On the Limits of Automated Root Cause Analysis in Network Virtualization Scenarios using Language Models

Ana Beatriz L. Romero[1,2], **Pedro R. X. do Carmo**[1,2], **Assis T. Oliveira Filho**[1,2], **Judith Kelner**[1], **Djamel Sadok**[1]

[1]Grupo de Pesquisas em Redes e Telecomunicações (GPRT) – Centro de Informática (CIn) – Universidade Federal de Pernambuco (UFPE) Recife, Pernambuco, Brazil

[2]Universidade Católica de Pernambuco (UNICAP), Recife, Pernambuco, Brazil

{ana.beatriz,pedro.carmo,assis.tiago,jk,jamel}@gprt.ufpe.br

Abstract. *Root Cause Analysis (RCA) in networked and virtualized infrastructures is a complex task due to the volume of low-level metrics and the ambiguity of observable symptoms. Although Large Language Models (LLMs) have recently been explored for automated diagnosis, their effectiveness in realistic network scenarios remains unclear. This paper investigates the use of small-scale LLMs for network fault diagnosis through a systematic experimental study. We introduce the NetPerf-RCA Benchmark, composed of 24 representative network and virtualization scenarios, and evaluate multiple diagnostic approaches. Our results show that diagnostic effectiveness is primarily constrained by scenario characteristics and system observability.*

1. Introduction

The evolution of network infrastructure, driven by virtualization, the consolidation of cloud architectures, and the increasing heterogeneity of workloads, have significantly increased operational complexity and the volume of data generated [Kosińska et al. 2023]. In this scenario, maintaining service availability and performance depend on the agility in diagnosing failures and specifically identifying Root Causes [Han et al. 2024]. However, such a diagnosis still largely depends on the manual interpretation of technical logs, configuration files, and the collected performance metrics by specialists, which constitutes a time-consuming process, susceptible to errors, and difficult to scale [Kosińska et al. 2023].

Recently, Language Models have emerged as powerful tools for processing unstructured information, demonstrating remarkable capabilities in natural language interpretation and production. Furthermore, recent research explores the use of these models as mechanisms for analyzing and diagnosing logs and anomalies in complex systems, leveraging their ability to extract semantic meaning from textual operational records [Guan et al. 2024, Akhtar et al. 2025]. These approaches investigate how language models can overcome limitations of traditional detection methods and offer richer, more contextualized explanations of flaws in observational data, opening new directions for automating Root Cause Analysis (RCA) in technical domains. Despite this potential, the direct application of generic models in specific domains, such as the computer networking sector, faces critical challenges, such as the occurrence of hallucinations, sensitivity to

noise, and limitations in causal inference from partial observations [Huang et al. 2025]. In network scenarios, where multiple distinct mechanisms can produce similar observational symptoms, causal distinction is even more challenging [Lakhina et al. 2004].

This work proposes an approach to optimize fault diagnosis across a network infrastructure through the specialization of language models. It utilizes efficient Fine-Tuning techniques via QLoRA (Quantized Low-Rank Adaptation) [Dettmers et al. 2023] and Retrieval-augmented Generation (RAG) systems [Lewis et al. 2020]. We investigated the way that *Llama-3.1, Mistral*, and *Gemma* models can be adapted to interpret complex RCA scenarios applied to networks. The focus is on mapping the manner in which characteristics of the analyzed scenarios impact the feasibility of automated diagnostics, going beyond conventional accuracy optimization. As such, we seek to reduce diagnostic time and increase analytical precision by correlating textual evidence with technical knowledge bases, transforming raw data into actionable diagnostics in an automated way.

The rest of this paper is organized into the following sections: Background and Related Works reviews relevant literature; Methodology details the models, datasets, and experimental procedures; the section on Results and Discussion presents quantitative and qualitative findings and limitations; Conclusions summarizes contributions and future work.

2. Background and Related Works

The adoption of language models for diagnostic automation has advanced in several areas, based on the application of specialization techniques such as fine-tuning, Retrieval-Augmented Generation (RAG), and hybrid pipelines, which adapt base models to domain-specific tasks. Although these approaches reduce hallucinations, they do not, by themselves, guarantee robust causal reasoning under limited observability [Pingua et al. 2025]. Recent work demonstrates that Large Language Models (LLMs) can act as effective mechanisms for interpreting logs, events, and operational data, exploiting their innate ability to capture semantic relationships in unstructured information [Guan et al. 2024, Akhtar et al. 2025]. In the context of computer networks and telecommunications, the integration of AI and LLMs in 5G/6G [Kan et al. 2024, Usman et al. 2025] architectures is already established as a trend for optimizing complex operations. However, Root Cause Analysis (RCA) in this sector requires identifying not only the symptom, but the real origin of the failure through metrics, logs, and traceability.

Current literature presents relevant RCA benchmarks for microservices and large-scale systems, such as [Pham et al. 2025, Zheng et al. 2025], but these focus on classical machine learning methods, failing to explicitly model Virtualized Network Functions (VNFs) and multi-layer network stacks. In parallel, studies focused on networks, such as [Tan et al. 2024, Tan et al. 2024], perform robust anomaly detection, but without exploring the potential of language models or the variation of observability regimes. On the other hand, there are recent approaches that integrate LLMs into the RCA process for cloud log and incident analysis [Guan et al. 2024, Akhtar et al. 2025], but they tend to evaluate few scenarios, rely heavily on large-scale proprietary models, and do not provide end-to-end public benchmarks. Consequently, there is a gap in the literature regarding the automation of diagnostics with LLMs specifically geared towards virtualized network infrastructures and the systematic variation of multi-layered complexity.

To our knowledge, there is still no open benchmark for RCA in virtualized network scenarios that simultaneously: (i) provides complete implementations and knowledge bases; (ii) covers diverse performance and failure conditions; and (iii) allows for a systematic comparison of LLM specialization strategies under different observability levels. To fill this gap, this work introduces the NetPerf-RCA benchmark, offering a scenario-driven study focused on the practical limitations of Small LLMs (such as Llama-3.1, Mistral, and Gemma) for automated diagnostics in NFV environments, mapping the direct impact of causal ambiguity and visibility constraints on model effectiveness.

3. Methodology

This section details the methodology employed, including the creation of the specialized benchmark *NetPerf-RCA* and the procedures for specializing the language models. The approach ranges from curating training data and performing fine-tuning via *QLoRA*, to implementing Retrieval-augmented generation (RAG) strategies and configuring the hybrid model, which integrates fine-tuning with RAG.

Figure 1 presents a high-level overview of the proposed methodology. The pipeline starts with raw observational data collected from networked systems, which is encapsulated into structured prompts and provided as input to different language models. We next evaluate three model families (*Llama*, *Mistral*, and *Gemma*) under four configurations: baseline inference, fine-tuning only, retrieval-augmented generation (RAG) only, and a hybrid approach combining fine-tuning with RAG. The technical specifications of the selected models are summarized in Table 1. The output of each configuration consists of a structured diagnosis, supporting evidence extracted from the logs, and suggested remediation steps.

Table 1. Technical specifications of the evaluated language models.

Model	Version	Parameters	Developer
Llama-3.1-8B-Instruct	3.1	8 Billion	Meta
Mistral-7B-v0.3-Instruct	v0.3	7 Billion	Mistral AI
Gemma-3-12b-it	3.0	12 Billion	Google

The selection of the models was based on their widespread adoption and their small-to-medium parameter scales. This choice was strictly conditioned by the available infrastructure, as the experiments were conducted on a single NVIDIA GeForce RTX 4090 GPU with 24 GB of VRAM, hosted on a workstation equipped with an Intel® Core™ i7-5820K CPU (6 cores, 12 threads) and 64 GB of system memory, running Ubuntu 24.04.3 LTS. This hardware configuration imposed an upper bound on the scale of supported architectures, since larger models with higher parameter counts, although potentially achieving superior generalization and causal inference capabilities, present a prohibitive computational cost for the environment used. Finally, the inclusion of three distinct model families aimed to ensure architectural diversity, allowing for a more robust comparative analysis of performance in network diagnostic tasks.

The following subsections describe each component of the methodology in detail, starting with the construction of the *NetPerf-RCA Benchmark*, which serves as the foundation for both model training and evaluation.

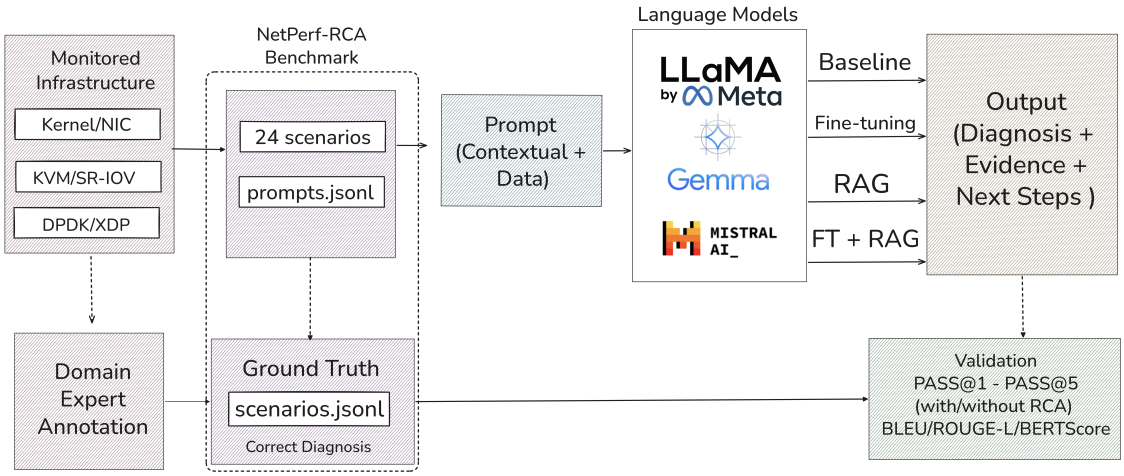

Figure 1. Methodology Flow

3.1. Dataset

To address the limited availability of public datasets focused on network diagnosis for language-model contexts, we propose the *NetPerf-RCA* Benchmark. This dataset comprises 24 scenarios covering critical failures across multiple layers of the technology stack, enabling a systematic evaluation of the limits of automated diagnosis in networking and virtualization environments.

Technological Domains. The Kernel/Base layer includes scenarios of *SoftIRQ* saturation, *bufferbloat*, and *MTU fragmentation*. At the Virtualization level (KVM/SR-IOV), we address issues of emulation overload, *VMExits*, *steal time,* and *VF spoofing*. Finally, the High-Performance Packet Processing layer (DPDK/XDP) focuses on PCIe versus CPU bottlenecks and logical failures in eBPF. This diversity of domains was intentionally chosen to reflect the heterogeneity of failures observed in real networking and virtualization environments.

Composition of Scenarios. Each dataset instance is rigorously structured into three fundamental components: *Prompt*, *Correct Answer*, and *Evidence*. The *Prompt* contains the problem context and the raw operational logs extracted from monitoring tools such as `mpstat`, `ethtool`, `top` and `perf`. The *Correct Answer* provides the precise diagnosis of the failure, while the *Evidence* correlates the numerical data from the logs with the root cause. The scenario names shown in Table 2 correspond to high-level descriptions used for categorization and analysis, while the specific root cause deterministically induced by system configurations is defined as the *ground truth* through domain expert annotation. This structure is intended to ensure that the model interprets quantitative metrics for diagnosis, rather than relying solely on explicit textual indicators, aligning the dataset with an evaluation focused on causal inference rather than mere textual similarity. Based on these scenarios, Table 3 organizes the dataset according to a network-subsystem-oriented taxonomy, which is later used to analyze how different classes of failures influence the observed success and error patterns in the experimental results. The *Observability* column indicates the extent to which the root cause can be inferred from the metrics and logs available in the *prompt*, i.e., the level of visibility provided by passive system instrumentation. Two independent domain experts assigned these ordinal levels: *High* (explicit

signatures strongly constraining plausible causes), *Medium* (partial evidence admitting multiple interpretations), and *Low* (subtle symptoms making causal inference underdetermined). It is important to note that this taxonomy is not mutually exclusive, as some scenarios involve multiple subsystems; in such cases, they are listed under more than one class to reflect their multi-layer nature.

Table 2. Network scenarios evaluated in the study. The labels represent the high-level characterization of each scenario; the specific root cause associated with each scenario is detailed in the dataset used for evaluation.

#	Scenario (Diagnosis)				
1	CPU SoftIRQ saturation	9	VirtIO single-queue	17	DPDK degradation
2	TCP bufferbloat	10	Packet drops (app CPU)	18	XDP complex BPF
3	Context switches	11	High latency (C-states)	19	PCIe link degradation
4	TCP listen overflows	12	E1000 driver overhead	20	XDP DROP action
5	Path MTU mismatch	13	E1000 throughput lim.	21	VirtIO TX saturation
6	Small NIC ring buffers	14	Docker/Netfilter ovh.	22	CPU steal time
7	TCP offloads disabled	15	RTT jitter (neighbor)	23	SR-IOV MAC spoofing
8	RTL8139 overhead	16	XDP CPU saturation	24	Netfilter conntrack

Dataset Structure. The *NetPerf-RCA Benchmark* consists of two main files in `JSON Lines` format and is publicly available as an open dataset[1]. The `prompts.jsonl` file contains the scenario descriptions and observable system data, including metrics and logs collected from monitoring tools, representing the input provided to the language models. The `scenarios.jsonl` file contains the expected diagnosis and the associated root cause for each scenario, and is used exclusively as *ground truth* during the evaluation stage. This separation ensures that models are evaluated solely based on observable information, reflecting realistic diagnostic conditions in networked environments.

Table 3. Taxonomy of the evaluated network scenarios, with explicit mapping of the 24 scenarios.

Class	Dominant Subsystem	Scenarios (#)	Observability
CPU-bound	CPU / Power Management	1, 3, 9, 11, 15, 16, 17, 18, 22	Low
NIC-bound	NIC / Queues / PCIe	2, 5, 6, 7, 19, 21	High
Kernel-path	Kernel / TCP / Netfilter	4, 14, 20, 24	High
Virtualization	VMM / Emulated Driver	8, 12, 13, 21	Medium
Mixed	Multi-layer	10, 15, 22	Low

3.2. Fine-tuning

Based on the scenarios defined in the NetPerf-RCA Benchmark, we apply parameter-efficient fine-tuning techniques to adapt each model to the network diagnosis domain. The goal of this step is not to develop new knowledge but to encourage structured reasoning over low-level system metrics and logs.

The fine-tuning process was applied to the *Llama-3.1-8B*, *Mistral-7B-v0.3*, and *Gemma-3-12B* models using the *QLoRA* (*Quantized Low-Rank Adaptation*) technique [Dettmers et al. 2023]. This approach combines the efficiency of *LoRA* (*Low-Rank Adaptation*), which keeps the original model weights frozen while introducing low-rank adaptation matrices, with 4-bit quantization, significantly reducing VRAM usage during

[1] `https://github.com/prximenes/NetPerf-RCA-Benchmark`

training. To further optimize memory efficiency and training stability, we employed the *Unsloth* framework [Daniel Han and team 2023], which provides optimized support for quantization-aware fine-tuning and *gradient checkpointing*. The overall setup and hyperparameter choices follow established practices in the literature for domain-specific adaptation of large language models [Pingua et al. 2025].

Given that effective fine-tuning typically requires a substantially larger number of training examples than the 24 original benchmark scenarios, we adopted an instruction expansion strategy to increase data diversity and coverage. Starting with the *NetPerf-RCA* scenarios, we generated an expanded instruction set following the workflow proposed in *Mobile-LLaMA* [Kan et al. 2024]. To control redundancy and encourage variability in the generated instructions, we employed the *ROUGE-L* metric as a similarity filter, discarding instructions with similarity above 70%. This iterative process was repeated until reaching a target of 500 qualified instructions. A key adaptation introduced in this work was restructuring the expanded data into a structured *Prompt*, *Correct Answer*, and *Evidence* format, with the explicit goal of encouraging the model to perform analytical reasoning over technical logs before producing a final diagnostic decision.

LoRA-based adaptation was applied using the `FastLanguageModel.get_peft_model()` function of the *Unsloth* framework, targeting the transformer attention and feed-forward layers. The main LoRA configuration parameters and training arguments used during fine-tuning are summarized in Table 4.

Table 4. LoRA adaptation configuration and training arguments used for fine-tuning.

LoRA Configuration		Training Arguments	
Parameter	**Value**	**Parameter**	**Value**
Rank (r)	16	Batch size (per device)	2
LoRA Alpha	16	Gradient accum. steps	4
Target modules	`q, k, v, o_proj` `gate, up, down_proj`	Learning rate	2×10^{-4}
		Optimizer	AdamW 8-bit
LoRA Dropout	0	Scheduler	Linear
Bias	None	Max steps	60
		Gradient checkpointing	Enabled (Unsloth)

Retrieval-Augmented Gener- ation (RAG). In addition to fine-tuning, we evaluate the use of RAG as a complementary strategy to support automated diagnosis. In this approach, external technical information is retrieved at inference time and injected into the prompt, following the classical RAG paradigm proposed by [Lewis et al. 2020]. The implementation and full knowledge base used in this study are publicly available[2]. This setup allows us to assess whether increased observability through explicit knowledge can reduce diagnostic ambiguity in complex network scenarios. The knowledge base used for RAG consists of technical documentation focused on Linux and virtualization[2], including man pages (manuals), kernel documentation, and references to technologies such as KVM, SR-IOV, DPDK, and XDP. The documents are normalized, segmented into overlapping textual chunks, and indexed in a local vector database (Qdrant) using dense embeddings. During inference, the scenario prompt is converted into a query vector, and the top-k semantically the closest chunks (with $k = 5$) are retrieved via cosine similarity and

[2]`https://github.com/prximenes/RAG-for-NetPerf-RCA/tree/full-kb`

appended to the original prompt as an explicit additional context section. The generator model is executed locally and conditioned solely on the augmented prompt, producing a structured output containing the diagnosis, evidence, and next steps. This experimental design isolates the effect of RAG as a context augmentation mechanism, enabling us to distinguish scenarios in which the retrieval of technical knowledge supports diagnosis from those in which performance limitations stem from insufficient observable signals, regardless of the specialization technique employed.

Evaluation Methodology. The evaluation of the models is conducted with a focus on the effectiveness of fault diagnosis in network scenarios, rather than on textual similarity between responses. For each benchmark scenario, five responses are generated per model. Considering the 24 evaluated scenarios, this results in 120 responses per model and per approach. In total, considering three base models and four distinct approaches (baseline, fine-tuning, RAG, and RAG+FT), the experimental process comprises **1,440 generated responses**. Each of these responses is manually analyzed by domain experts, compared against the dataset *ground truth*, and classified as either a correct or incorrect diagnosis. A response is considered correct if it accurately identifies the underlying fault, regardless of textual formulation or the order in which evidence is presented.

Based on this evaluation procedure, we adopt *Pass*@k-based metrics, which are widely used to assess diagnostic tasks and multiple-hypothesis generation [Qiu et al. 2025]. In our experiments, we use $k = 1$ and $k = 5$. Formally, the *Pass*@k metric is defined as Pass@$k = \frac{1}{N}\sum_{i=1}^{N} \mathbb{I}\,(\exists\, j \leq k \text{ such that } r_{i,j} \in \mathcal{C}_i)$, where N denotes the total number of evaluated scenarios, $r_{i,j}$ corresponds to the j-th response generated for scenario i, $\mathcal{C}_i$ represents the set of responses considered correct for scenario i, and $\mathbb{I}(\cdot)$ is the indicator function, which takes value 1 when the condition is satisfied and 0 otherwise. Specifically, *Pass@1* measures the proportion of scenarios in which the correct diagnosis is obtained in the **first response** generated by the model, reflecting a conservative operational setting in which only a single diagnostic hypothesis is considered. In contrast, *Pass@5* evaluates whether at least one of the five generated responses contains the correct diagnosis. {Following [Qiu et al. 2025], we adopt $k = 5$ to reflect operational scenarios where multiple hypotheses are presented to the operator for verification.Table 5 summarizes the metrics used and their interpretation in the context of this study. In addition, correct responses are further classified into diagnoses *with* and *without* explicit root cause analysis (RCA), enabling a separate assessment of problem recognition and causal inference. Classical textual similarity metrics like BLEU, ROUGE-L and BERTScore are reported only for complementary analysis.

4. Results and discussion

This section presents a detailed analysis of the experimental results obtained from applying language models to the automated diagnosis of network problems.

4.1. Effectiveness of the Diagnosis

Table 6 presents the aggregated diagnostic performance (Pass@1 and Pass@5) by base model and approach, considering all 24 evaluated scenarios. It can be observed that **Pass@5 is consistently higher than Pass@1** across all models, indicating that the main contribution of the evaluated approaches lies in expanding diagnostic coverage rather than

Table 5. Evaluation metrics used and their interpretation in the context of network diagnosis.

Metric	Definition	Interpretation in Networking
Pass@1	Proportion of scenarios in which the first generated answer contains the correct diagnosis.	Reflects a conservative operational setting in which only a single diagnostic hypothesis is considered.
Pass@5	Proportion of scenarios in which at least one of the five generated answers contains the correct diagnosis.	Models real-world troubleshooting practices, where multiple hypotheses are explored before confirming the root cause.
Pass@5 (with RCA)	Pass@5 considering only responses that explicitly include root cause analysis.	Evaluates the model's ability to perform causal inference beyond problem recognition.
Pass@5 (without RCA)	Pass@5 considering correct responses without explicit causal explanation.	Indicates symptom-level recognition, even in the absence of an explicit causal mechanism.
BLEU / ROUGE-L / BERTScore	Textual similarity metrics between generated responses and references.	Used only for complementary analysis; they do not reliably reflect diagnostic correctness.

improving the precision of the initial response. From an operational perspective, this suggests that the models are more effective when used as decision-support tools, providing a small set of plausible diagnostic hypotheses instead of a single definitive answer. The results also reveal relevant differences among base models. Gemma-based approaches achieve the highest diagnostic coverage, correctly identifying up to **20 out of 24 scenarios** (Pass@5 = 0.83) when combined with RAG. Mistral-based models exhibit intermediate performance, whereas LlaMA-based approaches show higher variability, with notable gains under fine-tuning but significant degradation when combined with RAG.

Overall, the aggregate results in Table 6 indicate that, under small-language-model constraints, base model choice has a stronger impact on diagnostic coverage than adding RAG or fine-tuning. These conclusions, however, only concern the ability to obtain at least one correct diagnosis (Pass@1/Pass@5) and do not account for whether the response explicitly states the root cause, which is analyzed in the next Section 4.2.

4.2. Diagnosis with and without Root Cause Analysis (RCA)

Although the aggregated results indicate high diagnostic coverage across several scenarios, particularly in terms of Pass@5, diagnostic effectiveness also depends on the model's ability to correctly articulate the root cause of the observed problem. To assess diagnostic depth, we separate correct responses into diagnoses *with RCA* and *without RCA*. Table 7 shows that requiring an explicit and correct root cause explanation substantially reduces performance in terms of Pass@5.

The consistent gap between diagnoses with and without RCA indicates that the primary challenge lies not in identifying the problem, but in correctly explicating its root cause. From a networking perspective, this reflects the difficulty of causal inference based on observable symptoms, particularly in failures involving interactions across multiple

Table 6. Aggregate diagnostic performance by baseline model and approach (24 scenarios).

Model / Approach	Pass@1	Pass@5	Scenarios (Pass@5)
Baseline Gemma	0.50	0.71	17
RAG Gemma	0.63	0.83	20
FT Gemma	0.21	0.79	19
RAG+FT Gemma	0.38	0.58	14
Baseline Mistral	0.29	0.50	12
RAG Mistral	0.25	0.42	10
FT Mistral	0.38	0.63	15
RAG+FT Mistral	0.29	0.54	13
Baseline LLaMA	0.29	0.54	13
RAG LLaMA	0.13	0.25	6
FT LLaMA	0.42	0.67	16
RAG+FT LLaMA	0.08	0.46	11

Table 7. Comparison between correct diagnoses with and without RCA (Pass@5), aggregated by base model.

Base Model	Pass@5 (with RCA)	Pass@5 (without RCA)
Gemma	0.46 – 0.54	0.25 – 0.71
Mistral	0.29 – 0.54	0.33 – 0.42
LLaMA	0.08 – 0.29	0.21 – 0.50

system layers, such as CPU, kernel, and network subsystems, where causal ambiguity is common. Figure 2 further confirms that gains in diagnostic coverage do not automatically translate into correct causal explanations.

4.3. Impact of RAG and Fine-Tuning

In this subsection, we evaluate the impact of using RAG and fine-tuning on diagnostic coverage, considering both RCA and non-RCA diagnoses as correct, in order to isolate the effect of these techniques on identifying the problem itself.

Impact of Retrieval-Augmented Generation (RAG): The use of RAG yields moderate gains that are strongly dependent on the base model and, more importantly, on the **level of observability of the network scenario**. As shown in Figure 3, which reports how many scenarios each model correctly diagnoses at least once across the five attempts, applying RAG to the Gemma model increases the concentration of correctly diagnosed scenarios, particularly in cases where relevant symptoms are explicitly represented in the retrieved data. However, this effect does not generalize across all scenarios. In cases characterized by subtle or ambiguous symptoms, the mere addition of textual context is insufficient when the model fails to correctly interpret low-level operational signals, such as CPU statistics, network queues, buffer behavior, and effects along the kernel execution path.

Effect of Fine-Tuning (FT) Fine-tuning exhibits a limited impact on diagnostic effectiveness. Although it improves textual similarity metrics, these gains do not persist when evaluated using operational metrics such as Pass@5 (Table 6). This behavior indicates that

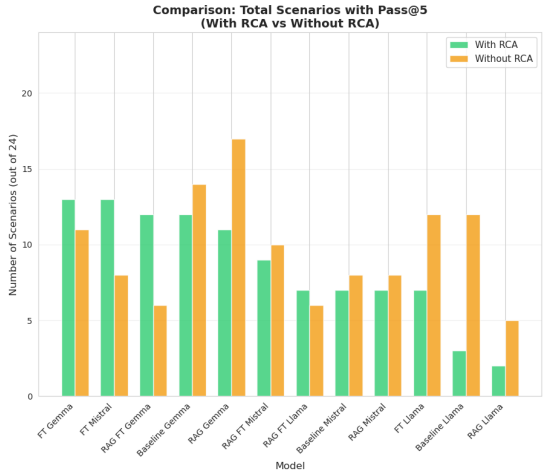

Figure 2. Comparison between diagnoses with and without RCA for different approaches.

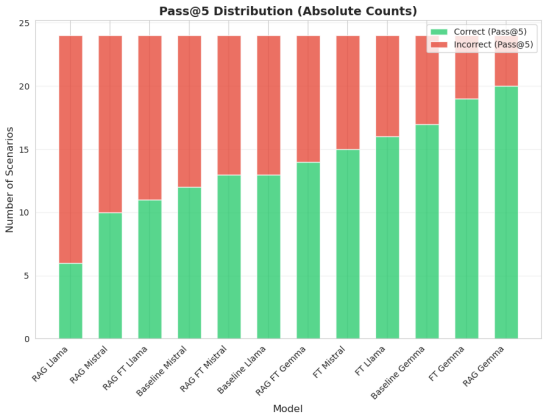

Figure 3. Aggregated distribution of correct diagnoses by approach.

FT primarily enhances the **textual form of the responses**, without necessarily strengthening the causal reasoning capabilities required for diagnosing complex networked systems.

Combination of RAG and fine-tuning. The hybrid RAG+FT approach often degrades Pass@5 for Gemma and LLaMA compared to either method in isolation. In this small-model regime, retrieved text likely dilutes the influence of low-level logs, causing over-reliance on generic documentation. This effect peaks in low-observability scenarios, where broad retrieved contexts introduce multiple unconstrained failure modes, increasing confusion rather than resolving it.

4.4. Textual Similarity versus Diagnostic Correctness

Figure 4 presents the average textual similarity metrics. It is observed that fine-tuned approaches achieve higher ROUGE-L values, indicating greater superficial similarity to the expected answers.

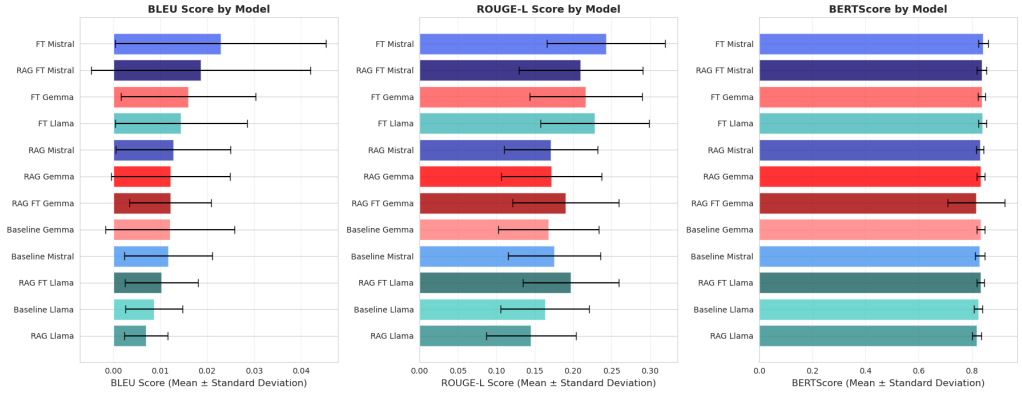

Figure 4. Average textual similarity metrics by model and approach.

However, Figure 5 shows that these metrics exhibit weak or only moderate correlation with Pass@5, reinforcing that textual similarity is not a reliable proxy for diagnostic correctness in network scenarios. This behavior indicates that fine-tuning tends to

improve the textual form of the responses and their superficial alignment with expected patterns, without consistently strengthening the causal inference capability required for diagnosing complex network systems.

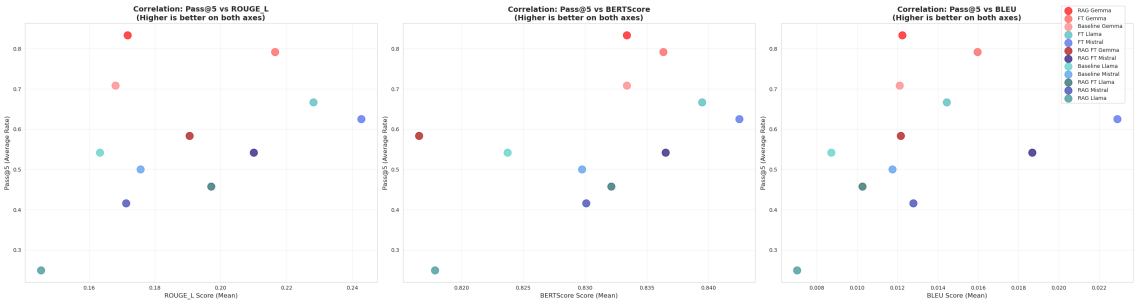

Figure 5. Correlation between textual similarity metrics and Pass@5.

4.5. Scenario-Oriented Network Analysis

A scenario-oriented analysis reveals that the performance of automated diagnosis is strongly conditioned by the intrinsic characteristics of network failures, rather than by the specific AI technique employed. In particular, the dominant affected subsystem and the level of symptom observability play a central role in diagnostic effectiveness.

Table 3 presents a taxonomy of the 24 evaluated scenarios, grouping them according to the primary subsystem impacted. This classification provides an interpretative lens for the patterns observed in performance metrics and aggregated visualizations. Scenarios classified as *NIC-bound* and *kernel-path* (e.g., Scenarios 2, 6, 14, and 24) exhibit high observability, as their symptoms manifest through explicit counters, error messages, and well-instrumented kernel events. A representative example is **Scenario 24 (Netfilter Conntrack Table Exhaustion)**, which is correctly diagnosed by all evaluated approaches (Figure 6). In such cases, the strong correlation between cause and effect enables consistent diagnosis even with small-scale models. In contrast, *CPU-bound* and *mixed* scenarios concentrate the highest error rates, as evidenced in Figure 6. **Scenario 11 (High latency under low load due to aggressive interrupt coalescing and/or CPU C-states)** exemplifies this behavior: subtle and counterintuitive symptoms emerge from the interaction between power management, CPU scheduling, and packet processing, resulting in high causal ambiguity. This pattern becomes even more evident when analyzing the aggregated performance by scenario and approach, as shown in the heatmap in Figure 6. The heatmap reveals that the same scenarios tend to be consistently diagnosed correctly or incorrectly across different approaches, reinforcing that the dominant factor is the scenario itself rather than the specific technique applied. Scenarios associated with virtualization (e.g., Scenarios 8, 12, and 13) exhibit intermediate behavior. Although CPU and I/O metrics are available, the mediation performed by the hypervisor and emulated drivers introduces additional layers of abstraction, reducing the clarity of the cause-effect relationship and resulting in variable diagnostic performance.

Overall, the results indicate that the difficulty of automated network diagnosis is strongly associated with the inherent causal ambiguity of certain scenarios, particularly those involving interactions across multiple system layers. Such scenarios impose fundamental limits on approaches based exclusively on passive observation of metrics and logs,

reinforcing that advances in this area depend not only on more sophisticated models but also on improved instrumentation and active diagnostic strategies.

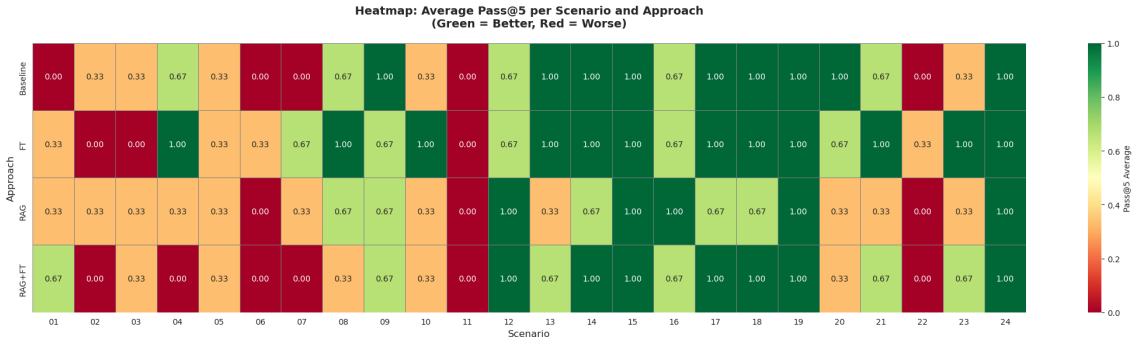

Figure 6. Aggregated heatmap of diagnostic performance by scenario and approach.

4.6. Discussion of Results

Implications for Network Instrumentation and Active Diagnosis

The results presented throughout this section indicate that automated diagnosis of network problems, when based exclusively on passive observation of metrics and logs, faces structural limitations imposed by the nature of the analyzed scenarios themselves. In particular, scenarios characterized by low observability, causal ambiguity, and interactions across multiple system layers prove to be intrinsically difficult to diagnose, regardless of the model and/or technique employed. From a network engineering perspective, these findings suggest that significant advances in automated diagnosis depend less on the choice of models or learning techniques and more on improvements in **system instrumentation**. CPU-bound and mixed scenarios, for instance, require more fine-grained metrics regarding kernel execution paths, *softirq* distribution, power management effects, and interference among concurrent workloads. Without this level of visibility, causal inference remains fundamentally underdetermined.

The results also indicate that higher-capacity models, such as *Gemma*, tend to achieve slightly better diagnostic coverage. This effect, however, does not alter the nature of the observed limitations and is likely associated with a greater ability to integrate multiple partial signals and maintain competing hypotheses, rather than with a deeper causal understanding of system behavior. Moreover, the findings reinforce the role of **active diagnosis** as a necessary complement to passive observation. Mechanisms such as controlled load injection, intentional variation of parameters (e.g., CPU affinity, interrupt coalescing, or NIC queue configurations), and guided experimentation can significantly reduce causal ambiguity, enabling the differentiation of hypotheses that produce indistinguishable symptoms under passive observation.

In this context, techniques such as RAG should be interpreted as auxiliary mechanisms whose benefits are conditioned on the availability of sufficient observability. When relevant signals are available and well instrumented, RAG can facilitate the association between symptoms and known causes. Conversely, in the absence of such signals, the addition of textual context does not compensate for the lack of causal information, limiting its practical effectiveness. Similarly, fine-tuning primarily demonstrates utility in

improving the textual form of responses and alignment with expected patterns, without consistently enhancing causal reasoning capabilities. This observation underscores that improvements in textual similarity metrics should not be interpreted as genuine advances in the diagnostic capability of complex networked systems.

5. Conclusion

This paper investigated the use of language models for automating Root Cause Analysis (RCA) in networking and virtualization scenarios, with an emphasis on understanding their practical limits under realistic constraints of small-scale models. Through a systematic evaluation, we demonstrate that the effectiveness of automated diagnosis is strongly conditioned by the intrinsic characteristics of network scenarios, particularly the dominant affected subsystem and the level of symptom observability available, and not solely by the chosen specialization technique. Results show that metrics such as *Pass@5* are better suited to capture the operational role of these models, reflecting their ability to expand diagnostic coverage by generating multiple hypotheses. Conversely, the separate analysis of diagnoses with and without explicit root-cause identification reveals that causal inference remains the primary challenge. Techniques such as *fine-tuning* and *Retrieval-Augmented Generation* (RAG) yield limited gains that are highly dependent on scenario observability and do not overcome difficulties posed by causal ambiguity, subtle symptoms, and interactions across multiple system layers such as CPU, kernel, networking, and virtualization.

As a contribution, this work validates a set of metrics and experimental procedures for evaluating language models applied to automated fault diagnosis in networks, emphasizing operational criteria rather than textual similarity. By adopting Pass@k-based metrics, distinguishing diagnoses with and without explicit root-cause identification, and organizing scenarios according to a subsystem- and observability-oriented taxonomy, we establish a reproducible methodology for more realistic analysis of these approaches in networking and virtualization contexts. Using this methodology, results indicate that for small-scale language models, the performance of the specialization techniques evaluated is strongly conditioned by scenario characteristics, particularly the available level of observability and the degree of causal ambiguity. These factors explain much of the variation observed across scenarios and approaches, while techniques such as fine-tuning and Retrieval-Augmented Generation (RAG) yield context-dependent, localized gains. Together, this evidence reinforces that progress in automated network diagnosis requires not only model improvements but also a systemic approach combining better instrumentation, increased observability, and active-diagnosis strategies.

For future work, we highlight (i) expanding *NetPerf-RCA* with new scenarios and greater diversity of multilayer interactions to reduce bias and increase fault-space coverage; (ii) investigating the impact of higher and larger-capacity models; and (iii) integrating active-diagnosis strategies and more refined instrumentation to reduce causal ambiguity and provide more informative operational signals to models.

Acknowledgement
This study was funded by FAPESP, CAPES (grant number 88887.003895/2024-00), and CNPq.

References

Akhtar, S., Khan, S., and Parkinson, S. (2025). Llm-based event log analysis techniques: A survey. *arXiv preprint arXiv:2502.00677*.

Daniel Han, M. H. and team, U. (2023). Unsloth.

Dettmers, T., Pagnoni, A., Holtzman, A., and Zettlemoyer, L. (2023). Qlora: Efficient finetuning of quantized llms. In *Advances in Neural Information Processing Systems*, volume 36, pages 10088–10115. Curran Associates, Inc.

Guan, W., Cao, J., Qian, S., Gao, J., and Ouyang, C. (2024). Logllm: Log-based anomaly detection using large language models. *arXiv preprint arXiv:2411.08561*.

Han, Y., Du, Q., Huang, Y., Li, P., Shi, X., Wu, J., Fang, P., Tian, F., and He, C. (2024). Holistic root cause analysis for failures in cloud-native systems through observability data. *IEEE Transactions on Services Computing*, 17(6):3789–3802.

Huang, L., Yu, W., Ma, W., Zhong, W., Feng, Z., Wang, H., Chen, Q., Peng, W., Feng, X., Qin, B., and Liu, T. (2025). A survey on hallucination in large language models: Principles, taxonomy, challenges, and open questions. *ACM Trans. Inf. Syst.*, 43(2).

Kan, K. B., Mun, H., Cao, G., and Lee, Y. (2024). Mobile-llama: Instruction fine-tuning open-source llm for network analysis in 5g networks. *IEEE Network*, 38(5):76–83.

Kosińska, J., Baliś, B., et al. (2023). Toward the observability of cloud-native applications: The overview of the state-of-the-art. *IEEE Access*, 11:73036–73052.

Lakhina, A., Crovella, M., and Diot, C. (2004). Diagnosing network-wide traffic anomalies. *ACM SIGCOMM computer communication review*, 34(4):219–230.

Lewis, P., Perez, E., Piktus, A., et al. (2020). Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in neural information processing systems*, 33:9459–9474.

Pham, L., Zhang, H., et al. (2025). Rcaeval: A benchmark for root cause analysis of microservice systems with telemetry data. In *Companion Proceedings of the ACM on Web Conference 2025*, New York, NY, USA. Association for Computing Machinery.

Pingua, B., Sahoo, A., Kandpal, M., Murmu, D., et al. (2025). Medical llms: Fine-tuning vs. retrieval-augmented generation. *Bioengineering*, 12(7).

Qiu, S., Wang, M., Afsharmazayejani, R., Shahmiri, M. M., Tan, B., and Pearce, H. (2025). Towards llm-based root cause analysis of hardware design failures.

Tan, Y., Wang, J., and Liu, J. (2024). Zoom-inrcl: Root cause localization at virtualized infrastructure layer for b5g/6g network slicing. In *2024 IEEE 99th Vehicular Technology Conference (VTC2024-Spring)*, pages 1–5.

Usman, Y., Oladipupo, H., During, A. D., Akl, R., and Chataut, R. (2025). Ai, ml, and llm integration in 5g/6g networks: A comprehensive survey of architectures, challenges, and future directions. *IEEE Access*, 13:168914–168950.

Zheng, L., Chen, Z., Wang, D., Deng, C., Matsuoka, R., and Chen, H. (2025). Lemma-rca: A large multi-modal multi-domain dataset for root cause analysis.