/TemplateVersion (2025.1)

www.xueshuxiangzi.com

TN-AutoRCA：用于电信网络中基于告警的根本原因分析的自改进基准构建与代理框架

Keyu Wu[1], Qianjin Yu[1], Manlin Mei[1], Ruiting Liu[1], Jun Wang[2], Kailai Zhang[2], Yelun Bao[2], Rui Ye[1], Bin He[1], Junfeng Liao[1], Lingjun Huang[1], Yongsheng Du[1], Zhiguo Yang[1], Kunlin Liu[1], Zhiwei Song[1], YanQin Gao[1,‡], Fang Tan[1,‡] Jin Yang[2,‡], Ninglun Gu[2,‡]

[1]Zhongxing Telecom Equipment(ZTE), China

[2]China Mobile Communications Group Co Ltd

AIM@zte.com.cn, { wangjunwl,zhangkailai,baoyelun,yangjin,guninglun } @chinamobile.com

Abstract

在电信网络中进行根本原因分析（RCA）是一项重要任务，但由于其复杂的基于图的推理要求和现实基准的稀缺性，对人工智能（AI）构成了巨大的挑战。为了促进这一领域的研究，我们在此提出 TN-RCA530，这是首个用于电信网络警报根本原因分析（RCA）的真实世界、公开可访问的基准，包含由专家验证的知识图谱（KG）构建的 530 个故障情景。我们的评估表明，即使是最先进的大型语言模型（LLMs）在这一任务上表现不佳，最佳模型的 F1 得分也低于 70 %，这突显了其显著的难度。为了解决这一挑战，我们提出了 Auto-RCA，这是一种新颖的代理系统，旨在自动改进基于代码的解决方案。Auto-RCA 的核心创新不仅仅是简单的自我纠正；它采用迭代的“评估-分析-修复”循环，系统地识别所有故障案例中的共性模式以生成对比反馈。该反馈指导 LLM 修复系统性逻辑缺陷，而不是孤立的错误。实验表明，这种代理框架显著提升了解决问题的性能，将 TN-RCA530 上的最终解决方案 F1 得分从基线 58.99 %（由 Gemini-2.5-Pro 直接实现）提升到 91.79 %。这项工作不仅为社区贡献了一个关键基准，还展示了自主的自优化代理架构是解决复杂、特定领域推理问题的强大范式。

介绍

基于告警的根因分析（RCA）是现代电信网络可靠性工程的基石，对于确保网络的弹性和最小化服务停机时间至关重要。然而，传统的告警根因分析方法严重依赖领域专家对复杂的告警日志数据进行详细分析，以便准确识别根因。这种依赖导致显著的主观性，与个人经验紧密相关，妨碍了统一数据管理协议和系统化建模框架的建立。近年来，大型语言模型（LLMs）在推理能力方面取得了显著进展，最新的模型在数学和编码基准测试中达到或超过人类水平（DeepSeek-AI et al. 2025; Yang et al. 2025; Team 2025; Team et al. 2025）。这一突破使得 LLMs 在解决复杂的现实世界问题上成为可能。受通信网络基础设施的基本拓扑相互依赖性的启发，我们引入了一种创新的方法，用于统一数据建模和生成。我们的解决方案在图 1 中描绘，将告警 RCA 关键的原始告警和拓扑资源通过知识图谱整合，随后将它们综合成一个根因推理图。

与一般推理任务不同，基于电信网络告警的根因分析本质上是一个基于复杂图结构数据的多标签分类问题。它需要基于深厚的领域知识进行复杂的推理，以在充满噪声和相互依赖的告警信号的情况下，准确定位级联故障的起源。一种有前景的自动化方法涉及将知识图

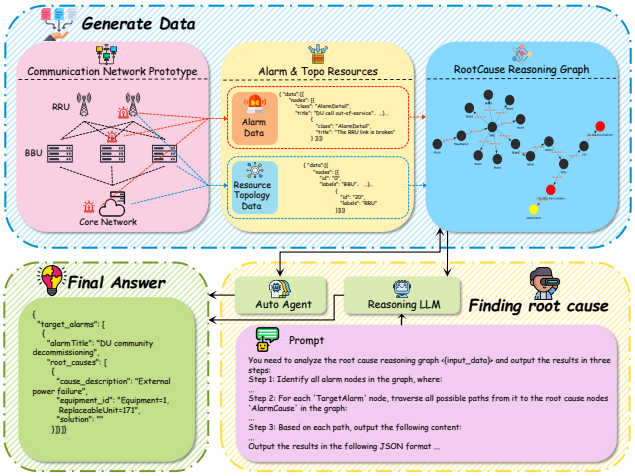

Figure 1: 基于大型语言模型（LLM）的电信基站根本原因分析框架。该过程首先通过知识图对真实世界的基站设备（如 RRU、BBU）及其连接进行建模。然后将该图与实时告警数据和专家定义的提示一起输入到大型语言模型（LLM）中。LLM 利用其推理能力进行根本原因分析，最终生成一个结构化输出，其中包括对根本原因的描述、涉及的具体设备以及建议的解决方案。

谱（KGs）的结构化知识表示与大型语言模型（LLMs）的推理能力结合起来。这种范式在图 1 中进行了说明，它将网络的物理设备及其连接建模为知识图谱，与实时告警数据一起输入到大型语言模型中，以推导出根本原因。

尽管这一框架在概念上很有吸引力，但其实际效能尚未得到验证。AI 驱动的 RCA 解决方案的发展受到两个重大缺口的阻碍。首先，缺乏全面的、真实世界的基准。现有的基准如 SWE-bench（Arora et al. 2024）或 OpenRCA（Xu et al. 2025）专注于一般软件工程，并未捕捉到电信领域的独特挑战，例如“警报风暴”以及 5G 和 NFV 引入的物理和逻辑网络层之间的复杂依赖关系。其次，也是因此，LLM 在此任务上的真正能力仍是一个未解的问题。

为了量化这一挑战并建立一个明确的基准，我们在自己开发的一个新的、现实世界的电信基准上评估了一组著名的 LLM。结果令人警醒。这种 LLM 的直接应用——衡量它们在任务上原始的、内在的推理能力——揭

www.xueshuxiangzi.com

示了显著的性能上限。即使是最先进的模型也仅能达到 58.99 % 的 F1 最高分。这一发现强调了，仅仅扩大通用模型的规模不足以掌握这一复杂领域，并且验证了该基准本身的难度。

为了弥合这些差距，本文提出了一个双重贡献，首先严格定义了问题空间，然后提供了一种强大的解决方法。我们介绍了 TN-RCA530，这是第一个专为电信 RCA 设计的大规模公共基准，为衡量和推动进展提供了一个标准化的"试验场"。随后，我们提出了 Auto-RCA，这是一个新的代理框架，展示了在这个基准上达到精通的路径。与其增强 LLM 的固有能力，Auto-RCA 在更大的系统中将模型作为一个推理引擎来利用。该系统通过系统地分析所有故障的模式来迭代地改进基于代码的解决方案，以生成针对性、对比性的反馈给 LLM。

我们的工作做出了以下关键贡献：

- 一种新颖的电信 RCA 基准（TN-RCA530）：我们引入了首个用于电信 RCA 的大规模真实世界基准，包含 530 个基于专家验证知识图的场景，具备可验证的构建方法和自动化的复杂性分级系统。
- 一个具有对比反馈循环的自主代理（Auto-RCA）：我们设计并实现了 Auto-RCA，这是一个模块化的智能代理系统，可以迭代地改进基于代码的解决方案。其核心创新是一个新颖的"评估-分析-修复"循环，通过系统地分析聚合的失败模式生成强大而结构化的对比反馈。这引导大语言模型（LLM）进行有针对性、假设驱动的代码修复，这比简单的自我反思有了显著的进步。
- 最先进的问题解决能力：通过大量实验，我们表明 Auto-RCA 框架极大地提高了解决我们基准中的任务的能力，将 TN-RCA530 的最终解决方案的 F1-score 提升到了 91.79 % 。这在该特定问题上建立了新的最先进的性能，并为实现复杂、领域特定推理的自动化绘制了一条可行的路径。

相关工作

将知识图谱（KGs）与大型语言模型（LLMs）整合是提高事实基础和减轻幻觉的关键策略。当前的研究集中在将 LLM 推理与 KG 路径对齐的框架上，使用 LLMs 作为导航 KGs 的策略模型，或利用 KGs 对生成的候选项进行重新排序。这些方法强调了一种走向结构化神经符号推理的趋势，这对于复杂的、基于事实的任务（如 RCA）是至关重要的。

在软件工程中，AI 驱动的根本原因分析（RCA）通过利用来自日志、指标和追踪的多模态数据，越来越多地取代了手动方法。尽管像 Defects4J 和 RCAEval 这样的通用基准非常重要，但它们并没有解决电信网络的独特挑战。由于大规模的"报警泛滥"、复杂的拓扑依赖性和严格的实时约束，电信行业的 RCA 具有独特性。尽管在这个领域中应用 AI/ML 取得了进展，但仍然缺乏一个大规模的公共基准，我们的 TN-RCA530 基准正是为填补这一空白而设计的。

将大型语言模型（LLM）用作自动程序修复（APR）自主代理的范式显示出显著的前景，SWE-bench 等基准测试已成为评估的标准 (Campos et al. 2025; Ma et al. 2025) 。一个显著的架构趋势是使用模块化代理系统，其中专门化的子代理协作解决复杂任务，这一理念被 MASAI (Arora et al. 2024) 和我们的 Auto-RCA 等框架所采用。

自主性自动程序修复（APR）的一个关键方面是自我纠正机制。早期的方法如 Self-Refine (Pan et al. 2024) ，依赖于大型语言模型（LLM）自身修复其输出的内在能力，但由于"认知固着" (Kamoi et al. 2024) ，这些方法在复杂推理任务中被证明无效。普遍共识是，有效改进需要可靠的外部反馈。在此基础上，方法如 ContrastRepair 使用失败和成功测试的对比反馈来指导 LLM (Kong et al. 2024) 。我们的 Auto-RCA 框架推进了这一概念，通过组织 LLM 超越实例级的修复。Auto-RCA 系统分析整个基准的失败模式，以生成更整体化和战略性的反馈。然后，这些反馈指导 LLM 修复解决方案代码，以应对系统性缺陷，展示了一个结构化自主过程如何实现直接 LLM 应用无法达到的结果。

电信网络的大语言模型

大型语言模型（LLM）在变革电信运营方面受到越来越多的关注，但通用模型通常缺乏必要的领域特定知识 (Maatouk et al. 2025; Ethiraj et al. 2025; Barriah et al. 2025) 。这导致了专门的"电信 LLM"和工具辅助代理的发展。例如，像 TAMO 这样的框架使用具有多模态数据的代理来进行 AIOps 中的细粒度根因分析（RCA） (Wang et al. 2025) 。这表明，在动态环境中进行有效和可靠的根因分析需要将 LLM 的推理与外部工具或结构化知识结合在一起。我们的工作通过提供严格的基准（TN-RCA530）以评估此任务的性能，以及展示如何有效地应用 LLM 以实现最先进的问题解决性能的高级代理框架（Auto-RCA），为这一方向做出贡献。

方法论

我们的方法通过两个部分来解决电信根因分析的挑战。首先，为了严格定义和度量问题，我们引入了 TN-RCA530，一个基于现实世界数据的新颖且具有挑战性的基准。其次，为了解决这个问题，我们提出了 Auto-RCA，一个旨在迭代学习并掌握基准复杂性的代理框架。本节详细介绍了 TN-RCA530 的构建原则、Auto-RCA 系统的架构以及用于评估性能的统一评估框架。

基准构建：TN-RCA530 数据集

我们工作的主要贡献是 TN-RCA530，一个包含 530 个不同电信网络故障场景的基准。其构建遵循四个核心原则：真实性、全面性、可验证性和复杂性判别能力。每个场景的输入是一个知识图谱，该图谱模拟了物理设备、它们的拓扑连接以及相关的告警数据。LLM 的任务是利用其领域知识和推理能力来分析此图谱，并从一组潜在候选项中识别出具体的根本原因节点。

为了实现这些原则，我们采用了一种结果导向的建设过程，具体细节如下。

真实性：为了确保基准反映现实世界中的挑战，所有数据直接来源于运营中的电信基站。这包括网络拓扑，描述设备之间的物理和逻辑连接（例如，基带单元（BBU），远程射频单元（RRU）），以及告警数据，这些告警数据是在实际故障事件中产生的真实信号。

全面性：通过穷举法收集所有可能的故障场景是低效的，而且容易出现遗漏。相反，我们采用了一种以结果为导向的方法。我们首先根据行业标准和专家知识，识别一套全面的已知根本原因。然后，我们从这些根本原因开始逆向推理，通过网络拓扑结构追踪其可能的告警表现。这确保了对根本原因类型的广泛和多样的覆盖，更有效地捕捉到常见和长尾故障场景。

www.xueshuxiangzi.com

可验证性：结果导向的方法为每种情境提供了一个内在的真实依据。由于每个案例的构建都是通过从已知的、专家验证的根本原因追踪到其结果警报，正确答案（根本原因）在设计时即嵌入数据中。因此，这为评估模型性能提供了一个可靠且明确的基础。

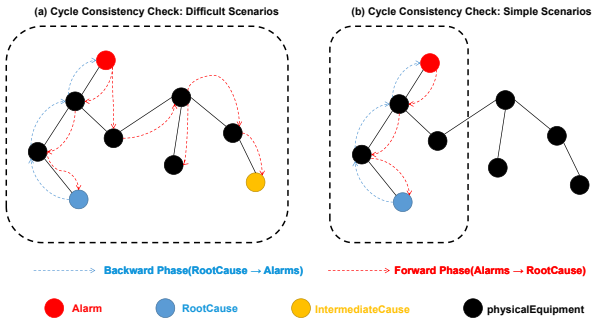

Figure 2: 说明自动难度分级的循环一致性检查。(a) 一个困难场景，其中正向推理阶段（红色箭头，Alarms → RootCause）导致一对多映射（$|m| > 1$）。初始 Alarm 可以追溯到真实的 RootCause 和其他合理的原因（例如，IntermediateCause），造成歧义。(b) 一个简单场景，其中正向映射是一对一（$|m| = 1$）。从 Alarm 到 RootCause 的因果路径不含歧义，反向阶段（蓝色箭头，RootCause → Alarms）确认了这个唯一且一致的循环。

复杂性可辨性：我们方法的一个关键创新是能够客观且自动地对情境的难度进行分层。这通过在图结构中基于一个向后推理和一个向前推理过程之间的循环一致性检查来实现。

- 向后阶段（根本原因 → 告警）：这一阶段建立了真实情况。对于给定的标准根本原因（例如，“外部电源故障”），我们利用资源拓扑图和专家定义的规则来确定此原因将触发的完整告警链。这创建了一个明确的映射：

$$M_{backward} : RootCause \rightarrow \{Alarm_1, Alarm_2, \ldots, Alarm_n\} \quad (1)$$

- 正向阶段（警报 → 根本原因）：这个阶段模拟 LLM 的任务。从反向阶段识别出的警报集合开始，我们构建输入知识图谱。然后，我们应用基于图的推理来生成一个潜在的推理图，将警报连接到一个或多个可能的根本原因。这确立了一个正向映射：

$$M_{forward} : Alarms \rightarrow \{InferredRootCause_1, \ldots, InferredRootCause_m\} \quad (2)$$

  图中因果路径的唯一性为这个过程提供了理论基础。
- 循环一致性检查 & 难度分级：我们通过将前向阶段的输出与后向阶段的输入进行比较来评估一致性。
  - 简单场景：如果存在一对一的映射（$|m| = 1$），并且推断出的根本原因与原始根本原因完美匹配，则该场景被归类为“简单”。从警报到根本原因的因果路径是明确的。
  - 困难情景：如果存在一对多映射（$|m| > 1$），则将情景分类为“困难”，其中警报可能合理地指向多个根本原因，包括真实原因和几个干扰项。困难程度与合理的替代根本原因的数量成正比，因为这要求模型进行更微妙的推理以排除不正确的假设。

这个自动化过程消除了主观的难度手动标记，并提供了一种在复杂性范围内分析模型性能的原则性机制。

数据表示和统计

TN-RCA530 基准测试中的每个场景都封装在一对 JSON 文件中：input.json，它包含供 LLM 分析的故障场景的知识图表示，以及 label.json，它提供了真实的答案。

知识图谱结构：输入.json 文件将故障场景建模为有向图 $G = (V, E)$，其中 V 是一组节点，而 E 是一组表示其关系的边。节点 V 主要分为三种类型，这对 RCA 任务至关重要：

- 资源拓扑节点：这些节点代表了电信基站的物理和逻辑组件，例如基站（BaseStation）、基带单元（BBU）、特定的功能板（Board，例如 VBPe5a）、远程射频单元（RRU）和端口（RiPort）。每个节点包含如其逻辑专用名称（ldn）和序列号等属性，构成了网络的结构骨干。
- 告警节点（AlarmDetail）：这些节点表示症状数据。它们被链接到特定的资源节点，指明是由哪一设备报告了告警。每个告警节点包含丰富的元数据，包括标题（例如，“LTE 小区不在服务范围内”）、代码、严重性以及精确的报告告警时间。
- 根本原因节点（AlarmCause）：这些节点表示警报的潜在根本原因。在 input.json 文件中，这些是与资源节点相关联的候选原因。LLM 的目标是识别出这些候选原因中哪个是真正的根本原因，用于给定的 TargetAlarm。准确的根本原因只在 label.json 文件中独家指定以供验证。

图中的边 E 定义了这些节点之间的关系。它们包括 dependOn 边，用于描述设备的物理或逻辑层次结构，generate 边用于将警报与产生警报的设备链接，以及 causedBy 边表示不同警报之间或根本原因与警报之间的因果假设。这种基于图结构的表示使得 LLM 能够通过遍历拓扑和因果链接来进行复杂的多跳推理，以从一系列症状中推断出主要故障源。

基准统计：对 TN-RCA530 的统计分析揭示了两个核心特征：现实的根本原因分布和以“困难”场景为主导的具有挑战性的设计。

首先，基准测试中的根本原因分布呈现出典型的”长尾”模式（如图 3 所示）。少数常见故障（例如，”外部断电”）占据了大多数案例，而大量多样的、不常见的故障构成了”尾部”。这种分布与现实世界的操作数据非常接近，确保了基准测试的实际相关性。

其次，也是我们工作的一个关键创新，我们应用了一种自动化的“循环一致性检查”方法论，将所有 530 个情境按难度进行分层。结果是由 29 个“简单”情境和 501 个“困难”情境组成的基准。这意味着 94.5 % 的情境被归类为“困难”。这种设计确立了 TN-RCA530 作为一个极具挑战性的基准，能够有效区分出各种模型推理能力的上限，特别是在处理那些仍未解决的复杂和模糊问题时。

www.xueshuxiangzi.com

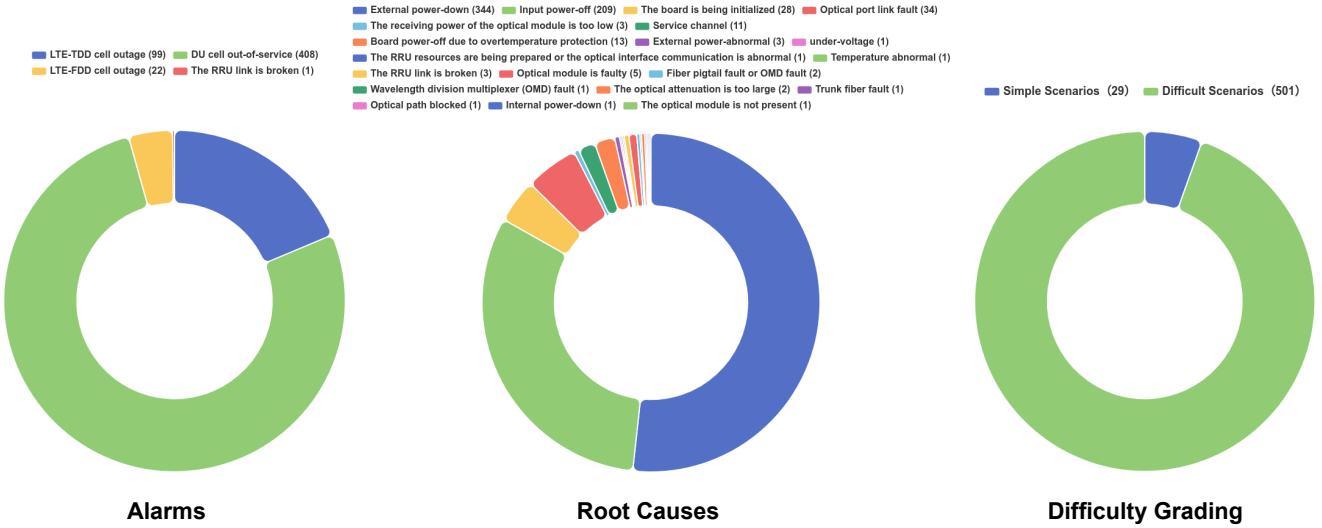

Figure 3: TN-RCA530 中的根本原因分布和复杂性分层。图表显示了主要根本原因类别的频率，展示了基准的长尾分布。每个柱状图都被分段，以显示该原因下“简单”与“困难”场景的绝对数量。该可视化揭示了两个关键特性：(1) 少数常见故障的主导性，以及 (2)“困难”场景（占总量的 94.5 %）在所有类别中的压倒性普遍性，突出了该基准的高挑战性。

在建立了用于量化电信 RCA 挑战的 TN-RCA530 基准后，我们现在引入 Auto-RCA，这是我们开发的用于掌握此任务的代理框架。Auto-RCA 的设计并不是为了增强 LLM 的内在推理能力，而是在一个结构化的、迭代的过程中协调其能力，通过从基准测试中的失败中学习，完善一个基于代码的程序化解决方案。

核心哲学：Auto-RCA 的核心原则是将失败视为一种系统性的学习机会，而不是孤立的事件。该框架不是逐一修复漏洞，而是分析所有失败案例（错误案例）中的模式，以识别当前解决方案代码中的主要逻辑缺陷。这使得系统能够引导 LLM 进行系统性、假设驱动的修复，而不是肤浅的、特例层面的修补。

系统架构：如图 4 所示，Auto-RCA 采用了一种模块化架构，由五个协同工作的模块组成，模拟了一个专家软件开发团队。这个结构将关注点分开，确保了一个稳健、可重复的工作流程。

- 协调者（项目经理）：管理端到端的“评估-分析-生成-验证”生命周期。它控制工作流程，并根据基准上的经验性能变化做出是否接受或拒绝所提议的代码修改的最终决定。
- 评估员（测试工程师）：量化给定代码解决方案的性能。它在整个 TN-RCA530 基准上运行代码，计算准确的 F1 分数，并将所有不正确的输出编译成一组不良案例以供分析。
- 故障案例分析器（高级分析师）：这是框架的智能核心。它接收来自评估者的所有故障案例，并进行系统分析以识别和分类最常见的故障模式（例如，“缺失根本原因”与“多余根本原因”）。这个分析是系统对比反馈的基础。
- LLM 代理（编码 & 思考者）：该模块作为大型语言模型的接口。它将来自故障案例分析器的结构化输出转换为一个高质量、针对性的提示。这个提示结合了故障分析报告、具代表性的故障案例及预配置的

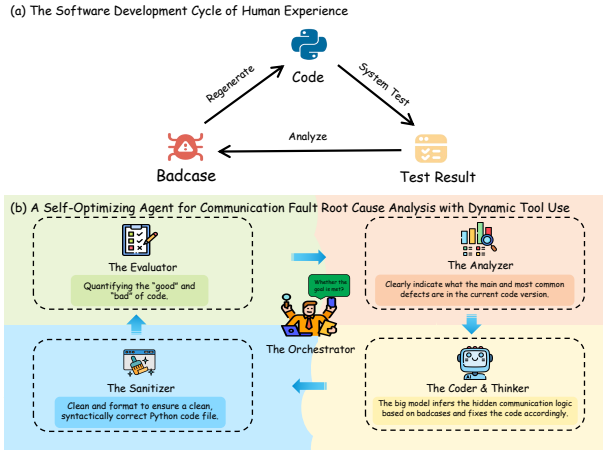

Figure 4: Auto-RCA 主体框架的架构。该系统在一个连续循环中运行，其中协调器管理用于评估代码性能、分析故障、为 LLM 生成有针对性的提示以及清理生成代码以供下一次迭代的模块之间的工作流程。

www.xueshuxiangzi.com

探索性问题，以引导大型语言模型的代码生成过程朝向一个特定且逻辑的修复方向。

- 审核者（代码审查者）：确保 LLM 输出的可靠性。它清理并格式化生成的文本，以产生一个纯粹且语法正确的 Python 文件，在代码传递给评估者之前，去除对话中的残留物或不一致性。

迭代工作流程：Auto-RCA 系统在一个封闭循环中运行，旨在逐步优化：

1. 基线测试（第 0 轮）：在基准上运行初始代码版本，建立基线 F1 分数并生成第一组错误案例。
2. 故障分析：编排器将坏案例交给坏案例分析器，该分析器识别出最关键的系统性缺陷（例如，一种生成"EXTRA_ROOT_CAUSE" 的一致模式）。
3. 引导代码生成：LLM 代理使用此分析来构建一个有针对性的提示，并查询 LLM 以获得一个新的、改进版本的解决方案代码。
4. 评估 & 决策：候选代码被清理并在基准上重新评估。协调器将新的 F1 分数与之前的最佳分数进行比较。如果分数有所提高，新代码将被接受并成为下一个迭代的基础。如果没有提高，则被拒绝。
5. 迭代：该过程会重复进行，使用成功运行后得到的新一组（较少的）坏情况下的数据作为下一轮分析的输入，逐步优化解决方案。

评估框架和指标

为了系统地衡量 LLM 在 TN-RCA530 上的表现，我们开发了一个全面的评估框架和一套精确的指标。

评估协议：我们的框架旨在实现自动化执行。给定模型的 API 端点，框架会将 530 个场景中的每一个呈现为输入知识图。模型会收到一组结构化的指令，要求分析该图并以标准化的 JSON 格式返回其发现。该格式要求模型识别出目标警报，并且对于每个警报，列出推断出的根本原因，包括文本描述（*cause_description* ），相关设备 ID（*equipment_id* ）以及一个建议的解决方案。

主要指标（F1-得分）：鉴于该任务的多标签特性（例如，可能存在多个警报，并且一个警报可能有多个根本原因），我们采用 F1-得分作为主要评估指标。它提供了一个平衡的衡量标准，评估模型正确识别所有真实根本原因而不引入错误的能力。

F1-Score 的计算公式如下：

- 提取：框架解析模型生成的 JSON 以提取预测根本原因的集合（P ）。它还从对应的标签文件中提取真实的根本原因集合（T ）。每个根本原因都被视为（*cause_description* ，*equipment_id* ）的一个元组。
- 匹配：如果预测的根本原因 $p \in P$ 与真实的根本原因 $t \in T$ 完全匹配，则将其视为真正例（TP）。
- 计算：我们计算精度和召回率：

$$\text{Precision} = \frac{|P \cap T|}{|P|}$$

$$\text{Recall} = \frac{|P \cap T|}{|T|}$$

- F1-Score 是精确率和召回率的调和平均值：

$$F_1 = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

模型的最终得分是所有 530 个情景中 F1-Score 的宏平均值。我们的框架还报告总体精确率和召回率，以深入了解模型行为。

实验

本节展示了一系列实验，旨在回答两个核心研究问题：

1. 当前最先进的大型语言模型（LLM）直接应用于 TN-RCA530 基准时的基线性能如何？
2. Auto-RCA agentic 框架能否有效提高解决 TN-RCA530 问题的性能？

实验设置

- 基准：所有实验均在完整的 TN-RCA530 基准上进行，该基准包括 530 个基于专家验证知识图的真实故障场景。该基准进一步分为 29 个“简单”场景和 501 个“困难”场景，“混合”代表整个数据集。
- 评价指标：主要的评价指标是宏平均 F1 分数，计算覆盖所有 530 种情境。为了提供一个更全面的性能视图，我们还报告 Precision@k（P@k）和 Recall@k（R@k），其中 k=1。
- 基线模型：为了建立一个稳健的基线，我们评估了各种知名的封闭源和开源的 LLM。如表 1 所示，这包括来自 DeepSeek 和 Qwen 系列的模型，以及谷歌的 Gemini 和 Anthropic 的 Claude 家族。
- 代理配置：对于 Auto-RCA 系统，我们选择了几种支持框架所需长上下文窗口的领先模型（至少 64K tokens）。每次实验运行由五个迭代优化轮次组成，从相同的初始代码库开始。

为了回答我们的第一个研究问题，我们测量了各种大型语言模型直接应用于 TN-RCA530 基准测试时的“开箱即用”性能。该实验评估了这些模型在直接解决 RCA 问题时的内在推理能力，而不依赖于迭代改进框架。TN-RCA530 基准测试特别具有挑战性，包括 530 个真实世界的故障场景，其中大多数（501 个“困难”场景）需要复杂的推理，此外还有一小部分 29 个“简单”场景。

在表 1 中总结的结果颇具启发性。对于“简单”场景，包括 Gemini-2.5-Pro、Claude-3.5-Sonnet、Claude-3.7-Sonnet、Claude-Sonnet-4、Qwen-QWQ 和 Qwen3-235B 在内的多个模型，都取得了相同且较高的 Precision@1、Recall@1 和 F1-Score@1，均为 0.8621。这表明对于不太复杂的故障诊断任务，许多最先进的大型语言模型展现了相似且优秀的表现。

然而，当模型面对“困难”场景时，性能显著下降。在这一类别中，Qwen3-235B 表现出最好的 Recall@1（0.9810）和 F1-Score@1（0.6049），而 Qwen-QWQ 则在 Precision@1（0.4381）上取得了最高值。反映整体数据集构成的“混合”场景中，Qwen-QWQ 在 Precision@1（0.4613）上领先，而 Qwen3-235B 则在 Recall@1（0.9736）和 F1-Score@1（0.6254）上领先。值得注意的是，Qwen 系列模型在“混合”和“困难”类别中一般表现出强大的性能。这可以归因于我们的基准测试中的故障描述主要是中文，而 Qwen 模型由于在开发中高度关注中文语言能力，可能在理解和处理问题陈述的细节上具有明显的优势。

即使在“混合”场景中最佳的 F1-Score@1 也仅达到 0.6254（Qwen3-235B），这一观察到的性能上限，强调

www.xueshuxiangzi.com

Table 1: TN-RCA530 场景难度的性能分解。每列中表现最好的值用粗体标出。

	Precision@1			Recall@1			F1-Score@1		
Model	Simple	Difficult	Mixed	Simple	Difficult	Mixed	Simple	Difficult	Mixed
DeepSeek-R1-671B	0.8448	0.4240	0.4471	0.8448	0.9600	0.9537	0.8448	0.5749	0.5897
Gemini-2.5-Pro	0.8621	0.4311	0.4356	0.8621	0.9658	0.9134	0.8621	0.5824	0.5899
Claude-3.5-Sonnet	0.8621	0.4021	0.4234	0.8621	0.9653	0.9537	0.8621	0.5677	0.5934
Claude-3.7-Sonnet	0.8621	0.4370	0.4441	0.8621	0.9800	0.9626	0.8621	0.5912	0.6078
Claude-Sonnet-4	0.8621	0.4323	0.4561	0.8621	0.9783	0.9586	0.8621	0.5995	0.6181
Qwen3-4B	0.6752	0.3680	0.3782	0.7040	0.8302	0.8038	0.6741	0.4955	0.5144
Qwen3-32B	0.8302	0.4277	0.4498	0.8563	0.9756	0.9691	0.8311	0.5789	0.6142
Qwen-QWQ	0.8621	0.4381	0.4613	0.8621	0.9744	0.9682	0.8621	0.5900	0.6248
Qwen3-235B	0.8621	0.4374	0.4607	0.8621	0.9810	0.9736	0.8621	0.6049	0.6254

Table 2: 按情景难度划分的最佳解决方案代码性能分析。每列中性能最佳的值以粗体显示。

	Precision@1			Recall@1			F1-Score@1		
Model	Simple	Difficult	Mixed	Simple	Difficult	Mixed	Simple	Difficult	Mixed
DeepSeek-R1-671B	0.7832	0.8386	0.8115	0.8409	0.9627	0.9288	0.7997	0.8746	0.8450
Gemini-2.5-Pro	0.9483	0.8920	0.8951	0.9828	0.9763	0.9767	0.9540	0.9158	0.9179
Claude-3.5-Sonnet	0.9234	0.8547	0.8097	0.7453	0.9113	0.8633	0.8250	0.8721	0.8262
Claude-3.7-Sonnet	0.9310	0.8500	0.8544	0.7586	0.7660	0.7656	0.8161	0.7908	0.7922
Claude-Sonnet-4	0.9483	0.4407	0.4685	0.9828	0.9823	0.9823	0.9540	0.5943	0.6140
Qwen3-235B	0.9109	0.6950	0.6682	0.7839	0.6930	0.6626	0.8422	0.6923	0.6632

了 TN-RCA530 基准测试对直接应用大规模语言模型的固有困难。这表明在复杂领域中进行专业的基于图的推理对通用大模型仍然是一个显著的挑战，仅仅通过扩大模型规模对于掌握该领域是不足的，除非进行进一步的改进或采用代理方法。

为了回答我们的第二个研究问题，我们评估了 Auto-RCA 框架的有效性。该实验并不衡量 LLM 基础智能的提升，而是整个问题解决系统的性能，该系统使用 LLM 作为代码生成组件。目标是确定我们的代理化、迭代精炼过程是否能在基准测试上生成更高准确度的最终解决方案。

图 5 展示了最终解决方案代码在 Auto-RCA 框架中经过五轮迭代改进的性能。结果显示，与基线直接应用 LLMs 相比，问题解决能力有了显著提升。值得注意的是，当在框架中使用 Gemini-2.5-Pro 作为推理引擎时，解决方案的 F1-score 从最初的基线 0.5899 提升到第 1 轮的 0.9179，这是框架中任何模型达到的最佳总体性能。这一显著的增长突显了该框架在显著提升问题解决准确性方面的能力。

在多个模型的不同轮次中，迭代进展显而易见。得分为 0.0000 表示本轮生成的代码在语法上无效或表现不如之前的最佳代码，从而导致协调器拒绝更改——这是该系统鲁棒性的关键特性，防止性能退化并确保仅采用改进的部分。

在 Auto-RCA 框架中，Gemini-2.5-Pro 优异性能的一个关键因素是其显著较大的最大令牌容量（1M）。这种广泛的上下文窗口使得 Gemini-2.5-Pro 能够处理更大量的有效信息，包括详细的问题描述、历史交互和中间推理步骤，而不被截断。相比之下，其他上下文窗口较小的模型（例如，Claude-Sonnet-4 和 Qwen3-235B，均为 64K 令牌）更容易因为截断导致信息丢失，这可能会妨碍它们生成最佳代码修改的能力并导致次优性能。这表明对于复杂的、迭代的代码生成任务，保持全面上下文的能力至关重要。

为了进一步分析该框架的输出，我们评估了在不同情境难度下表现最好的解决方案代码（来自 Gemini-2.5-Pro，F1 得分 0.9179），如表 2 所详述的。

- 简单场景：该解决方案取得了接近完美的 F1 分数 0.9540，展示了对具有明确因果关系的基础问题的掌握。
- 困难场景：性能下降到 F1 分数 0.9158，这揭示了该解决方案在面对复杂因果链或模糊信号时的当前限制。相比于准确率（0.8920），较高的召回率（0.9763）表明模型倾向于识别大多数真实正例，但也标记了

www.xueshuxiangzi.com

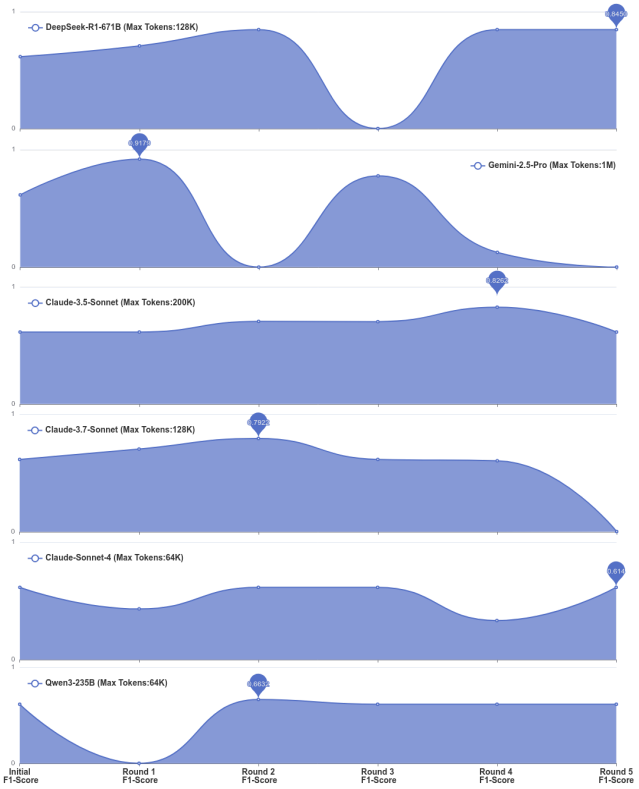

Figure 5: 通过使用 Auto-RCA 框架的 TN-RCA530 性能改进。折线图展示了最佳解决方案代码在五次精炼迭代中的 F1-分数演变。标记点表示每个模型在混合数据集上获得的最高 F1-分数。0.0000 的分数表示由于性能下降或错误而被拒绝的代码修改，从而确保了单调改进。

一些误报，在复杂情况下采取“激进”的诊断策略。

- 混合场景：F1-得分为 0.9179，与整体迭代性能非常接近，确认了基准的平衡构成和解决方案在实际应用中的有效性。

总而言之，Auto-RCA 框架是一个可以有效执行复杂推理任务的范式。它达到了 0.9179 的最终 F1 得分（Gemini-2.5-Pro），相比于最佳基线 F1-Score@1 的 0.6254（Qwen3-235B 在混合场景中）的显著提升。这验证了我们的假设，即一个自主的、自优化的代理架构可以有效地掌握像 TN-RCA530 这样领域特定的挑战。

结论

本文对 AI 驱动的网络管理做出了两个核心贡献。首先，我们引入了 TN-RCA530，这是一个公开的、真实世界的基准，为衡量和推动电信 RCA 的进步提供了一个标准化的平台。其次，我们提出了 Auto-RCA，这是一种新颖的代理框架，展示了掌握该基准的明确路径。

我们工作的核心观点是，对于复杂的、特定领域的推理任务，最有效的范式可能不是直接应用大型语言模型。相反，可以通过在自主代理框架中利用它们，逐步改进可验证的解决方案，以获得更优越的性能。未来的工作将侧重于通过增加复杂性扩展基准，并将 Auto-RCA 框架推广到其他具有挑战性的、基于图的问题领域。

References

Arora, D.; Sonwane, A.; Wadhwa, N.; Mehrotra, A.; Utpala, S.; Bairi, R.; Kanade, A.; and Natarajan, N. 2024. MASAI: Modular Architecture for Software-engineering AI Agents. arXiv:2406.11638.

Barriah, L.; De Domenico, A.; Powell, L.; Sana, M.; Wang, P.; Piovesan, N.; and Debbah, M. 2025. GSMA Open-Telco LLM Benchmarks. https://huggingface.co/blog/otellm/gsma-benchmarks.

Campos, V.; Shariffdeen, R.; Ulges, A.; and Noller, Y. 2025. Empirical Evaluation of Generalizable Automated Program Repair with Large Language Models. arXiv:2506.03283.

DeepSeek-AI; Guo, D.; Yang, D.; Zhang, H.; Song, J.; Zhang, R.; Xu, R.; Zhu, Q.; Ma, S.; Wang, P.; Bi, X.; Zhang, X.; Yu, X.; Wu, Y.; Wu, Z. F.; Gou, Z.; Shao, Z.; Li, Z.; Gao, Z.; Liu, A.; Xue, B.; Wang, B.; Wu, B.; Feng, B.; Lu, C.; Zhao, C.; Deng, C.; Zhang, C.; Ruan, C.; Dai, D.; Chen, D.; Ji, D.; Li, E.; Lin, F.; Dai, F.; Luo, F.; Hao, G.; Chen, G.; Li, G.; Zhang, H.; Bao, H.; Xu, H.; Wang, H.; Ding, H.; Xin, H.; Gao, H.; Qu, H.; Li, H.; Guo, J.; Li, J.; Wang, J.; Chen, J.; Yuan, J.; Qiu, J.; Li, J.; Cai, J. L.; Ni, J.; Liang, J.; Chen, J.; Dong, K.; Hu, K.; Gao, K.; Guan, K.; Huang, K.; Yu, K.; Wang, L.; Zhang, L.; Zhao, L.; Wang, L.; Zhang, L.; Xu, L.; Xia, L.; Zhang, M.; Zhang, M.; Tang, M.; Li, M.; Wang, M.; Li, M.; Tian, N.; Huang, P.; Zhang, P.; Wang, Q.; Chen, Q.; Du, Q.; Ge, R.; Zhang, R.; Pan, R.; Wang, R.; Chen, R. J.; Jin, R. L.; Chen, R.; Lu, S.; Zhou, S.; Chen, S.; Ye, S.; Wang, S.; Yu, S.; Zhou, S.; Pan, S.; Li, S. S.; Zhou, S.; Wu, S.; Ye, S.; Yun, T.; Pei, T.; Sun, T.; Wang, T.; Zeng, W.; Zhao, W.; Liu, W.;

www.xueshuxiangzi.com

Liang, W.; Gao, W.; Yu, W.; Zhang, W.; Xiao, W. L.; An, W.; Liu, X.; Wang, X.; Chen, X.; Nie, X.; Cheng, X.; Liu, X.; Xie, X.; Liu, X.; Yang, X.; Li, X.; Su, X.; Lin, X.; Li, X. Q.; Jin, X.; Shen, X.; Chen, X.; Sun, X.; Wang, X.; Song, X.; Zhou, X.; Wang, X.; Shan, X.; Li, Y. K.; Wang, Y. Q.; Wei, Y. X.; Zhang, Y.; Xu, Y.; Li, Y.; Zhao, Y.; Sun, Y.; Wang, Y.; Yu, Y.; Zhang, Y.; Shi, Y.; Xiong, Y.; He, Y.; Piao, Y.; Wang, Y.; Tan, Y.; Ma, Y.; Liu, Y.; Guo, Y.; Ou, Y.; Wang, Y.; Gong, Y.; Zou, Y.; He, Y.; Xiong, Y.; Luo, Y.; You, Y.; Liu, Y.; Zhou, Y.; Zhu, Y. X.; Xu, Y.; Huang, Y.; Li, Y.; Zheng, Y.; Zhu, Y.; Ma, Y.; Tang, Y.; Zha, Y.; Yan, Y.; Ren, Z. Z.; Ren, Z.; Sha, Z.; Fu, Z.; Xu, Z.; Xie, Z.; Zhang, Z.; Hao, Z.; Ma, Z.; Yan, Z.; Wu, Z.; Gu, Z.; Zhu, Z.; Liu, Z.; Li, Z.; Xie, Z.; Song, Z.; Pan, Z.; Huang, Z.; Xu, Z.; Zhang, Z.; and Zhang, Z. 2025. DeepSeek-R1: Incentivizing Reasoning Capability in LLMs via Reinforcement Learning. arXiv:2501.12948.

Ethiraj, V.; Vijay, D.; Menon, S.; and Berscilla, H. 2025. Efficient Telecom Specific LLM: TSLAM-Mini with QLoRA and Digital Twin Data. arXiv:2505.07877.

Kamoi, R.; Zhang, Y.; Zhang, N.; Han, J.; and Zhang, R. 2024. When Can LLMs Actually Correct Their Own Mistakes? A Critical Survey of Self-Correction of LLMs. Transactions of the Association for Computational Linguistics, 12: 1417–1440.

Kong, J.; Cheng, M.; Xie, X.; Liu, S.; Du, X.; and Guo, Q. 2024. ContrastRepair: Enhancing Conversation-Based Automated Program Repair via Contrastive Test Case Pairs. arXiv:2403.01971.

Ma, Y.; Li, Y.; Dong, Y.; Jiang, X.; Cao, R.; Chen, J.; Huang, F.; and Li, B. 2025. Thinking Longer, Not Larger: Enhancing Software Engineering Agents via Scaling Test-Time Compute. arXiv:2503.23803.

Maatouk, A.; Ampudia, K. C.; Ying, R.; and Tassiulas, L. 2025. Tele-LLMs: A Series of Specialized Large Language Models for Telecommunications. arXiv:2409.05314.

Pan, L.; Saxon, M.; Xu, W.; Nathani, D.; Wang, X.; and Wang, W. Y. 2024. Automatically Correcting Large Language Models: Surveying the Landscape of Diverse Automated Correction Strategies. Transactions of the Association for Computational Linguistics, 12: 484–506.

Team, K.; Du, A.; Gao, B.; Xing, B.; Jiang, C.; Chen, C.; Li, C.; Xiao, C.; Du, C.; Liao, C.; Tang, C.; Wang, C.; Zhang, D.; Yuan, E.; Lu, E.; Tang, F.; Sung, F.; Wei, G.; Lai, G.; Guo, H.; Zhu, H.; Ding, H.; Hu, H.; Yang, H.; Zhang, H.; Yao, H.; Zhao, H.; Lu, H.; Li, H.; Yu, H.; Gao, H.; Zheng, H.; Yuan, H.; Chen, J.; Guo, J.; Su, J.; Wang, J.; Zhao, J.; Zhang, J.; Liu, J.; Yan, J.; Wu, J.; Shi, L.; Ye, L.; Yu, L.; Dong, M.; Zhang, N.; Ma, N.; Pan, Q.; Gong, Q.; Liu, S.; Ma, S.; Wei, S.; Cao, S.; Huang, S.; Jiang, T.; Gao, W.; Xiong, W.; He, W.; Huang, W.; Xu, W.; Wu, W.; He, W.; Wei, X.; Jia, X.; Wu, X.; Xu, X.; Zu, X.; Zhou, X.; Pan, X.; Charles, Y.; Li, Y.; Hu, Y.; Liu, Y.; Chen, Y.; Wang, Y.; Liu, Y.; Qin, Y.; Liu, Y.; Yang, Y.; Bao, Y.; Du, Y.; Wu, Y.; Wang, Y.; Zhou, Z.; Wang, Z.; Li, Z.; Zhu, Z.; Zhang, Z.; Wang, Z.; Yang, Z.; Huang, Z.; Huang, Z.; Xu, Z.; Yang, Z.; and Lin, Z. 2025. Kimi k1.5: Scaling Reinforcement Learning with LLMs. arXiv:2501.12599.

Team, Q. 2025. QwQ-32B: Embracing the Power of Reinforcement Learning.

Wang, Q.; Zhang, X.; Li, M.; Yuan, Y.; Xiao, M.; Zhuang, F.; and Yu, D. 2025. TAMO:Fine-Grained Root Cause Analysis via Tool-Assisted LLM Agent with Multi-Modality Observation Data in Cloud-Native Systems. arXiv:2504.20462.

Xu, J.; Zhang, Q.; Zhong, Z.; He, S.; Zhang, C.; Lin, Q.; Pei, D.; He, P.; Zhang, D.; and Zhang, Q. 2025. OpenRCA: Can Large Language Models Locate the Root Cause of Software Failures? In The Thirteenth International Conference on Learning Representations.

Yang, A.; Li, A.; Yang, B.; Zhang, B.; Hui, B.; Zheng, B.; Yu, B.; Gao, C.; Huang, C.; Lv, C.; Zheng, C.; Liu, D.; Zhou, F.; Huang, F.; Hu, F.; Ge, H.; Wei, H.; Lin, H.; Tang, J.; Yang, J.; Tu, J.; Zhang, J.; Yang, J.; Yang, J.; Zhou, J.; Zhou, J.; Lin, J.; Dang, K.; Bao, K.; Yang, K.; Yu, L.; Deng, L.; Li, M.; Xue, M.; Li, M.; Zhang, P.; Wang, P.; Zhu, Q.; Men, R.; Gao, R.; Liu, S.; Luo, S.; Li, T.; Tang, T.; Yin, W.; Ren, X.; Wang, X.; Zhang, X.; Ren, X.; Fan, Y.; Su, Y.; Zhang, Y.; Zhang, Y.; Wan, Y.; Liu, Y.; Wang, Z.; Cui, Z.; Zhang, Z.; Zhou, Z.; and Qiu, Z. 2025. Qwen3 Technical Report. arXiv:2505.09388.

www.xueshuxiangzi.com