NetAutoLLM:基于LLM的自主化网络健康管理

唐枫枭*,王啸楠,张君健,赵 明
(中南大学计算机学院,湖南长沙 410083)

摘 要: 在现代网络运营和维护中,可靠的网络服务与系统健康管理至关重要。传统的网络健康管理方案依赖于规则匹配的简单算法或机器学习模型,难以应对通信状况复杂、信息难以拟合的网络通信环境。大型语言模型(Large Language Model,LLM)因其具备自主的思考与推理能力可以解决这些问题。目前,最先进的基于LLM的算法在通信网络领域取得了显著进展。然而,现有方案通常将LLM作为特征梳理与学习的工具,LLM本身被动接收数据,其自主性和思考能力受到了极大的限制。这导致目前基于LLM的方案无法直接应用于网络运营与维护任务中。为了解决这个问题,本文通过外接行为树(Behavior Trees,BTs)赋予LLM感知与修改网络环境的能力,并构建自主化的网络健康管理框架。此外,该框架通过双池设计保证各专家信息隔离与案例经验更新。同时,通过多专家讨论的方式加固操作安全性,以保证网络运维任务的完成。实验证明:相较于目前流行的模型,基于LLM的自主化网络健康管理(Network Autonomous health management based on LLM,NetAutoLLM)模型在异常检测任务中精度提高了8.44个百分点;在故障溯源任务中,精度提高了21.7个百分点,同时可以自动缓解故障。

关键词: 大语言模型(LLM);主动运维;网络健康管理;行为树(BTs);多专家讨论;双池信息存储

基金项目: 国家自然科学青年基金(No.62302527);湖南省自然科学基金(No.2025JJ90177,No.2024JJ9173);湖南省青年基金(No.2023jj40774)

中图分类号: TP277 **文献标识码:** A **文章编号:** 0372-2112(2026)03-1147-14

电子学报URL:http://www.ejournal.org.cn **DOI:**10.12263/DZXB.20250440

NetAutoLLM: Network Autonomous Health Management Based on LLM

TANG Fengxiao*, WANG Xiaonan, ZHANG Junjian, ZHAO Ming
(*School of Computer Science and Engineerings*, *Central South University*, *Changsha*, *Hunan* 410083, *China*)

Abstract: In modern network operation and maintenance, reliable network services and system health management are paramount. Traditional network health management solutions relying on machine learning models or rule-based algorithms struggle to address complex network communication environments with heterogeneous devices. Large language models (LLMs), with their reasoning and generalization capabilities, offer potential solutions to these challenges. While state-of-the-art LLM-based algorithms have achieved significant progress in network domains, existing approaches typically utilize LLMs merely as tools for feature extraction and learning, where models passively receive data with severely constrained autonomy and cognitive capabilities. This limitation hinders direct application of current LLM-based solutions to network operation and maintenance tasks. To address this issue, this paper proposes an autonomous network health management framework by integrating external behavior trees (BTs) that empower LLMs with network environment perception and modification capabilities. The framework ensures information authenticity through a dual-pool design distinguishing public and private information, while employing multi-expert discussions to verify action effectiveness and guarantee successful network maintenance. Experimental results demonstrate that compared to conventional models, network autonomous health management based on LLM (NetAutoLLM) achieves an 8.44 percentage points improvement in anomaly detection accuracy over mainstream models, and enhances fault localization precision by 21.7 percentage points, while enabling automated fault mitigation.

Keywords: large language models (LLM); autonomy; network health management; behavioral trees (BTs); multi-expert discussions; dual-pool information storage

Foundation Item(s): National Natural Science Foundation of China (No.62302527); Hunan Provincial Natural Science Foundation (No.2025JJ90177, 2024JJ9173); Hunan Provincial Natural Science Foundation (No.2023jj40774)

收稿日期:2025-05-29;录用日期:2026-02-26;责任编辑:覃怀银

*通信作者:唐枫枭

0 引言

近年来，随着5G/6G网络复杂度呈指数级增长，通信网络健康管理面临严峻挑战[1]。传统基于规则[2-3]或单一机器学习模型[4-6]的方法，难以应对无流量时指标空白的特征缺失场景以及多模态异构数据的融合挑战。如图1所示，在包含“无数据传输”的非平衡数据集上，模型的精度出现断崖式下跌，暴露了其对数据集的苛刻要求。

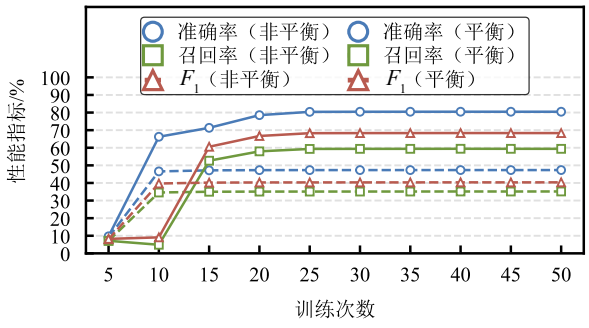


图1 LSTM模型在特征缺失与特征完整数据集上性能的对比

Figure 1 Comparison of the performance of the LSTM model on datasets with missing features and datasets with complete features

近期，大语言模型（Large Language Model，LLM）凭借其语义理解与推理能力为网络运维带来新契机[7]。然而，直接应用单智能体LLM面临三大瓶颈[8]。通用LLM缺乏通信领域的深度先验，在面对数据不平衡或特征缺失时，虽能推理但易产生致命幻觉[9]。此外，纯文本生成的决策过程缺乏显式的逻辑约束，难以满足网络的安全要求[10]。网络健康管理涉及海量多模态信息，单一模型难以在有限上下文窗口中同时兼顾全局态势与局部细节。

受此驱动，多智能体协作成为突破单模型能力边界的关键范式。通过分解任务并将任务分配给不同“专家模型”，可有效提升问题解决的精度。然而，现有多智能体框架在网络场景中存在明显短板。首先，缺乏确定性的执行逻辑使协作流程松散，难以完成闭环的健康管理任务。其次，依赖大量特定场景标注数据致使微调成本高昂。最后，专家间信息边界模糊，公有/私有信息混用导致推理干扰。

在此基础上，本文将行为树（Behavior Trees，BTs）的结构化决策能力与多专家LLM的领域认知能力相结合，提出了基于LLM的自主化网络健康管理（Network Autonomous health management based on LLM，NetAutoLLM）模型。与传统的多智能体协作方式不同，NetAutoLLM模型通过BTs实现分层任务分解与渐进式执行的策略。

针对缺乏确定性的执行逻辑问题，该策略将复杂任务自顶向下拆解为逻辑关联的子任务，并将其嵌入BTs的多层结构中。执行过程从根部开始逐层递进，每一层专注于完成一个原子化的子任务，其输出为下一层提供决策依据。每一层的原子节点严格限制在细粒度的预定义操作并辅以内置安全校验，有效约束了LLM潜在的幻觉风险。

针对传统运维数据复用性低的问题，本文提出认知迁移新范式。通过构建“专家认知流程数据集”微调LLM，使其学习通用的专家思维模式而非记忆特定案例，从而具备跨任务泛化能力。

针对专家间信息边界模糊，公有/私有信息混用的问题，本文引入双池信息存储机制。各专家通过公有池同步网络态势与任务进度，通过私有池隔离可调用工具与历史案例。该机制既避免了冗余信息干扰，又通过私有池的历史积累实现了模型的持续迭代。

本文在真实的网络上对NetAutoLLM模型和最近的各类算法模型进行了评估与对比。NetAutoLLM模型展现出卓越的性能。异常检测准确率达到91.89%，较现有最佳方法提升8.44个百分点，故障溯源平均定位精度提高21.7个百分点，同时支持典型故障场景的自动修复。这些结果充分验证了该框架在实际网络环境中的有效性和可靠性。

综上所述，本文的主要贡献如下：

（1）提出基于BTs的多专家LLM协作框架模型NetAutoLLM，利用结构化决策约束解决了网络健康管理中高维、缺失、强上下文依赖的复杂决策难题。

（2）设计认知迁移范式与专家认知流程数据集，使模型具备低成本、跨任务的泛化能力，有效缓解了高质量训练数据稀缺的问题。

（3）构建双池信息存储机制，实现了多专家间通用信息与私有信息的安全隔离与动态更新，保障了协作的高效性与可靠性。

1 相关工作

本节首先回顾传统网络健康管理方法的局限性，其次分析LLM在网络领域的应用现状及其作为单智能体决策时的瓶颈，最后探讨多智能体协作的研究进展，从而引出本文提出的NetAutoLLM模型的必要性。

1.1 传统网络健康管理方法

传统方法分为基于人工设计规则和基于机器学习的方法。基于人工设计规则的算法在优化网络系统方面发挥了重要作用。例如，Szilagyi等人[11]依据测量视无线电测量数据，并与配置文件捕获的正常行为进行比较。然而，随着网络规模指数级增长，人工设计的规则工程易面临组合爆炸问题，难以覆盖复杂的故障场景，且规则构建对运维人员的领域知识要求

极高[12-13]。

基于机器学习的算法尤其是深度神经网络(Deep Neural Networks,DNNs)试图通过数据驱动的方式解决上述问题[14-17]。在监督学习(Supervised Learning,SL)范式下,模型被用于异常检测和攻击分类[18]。在强化学习(Reinforcement Learning,RL)范式下,模型通过与环境交互优化资源分配和拥塞控制策略[19]。尽管取得了一定进展,但模型性能高度依赖输入特征的完整性[20]。如图1所示,在"无流量"导致指标空白的特征缺失场景下,传统模型因无法拟合缺失特征导致精度断崖式下跌。DNNs通常针对特定任务和数据分布进行训练,难以适应动态变化的网络环境,且在面对未知故障模式时表现不佳[21]。此外,高质量标注数据的稀缺也限制了其在真实网络中的部署。

1.2　大语言模型算法在网络中的应用

近期,LLM凭借强大的语义理解与推理能力,为网络运维带来了新范式。现有研究主要集中在利用LLM处理多模态信息感知任务。例如,网络大语言模型(Networking Large Language Model,NetLLM)[22]通过适配不同任务头来处理视口预测和集群调度。在空天地场景中,用于信道预测的大语言模型(Large Language Model for Channel Prediction,LLM4CP)[23]通过信道特征对齐提升了B5G/6G的信道预测精度。多尺度语义化异常检测模型(Multi-scale Semanticized Anomaly Detection Model,MSADM)[24]通过大小模型结合进行网络故障诊断任务。然而,仅利用LLM的参数量进行复杂模态的学习并非其极限,现有应用大多将LLM视为被动的知识库或预测器,而非主动的决策者。

这些通用的LLM往往缺乏通信领域的深度先验。在面对数据不平衡或特征缺失(如无流量场景)时,单智能体LLM虽能进行文本推理,但缺乏外部逻辑约束,易产生"一本正经胡说八道"的致命幻觉,这在网络中不可接受。此外,网络健康管理涉及海量时序流量、拓扑结构和日志文本。单一LLM受限于上下文窗口,难以同时兼顾全局态势感知与局部细节推理,导致在复杂决策中顾此失彼。同时,目前多智能体面临的另外一个问题在于,虽然文献[25]指出LLM已具备任务分解能力,但现有的微调或检索增强生成(Retrieval-Augmented Generation,RAG)方案仍停留在"问答"层面。LLM缺乏将推理结果转化为具体执行动作的闭环机制,无法真正实现"思考-决策-执行"的自主化运维。

1.3　多智能体协作健康管理方案

为突破单模型限制,多智能体协作成为前沿趋势。多智能体系统通过"协调行为"和"异构目标跟踪"显著提升了复杂环境下的协作能力[26]。在医疗健康领域,基于多智能体的协同感知-动态决策-精准干预范式[27]已证明其能够有效融合多模态数据。

然而,直接将通用多智能体框架迁移至网络领域存在明显短板。基于自然语言对话的协作缺乏确定性的执行逻辑,难以满足网络运维对实时性和安全性的严格要求。现有框架缺乏对公有态势信息与私有工具/案例信息的有效隔离,导致专家间推理干扰或敏感信息泄露[28]。

与此同时,BTs因其模块化、可解释性和分层结构,被广泛用于机器人与游戏人工智能(Artificial Intelligence,AI)的决策控制。一方面,基于攻防BTs的网络安全态势分析模型能够有效计算攻击路径与防御效果[29]。另一方面,融合YOLO11(You Only Look Once 11)与BTs的人机协作框架研究,通过实验表明:BTs实现的自适应角色管理比基于规则的系统响应速度更快[30]。

尽管多智能体协作与BTs在各自领域取得了进展,但现有研究尚未解决如何将两者融合以应对通信网络中的特征缺失与多模态异构挑战[31]。如何利用BTs的结构化决策能力约束多专家LLM的松散协作,并通过认知迁移机制降低数据依赖,仍是当前亟待解决的关键问题。

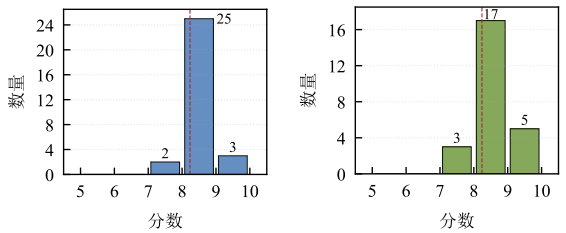
2　主要成果论述

2.1　专家微调

现有基于大语言模型的网络健康管理方案通常采用领域微调的方式将专业知识注入模型参数。然而,这种范式存在两个根本性局限。一方面,高质量的网络运维数据获取成本极高,尤其是故障诊断等关键场景的标注数据极为稀缺。另一方面,过度依赖特定领域数据会严重制约模型的泛化能力,难以适应快速演进的网络环境。

为克服上述方法的缺陷,本文提出了一种基于认知迁移的创新范式。该范式假设预训练的LLM已具备阅读网络管理任务相关文档所需的基础知识、基本的推理能力以及撰写简单的修复脚本的能力。其性能瓶颈主要源于在特定运维场景下缺乏专家级思维流程和经验性决策方案的注意力关注。如图2所示,本文选择Llama3-7b作为基座且在四类基础能力上得分较高,证明其具备完成基运维任务的知识储备。

本文基于信息技术基础架构库事件管理标准流程,将复杂的网络运维任务解耦为任务分配、数据检测与收集、数据分析与诊断、脚本执行、测试验证以及复盘优化六个核心职能。该划分对应运维标准程序操作中的"事件分发—监控与观察—根因分析—执行与变更—验证—沉淀"六个步骤。随后,通过这六



(a) 网络基础知识 (b) 设备配置能力

(a) Network basic knowledge (b) Device config skill

(c) 代码撰写能力 (d) 异常判断能力

(c) Coding skill (d) Abnormal detection skill

图2 Llama3-7b网络运维知识答题得分情况

Figure 2 Score distribution of Llama3-7b network operation knowledge

个步骤设计相应的专家角色。

为实现这一目标，本文构建了面向专家认知流程的多领域数据集，其结构如表1所示。该数据集主要包括专家类型与角色定位、任务描述与上下文、决策思考路径、关键判断节点、预期成果标准。同时，该数据集规避了具体的网络配置参数或故障解决方案等难以收集且不能泛用的细节，转而聚焦于专家在问题分析、决策制定和验证评估等环节的通用方法论。

在数据集构建后，本文通过低秩矩阵微调（Low-Rank Adaptation，LoRA）的方式对预训练后的Llama3-7b进行微调。所有的专家角色共享同一个预训练后的Llama3-7b基座模型，保留通用的网络协议知识与语言能力，实现认知迁移中的泛化基础。随后，针对上述六个角色，本文分别训练独立的低秩矩阵适配器。

在推理阶段，NetAutoLLM依据BTs的节点指令，通过动态地切换专家的低秩矩阵来调用对应的专家模型，以保证多角色间的参数隔离与协同。

评估结果如图3所示，经过认知迁移微调后，模型在分配不同角色时，其输出的信息密度与专业度显著提升。尤其是“统筹专家”因需兼顾全局，其输出覆盖面最广，而“诊断专家”则展现出更深层次的逻辑推理。由此证明了通过LoRA适配器注入专家思维流程，能够有效引导基座模型的注意力机制，使其生成更贴合真实专家思维的输出。

表1 模拟专家数据集重要字段展示

Table 1 Display of key fields in the simulated expert dataset

字段	介绍	案例
Expert	专家类型	任务分配专家
Task	任务描述	审核与优化现有的工作流程
ThinkingFlow	思考流	分析现有工作流程中的节点是否可完成任务； 识别不能单次执行的行为节点
Decision	在特定情况下可能需要做出决策或使用工具	“对行为是否可行产生困惑时”：通过测试专家验证流程与效果
Output	成果	优化后的任务行为流程

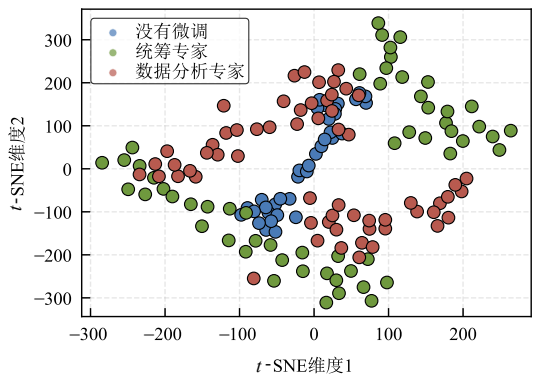


图3 不同角色的LLM对同一问题回答的tokens分布情况

Figure 3 Distribution of tokens in the responses of different roles' LLMs to the same question

2.2 BTs构建与更新

与传统依托于条件判定与动作执行结合的BTs不同。本文设计的BTs架构如图4所示，采用分层递归结构实现任务逻辑的模块化表达。每棵BTs对应一个完整网络任务，树状层级映射子任务依赖关系，叶节点则代表原子级可执行行为。

此外，本文定义了四类核心行为节点。感知节点（Perception）负责环境状态获取，实时采集网络指标与配置数据；思考节点（Thinking）执行数据分析，包括异常检测、根因推理等；行动节点（Action）实施具体操作，如参数调整、配置修改等；装饰节点（Decoration）进行结果验证与安全审计。

在实现层面，BTs的构建并非全部由LLM实现。每个复杂网络任务的完成均需一些基础任务，如数据收集、数据分析、数据整理等。因此，本文采用指导与启发式相结合的方式来辅助LLM生成BTs。首先，设计BTs的框架，即各层的通用子任务；然后，由LLM

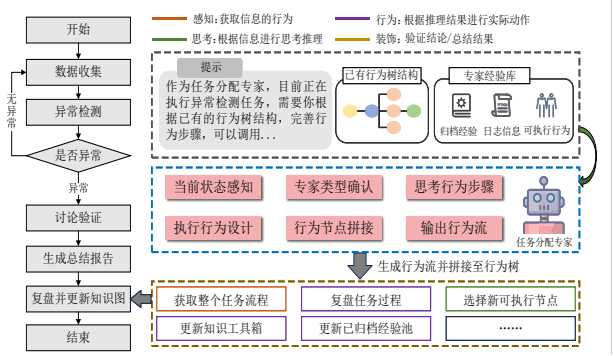


图4　BTs生成

Figure 4　Generation of behavior tree

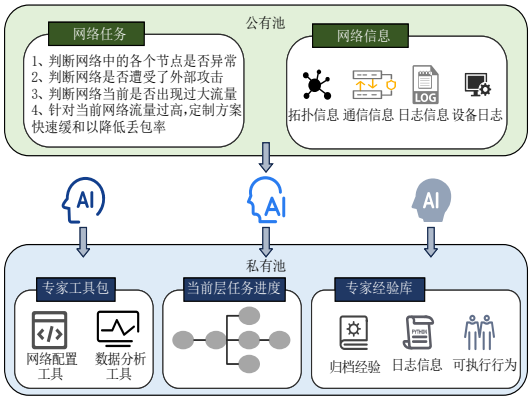


图5　双池存储方案

Figure 5　Dual-tank storage solution

依据网络任务与通信网络的状态来填充与重规划BTs的每一层子任务。

对于每一层的子任务,本文构建了分层可扩展的行为节点库。预置通用行为节点,如各类数据采集的接口调用工具。专家层则存储行为专用节点,如基础数据的分析工具。当LLM进行任务分解时,首先在宏观层面自主划分任务阶段,然后在微观层面从行为库匹配或生成具体节点。这种分层决策机制可以保留LLM主动思考的优势,同时通过预定义节点规范了执行过程。

2.3　双池信息存储

在多专家协作执行复杂网络健康管理任务时,长链条的多轮对话极易引发信息完整性与一致性挑战,具体表现为关键数据篡改、重要特征忽略及推理逻辑断层。

针对上述风险,本文提出一种双池的信息存储机制,通过构建全局共享与私有专属的信息空间,确保任务执行的可靠性与高效性。

如图5所示,系统将任务流中的信息划分为两个层级。全局公有池包含原始多模态网络数据、基础任务描述及全局进度状态。该域对所有专家透明开放,通过“单一数据源”原则强制保证基础数据的一致性,杜绝篡改风险。专家私有池封装了各专家的局部推理结果、历史私有案例库及专用工具接口。通过细粒度的访问控制列表,实现严格的物理隔离,防止外部信息污染私有推理链,降低专家模型的认知负荷与冗余干扰。

为保证全局共享域中信息的一致性与任务的有序推进,系统为每个BTs节点设计了轻量级的状态编码机制。当某位专家完成其节点任务后,更新该节点的状态编码为“已完成”。随后保存经提炼的、对后续步骤必要的输出信息。后续专家在启动自身任务前,通过查询相关节点的状态编码来确认前置依赖已全部满足,并获取已更新的全局信息作为输入。若检测到必要的前置节点尚未完成,系统将自动激活对应专家任务执行相应行为。

为实现运维经验的有效沉淀与系统能力的持续进化,本文设计了基于语义相似度的知识库异步更新策略。该策略与在线任务执行过程在时间上进行解耦,仅在每次完整任务结束后触发。

系统首先从本次任务过程中提取关键要素,如故障现象、诊断路径与解决方案,生成结构化候选案例。然后,利用预训练模型[如基于Transformer的双向编码器表示(Bidirectional Encoder Representations from Transformers,BERT)]计算候选案例与历史归档案例在场景描述上的语义向量相似度σ。更新策略遵循两条规则。若相似度σ低于阈值(如0.3),则判定为新颖故障模式,自动将该案例注入知识库并创建新索引。若相似度σ高于阈值(如0.8),则触发复盘专家模块,依据“主题鲜明度、逻辑完整性、范围覆盖度、可理解性”四项指标对新旧案例进行对比评估,以决定是否选用质量更高的新案例替换旧条目。

3　关键方案流程

本文提出的NetAutoLLM方案完成任务流程如图6所示。首先,统筹专家通过目前的网络情况指定任务,并依据BTs对任务进行分解,转化为可执行的BTs结构。其次,依据BTs,不同专家通过工具包完成行为节点。最后,为了保证结果可信,本文设计了行为验证章节保证其安全与可靠性。

本节将对本研究方案完成任务的各个环节进行详细阐明,并说明本文算法与设计方案的作用。在方案的初始阶段,LLM需以统筹专家的身份依据网络任务细分子任务,从而构建BTs主干。并依据可执行节点对BTs中的每层行为进行扩充。

3.1　行为执行

尽管微调后的专家角色具备任务的理解和推理

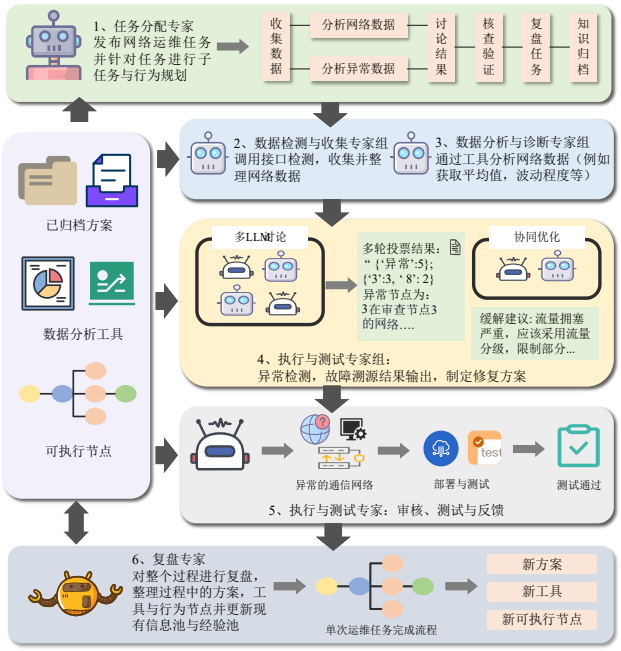


图6 NetAutoLLM任务完成结构

Figure 6 Structure of the NetAutoLLM task completion process

能力，但其感知数据与修改设备配置仍需借助外部工具的调用。此外，由于网络数据结构与关系复杂，LLM直接处理原始网络数据效率低下。LLM也可通过调用工具包，进行时序数据分析等行为。在需修改网络信息时，同样需调用类似于防火墙名单修改、配置校验等标准化工具或接口。

数据整理专家更多从全局出发，将全局网络信息进行整理形成文本信息。分析专家更多分析局部数据。多专家结合可有效防止信息遗漏。

当专家节点需执行具体行为时，自主选择工具，例如通过预置的Python max(·)函数获取指标峰值。对于工具包未覆盖的特殊需求，生成专家可生成相应的工具，从而保证任务的可完成性。

3.2 行为验证

当各专家模型产生幻觉问题时，为确保NetAutoLLM的可靠性与可控性，本文在行为验证阶段引入了多专家协同机制。该机制首先将不同阶段的任务输出划分为确定性任务与生成性任务两类。前者如检测或故障分类结果，具备明确的判定边界；后者如数据分析报告或故障缓解方案，则具有更强的生成性与开放性。其次，针对确定性任务，本文设计了一种基于多智能体投票的幻觉缓解机制。利用微调后的多个具备相同知识与提示的分析与诊断专家，在相同提示下并行推理并进行多数投票。最后，为避免平局，专家数量被配置为奇数。若检测到半数以上票类不一致、置信度低或意见严重分歧的情况，则判定为重大异常，此时触发异常上报与复盘模块生成报告，并强制重置健康管理流程（回滚至数据采集阶段）。

针对生成性任务，本文构建了迭代优化的协作方案。首先，由执行专家生成初步解决方案。其次，测试专家通过思维链提示对操作动作进行细粒度拆解与推演，系统性评估每一步的可行性、有效性、成本及风险。若推演发现不可执行或高风险步骤，系统则反馈优化信号至执行专家进行迭代修正；若方案被判定完全无效，则重启执行智能体生成新方案。最后，针对涉及物理硬件协调的关键操作，系统设置了人工介入阈值，一旦触发即挂起并告警。

4 实验结果与分析

4.1 实验设置

4.1.1 实验环境设置

在默认情况下，本文采用Llama3-7b作为基础的LLM。随后，本文使用NetAutoLLM对Llama3进行适配，以处理三个网络任务：网络异常检测（Network Anomaly Detection，NAD）、故障溯源（Network Fault Diagnosis，NFD）和故障缓解（Network Fault Mitigation，NFM）。

为抑制模型幻觉并防止灾难性故障，本文设计了动态监控与中止机制：当对话轮次超过预设上限或智能体间分歧度超出阈值时，系统将主动中止当前管理流程并上报运维人员，从而确保复杂场景下系统的稳健性与可控性。

本文基于通用Wi-Fi开发版构造真实的具有27个节点的物联网（Internet of Things，IoT）网络场景。并在网络中，注入应用程序崩溃、恶意流量、节点崩溃、信号弱、中央处理器（Central Processing Unit，CPU）过载与防火墙配置错误六个不同的网络环境，用于训练和测试。这些设置涵盖了影响模型性能的关键因素。例如，在异常检测任务中，本文的环境设置考虑了设备的发包数量和变化频率，以及通信过程中的带宽。

本文在一台配备四个Intel（R）Xeon（R）Silver 4310 CPU和四个NVIDIA RTX A6000 GPU的Linux服务器上对Llama3-7b模型进行部署与微调。通过构建Java服务器中包含大量监控和收集网络中各节点的通信流量、设备状态、日志等信息的接口，以模拟真实场景环境。

4.1.2 基线方法

针对网络异常检测任务，本研究选取两类典型基线模型：基于SL的谱残差卷积神经网络（Spectral Residual Convolutional Neural Network，SR-CNN）[32]、混合概率主成分分析的卷积长短期记忆网络（Convolutional LSTM with Mixtures of Probabilistic Principal Component analysis，CL-MPPC）[33]、异常检测BERT（Anoma-

lyBERT)和双注意力对比表示学习模型(Dual Contrastive detector, DCdetector)[34]、基于生成或重建的基于Transformer的异常检测模型(Transformer-based Anomaly Detection model, TranAD)[35]、插补扩散模型(Imputed Diffusion models, ImDiffusion)[36],以及基于大语言模型的NetLLM。

在故障溯源任务中,重点比较时序假设检验方法Randomwalk(随机游走)[37]和因果推理框架(Causal Inference-based Root Cause Analysis, CIRCA)[38]的性能表现。由于FS任务领域缺乏通用解决方案,故多数方法局限于特定故障类型修复。为此,本研究通过构建恶意流量注入的动态攻防场景,系统性地验证NetAutoLLM在故障自愈方面的闭环处理能力。

同时,本文对比了均为7×10^9大小的Llama3、Qwen与DeepSeek-r1证明方案的有效性。此外,自动化的故障诊断需LLM具有任务分解与任务执行能力。为评估智能体的任务分解效能,本文选取其他领域的任务分解方案,主要包括提示范式可靠架构表征工具(Reliable Architecture Characterization Tool, ReACT)、多层级决策树(Multi-Level Decision Tree, MLDT)[39]和知识增强的推理算法WorfBench评估协议(WORFbench EVALuation protocol, WORFEVAL)[40]作为对比基准。本文在网络运维场景中复现所有模型,并通过与NetAutoLLM的任务分解能力进行对比,以证明其在任务分解中的有效性。

4.1.3　评估指标

对于任务分解,考虑分解结果的准确性、相关性、覆盖度、平均步数与是否需要多轮的人机交互。

针对异常检测、故障诊断和故障溯源任务,本文采用了通用的准确率(Accuracy)、精确率(Precision)、召回率(Recall)和F_1分数性能指标。

针对LLM生成特性,引入基于Rouge的文本评分以评估NetAutoLLM生成的报告质量。特别地,在NFD任务中设计动态候选评估机制TOP@k,通过调节候选集数量k分析模型鲁棒性。对于NFM任务,通过检测从故障注入到故障缓解过程中的CPU利用率、丢包率、延迟与带宽变化情况,展示修复方案的有效性。对于NFD,本文主要对比了具有代表性的ReACT范式、MLDT及知识增强型WORFEVAL框架。设计输入均为网络健康管理中的各项任务,即异常检测,故障溯源和故障修复。且本文统一采用网络运维知识库作为输入源。约束各方案的LLM参数规模在7×10^9量级,并且通过专家标注构建标准流程作为评估基准。

4.2　任务分解与分配

在评估体系构建方面,验证准确性(分解结果与标准流程的文本相似度)、相关性(关键步骤召回率)、覆盖度及平均执行步数四个指标。

如表2所示,本方案在标准测试集上取得96%平均覆盖度,较最优基线WORFEVAL提升5个百分点,准确率提高16个百分点。相较ReACT范式与MLDT,平均步数减少近2/3,子行为划分与真实基准相关性达86%、覆盖度达96%。ReACT因缺乏先验知识导致步骤冗余效率低下,MLDT因结构僵化在训练数据外任务变体中覆盖度与准确性下降,WORFEVAL虽引入先验知识但相比经该类型专家微调后的专家模型在任务分解全面性与效率上仍存差距,本方案通过动态任务分解机制实现覆盖度、准确率及执行效率的综合优势。

表2　各方案任务分解能力对比

Table 2　Comparison of task decomposition capabilities of each scheme

模型	准确性/%	相关性/%	覆盖度/%	是否需要多轮交互	平均步数
ReACT	42	46	11	是	23
MLDT	46	52	49	是	41
WORFEVAL	66	74	91	否	7
Ours	82	86	96	否	9

4.3　异常检测

异常检测作为网络基础且至关重要的任务,通过对数据异常点(如平均值、极值点或抖动情况)的判断,检测网络中的异常现象,并依据异常表现特征有效推断出潜在的故障类型。本文收集了不同网络场景下的节点流量信息、节点配置信息以及节点的状态信息,它们本质上均为结构化或非结构化的文本。LLM能够有效理解并分析这些信息,从而完成异常检测任务。

然而,传统模型无法有效处理这些信息。因此,为了确保不同算法能够在公平且可比的基础上进行评估,本文将所有原始文本数据统一转换为结构化的时间序列数据格式。数据按固定时间间隔进行采样和聚合,形成一个等间隔的时间序列。对于部分数据中明确的数值指标,如CPU利用率、网络延迟等直接提取其数值作为时间序列的维度。

针对文本中表示状态或类别的信息(如节点类型、通信方式等)通过统一编码来表示。在该过程中,尽可能保留原始文本中蕴含的关键状态信息和语义含义,并将其量化为模型可处理的数值形式。最终,本文设计的数据集包含网络节点的特征及节点间的通信流信息,训练目标即为判定数据是否存在异常或识别具体的故障类型。

如图7所示,混淆矩阵呈现了NetAutoLLM模型在异常检测任务中的预测结果与实际标签之间的匹配

关系。结果表明：NetAutoLLM 在异常检测场景下对异常样本展现出较高的分类准确率。

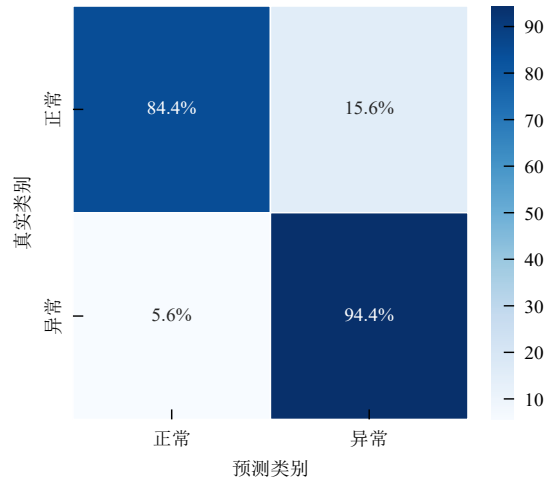


图 7 异常检测混淆矩阵

Figure 7 Confusion matrix for anomaly detection

本节展示了不同模型在网络领域异常检测的结果。包括基于学习的模型、基于生成模型以及基于 LLM 的模型。

如表 3 所示，在网络领域，NetAutoLLM 相较于其他模型均有更好的效果。在异常检测的基础上，本文同样对比了算法在故障诊断即多分类任务上的精度。由于基于生成的模型几乎完全无法学习多故障场景下性能变化的趋势，因此，该类模型不参与对比。为了公平对比，本文对所有基于分类的模型进行对比。

本文进一步评估了 NetAutoLLM 在三个典型运维场景下的准确性、推理时间以及总时间。推理时间主要为诊断专家模型诊断所需时间；总时间是从收集数据开始到结束的时间，包括了请求接口的等待时间以及信息传递时间。

如表 4 所示，在应用程序崩溃诊断场景中，模型表现最优，准确率最高，且核心推理耗时最短，这是由于程序崩溃后易在设备的运行状态以及日志中发现问题，推理时间主要用于推理与投票。在配置信息错误识别场景中，配置错误会导致日志异常，因此模型仍可较快地检测异常。在恶意流量检测场景中，模型面临较大挑战，其推理时间最长，准确率最低，其主要原因在于模型难以区分网络中的流量是恶意还是正常的访问，仅当流量显著区别于正常情况时才可以识别。此外，在故障分类中，该类情况极大概率被 NetAutoLLM 认为是流量拥塞故障而非恶意的攻击。

在异常检测和故障溯源的过程中，双池的信息存储方式很大程度地保证了信息的完整性。如图 8 所示，在失去私有池时，异常检测与故障溯源的精度下降。在失去公有池后，由于当前专家并不了解整个任务以及当前的进度，同时数据的丢失较为严重，导致模型精度显著下降。

4.4 故障溯源

故障溯源是网络运维中定位根本故障节点的关键环节，其准确性直接影响故障恢复效率。为评估不同算法的故障定位能力，本文采用故障注入方法，通过修改网络设备配置模拟故障场景。并基于该场景对比了 RandomWalk、CIRCA 和 NetAutoLLM 三种算法的性能差异。

需要注意的是，RandomWalk 和 CIRCA 并不能直接使用原始网络数据进行故障溯源。针对 RandomWalk，本文将原始数据中的流量信息与日志信息转化为可输入的时序信息。CIRCA 方法需使用关系图结构作为输入。因此，本文将网络拓扑结构作为图主干，并将每一个节点的特征作为影响因素，从而实现基于因果推理的故障溯源。而 NetAutoLLM 凭借其卓越的特征提取能力，可直接处理原始网络流量数据、设备日志及性能指标，无需复杂预处理，显著简化故障溯源流程并减少信息损失。

表 3 各算法的异常检测性能展示 单位：%

Table 3 Presentation of anomaly detection performance for each algorithm unit：%

模型	异常检测				故障诊断			
	准确率	召回率	精确率	F_1分数	准确率	召回率	精确率	F_1分数
SR-CNN	73.45	78.85	83.67	76.05	63.25	71.67	61.25	66.05
CL-MPPCA	74.80	79.72	84.91	77.18	67.61	73.24	71.13	72.17
AnomalyBERT	79.31	84.66	86.71	81.90	77.78	77.78	79.94	78.85
DCdetector	82.51	88.58	87.50	85.44	80.32	80.56	82.38	80.58
TranAD	87.62	89.69	88.64	94.00	—	—	—	—
ImDiffusion	83.45	87.23	90.25	85.30	—	—	—	—
NetLLM	82.00	81.20	87.70	84.32	79.46	78.33	79.12	78.72
Ours	**91.89**	**93.44**	**95.83**	**94.61**	**86.11**	**86.11**	**86.90**	**86.31**

注：加粗字体为最优结果，下同。

表4　NetAutoLLM在不同场景下的异常检测所需时间

Table 4　Time required for NetAutoLLM to perform anomaly detection in different scenarios

场景	准确率/%	推理时间/min	总时间/min
恶意流量	91.33	2.57	9.26
应用程序崩溃	96.67	0.58	5.38
配置信息错误	92.45	0.88	7.92

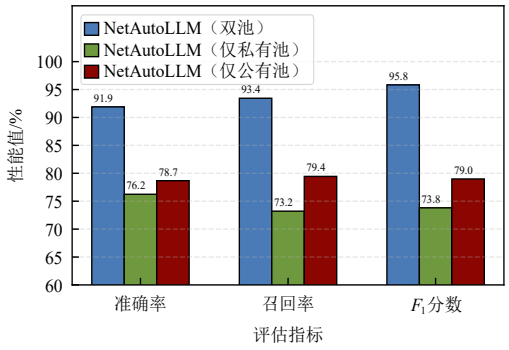


图8　双池信息存储对NetAutoLLM性能指标的影响

Figure 8　The Impact of Dual-Pool information storage on the performance indicators of NetAutoLLM

为了保证实验的公平性,本文对三种方法均只使用网络拓扑、网络流量与日志信息三个模态数据作为输入。考虑RandomWalk本身不接受时序外的数据,本文将拓扑信息压缩为列表,同样作为时序特征的方式并入RandomWalk的输入中,以保证RandomWalk可得到网络拓扑信息。

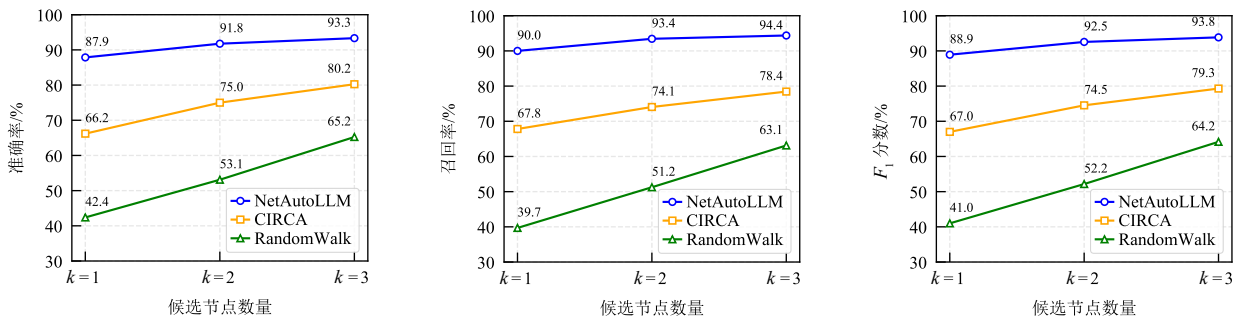
实验结果如表5所示,NetAutoLLM在故障节点识别准确性和稳定性方面均显著优于传统方法,为网络故障的快速定位与修复提供了可靠的技术支撑。

表5　各算法的故障溯源任务性能展示　　单位:%

Table 5　Performance comparison of fault traceback tasks　unit:%

算法	准确率	召回率	F_1分数
RandomWalk	53.10	51.25	52.16
CIRCA	75.00	74.06	74.53
NetAutoLLM	**91.67**	**93.44**	**92.55**

如图9所示,当候选节点数量从1增加到3时,NetAutoLLM在三项指标上均保持领先优势。在单候选节点场景下,NetAutoLLM的准确率达到87.9%,较CIRCA和RandomWalk分别提升了21.7个百分点和45.5个百分点。在三候选节点中,其F_1分数仍维持在93.8%。NetAutoLLM的优势在于采用的层次化与多专家讨论的推理机制,能够通过多粒度特征分析与多次讨论完善推理结果,有效排除干扰节点。



(a) 准确率　(b) 召回率　(c) F_1分数

(a) Accuracy　(b) Recall　(c) F_1-Score

图9　故障溯源模型在Top@k下的准确度、召回率以及F_1分数变化情况

Figure 9　Changes in accuracy, recall rate and F_1-score of the fault traceability model under Top@k

图10中的柱状图证明了多专家讨论对NetAutoLLM溯源任务的影响。由图10可知,在多个候选项的讨论与投票后,模型的稳定性与性能指标均显著提高。

4.5　故障缓解

故障缓解方案是网络健康管理体系中的核心环节。当网络遭遇流量激增,设备资源面临过载风险时,CPU使用率飙升、延迟显著增加、丢包现象加剧以及带宽占用激增等问题将接踵而至,在极端情况下,超大数据包甚至可能直接导致网络瘫痪。

本文以恶意流量场景为例,证明方案的有效性。在传统处理方式中,运维人员通常采用手动方式降低发送端码率来缓解压力。而当NetAutoLLM系统检测到目前高流量引发的异常丢包后,任务分配专家模型设计并发布缓解流量异常的任务,数据分析与诊断专家模型排查问题,并由执行专家生成并执行缓解脚本。

如图11所示,本文首先尝试了低强度的恶意流量,大致的攻击强度在200包/s以内。丢包率上升为接近10%,延迟超过了48 ms,而CPU使用率已达到约50%。此时,NetAutoLLM并未发现异常,在高强度攻

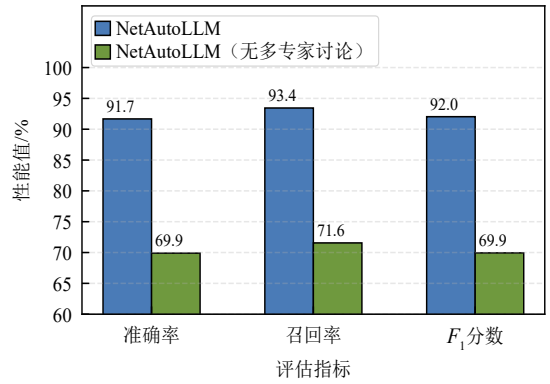
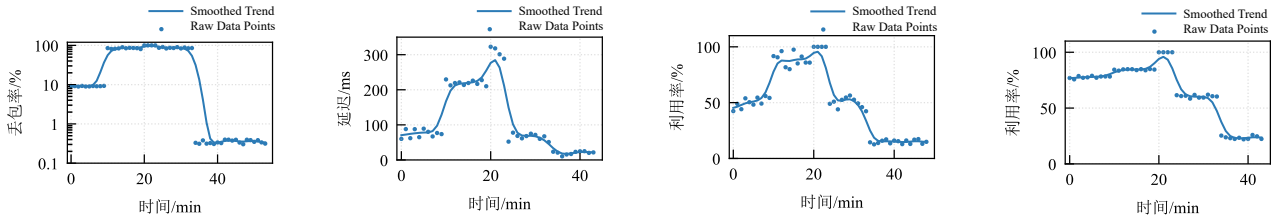


图 10 多专家讨论对 NetAutoLLM 的性能影响

Figure 10 Discussion among multiple experts on the impact on the performance of NetAutoLLM

击（1 000 包/s）后，丢包率上升为约 80%，延迟超过 220 ms，CPU 利用率超过 89%。此时，数据分析专家已经发现异常问题，但不清楚是否为恶意流量。然后，本文通过传入大流量包导致了整个网络的崩溃即丢包率 100%，延迟超过 300 ms，CPU 也已经过载。在经过 NetAutoLLM 进行的故障缓解后，丢包率、CPU 利用率和延迟都明显降低最终达到与正常情况下相持平。

部分信息如图 12 所示，本文在注入高强度恶意流量之后，数据检测与收集专家根据日志信息判断出性能变化，随后根据数据分析专家给出故障原因并由执行专家生成解决方案。针对 200~1 000 包/s 的攻击强度梯度，动态生成分级限流规则，并通过自动生成的 Python 脚本实现实时流量整形。



(a) 丢包率变化情况 (b) 平均延迟变化情况 (c) CPU 利用率变化情况 (d) 带宽利用率变化情况

(a) Packet loss rate changes (b) Average latency change trend (c) CPU utilization change trend (d) Band utilization change trend

图 11 恶意流量下网络中丢包率、CPU 利用率、带宽以及延迟的指标变化情况

Figure 11 Changes in packet loss rate, CPU utilization, bandwidth, and latency under malicious traffic

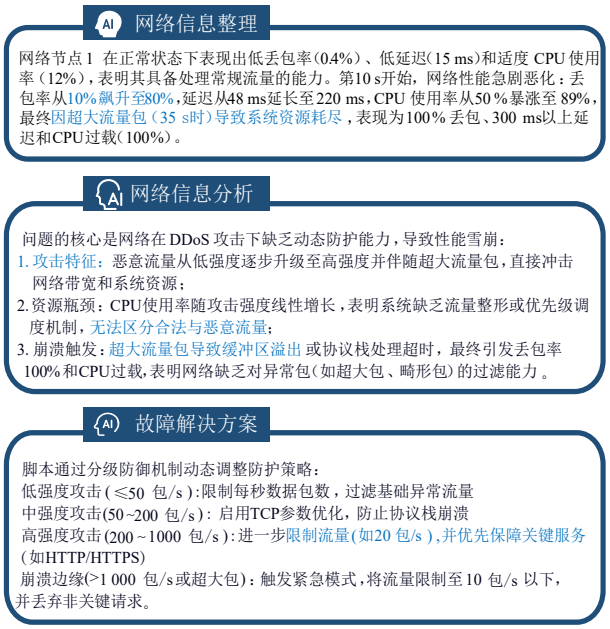


图 12 LLM 整理、分析并生成缓解方案部分文本展示

Figure 12 Text display of LLM-Based organization, analysis, and mitigation strategy generation

脚本信息如图 13 所示，NetAutoLLM 生成的防御脚本实现了分级流量控制机制。该脚本采用四级防御策略设计，基础防护通过 iptables 实现 100 包/s 的流量限制；中级防护增加传输控制协议（Transmission Control Protocol，TCP）栈的参数调优；高级防护进一步优化连接队列参数；紧急模式对关键服务端口实施严格限速文本质量评估。

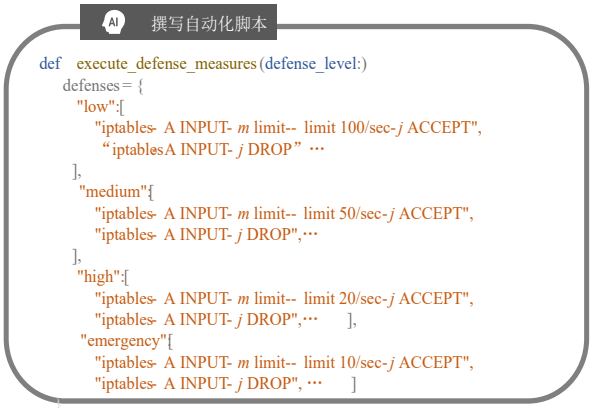


图 13 LLM 自动化生成脚本部分展示

Figure 13 Partial demonstration of LLM-Automated script generation

在模型推理阶段，为全面评估生成文本的质量，本研究采用 Rouge 评分体系对网络信息总结报告、任务分解方案以及故障缓解解决方案进行量化分析。

Rouge评分作为衡量生成文本与参考文本相似度的常用指标,能够有效反映模型在文本生成任务中的性能表现,其评分范围通常在0~1。在学术研究中,Rouge-L得分达到0.3~0.5,通常表明生成文本与参考文本具有较好的语义一致性,而专业领域文本由于术语特异性,其基准分数往往比通用领域低10~15个百分点。

本研究以GPT-4o生成并经专家审核修改的网络信息报告作为标准文本,计算各模型在所有任务中的Rouge评分平均值。该评估方法能够客观、准确地衡量不同模型在生成网络相关文本时的质量差异,为后续模型优化与应用提供有力依据。

本文在此处的质量评估主要针对网络信息总结出的报告、任务分解的方案以及故障缓解的解决方案。并对任务过程中出现的文本计算Rouge评分后,计算其平均值。

由于NetAutoLLM本身基于Llama3(7×10^9)实现。因此,本文对比的生成方案包括基线Llama3(7×10^9)和同样7×10^9参数量的QWen与DeepSeek-r1。

本文计算了模型在Rouge评分中的表现。最终结果如表6所示,NetAutoLLM相较于未训练的Qwen、Llama3及DeepSeek-r1在文本生成质量上具有明显的进步。尤其是在短文本的生成结果中表现优异。

表6　文本生成质量评估表　　单位:%

Table 6　Text generation quality evaluation form　　unit:%

Method	ROUGE-1		ROUGE -2		ROUGE-L	
	Precision	Recall	Precision	Recall	Precision	Recall
QWen(7×10^9)	14.15	17.21	1.12	1.28	11.88	18.28
Llama3(7×10^9)	14.61	21.43	4.18	5.23	17.12	20.95
DeepSeek-r1(7×10^9)	14.88	20.19	5.77	6.75	19.62	23.47
NetAutoLLM	**45.34**	**54.76**	**16.15**	**18.92**	**34.09**	**26.32**

4.6　在线学习能力验证

为了进一步证明NetAutoLLM面对未知故障时的表现以及信息池更新的重要性。

由于单恶意流量的故障场景过于经典可能已经被基座模型学习。本文在存在恶意流量故障场景的网络中,注入了信号弱故障。由于NetAutoLLM的各个专家私有池中无关于信号弱故障的相关场景描述以及介绍。如图14所示,NetAutoLLM虽然可以检测出网络中似乎存在异常,但是无法有效识别故障节点,因此故障溯源的F_1分数仅有43%。但是随着故障次数的增多,信号弱的故障表现被加入到专家私有池的案例中,故障溯源的精度也有明显增长,且随着案例描述的不断优化,故障溯源的精度也不断上涨。但需要承认的是,NetAutoLLM自己生成的案例描述相较于人工撰写的案例描述质量有差距,因此最终的精度相较于人工撰写的案例降低了13%。

同时,本文展示了NetAutoLLM关于更新案例的评分变化情况。为了更好地在图中展示内容,本文将分数从0~10分归一化为0~1。如图14所示,随着更新轮次的变化,故障现象的描述分数也在不断变化,并最终能够达到0.83(即8.3分)。

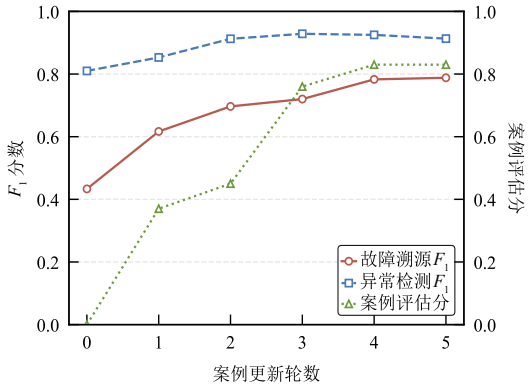


图14　随着案例更新的次数增加异常检测与故障溯源F_1分数变化

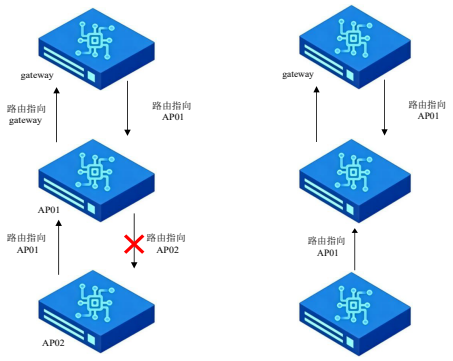
Figure 14　F_1-Score variation in anomaly detection and fault localization with increasing case updates

4.7　性能极限探索

为了进一步探索NetAutoLLM的极限。如图15所示,本文通过修改路由表,在网络中设计了更复杂的路由配置错误场景。场景图15(a)为节点AP02,需通过AP01才可联网。本文修改了AP01的路由表参数,使其无法回应AP02的请求。

实验证明:在无相关案例支撑的条件下,NetAutoLLM可轻松识别性能下降明显的异常节点AP02(丢包率、延迟等骤降)。但由于缺少相关知识储备,在追溯问题根源时始终需要更多轮的对话。初期模型会收集已检测的异常节点信息,并在后续过程中分析相邻节点信息。由于当前故障涉及节点较少,因此在分析AP02附近的AP01时,可快速定位故障原因。对于修复方案,初期生成专家给出修改AP02使其可以直接与gateway通信的修复方案,当测试专家意识到用户意图后,在多轮优化后给出修改AP01的路由表的正确方案。

在首次遇到并解决问题后,在给出相似的故障案



(a) 单向通信 (b) 路由环路

(a) One-way communication (b) Loop routing

图15 路由错误配置故障场景子结构展示

Figure 15 Substructure bisualization of routing misconfiguration failure scenario

例现象中，NetAutoLLM可以准确定位故障节点AP01的配置问题，并给出正确的缓解方案。即对AP01的网关进行更正。本文整理并简化了NetAutoLLM对注入故障的现象和解决方案，输出结果如图16所示。

网络信息分析

数据分析专家问题描述:
发现Mesh-AP01节点的路由配置存在问题，经分析，Mesh-Gateway节点路由表配置合理，可正常访问外部网络，Mesh-AP02节点路由配置基本正确，能与其他节点通信，而Mesh-AP01节点存在错误默认网关设置(Gateway:10.0.99.99)，可能导致数据包无法到达目的地址。
问题所在: 主要出现在Mesh-AP01节点
推理专家推理分析:
默认网关配置错误:Mesh-AP01节点默认网关设置为10.0.99.99，此地址可能无效或不可达，会导致所有经该节点的数据包无法到达目标网络。可能原因: 一是配置工具使用错误或手动配置失误; 二是硬件或软件故障导致配置不一致。建议: 立即纠正Mesh-AP01的默认网关设置，重新验证网络连通性。若问题持续，进一步检查相关配置文件和系统状态。
生成解决方案:
假设需要将Mesh-AP01的默认网关更正为'10.0.1.1'，可以执行以下命令:

```bash
sudo ip route del default via 10.0.99.99 dev eth1
sudo ip route add default via 10.0.1.1 dev eth0
```

图16 NetAutoLLM分析并缓解单向通信故障关键内容展示

Figure 16 Key content demonstration of NetAutoLLM-Based analysis and mitigation for unidirectional communication failures

然而，在面对图15(b)所示的多节点环路故障等更具挑战性的场景时，即便为NetAutoLLM提供了相关案例，系统也仅能检测到异常节点(如gateway和AP02)，而难以精准定位故障源头。该现象揭示了NetAutoLLM在当前阶段的一个核心瓶颈：当多个节点因故障耦合而同时表现出严重异常时，模型在复杂因果推理与证据去歧义化的能力上存在局限。尽管在少数能够准确定位故障源的案例中，NetAutoLLM证明了其生成有效解决方案的潜力，但定位环节的不可靠性从根本上制约了其在复杂拓扑中的整体效能。

上述性能边界的探索表明：受限于使用的基座模型推理能力与案例知识的完备性，NetAutoLLM在处理涉及多节点、多链路故障耦合的复杂场景时面临挑战。尽管如此，NetAutoLLM框架的出现使大语言模型向网络全自动健康管理领域的应用迈出了关键的第一步。它成功地构建了一个从数据收集到数据分析到异常检测和故障缓解的自主闭环原型。未来，研究人员可基于NetAutoLLM，通过集成更具针对性的专业化模型、引入更强大的因果发现算法，以及构建大规模的网络故障仿真环境，以逐步完善这片技术拼图，最终推动通信网络运维走向高度的智能化与自动化。

5 结论

本文探索了利用大型语言模型LLMs进行主动式网络运维的能力，以降低网络运维的人工设计成本，并实现强大的泛化能力。为实现这一目标，本文提出了NetAutoLLM。在宏观上，将BTs与经过认知迁移微调的LLM相结合。利用BTs对LLM进行约束和引导，通过设计与迭代分层BTs，使其能够自主完成网络健康管理任务。在微观上，通过双池信息存储将任务与网络等通用信息和专家当前所操作的私人信息进行隔离，以保证数据的完整性，并通过多专家讨论保证结果稳定性。通过在网络领域的三个应用案例，本文展示了NetAutoLLM如何能够自主在多个网络任务中展现其解决问题的能力。并且通过与基准模型的对比，证明了NetAutoLLM相较于传统模型在网络领域完成任务的优势。

参考文献

[1] Yuan X, Tang F X, Zhao M, et al. Joint rate and coverage optimization for the THz/RF multi-band communications of space-air-ground integrated network in 6G[J]. IEEE Transactions on Wireless Communications, 2024, 23(6): 6669-6682.

[2] Arun V, Balakrishnan H. Copa: Practical delay-based congestion control for the Internet[C]//Proceedings of the 2018 Applied Networking Research Workshop. New York: ACM, 2018: 19.

[3] Meng Z L, Wang M H, Bai J S, et al. Interpreting deep learning-based networking systems[C]//Proceedings of the Annual Conference of the ACM Special Interest Group on Data Communication on the Applications, Technologies, Architectures, and Protocols for Computer Communication. New York: ACM, 2020: 154-171.

[4] Yuan X, Wang X N, Tang F X, et al. MPITE: Multidimensional performance evaluator for interpretable and traceable network performance evaluation[J]. IEEE Transac-

tions on Networking, 2025, 33(5): 2458-2473.

[5] Li Z, Zhu X Q, Gahm J, et al. Probe and adapt: Rate adaptation for HTTP video streaming at scale[J]. IEEE Journal on Selected Areas in Communications, 2014, 32(4): 719-733.

[6] Lin X J, Xiong G, Gou G P, et al. ET-BERT: A contextualized datagram representation with pre-training transformers for encrypted traffic classification[C]//Proceedings of the ACM Web Conference 2022. New York: ACM, 2022: 633-642.

[7] Guo Z Q, Tang F X, Luo L F, et al. A survey on applications of large language model-driven digital twins for intelligent network optimization[J]. IEEE Communications Surveys & Tutorials, 2026, 28: 3388-3411.

[8] Tang J, Tang F X, Long S F, et al. Utilizing large language models for advanced optimization and intelligent management in space-air-ground integrated networks[J]. IEEE Network, 2025, 39(5): 173-181.

[9] Xia Z X, Zhou Y J, Yan F Y, et al. Automatic curriculum generation for learning adaptation in networking[PP/OL]. V2.arXiv (2022-09-08)[2025-05-29]. https://doi.org/10.48550/arXiv.2202.05940.

[10] Dhamdhere A, Teixeira R, Dovrolis C, et al. NetDiagnoser: Troubleshooting network unreachabilities using end-to-end probes and routing data[C]//Proceedings of the 2007 ACM CoNEXT conference. New York: ACM, 2007: 1-12.

[11] Szilagyi P, Novaczki S. An automatic detection and diagnosis framework for mobile communication systems[J]. IEEE Transactions on Network and Service Management, 2012, 9(2): 184-197.

[12] Peng Y H, Bao Y X, Chen Y R, et al. DL2: A deep learning-driven scheduler for deep learning clusters[J]. IEEE Transactions on Parallel and Distributed Systems, 2021, 32(8): 1947-1960.

[13] Mao H Z, Netravali R, Alizadeh M. Neural adaptive video streaming with pensieve[C]//Proceedings of the Conference of the ACM Special Interest Group on Data Communication. New York: ACM, 2017: 197-210.

[14] Bentaleb A, Timmerer C, Begen A C, et al. Bandwidth prediction in low-latency chunked streaming[C]//Proceedings of the 29th ACM Workshop on Network and Operating Systems Support for Digital Audio and Video. New York: ACM, 2019: 7-13.

[15] Mei L F, Hu R C, Cao H W, et al. Realtime mobile bandwidth prediction using LSTM neural network[M]//Passive and Active Measurement. ChamSpringer International Publishing, 2019: 34-47.

[16] 朱晓荣, 张佩佩. 基于GAN的异构无线网络故障检测与诊断算法[J]. 通信学报, 2020, 41(8): 110-119.
Zhu Xiaorong, Zhang Peipei. Fault detection and diagnosis method for heterogeneous wireless network based on GAN[J]. Journal on Communications, 2020, 41(8): 110-119. (in Chinese)

[17] Kan N W, Jiang Y K, Li C L, et al. Improving generalization for neural adaptive video streaming via meta reinforcement learning[C]//Proceedings of the 30th ACM International Conference on Multimedia. New York: ACM, 2022: 3006-3016.

[18] Jeong Y, Yang E, Ryu J H, et al. AnomalyBERT: Self-supervised transformer for time series anomaly detection using data degradation scheme[PP/OL]. V1. arXiv (2023-05-08)[2025-05-29]. https://doi.org/10.48550/arXiv.2305.04468.

[19] Mao H Z, Schwarzkopf M, Venkatakrishnan S B, et al. Learning scheduling algorithms for data processing clusters[C]//Proceedings of the ACM Special Interest Group on Data Communication. New York: ACM, 2019: 270-288.

[20] Yen C Y, Abbasloo S, Chao H J. Computers can learn from the heuristic designs and master Internet congestion control[C]//Proceedings of the ACM SIGCOMM 2023 Conference. New York: ACM, 2023: 255-274.

[21] Wang X W, Lin X H, Dang X C. Supervised learning in spiking neural networks: A review of algorithms and evaluations[J]. Neural Networks, 2020, 125: 258-280.

[22] Wu D, Wang X D, Qiao Y Q, et al. NetLLM: Adapting large language models for networking[C]//Proceedings of the ACM SIGCOMM 2024 Conference. New York: ACM, 2024: 661-678.

[23] Liu B X, Liu X Y, Gao S J, et al. LLM4CP: Adapting large language models for channel prediction[J]. Journal of Communications and Information Networks, 2024, 9(2): 113-125.

[24] Tang F X, Wang X N, Yuan X, et al. MSADM: Large language model (LLM) assisted end-to-end network health management based on multi-scale semanticization[PP/OL]. V4.arXiv (2026-03-23)[2025-05-29]. https://doi.org/10.48550/arXiv.2406.08305.

[25] Yao S, Zhao J, Yu D, et al. ReAct: Synergizing reasoning and acting in language models[C]//The Eleventh International Conference on Learning Representations. 2023.

[26] Qin Y G, Tang J, Tang F X, et al. Multi-agent reinforcement learning in adversarial game environments: Personalized anti-interference strategies for heterogeneous UAV communication[J]. IEEE Transactions on Mobile Computing, 2025, 24(9): 8886-8898.

[27] Zhang L, Wang B L, Zhao Y Q, et al. Collaborative multimodal fusion network for multiagent perception[J]. IEEE

Transactions on Cybernetics, 2025, 55(1): 486-498.
[28] Peigné P, Kniejski M, Sondej F, et al. Multi-agent security tax: Trading off security and collaboration capabilities in multi-agent systems[J]. Proceedings of the AAAI Conference on Artificial Intelligence, 2025, 39(26): 27573-27581.
[29] Iodice F, De Momi E, Ajoudani A. Intelligent framework for human-robot collaboration: Dynamic ergonomics and adaptive decision-making[J]. Journal of Intelligent & Robotic Systems, 2026, 112: 5.
[30] Xu M R, Peng J L, Gupta B B, et al. Multiagent federated reinforcement learning for secure incentive mechanism in intelligent cyber-physical systems[J]. IEEE Internet of Things Journal, 2022, 9(22): 22095-22108.
[31] Pacheco F, Exposito E, Gineste M, et al. Towards the deployment of machine learning solutions in network traffic classification: A systematic survey[J]. IEEE Communications Surveys & Tutorials, 2019, 21(2): 1988-2014.
[32] Ren H S, Xu B X, Wang Y J, et al. Time-series anomaly detection service at microsoft[C]//Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining. New York: ACM, 2019: 3009-3017.
[33] Tariq S, Lee S, Shin Y, et al. Detecting anomalies in space using multivariate convolutional LSTM with mixtures of probabilistic PCA[C]//Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining. New York: ACM, 2019: 2123-2133.
[34] Yang Y Y, Zhang C L, Zhou T, et al. DCdetector: Dual attention contrastive representation learning for time series anomaly detection[C]//Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining. New York: ACM, 2023: 3033-3045.
[35] Tuli S, Casale G, Jennings N R. TranAD: Deep transformer networks for anomaly detection in multivariate time series data[PP/OL]. V6. arXiv (2022-05-14) [2025-05-29]. https://doi.org/10.48550/arXiv.2201.07284.
[36] Chen Y H, Zhang C Y, Ma M H, et al. ImDiffusion: Imputed diffusion models for multivariate time series anomaly detection[PP/OL]. V2. arXiv (2023-11-14) [2025-05-29]. https://doi.org/10.48550/arXiv.2307.00754.
[37] Ma M, Xu J M, Wang Y, et al. AutoMAP: Diagnose your microservice-based web applications automatically[C]//Proceedings of The Web Conference 2020. New York: ACM, 2020: 246-258.
[38] Li M J, Li Z Y, Yin K L, et al. Causal inference-based root cause analysis for online service systems with intervention recognition[C]//Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining. New York: ACM, 2022: 3230-3240.
[39] Wu Y K, Zhang J T, Hu N, et al. MLDT: Multi-level decomposition for complex long-horizon robotic task planning with open-source large language model[C]//Database Systems for Advanced Applications. Singapore: Springer, 2024: 251-267.
[40] Qiao S F, Fang R N, Qiu Z S, et al. Benchmarking agentic workflow generation[PP/OL]. V3.arXiv (2025-02-23)[2025-05-29]. https://doi.org/10.48550/arXiv.2410.07869.

作者简介

唐枫枭 男，1990 年 6 月出生于重庆市。现为中南大学计算机科学与工程学院正教授。主要研究方向为网络智能化、复杂系统数字孪生与故障诊断、网络智能运维、空天地一体化网络运维与优化等。
E-mail: tangfengxiao@csu.edu.cn

王啸楠 男，2000 年 6 月出生于河南省郑州市。现为中南大学计算机科学与工程学院博士研究生。主要研究方向为大语言模型、通信网络健康管理等。
E-mail: blackbug2021@163.com

张君健 男，1994 年 3 月出生于湖南省长沙市。现为中南大学计算机科学与工程学院博士研究生。主要研究方向为数字孪生网络、网络通信和网络故障诊断。
E-mail: 244701030@csu.edu.cn

赵 明 男，1975 年 11 月出生于湖南省益阳市。现为中南大学计算机科学与工程学院教授。主要研究方向为网络智能、机器学习、健康评估。
E-mail: meanzhao@csu.edu.cn