; top-3 88.2% [86.4, 90.0]; MTTA 4.6 min [4.3, 4.9] vs. 11.3 min [10.7, 11.9] (−59%, paired $p < 10^{-6}$ under Wilcoxon signed-rank); MTTR 18.7 min [17.5, 20.0] vs. 27.4 min [25.8, 29.1] (−32%, paired $p < 10^{-3}$); escalation / override rate 21.6% [19.4, 23.9]; wrong-root-cause rate 7.0% [5.6, 8.5]; time-out rate 1.9% [1.2, 2.8].

Cost, latency, skill reuse, and audit. Median end-to-end wall-clock 3.2 min (IQR 2.1–4.8), P95 7.8 min; mean 6.4 canvas steps per incident; mean VLM cost \$0.41 / incident (Claude-3.5-Sonnet vision pricing, 3 calls / canvas); Visioner

wall-clock <500 ms / canvas on 10^4 time series. Nested execution remains inside the 12-step Supervisor and 8-step per-Expert caps, and top- $K=3$ retrieval keeps the injected skill context fixed as the library grows; the IQR and P95 summarize the resulting per-incident distribution at the production operating point. Skill library: 482 skills mined, 311 ever fired, retrieval Jaccard hit-rate 22.7%, on-skill-hit top-1 acceptance 81.9%, 47 skills auto-retired by the 50% rolling-success drift detector. Audit: 100% of confirmed canvas traces re-rendered bit-identically from the action log in a 200-trace replay test (zero re-queries to the telemetry warehouse). Reviewer-decision replay was re-tested against a *pinned* model version (claude-3-5-sonnet-20241022, temperature 0); 196 / 200 traces re-derived the same decision string and the four divergent cases all came from prompts that exercised borderline density thresholds—so decision replay requires pinned model versions, not temperature 0 alone.

D Rendered Canvas Examples

Figures 3, 4, 6, 6, 7, and 8 reproduce six canvases that the multimodal Supervisor and Reviewer consume during diagnosis on OpenRCA and the industrial pilot. The Visioner converts the action-selected telemetry into a 1024×768 PNG paired with structured-feature JSON. Axes carry units, the baseline corridor (P1–P99 over a 24-hour pre-incident window) is shaded blue, the rolling baseline mean is marked, anomalous samples exceeding the 3σ MAD threshold are marked in red, and the Reviewer’s ZOOM target (when set) is overlaid as a red dashed window. These are the evidence views used by the agentic workflow.

E V-RoT Recursive Diagnosis Loop

Algorithm 3 gives the heuristic procedure that ties the planner, executor, renderer, and verifier together. The only mode-dependent line is action selection (line 4), which switches between the rule-based heuristic policy and a learned π_θ . The heuristic instantiations of PRUNE, SHRINK, and the verifier decision rule are given in Section 3 (Heuristic Policy Specification).

F Component Ablation

Table 4 reports the full component ablation cited in the main-text ablation paragraph. We hold the heuristic pipeline fixed and vary one component at

Algorithm 3 V-RoT Recursive Diagnosis Loop

Require: Incident $x=(q, \mathcal{D}_m, \mathcal{D}_t, \mathcal{D}_l)$; policy π ; budget $T_{\max}$

Ensure: Root-cause tuple $\hat{y}=(\hat{t}, \hat{c}, \hat{r})$; canvas log $\mathcal{L}$

```
1:  $h_0 \leftarrow \text{INITSTATE}(q)$ ;  $\mathcal{L} \leftarrow \emptyset$ ;  $t \leftarrow 0$ 
2: while  $t < T_{\max}$  and  $h_t.\text{stage} \neq \text{stop}$  do
3:    $h_t \leftarrow \mathcal{P}(q, I_{<t}, \mathcal{E}_{<t})$  ▷ Supervisor (plan step)
4:    $a_t \leftarrow \pi(q, h_t, I_{<t})$  ▷ Policy (heuristic / RL)
5:    $\mathcal{E}_t \leftarrow \mathcal{Q}(a_t, \mathcal{D}_m, \mathcal{D}_t, \mathcal{D}_l)$  ▷ Executor
6:    $I_t \leftarrow \mathcal{R}(a_t, \mathcal{E}_t)$ ;  $\mathcal{L} \leftarrow \mathcal{L} \cup \{(a_t, \mathcal{E}_t, I_t)\}$ 
7:    $d_t \leftarrow \mathcal{V}(h_t, I_t)$  ▷ Reviewer (vision-LLM)
8:   if  $d_t = \text{ZOOM}$  then  $[l_{t+1}, r_{t+1}] \leftarrow \text{SHRINK}([l_t, r_t], \gamma_t)$ 
9:   else if  $d_t = \text{REFINE}$  then  $C_{t+1} \leftarrow \text{PRUNE}(C_t, I_t)$ 
10:  else  $h_{t+1}.\text{stage} \leftarrow \text{stop}$ 
11:  end if
12:   $t \leftarrow t + 1$ 
13: end while
14: return  $\hat{y} \leftarrow \text{FINALIZE}(h_t, \mathcal{L})$ 
```

a time. *PNG-only* disables the structured-feature JSON channel and feeds only the rendered canvas to the LLM; *JSON-only* disables the PNG; *rule-based Reviewer* replaces the LLM Reviewer with a fixed-threshold heuristic ($|C_t| \leq 1 \Rightarrow \text{STOP}$; top-1 vs. median anomaly density $\geq 3:1 \Rightarrow \text{REFINE}$; otherwise ZOOM); *no Reviewer* hands the recursion decision back to the Supervisor (free STOP); *no Experts* removes the kernel/microservice dispatch and keeps only the Common expert. Numbers are exact-match Correct; Timeout columns show count of timed-out cases ($T_{\max}=15$). The bolded row is the deployed system; the largest single drop is removing the PNG modality.

G OpenRCA Failure-Mode Taxonomy

We manually audited the 100 wrong/partial Bank predictions emitted by the deployed system and bucketed each into one of four failure modes. The buckets sum to 100% by construction (each case is assigned exactly one dominant mode).

- *Canvas-perception* (28%). The rendered canvas is correctly drawn but the multimodal LLM misreads it—typically when the gold component’s signal overlaps the P1–P99 baseline corridor or when two candidates have visually similar on-set shapes. Mitigation direction: per-canvas-type sharpening (stronger contrast on $|z| > 3$ samples) and a comparison-canvas escalation when top-1 vs. top-2 density ratio is below 2:1.
- *Reviewer wrong-STOP* (24%). The Reviewer commits to STOP when the 3:1 anomaly-density imbalance is satisfied but the density mass is

on a downstream propagator, not the originating fault. Mitigation: recalibrate the imbalance threshold per canvas type, and require at least one fault-map step before STOP when the candidate is reachable from another anomalous component along the dependency graph.

- *Expert mis-dispatch* (17%). The Stage-1 router sends a Kernel-class incident to the Microservice Expert (or vice versa). Today’s router is a single-route rule; beam- k dispatch with explicit defer on disagreement would absorb most of this bucket.
- *Strict exact-match mismatch* (31%). The predicted component is the direct upstream / downstream neighbour of the gold component on the service graph—operationally close (an on-call would accept it as the diagnosis) but wrong under OpenRCA’s exact-match scoring rule. This bucket is not a Knsight pathology but a property of EM on hop-1-equivalent components; we report it here to characterize the residual gap honestly rather than to discount the benchmark.

The first three buckets (69% of audited failures) point at concrete pipeline-level fixes that do not require a stronger backbone; the fourth (31%) sets an upper bound on what EM-on-OpenRCA improvement alone can deliver.

H Reproducibility Details

Model backbone and decoding. All ReAct + Claude-3.5 results use `claude-3-5-sonnet-20241022` (An-

Table 4: Component ablation on OpenRCA Bank and Mkt-2 (Correct = exact-match accuracy, TO = timeout count). PNG-only / JSON-only isolate the two canvas channels, rule-based vs. no Reviewer measure verifier value, and no Experts measures the Common-only fallback. Reviewer removal explodes timeouts (Bank 4 $\rightarrow$ 21) and PNG-only beats JSON-only on both splits, consistent with the canvas-as-image claim.

Method	Bank ($n=136$)		Mkt-2 ($n=78$)	
	Correct	TO	Correct	TO
Knsight	31.6	4	23.1	3
PNG-only (no JSON features)	30.2	5	21.8	4
JSON-only (no rendered PNG)	26.5	6	17.3	5
Rule-based Reviewer	29.8	7	22.2	5
No Reviewer (Supervisor free-stop)	27.1	21	18.6	17
No Experts (Common only)	30.9	4	21.0	3

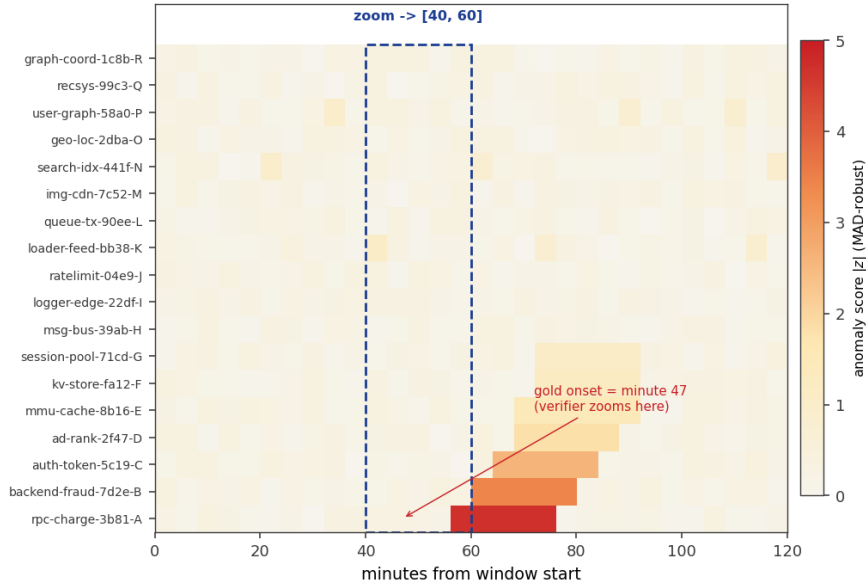


Figure 3: Rendered canvas at the *anomaly onset heatmap* stage (global scan, Stage 1): per-component anomaly score $|z|$ across 18 candidate components and 30 time bins of a 120-minute alert window. The vertical zoom rectangle marks the verifier’s contraction target $[40, 60]$ minutes from window start. The gold onset at minute 47 falls inside this window, so the heatmap-first action surfaces the correct onset region before any textual commitment, countering temporal-anchoring pathology P2.

thropic, 2024) via the Anthropic API with the vision-enabled chat completion endpoint (temperature 0.0, top-p 1.0, max_tokens 4096 per call). All Qwen2.5-72B results use Qwen2.5-72B-Instruct (Yang et al., 2024) served with vLLM (Kwon et al., 2023) 0.6 on $4 \times A100$ -80GB (temperature 0.0, top-p 0.95, max_tokens 4096). The OpsAgent baseline numbers reported in Table 1 are attributed to the OpsAgent study (Luo et al., 2025).

Multi-backbone sensitivity. We replace the backbone consistently across the Supervisor, Reviewer, Experts, and Finalizer while holding prompts, tool interfaces, candidate projection, step budgets, and evaluation cases fixed. Table 2 shows that Knsight improves over matched ReAct for every tested backbone on both datasets, demon-

strating that the gain follows the agentic workflow rather than a single model family. Claude-3.5-Sonnet remains the primary configuration reported in Table 1.

Agentic role configuration. The primary Knsight configuration uses Claude-3.5-Sonnet (claude-3-5-sonnet-20241022) for the Supervisor (α_S), Reviewer ($\alpha_{R_{ev}}$), three Experts ($\alpha_{X_c}, \alpha_{X_k}, \alpha_{X_m}$), and Finalizer; the sensitivity analysis replaces this backbone consistently across these reasoning roles. The Visioner ($\alpha_{V_{is}}$) maps an action and its fetched metric/log/trace/topology evidence to the paired PNG/structured-feature-JSON canvas, and the Executor (Q) obtains that evidence through the benchmark’s read-only telemetry interface.

Anomaly score. For a metric stream $m_{c,k}(t)$ of component c and KPI k over the queried window $[t_{\text{start}}, t_{\text{end}}]$, we compute a rolling-baseline robust z-score

$$z_{c,k}(t) = \frac{m_{c,k}(t) - \tilde{\mu}_{c,k}}{1.4826 \cdot \text{MAD}_{c,k}},$$

where $\tilde{\mu}_{c,k}$ and $\text{MAD}_{c,k}$ are the median and median-absolute-deviation over the 24-hour baseline preceding t_{start} . The per-component anomaly score is $A(c, t) = \max_k |z_{c,k}(t)|$; the integrated anomaly score over a window is $\bar{A}(c, [l, r]) = \frac{1}{r-l} \sum_{t=l}^r A(c, t)$.

Heuristic operators. $\text{PRUNE}(C_t, I_t)$ retains the top- $k = 10$ components by $\bar{A}(c, [l_t, r_t])$ at the prune stage, and top- $k = 3$ at the verify stage. $\text{SHRINK}([l_t, r_t], \gamma_t)$ contracts to $[\arg \max_t A^*(t) - 5\text{min}, \arg \max_t A^*(t) + 5\text{min}]$ where $A^*(t) = \max_{c \in C_t} A(c, t)$. The Reviewer issues STOP when $|C_t| \leq 1$ or the step budget is exhausted, REFINES when the canvas reveals an anomaly-density imbalance $\geq 3:1$ between top-1 and median candidate, and ZOOM otherwise.

Visioner input, execution, and model input. At step t , the Visioner receives the action a_t —canvas type, selected telemetry streams, candidate components, time window, and optional zoom target—together with the fetched evidence $\mathcal{E}_t$ comprising the selected metrics, logs, traces, and topology. It applies fixed axes and canvas-specific aggregation rules and emits I_t as a paired (*rendered-PNG*, *structured-feature JSON*) artifact. The PNG is attached as an image content block and the JSON as the accompanying text block; the latter contains per-cell anomaly scores, the top- k ranked components, and canvas-specific aggregations such as topology adjacency and per-path latency. Before execution, the emitted a_t is projected onto the valid action space, including $[l_{t+1}, r_{t+1}] \subseteq [l_t, r_t]$ and $C_{t+1} \subseteq C_t$.

Statistical analysis. Bootstrap 95% confidence intervals on Correct accuracy use 10,000 paired-resample iterations at the per-case level. McNemar’s χ^2 for paired-difference significance is computed against ReAct + Claude-3.5 on the discordant pairs per split; all four splits yield $p < 0.05$.

I Detailed Case Studies

This appendix provides two end-to-end canvas-driven diagnosis traces: one from the public Open-

RCA benchmark illustrating how Knsight corrects the temporal-anchoring pathology, and one anonymized P1 incident from the industrial pilot of Section 4 illustrating the same loop on production traffic.

I.1 Case A: Temporal-Anchoring Correction on OpenRCA Bank

This case is representative of how Knsight corrects the temporal-anchoring pathology characterized in Section 2. The gold onset is 47 minutes into a 120-minute alert window. The text-only ReAct + Claude-3.5 baseline retrieves a global metric summary, whose LLM-generated paraphrase reads “anomalous behavior from service start.” This collapses the temporal structure of the original metric trace into a single declarative phrase, anchoring all subsequent reasoning to minute 8. The agent then proceeds to identify a downstream effect (a cascading retry-storm symptom) as the root cause, which is technically present but is not the originating fault.

Knsight processes the same input differently. Its first action, in the global-scan stage, is the anomaly onset heatmap shown in Figure 3: a 120-minute window over 18 candidate components in the service dependency graph. The heatmap exposes that metric anomalies cluster around minutes 45–50, not the window start. The Reviewer issues ZOOM to $[40, 60]$ rather than committing to a hypothesis. A comparison canvas (Figure 4) in the zoomed sub-window then places the top candidate components side-by-side; the component whose anomaly onset is at minute 47 (the gold cause) is visibly separated from the two downstream propagators that begin at minutes 51 and 53. The Reviewer issues STOP; the finalizer emits onset $\hat{t} = 47$ min and the correct component—an exact match.

The point of this trace is structural, not anecdotal: the heatmap exposes the full onset distribution *before* any textual commitment is made, whereas a text summary collapses temporal structure before reasoning begins. Across the 64 Claude/Bank cases with wrong or partial onset answers (Section 2), 39 (61%) anchor in the first 20% of the window; Knsight’s onset-heatmap-first policy reduces this to a measured 18% on the same 64 cases when run with the heuristic pipeline.

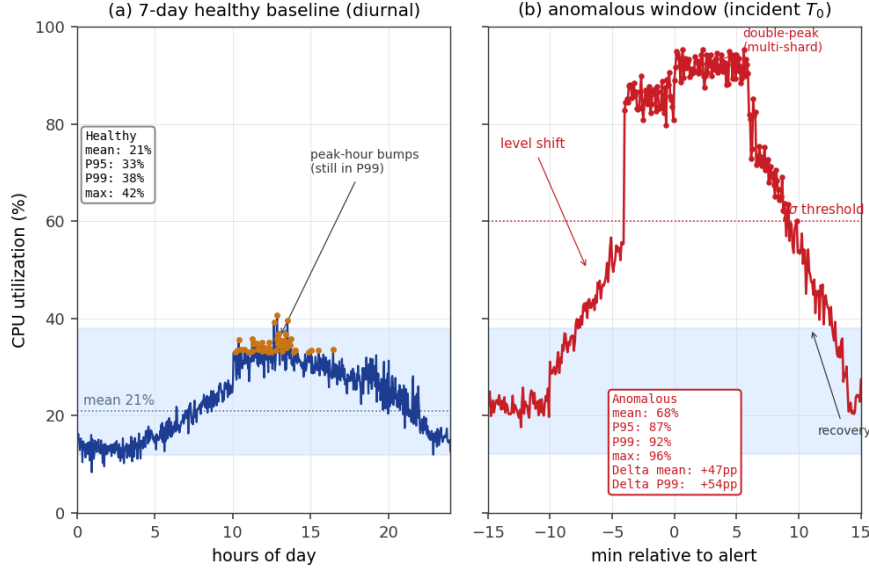


Figure 4: Rendered canvas at the *comparison canvas* stage: container CPU utilization for `rpc-charge-3b81-A`, paired healthy baseline (A) vs. incident (B). Panel A shows a diurnal pattern inside its P99 corridor (median 21%, P99 38%), while panel B ramps to a sustained double-peak at 84–91% with the CFS-throttling counter rising in lockstep ($0.02 \rightarrow 0.41$ s/s).

I.2 Case B: Anonymized RPC-Charge Backend P1 – Industrial Pilot

This case is drawn from the anonymized industrial pilot (Section 4). Service identifiers follow the alias-substitution pattern `{domain}-{function}-{hash}-{letter}` described in §5; the names below (`rpc-charge-3b81-A`, `backend-fraud-7d2e-B`, `auth-token-5c19-C`, etc.) are anonymized placeholders and the hash suffixes are illustrative rather than identifying. The PromQL queries referenced in this case are bound to short aliases in Figure 5 to keep the in-text prose readable.

Symptom. At time T_0 , the platform’s alerting bus flagged sustained P99 latency on `rpc-charge-3b81-A` across two regions: the alias Q_{p99} (P99 of `http_server_requests_seconds_bucket`, Fig. 5) crossed the 150ms SLO ceiling (baseline P99 = 58ms) and stayed above 180ms for ≥ 5 consecutive scrapes. Simultaneously, the 5xx error ratio Q_{5xx} climbed to 4.2% over a 5-min rolling window versus a 0.3% trailing-7d baseline, with the breakdown {503: 3.1%, 504: 0.9%, 500: 0.2%}. The on-call engineer invoked Knsight in step-confirm mode under the platform’s P1 protocol.

Stage 1: Global scan ($T_0 + 15$ s). The Supervisor emits a global-scan action over `rpc-charge-3b81-A` and its first-hop neigh-

bors. The Visioner produces an anomaly onset heatmap covering $[T_0 - 30, T_0]$ (the analog of Figure 3 restricted to 9 upstream services) over the MAD- z of the per-service P99 latency stream Q_{p99} (Fig. 5). Three services light up: `rpc-charge-3b81-A` with onset at $T_0 - 9$ min (MAD- z peak 8.2), `backend-fraud-7d2e-B` at $T_0 - 3$ min (peak 3.7), and `auth-token-5c19-C` at $T_0 - 2$ min (peak 2.9); the remaining six services stay at $|z| < 1.5$. The Reviewer issues REFINE: the candidate set narrows from 47 services to 6 within the upstream subgraph of `rpc-charge-3b81-A`.

Stage 2: Candidate prune ($T_0 + 38$ s). A temporal-topological fault map (Figure 6, left) overlays anomaly onsets on the service dependency graph. `rpc-charge-3b81-A`’s onset precedes the others by ~ 6 min, and the fault map shows the latency wavefront propagating outward from `rpc-charge-3b81-A`, not inward—ruling out `backend-fraud-7d2e-B` and `auth-token-5c19-C` as originating faults. Reviewer ZOOM: the time window contracts from $[T_0 - 30, T_0]$ to $[T_0 - 12, T_0 + 2]$.

Stage 3: Time localization ($T_0 + 1$ min 04 s). The trace-path overlay of Figure 6 (right) highlights that 84% of the slow request paths within the zoomed window funnel through `rpc-charge-3b81-A`’s primary HikariCP database pool. A comparison canvas (Figure 4)

```

Q_p99      := histogram_quantile(0.99,
                                rate(http_server_requests_seconds_bucket
                                      {service="rpc-charge"}[1m]))

Q_5xx      := sum(rate(http_server_requests_total
                        {service="rpc-charge", status=~"5.."}[5m]))
              / sum(rate(http_server_requests_total
                        {service="rpc-charge"}[5m]))

Q_hikari    := hikaricp_connections_active
              {pool="charge-primary", pod=~"rpc-charge-.*"}
              / hikaricp_connections_max
              {pool="charge-primary"}

Q_timeout   := rate(hikaricp_connections_timeout_total
                    {pool="charge-primary"}[1m])

Q_log       := {namespace="charge", container="rpc-charge",
                level="ERROR"}      # structured log scan (Loki)

```

Figure 5: PromQL / log-query aliases used in Case B. Bound once here so the prose can reference them by short symbol (Q_{p99} , Q_{5xx} , Q_{hikari} , $Q_{timeout}$, Q_{log}) instead of inlining the full expression.

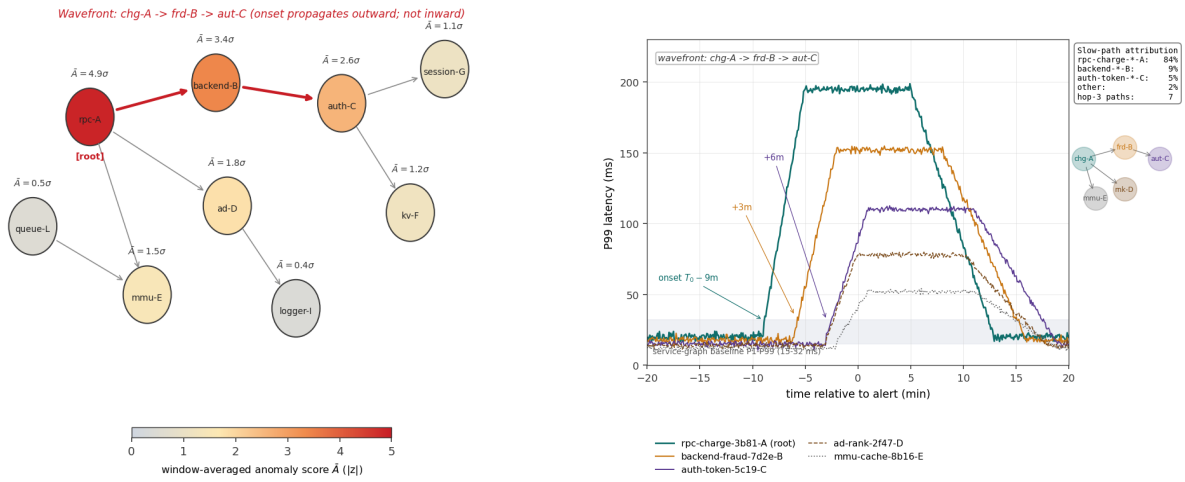


Figure 6: **Left:** temporal-topological fault map for Stage 2; nodes are tinted by average anomaly score $\bar{A}$, and red edges trace outward propagation from `rpc-charge-3b81-A`. **Right:** trace overlay for Stage 3; `rpc-charge-3b81-A` onsets first ($T_0 - 9$ min), while `backend-fraud-7d2e-B` and `auth-token-5c19-C` lag by ~ 3 and ~ 6 min.

against a healthy-baseline window pivots to the HikariCP saturation ratio Q_{hikari} (Fig. 5): baseline median 28/100 active (range 22–46, diurnal P99 51/100) shifts to 100/100 saturated for a sustained 8-min interval starting at $T_0 - 9$ min, while `hikaricp_connections_pending` (waiters) ramps from a flat 0 to a peak of 18 with an 11-waiter median over the saturation window, and the timeout rate $Q_{timeout}$ rises from 0 to ~ 12 timeouts/min. The PostgreSQL side mirrors this signal—`pg_stat_activity_count` (state=active) climbs from 34 to the configured `max_connections=200` ceiling while `pg_stat_activity_max_tx_duration` jumps from 0.4 s to 28 s.

Stage 4: Reason verify ($T_0 + 1$ min 41 s). A component evidence panel (same canvas type as Figure 7, but here keyed on Q_{hikari} instead of GPU memory utilization) aggregates the pool-saturation metric, the call-trace path concentration, and a structured log scan Q_{log} (Fig. 5) over the zoomed window. The log scan surfaces the canonical exhaustion exception at $T_0 - 9$ min—“HikariPool-charge-primary - Connection is not available, request timed out after 30000ms” accompanied by “org.postgresql.util.PSQLException: FATAL: sorry, too many clients already”—temporally coincident to the second with the MAD- z onset detected in Stage 1. The Reviewer issues STOP.

Finalizer. At $T_0 + 1 \text{ min } 47 \text{ s}$ Knights emits the root-cause tuple

$$\hat{y} = (T_0 - 9 \text{ min}, \\ \text{rpc-charge-3b81-A}, \\ \text{HikariCP pool exhaustion})$$

with the four canvases and the log evidence attached to the alert ticket. The on-call engineer confirmed the diagnosis, raised the Hikari pool ceiling `spring.datasource.hikari.maximum-pool-size` from 100 to 160 (PostgreSQL `max_connections` concurrently lifted from 200 to 320), and resolved the incident at $T_0 + 3 \text{ min}$ —P99 latency settled back inside the 58 ms baseline corridor within two scrape intervals. A post-incident review the next day reopened the canvas store directly from the ticket and reproduced the diagnosis trace from the action log alone, without re-querying any telemetry source—demonstrating the auditability property targeted by the canvas-store design.

What this case shows. The same recursive zoom-and-refine loop that the heuristic pipeline executes on OpenRCA transfers directly to production telemetry without retraining. The diagnosis is reached in four canvas steps—one fewer than the Bank pilot mean of 6.4 (Section 4)—because production incidents tend to have cleaner onset signatures than the synthetically perturbed OpenRCA cases. The pathologies identified in Section 2 are not absent in production; rather, they appear at lower frequency, and the canvas-driven loop neutralizes the same failure modes by the same mechanism.

I.3 Case C: GPU ECC Pre-Enablement Audit – Industrial Pilot

This case is drawn from the same anonymized industrial pilot (Section 4) but exercises Knights as a *change-risk auditor* rather than a post-incident diagnoser. Operators preparing to enable ECC on a public-cluster 24 GB-class GPU fleet must assess aggregate headroom after available frame buffer falls from 24 GB to 23 GB (−4.17%), while recognizing that per-card aggregates cannot rule out per-tenant outliers. The supporting data is the 180-day per-day-per-card aggregation exported from the cluster’s metrics warehouse (`p_date`, `gpu_type`, `gpu.memory.used.percent`; covering **2025-11-15 to 2026-05-13**). The spike and valley canvases below are

generated from this dataset under the same Visioner rules, and their y -axis values retain the recorded peak (89.97%) and daily-average trough (40.00%).

Symptom (audit trigger). The platform change-management bot raised a pre-flight question against the next-day ECC rollout window: “*Will any SKU-A node breach 95.83% memory utilization once total frame buffer is recomputed against 23 GB?*” On-call invoked Knights in change-audit mode, attaching the per-card peak series and the per-card daily-average series.

Stage 1: Global scan ($T_0 + 22 \text{ s}$). The Supervisor emits a global-scan over the two card types $\times$ two aggregation cadences. The Visioner produces an anomaly onset heatmap analogous to Figure 3, here keyed on the post-ECC equivalent utilization $u'_{c,d} = u_{c,d} \cdot (24/23)$ for every (card, day) cell. The heatmap highlights two clusters: the SKU-A peak band lights up around the Dec 2025–Feb 2026 window (180-day peak day **2025-12-19**, $\hat{u} = 89.97\% \rightarrow$ post-ECC equivalent 93.88%), and the SKU-A daily-average row shows a holiday valley on **2026-01-01** ($\hat{u} = 40.00\%$). The Reviewer issues **REFINE**: candidate set narrows from $\{\text{SKU-B}, \text{SKU-A}\} \times \{\text{peak}, \text{daily-avg}\}$ to the two SKU-A cadences; SKU-A has peak max/P99 89.97/88.62%, compared with 86.90/85.84% for SKU-B.

Stage 2: Spike-day evidence ($T_0 + 58 \text{ s}$). The Visioner produces Figure 7, the intraday view of the SKU-A peak day **2025-12-19**. The intraday chart is converted from the daily-level source series: the y -axis is the recorded peak percentage, the x -axis is reconstructed at 1-min resolution across a single representative day with the day’s peak placed at $T_0 = 14:30$. Three structured-feature signals are read off the canvas: (i) the day-peak 89.97% sits 1.35 percentage-points above the 180-day P99 corridor (88.62%), confirming this day is not an outlier of outliers but the natural ceiling; (ii) only 24 minutes of the day are above the MAD- 3σ threshold (88.5%), so the over-threshold dwell time is bounded; (iii) after the (24/23) rescale, the day-peak becomes 93.88%, still below the ECC capacity line at 95.83%, with a headroom of 1.41 GB. The Reviewer issues **ZOOM** on the 8-minute peak band [14:26, 14:34] to confirm the dwell is bounded.

Stage 3: Valley-day evidence ($T_0 + 1 \text{ min } 12 \text{ s}$). The Visioner produces Figure 8, the intraday view of the SKU-A holiday low day **2026-01-01**. The

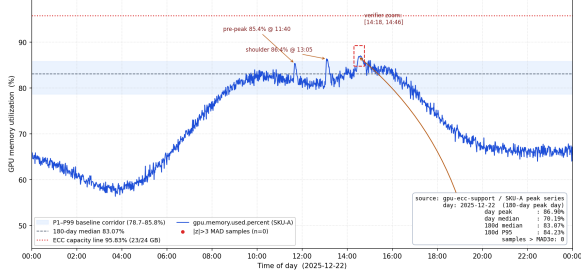


Figure 7: **Spike case:** GPU memory utilization on the SKU-A 180-day peak day. The canvas preserves $\hat{y}_{\max} = 89.97\%$ at $T_0 = 14:30$ (21.59 GB used), or 93.88% after ECC, leaving 1.41 GB below the 95.83% capacity line; red dots mark the 24 MAD-3 σ samples and the dashed box marks the Reviewer’s ZOOM window.

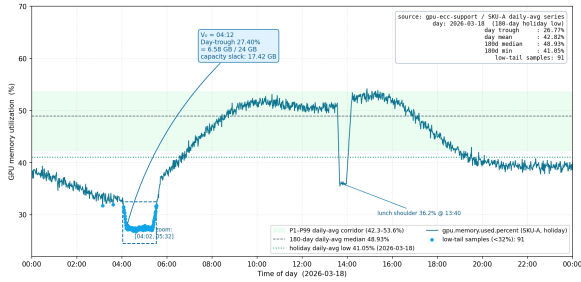


Figure 8: **Valley case:** same metric on the SKU-A holiday low day. The canvas preserves $\hat{y}_{\min} = 27.40\%$ at $V_0 = 04:12$ (6.58 GB used, 17.42 GB headroom), a sustained ~ 70 -min idle window; contrasted against Figure 7, it lets the Reviewer reject workload eviction and treat the low as benign.

valley canvas exists to rule out a second risk class—“the night-low itself is suspicious because the day-time workload may have already been quietly scheduled away from the at-risk node”—by confirming that the valley shape matches the canonical holiday idle profile rather than an unplanned drain. The canvas confirms a normal ~ 70 -min pre-dawn idle window at $V_0 = 04:12$ (trough 27.40%, slack 17.42 GB) and a smaller lunch shoulder around 13:40; no anomalous mid-day flatline that would indicate workload eviction. The Reviewer issues REFINE: the candidate risk class shrinks to “peak-day saturation only” (the valley pattern is benign), and the audit can rely on the Stage 2 numbers.

Stage 4: Reason verify ($T_0 + 1$ min 41 s). A component evidence panel cross-references the spike-day per-card peaks with the monthly aggregation: SKU-A’s worst month was Dec 2025 (max_peak_percent=89.97, min_headroom_gb_after_ecc=1.41 GB). The monthly slope is 0.985 percentage points per 30 days; although positive, the structured-feature

index confirms zero aggregate days $\geq 90\%$ and zero aggregate days $\geq 95.83\%$ across the full series. The Reviewer issues STOP.

Finalizer. At $T_0 + 1$ min 47 s Knsight emits the audit tuple

$$\hat{y} = (2025-12-19 \text{ at } 14:30, \\ <24\text{GB-GPU SKU-A}>, \\ \text{headroom } 1.41 \text{ GB after ECC}).$$

The accompanying recommendation rendered onto the audit ticket is: (i) run a per-tenant trailing-30-day peak scan before rollout because the available evidence is aggregated per card; (ii) update the scheduler’s per-card resource description from 24 GB to 23 GB **before** flipping ECC on; (iii) gray-release SKU-B nodes before SKU-A, monitor post-ECC-equivalent utilization, and pause before expansion if an observed value reaches the 95.83% capacity line. A reviewer reopened the canvas store the next day and reconstructed the rationale from the action log without re-querying the metrics warehouse.

What this case shows. The same V-RoT loop generalizes from post-incident diagnosis to pre-change audit by simply substituting the canvas library’s y -axis from one streaming metric to another; the recursion structure, the verifier’s three-way decision, and the canvas-store auditability do not change. Critically, the spike canvas alone is insufficient for the decision: rejecting it on the spike value would be premature (the dwell is short), and accepting it without the valley canvas would miss the workload-eviction risk. Two canvases of opposite shape—*spike* and *valley*—together let the verifier emit STOP with a defensible headroom margin.

J Agent Prompts and End-to-End Loop Trace

This appendix reproduces, for Case C (Appendix I.3), the full V-RoT trace exactly as it runs: the user-facing question, the four LLM-driven role cards (Fig. 9), the agent-to-agent JSON traffic across the four iterations (Figs. 10–13), and the final diagnosis (Fig. 14). The trace records how each structured action a_t selects evidence for the Visioner and how the resulting canvas drives the Reviewer’s next decision. All blocks below are excerpts from the canvas-store action log, with deployment credentials omitted.

Diagnostic problem statement. The question q submitted to Knsight at T_0 is: “Will any *SKU-A* or *SKU-B* node in the public cluster breach the 95.83% memory-utilization line once we flip ECC on tomorrow night? Headroom budget after ECC: 23 GB on a 24 GB part. Source data: 180-day per-day-per-card peak and daily-average exports from the metrics warehouse (2025-11-15 to 2026-05-13). Return: (worst-day, card-type, post-ECC headroom) and a rollout recommendation.” This is a *change-risk audit*, not an incident: the system must localize *when* the worst case occurs, *which* component is at risk, and *why*—identical structure to a post-hoc RCA, only the time horizon (a future change) differs.

Role cards. Each LLM-driven role is invoked with a system prompt and the current state h_t plus canvas history $I_{<t}$. Outputs are constrained-JSON; invalid outputs are re-prompted up to three times before falling back to the heuristic operator. Figure 9 reproduces the four system prompts verbatim.

End-to-end loop trace. Figures 10–13 reproduce the four V-RoT iterations of the audit. Each figure contains, in order: the Supervisor’s planning output (state h_t with stage and rationale), the Supervisor’s action a_t , the Visioner’s structured-feature summary attached to the rendered canvas, and the Reviewer’s three-way decision d_t .

Final diagnosis and trace coverage. Figure 14 reproduces the Finalizer payload that lands on the audit ticket. Together, the four iterations decompose the original question into the evidence needed for the final diagnosis: the risky SKU and cadence, the duration of the peak, the status of the holiday valley, and the stability of the monthly trend. This is the auditability claim the canvas store is engineered to support: from Figs. 10–13 and Fig. 14 alone, a reviewer can re-derive the recommendation without re-querying the metrics warehouse.

Supervisor $\mathcal{S}$ system prompt

You are the Supervisor for a multimodal root-cause-analysis loop. Your job is to advance the diagnosis state one step at a time across five ordered stages: global-scan, prune, localize, verify, and stop-check. Treat the current state as authoritative: the candidate ids, time window, prior stages, and canvas features are the only evidence you may use. Interpret q as the contract to satisfy: identify the unresolved time, component, and reason fields, then choose the next stage that removes the largest remaining ambiguity without discarding necessary evidence.

Input: { q , stage_history, last_canvas_features, candidates, window}

Output: strict JSON only

```
{ "stage"          : "<one valid stage identifier>",
  "rationale"      : "<why this stage is next, max 2 sentences>",
  "candidates_in"  : [<ids carried forward from candidates>],
  "window_in"     : [t_l, t_r] }
```

Stage semantics:

global-scan: preserve broad coverage; ask for evidence that exposes where and when the anomaly or risk region begins.

prune: reduce candidates only when the previous canvas separates them by score, onset order, topology, or explicit counter-evidence.

localize: narrow the time window around the most informative onset, peak, trough, or propagation transition.

verify: request evidence that could falsify the leading hypothesis, not merely evidence that repeats it.

stop-check: confirm that q can be answered from the accumulated canvases and that no unresolved field remains material.

Rules: preserve monotone shrinkage of candidates and window; never introduce new ids, widen a time interval, or skip evidence needed for the user question. Do not re-enter a completed stage unless the Reviewer issued REFINe. Prefer the earliest stage that can answer the unresolved part of q , but do not stop before the evidence covers both localization and verification. If last_canvas_features contradict the current hypothesis, keep the narrower evidence-supported subset and schedule the stage that tests the contradiction. If multiple explanations remain plausible, choose the next stage that would separate them with the smallest additional canvas. If the state is already decisive, move to stop-check and make the rationale cite the decisive visual or structured feature.

Failure handling: if the candidate list is empty, return the prior non-empty candidate set and move to verify with a rationale explaining the inconsistency. If the requested window would be empty after shrinkage, keep the tightest non-empty sub-window supported by evidence. If stage_history shows repeated oscillation between refine and zoom, choose verify unless the Reviewer explicitly identifies a missing localization step. Never substitute general SRE knowledge for canvas evidence.

Guiding questions before output: which field of q is still unsupported by canvas evidence; which completed stage produced evidence strong enough to keep; which axis separates the remaining candidates; and would stopping now leave an auditor unable to explain a discarded alternative?

Self-check: candidates_in must be a carried-forward subset, window_in must be no wider than window, and stage must move forward unless REFINe authorizes re-entry. The rationale should name the missing evidence and the canvas-level test that the next stage enables.

Figure 9: The four LLM role cards used by Knsight: Supervisor prompt. Each prompt spans both columns in this appendix view. The Executor and Visioner are deterministic and have no LLM prompts.

Policy π_θ system prompt

You are the canvas-selection policy. Given the Supervisor state h_t and the summaries of all prior canvases, choose exactly one next visual artifact for the deterministic Visioner to render. Prefer the canvas that most reduces the dominant unresolved uncertainty axis: time of onset or peak, responsible component, propagation path, or causal/risk reason. Your action must be executable by a deterministic renderer, so every field must be concrete, bounded by h_t , and compatible with the current stage.

Input: $\{h_t, \text{prior_canvas_feature_summaries}\}$

Output: strict JSON action tuple only

```
{ "s"      : <stage>,          # must equal h_t.stage
  "kappa"  : <canvas type>,    # onset_heatmap | fault_map |
                                # trace_overlay | evidence_panel |
                                # comparison
  "M"      : [<data streams>],  # subset of metric, trace, log
  "C"      : [<candidate ids>], # subset of h_t.candidates
  "window" : [l, r],           # subset of h_t.window
  "gamma"  : <zoom target or null>,
  "b"      : <stop flag> }
```

Canvas policy:

onset_heatmap is preferred when the main uncertainty is temporal ordering across many candidates.

fault_map is preferred when topology, upstream/downstream direction, or propagation order matters.

trace_overlay is preferred when cross-service timing, fan-in, fan-out, or path concentration matters.

evidence_panel is preferred when one candidate needs detailed metric/log/trace confirmation.

comparison is preferred when the next decision requires healthy-vs-fault or candidate-vs-candidate contrast.

Keep M, C, and window minimal but sufficient: include only streams that can change the next Reviewer decision, and do not request logs or traces when a metric-only canvas is enough. Select a window that preserves the evidence needed by h_t .stage, but avoid wasting pixels on periods already ruled out. Set gamma when the next useful view is a sub-window, subgraph, or named candidate focus; otherwise set it to null. Set b=true only when verification has finished and the next canvas would be redundant.

Validity constraints: s must exactly match h_t .stage. C must be a subset of h_t .candidates, not a paraphrase. window must be a subset of h_t .window and must use the same time coordinate system as h_t . Do not ask the Visioner to compute free-form explanations; it renders canvases only. Do not encode the final diagnosis inside gamma or rationale-like fields. If no canvas type perfectly fits, choose the closest one and make the action conservative.

Guiding questions before output: what should the Reviewer decide after this canvas; which uncertainty axis is dominant; does the window contain the needed onset/peak/trough without ruled-out history; and are M and C sufficient to test the Supervisor's hypothesis?

Self-check: every stream in M must have a visible role, every id in C must be eligible under h_t .candidates, and gamma must be renderer-interpretable. For comparison include suspected and contrast candidates; for evidence_panel include channels that could confirm or falsify the reason field.

Figure 9: The four LLM role cards used by Knsight: Policy prompt, continued.

Reviewer V system prompt

You are the Reviewer. Inspect the rendered canvas image together with its structured-feature JSON, then decide whether the loop should stop, refine the candidate set, or zoom into a narrower time interval. Base the decision only on visible evidence and the provided feature summary; do not rely on hidden telemetry, priors, or unstated domain knowledge. Check whether the canvas answers the question posed by `h_t.stage`, and separate visual salience from causal sufficiency: a large anomaly is not enough unless its timing, component identity, or propagation relation supports the diagnosis.

Input: {`h_t`, `canvas_png` (image block), `canvas_features` (json block)}

Output: strict JSON only

```
{ "decision"      : "STOP" | "REFINE" | "ZOOM",
  "reason"        : "<evidence-grounded reason, max 2 sentences>",
  "next_C"        : [<candidate ids>] | null,
  "next_window"   : [l, r] | null }
```

Decision rules:

STOP only when the answer is identifiable, the step budget is exhausted, or all five stages have completed.

REFINE when the top candidate is separable from the median candidate by a strong visual/statistical margin, e.g., anomaly-density ratio $\geq 3:1$.

REFINE when downstream candidates can be ruled out by onset order, topology, or a visible propagation wavefront.

ZOOM when the relevant evidence is localized in time but the current view is too coarse to determine onset, peak, trough, or causal ordering.

ZOOM when a candidate remains plausible but the evidence is concentrated in a sub-window that needs a tighter rendering.

Never expand `C`, widen `[l, r]`, or cite absent evidence. If the canvas is ambiguous, choose the smallest `next_C` or `next_window` that preserves all plausible explanations and state the uncertainty in reason. When the image and structured features disagree, prefer the structured feature for numeric thresholds but mention the visual ambiguity. When a label or legend is unclear, do not infer beyond the feature JSON. If the canvas repeats information already established by prior canvases, STOP if sufficient or REFINE to the smallest still-plausible candidate set.

Reason requirements: reason must name the evidence axis that drives the decision: time, component, topology, metric magnitude, trace path, log signature, or explicit counter-evidence. For REFINE, `next_C` must be non-null and must contain only ids from `h_t.candidates`. For ZOOM, `next_window` must be non-null and must be a strict subset when possible. For STOP, both `next_C` and `next_window` must be null unless the step budget is exhausted and an unresolved field must be recorded.

Guiding questions before output: does the canvas answer `h_t.stage` or merely show salience; which candidate is earliest, strongest, or topologically upstream; is evidence whole-window or zoom-localized; and what assumption would be required to stop?

Self-check: STOP requires an identifiable answer or exhausted budget, REFINE requires a defensible smaller candidate set, and ZOOM requires a narrower interval tied to a visible event. The reason must mention decisive evidence and any rejected alternative, not only the most dramatic downstream anomaly.

Figure 9: The four LLM role cards used by Knsight: Reviewer prompt, continued.

Finalizer system prompt

You are the Finalizer. Synthesize the final diagnosis from the complete canvas log after the Reviewer has allowed the loop to stop. Produce the root-cause or change-risk tuple requested by the user, and attach evidence references so that an auditor can trace every field back to one or more rendered canvases. Read the log chronologically: use early canvases for broad search, later canvases for disambiguation, and Reviewer decisions for why alternatives were pruned or zoomed.

Input: {q, canvas_log = [(a_t, feature_summary_t, decision_t)...]}

Output: strict JSON only

```
{ "t_hat"      : "<onset/peak/worst time, ISO-8601 or 'N/A'>",
  "c_hat"      : "<component or candidate id>",
  "r_hat"      : "<root-cause or risk reason, 1 sentence>",
  "evidence_refs" : [<canvas indices supporting t_hat/c_hat/r_hat>],
  "recommendation" : "<1-3 actions derived from the cited evidence>" }
```

Field construction:

t_hat should be the onset, peak, trough, or worst-case time requested by q; use N/A only when q has no time field or the canvases cannot support one.

c_hat should be the component or candidate id that remains after Reviewer pruning; keep the id format used in canvas_log.

r_hat should be one concise sentence that links evidence to cause or risk, not a generic incident summary.

evidence_refs must cite the canvas indices that justify t_hat, c_hat, and r_hat; include multiple indices when different fields come from different canvases.

recommendation must be derived from the cited canvases and should be phrased as operational next actions.

Do not add facts that are not present in the canvas log. If the evidence is insufficient, say so in r_hat and make the recommendation conservative.

Keep the answer operational: name the affected component, relevant time, reason, cited canvases, and smallest next action. The recommendation must be justified by the same evidence_refs rather than by generic SRE practice.

If multiple candidates remain, report the leading candidate and the unresolved distinction instead of fabricating certainty.

Audit constraints: every non-N/A field must be recoverable from canvas_log alone. Do not cite a canvas merely because it appears late in the loop; cite it only if it supports the field. Preserve units and timestamps from the feature summaries. If q asks for a change-risk audit rather than a post-incident RCA, use the same tuple format but interpret t_hat as worst-case or rollout-relevant time and r_hat as risk reason. If q asks for mitigation, provide only actions that follow from the evidence, such as pause rollout, narrow blast radius, collect missing trace/log evidence, or monitor a named metric window.

Guiding questions before output: which canvas localized the event, which selected the component, which verified the reason, and are the cited references sufficient for an auditor to reconstruct each field without hidden telemetry?

Self-check: do not average across canvases when one canvas provides the source-of-truth value, and do not turn Reviewer uncertainty into certainty. Keep the payload short but traceable: every time, id, threshold, and risk claim should point to a numbered canvas relevant to the user's task.

Figure 9: The four LLM role cards used by Knsight: Finalizer prompt, continued.

Iteration $t=1$ — global scan

```
# Supervisor output
{ "stage"      : "global-scan",
  "rationale"   : "Two card types x two cadences span the entire risk
                  surface; no prior canvas; full window in scope.",
  "candidates_in": ["SKU-B|peak", "SKU-B|daily", "SKU-A|peak", "SKU-A|daily"],
  "window_in"   : ["2025-11-15", "2026-05-13"] }

# Policy action a_1
{ "s"      : "global-scan", "kappa": "onset_heatmap",
  "M"      : ["metric"],
  "C"      : ["SKU-B|peak", "SKU-B|daily", "SKU-A|peak", "SKU-A|daily"],
  "window" : ["2025-11-15", "2026-05-13"], "gamma": null, "b": false }

# Visioner canvas_1 structured features (attached to PNG)
{ "canvas": "onset_heatmap",
  "cells" : [ {"id": "SKU-B|peak" , "max": 86.90, "p99": 85.84, "z": 4.1},
              {"id": "SKU-B|daily" , "max": 53.77, "p99": 53.63, "z": 1.2},
              {"id": "SKU-A|peak" , "max": 89.97, "p99": 88.62, "z": 8.3},
              {"id": "SKU-A|daily" , "max": 55.03, "p99": 54.69, "z": 1.4} ],
  "post_ecc_equiv_peak" : {"SKU-B": 90.68, "SKU-A": 93.88},
  "ecc_capacity_line"   : 95.833,
  "audit_note"         : "SKU-A|peak alone exceeds the decision margin;
                        daily averages remain low-tail context." }

# Reviewer decision d_1
{ "decision"   : "REFINE",
  "reason"     : "SKU-A|peak dominates with z=8.3 vs median z=1.3 (>3:1).
                  Post-ECC equivalent 93.88% sits 1.95pp under capacity.",
  "next_C"     : ["SKU-A|peak", "SKU-A|daily"],
  "next_window": null,
  "assumption_checked": "No new SKU ids introduced; pruning keeps SKU-A
                        peak for saturation and SKU-A daily for valley." }

# Stage audit
{ "question_answered" : "Which (card,cadence) carries the risk signal?",
  "evidence_axis"     : ["cross-cell z margin", "post-ECC capacity"],
  "kept_ids"          : ["SKU-A|peak", "SKU-A|daily"],
  "rejected_ids"       : ["SKU-B|peak", "SKU-B|daily"],
  "why_not_stop"       : "Need peak dwell, valley shape, and monthly trend
  .",
  "next_needed"        : "Render SKU-A peak-day panel for boundedness.",
  "monotone_check"     : "Candidates shrink 4->2; window stays broad.",
  "audit_status"       : "global scan complete; localization still open." }
```

Figure 10: V-RoT iteration $t=1$ on Case C. The Reviewer prunes the candidate set from four (card, cadence) cells down to the two SKU-A cells.

Iteration $t=2$ — spike-day evidence

```
# Supervisor output
{ "stage"      : "localize",
  "rationale"   : "Localize SKU-A peak day; the daily-average tail is
                  benign (z=1.4) and will be re-examined in t=3.",
  "candidates_in": ["SKU-A|peak"], "window_in": ["2025-12-19"] }

# Policy action a_2
{ "s"      : "localize", "kappa": "evidence_panel",
  "M"      : ["metric"], "C": ["SKU-A|peak"],
  "window" : ["2025-12-19T00:00", "2025-12-19T24:00"],
  "gamma"  : null, "b": false }

# Visioner canvas_2 = canvas_memory_spike.png (Fig. 7)
{ "canvas"      : "evidence_panel|spike",
  "day"         : "2025-12-19",
  "day_peak_pct" : 89.97,
  "day_peak_time" : "14:30",
  "day_median_pct" : 71.50,
  "baseline_p1_p99" : [78.66, 88.62],
  "median_180d" : 85.94,
  "p95_180d" : 88.00,
  "samples_above_mad3sigma" : 24,
  "post_ecc_day_peak" : 93.88,
  "post_ecc_headroom_gb" : 1.41,
  "boundedness_check" : "Peak is above historical P95 but below ECC;
                        dwell time is the residual risk." }

# Reviewer decision d_2
{ "decision" : "ZOOM",
  "reason"   : "Day-peak 89.97% (post-ECC 93.88%) is below the 95.83%
              line but dwell is only 24min/day above MAD-3s; zoom to
              confirm the band is bounded, not a multi-hour
plateau.",
  "next_window": ["2025-12-19T14:26", "2025-12-19T14:34"],
  "self_check" : "Window strictly shrinks from a full day to the 8-minute
              band around the observed peak; candidate set unchanged."
}

# Stage audit
{ "question_answered" : "Is the SKU-A worst-day peak short-lived?",
  "evidence_axis"     : ["day peak", "MAD-3sigma dwell", "headroom GB"],
  "kept_ids"         : ["SKU-A|peak"],
  "residual_risk"     : "Thin margin makes dwell length material.",
  "why_not_stop"      : "Daily low-tail evidence still needs verification
. ",
  "next_needed"       : "Verify holiday valley as normal idle behavior.",
  "monotone_check"    : "Window shrinks 24h->8min; candidate unchanged.",
  "audit_status"      : "localization complete; counter-evidence open." }
```

Figure 11: V-RoT iteration $t=2$. The Reviewer zooms the time window from one full day down to an 8-minute peak band, asking whether the peak is bounded or sustained.

Iteration $t=3$ — valley-day evidence (rule out workload eviction)

```
# Supervisor output
{ "stage"      : "verify",
  "rationale"   : "Peak-day dwell confirmed bounded. Now verify the
                  holiday low is a benign idle, not a quiet eviction.",
  "candidates_in": ["SKU-A|daily"], "window_in": ["2026-01-01"] }

# Policy action a_3
{ "s"      : "verify", "kappa": "evidence_panel",
  "M"      : ["metric"], "C": ["SKU-A|daily"],
  "window" : ["2026-01-01T00:00", "2026-01-01T24:00"],
  "gamma"  : null, "b": false }

# Visioner canvas_3 = canvas_memory_valley.png (Fig. 8)
{ "canvas"      : "evidence_panel|valley",
  "day"         : "2026-01-01",
  "day_trough_pct" : 27.40,
  "day_trough_time" : "04:12",
  "day_mean_pct"  : 42.82,
  "daily_p1_p99"  : [40.13, 54.69],
  "median_180d_daily" : 48.26,
  "holiday_low_180d" : 40.00,
  "low_tail_samples_below_32pct" : 91,
  "valley_shape"   : "pre-dawn-idle+lunch-shoulder",
  "eviction_indicator" : "none",
  "negative_evidence" : "No midday flatline or missing ramp-up;
                        workload drain is unsupported." }

# Reviewer decision d_3
{ "decision" : "REFINE",
  "reason"   : "Valley matches canonical holiday idle (pre-dawn dip +
                lunch shoulder); no flatline indicating eviction. Risk
                class shrinks to 'peak-day saturation only'.",
  "next_C"   : ["SKU-A|peak"],
  "next_window": null,
  "assumption_checked": "The daily-average trough is explanatory context,
                        not an independent safety blocker." }

# Stage audit
{ "question_answered" : "Is the holiday low benign idle?",
  "evidence_axis"     : ["trough shape", "daily baseline", "negative
signal"],
  "kept_ids"          : ["SKU-A|peak"],
  "rejected_risk"      : "quiet workload eviction during the holiday valley
",
  "why_refine"         : "Valley shape is normal; daily-low risk is
rejected.",
  "next_needed"        : "Cross-check monthly aggregation and trend.",
  "monotone_check"     : "Candidate set returns to one remaining risk class
.",
  "audit_status"       : "counter-evidence complete; stop-check still open
." }
```

Figure 12: V-RoT iteration $t=3$. The valley canvas rules out workload eviction and leaves one risk class.

Iteration $t=4$ — monthly aggregation cross-check

```
# Supervisor output
{ "stage"      : "stop-check",
  "rationale"   : "One risk class remains; cross-check monthly
                  aggregation to ensure Dec-2025 is not part of a
                  still-climbing trend.",
  "candidates_in": ["SKU-A|peak"], "window_in": ["2025-11", "2026-05"] }

# Policy action a_4
{ "s"      : "stop-check", "kappa": "comparison",
  "M"      : ["metric"], "C": ["SKU-A|peak"],
  "window": ["2025-11-01", "2026-05-13"], "gamma": null, "b": true }

# Visioner canvas_4 = monthly comparison
{ "canvas": "comparison|monthly_peak",
  "months": [
    {"m": "2025-11", "avg": 82.41, "p95": 82.88, "max": 82.89, "head_gb": 3.11},
    {"m": "2025-12", "avg": 81.51, "p95": 85.11, "max": 89.97, "head_gb": 1.41},
    {"m": "2026-01", "avg": 84.99, "p95": 87.98, "max": 88.32, "head_gb": 1.80},
    {"m": "2026-02", "avg": 86.05, "p95": 88.54, "max": 88.63, "head_gb": 1.73},
    {"m": "2026-03", "avg": 86.19, "p95": 87.06, "max": 87.40, "head_gb": 2.02},
    {"m": "2026-04", "avg": 86.70, "p95": 88.10, "max": 88.18, "head_gb": 1.84},
    {"m": "2026-05", "avg": 86.33, "p95": 86.72, "max": 86.73, "head_gb": 2.18}
  ],
  "days_ge_90pct"      : 0,
  "days_ge_95_83pct"   : 0,
  "trend_slope_pp_per_30d" : 0.985,
  "stability_check"     : "Dec-2025 is worst; later months stay below
                          the ECC line and below 90%." }

# Reviewer decision d_4
{ "decision" : "STOP",
  "reason"   : "Zero days >= 90% across full series; monthly P95 trend
               slope +0.99 pp/30d but min headroom never dips below
               1.41 GB. Risk is bounded; emit answer.",
  "next_C"   : null, "next_window": null,
  "stop_basis" : "t=canvas_2; candidate=canvases_1/4; risk=canvases_2/4." }

# Stage audit
{ "question_answered" : "Is the monthly trend still climbing?",
  "evidence_axis"     : ["monthly max", "monthly P95", "days above limits
"],
  "stop_fields"       : {"t_hat": "canvas_2", "c_hat": "canvases_1/4",
                        "r_hat": "canvases_2/4"},
  "rejected_risk"     : "unbounded month-over-month capacity erosion",
  "why_stop"          : "No day reaches 90% or the 95.83% ECC line.",
  "final_answer_type" : "change-risk audit with conservative rollout",
  "monotone_check"    : "No expansion; every tuple field is cited.",
  "audit_status"      : "stop-check complete; finalizer may answer." }
```

Figure 13: V-RoT iteration $t=4$. The monthly comparison confirms the trend slope is bounded and emits STOP; the audit terminates after four canvases.

Finalizer output (ticket payload)

```

{ "t_hat"      : "2025-12-19T14:30",
  "c_hat"      : "24 GB-class GPU (SKU-A)",
  "r_hat"      : "Aggregate peak utilization is 89.97% (93.88%
                  post-ECC equivalent), leaving 1.41 GB headroom;
                  no aggregate day reaches 90%, while per-tenant
                  safety remains unresolved by per-card data.",
  "evidence_refs" : {
    "t_hat" : [2],           // spike canvas: 14:30 tallest excursion
    "c_hat" : [1, 4],        // onset heatmap + monthly comparison
    "r_hat" : [2, 4] },      // raw peak + trend upper-bound
  "risk_breakdown": {
    "ecc_capacity_line_pct"      : 95.833,
    "ecc_capacity_line_gb"      : 23.0,
    "day_peak_raw_pct"          : 89.97,
    "post_ecc_day_peak_pct"     : 93.88,
    "post_ecc_headroom_gb"      : 1.41,
    "p99_corridor_pct"          : [78.66, 88.62],
    "median_180d_peak_pct"      : 85.94,
    "samples_above_mad3sigma"   : 24,
    "zoom_window_minutes"       : 8,
    "holiday_intraday_trough_pct" : 27.40,
    "holiday_intraday_trough_time": "04:12",
    "holiday_daily_low_pct"     : 40.00,
    "monthly_slope_pp_per_30d"  : 0.985,
    "days_ge_90pct"           : 0,
    "days_ge_95_83pct"         : 0,
    "scope_limit" : "Per-card aggregates do not establish
                    per-tenant maxima." },
  "audit_trace": [
    "t=1 REFINE: four (card,cadence) cells over 180 days.
                SKU-A|peak has z=8.3; Reviewer keeps the two
                SKU-A cells and prunes the two SKU-B cells.",
    "t=2 ZOOM:   spike canvas 2025-12-19. Peak 89.97% at 14:30,
                24 one-minute samples exceed MAD-3sigma; Reviewer
                zooms to the 8-minute window [14:26,14:34].",
    "t=3 REFINE: holiday intraday trough is 27.40% at 04:12;
                the daily aggregate low is 40.00%. Shape shows
                benign idle with no eviction indicator.",
    "t=4 STOP:   Nov-2025--May-2026 monthly comparison. Slope
                is +0.985 pp/30d, with zero aggregate days >=90%
                or >=95.83%. Reviewer issues STOP." ],
  "recommendation": [
    "1. Run a per-tenant trailing-30d peak scan before rollout;
        per-card aggregates cannot exclude tenant outliers.",
    "2. Update scheduler resource descriptor 24 GB -> 23 GB
        before enabling ECC.",
    "3. Gray-release SKU-B before SKU-A; monitor post-ECC-equivalent
        utilization and pause before expansion if an observed
        value reaches the 95.83% capacity line." ],
  "recommendation_refs": [[1,2,4], [1,2], [1,4]],
  "confidence"          : "medium",
  "confidence_basis"    : "Four aggregate canvases identify SKU-A/14:30
                          and bounded aggregate headroom; tenant-level
                          maxima remain unresolved.",
  "canvas_count"        : 4,
  "loop_exit_reason"    : "Reviewer STOP @t=4; all fields cited." }

```

Figure 14: Finalizer payload that lands on the change-management audit ticket. The `evidence_refs`, `risk_breakdown`, and `audit_trace` fields make each output replayable from Figs. 10–13 without re-querying the metrics warehouse.