TELEAGENT-LOGS: BENCHMARKING EVIDENCE-GROUNDED AGENTS FOR INTERACTIVE 5G TROUBLESHOOTING

Anonymous authors
Paper under double-blind review

ABSTRACT

Large language model agents offer a promising way to automate telecommunications network operations, improve efficiency, and deliver better services to end users, yet realistic benchmarks for network troubleshooting remain scarce. In telecommunication networks, effective diagnosis requires more than predicting a root cause: an engineer must identify where and when degradation occurs, analyze heterogeneous network logs, gather supporting evidence, and determine actionable remediation. We introduce TeleAgent-Logs, an agentic benchmark that reformulates network fault diagnosis and remediation as an interactive decision-making problem. Each task is based on a network drive test and exposes five complementary data sources: user-plane data, signaling-plane data, measurement-report data, network-configuration data, and traffic data—through a fixed set of diagnostic tools for both data access and radio and network analysis. Agents must decide which observations to inspect, which tools to invoke, and how to combine their results into structured fault hypotheses covering four capabilities: localization, fault identification, evidence grounding, and remediation. We consider two evaluation settings that require either one or all evidence-supported fault hypotheses, and introduce taxonomy-aware matching together with component-wise evaluation of each troubleshooting stage. By shifting evaluation from static answer selection to sequential investigation, analysis, and corrective actions, the benchmark measures the full troubleshooting loop and provides a challenging testbed for tool-using agents in 5G network operations. We evaluate four models across five harnesses and observe substantial performance variation across model–harness combinations. Overall, agents are better at locating degradations and identifying faults than they ground diagnoses or propose correct remediation. Finally, our comparison across fault categories highlight systematic confusion between related radio faults such as antenna tilt, transmission power, and mobility configuration.

1 INTRODUCTION

Modern cellular networks are complex, highly configurable systems whose performance depends on the interaction of radio conditions, mobility procedures, network configuration, traffic load, and external infrastructure. Diagnosing network performance degradation therefore rarely reduces to identifying a single anomaly. In practice, network troubleshooting requires an engineer to determine *when and where* performance degrades, inspect multiple sources of network logs, relate protocol events to radio measurements and configuration, formulate candidate root causes, and identify an appropriate corrective action. This process is inherently sequential: each observation informs which data should be inspected or which analysis should be performed next.

Recent progress in large language models (LLMs) has motivated growing interest in agents that can interact with tools, gather information, and perform multi-step reasoning. Such capabilities are particularly relevant to network operations, where raw observations are distributed across heterogeneous data sources and many useful signals must be obtained through targeted analysis rather than direct inspection alone. Existing agent benchmarks primarily evaluate whether an agent can achieve a final task objective. TeleAgent-Logs instead decomposes troubleshooting into acquisition, localization, diagnosis, evidence grounding, and remediation, allowing us to study how agent failures

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

propagate across a structured diagnostic process. The benchmark uses a real-world anonymized network topology and fault-frequency distribution, while the network logs and fault scenarios are synthetically generated. Importantly, we construct troubleshooting tasks in which the underlying network data are hidden behind a controlled tool interface. Each task corresponds to a scenario in which a user traverses a region covered by multiple gNodeBs[1]. The environment exposes five complementary sources of information: user-plane data, signaling-plane data, measurement-report data, network-configuration data, and traffic data.

A first research question we tackle in this work is: *how does interactive information acquisition affect troubleshooting performance compared with fully observed diagnosis?* TeleAgent-Logs provides a fixed set of tools for data access, cell and identifier resolution, and radio/network analysis. The agent must therefore decide both *what to inspect* and *which computations to perform* as its investigation progresses.

A second research question we address is: *where in the troubleshooting pipeline do current agents fail?* Our formulation extends beyond conventional root-cause classification. We view troubleshooting as an end-to-end task comprising four coupled capabilities: **localization**, identifying the affected time interval and network entities; **identification**, determining the fault category and subcategory; **evidence grounding**, supporting the diagnosis with observations obtained from network data; and **remediation**, proposing the corrective action associated with the identified fault. This distinction is important: an agent that predicts the right fault without locating it or grounding it in the observed network state is qualitatively different from one that reconstructs the diagnostic path and proposes an actionable fix.

A further challenge is that a single performance symptom may admit more than one plausible diagnosis, any one of which may be sufficient to recover from the symptom. We therefore consider two evaluation settings. In *Discovery mode*, the evaluation verifies if the agent is able to provide one valid fault hypothesis. In *Complete mode*, the evaluation requires the agent to recover the full set of evidence-supported fault hypotheses associated with the task. To evaluate the latter, we introduce a taxonomy-aware hierarchical matching procedure that first aligns predictions with reference answers sharing the same category–subcategory identity and then allows partial matching only within the same fault category. This prevents predictions from receiving credit through accidental overlap in time or evidence when they correspond to semantically unrelated fault categories.

The resulting benchmark evaluates not only whether an agent arrives at a plausible final diagnosis, but whether it can carry out the complete troubleshooting loop under an interaction budget. This setting introduces several sources of difficulty: the agent must select useful tool calls, integrate asynchronous and heterogeneous network observations, reason across multiple cells and time scales, distinguish symptoms from causes, and translate a diagnosis into a structured remediation.

We address the following research questions and provide the related contributions:

- We introduce an agentic formulation of 5G network troubleshooting transforming static diagnostic questions into interactive tasks in which data are accessible only through a controlled tool interface.
- We design an environment exposing five complementary network-data sources together with tools for data access, identifier resolution, and radio/network analysis, requiring agents to actively investigate each task rather than reason from a fully observed prompt.
- We formulate end-to-end troubleshooting around four structured capabilities–localization, fault identification, evidence grounding, and remediation–thereby evaluating both root-cause analysis and the ability to act on the resulting diagnosis.
- We introduce Discovery and Complete evaluation settings together with taxonomy-aware fault-hypothesis matching and component-wise scoring, enabling fine-grained analysis of where agentic troubleshooting systems succeed or fail.

[1] A base station providing 5G radio connectivity to mobile users.

2 Related Work

LLM agents for interactive diagnosis. LLM agents increasingly solve tasks by interleaving reasoning with actions whose observations guide subsequent decisions (Yao et al., 2022). Agent evaluations show that performance depends not only on the underlying model but also on the interface through which it acts (Yang et al., 2024), and that success can vary substantially across repeated trials of the same task (Yao et al., 2024). Operational diagnosis is a natural test of these capabilities. OpenRCA evaluates agents that identify the root causes of software failures from logs, metrics, and traces (Xu et al., 2025), while AIOpsLab and ITBench provide interactive environments for incident localization, diagnosis, and mitigation in cloud and IT systems (Chen et al., 2025; Jha et al., 2025). These environments target software and IT infrastructure. Radio access networks pose a different challenge: degradations arise from the interaction of radio propagation, mobility procedures, cell configuration, and traffic load, and must be diagnosed from asynchronous telemetry spanning multiple cells and time scales.

Benchmarks for network troubleshooting and automation. Within telecommunications, TeleLogs (Sana et al., 2025) evaluates root-cause reasoning over 5G troubleshooting data provided directly to the model, and TeleCom-Bench covers industrial telecom workflows including tool invocation, root-cause analysis, and solution generation (Xiao et al., 2026). These benchmarks measure whether a model reaches a correct answer, but not whether it can acquire the information needed to support it. Interactive network benchmarks address this gap. NIKA lets agents diagnose reproducible incidents through diagnostic interfaces (Wang et al., 2025), NetOpsBench exposes live observations from fault scenarios in AI infrastructure networks (Yang & Xu, 2026), and NetArena generates tasks dynamically across network automation applications (Zhou et al., 2026b). Configuration-focused benchmarks address the neighboring problems of synthesizing network configurations (Wang et al., 2024) and verifying configuration repairs (Protogeros et al., 2026). Closest to our setting, CTBench evaluates agents that gather evidence for telecom root-cause analysis and path restoration, scoring localization, identification, and evidential support (Yan et al., 2026). TeleAgent-Logs differs from these benchmarks in three respects: it targets radio access network degradations that must be diagnosed across five heterogeneous drive-test data sources; it evaluates remediation as structured parameter corrections, credited only when the fault is correctly identified; and it admits multiple valid fault hypotheses per task, assessed through Discovery and Complete modes with taxonomy-aware matching.

Specialized LLMs for telecommunications. The domain knowledge required for network operations has motivated telecom-specialized language models. OTel post-trains models for context-grounded telecom answers, together with dedicated retrieval and reranking models (Tavakkoli et al., 2026), and TelecomGPT-R1 trains a reasoner across protocol, knowledge, network-modeling, and fault-related tasks (Wang et al., 2026). For 5G diagnosis specifically, the work introducing TeleLogs combines supervised fine-tuning and reinforcement learning to improve root-cause prediction and structured explanations (Sana et al., 2025), while Zhou et al. (2026a) structure telemetry and constrain the reasoning path to ground diagnoses in network evidence. These approaches are evaluated primarily on questions or fully observed diagnostic inputs. Whether domain specialization transfers to interactive investigation, where the model must also decide what to observe, remains open.

3 TeleAgent-Logs: An Agentic Benchmark for 5G Troubleshooting

TeleAgent-Logs is an agentic benchmark for troubleshooting performance degradations in 5G radio access networks. To develop TeleAgent-Logs, we consider a real network topology that has been anonymized for privacy. On top of this topology, we construct a set of possible network faults, with the occurrence probability of each fault following a distribution derived from real network data. Each benchmark instance is then generated from a synthetic scenario designed to mimic a real drive-test in which a user equipment (UE) mounted on a moving vehicle traverses a geographic region served by multiple gNodeBs. For each instance, a subset of the constructed faults is injected according to the underlying fault distribution. To further emulate the drive test, we generate telemetry related to multiple layers of the network, including radio measurements, protocol events, cell configuration, and network load.

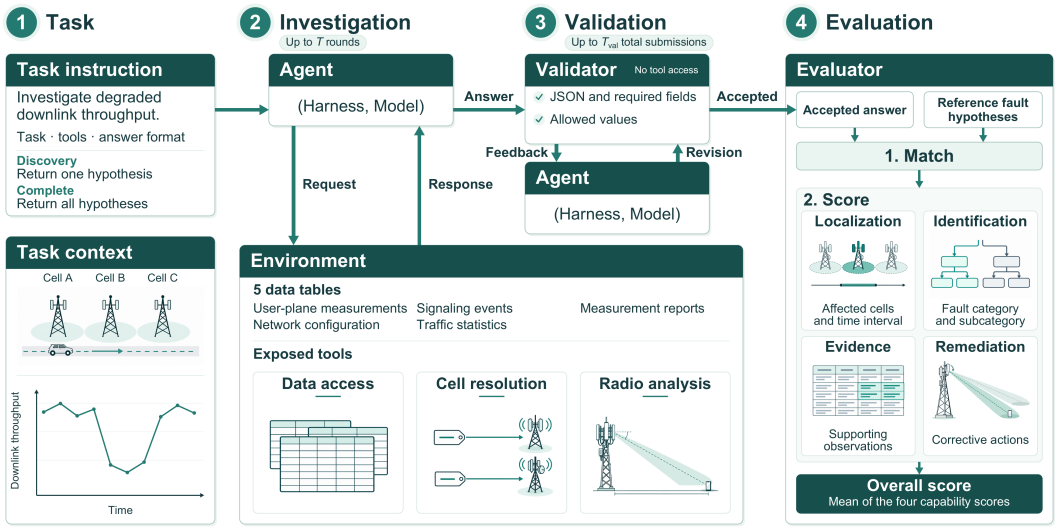

Figure 1: Benchmark overview.

In total, the dataset comprises 500 tasks (Appendix Table 3). Before inclusion in the benchmark, 3 expert telecommunications engineers, not involved in the benchmark creation, reviewed the data and reference fault hypotheses, checking the consistency of the observations and whether additional plausible fault explanations and remediation were apparent. Ambiguous or inconsistent benchmark instances were revised or removed. Human reviewers assessed each task contains five primary data sources:[2]

1. **User-plane data**, containing timestamped UE-level measurements such as serving PCI, ARFCN, RSRP, SINR, throughput, MCS, rank, and neighboring-cell measurements;
2. **Signaling-plane data**, containing timestamped radio and mobility events such as measurement configurations, measurement reports, random access, handover attempts, and RRC procedures;
3. **Measurement-report (MR) data**, containing serving- and neighboring-cell radio measurements together with observed user performance;
4. **Network-configuration data**, describing the deployed cells and their configuration, including PCI, ARFCN, antenna parameters, transmission power, mobility thresholds, and neighbor relationships; and
5. **Traffic data**, containing cell-level load and capacity indicators such as PRB utilization, CCE utilization, throughput, interference, and allocation success rates.

Each task is instantiated as an interactive troubleshooting environment. The evaluated agent must decide which information to retrieve, which cells or time periods to inspect, and which radio analyses to perform using a fixed set of 14 tools (see details in Appendix Table 4). These tools cover telemetry access, cell and identifier resolution, and radio/network analysis. Consequently, performance depends not only on the final diagnosis, but also on the agent's ability to actively acquire the information needed to support it. The investigation proceeds over multiple turns, up to a fixed maximum number of iterations. At each turn, the agent reasons over the available information and may conclude the turn by issuing a tool call, after which the environment returns the corresponding output. Figure 1 summarizes the investigation and evaluation workflow.

3.1 Problem Formulation

We formulate network troubleshooting as an interactive decision-making problem. Let π_θ denote the evaluated agent, comprising both the underlying language model with parameters θ and its agent

[2]Appendix A.1 provides explanation of the telecommunication terms used in this work.

harness. Each benchmark instance is associated with an underlying task $s \in \mathcal{S}$ and a corresponding task instruction q_s.

The task instruction describes the troubleshooting objective and the required output format, but does not directly reveal the underlying telemetry. The agent can access the task data only through the tools exposed by the environment.

At interaction step t, the agent selects an action conditioned on the task instruction and its previous interaction history:

$$a_t \sim \pi_\theta(\cdot \mid q_s, h_{t-1}), \tag{1}$$

where $h_{t-1} = (a_1, o_1, \ldots, a_{t-1}, o_{t-1})$ denotes the sequence of previous actions and observations.

An action a_t corresponds to a diagnostic operation exposed by the benchmark environment, such as retrieving telemetry over a specified time interval, inspecting a cell configuration, resolving a cell identifier, or requesting a radio-analysis operation.

The environment $\mathcal{E}$ executes the requested action and returns an observation

$$o_t = \mathcal{E}(s, h_{t-1}, a_t). \tag{2}$$

An observation may contain a subset of user-plane, signaling-plane, MR, network-configuration, traffic data, or the result of an analysis tool, depending on the requested operation. Invalid or unavailable operations return an explicit error observation.

The agent continues interacting with the environment until it voluntarily terminates or reaches the maximum investigation budget T. Let ℓ denote the number of investigation rounds completed. The resulting trajectory is

$$\mathcal{T}(q_s) = (q_s, a_1, o_1, a_2, o_2, \ldots, a_\ell, o_\ell), \qquad \ell \le T. \tag{3}$$

At termination, the agent produces a structured final answer $\hat{y}_s$ containing fault hypotheses supported by the inspected observations.

3.2 Troubleshooting Objective

TeleAgent-Logs evaluates end-to-end troubleshooting rather than root-cause classification alone. For each task, the agent must determine *where and when* the degradation occurs, infer *why* it occurs, support its diagnosis with observed data, and propose an appropriate remediation.

Each task is associated with a reference answer represented as a non-empty set of reference fault hypotheses,

$$y_s = \{p_1, \ldots, p_m\}, \qquad m \ge 1, \tag{4}$$

with each hypothesis

$$p_i = (L_i, I_i, E_i, R_i), \tag{5}$$

corresponding to four troubleshooting capabilities:

- **Localization L_i.** The agent must identify the affected portion of the drive test, including both the temporal interval and the implicated network entities. Specifically,
$$L_i = (\tau_i, \mathcal{A}_i), \tag{6}$$
where $\tau_i = [t_i^{\text{start}}, t_i^{\text{end}}]$ is the fault interval and $\mathcal{A}_i$ is the set of affected cells or other network entities.
- **Identification I_i.** The agent must identify the underlying root cause according to the benchmark taxonomy. Mathematically,
$$I_i = (c_i, z_i), \tag{7}$$
where c_i is the fault category and z_i the corresponding subcategory. Note that each category–subcategory pair uniquely identifies a reference fault hypothesis within each task.
- **Evidence grounding E_i.** The agent must support the diagnosis using observations present in the task data. E_i denotes the set of supporting evidence items, represented as table–row–column triplets, each identifying a data table, a row selector, and a column.
- **Remediation R_i.** The agent must propose an actionable correction addressing the identified fault. R_i is a set of proposed corrections, each specifying a parameter, an operation, target entities, and, where applicable, target values. External or inconclusive diagnosis may require no network-side remediation.

The evaluated agent returns predictions in the same structured form, i.e., a set of fault hypotheses:

$$\hat{y}_s = \{\hat{p}_1, \ldots, \hat{p}_k\}. \tag{8}$$

This formulation evaluates the full troubleshooting loop from fault localization and diagnosis to evidence grounding and corrective action.

Before scoring, a validator checks that $\hat{y}_s$ satisfies the answer requirements (e.g., valid JSON, permitted field values, references to existing observations, and no repeated category–subcategory pairs), without assessing diagnostic correctness. The agent receives feedback on violations and may revise its answer without further tool access, up to T_{val} total submissions per attempt, including the initial answer (see the validation block in Figure 1). Appendix A.2 presents a worked investigation that includes an example of validation feedback and answer revision.

The benchmark covers 15 parameterized remediation patterns spanning radio coverage, mobility, neighborhood configuration, control-channel capacity, and non-radio failure modes. Appendix Table 5 presents the taxonomy of the complete remediation patterns in TeleAgent-Logs.

3.3 Evaluation Settings

Since network symptoms in TeleAgent-Logs may admit multiple plausible diagnoses, we consider two evaluation settings that differ in the required number of returned fault hypotheses.

Discovery mode. The agent objective is to discover *one* supported fault hypothesis from the reference set of fault hypotheses; therefore, the agent must return exactly one fault hypothesis, i.e., $|\hat{y}_s| = 1$.

Complete mode. The agent must provide all correct fault hypotheses for the task, with $|\hat{y}_s| \geq 1$. This setting evaluates both diagnostic precision and coverage, since returning only one valid explanation is insufficient when the reference answer contains multiple fault hypotheses.

3.4 Scoring Metrics

For each task, using the category–subcategory pair, we first determine a matched reference – prediction fault hypothesis pair, denoted as $p = (L, I, E, R)$ and $\hat{p} = (\hat{L}, \hat{I}, \hat{E}, \hat{R})$, respectively. We then compute four component scores evaluating the diagnostic process. This allows not only to understand whether an off-the shelf agent can solve a network troubleshooting task but importantly also to identify why it fails in its duty.

Localization. Localization evaluates both the temporal fault interval and the affected cells. Let τ and $\hat{\tau}$ denote the reference and predicted time intervals, and let $\mathcal{A}$ and $\hat{\mathcal{A}}$ denote the corresponding affected-cell sets. We define

$$S_{\text{loc}} = \frac{1}{2}\,\text{IoU}(\tau, \hat{\tau}) + \frac{1}{2}\,\text{IoU}(\mathcal{A}, \hat{\mathcal{A}}), \tag{9}$$

where IoU represents the intersection-over-union operator, i.e., $\text{IoU}(A, B) = |A \cap B|/|A \cup B|$.

Identification. Identification is decomposed into category and subcategory correctness:

$$S_{\text{id}} = 0.75 \cdot \mathbb{1}[\hat{c} = c] + 0.25 \cdot \mathbb{1}[\hat{z} = z]. \tag{10}$$

This weighting places greater emphasis on identifying the correct fault category while still requiring fine-grained fault discrimination.

Evidence. We evaluate evidence quality as the IoU of the predicted and reference evidence set:

$$S_{\text{evid}} = \text{IoU}(E, \hat{E}). \tag{11}$$

Remediation. Remediation is conditioned on correct fault identification. We compare the predicted and reference correction sets using an F1-like score:

$$S_{\text{rem}} = \begin{cases} 0, & (\hat{c}, \hat{z}) \neq (c, z), \\ \text{F1}(R, \hat{R}), & \text{otherwise.} \end{cases} \tag{12}$$

Appendix A.4 defines the correction matching, pair scores, and F1 aggregation.

Fault score. The fault score for one matched fault hypothesis is

$$S(p, \hat{p}) = \frac{1}{4}\left(S_{\text{loc}} + S_{\text{id}} + S_{\text{evid}} + S_{\text{rem}}\right). \tag{13}$$

The weighting reflects that domain experts prioritize these component metrics independently as primary metrics, with the aggregated score serving as a secondary metric.

Table 1: Agent scores by configuration (%, mean ± mean SD).

	Localization		Identification		Evidence		Remediation		Success Score	
	Discovery	Complete	Discovery	Complete	Discovery	Complete	Discovery	Complete	Discovery	Complete
Claude harness										
DeepSeek	96.9±3.3	56.9±15.0	64.0±11.7	51.3±13.3	25.6±6.7	15.5±6.3	34.0±9.0	25.1±8.1	55.1±6.8	37.2±10.0
Gemma	99.0±2.0	44.5±11.3	48.0±8.3	39.8±9.8	18.3±6.3	6.9±3.5	27.9±4.8	21.0±4.8	48.3±4.8	28.1±7.1
Muse	71.3±36.5	38.6±32.4	43.1±31.7	33.7±27.9	13.6±9.6	8.1±7.6	19.8±13.6	14.6±12.0	37.0±21.1	23.8±19.4
Qwen	99.1±2.1	58.1±16.5	64.8±15.4	54.7±16.0	24.4±5.9	13.6±6.3	38.4±15.4	31.3±14.7	56.7±8.5	39.4±12.4
Mean	**91.6**±11.0	**49.6**±18.8	**55.0**±16.8	**44.9**±16.8	**20.5**±7.1	**11.0**±5.9	**30.0**±10.7	**23.0**±9.9	**49.3**±10.3	**32.1**±12.2
Codex harness										
DeepSeek	85.5±23.7	54.6±26.2	58.5±25.7	48.9±23.3	22.7±9.4	15.3±8.3	29.7±16.2	23.4±12.8	49.1±17.5	35.5±16.9
Gemma	93.4±9.8	35.6±16.2	36.9±15.4	32.6±15.0	15.3±6.4	5.1±3.1	22.2±8.7	18.2±8.0	41.9±8.9	22.9±10.4
Muse	69.3±39.9	39.7±33.0	44.3±32.1	34.0±28.1	15.0±10.5	9.2±8.3	17.6±14.1	13.2±11.5	36.5±22.8	24.0±19.7
Qwen	49.1±48.0	26.2±30.6	31.9±34.6	24.9±28.9	11.8±12.2	6.2±7.6	19.0±21.9	14.7±18.1	27.9±28.2	18.0±20.9
Mean	**74.3**±30.4	**39.0**±26.5	**42.9**±27.0	**35.1**±23.8	**16.2**±9.6	**8.9**±6.8	**22.1**±15.2	**17.4**±12.6	**38.9**±19.3	**25.1**±17.0
DeepSeek harness										
DeepSeek	96.1±5.5	45.9±19.3	61.0±9.3	41.2±17.4	24.8±6.9	13.6±6.9	34.4±6.9	21.6±9.1	54.1±6.5	30.6±12.7
Gemma	99.7±0.7	44.7±9.1	47.4±8.8	40.0±7.8	17.7±5.7	6.5±3.1	27.0±5.9	21.2±4.1	48.0±4.8	28.1±5.8
Muse	90.3±16.5	53.6±22.0	55.5±23.5	47.4±20.9	16.9±6.9	11.1±6.2	24.0±16.5	20.2±14.1	46.7±14.1	33.1±14.9
Qwen	89.7±15.2	42.6±22.0	54.6±19.5	39.4±21.1	22.9±8.6	10.7±6.7	31.8±12.5	22.0±13.6	49.7±12.6	28.7±15.3
Mean	**93.9**±9.5	**46.7**±18.1	**54.6**±15.3	**42.0**±16.8	**20.6**±7.0	**10.5**±5.7	**29.3**±10.5	**21.3**±10.2	**49.6**±9.5	**30.1**±12.2
Hermes harness										
DeepSeek	72.1±27.9	28.2±21.5	49.2±22.0	25.3±19.4	18.5±9.9	7.4±6.2	26.4±13.4	12.9±10.0	41.6±17.4	18.5±14.1
Gemma	97.1±4.6	36.8±12.0	37.2±14.4	33.5±11.0	16.6±6.1	5.6±2.8	21.9±7.5	18.4±5.5	43.2±7.1	23.6±7.7
Muse	90.5±16.0	53.2±23.9	56.1±24.5	47.3±22.5	17.2±7.2	11.0±6.3	24.7±17.1	20.5±15.4	47.1±14.3	33.0±16.0
Qwen	48.5±35.5	12.9±15.0	25.1±21.2	12.6±14.5	11.8±10.6	2.7±3.6	17.1±12.7	8.8±8.9	25.6±19.5	9.3±10.4
Mean	**77.1**±21.0	**32.8**±18.1	**41.9**±20.5	**29.7**±16.9	**16.1**±8.4	**6.7**±4.7	**22.5**±12.7	**15.2**±10.0	**39.4**±14.6	**21.1**±12.1
ReAct harness										
DeepSeek	79.0±27.4	31.5±24.8	51.2±22.4	28.4±22.0	20.7±10.2	8.6±7.6	29.5±12.9	15.5±11.3	45.1±17.4	21.0±16.1
Gemma	91.5±8.8	33.1±13.1	37.2±13.2	30.6±11.8	19.7±5.4	5.7±2.8	24.2±8.1	18.4±6.6	43.1±8.2	22.0±8.4
Muse	85.5±24.0	51.2±25.4	53.6±27.1	44.5±22.5	18.4±8.7	11.9±7.2	23.0±15.0	18.5±12.2	45.1±16.7	31.5±16.0
Qwen	88.1±21.3	44.3±25.5	53.6±26.0	41.5±23.7	21.4±9.4	9.8±6.9	30.7±17.7	23.0±14.6	48.4±16.9	29.6±17.1
Mean	**86.0**±20.4	**40.0**±22.2	**48.9**±22.2	**36.3**±20.0	**20.1**±8.4	**9.0**±6.1	**26.8**±13.4	**18.8**±11.2	**45.5**±14.8	**26.0**±14.4
Model averages										
DeepSeek	**85.9**±17.6	**43.4**±21.3	**56.8**±18.2	**39.0**±19.1	**22.5**±8.6	**12.1**±7.1	**30.8**±11.7	**19.7**±10.3	**49.0**±13.1	**28.6**±14.0
Gemma	**96.1**±5.2	**39.0**±12.4	**41.3**±12.0	**35.3**±11.1	**17.5**±6.0	**6.0**±3.1	**24.6**±7.0	**19.4**±5.8	**44.9**±6.8	**24.9**±7.9
Muse	**81.4**±26.6	**47.3**±27.4	**50.5**±27.8	**41.4**±24.4	**16.2**±8.6	**10.3**±7.1	**21.8**±15.3	**17.4**±13.0	**42.5**±17.8	**29.1**±17.2
Qwen	**74.9**±24.4	**36.8**±21.9	**46.0**±23.3	**34.6**±20.8	**18.5**±9.3	**8.6**±6.2	**27.4**±16.1	**20.0**±14.0	**41.7**±17.1	**25.0**±15.2
All	**84.6**±18.4	**41.6**±20.7	**48.7**±20.3	**37.6**±18.9	**18.7**±8.1	**9.2**±5.9	**26.2**±12.5	**19.1**±10.8	**44.5**±13.7	**26.9**±13.6

Discovery mode matching. In TeleAgent-Logs, a task may include multiple reference faults. Therefore, in Discovery mode, where a single prediction is returned, the matching reference fault index i^* is selected based on the highest identification score, with the highest fault score used as a tie-breaker when two or more faults have the same identification score.

The success score per task in Discovery mode is

$$S_s^{\text{disc}} = S(p_{i^*}, \hat{p}_1). \tag{14}$$

Complete mode taxonomy-aware matching. In Complete mode, the reference and predicted fault hypotheses form unordered sets, with $m = |y_s|$ and $k = |\hat{y}_s|$. For each benchmark instance, the validator ensures that no two predicted fault hypotheses share the same category–subcategory pair. We construct a deterministic one-to-one matching $\mathcal{M}_s \subseteq \{1, \ldots, m\} \times \{1, \ldots, k\}$ hierarchically. We first match fault hypotheses with identical category–subcategory pairs. We then match the remaining fault hypotheses only within the same category, using a lexicographic objective that (i) maximizes the number of pairs, (ii) maximizes their summed per-fault score, and (iii) breaks ties by reference order and then prediction order. Unmatched references and predictions remain as penalties.

The success score per task is

$$S_s^{\text{comp}} = \frac{2 \sum_{(i,j) \in \mathcal{M}_s} S(p_i, \hat{p}_j)}{m + k}, \tag{15}$$

which penalizes both missing reference hypotheses and extra predictions.

Benchmark success score. Finally, the benchmark success score is the macro-average over the tasks:

$$S_{\text{benchmark}} = \frac{1}{|\mathcal{S}|} \sum_{s \in \mathcal{S}} S_s. \tag{16}$$

Table 2: Scores and fault hypothesis coverage.

Reference fault hypotheses	Tasks	Success Score (%)		Predicted fault hypotheses in Complete mode (%)				
		Discovery	Complete	Failure	1	2	3	4+
1	326	39.0	24.7	23.7	71.2	5.0	0.2	< 0.1
2	160	54.1	31.1	18.2	77.6	4.0	0.2	< 0.1
3	14	63.5	30.5	21.1	63.9	14.4	0.6	0.0

Failures include mainly investigation budget failures, validation failures, context/token limit, timeout and no response.

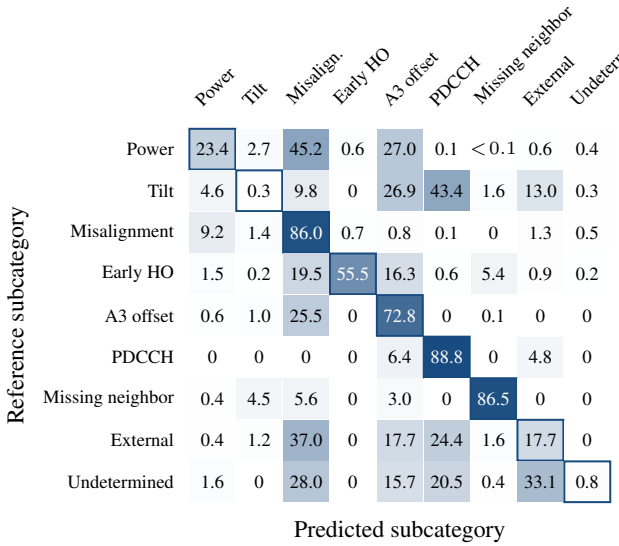

Figure 2: Fault identification confusion matrix computed in Discovery mode on single-reference fault tasks (see details in Appendix A.6). Scores are conditioned to the validation of the provided answers.

4 Experiments and Analysis

4.1 Experimental Setup

We compare four models (DeepSeek-V4-Flash-0731, Gemma-4-31B-IT, Muse-Glimmer-30B, and Qwen3.8-27B) using Claude, Codex, DeepSeek, Hermes, and a basic ReAct-style harness. Each configuration is evaluated in both Discovery and Complete modes on all 500 tasks, with five attempts per task and mode. We also evaluate OTel-2.0-LLM-31B-IT, a telecom-specialized Gemma 4 31B-IT derivative, using Claude in both modes and compare its results with Gemma 4 in Appendix A.7. To account for the agent stochastic variability, we report, for each scoring metric, the mean and mean standard deviation computed over 5 independent runs. The full experiment setup is reported in Appendix A.5. Appendix A.9 provides an analysis of the reasons that lead the agents under test to fail the tasks in TeleAgent-Logs. For completeness, Appendix A.10 also reports results conditioned on successful benchmark runs without failures. We wish to highlight that some fault types are represented by only a few samples in TeleAgent-Logs. We retain them to reflect their occurrence in real-world network data; however, the following analysis primarily focuses on the most frequent and challenging faults.

4.2 Performance Analysis

Table 1 presents configuration-level results alongside averages by harness and model. Together with the success score, we also report localization, identification, evidence, and remediation scores to expose differences in agent capabilities. The results confirm that performance in Complete mode is worse than those in Discovery mode. Indeed, Table 2 shows that off-the-shelf agents are relatively good at identifying at least one of the network faults in the task but rarely match all the correct fault hypotheses. In addition, it also highlights that performance in Discovery mode increases substantially with the number of reference hypotheses.

In addition, Table 1 highlights that the agent performance depends strongly on the model–harness combination. For instance, in Discovery mode, Qwen gets a success score of 56.7% with Claude Code and 25.6% with Hermes, whereas Muse scores 37.0% and 47.1%, respectively. This ranking reversal also occurs in Complete mode, showing that a harness that works well with one model may not work equally well with another. Importantly, similar success scores can also conceal different consistency across repeated investigations: Gemma and Muse achieve Discovery success scores of 44.9% and 42.5%, but their mean standard deviations are 6.8 and 17.8 percentage points, highlighting the larger stochastic behaviour of Muse in TeleAgent-Logs.

Focusing on the harness, we show that Claude Code and DeepSeek harness obtain the highest success score across the models under test, achieving respectively a success score of 32.1% and 30.1% in Complete mode. Moreover, analysing the resource usage per task presented in Appendix A.8, we observe that DeepSeek harness is characterized by the lowest number of total tokens, ReAct provides the smallest latency and requires the smallest number of investigation rounds.

Our results also highlight a general capability pattern: agents perform well in fault localization and identification while struggle to provide grounded reasoning evidence (18.7% and 9.2% in Discovery and Complete modes, respectively) and appropriate fault remediation (26.2% and 19.1% in Discovery and Complete modes, respectively) Performance also varies by fault type (see details in Appendix A.6). Our results show that the average agent success score for coverage and mobility faults is only 20.7% and 34.8% (in Complete mode) while it reaches 63.3% for control channel faults. Interesting, the score for coverage-related faults conceals a substantial contrast between the ability of agents to remediate to issues of antenna tilt (success score of 7.8%) and antenna misalignment (success score of 44.2%).

Figure 2 confirms that faults related to antenna tilt are rarely identified, whereas agents demonstrate higher success in identifying faults related to antenna misalignment, A3 offset, and PDCCH capacity. Notably, among the most frequent and challenging network faults, transmission power is often confused with antenna misalignment and A3 offset issues, while antenna tilt is frequently confused with A3 offset and PDCCH capacity.

5 Conclusion

We presented TeleAgent-Logs, a novel benchmark to measure agent capabilities for realistic telecom network troubleshooting. Unlike static root-cause prediction, the benchmark requires agents to actively acquire heterogeneous network observations, localize degradations, identify their causes, ground their diagnoses in evidence, and propose corrective actions. Grounded on fine-grained data annotations and multi-dimensional metrics, TeleAgent-Logs enables separating diagnostic process from final answers, which reveals key gaps in fault identification, remediation, and evidence extraction. We observed that agents can often identify at least one plausible fault in Discovery mode, but struggle to recover all supported fault hypotheses in Complete mode. Experiments with representative agent-model combinations reveal that current agents exhibit substantial gaps in evidence-grounded diagnosis and remediation, especially in solving faults related to antenna tilts. Future extensions of TeleAgent-Logs will integrate a realistic network simulator for testing active closed-loop system interaction and multi-domain network data with a larger number of tool set. We hope TeleAgent-Logs will support future research on trustworthy, domain-grounded, and operationally safe AI agents for telecommunication networks.

6 AI use Statement

In this work, we use generative AI tools to provide feedback on experiments, implement methods, support qualitative data analysis and writing. We have not used AI tools to interpret results, generate synthetic datasets, assist with translation, clean and reformat datasets, and perform other required-disclosure tasks that are not applicable to this work. We have reviewed all AI-assisted work. We take responsibility for the final content of this work, including text, claims, and artifacts produced with the aid of generative AI.

REFERENCES

Yinfang Chen, Manish Shetty, Gagan Somashekar, Minghua Ma, Yogesh Simmhan, Jonathan Mace, Chetan Bansal, Rujia Wang, and Saravan Rajmohan. AIOpsLab: A Holistic Framework to Evaluate AI Agents for Enabling Autonomous Clouds. In *Eighth Conference on Machine Learning and Systems*, 2025. URL https://openreview.net/forum?id=3EXBLwGxtq.

Saurabh Jha, Rohan R. Arora, Yuji Watanabe, Takumi Yanagawa, Yinfang Chen, Jackson Clark, Bhavya Bhavya, Mudit Verma, Harshit Kumar, Hirokuni Kitahara, Noah Zheutlin, Saki Takano, Divya Pathak, Felix George, Xinbo Wu, Bekir O Turkkan, Gerard Vanloo, Michael Nidd, Ting Dai, Oishik Chatterjee, Pranjal Gupta, Suranjana Samanta, Pooja Aggarwal, Rong Lee, Jae wook Ahn, Debanjana Kar, Amit Paradkar, Yu Deng, Pratibha Moogi, Prateeti Mohapatra, Naoki Abe, Chandrasekhar Narayanaswami, Tianyin Xu, Lav R. Varshney, Ruchi Mahindru, Anca Sailer, Laura Shwartz, Daby Sow, Nicholas C. M. Fuller, and Ruchir Puri. ITBench: Evaluating AI Agents across Diverse Real-World IT Automation Tasks. In *Forty-second International Conference on Machine Learning*, 2025. URL https://openreview.net/forum?id=jP59rz1bZk.

Ioannis Protogeros, Rufat Asadli, Benjamin Hoffman, and Laurent Vanbever. Benchmarking LLM-Driven Network Configuration Repair, 2026. URL https://arxiv.org/abs/2604.22513.

Mohamed Sana, Nicola Piovesan, Antonio De Domenico, Yibin Kang, Haozhe Zhang, Merouane Debbah, and Fadhel Ayed. Reasoning Language Models for Root Cause Analysis in 5G Wireless Networks, 2025. URL https://arxiv.org/abs/2507.21974.

Farbod Tavakkoli, Roderic Paulk, Jorden Terrazas, Kenneth Church, Mark Austin, Louis Powell, Gregory Diamos, Lina Bariah, Syed Ali Raza Zaidi, Maryam Hafeez, Ali Maatouk, and Imtiaz Karim. Otel: Building domain-specialized telecom llm foundations for intelligent networks, 2026. URL https://arxiv.org/abs/2608.15436.

Bohao Wang, Chenwei Wu, Hang Zou, Yu Tian, Lina Bariah, Li Wei, Chongwen Huang, Yongliang Shen, Zhaoyang Zhang, and Merouane Debbah. TelecomGPT-R1: Unified Post-Training for Reasoning Across Heterogeneous Telecom Tasks, 2026. URL https://arxiv.org/abs/2609.25356.

Changjie Wang, Mariano Scazzariello, Alireza Farshin, Simone Ferlin, Dejan Kostić, and Marco Chiesa. NetConfEval: Can LLMs Facilitate Network Configuration? *Proc. ACM Netw.*, 2 (CoNEXT2), June 2024. doi: 10.1145/3656296. URL https://doi.org/10.1145/3656296.

Zhihao Wang, Alessandro Cornacchia, Alessio Sacco, Franco Galante, Marco Canini, and Dingde Jiang. A Network Arena for Benchmarking AI Agents on Network Troubleshooting. arXiv:2512.16381, 2025. URL https://arxiv.org/abs/2512.16381.

Jieting Xiao, Yun Lin, Huizhen Qiu, Rui Ma, Chen Zhong, Dongyang Xu, Xiao Long, Chaoyu Zhang, Qiaobo Hao, Ding Zou, Zhiguo Yang, Yanqin Gao, and Fang Tan. TeleCom-Bench: How Far Are Large Language Models from Industrial Telecommunication Applications?, 2026. URL https://arxiv.org/abs/2605.18025.

Junjielong Xu, Qinan Zhang, Zhiqing Zhong, Shilin He, Chaoyun Zhang, Qingwei Lin, Dan Pei, Pinjia He, Dongmei Zhang, and Qi Zhang. OpenRCA: Can Large Language Models Locate the Root Cause of Software Failures? In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=M4qNIzQYpd.

Xingyu Yan, Tingting Dai, Antonio De Domenico, Mohamed Sana, Nicola Piovesan, Changchang Li, Bowen Liu, Kun Jiang, Mengjie Zhang, Dingcheng Shan, Jing-Cheng Pang, Chenwei Wu, Sijie Wu, Lianying Chao, Haoran Cai, Jiantao Ye, Xubin Li, Simon Mark Lucas, and Xin Chen. CTBench: Evaluating Troubleshooting Capabilities of AI Agents in Realistic Telecom Network Operations. 2026. URL https://arxiv.org/abs/2608.12002.

John Yang, Carlos E Jimenez, Alexander Wettig, Kilian Lieret, Shunyu Yao, Karthik R Narasimhan, and Ofir Press. SWE-agent: Agent-Computer Interfaces Enable Automated Software Engineering. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. URL https://arxiv.org/abs/2405.15793.

Yitao Yang and Hong Xu. NetOpsBench: Open Arena for NetOps in AI Infrastructure, 2026. URL https://github.com/netx-lab/NetOpsBench.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. ReAct: Synergizing Reasoning and Acting in Language Models. arXiv:2210.03629, 2022. URL https://arxiv.org/abs/2210.03629.

Shunyu Yao, Noah Shinn, Pedram Razavi, and Karthik Narasimhan. τ-bench: A Benchmark for Tool-Agent-User Interaction in Real-World Domains, 2024. URL https://arxiv.org/abs/2406.12045.

Hao Zhou, Mandar Kulkarni, Hao Chen, Yan Xin, Charlie, and Zhang. Large Language Models (LLMs) for Telecom Root Cause Analysis (RCA): A Structured Reasoning Framework for Evidence-Grounded Diagnosis, 2026a. URL https://arxiv.org/abs/2609.02805.

Yajie Lesley Zhou, Jiajun Ruan, Eric Wang, Sadjad Fouladi, Francis Yan, Kevin Hsieh, and Zaoxing Liu. NetArena: Dynamic benchmarks for AI agents in network automation. In *International Conference on Learning Representations*, volume 2026, pp. 121053–121081, 2026b.

A Appendix

A.1 Acronyms

Abbreviation	Full term	Description
A2	Measurement event A2	The serving-cell measurement falls below a configured threshold.
A3	Measurement event A3	A neighboring cell becomes better than the serving cell by a configured offset.
A5	Measurement event A5	The serving-cell measurement falls below one threshold while a neighbor exceeds another.
ARFCN	Absolute Radio Frequency Channel Number	Numerical index identifying a radio carrier frequency.
CCE	Control Channel Element	Resource unit used to allocate downlink control-channel transmissions.
DL	Downlink	Transmission from the network to the user device.
gNodeB	Next-generation Node B	A 5G base station providing radio connectivity to user devices.
HO	Handover	Transfer of a device's connection between serving cells.
Mbps	Megabits per second	Throughput unit corresponding to one million bits per second.
MCS	Modulation and Coding Scheme	Determines the modulation format and coding rate used for transmission.
MR	Measurement Report	Report containing radio measurements of serving and neighboring cells.
PCell	Primary Cell	The main serving cell supporting the device's radio connection.
PCI	Physical Cell Identity	Identifies a cell at the physical layer; it is not a globally unique identifier.
PDCCH	Physical Downlink Control Channel	Carries control information, including resource assignments and scheduling grants.
PRB	Physical Resource Block	Unit of radio resources; its utilization indicates resource occupancy.
RLC	Radio Link Control	Radio protocol layer handling data segmentation, reassembly and, when configured, retransmission.
RRC	Radio Resource Control	Protocol managing radio connections, measurement reporting and mobility configuration.
RSRP	Reference Signal Received Power	Received reference-signal strength, used to assess serving and neighboring cells.
SINR	Signal-to-Interference-plus-Noise Ratio	Desired signal power relative to interference and noise, indicating radio-link quality.
SS-SINR	Synchronization-signal SINR	SINR measured using synchronization signals; plotted in the worked investigation.
UE	User Equipment	The user device whose throughput degradation is investigated.
UL	Uplink	Transmission from the user device to the network.

A.2 Example of a Troubleshooting Task Resolution

This appendix provides a detailed example of a troubleshooting task resolution evaluated in Discovery mode. Qwen3.8-27B with DeepSeek Harness achieves a task success score of 87.26%.

Run overview.

	Item	Value
Task ID	`f4bb1435-374a-4b09-a84e-d0d596a6d8ed`	
Configuration	Model	Qwen3.8-27B
	Harness	DeepSeek
	Evaluation mode	Discovery
Execution	Investigation rounds	10
	Validation resubmissions	1
	Duration	422.2 seconds
	Total tokens	130,683
Outcome	Task success score	87.26%

Initial instructions.

You are a diagnostic agent for the supplied network scenario.

Follow the public task, output contract, possible-correction templates, and interaction protocol exactly.

Use only the scenario context, observable scenario data, and information returned through the listed information sources.

The listed categories, subcategories, metrics, actions, and correction shapes describe legal output possibilities.

Validation feedback reports only violations of the public output contract and observable-data consistency rules. It does not confirm or reject the diagnostic correctness of an otherwise legal choice.

Return responses only in the form requested by the interaction protocol.

Benchmark mode: discovery

Mode requirement: Return exactly one fault.

Dataset task description: Analyze the supplied 5G drive-test scenario, identify the fault or faults responsible for significant user-throughput degradation, localize their affected serving cells and time interval, cite supporting evidence from the five supplied tables, and return the required corrective actions.

Investigation.

Table A1: Investigation requests.

Round	Tool and arguments	Returned observations
1	`list_tables` No arguments.	Five tables: 12 user-plane rows, four configuration rows, ten signaling events, four traffic rows, and eight measurement reports; available columns and selectors.
2	`get_throughput_over_time` No arguments.	Twelve samples, 22:21:31.230–42.827. A drop from 351–433 Mbps to 23–61 Mbps, followed by recovery.
3	`query_user_plane_data` Ten exact timestamps, 31.230–40.757; 19 columns.	Serving and neighbor PCI, frequency, signal power, SINR, throughput, and link statistics. PCI 728 serves through 38.539; PCI 929 serves from 39.476.
4	`resolve_cell_identity` `pci: 728`.	Base station `3270010`, cell `1`.
5	`resolve_cell_identity` `pci: 929`.	Base station `3210665`, cell `2`.
6	`query_traffic_data` Eight columns; `gnodeb_ids: null`.	All four cells. Serving-cell downlink PRB utilization is 9.49%; traffic statistics also include CCE utilization and allocation success.
7	`query_network_configuration_data` All four base-station IDs; 16 columns.	Serving A3 offset 10 half-dB units, versus 6 for the other cells; neighbor PCI 929 is configured. Power, antenna, bandwidth, and handover parameters are returned.
8	`query_signaling_plane_data` `time: 22:21:42.827`; all three columns.	Ten events, including A3 at 37.906 and the 728-to-929 handover attempt at 38.146.
9	`query_mr_data` All 14 columns; `call_ids: null`.	Eight reports. Six have PCI 728 serving and PCI 929 as the stronger first neighbor.

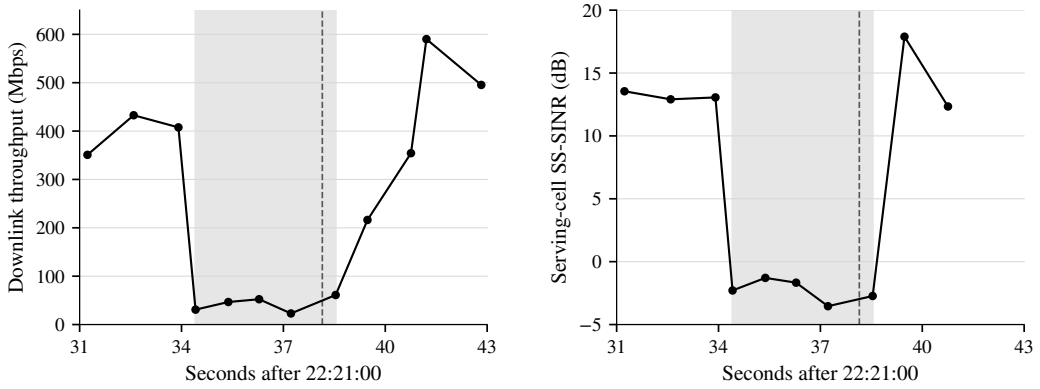

Figure A1: Example measurements. Shading: fault interval; dashed line: handover attempt.

Validation and revision. The first answer cites the user-plane timestamp `22:21:38.539` as signaling evidence. It is absent from that table, producing this feedback:

> **Path:** `final_answer[0].diagnostic_metrics[2].Timestamp[0]`
> **Message:** The selector does not exist in signaling_plane_data.
> **Actual:** `2024-09-20 22:21:38.539`
> **Expected:** an exact Timestamp value from signaling_plane_data
> **Correction:** Copy an exact selector value from the named table.

With further requests unavailable, the agent replaces this selector with `22:21:37.906`, the A3 event returned in round 8, and resubmits. The diagnosis and correction remain unchanged; validation accepts this submission.

Final answer and evaluation. The `final_answer` response contains one fault hypothesis:

Category / subcategory	Mobility / `A3_OFFSET_ISSUE`
Affected cell	`3270010_1` (PCI 728)
Fault interval	`2024-09-20 22:21:34.413` to `22:21:38.539`
Remediation	Decrease `IntraFreqHoA3Offset`; value `[2]`; target `["3270010_1"]`

Table A2 summarizes the selected evidence using shorter column names.

Table A2: Selected evidence.

Table	Rows	Selected columns	Triplets
User plane	Six samples: 34.413, 35.378, 36.285, 37.223, 38.539, 39.476.	Serving PCI, RSRP, SINR, throughput; first-neighbor PCI and signal power.	36
Configuration	Base stations 3270010 and 3210665.	PCI and A3 offset.	4
Signaling	A3 event at 37.906.	Event name and content.	2
Traffic	Base stations 3270010 and 3210665.	Cell ID, downlink throughput, weak-coverage ratio.	6
Measurement reports	Six reports with PCI 728 serving.	Serving PCI and RSRP; first-neighbor PCI and RSRP; throughput.	30
		Total predicted triplets	**78**

Category, subcategory, cell and interval agree with the reference, giving $S_{\text{loc}} = S_{\text{id}} = 1$. All correction components agree, giving $S_{\text{rem}} = 0.50 + 0.25 + 0.15 + 0.10 = 1$. Of 78 predicted and 80 reference evidence triplets, 52 are shared, 28 omitted and 26 additional. Hence

$$S_{\text{evid}} = \frac{52}{106} = 0.490566, \qquad \text{and} \qquad S^{\text{disc}} = \frac{1 + 1 + 52/106 + 1}{4} = 0.872642.$$

A.3 Dataset and Tools

Tables 3–5 respectively provide information on the dataset composition, tools, and fault remediation.

Table 3: Dataset composition.

Property	Count	Reference subcategory	Count
Tasks	500	Transmission power	268
Reference fault hypotheses	688	Antenna tilt	214
Categories	6	Antenna misalignment	75
Subcategories	9	Premature handover	57
Correction parameters	8	A3 offset	34
Tasks with 1 reference hypothesis	326	PDCCH bottleneck	31
Tasks with 2 reference hypotheses	160	Missing neighbor	3
Tasks with 3 reference hypotheses	14	Transport/test server	3
		Undetermined	3

Table 4: Benchmark tools.

Tool	Arguments	Returned information
Data access		
`list_tables`	None	Table names, selectors, columns, and row counts.
`get_throughput_over_time`	None	User-plane timestamps and downlink throughput.
`query_user_plane_data`	`timestamps`: list of strings; `columns`: list of strings	Selected measurements at exact timestamps.
`query_network_configuration_data`	`gnodeb_ids`: list of strings; `columns`: list of strings	Selected configuration fields for the named base stations.
`query_signaling_plane_data`	`time`: string; `columns`: list of strings	Signaling events up to and including the given time.
`query_traffic_data`	`columns`: list of strings; `gnodeb_ids`: optional list of strings	Selected traffic statistics, optionally filtered by base station.
`query_mr_data`	`columns`: list of strings; `call_ids`: optional list of strings	Selected measurement reports, optionally filtered by Call ID.
Cell resolution		
`resolve_cell_identity`	`pci`: integer	Cell identity associated with a physical cell identifier (PCI).
Radio analysis		
`compute_mainlobe_status`	`time`: string; `pci`: integer	Whether the device lies outside the selected cell's main lobe.
`compute_horizontal_angle`	`time`: string; `pci`: integer	Horizontal difference between the device direction and cell azimuth.
`compute_tilt_angle`	`time`: string; `pci`: integer	Vertical difference between the device direction and cell downtilt.
`compute_pathloss`	`time`: string; `pci`: integer	Path loss from transmission power and observed serving-cell RSRP.
`compute_overlap_ratio`	`pci_serving`: integer; `pci_neighbor`: integer	Calculated coverage overlap between two cells.
`compute_antenna_gain_change`	`time`: string; `pci`: integer; `adjust_horizontal_angle`, `adjust_tilt_angle`: numbers	Simulated antenna gain change for the proposed angular adjustments.

Table 5: Fault categories and remediation patterns.

Category and Subcategory	Parameter	Operation	Target–value pairs	Combinations
Coverage				
Transmission power issue	Transmission Power	Increase/decrease	1	2
Antenna tilt issue	Antenna Tilt Angle	Increase/decrease	1	2
Antenna misalignment	Antenna Azimuth	Toward the UE	1	1
Mobility				
A3 offset issue	IntraFreqHo A3Offset	Increase/decrease	1 or 2	4
Premature inter-frequency handover	A2 or A5 RSRP threshold	Increase/decrease	1	4
Neighbourhood				
Missing neighbor relationship	PCell Neighbor Cell	Update	2	1
Control-Channel				
PDCCH capacity bottleneck	PdcchOccupied SymbolNum	Update	1	1
External Issue				
Transport or test-server failure	No correction object			0
Inconclusive Evidence				
Undetermined fault	No correction object			0
			Total correction-object forms	**15**

A.4 Details on remediation scoring

The remediation score is zero when the predicted fault category or subcategory differs from the reference (see eq. (12)). Otherwise, it is $\mathrm{F1}(R, \hat{R})$, where R and $\hat{R}$ denote the reference and predicted remediation sets. If both sets are empty, $\mathrm{F1}(R, \hat{R}) = 1$; if exactly one is empty, it is zero.

For nonempty sets, write $R = \{r_1, \ldots, r_{m_R}\}$ and $\hat{R} = \{\hat{r}_1, \ldots, \hat{r}_{k_R}\}$. We calculate their F1 score in three steps.

Matching. Within each fault hypothesis, there is a single correction for network parameter; the validator enforces this requirement on predictions. We construct a one-to-one matching $\mathcal{M}_R$ between reference and predicted corrections concerning the same network parameter.

Pair scoring. For each matched pair, we compute

$$s(r_i, \hat{r}_j) = 0.50 s_{\text{parameter}} + 0.25 s_{\text{operation}} + 0.15 s_{\text{target}} + 0.10 s_{\text{value}}. \tag{17}$$

Parameter and operation similarities ($s_{\text{parameter}}$ and $s_{\text{operation}}$) use exact agreement, while target similarity is the IoU of the target sets. Value similarity compares target–value pairs using IoU, with a tolerance of ± 1 for numerical values and exact agreement for strings. When the reference correction has no applicable value, the value weight is transferred to the target component, which becomes 0.25.

Aggregation. The correction F1 score is

$$\mathrm{F1}(R, \hat{R}) = \frac{2 \sum_{(i,j) \in \mathcal{M}_R} s(r_i, \hat{r}_j)}{m_R + k_R}. \tag{18}$$

Matched corrections contribute according to their agreement, while missing and extra corrections remain in the denominator and reduce the score.

A.5 Experimental Setup

Table 6: Experimental setup.

Configurations	20 model–harness pairs in two modes; OTel–Claude in two modes (42 in total)
Tasks and attempts	500 scenarios; five attempts per task and mode; 105,000 attempts
Rounds	30 investigation rounds; five validation submissions
Timeouts	CLI/provider: 600 s; attempt: 3,600 s; tool: 15 s
Harness versions	Claude Code 2.1.234; Codex CLI 0.147.0; DeepSeek runtime 0.1.1rc1; Hermes Agent 0.20.4
Server temperature	Qwen, DeepSeek, Muse, Gemma: 1.0
Capability weights	Localization, identification, evidence, remediation: 0.25 each
Identification weights	Category: 0.75; subcategory: 0.25
Localization weights	Time: 0.5; affected entities: 0.5
Remediation weights	Metric: 0.5; action: 0.25; target: 0.15; value: 0.1

A.6 Performance by Fault Type

In this Appendix, we analyse the agentic troubleshooting capabilities for each fault type. We report results averaged across all the 20 agent configurations, each of which has been tested in 5 independent runs (K=100 experiments). We compare two settings:

- **S1: Discovery on single-reference tasks.** We evaluate the agent performance using Discovery mode, and we focus on the 326 tasks containing exactly one fault hypothesis.
- **S2: Complete on all tasks.** We evaluate the agent performance using Complete mode, and we focus on the full set of tasks in TeleAgent-Logs, including 688 reference fault hypotheses. Performance are presented at the level of fault-type, computed following the matching between reference hypotheses and predicted hypothesises.

For each row in Table 7, let N be the number of reference fault hypotheses belonging to a fault category or subcategory. The identification and success score are calculated as

$$\text{Identification} = \frac{1}{KN}\sum_{r=1}^{N}\sum_{j=1}^{K} S_{\text{id}}(r,j),$$

$$\text{Success score} = \frac{1}{4KN}\sum_{r=1}^{N}\sum_{j=1}^{K} \left[S_{\text{loc}}(r,j) + S_{\text{id}}(r,j) + S_{\text{evid}}(r,j) + S_{\text{rem}}(r,j)\right].$$

We associate zero to scores corresponding to unmatched references fault hypotheses and predictions.

Table 7: Fault-type performance. Identification and success score are presented as percentages (%).

	Number of reference hypotheses		Identification		Success score	
Category / subcategory	S1	S2	S1	S2	S1	S2
Coverage	**247**	**557**	**35.7**	**29.0**	**38.3**	**20.7**
Transmission power	127	268	50.7	35.2	43.0	24.5
Antenna tilt	103	214	9.7	11.7	28.3	7.8
Antenna misalignment	17	75	81.1	56.5	64.1	44.2
Mobility	**68**	**91**	**53.2**	**49.8**	**40.5**	**34.8**
Premature inter-frequency handover	57	57	50.8	39.1	37.2	25.0
A3 offset	11	34	65.5	67.8	57.5	51.3
Control-Channel	**2**	**31**	**83.5**	**81.3**	**67.2**	**63.3**
PDCCH capacity	2	31	83.5	81.3	67.2	63.3
Neighbourhood	**3**	**3**	**77.0**	**73.0**	**67.8**	**60.8**
Missing neighbor relation	3	3	77.0	73.0	67.8	60.8
External Issue	**3**	**3**	**15.0**	**14.0**	**28.6**	**10.5**
Transport / test server	3	3	15.0	14.0	28.6	10.5
Inconclusive Evidence	**3**	**3**	**0.7**	**0.7**	**23.6**	**0.5**
Undetermined	3	3	0.7	0.7	23.6	0.5
All	**326**	**688**	**39.5**	**34.1**	**39.0**	**24.6**

A.7 Comparison of a Telecom-specialized model and its baseline on Claude Harness

Our results show that models specialized on telco asset not necessarily perform better than their base model on TeleAgent-Logs. Interestingly, the telecom-specialized model improves its baseline capabilities to deal with antenna tilt issues. Importantly, our results highlight that the performance drop of OTel-2.0 with respect to its baseline is mainly related to the limited capability of providing a valid answer: the model frequently consumes all possible submissions per task, T_{val} (see Figure 1), or produces timeouts or no response.

Table 8: OTel-2.0-LLM-31B-IT vs Gemma 4 31B-IT capability and success scores (%; mean ± mean SD).

Mode	Model	Localization	Identification	Evidence	Remediation	Success Score
Discovery	Gemma 4	99.0±2.0	48.0±8.3	18.3±6.3	27.9±4.8	48.3±4.8
	OTel-2.0	36.4±41.1	27.0±33.6	4.7±6.3	12.9±19.6	20.3±23.9
Complete	Gemma 4	44.5±11.3	39.8±9.8	6.9±3.5	21.0±4.8	28.1±7.1
	OTel-2.0	24.6±30.6	22.7±28.5	3.4±4.8	10.1±15.2	15.2±19.2

Table 9: Performance of OTel-2.0-LLM-31B-IT vs Gemma 4 31B-IT in Discovery mode on single-reference tasks (%).

	Reference	Identification		Success Score	
Category / subcategory	hypotheses	Gemma	OTel	Gemma	OTel
Coverage	**247**	**29.4**	**29.8**	**40.3**	**21.9**
Transmission power	127	43.7	30.9	45.7	23.1
Antenna tilt	103	0.9	26.5	28.2	19.1
Antenna misalignment	17	95.3	41.5	73.4	29.2
Mobility	**68**	**75.9**	**6.8**	**49.7**	**7.0**
Premature inter-frequency handover	57	71.6	4.2	45.0	5.1
A3 offset	11	98.2	20.0	74.5	17.2
Control-Channel	**2**	**80.0**	**0.0**	**66.1**	**14.0**
PDCCH capacity	2	80.0	0.0	66.1	14.0
Neighbourhood	**3**	**93.3**	**0.0**	**77.6**	**8.2**
Missing neighbor relation	3	93.3	0.0	77.6	8.2
External Issue	**3**	**20.0**	**0.0**	**33.2**	**9.3**
Transport / test server	3	20.0	0.0	33.2	9.3
Inconclusive Evidence	**3**	**0.0**	**0.0**	**29.4**	**19.6**
Undetermined	3	0.0	0.0	29.4	19.6
All	**326**	**39.6**	**24.0**	**42.6**	**18.5**

Table 10: OTel-2.0-LLM-31B-IT vs Gemma 4 31B-IT failure analysis (%).

Failure cause	Gemma 4	OTel-2.0
Validation failure	0.88	49.54
Investigation budget failure	0.00	0.20
Context/token limit	0.00	0.06
Timeout/no response	0.66	10.28
Other	0.02	0.02
All failures	**1.56**	**60.10**

Table 11: Resource use per task.

Mean	Gemma 4	OTel-2.0
Recorded tokens (k)	147.5	130.4
Investigation rounds	13.0	11.5
Duration (min)	6.3	2.7

A.8 Resource Use

This appendix provides details on the per-task resource usage of the agents under test.

Table 12: Mean number of Tokens (in millions) per task and input–output-weighted USD per million tokens. OpenRouter snapshot: 14 September 2026

Harness	DeepSeek	Gemma	Muse	Qwen
Claude	0.286 / 0.051	0.148 / 0.124	0.367 / 0.323	0.414 / 0.269
Codex	0.722 / 0.041	0.215 / 0.098	0.547 / 0.310	0.361 / 0.179
DeepSeek	0.261 / 0.050	0.127 / 0.122	0.327 / 0.327	0.186 / 0.456
Hermes	0.258 / 0.050	0.128 / 0.100	0.366 / 0.325	0.233 / 0.620
ReAct	0.477 / 0.050	0.180 / 0.099	0.768 / 0.316	0.330 / 0.354
All harnesses	0.401 / 0.047	0.160 / 0.107	0.475 / 0.319	0.305 / 0.343

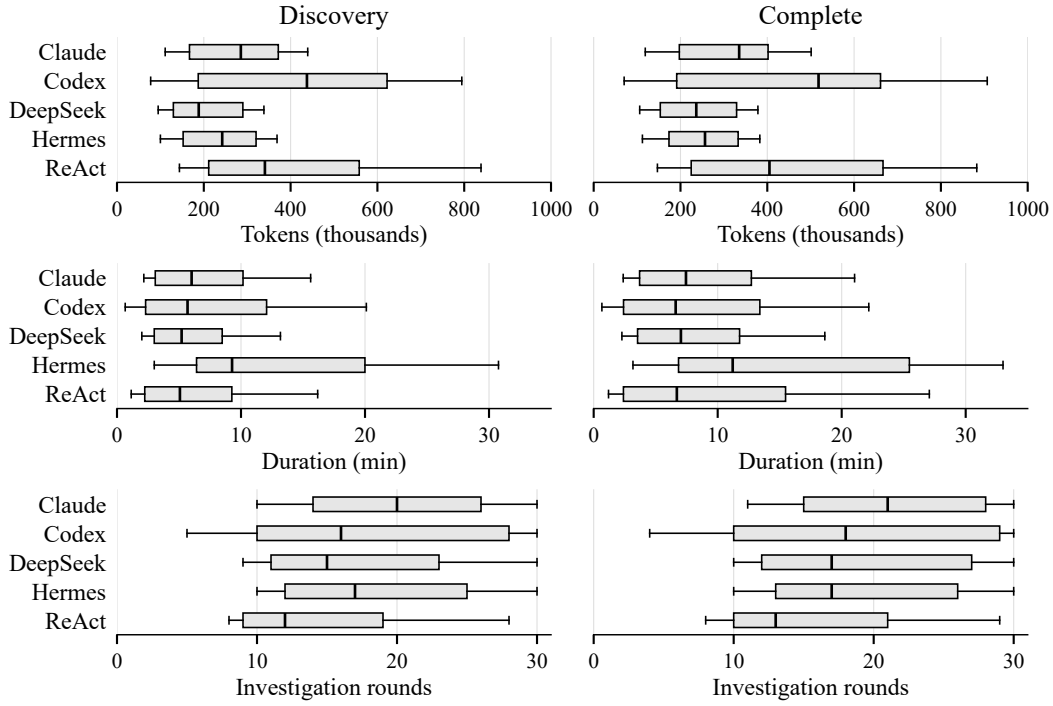

Figure 2: Per task total tokens, duration and rounds. Boxes: quartiles and median; whiskers: 10th–90th percentiles.

A.9 FAILURE ANALYSIS

This appendix provides details on the reasons behind failure events of the agents under test.

Table 13: Failure rates by configuration (%).

Harness	Model	Validation failure	Round budget exhausted	Context/ token limit	Timeout/ no response	Other
Claude	DeepSeek	0.06	0.04	0.00	4.08	0.00
	Gemma	0.88	0.00	0.00	0.66	0.02
	Muse	0.56	29.40	0.00	0.00	0.00
	Qwen	0.04	0.00	0.02	0.74	0.02
	Mean	**0.39**	**7.36**	**0.00**	**1.37**	**0.01**
Codex	DeepSeek	1.34	0.02	0.88	12.78	0.12
	Gemma	5.70	0.00	0.00	0.36	0.00
	Muse	0.94	22.88	0.00	9.52	0.00
	Qwen	1.60	0.00	0.16	50.84	0.20
	Mean	**2.40**	**5.72**	**0.26**	**18.38**	**0.08**
DeepSeek	DeepSeek	0.02	0.00	12.34	1.44	0.00
	Gemma	0.34	0.00	0.00	0.00	0.04
	Muse	4.34	2.70	0.06	0.12	0.02
	Qwen	0.02	0.00	16.16	0.00	0.02
	Mean	**1.18**	**0.68**	**7.14**	**0.39**	**0.02**
Hermes	DeepSeek	0.04	0.02	30.46	10.22	0.04
	Gemma	2.58	0.00	0.00	0.60	0.00
	Muse	3.70	4.30	0.02	0.06	0.00
	Qwen	0.04	0.00	34.44	24.98	0.08
	Mean	**1.59**	**1.08**	**16.23**	**8.96**	**0.03**
ReAct	DeepSeek	1.56	0.00	33.58	0.76	0.00
	Gemma	7.80	0.00	0.00	0.02	0.00
	Muse	14.04	0.12	0.00	0.24	0.00
	Qwen	2.28	0.00	2.36	10.16	0.00
	Mean	**6.42**	**0.03**	**8.98**	**2.79**	**0.00**
All	**DeepSeek**	**0.60**	**0.02**	**15.45**	**5.86**	**0.03**
All	**Gemma**	**3.46**	**0.00**	**0.00**	**0.33**	**0.01**
All	**Muse**	**4.72**	**11.88**	**0.02**	**1.99**	< 0.01
All	**Qwen**	**0.80**	**0.00**	**10.63**	**17.34**	**0.06**
All	**All**	**2.39**	**2.97**	**6.52**	**6.38**	**0.03**

A.10 SCORES CONDITIONED ON VALIDATED RUNS

In this appendix, we present scores by considering only the tasks where all the agent configurations have validated at least one run. This brings our evaluation to 384 tasks in Discovery mode and 220 tasks in Complete mode. In additions, failures are ignored in the computation. The results shown in Table 14 confirms that agents perform better in localization and identification tasks while they struggle to provide data-grounded evidence and remediation to the network faults. Also, they indicate that the different harnesses have similar performance, highlighting that they main distinction is their ability to satisfy the benchmark budget in terms of rounds, format, latency, token limit, and validation.

Table 14: Agent scores by configuration (%, validated attempts only).

	Localization		Identification		Evidence		Remediation		Success Score	
	Discovery	Complete	Discovery	Complete	Discovery	Complete	Discovery	Complete	Discovery	Complete
Claude harness										
DeepSeek	99.7	57.6	62.1	51.8	25.9	16.1	32.4	26.5	55.0	38.0
Gemma	99.9	36.7	44.8	34.4	18.5	6.3	25.8	23.1	47.2	25.1
Muse	97.7	57.6	56.5	50.3	18.4	12.3	22.9	21.9	48.9	35.5
Qwen	99.8	54.9	62.9	51.6	24.3	13.2	37.4	31.0	56.1	37.7
Mean	**99.3**	**51.7**	**56.6**	**47.0**	**21.8**	**12.0**	**29.6**	**25.6**	**51.8**	**34.1**
Codex harness										
DeepSeek	99.5	64.3	65.3	57.5	25.7	18.3	32.4	28.0	55.7	42.0
Gemma	99.2	32.0	34.6	29.7	15.9	4.8	20.4	18.5	42.6	21.3
Muse	97.9	63.6	60.7	55.0	20.7	15.0	21.5	22.8	50.2	39.1
Qwen	99.6	56.4	64.5	53.3	23.8	13.9	37.6	32.7	56.4	39.1
Mean	**99.1**	**54.1**	**56.3**	**48.9**	**21.5**	**13.0**	**28.0**	**25.5**	**51.2**	**35.4**
DeepSeek harness										
DeepSeek	99.9	54.8	58.9	49.5	25.6	15.7	32.6	26.7	54.3	36.7
Gemma	99.8	36.5	44.0	34.1	17.5	5.8	25.2	22.7	46.6	24.8
Muse	97.1	61.6	57.3	54.5	17.6	12.8	23.2	24.0	48.8	38.2
Qwen	99.8	49.6	58.4	46.5	24.6	12.3	32.9	28.8	53.9	34.3
Mean	**99.2**	**50.6**	**54.6**	**46.2**	**21.3**	**11.7**	**28.5**	**25.6**	**50.9**	**33.5**
Hermes harness										
DeepSeek	99.5	56.6	60.8	51.2	25.3	14.8	32.5	26.5	54.5	37.3
Gemma	99.6	32.2	33.8	29.8	16.9	4.9	19.8	18.4	42.5	21.3
Muse	97.1	59.3	57.8	53.2	18.2	12.1	23.5	24.2	49.1	37.2
Qwen	99.9	44.1	53.3	42.8	25.8	10.7	33.4	29.6	53.1	31.8
Mean	**99.0**	**48.0**	**51.4**	**44.2**	**21.5**	**10.6**	**27.3**	**24.7**	**49.8**	**31.9**
ReAct harness										
DeepSeek	100.0	56.0	59.6	50.7	25.6	15.1	33.6	27.5	54.7	37.3
Gemma	98.0	31.3	37.7	29.5	20.2	5.3	22.9	19.7	44.7	21.4
Muse	98.3	61.9	58.3	54.0	20.4	14.4	22.8	23.1	49.9	38.4
Qwen	99.9	50.4	59.1	47.2	24.2	11.4	33.6	28.7	54.2	34.4
Mean	**99.0**	**49.9**	**53.7**	**45.3**	**22.6**	**11.5**	**28.2**	**24.7**	**50.9**	**32.9**
Model averages										
DeepSeek	**99.7**	**57.9**	**61.3**	**52.1**	**25.6**	**16.0**	**32.7**	**27.0**	**54.9**	**38.3**
Gemma	**99.3**	**33.7**	**39.0**	**31.5**	**17.8**	**5.4**	**22.8**	**20.5**	**44.7**	**22.8**
Muse	**97.6**	**60.8**	**58.1**	**53.4**	**19.0**	**13.3**	**22.8**	**23.2**	**49.4**	**37.7**
Qwen	**99.8**	**51.1**	**59.6**	**48.3**	**24.6**	**12.3**	**35.0**	**30.2**	**54.7**	**35.4**
All	**99.1**	**50.9**	**54.5**	**46.3**	**21.8**	**11.8**	**28.3**	**25.2**	**50.9**	**33.5**