

大语言模型赋能的网络配置分析技术研究

李鹏飞¹, 刘宇靖¹, 苏金树^{1,2}, 喻波¹

(1. 国防科技大学计算机学院, 湖南 长沙 410073; 2. 军事科学院军事智能研究院, 北京 100091)

摘要: 针对现有网络配置分析工具智能化水平有限、操作复杂且依赖专业知识导致人力成本高的问题, 提出了一个大语言模型 (LLM) 赋能的网络配置分析技术框架。该框架融合 LLM 与形式化验证工具, 通过理解配置语义自动构建网络知识图谱, 并基于图谱推理补全以还原网络状态。进一步引导 LLM 调用领域知识, 实现配置错误检测、关键节点识别等任务的自动化分析。实验基于 EVE-NG 网络模拟实验室的官方配置、思科 CCNA 培训的配置案例及真实互联网拓扑配置, 从网络建模效果、应用效果验证、运行时间成本及鲁棒性等方面进行评估。结果表明, 该框架能够根据配置快速准确地还原网络状态, 适应多种网络场景的分析需求, 实现一键式分析, 为网络管理提供了重要技术支撑。

关键词: 网络管理; 网络配置; 大语言模型; 知识图谱

中图分类号: TP393

文献标志码: A

DOI: 10.11959/j.issn.1000-436x.2026014

Research on network configuration analysis technology empowered by large language models

Li Pengfei¹, Liu Yujing¹, Su Jinshu^{1,2}, Yu Bo¹

1. College of Computer Science and Technology, National University of Defense Technology, Changsha 410073, China

2. College of Military Intelligence, Academy of Military Science, Beijing 100091, China

Abstract: Due to the limited intelligence, complex operation and reliance on specialized expertise of the existing network configuration analysis tools, leading to significant human effort, a network configuration analysis framework empowered by large language model (LLM) was proposed. By integrating LLM with formal verification tools, the framework interpreted the semantics of network configurations to automatically construct a network knowledge graph, and performed reasoning and completion based on this graph to reconstruct the network state. It further guided the LLM to apply domain knowledge, enabling automated analysis for tasks such as network configuration error detection and key node identification. Experiments based on configuration files from the EVE-NG, Cisco CCNA training cases, and real-world Internet topologies were conducted from multiple perspectives, including network modeling accuracy, application effectiveness, runtime efficiency, and robustness. Results indicate that the proposed framework can quickly and accurately restore the network state based on configurations, adapt to the analysis requirements of diverse network scenarios, and achieve one-click analysis, thus providing important technical support for network management.

Keywords: network management, network configuration, large language model, knowledge graph

收稿日期: 2025-12-09; 修回日期: 2026-01-13

通信作者: 刘宇靖, liuyujing@nudt.edu.cn

基金项目: 国家自然科学基金资助项目 (No.62372462); 国家重点研发计划基金资助项目 (No.2022YFB3104800); “芙蓉计划”青年英才基金资助项目 (No.2024RC3136)

Foundation Items: The National Natural Science Foundation of China (No.62372462), The National Key Research and Development Program of China (No.2022YFB3104800), The Science and Technology Innovation Program of Hunan Province (No.2024RC3136)

0 引言

随着信息技术持续发展,大量网络设备接入组网,使网络配置管理成为实现多样化网络功能所需面对的关键问题之一。配置文件的语句具有显著的语义性和层次性,涵盖接口、路由协议、网络服务等多种要素的配置信息,能够据此推断网络拓扑、网络连通状况以及网络服务等网络状态。在网络设备的配置与实际部署过程中,网络管理员依托专业经验对网络运行进行调控,以满足实际业务需求。文献[1]指出,网络运营商的网络管理员需要维护大规模网络,实现多样化功能并确保快速部署,其操作往往基于经验与直觉。网络管理员对大规模网络进行配置管理,并在其中开展配置检查和信息推理的任务极具挑战性。文献[2]的调研表明,网络运营商通常借助通用配置模板与自动化脚本进行配置调整,主要关注管理目标的实现,而较少评估配置变更可能引发的潜在影响。然而,网络配置存在脆弱性,损害网络性能的配置失误在实践中较为常见^[3],严重时甚至可能引发网络故障。例如,2021年Facebook因配置故障导致其网络服务中断近6h^[4]。因此,如何提供高效的配置管理与分析方法,缓解网络配置脆弱性,已成为学术界与产业界广泛关注的问题。

为应对复杂网络场景下大量网络设备的配置管理需求,网络配置的智能化分析日益受到重视。网络中设备的相互连接形成了网络拓扑,启用的路由协议支撑了网络连通,网络服务构成了整体的网络功能,这些网络状态通过网络设备的配置文件实现。通过还原配置对应的网络状态并验证其是否符合预期目标,能够及早排除配置故障,降低配置脆弱性风险。然而,从设备配置中推理网络状态是一个从低层次文字语义到高层次服务功能的抽象过程,依赖人工分析,耗时多且易出错,成为网络配置管理的核心难点。

网络验证是缓解配置脆弱性的一类关键技术。已有研究通过不同方法对网络控制和数据平面进行建模,以验证网络配置的预期意图。文献[3]提出的Batfish将网络配置迁移至高级编程框架,采用声明性方法将网络配置建模为网络控制与数据平面。文献[5]提出的Minesweeper将网络配置转化为逻辑公式,并结合描述预期特性的约束条件,通过约束求解判断当前网络状态。文献[6]提出的Tiramisu是一种通用的多层次网络验证框架,基于图对

网络控制平面进行建模。文献[7]利用图进行配置检查,从网络配置中提取组件及其相互引用关系以构建图的节点,通过比较图中环路的频率作为配置检查的判定标准。文献[8]提出了一种基于差分网络分析的方法,通过网络控制平面的变化,增量计算控制和数据平面的状态,并输出端到端的行为差异。文献[9]提出的ConfigKG基于配置文件将路由网络建模于知识图谱,将配置经验转化为规则集,从而把配置错误检测问题转化为图上的规则检查问题。文献[10]提出的XNV在知识图谱中构建语法树,采用可解释的网络模型对网络控制平面进行建模,并提供策略违规的根源定位、语义解释和修复建议。尽管网络验证领域已存在众多成熟技术,但现有方法仍存在以下不足:一是智能化水平有限。现有方法多为针对特定问题的点对点工具,面对复杂场景的分析需求,网络管理员需综合使用多种方法进行分析,缺乏全局视角的一键式分析能力;二是通用性与整体性不足。多数研究集中于连通性、路由策略等局部功能的验证,未能构建对网络全局服务能力的整体分析框架,难以适应多样化场景的综合需求;三是可维护性较弱,多数工具仅提供一次性分析结果,缺乏持续维护配置状态的存储机制与交互接口,不利于长期部署使用。针对上述不足,亟须一种智能化的网络配置分析方案以适应多种网络环境的分析需求,降低网络管理员的使用成本与维护负担。因此,大语言模型(large language model, LLM)赋能的网络配置分析技术研究具有重要的现实意义。

本文工作的主要贡献如下。

1) 提出了一种LLM赋能的网络配置分析技术框架,实现全局视角的一键式配置分析。该技术利用LLM从配置文件中提取实体与关系,自动构建并补充知识图谱,从而实现对网络状态的精准建模。知识图谱作为外部结构化知识库,增强了LLM对配置文件语义的深度理解,提高了配置分析的准确性。通过LLM与知识图谱的协同互动,该技术实现了配置文件的自动化解析与网络状态的高效推理,能够适应多样化的分析需求和网络场景,提供了一键式智能配置分析新思路。

2) 设计实现了面向具体配置问题的LLM引导交互机制,提升了在多样化场景下的整体适应能力。针对网络配置错误检测、关键节点识别等典型

任务，构建了相应的特征描述和推理规则，结合形式化验证工具组件，引导 LLM 调用相关专业进行分析推理和输出。该技术利用 LLM 对自然语言需求的理解能力，能够灵活适配不同网络场景的配置分析任务。

3) 基于多源配置数据开展系统实验验证。使用 EVE-NG 网络模拟实验室的官方配置文件^[11]、思科公司 CCNA 培训的配置案例^[12]以及主干网真实拓扑配置^[13-14]进行测试。实验结果表明，该技术能够准确推理网络状态，根据需求提供详细的分析报告，在合理的时间开销内完成自动化分析。技术支持构建可持续管理与维护的特色数据库，同时提供基于自然语言交互的人机交互平台，全面验证了该技术在网络配置管理中的应用价值。

1 技术框架概览

针对特定网络内配置文件的分析任务，需要从配置文件的逐个解析、网络状态的推理还原和需求引导的智能分析 3 个主要流程进行，如图 1 所示。首先，在预设的语义框架引导下，LLM 从配置文件中抽取统一的“实体-关系-实体”三元组结构数据，以构建知识图谱。随后，基于专业领域知识，LLM 对知识图谱进行分析补全，实现对网络状态的推理还原。最后，结合分析需求与引导，LLM 基于知识图谱获取配置所反映的网络状态，理解网络管理员的具体分析需求，并提供详尽的分析报告。

配置文件的逐个解析核心挑战在于将不同厂商的配置文件语义内涵建模为统一的数据结构形式。在实现相同的网络功能时，由于不同厂商的配置语法不同，配置的表达形式可能存在较大差异。为还原准确的网络状态，本文采用知识图谱作为层次化建模介质，其以节点和边直观表征网络实体与关系，天然契合配置的语义表达^[15]。并且文献^[16]展示了知识图谱在大规模网络配置分析中的应用能力，使用公开数据集构建互联网知识图谱，将复杂的分析问题转化为查询问题。这些研究说明了知识图谱非常契合网络配置的分析任务，具有对大规模配置文件建模的潜力。此外，LLM 在文本理解方面表现优异^[17]，能够高效处理格式化和结构化的文本数据，与配置文件的逐个解析任务高度契合。具体实现上，本文在构建该统一模型时，首先引入基于 Batfish 开发的语法解析器对原始配置进行预处理与标准化，将其转化为规范化的中间表示，以加速解析过程并消除语法差异。在此基础上，利用 Batfish 解析原始配置形成的基本数据结构框架，构建以知识图谱为目标的语义引导框架，驱动 LLM 精准抽取规范化数据并映射为三元组数据。流程以单个文件为单位进行解析，最终将全部三元组数据整合导入知识图谱，为后续推理奠定基础。

网络状态的推理还原核心挑战在于利用网络知识推理配置语义中隐含的关联关系。基于知识图谱的推理能够从现有信息中挖掘潜在信息。文献^[9]依托网络管理经验，在知识图谱中推理隐含的网络状态，并通过图规则检查识别配置中的安全隐患。然而，该过程高度依赖专家制定的规则集，难以适应多样化的网络场景，因此亟须更加灵活的智能化方案。随着人工智能的发展，LLM 在应用中展现了丰富的背景知识，能够替代人类专家实现自动化分析^[18-21]。在具体实现上，本文采用规则与启发式推理相结合的还原路径。首先，通过预先实现的确定性规则组件，严格依据网络协议逻辑，自底向上（从物理接口、设备拓扑到路由区域）自动推导基本网络状态。在此基础上，针对域名系统（domain name system, DNS）、虚拟专用网络（virtual private network, VPN）等复杂网络服务，引导 LLM 进行启发式分析，识别其连通性与配置策略依赖，并在知识图谱中构建 DEPENDS_ON、SERVES 等高阶语义关系。整个流程在形式化验证工具提供的

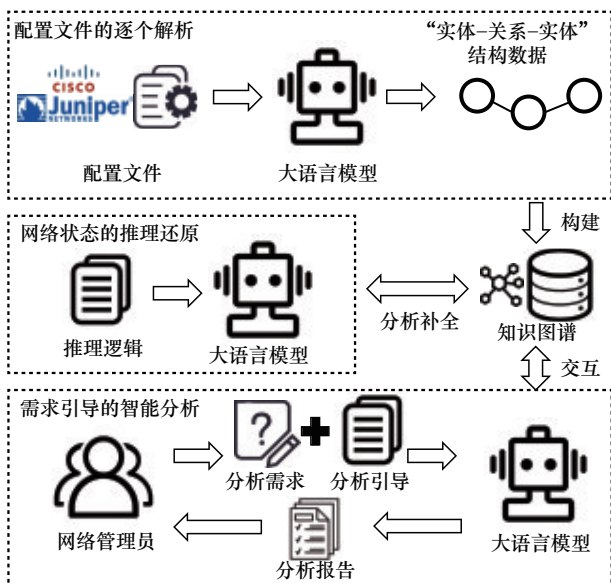


图 1 LLM 赋能的网络配置分析技术框架

事实基准与规则组件的逻辑约束下进行, 确保还原结果准确可靠, 同时充分发挥 LLM 在语义关联与灵活推理方面的优势。

需求引导的智能分析核心挑战在于引导 LLM 理解配置情况并提供准确的分析报告。文献[22-23]指出, 整合显性知识(如知识图谱)能显著增强 LLM 的认知与推理性能。基于此, 本文将前序流程所构建的、完整刻画网络状态的知识图谱作为核心事实库, 为 LLM 提供精准的语义上下文。在实际分析过程中, LLM 严格按照给定的思维链进行分析, 并主动调用框架中集成的多个形式化验证工具组件来获取佐证或执行验证, 从而确保分析过程的严谨性与结论的可信度。例如, 模型可调用基于 Batfish 的形式化验证组件模拟数据包传播路径, 在网络连通性事实的基础上进一步验证配置策略。通过将 LLM 强大的语义理解能力与这些形式化验证工具的验证能力相结合, 本文实现了分析的自动化与智能化。同时, 该框架具备高度可扩展性, 通过引入新的工具组件或定义新的提示词引导, 能够灵活适配多种定制化分析场景。本文以网络配置错误检测与关键节点识别两个典型网络运维任务为例, 展示了该引导分析范式的有效性。

LLM 与知识图谱的双向赋能是技术框架的智能化特点。在 LLM 与知识图谱的结合中^[24], LLM 能够从头构建并推理补全知识图谱, 知识图谱能够作为外部知识库进一步增强 LLM 的推理分析能力。本文提出的 LLM 与知识图谱的结合框架如图 2 所示, 该框架结合了检索增强生成(retrieval-augmented generation, RAG)模型^[25]与推理和行动(reasoning and acting, ReAct)模式^[26], 汲取二者优势, 能够在多次获取外部网络配置信息的同时, 通过多轮迭代分析确保结果的准确性。在推理过程中, LLM 根据分析需求持续调用工具组件, 自由地从知识图谱中提取所需信息进行分析, 并在阶段性分析基础上不断丰富和完善知识图谱。知识图谱则提供网络状态的具体信息, 满足 LLM 在分析过程中对网络配置及状态的知识需求。这些工具组件主要负责在图数据库中执行以下常见操作: 1) 实体创建, 创建所需实体, 并为其分配相应标签和属性; 2) 关系创建, 基于指定实体 ID, 建立实体间关系并赋予相应类型; 3) 实体删除, 根据实体 ID 删除指定实体; 4) 关系删除, 根据

关系 ID 删除指定关系; 5) 查询信息, 通过自行构建的查询语句获取所需网络信息; 6) 自定义组件, 为加速分析过程, 提供自定义查询工具的接口。例如, 提供预构建的查询语句, 快速识别配置中潜在的连接关系接口, 降低可能因 LLM 构建查询语句错误或多轮信息获取导致的分析成本。此外, 本文在自定义组件中提供了一系列形式化验证工具, 辅助 LLM 准确完成各项任务。例如, 在配置文件解析中, Batfish 查询组件用于查询 Batfish 数据结构语义的官方说明文档, 从而获取准确的配置语义定义与约束条件; 在网络状态还原中, 规则推理组件用于快速提供配置中强逻辑的关联关系, 提升推理效率; 在实际场景分析中, 基于 Batfish 的形式化验证组件能够模拟数据包传播路径, 为连通性推断提供数据参考。该结合框架应用于网络状态的推理还原与需求引导的智能分析的流程中, 支撑了 LLM 与知识图谱的有效交互, 保证了分析的准确性。

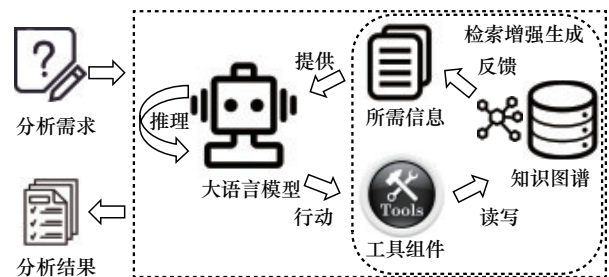


图2 LLM与知识图谱的结合框架

2 配置文件的逐个解析

配置的语义决定了网络的具体功能。在配置文件的逐个解析流程中, 本文以成熟的网络验证工具 Batfish 所定义的规范化语义框架为基础进行引导。该框架源于长期和广泛的工程实践, 其结构语义已能完整涵盖主流配置场景, 为知识图谱的实体与关系设计提供了权威且稳定的模式基准。在此框架下, 本文引导 LLM 根据配置语句的具体语义, 将经过 Batfish 预处理和标准化的配置数据, 精准映射并抽取为知识图谱的实体与关系。该流程赋予一定的自由度, 但并非让 LLM 无约束地创建全新语义, 而是在该语义集合内进行识别、匹配与实例化。例如, 对非标准化的配置语法, 模型可依据 Batfish 的语义逻辑, 将其映射至基准框架内最适配的实体类型, 或按既定规则扩展同质化的实体属性

与关系，从而在严控语义一致性的前提下，更灵活且全面地完成配置建模。最终，该框架将对结果进行校验检查，以保证语义抽取的可靠性。整体流程旨在构建精准且规范的初步知识图谱，涵盖对各个配置文件的标准化表征。

2.1 配置文件的建模

配置文件具有语法多样性和内容层次性强的特点。不同厂商的路由设备配置语法存在差异，即使实现功能相同，其配置语句也会存在不同。通过知识图谱语义网络的统一描述，可以消除配置文件的语法差异，增强 LLM 在推理分析阶段对配置语义内涵的理解能力。本文在构建统一语义描述时，参考了成熟网络形式化验证工具 **Batfish** 所定义的数据结构，其官方文档中有核心数据结构的组成与示例，简要分类的部分示例如表 1 所示。**Batfish** 的结构设计完备，但其属性定义更侧重于服务程序化分析与验证，定义数量较多，内涵复杂，各属性间还存在隐含关联。直接用于构建知识图谱时，存在语义细节复杂、信息冗余较多等问题。为此，本文对其进行了有针对性的梳理与语义抽象，配置文件在知识图谱中建模的本体结构如图 3 所示，相关实体与关系分别如表 2 和表 3 所示。对于知识图谱中的实体关系，本文采用英文命名，以尽量保持配置原有语义，避免翻译过程中可能带来的信息误差。

表 1 Batfish 定义的数据结构部分示例

类别	结构名称	属性数量
设备层面	Node Properties	37
	Interface Properties	37
路由协议层面	BGP Process Configuration	11
	OSPF Process Configuration	8
	⋮	⋮
网络服务层面	VLAN Properties	4
	VRRP Properties	7
	⋮	⋮
其他	Named Structures	4
	Defined Structures	3
	Undefined References	5
	⋮	⋮

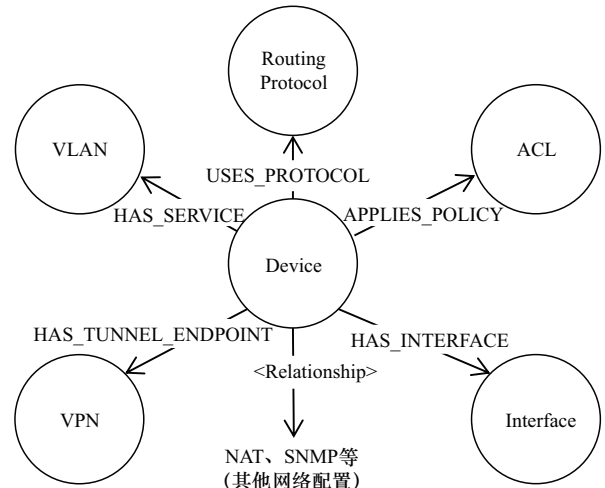


图 3 配置文件建模的知识图谱本体结构

表 2 配置文件建模的知识图谱实体

实体	属性	语义
Device	config_file; router_type	配置文件对应的路由设备
Routing Protocol	type; networks	配置文件中具有的路由协议
ACL	name; rules	配置文件中确立的访问控制列表
Interface	name; ip_address; status; subnet_mask	配置文件中涉及的接口
VPN	type; peer_address; encryption_algorithm	配置文件中开启的 VPN 服务
VLAN	vlan_id	配置文件中开启的 VLAN 服务
⋮	⋮	配置文件中的更多成分

表 3 配置文件建模的知识图谱关系

关系	语义
USES_PROTOCOL	设备使用的路由协议
APPLIES_POLICY	设备开启的路由策略
HAS_INTERFACE	设备具备的接口
HAS_TUNNEL_ENDPOINT	设备具有的 VPN 端点
HAS_SERVICE	设备拥有的网络服务（如 VLAN 等）
⋮	设备启用的其他网络配置

对配置文件的建模主要依赖 LLM 进行实体与关系的抽取。每个实体具备一些属性，以进一步描述配置内容。表 2 中标注的属性用于表示该配置成分的基本语义，额外所需的属性则由 LLM 决定并抽取。在不同的网络环境中，配置所涉及的功能存

在差异,需要多样化的实体以丰富语义网络,从而更好地描述配置的内涵。例如,在边界网关协议(border gateway protocol, BGP)主干网中,可能会启用二层虚拟专用网(layer 2 virtual private network, L2VPN)、多协议标签交换(multi-protocol label switching, MPLS)等服务配置。在企业主干网中,可能存在安全套接层虚拟专用网(secure socket layer VPN, SSL VPN)等服务配置。这些多样且复杂的服务语义,在底层配置中通常有具体的实现模式。例如,在 Batfish 的解析框架中, L2VPN 的语义信息常被规范地映射并定义于 BGP Process Configuration 的 VRF 等相关属性中。这些例子表明,对 LLM 而言,配置的解析与语义映射需要足够灵活以覆盖多样的实体关系,同时又需保持底层语义的准确性。因此,需要 LLM 对实体与关系进行启发式扩展。LLM 并非完全自由创造,本文要求其在 Batfish 所厘清与规范的语义框架约束下进行。具体而言,LLM 首先依据 Batfish 解析后产生的结构化语义中间表示进行识别与匹配。当遇到未在基准表中明确定义的新型或复合配置模式时,则参考 Batfish 已定义的数据结构属性关联逻辑,启发式地推导并实例化新的实体关系,从而在保证配置语义一致性的前提下,增强知识图谱对复杂配置的描述能力。例如,对于 BGP 的 L2VPN 配置,LLM 将创建 BGP Process Configuration 实体及其相应属性,并将其关联至对应的 Device 节点。

2.2 语义框架引导的实体关系提取

配置文件解析的过程本质上是将配置语义提炼并转化为对应实体与关系的过程。经过大量语料的训练,LLM 已具备理解网络配置语义的能力。因此,实现解析目标的关键在于设计用于语义框架引导的提示词,通过提示词引导 LLM 检索并调用相关知识,从而完成配置文件的建模任务。

本文设计的解析配置文件的提示词如图 4 所示。为便于阅读,原文英文的提示词均已翻译为中文。在提示词中,输出格式指引明确要求 LLM 输出“实体-关系-实体”三元组数据,并提供了表 2 和表 3 所示的实体与关系作为示例。此外,提示词也说明了模型可调用 Batfish 查询组件:当需要识别或扩展实体关系时,模型可据此获取 Batfish 中已定义的结构体名称和语义规范,从而在统一的语义框架内补充和映射实体关系。

```

【System】你是路由配置文件的专家。你的工作是将提供的配置文件转换为
“实体-关系-实体”三元组。这些三元组将用于构建知识图谱。用“json”标签
包装输出: {输出格式指引}
可用的工具组件:
* Batfish 查询组件: 当你需要确认或扩展现有实体、关系的规范名称时,请
调用此工具获取 Batfish 中已定义的结构体名称与语义规范。
【用户提示词】
##您必须遵守以下规定:
* 请确保每个实体的 ID 必须是唯一的,因为这会影响到知识图谱中实体和
关系的构建。
* 您可以根据需要为实体和关系分配属性,以进一步阐明其含义。请尽量确
保这些属性的名称对于相同的实体或关系保持一致。
* 为配置文件分配一个标有“设备”的实体,并与与之相关的实体建立关系。
* 关注配置的实际语义,例如接口配置、启用的服务、启用的路由协议、对
应配置后的注释等等。
* 请忽略无用的内容,例如配置文件中的标语,它对路由网络的整体运行没
有影响。

## 定义的实体与关系样例:
-- 实体:
{实体列表}
-- 关系:
{关系列表}
## 遵守以下的指导:
1. 实体识别:
- 优先使用定义的实体标签和属性名称
- 将特定于供应商的术语转换为统一的规范名称
(样例: "ge-0/0/0" 对应 Interface, "IP-SEC" 对应 IPsecVPN)
- 如果还有未定义的实体,可以自行添加新的实体标签和属性
2. 属性处理:
- IP格式规范化: 192.168.1.1/24→CIDR表示法
- 在适用的情况下将数值保留为整数
- 用于表示配置中具有意义的字段
3. 关系建立:
- 根据给出的关系语义建立实体间关系
- 如果配置设计了未定义的关系,则可以参考给出的关系样例自行创建,但需
要保证其名字具有语义
##现在根据配置文件构建三元组:
- 配置文件名: {本地配置文件名}
- 配置文件内容: {配置文件的具体内容}

- 结果:

```

图 4 解析配置文件的提示词

为控制输入长度并提升解析效率,本文基于 Batfish 构建了语法解析器。配置文件在输入模型之前,先经该语法解析器进行预处理与语义标准化,滤除冗余信息,提取结构化的语义表示。随后,将配置文件名称、预处理后的结构化内容及对应的提示词一并输入 LLM,启动解析过程。在语义框架的约束与引导下,LLM 抽取的实体关系及属性保持较高的稳定性。

2.3 结果校验检查

为确保 LLM 输出结果的稳定性与可靠性,本文设计了相应的结果校验与迭代机制,用于对 LLM 解析生成的内容进行自我检查与修正。LLM 在生成新增内容时可能存在一定的不稳定性,具体表现为可能引入未经充分验证的实体或关系。在本文框架中,实体通常对应具有实际网络功能的配置对象,其定义应具备明确的语义基础。因此,校验程序重点关注新生成实体的有效性。

在输出最终解析结果前,系统会启动语义一致性校验程序。该程序将逐一核对解析结果中出

现的每个实体是否已存在于预定义的实体列表中，或通过 Batfish 查询组件确认其是否符合 Batfish 文档中已定义的结构体名称的语义规范。若发现无法匹配的实体，则认为该部分结果的可靠性不足，程序会自动将未通过校验的内容反馈给 LLM，并要求其重新生成。此过程可迭代进行，直至所有内容通过校验。最终，所有配置文件解析无误后，对应的三元组数据将在检查冗余内容后被合并，并统一导入知识图谱，从而完成配置文件的整体建模。

3 网络状态的推理还原

网络状态反映了配置所体现的网络功能情况，通常涵盖多个层面，如接口连接、网络拓扑、路由通信等。在网络状态的推理还原流程中，本文设计了表征网络状态所需的实体关系，并引导 LLM 对配置文件建模的知识图谱进行推理补全。在该流程中，LLM 主要从物理层面、设备层面和区域层面对网络状态进行基于规则的层次化推理还原，经过推理结果校验后，进一步在服务层面开展基于网络状态的启发式推理。

3.1 网络状态的建模

配置文件的具体语义能够有效反映网络状态。设备通过配置实现网络中的互联，如配置的路由协议、VPN 等网络服务，从而支撑整体网络功能。因此，配置文件之间存在显著的关联性，能够揭示整体的网络状态。知识图谱的建模将配置文件的语义转化为实体关系，但每个配置文件相对独立，需要进一步推理分析以建立配置文件之间的关联。此外，配置文件之间的关联同样反映了网络路由系统的层次性：在物理层面，通过接口配置推理相连的物理接口；在设备层面，通过设备连接关系反映网络连接的拓扑结构；在区域层面，通过路由协议等配置划分网络功能的服务范围。基于这种层次性，对知识图谱中需要创建新的实体关系进行表征，从而建模配置所反映的网络状态。网络状态建模的知识图谱本体结构如图 5 所示，其相应的新实体和关系分别如表 4 和表 5 所示。在需要引入新实体关系时，采用的方法与建模配置文件的过程类似，LLM 在 Batfish 的语义约束引导下进行补充。

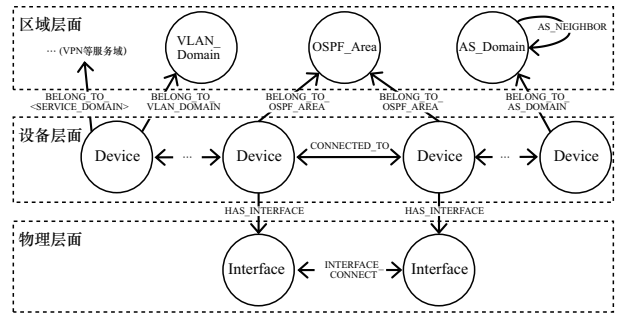


图 5 网络状态建模的知识图谱本体结构

表 4 网络状态建模的知识图谱实体

实体	属性	语义
VLAN_Domain	vlan_id	网络中的 VLAN 区域
OSPF_Area	area_id	网络中的 OSPF 域
AS_Domain	as_number	网络中的自治系统
⋮	⋮	其他描述网络状态的实体

表 5 网络状态建模的知识图谱关系

关系	语义
INTERFACE_CONNECT	设备接口的物理连接
CONNECTED_TO	设备间的网络连接
BELONG_TO_VLAN_DOMAIN	设备归属的 VLAN 域
BELONG_TO_OSPF_AREA	设备归属的 OSPF 域
BELONG_TO_AS_DOMAIN	设备归属的自治系统
AS_NEIGHBOR	自治系统间的连接
⋮	其他描述网络状态的关系

网络状态的建模主要体现在网络各层次上的实体关系推理与构建。本文将该过程分为两个阶段：基于规则的层次化推理和基于网络状态的启发式推理。由于不同网络环境下的实际需求存在差异，所配置启用的网络服务也各不相同，网络状态也呈现多样性。为描述这些状态，本文定义了基本网络状态，包括物理接口连接状态、设备连接状态和路由区域连接状态。这些网络状态反映了网络是否具备最基本的通信功能，是绝大多数高级网络服务的基础。此阶段的推理（详见 3.2 节）可通过预先实现的工具组件，以确定性规则匹配的方式高效、准确地完成。通过层次化推理，能够获取网络的基本状态。随后，对推理结果进行校验，以确保其准确性。

在此基础上，针对多样化且复杂的网络服务功能状态还原，通常缺乏唯一且确定的规则。这些服务的有效性与依赖性需要结合网络上下文与领域知识进行推断。因此，在第二阶段，LLM 基于第一阶段构建的基础状态，采用启发式推理方法挖掘新的

实体关系,以涵盖多种复杂的网络服务状态。网络服务状态的定义与推理参考基础网络状态的推理结果,并深度融合LLM自身的专业知识,从而在保证基础准确性的前提下,拓展知识图谱的语义覆盖广度和深度。

3.2 基于规则的层次化推理

基本网络状态从原始的配置文件建模进行推理,形成各层面的连接关系,推断各层面存在的相互关联。根据物理层面的接口连接情况,能够推断设备层面潜在的连接;根据设备层面的连接情况,能够进一步推理区域层面网络服务的范围与状态。此外,层面间的关联也能够辅助判断各个层面的连接是否正确。例如,在不同自治系统中,可能存在设备具有相同IP配置的情况,这可能导致对设备和物理层面连接的错误推理,而该问题可通过设备所属区域信息加以避免。这些基本网络状态之间通常存在严密的逻辑关联,能够通过严谨的规则进行推导。为此,本文为LLM提供了明确的推理规则,并配备相应的规则验证组件,辅助LLM完成分析,从而系统性地构建可靠的基本网络状态。

1) 物理层面的推理。根据接口配置的IP地址及其子网掩码,可以判断接口是否处于同一子网内。当接口处于同一子网内时,可在它们之间建立INTERFACE_CONNECT关系。

2) 设备层面的推理。基于物理层面建立的INTERFACE_CONNECT关系,通过接口的HAS_INTERFACE关系查询对应的Device实体,并在这些Device实体之间建立CONNECTED_TO关系。

3) 区域层面的推理。根据Device实体所关联的网络服务情况,按照自治系统归属、开放最短路径优先(open shortest path first, OSPF)协议域归属的顺序进行推理。如果Device实体启用了BGP,则创建AS_Domain实体以代表自治系统,并在属性中标注自治系统号,同时建立设备与之对应的归属关系。在自治系统内的OSPF域,以相同方式创建相应的实体与关系,依此类推至其他区域。完成新的实体及其归属关系的构建后,需进一步推理设备共同所属的区域。依据设备层面的连接情况,若相互连接的设备归属于同一区域,则应合并其对应的区域实体。例如,Device实体A与B具有CONNECTED_TO关系,且二者位于同一自治系统下相同的OSPF域,则应合并其归属的OSPF_Area实

体。自治系统则较为特殊,其域内并非所有路由设备都启用BGP,因此需要判断区域中的边界路由设备以确认范围,通常这些边界路由设备为eBGP对等体。此外,为了表示更高层次的网络拓扑结构,依据eBGP对等体的配置,AS_Domain间建立相互的AS_NEIGHBOR关系,以表明自治系统间的关联。

规则验证组件嵌入在如图2所示的自定义工具组件中。对于物理与设备层面所需的同一子网判断,本文采用编程实现的方式完成,LLM只需查询并提供相应参数即可使用。在区域层面推理时,本文提供了集成最短路径算法的工具组件,能够在知识图谱中快速查找两个节点之间的最短路径,从而支持LLM寻找最近的BGP边界路由器,提升了规则推理的效率。

LLM的推理过程严格按照物理、设备和区域逐步进行。为引导LLM的推理,每一步所用的通用提示词如图6所示。该提示词开头总结了当前知识图谱的建模情况,通常为已完成配置文件建模的知识图谱。并且,提示词提供了图2中工具组件的描述,LLM能够直接调用这些工具组件。提供的实体与关系列表与表4和表5中的内容一致,同时赋予LLM一定的自由度,在语义框架的约束下智能地创建新的实体关系以表征多种网络服务状态。推理逻辑遵循给定的规则进行,并对推理结果进行校验。为进一步稳定LLM的性能,提示词中还包含若干注意事项,防止模型一次性获取过多数据而超出上下文限制,同时避免LLM创建冗余实体,确保知识图谱的精炼与准确。

```

[System] 你是路由网络领域的专家,负责帮助推理基本的网络情况。目前你可以依赖的知识图谱数据库的基本情况为: {当前的知识图谱本体结构}
为你提供用的工具组件为:
{工具组件}
* Batfish 查询组件: 当你需要确认或扩展现有实体、关系的规范名称时,请调用此工具获取 Batfish 中已定义的结构体名称与语义规范。

【用户提示词】
##如果知识图谱已经足够完备,那么你就无需再创建新的实体和关系了。
##以下是一些供您补充知识图谱的参考说明:
1. 实体定义标准: {实体列表}
2. 关系定义标准: {关系列表}
你可以根据需要定义新的实体关系,只要它们能够更加清晰地表达网络状态。请参照定义的实体关系进行命名。
## 推理应遵守的逻辑:
{推理各个层面的引导规则}
在进行推理时,请灵活运用提供的工具组件提高效率。
示例:
{一些推理示例}
##注意:
在进行分析时,请你依照上述给出的逻辑来进行推理。此外,如有需要,您还可以获取额外的信息以提高您的推理的准确性程度。
请避免直接获取知识图谱中的所有节点信息(例如MATCH(n)、RETURN(n))。当数据量较大时,您将难以处理!
请不要返回任何问题!
请自行在知识图谱中实现构建!
如果知识图谱中已有实体,请不要创建重复的实体!请使用现有的实体或者在删除后再创建!

```

图6 网络状态推理还原的通用提示词

3.3 基本网络状态的推理结果校验

推理结果校验旨在确保 LLM 还原基本网络状态的准确性，从而支撑服务层面的启发式推理。由于初始物理与设备层面的推理可能建立错误关系，因此需在各层面对推理结果进行回溯检查。与推理过程类似，结果校验同样能够通过严谨的规则进行。

1) 若存在归属不同 AS_Domain 的设备间具有 CONNECTED_TO 关系，则其应为 eBGP 对等体。

2) 若具有 VPN 等复杂网络服务的接口间具有 INTERFACE_CONNECT 关系，则配置中应包含对应的网络服务及标识，如 VRF 实例命名需保持一致。

3) 若具有访问控制列表 (access control list, ACL) 等访问策略的设备间存在 CONNECTED_TO 关系，则其对应接口的 IP 连接不能违反 ACL 路由策略。

当违反上述任何一条规则时，其对应关系将被移除。为提升规则校验结果的准确性，本文基于 Batfish 构建了形式化验证组件，并将其集成至校验流程中。该组件能够直接调用 Batfish 提供的网络数据平面形式化模型，对特定网络行为进行模拟分析。如图 7 所示，该组件可模拟执行类似 “Traceroute” 的端到端路径探测，精确返回数据包在给定策略下的实际传播路径、经过的设备及其被允许或拒绝的状态等关键信息。随后，模型将使用与推理阶段一致的知识图谱查询工具，检索相关实体与关系，自动比对自身的推理结论与形式化验证结果。若存在冲突，则触发针对性的重新推理或生成修订提示，从而在语义层面确保网络状态建模的最终准确性。

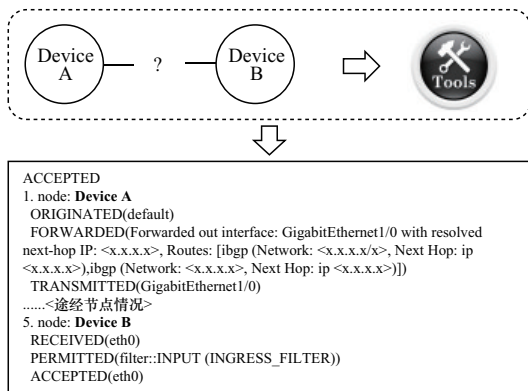


图 7 形式化验证组件提供的数据平面输出示例

3.4 基于网络状态的启发式推理

与基础网络状态的推理不同，服务状态的判断难以通过有限的确定性规则进行泛化，其复杂性主要体现在依赖的多维性与语义的隐含性上。一项服务的正常运行不仅依赖于 IP 可达性，还可能依赖访问控制策略、服务配置和物理端口的开放状态。这种依赖关系是跨层次和多因素的，需要结合具体配置情况进行判断，并且配置文件中不会出现类似 “Service_A 依赖于 Routing_B 和 Policy_C” 的显式声明，服务之间以及服务与底层网络状态之间的依赖和制约关系隐含于整体配置语义中，需结合网络上下文进行推断。因此，本文在此阶段引入网络服务的启发式推理机制，旨在解决规则无法覆盖的复杂语义关联问题。其核心任务是基于知识图谱中已建立的基础网络状态，结合配置的全局上下文，推导出高级网络服务的运行状态与依赖关系。

1) 服务依赖的识别与分析。首先引导 LLM 以设备为基本单元，识别配置中明确定义的服务实体，并分析其正常运行所必须满足的潜在条件。本文将 LLM 需要推断的潜在条件规范为两大类：连通性与配置策略依赖。

连通性依赖指服务对底层网络路由可达性的要求，它决定了服务能够辐射的设备范围。此类依赖可直接通过 3.2 节推理出的基础网络状态进行验证，从而锁定服务可覆盖的潜在设备子集。

配置策略依赖指在该设备子集中，服务相关配置的策略要求。例如，DNS 服务的可用性依赖于相关 ACL 策略是否允许放行 DNS 协议（如协议默认使用的 53 端口）。对此类依赖的分析，需要模型具备特定网络服务的专业知识，以理解其通信流程与协议要求。尽管基于 Batfish 的形式化验证组件可以校验配置的具体行为（如在数据平面中验证某协议的连通性），但正确调用该组件并构建有意义的验证查询本身则依赖于 LLM 对服务语义的理解。

2) 语义关系的构建。基于上述分析结果，LLM 在知识图谱中创建新的语义关系，以精确描述服务状态，主要构建了以下两类关系。

DEPENDS_ON 关系：用于编码服务对其关键前提条件的依赖。例如，若某个 DNS 服务的有效性依赖于一组设备之间特定路由由协议的连通性，则在知识图谱中建立该 DNS 实体与对应 Routing Pro-

tocol 实体间的 DEPENDS_ON 关系。

SERVES 关系：用于界定服务的有效覆盖范围，即为服务实体与经验证满足其所有访问条件（即连通性与配置策略依赖）的网络设备实体之间建立 SERVES 关系，从而明确标识服务的提供对象。

总之，通过引入 LLM，本文充分发挥了其理解复杂网络服务语义的能力，实现了对多样化服务的灵活、启发式分析。上述流程不仅提升了推理的适应性，而且最终构建出了更高质量和语义丰富的网络状态知识图谱，为后续需求引导的智能分析提供了坚实支撑。

4 需求引导的智能分析

需求引导分析是 LLM 赋能的网络配置分析技术的智能化特点。在需求引导的智能分析流程中，本文构建了智能问答平台，支持网络管理员通过自然语言动态地对知识图谱进行个性化更改。该流程的分析基础是前序章节所构建的、能够精准描述网络状态的知识图谱。该图谱融合了形式化工具的验证结果与启发式推理的语义关联，具备高度的准确性与完备性。基于此，针对网络配置的具体分析任务可以转化为 LLM 对知识图谱进行自主查询、理解与深度推理的过程。本文结合 RAG 框架^[25]、ReAct 框架^[26]和精心的提示词工程，灵活引导 LLM 的分析思维链。该方法能够有效激发模型检索相关领域知识，并使其与知识图谱中的结构化事实相结合，从而完成有针对性的复杂分析。在该流程中，本文以网络配置错误检测和网络关键节点识别为范例，引导 LLM 按照合理逻辑并结合自身专业知识进行深入分析，最终输出供网络管理员参考的具体报告。

4.1 基于 LLM 智能体的问答平台

问答平台作为本文网络配置分析技术框架的人机交互平台，利用 LLM 的能力，使网络管理员能够通过自然语言进行查询，从而获得直观的配置分析结果，了解配置所反映的网络状态。该问答平台基于图2所示的流程构建，LLM 为智能体 (Agent)，在理解网络管理员的分析需求后，经过多轮的知识获取与推理分析，最终得出分析结果。在自由使用与知识图谱交互的形式化工具组件的基础上，LLM 能够根据网络管理员的实际需求构建额外的

实体关系。通过这一机制，知识图谱能够根据需求进行动态调整，以满足不同网络场景下的个性化要求，从而实现网络管理的目标。本文将复杂的配置分析问题转化为引导 LLM 自主获取相关信息并进行推理分析的问题。这些引导信息以提示词的形式传递给 LLM，涵盖应关注的分析要点以及应遵循的分析步骤，从而促使 LLM 回忆相关的专业知识，并按照合理的逻辑进行推理。此外，本文讨论了网络管理中常见的网络配置错误检测和关键节点识别的需求，并给出了相应的引导框架，在此基础上能够进行改编或扩展，以满足多样化的分析需求。

4.2 网络配置错误检测

网络配置错误检测关键在于识别网络状态能否支持网络基本功能。网络的基本功能是实现设备之间的连通性，在此基础上，才能支持其他网络功能。因此，在进行配置检测时，应当设定检测的优先级。按优先级从高到低的顺序，在网络拓扑、路由状态和服务状态检测3个方面进行，如图8所示。多片独立连通的网络可以被拆分为多个单一连通的网络进行研究分析。因此，本文主要讨论单片连通网络的配置，即知识图谱中不涉及多片连通网络的配置文件。

网络拓扑检测主要检查设备层面的连接状态。对于 Device 实体节点及 CONNECTED_TO 关系，在配置正确的情况下，这些实体节点与关系将组成一个连通图。如果出现孤立节点或多个连通图，说明设备的接口配置错误，导致设备无法建立连接。在这种情况下，通过查询状态为开启但缺乏 INTERFACE_CONNECT 关系的 Interface 实体，可以迅速定位配置错误的相关设备。同时，在检查设备接口配置的过程中，还需检查该设备的 ACL 策略是否对连接情况产生影响。

路由状态检测主要关注设备的路由配置情况。首先，从设备层面看，配置启用的路由协议必须正确声明其所连接的网络。在此过程中，还需检查设备的静态路由，以确保其与相连网络的 IP 地址相匹配。其次，路由协议应在多个相邻的设备间建立有效的联系。对于具有连接关系的设备节点，若需实现互通，则必须启用相同的路由协议。最后，需检查路由协议本身的配置是否符合路由协议规范。根据路由协议的 RFC 标准要求，检查相应路由协议配置的合规性。例如，OSPF 协议要

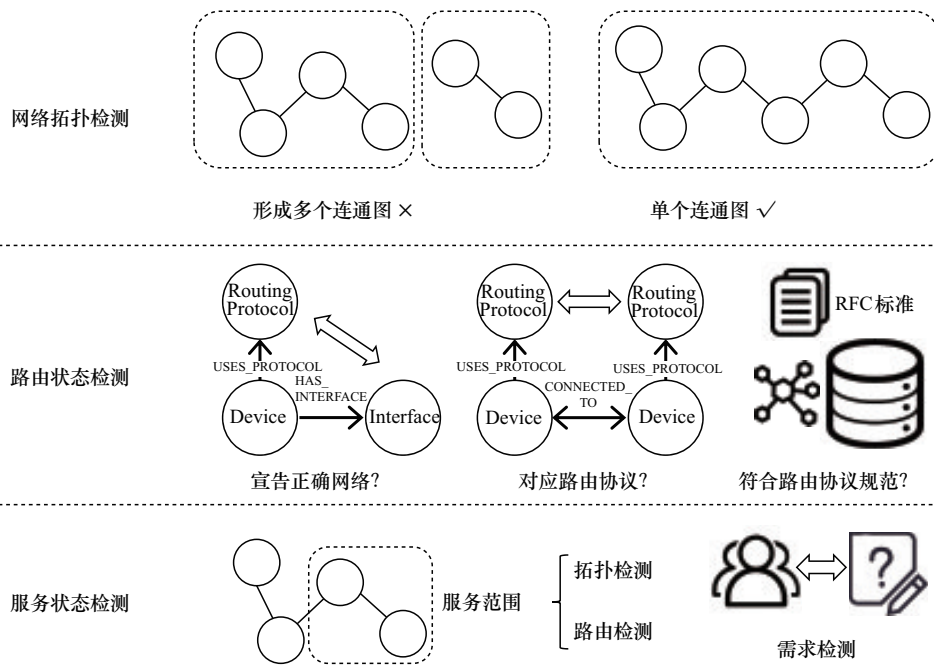


图8 网络配置错误检测

求 Area 0 作为核心传输区域，禁止其他区域直接进行路由通信^[27]。BGP 则要求域内具备 iBGP 全连接结构或正确使用路由反射器，以确保路由传输的准确性^[28]。

服务状态检测主要关注网络服务的正常运行。对于 VPN 类的服务而言，其通常与一般业务隔离，因此存在独立的路由配置。在这种情况下，需要重新检查该服务范围内的网络拓扑和路由状态，以确保其基本网络状态能够支撑服务。对于其他功能型服务，如 DNS 服务、安全外壳 (secure shell, SSH) 协议服务，则需在分析相关配置以判断其可用性的基础上，根据网络管理员的具体需求进行相应的检查。

综上所述，网络配置错误检测的过程始于对知识图谱网络状态的检查，随后根据网络管理员的实际需求进行深入分析。在具体实现过程中，本文采用 Lang Chain 框架对思维链的流程进行了严格规范，要求按照网络拓扑、路由状态和服务状态检测的顺序依次进行。每个步骤的结果均被记录在分析上下文中，以供后续检查参考，从而确保配置检测的优先级得以正确执行。其通用提示词如图 9 所示。提示词阐明了知识图谱的本体结构及可用的形式化工具组件，这些工具组件包含在推理阶段使用的形式化验证工具，用于满足特定的分析需求。此外，提示词还包含检测逻辑

和引导内容，这些引导采用自然语言对如图 8 所示的过程进行了详细描述。最终，LLM 将输出网络配置错误检测的分析结果。

```

[System] 你是路由网络领域的专家，负责帮助分析网络的运行情况。目前你可以依赖的知识图谱数据库的基本情况为：{当前的知识图谱本体结构} 为你提供的与知识图谱交互的工具组件为：{工具组件}
【用户提示词】
##请根据知识图谱中的信息进行推理分析。
需求：
请你对该网络的配置错误进行检查。{自定义的需求}
当前的任务阶段：
{网络拓扑/路由状态/服务状态}检测
##以下是配置检查的逻辑参考：
{检测引导}

##注意：
在进行分析时，请你依照上述给出的逻辑来进行推理。此外，如有需要，您还可以获取额外的信息以提高您的推理的准确性程度。
请避免直接获取知识图谱中的所有节点信息（例如 MATCH (n)、RETURN (n)）。当数据量较大时，您将难以处理！

请最后输出你的分析结果：

```

图9 网络配置错误检测的通用提示词

4.3 网络关键节点识别

网络关键节点是指在网络中承担重要角色以支持网络功能的路由设备。常见的相关配置主要集中于网络路由与服务的实现上，因此本文重点讨论网络路由与服务中的关键节点识别，其概况如图 10 所示。

在网络路由方面，路由信息通过动态选路确定。对于静态配置分析，路由信息只能通过设备所形成的拓扑结构及路由协议配置参数进行推导。首先，从整体网络拓扑的视角出发，网络中的设

备通过启用路由协议形成多个路由区域，这些路由区域的边界路由器通常承担着路由重分发的任务，因此被视为相对重要的节点。例如，自治系统内的边界路由器在传递 BGP 路由的同时，可能重分发自治系统内的域内路由，以宣告自治系统所拥有的网络前缀。在知识图谱中，这一识别过程可转化为优先查询具备多个路由协议实体的 Device 实体，以确认其重要性。其次，从各个路由协议形成的路由区域的视角出发，关键节点通常使用常见的网络指标进行识别。例如，网络拓扑中的节点度数能够反映设备节点的连通性，度数较高的设备节点通常具有更高的重要性。在知识图谱中，这一识别可转化为在相应路由区域下查询设备层面拓扑的节点度数，从而识别出关键实体节点。最后，从网络管理员的需求出发，LLM 在理解需求的基础上自行构建查询语句进行分析。例如，识别设备间路由情况的关键节点，可以在知识图谱中获得前往目标 T 的路径中必经的 Device 实体节点。

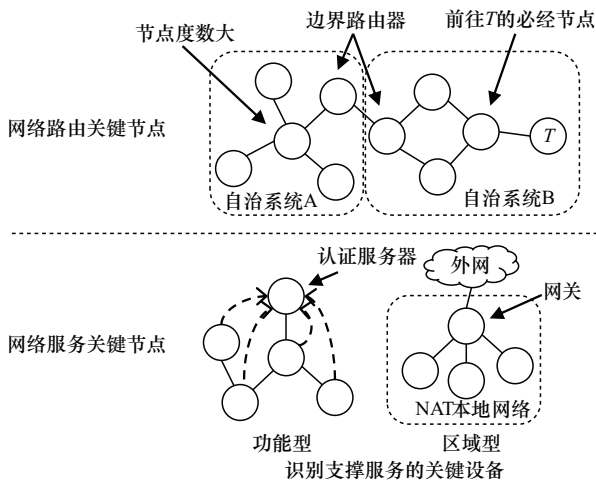


图 10 网络关键节点识别

在网络服务方面，服务范围通过配置进行划分，其运行通常需要网络路由的支持。因此，识别网络服务关键节点的第一步是在服务范围内确定网络路由的关键节点，其步骤与前文的方法一致。其次，根据服务的实质类型，确定网络服务的关键节点。对于功能型的网络服务，网络中一般具有关键的服务设备，如 SSH 等登录验证服务，通常在网络中使用认证服务器统一管理密钥。对于区域型的网络服务，网络中一般具有关键的区域边界设备，

例如，在开启网络地址转换 (network address translation, NAT)、动态主机配置协议 (dynamic host configuration protocol, DHCP) 等服务网络中，服务对应的网关设备即网络中的关键节点。最后，从网络管理员的需求出发，如需管理服务，通过 LLM 能够直接关联知识图谱中的相关设备，从而获取它们的服务状态。

综上所述，网络关键节点识别在一定程度上依赖于对网络路由的分析结果。因此，本文要求 LLM 依据预设的思维链，首先进行网络路由关键节点的检测，随后开展网络服务关键节点的识别工作。根据逻辑方法的引导，结合自身的专业知识赋能网络关键节点的识别任务，运用提供的形式化工具组件，实现自动化的分析过程。其通用提示词如图 11 所示，引导使用了自然语言描述图 10 中的方法，最终 LLM 将输出网络关键节点识别的分析结果。

```

[System] 你是路由网络领域的专家，负责帮助分析网络的运行情况。目前你可以依赖的知识图谱数据库的基本情况为：{当前的知识图谱本体结构}
为你提供提供的与知识图谱交互的工具组件为：{工具组件}
【用户提示词】
##请根据知识图谱中的信息进行推理分析。
需求：
请你识别该网络的关键节点。{自定义的服务分析需求}
##以下是进行识别的逻辑参考：
{识别路由/服务关键节点的引导}
##注意：
在进行分析时，请你依照上述给出的逻辑来进行推理。此外，如有需要，您还可以获取额外的信息以提高您的推理的准确性程度。
请避免直接获取知识图谱中的所有节点信息（例如 MATCH (n)、RETURN (n)）。
当数据量较大时，您将难以处理！

请最后输出你的分析结果：

```

图 11 网络关键节点识别的通用提示词

5 实验与评估

本文使用 Python 编写技术框架，通过 LangChain 框架与 OpenAI 库搭建 LLM 分析模块，使用 Neo4j 社区版图数据库^[29]构建知识图谱。该框架的原型在一台配备两个 Intel(R) Xeon(R) Silver 4210R x86_64 2.40 GHz CPU 的 Linux 服务器上运行。实验主要进行了以下评估：网络建模效果评估、分析应用效果验证、运行时间成本评估和基于真实拓扑的鲁棒性分析。实验配置源自 EVE-NG 官方提供的网络模拟实验室配置文件^[11]和思科公司 CCNA 培管所使用的配置案例^[12]。本文将这些配置输入技术框架，并对生成的分析结果进行评估。此外，本文基于现实 Internet Topology Zoo 中不同规模的骨干网拓扑结构^[13]，利用模拟生成的配置文件^[14]对

技术框架的鲁棒性进行评估与分析。最后，本文还与相关领域的已有工作进行对比与讨论，其在实验中使用 DeepSeek-R1 API^[30]。

5.1 网络建模效果评估

网络建模的效果主要基于以下 4 个指标进行评估：1) 设备识别的准确性，即识别网络中存在的设备品牌及其配置的建模是否正确；2) 网络拓扑构建的准确性，即推理还原的网络各层面的拓扑结构是否与实际情况相符；3) 判断网络路由状态的准确性，即识别启用的路由协议及其相互关系是否准确，以及网络中的路由可达性判断是否正确；4) 识别网络服务状态的准确性，即对网络中服务的识别及服务状态推理是否准确。评估结果分为√、△和×这 3 个等级，其中√表示能够满足对应指标，△表示在指标上部分正确但功能存在缺漏，×表示在指标上存在错误。本文针对不同需求下的网络场景进行了实验评估，涵盖交换机配置、路由器配置及网络服务配置等场景。场景名称反映了该场景的主要网络功能需求。除 NAT 配置场景采用了 Juniper 设备外，其他场景均使用 Cisco 设备。评估结果如表 6 所示，其中“—”表示该场景无法获取对应指标的评估结果或场景中不包含涉及该指标的相关内容。

表 6 多个场景下的网络建模效果评估

场景	指标 1)	指标 2)	指标 3)	指标 4)
交换机 STP VTP	√	—	—	√
单区域 OSPF	√	√	√	√
多区域 OSPF	√	√	√	—
多自治系统 BGP 配置	√	√	√	√
NAT 配置 (Juniper 设备)	√	√	√	√
MPLS VPN 配置	√	△	√	√

在本文实验中，LLM 依据配置文件的语法成功识别了相应的设备品牌，并在知识图谱中对应了每一个配置文件，按照 2.1 节的方法进行了准确建模，这表明了在形式化工具的辅助下，LLM 对配置语义提取建模的有效性与准确性。在交换机配置场景中，由于交换机接口通过二层协议直接相连进行数据转发，缺乏 IP 地址等配置信息，LLM 所依赖的推理信息不足，因此无法构建网络

拓扑或推断路由状态。但是模型仍然识别到了其中的生成树协议 (spanning tree protocol, STP) 和虚拟局域网干道协议 (VLAN trunking protocol, VTP)。面对仅启用二层交换协议的接口场景，形式化工具和 LLM 均无法判断连接情况，需借助进一步的外部信息进行引导。在路由协议配置的多个场景中，LLM 准确判断了路由协议所宣告的网络，并结合拓扑和路由情况识别了网络中的服务状态。在复杂的网络应用场景中，LLM 保持了良好的分析性能，还原了复杂的网络状态。值得注意的是，在 MPLS VPN 配置中，受限于以 Batfish 为基础的形式化工具对路由协议解析的支持范围，其略去了对思科专有协议 EIGRP 相关配置的解析，因此需要 LLM 根据外部知识进行额外补充。这些外部知识来源于 Cisco 官方文档，通过在思考解析过程中插入上下文的方式提供。经过上述过程，虽然问题得以解决，但仍需额外介入成本，因此本文将其评价为△。

实验结果表明，LLM 在理解配置文件和推理网络状态上展现了强大的能力。该技术框架的有效运行依赖于配置文件与形式化工具，网络中更加完备的配置文件会增强网络建模的效果，形式化工具能够快速提供 (见 5.3 节) 准确的结果以提升分析性能。对于一些较为冷门的配置知识 (如 EIGRP 配置)，LLM 需要更多的知识引导，以确保分析的准确性。通过知识图谱，LLM 在保留自身推理结果的基础上，能够支撑进一步的分析应用。总之，本文技术框架产出的网络建模与实际情况基本一致，能够满足现实的管理与分析需求。

5.2 分析应用效果验证

本文将从网络配置错误检测和网络关键节点识别两个角度进行分析应用效果的验证。对于网络配置错误检测，本文分析了 CCNA 培训中的故障排除场景，并将分析结果与实际错误进行对比，以验证配置检测的准确性。对于网络关键节点识别，本文分析了 EVE-NG 官方实验室中的一个企业网络通信场景，识别出其中的关键节点，并通过分析结果与现实情况的对比进行验证。

故障排除场景的拓扑结构通过 Cisco Packet Tracer 软件展示，如图 12 所示，中间 3 个设备为三层路由设备，两侧设备为部门的个人计算机。在该场景中，IT 部门 (VLAN 10) 与 Sales 部门 (VLAN

20) 无法通信, 需排查配置错误。经过网络状态的推理还原, 形成了如图 13 所示的知识图谱。图 13 中分布的节点代表多种不同含义的实体, 其中由黑色框标识的节点表示设备, 其 CONNECTED_TO 关系清晰地构成了与图 12 中主要三层路由设备的连接情况一致的网络拓扑, 验证了网络状态推理还原的有效性。由于各部门的个人计算机设备不存在配置文件, 实验中通过问答的方式确定了这些计算机设备与其他设备的连接方式。通过模拟网络管理员问答, 本文动态补全了知识图谱, 其过程及结果如图 14 所示。

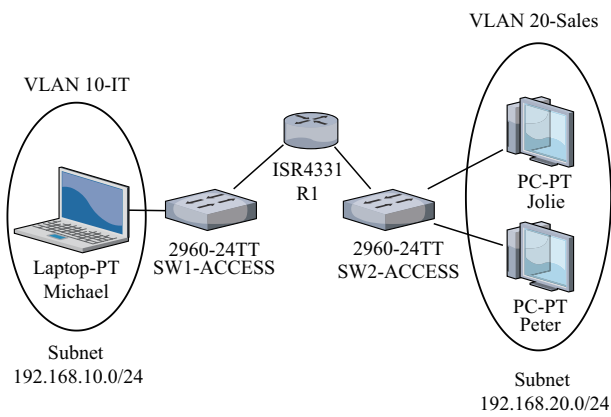


图 12 故障排除场景拓扑结构

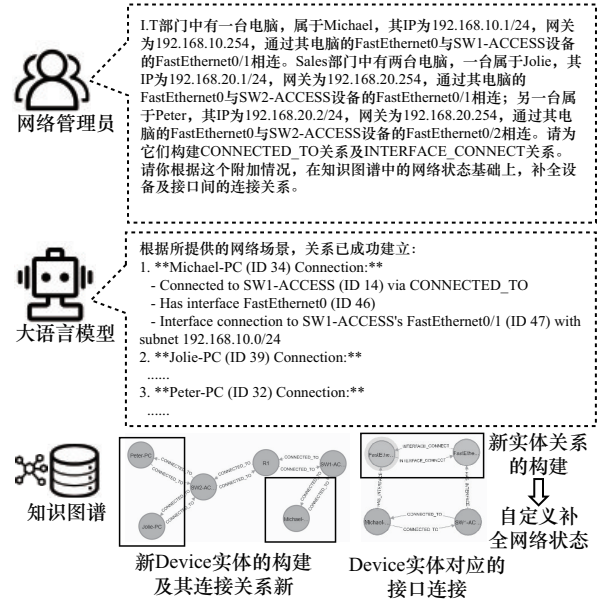


图 14 网络管理员问答补全知识图谱

通过将网络故障描述输入 LLM, 能够获得 LLM 对于该网络的分析报告, 如图 15 所示。LLM 从多个层面分析了网络配置错误, 包括 SW1 与 SW2 设备的配置缺失、R1 设备的路由配置缺失, 以及网络中需进一步确认的 IP 设置。其中, 配置缺失与实际错误情况相符, 是造成通信障碍的主

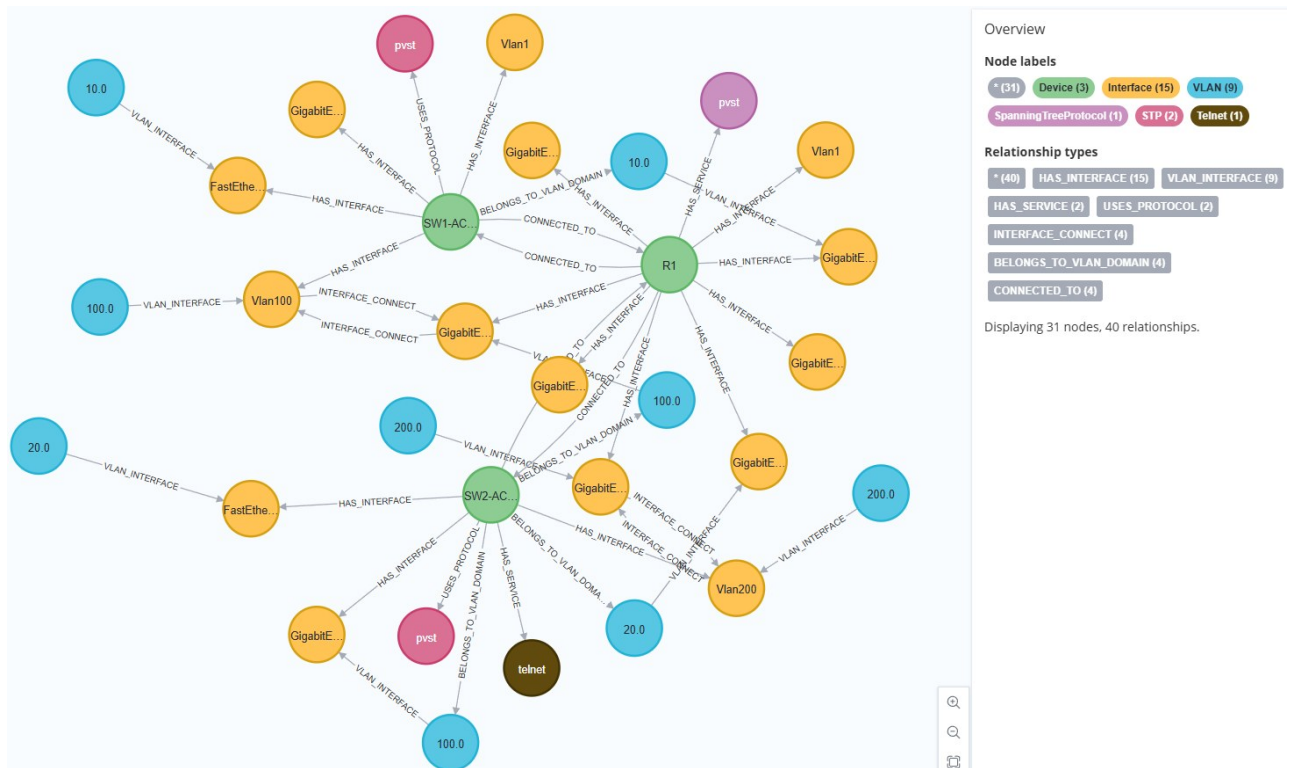




图 13 故障排除场景的网络状态知识图谱

要原因：SW1 缺少跨端口的 VLAN 转发权限配置，SW2 缺少对 VLAN 20 的跨端口的转发权限配置，R1 设备未开启地址解析协议（address resolution protocol, ARP）的代理转发服务。配置检测结果表明，通过该技术框架，网络管理员能够快速有效地定位并排除故障。



网络管理员

在知识图谱所示的网络中，Jolie无法与Michael和Peter进行通信。请你根据自己的专业知识，自行在知识图谱中获取对应信息，推断配置中存在的错误。



大语言模型

****已识别网络配置错误：****

- **VLAN 跨接配置错误（拓扑结构违规）****
-证据**:**
 -SW2 的接入端口（FastEthernet0/1-0/2）被配置为处于接入模式下的 ****VLAN 20****。
 -R1 的子接口 `GigabitEthernet0/0/1.200` 处理 VLAN 20，其 IP 地址为 `192.168.200.254/24`。
-SW1 的接入端口缺少允许 VLAN 20 的跨接端口**（执行 `trunk allowed_vlans` 查询未获取到结果）。**
-根本原因**:**
 -SW2 的接入端口无法将 VLAN 20 的流量传递到 SW1 的接入端口，原因是缺少允许 VLAN 20 的跨接端口权限。
 -VLAN 20 的交换机间通信被阻断，致使 Jolie-PC（VLAN 20）与其他设备隔离。
2. **三层路由由缺口（路由违规）**
-证据**:**
 -R1 的子接口 `GigabitEthernet0/0/1.200`（VLAN 200）拥有 IP 地址 `192.168.200.254/24`，但与 SW2-ACCESS 的接口 `192.168.200.1/24` 相连，两者之间没有路由协议。
-根本原因**:**
 -R1 和 SW2-ACCESS 之间缺少针对子网 `192.168.200.0/24` 的静态路由或动态路由（例如 OSPF）。
 -这使得 VLAN 20（Jolie-PC）无法通过 R1 路由到其他 VLAN 中。
3. **IP 地址冲突（服务相关）：**
-证据**:**
 -Peter 电脑的 IP 地址为 192.168.20.2/24
 -初始查询显示，Jolie 电脑的接口 IP 可能为 192.168.20.1/24（需进一步确认）
-根本原因**:** **Jolie 电脑与 Peter 电脑同属 VLAN 20 但 IP 地址冲突，会导致通信异常。

图 15 故障排除场景下 LLM 提供的网络配置错误分析报告

网络关键节点识别的应用通过 EVE-NG 实验室中一个典型的企业网络通信场景进行评估，其拓扑

如图 16 所示。该场景描述了企业 3 个站点的私有网络通过 IPsec VPN 服务构建公网上的安全通信通道以实现安全连接的网络配置。通过本文技术框架的推理还原，得到了如图 17 所示的网络状态知识图谱。知识图谱中的网络设备还原了与实际一致的网络拓扑。此外，多种节点关系准确反映了网络中启用的服务及路由协议。LLM 创建了一些语义重复的实体（如图 17 中的 IPsec VPN 与 VPN）以及极少量语义偏颇的关系（如图 17 中互联网服务提供商（Internet service provider, ISP）设备到 AS_Domain 的 AS_NEIGHBOR 关系），在检查验证与迭代分析过程中得以整理或删除。因此，系统框架在后续使用中保持了分析性能，证明了本文技术框架的鲁棒性。

图 18 展示了本文技术框架对网络关键节点识别的分析结果。LLM 根据网络的路由和服务状态，结合支撑网络功能的配置识别出关键设备并赋予其重要性评分。值得注意的是，LLM 在识别 VPN 服务的基础上，还关注到可能存在的失效问题，并合理推理出该情况下 ISP 路由器将重新进行路由。因此，LLM 指出 ISP 路由器对广域网连接至关重要。总体而言，LLM 通过与知识图谱的交互，获得了网络状态的背景知识，在结合分析引导与自身专业知识的基础上，识别出网络关键节点并提供了详细的报告。这些报告与实际网络情况基本一致，为网络管理员提供了直观可靠的参考。

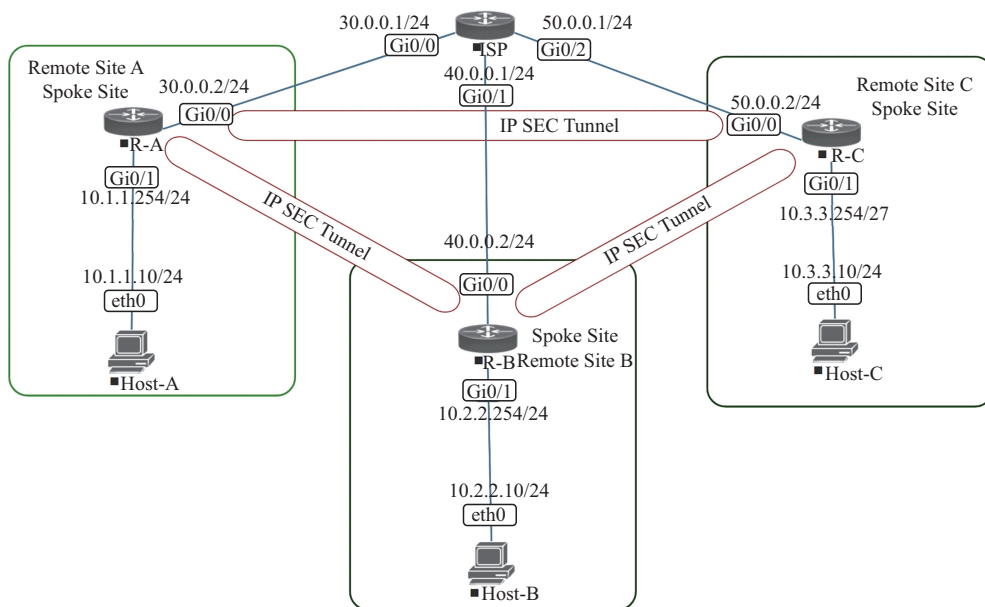


图 16 EVE-NG 实验室的企业网络 IPsec 通信场景拓扑

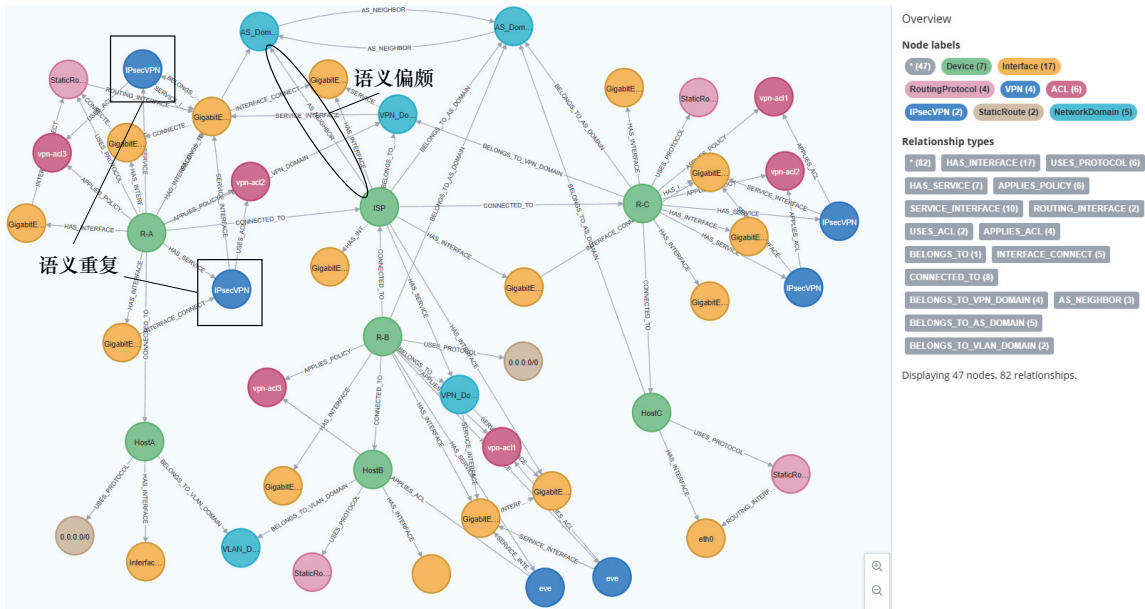


图 17 企业网络 IPsec 通信场景的网络状态知识图谱



网络管理员

在这片网络中，我想要找到网络中的关键设备节点，通常它的性能决定了这篇网络是否能够正常运作。你能够通过分析给出这个关键设备节点吗？



大语言模型

```

**关键网络设备分析**
#### 1. **关键路由节点**
##### **R-A (路由器 A)**
- **类型**：路由/服务
- **证据**：
  - **边界路由器**：连接至 AS_Domain (BGP) 和 VPN_Domain (IPsec)。
  - **节点度**：度数 = 2 (连接至 R-B 和 ISP)。
  - **VPN 服务**：承载 2 项 VPN/IPsec 服务 (对安全流量至关重要)。
  - **子网验证**：直接连接至 ISP (30.0.0.0/24) 和内部网络 (10.1.1.0/24)。
  - **重要性评分**：5 分
##### **R-B (路由器 B)**
- **类型**：路由/服务
- **证据**：
  - **边界路由器**：跨越 AS_Domain 和 VPN_Domain。
  - **节点度**：度数 = 2 (连接 R-A 和 HostA)。
  - **VPN 服务**：管理 2 个 VPN/IPsec 服务。
  - **子网验证**：连接到内部子网 (10.2.2.0/24)。
  - **重要性评分**：5 分
##### **ISP (互联网服务提供商路由器)**
- **类型**：路由
- **证据**：
  - **瓶颈**：所有外部流量均需通过 ISP (例如，从子网 50.0.0.0/24 到 R-C)。
  - **城域网**：连接 AS_Domain 和 VPN_Domain。
  - **子网验证**：直接连接到 R-A (30.0.0.0/24) 和 R-C (50.0.0.0/24)。
  - **重要性评分**：4 分
#### 2. **关键服务节点**
##### **R-C (路由器 C)**
- **类型**：服务
- **证据**：
  - **VPN 服务**：拥有 2 条 VPN/IPsec 隧道 (例如，IPsec_Tunnel_2 至 ISP)。
  - **子网验证**：连接至 ISP (50.0.0.0/24) 和内部网络 (10.3.3.0/24)。
  - **城域网**：属于 AS_Domain 和 VPN_Domain。
  - **重要性评分**：4 分
### **验证与上下文**
- **动态依赖关系**：
  - R-A 和 R-B 的重要性取决于 VPN 隧道的状态 (例如，如果 IPsec_Tunnel_1 失效，内部流量会通过 ISP 重新路由)。
  - ISP 作为唯一的外部网关对于广域网连接至关重要。
- **排除项**：
  - HostC (度数 = 1，单个域) 对于路由来说不是那么关键，但可能承载着关键的应用程序。
### **结论**
最关键的节点为 **R-A** 和 **R-B** (路由/服务) 以及 **ISP** (外部网关)。这些设备确保了跨域路由、VPN 安全性以及外部连接性。
  
```

图 18 IPsec 通信场景下 LLM 提供的网络关键节点识别报告

两个配置分析应用的实验表明，本文技术框架在实际应用中具有重要意义，能够根据需求一键提供详细的配置分析报告，从而帮助网络管理员更好地理解并管理网络。

5.3 运行时间成本评估

本文从构建知识图谱和推理网络状态两个方面对该技术的运行时间成本进行评估。运行时间成本主要源于 LLM 在 ReAct 过程中的时间消耗。为了验证形式化工具组件在提升网络状态还原效率方面的有效性，本文设计了消融实验：在相同实验环境下，比较了 LLM 在无形式化工具组件辅助与有形式化工具组件辅助两种条件下完成网络状态还原所需的时间成本。对比结果如表 7 所示。

表 7 各个场景还原网络状态的时间成本

场景	配置数量/个	无形式化工具组件/min	有形式化工具组件/s
交换机 STP VTP	5	7.66	6.43
单区域 OSPF	8	15.84	39.73
多区域 OSPF	7	11.3	45.80
多自治系统 BGP 配置	7	14.43	46.87
NAT 配置 (Juniper 设备)	3	8.48	64.11
MPLS VPN 配置	13	23.98	54.16
错误排除	3	16.01	61.32
关键节点识别	7	15.76	50.15

表7数据显示, 构建和推理时间与分析的有效配置数量密切相关。当配置中涉及的服务与接口数量较多时, LLM需要提取的实体关系也随之增加, 因此构建知识图谱所需的时间相对较长。在复杂场景下, 网络的内在拓扑结构与服务之间的关联性更为显著, 需要还原的网络状态更多, 导致推理所需的时间相对较长。分析框架通过结合形式化工具组件, 使推理还原网络状态的任务部分转化为综合使用形式化工具组件的调度问题, 充分利用LLM的智能理解能力实现高效操作。相比纯粹让LLM进行思考分析, 还原网络状态的时间从分钟级缩短至秒级, 充分证明了形式化工具组件协同工作的有效性。总体而言, 该技术框架能够在可接受的时间范围内完成配置文件的分析与网络状态的还原, 并以知识图谱形式进行存储。该过程可视为一种“冷启动”过程。在后续的维护和使用中, 仅需几秒钟即可加载本质为图数据库的知识图谱。因此, 该技术框架能够满足网络管理员维护多片网络配置的实际需求。

针对网络管理员日常使用的场景, 本文对问答分析的时间成本进行了评估。实验发现, 回答所需时间会因问题复杂程度的不同而有所差异, 通常介于几十秒到几分钟之间。对于简单问题, LLM能够在秒级时间内给出回答; 对于复杂问题, 其推理所需的时间成本则可参考推理网络状态所需时间, 通常为分钟级。

综上所述, 在网络配置维护场景中, 相较于易出错且耗时的人工分析, 本文技术框架显然更具优势。LLM赋能的分析技术框架能够有效帮助网络管理员在短时间内理解多种配置场景, 其较低的时间成本进一步证明了其在实际应用中的可行性。

5.4 基于真实拓扑的鲁棒性分析

为了模拟真实的网络配置环境, 本文使用了来自Internet Topology Zoo^[13]的真实互联网拓扑数据, 拓扑规模涵盖41~166个节点, 并使用Net-Complete^[14]生成了相应的配置文件。所生成的配置文件包含支持节点间互联的路由协议配置, 如BGP和OSPF, 并附带注释等信息。此外, 在不影响原有配置功能的前提下, 本文随机引入了一些冗余片段, 包括废弃的接口配置、废弃的ACL配置以及配置中常见的无意义内容(示例如图19所示)。通过这些配置文件进行输入分析, 本文验

证了所提系统框架在处理现实复杂配置时的鲁棒性。实验结果如表8所示, 本文从应用视角出发, 基于网络状态还原能力和问答响应时间对系统进行了评估, 并通过人工审计检测两个过程是否存在错误(如运行报错、建模冗余、答非所问及分析错误等情况)。问答环节的问题为随机选取网络中的两个节点, 针对其路由配置及与其他节点的连通性进行提问。

```

!
banner exec
*****
* IOSv is strictly limited to use for evaluation, demonstration and IOS *
* education. IOSv is provided as-is and is not supported by Cisco's *
* Technical Advisory Center. Any use or disclosure, in whole or in part, *
* of the IOSv Software or Documentation to any third party for any *
* purposes is expressly prohibited except as otherwise authorized by *
* Cisco in writing. *
*****
banner incoming
*****
* IOSv is strictly limited to use for evaluation, demonstration and IOS *
* education. IOSv is provided as-is and is not supported by Cisco's *
* Technical Advisory Center. Any use or disclosure, in whole or in part, *
* of the IOSv Software or Documentation to any third party for any *
* purposes is expressly prohibited except as otherwise authorized by *
* Cisco in writing. *
*****
banner login
*****
* IOSv is strictly limited to use for evaluation, demonstration and IOS *
* education. IOSv is provided as-is and is not supported by Cisco's *
* Technical Advisory Center. Any use or disclosure, in whole or in part, *
* of the IOSv Software or Documentation to any third party for any *
* purposes is expressly prohibited except as otherwise authorized by *
* Cisco in writing. *
*****
!

```

图19 配置中包含的无意义内容示例(Cisco设备)

表8 系统框架的鲁棒性分析结果

拓扑场景	配置数量/个	网络状态还原时间/s	问答响应时间/s	出现错误
Aries	42	70.36	318.42	否
Bics	41	51.19	240.07	否
Columbus	78	44.09	313.96	否
Latnet	77	44.89	224.18	否
UsCarrier	166	65.53	306.37	否
Colt	161	51.51	249.37	否

由表8可知, 在不同规模的网络环境中, 系统框架还原网络状态所需时间基本稳定在数十秒内。对于复杂查询问题, 能够在220~320s准确响应。基于Batfish开发的形式化工具组件在预处理阶段有效识别并剔除冗余配置, 同时初步过滤与配置无关的无效内容, 既加快了解析速度, 又保证了建模的准确性。在问答应用中, 该工具组件所提供的模拟数据平面信息可以有效辅助LLM分析路由状态,

促进其快速生成正确结果。表8中所述的“出现错误”特指LLM在用户问答过程中输出的回答是否存在事实性错误或逻辑矛盾。在网络状态还原的实际实现中,系统基于Lang Chain框架,按照预定流程进行分析,该流程的最终环节包含校验步骤。在校验过程中,LLM对已基本完善的结果进行用户无感知的必要调整。因此,在实际使用中,实验中网络拓扑状态还原的结果均为一次性准确生成。依托还原结果,LLM得以在结合形式化工具组件的基础上,提供准确的回答。经人工验证,系统的回答与网络事实保持一致,体现了其在实际应用场景中的鲁棒性。

值得注意的是,DeepSeek-R1在应对复杂任务时,其庞大的模型规模与冗长的思维过程,导致显著的推理开销和响应时延^[31],这成为系统框架在多轮分析过程中的主要时间瓶颈。作为对比,本文还对GPT-4o-mini^[32]等小参数模型进行了初步测试。此类模型能够显著加快响应速度(通常在10~20 s即可完成问答任务),因此能更适合需要高频、实时交互的应用场景。本文在设计上兼容多种LLM,可根据不同任务对性能与速度的需求,灵活选用合适的模型,以平衡分析深度与响应效率。然而,这些LLM在具体任务上的性能仍需进一步评估,以便选择最适合的模型来适配不同应用场景。

5.5 与现有工作的对比

本文构建了一个由LLM赋能的网络配置分析框架,使网络管理员能够通过自然语言交互,直观便捷地完成配置的查询、分析与维护。与Batfish^[3]、Minesweeper^[5]、Tiramisu^[6]、ConfigKG^[9]、XNV^[10]等现有工作相比,本文工作在分析范式与用户体验层面实现了显著差异与扩展。现有工作大多依赖特定的中间表示,要求使用者深入理解其内部语义并编写相应代码或查询语句,在面对复杂和跨领域的分析需求时存在较高的使用门槛和可解释性壁垒。相比之下,本文工作以可解释的知识图谱作为统一的、符合人类认知的语义中介,并借助LLM的自然语言理解与推理能力,将复杂的配置分析任务转化为对知识图谱的语义查询与结合形式化工具组件的分析推理问题。这一设计使本文框架能够灵活适配多种运维场景,自动化生成直观、可解释的分析报告,有效降低了网络管理的认知负荷与操作成本。

表9从使用方式、建模介质可解释性、在线动态维护更迭与多场景应用扩展4个维度,对本文工作与现有工作进行了系统性对比,其中√表示在该维度上表现优异,×表示能力不足或实现成本较高。现有工具(如Batfish、Minesweeper、Tiramisu)均要求使用者具备专业技术背景,通过编写交互代码、约束条件或命令或脚本来实现分析目标,存在较高的使用门槛。近年来,研究趋势开始注重提升工具的可解释性,ConfigKG与XNV等工作均采用知识图谱作为建模媒介,验证了该途径在增强模型语义透明度方面的有效性。本文的核心贡献在于结合知识图谱的语义描述能力与LLM的问答推理能力,实现了交互范式与应用能力的智能化转型。具体而言,首先,在交互方式上,本文构建了自然语言对话平台,使网络管理员不需要编写复杂的规约脚本,即可通过直观的问答形式完成分析任务。其次,在应用能力方面,本文依托LLM的认知与推理功能,自主调用提供的工具组件,实现了知识图谱的在线动态维护及多场景的灵活扩展。该系统能够根据用户的动态需求实时更新知识图谱,并通过不同问题的询问方式,快速适配配置错误检测、关键节点识别等多种运维场景,避免了人工修改代码和人力分析等烦琐过程。

表9 本文工作与现有工作的对比

工作	使用方式	建模介质 可解释性	在线动态 维护更迭	多场景 应用扩展
Batfish	编写交互代码	×	×	×
Mine-sweeper	编写约束条件	×	×	×
Tiramisu	编写命令脚本	×	×	×
ConfigKG	编写数据查询 规则	√	×	×
XNV	编写规则文件	√	×	×
本文工作	自然语言 对话交互	√	√	√

6 结束语

本文针对当前网络配置分析工具智能化水平有限、使用门槛高的问题,对LLM赋能的网络配置分析技术进行了深入研究,提出了一种全新的分析框架。该框架创新性地形式化工具与LLM相结合,通过自动化实体关系抽取构建可解释的知识图

谱,并基于此开展多层次的网络状态推理与补全,实现了对网络状态的精准还原。通过 LLM 与知识图谱之间的自动化交互,本文框架实现了从配置解析、状态还原到智能问答的完整闭环,有效支持了网络配置错误检测与关键节点识别等核心应用,显著提升了网络配置分析工具智能化水平。此外,本文提供的基于知识图谱的数据库与自然语言问答平台,构建了一个低门槛且具备可持续运维能力的人机交互系统。实验结果表明,该框架能够在多种真实及模拟场景下高效且准确地分析配置文件,验证了其可行性、有效性与实用价值。本文为网络配置分析技术的发展提供了新的思路与实现路径,为网络管理提供了重要技术支撑。

参考文献:

- [1] Li Y H, Yin X, Wang Z L, et al. A survey on network verification and testing with formal methods: approaches and challenges[J]. *IEEE Communications Surveys & Tutorials*, 2019, 21(1): 940-969.
- [2] Abhashkumar A, Gember-Jacobson A, Akella A. AED: incrementally synthesizing policy-compliant and manageable configurations[C]//*Proceedings of the 16th International Conference on Emerging Networking EXperiments and Technologies*. New York: ACM Press, 2020: 482-495.
- [3] Fogel A, Fung S, Pedrosa L, et al. A general approach to network configuration analysis[C]//*Proceedings of the 12th USENIX Symposium on Networked Systems Design and Implementation*. Berkeley: USENIX Association, 2015: 469-483.
- [4] Martinho C, Strickx T. Understanding how facebook disappeared from the Internet[R]. CloudFlare, 2021.
- [5] Beckett R, Gupta A, Mahajan R, et al. A general approach to network configuration verification[C]//*Proceedings of the Conference of the ACM Special Interest Group on Data Communication*. New York: ACM Press, 2017: 155-168.
- [6] Abhashkumar A, Gember-Jacobson A, Akella A. Tiramisu: fast multi-layer network verification[C]//*Proceedings of the 17th USENIX Symposium on Networked Systems Design and Implementation*. Berkeley: USENIX Association, 2020: 201-219.
- [7] Alam S, Lee D, Gember-Jacobson A. Poster: identifying syntactic motifs and errors in router configurations using graphs[C]//*Proceedings of the 2022 IEEE 30th International Conference on Network Protocols (ICNP)*. Piscataway: IEEE Press, 2022: 1-2.
- [8] Zhang P, Gember-Jacobson A, Zuo Y S, et al. Differential network analysis[C]//*Proceedings of the 19th USENIX Symposium on Networked Systems Design and Implementation*. Berkeley: USENIX Association, 2022: 601-615.
- [9] Li P F, Liu Y J, Su J S, et al. ConfigKG: identify routing security issues from configurations based on knowledge graph[C]//*Proceedings of the 2024 IEEE 23rd International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom)*. Piscataway: IEEE Press, 2025: 1060-1069.
- [10] Li F L, Li M L, Pu Y H, et al. XNV: explainable network verification[J]. *IEEE/ACM Transactions on Networking*, 2024, 32(6): 5097-5111.
- [11] EVE-NG. Enjoy EVE-NG labs library section[R]. 2025.
- [12] LuizMurosLima. TshootLabsCCNA[R]. 2024.
- [13] Knight S, Nguyen H X, Falkner N, et al. The Internet topology zoo[J]. *IEEE Journal on Selected Areas in Communications*, 2011, 29(9): 1765-1775.
- [14] El-Hassany A, Tsankov P, Vanbever L, et al. NetComplete: practical network-wide configuration synthesis with autocompletion[C]//*Proceedings of the 15th USENIX Conference on Networked Systems Design and Implementation*. Berkeley: USENIX Association, 2018: 579-594.
- [15] 王萌, 王昊奋, 李博涵, 等. 新一代知识图谱关键技术综述[J]. *计算机研究与发展*, 2022, 59(9): 1947-1965.
Wang M, Wang H F, Li B H, et al. Survey on key technologies of new generation knowledge graph[J]. *Journal of Computer Research and Development*, 2022, 59(9): 1947-1965.
- [16] Liu Y J, Wang Z L, Li P F, et al. Research on modeling technology for the Internet routing system based on knowledge graph[C]//*Proceedings of the 2024 9th International Conference on Computer and Communication Systems (ICCCS)*. Piscataway: IEEE Press, 2024: 660-665.
- [17] 秦小林, 古徐, 李弟诚, 等. LLM 综述与展望[J]. *计算机应用*, 2025, 45(3): 685-696.
Qin X L, Gu X, Li D C, et al. Survey and prospect of large language models[J]. *Journal of Computer Applications*, 2025, 45(3): 685-696.
- [18] Anepaka Y, Pakray P. Large language models: a survey of their development, capabilities, and applications[J]. *Knowledge and Information Systems*, 2025, 67(3): 2967-3022.
- [19] Gui Z W, Wang E Z, Deng B B, et al. SqliGPT: evaluating and utilizing large language models for automated SQL injection black-box detection[J]. *Applied Sciences*, 2024, 14(16): 6929.
- [20] Yan Y Q, Zheng P, Wang Y J. Enhancing large language model capabilities for rumor detection with knowledge-powered prompting[J]. *Engineering Applications of Artificial Intelligence*, 2024, 133: 108259.
- [21] Zhang Y H, Du T W, Ma Y S, et al. AttacKG+: boosting attack graph construction with large language models[J]. *Computers & Security*, 2025, 150: 104220.
- [22] Tan W C, Li Y L, Rodriguez P, et al. Reimagining retrieval augmented language models for answering queries[C]//*Proceedings of the Findings of the Association for Computational Linguistics*. Stroudsburg: ACL, 2023: 6131-6146.
- [23] Pan J Z, Razniewski S, Kalo J C, et al. Large language models and knowledge graphs: opportunities and challenges[J]. *Transactions on Graph Data and Knowledge*, 2023, 1(1): 1-38.
- [24] 冯拓宇, 李伟平, 郭庆浪, 等. LLM 增强的知识图谱问答研究进展综述[J]. *计算机科学与探索*, 2024, 18(11): 2887-2900.
Feng T Y, Li W P, Guo Q L, et al. Overview of knowledge graph question answering enhanced by large language models[J]. *Journal of Frontiers of Computer Science & Technology*, 2024, 18(11): 2887-2900.
- [25] Lewis P, Perez E, Piktus A, et al. Retrieval-augmented generation for knowledge-intensive NLP tasks[C]//*Proceedings of the 34th International Conference on Neural Information Processing Systems*. Massachusetts: MIT Press, 2020: 9459-9474.

- [26] Yao S Y, Zhao J, Yu D, et al. React: synergizing reasoning and acting in language models[C]//Proceedings of the International Conference on Learning Representations. Vancouver: ICLR, 2023: 30084-30116.
- [27] IETF. OSPF version 2: RFC 2328[R]. 1998.
- [28] IETF. RFC 4271: a border gateway protocol 4 (BGP-4)[R]. 2006.
- [29] NEO4J. Deployment center[R]. 2025.
- [30] Guo D Y, Yang D J, Zhang H W, et al. DeepSeek-R1: incentivizing reasoning capability in LLMs via reinforcement learning[J]. arXiv Preprint, arXiv: 2501.12948, 2025.
- [31] Wang J K, Li J T, Hou J Y, et al. Efficient reasoning for LLMs through speculative chain-of-thought[J]. arXiv Preprint, arXiv: 2504.19095, 2025.
- [32] Achiam J, Adler S, Agarwal S, et al. GPT-4 technical report[J]. arXiv Preprint, arXiv: 2303.08774, 2023.

[作者简介]



李鹏飞 (2000-), 男, 安徽无为, 国防科技大学博士生, 主要研究方向为计算机网络、网络安全。



刘宇靖 (1985-), 女, 山东曹县人, 博士, 国防科技大学副研究员、硕士生导师, 主要研究方向为网络路由安全等。



苏金树 (1962-), 男, 福建莆田人, 博士, 国防科技大学教授、博士生导师, 军事科学院研究员, 主要研究方向为网络体系结构、网络空间安全、智能网络等。



喻波 (1985-), 男, 湖南宁乡人, 博士, 国防科技大学研究员、博士生导师, 主要研究方向为软件安全、系统安全等。