GAI使能网络智能化：基于LLM的网络操作系统

霍如[1,2]，沙宗轩[1]，吕科呈[1]，陈伟[3]，汪硕[3]，黄韬[3]，F. Richard Yu[4]
（1.北京工业大学信息科学技术学院，北京 100124；2.紫金山实验室，江苏 南京 211111；
3.北京邮电大学网络与交换国家重点实验室，北京 100876；4.卡尔顿大学，渥太华 K1S 5B6）

摘　要：为了实现对网络高智能化可管可控的目标，提出了基于大语言模型（LLM）的高可控大网级网络操作系统。通过网络态势感知和微调大模型的意图理解优化资源配置，实现智能化运维。同时，设计了面向中国网络操作系统（CNOS）的网络大模型微调和推理流程，可识别CNOS指令并对系统反馈结果进行归纳总结，周期性的自主训练及模型迭代更新。实验结果表明，所提系统能够快速识别并准确转换用户指令，有效降低操作系统管理任务时间，实现网络资源的智能化调度和配置，提升网络操作系统的可控性和人机交互的友好程度。

关键词：网络大模型；网络操作系统；大语言模型；流量调度；智能运维
中图分类号：TP393
文献标志码：A
DOI：10.11959/j.issn.1000–436x.2025113

General artificial intelligence enables network intelligence: a network operating system based on large language model

HUO Ru[1,2], SHA Zongxuan[1], LYU Kecheng[1], CHEN Wei[3], WANG Shuo[3], HUANG Tao[3], YU F. Richard[4]
1. School of Information Science and Technology, Beijing University of Technology, Beijing 100124, China
2. Purple Mountain Laboratories, Nanjing 211111, China
3. State Key Laboratory of Networking and Switching Technology, Beijing University of Posts and Telecommunications, Beijing 100876, China
4. Carleton University, Ottawa K1S 5B6, Canada

Abstract: To achieve the goal of highly intelligent and controllable network, and a highly controllable large-scale network operating system was proposed based on large language model (LLM). Through network situation awareness and the intention understanding of fine-tuning LLM, resource allocation was optimized to achieve intelligent operation and maintenance. Meanwhile, the fine-tuning and inference flow for China network operation system (CNOS) oriented network large model was designed to recognize the CNOS system instructions and summarize the feedback results of the system, as well as periodic autonomous training and iterative model updating. The experimental results show that the proposed system can quickly identify and accurately convert user instructions, effectively reduce the management task time of the operating system, realize the intelligent scheduling and configuration of network resources, and improve the controllability of the network operating system and the friendliness of human-computer interaction.

Keywords: network large model, network operating system, large language model, traffic scheduling, intelligent operation and maintenance

收稿日期：2025–04–28；修回日期：2025–06–09
通信作者：霍如，huoru@bjut.edu.cn
基金项目：北京市自然科学基金资助项目（No.4254064）
Foundation Item：The Natural Science Foundation of Beijing (No.4254064)
*第二十七届中国科协年会“AI时代网络技术创新”专题

0　引言

随着ChatGPT等大型人工智能（AI, artificial intelligence）模型的广泛应用，生成式人工智能（GAI, generative artificial intelligence）得到越来越多的关注。GAI是一种利用人工智能算法创造性地生成和修改多样化类人数据的自动化方法[1]，可应用于各种形式的内容生成，包括文本、图像、视频、增强训练样本以及交互式3D内容等[2]。根据Gartner的数据，到2025年，GAI产生的数据将约占总数据的10%。

GAI产生数据的过程可分为2个主要阶段：首先提取和理解用户意图，然后根据意图生成内容。2022年，由OpenAI发布的ChatGPT是一种基于Transformer的大语言模型（LLM, large language model）。该模型可以生成代码、编写文档、翻译以及归纳总结内容。ChatGPT利用思维链（CoT, chain of thought）提示实现了小样本和零样本推理以及基于人类反馈的强化学习（RLHF, reinforcement learning from human feedback）的LLM微调[3-4]。凭借出色的自然语言处理、多领域知识和上下文理解能力，ChatGPT在公开发布仅2个月后实现了每日约1 300万用户的交互记录。此外，稳定扩散模型和DALL系列模型验证了大型视觉模型根据文本描述产生高分辨率图像的出色性能。作为上游技术，GAI拥有巨大的潜力支持不同的下游应用，目前已经初步在物联网和车联网等垂直领域用来生成模拟数据支持深度模型训练，然而在网络领域的赋能还有待拓宽。

随着信息技术的发展，基础网络服务性能提升，深刻地改变社会的生产方式。从简单的设备数据传递，到万物互联的智能时代，互联网在各个领域发挥着越来越重要的作用，促进新业态新模式快速发展。将大量应用和设备连接起来的通信网络，其规模和复杂性也在急剧增加。这种复杂性不仅体现在设备数量以及流量的增长上，还涉及网络架构、协议标准和操作系统。网络操作系统（NOS, network operating system）作为一种能够管理和控制网络资源、任务分配以及协调各单元之间通信的软件系统，为网络用户提供资源共享、协同工作，以及使用网络底层资源的桥梁[5-6]。加强NOS的研发和应用，对于推动未来网络的发展具有重要意义。

NOS建设基于服务器集群构建的全网控制平台，整体建设分为主干网操作系统、边缘网操作系统、云数据中心操作系统和协同操作系统。其中，主干网操作系统部署在主干网中心控制区域的节点上，成为集中式的软件定义网络（SDN, software defined networking）主干网控制器集群，提供主干网全网拓扑和路由转发策略的统一获取和管理，实现主干网全局流量调度和网络优化。边缘网操作系统部署在全国多个边缘网络中，对本地边缘网络设备和试验进行管理控制，实现局部网络控制。云数据中心操作系统部署在SDN云数据中心中，实现多资源池统一管理、网络资源与计算资源的解耦，并支持租户逻辑网络业务自动化编排和部署。协同操作系统部署在SDN云数据中心，通过互通协议协同连接各个区域的边缘网控制器和主干网操作系统，提供整个网络跨域试验的协同控制。SONiC（software for open networking in the cloud）是微软开发的基于容器的网络操作系统，可以灵活部署功能和应用[7]。FBOSS（facebook open switching system）是脸书公司开发的可屏蔽底层设备差异性的网络操作系统[8]。此外，谷歌推出的Stratum项目也实现了基于SDN的数据平面参考平台。我国科研团队也自主研发了高性能、可扩展、服务化、开放性的中国网络创新环境（CENI, China environment for network innovations）和中国网络操作系统（CNOS, China network operation system），实现了大网级网络操作系统的设计、编程和应用。

然而，NOS面临应用部署难度高、智能化程度低、扩展性差、依赖专家维护、存在较高的使用门槛、对新需求的响应时间长等方面的问题。为了使NOS从规则驱动向意图驱动的运维范式转变，提升网络管理的可编程性，实现大规模网络“全网智能管控”的目标，本文提出了基于LLM的网络操作系统，具体工作如下。

1)设计了基于LLM的高可控大网级网络操作系统的层次化体系架构。进一步在CNOS的基础上，通过微调和部署网络大模型转变运维范式，为CNOS带来构建智能化意图解析与自动化业务配置的能力。

2)构建了面向CNOS的网络大模型。基于系统控制指令构建微调数据集，通过对基座大模型的微

调，实现对网络操作指令的理解。一方面，可以将用户意图转换成系统指令；另一方面，可以将执行结果进行归纳总结，提高系统交互的友好性。

3)设计了网络大模型的动态微调和推理流程。动态微调流程实现系统使用数据积累，并周期性地对基座模型进行微调，实现模型性能的迭代上升。推理流程实现网络大模型从用户侧到系统侧的双向使用过程。

4)在真实大网场景中部署本文所提架构和模型，验证了网络大模型将用户意图识别为系统控制指令的准确性。模型取得了最优的ROUGE-L F1分数，针对典型单一场景和全部网络管理场景，指令生成平均时间最低为1.463 5 s，最高为3.563 1 s，执行耗时普遍在100 ms以内。实验证明了网络大模型对业务快速响应的能力，能够有效提升人机交互友好度，以及系统运维和管理效率。

1　相关工作

1.1　LLM技术

随着自然语言处理（NLP, natural language processing）领域中基于神经网络方法的发展，LLM应运而生。为了处理语言，早期的自然语言处理系统利用了基于规则的技术和统计模型。然而，这些方法在理解特定话语中的文本上下文时经常遇到困难，如短期记忆容量有限以及容易出现过度拟合[9]。为了解决这些问题，LLM利用深度学习技术，特别是Transformer架构，来学习和理解语言数据中存在的复杂模式和结构[10]。这些模型具有大量参数（通常为数十亿或更多权重），并使用自监督或半监督学习在大量未标记文本上对这些参数进行训练。得益于基于自注意机制的Transformer模型，LLM实现了并行化训练和对长距离依赖关系的有效处理，使模型能够考虑句子或文档的整个上下文，从而实现真正的上下文理解。

LLM准备流程如图1所示，一般可以分为数据收集、数据处理、预训练和模型微调。数据收集阶段需要为LLM准备充足的训练数据，这些数据可以来自互联网论坛、书籍、新闻或者已有的公开数据集。由于LLM对数据的质量有着更高的要求，因此还需要对数据进行处理，LLM能力在很大程度上依赖于预训练语料库及其预处理方式。在数据处理过程中，首先，需要进行质量筛选以从收集的语料库中去除低质量的数据。然后，需要对数据进行数据清洗，其中包括重复数据剔除、缺失值处理、异常值处理等操作。之后，由于大多数预训练文本数据都是从网络来源获得的，可能涉及敏感信息或个人信息，因此需要对涉及隐私的数据进行删除和模糊化处理。最后，需要对数据进行标记化，将原始文本分割成单个标记的序列，这些标记的序列随后将用作LLM的输入。利用处理完成后的数据对模型进行预训练，使模型可以从大规模未标注文本数据中学习到通用语言表示，从而捕获语言的深层结构和统计规律。

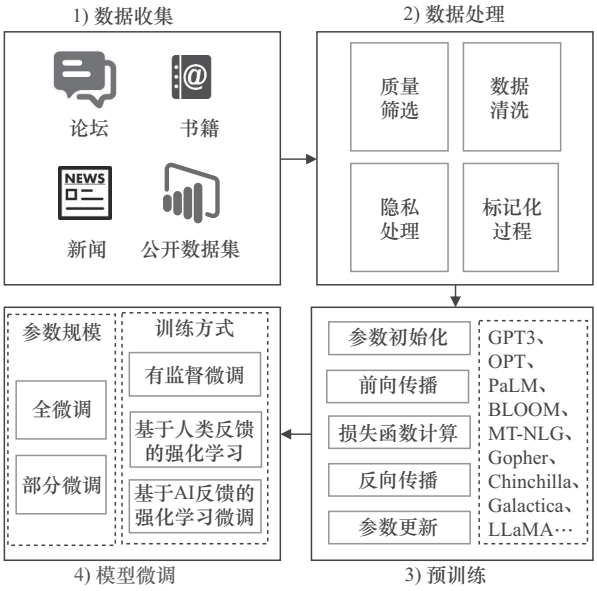

图1　LLM准备流程

当前，已经有很多LLM模型架构可以作为基础模型的选择，如GPT3[11]、OPT[12]、PaLM[13]等。这些模型经过参数初始化、前向传播、损失函数计算、反向传播、参数更新等流程后就初步具备了学习词汇、句法和语义的规律，以及上下文之间关系的能力。预训练的模型在实际使用前还需要进行模型微调，以根据用户自身的具体需求定制模型，从而提高准确性和性能，更好地适应特定场景。模型微调的方法可以分成不同的类别。例如，按照微调参数规模可以分为全微调和部分微调，它们在参数更新上分别进行全部参数更新和部分参数更新。此外，按照训练方式可以分为有监督微调、基于人类反馈的强化学习微调和基于AI反馈的强化学习微调。其中，有监督微调用人工标注的数据和监督学习的方法对大模型

进行微调；基于人类反馈的强化学习微调将人类的反馈和强化学习方式引入大模型的微调；基于AI反馈的强化学习微调则用AI取代了基于人类反馈的强化学习微调中的人类反馈以提高效率。经过不同方式微调并且通过性能评估测试的LLM就可以在不同场景中展开应用。

1.2　网络操作系统

网络操作系统是网络的运行管理者，在整个网络中承担着承上启下的作用，具有网络环境下对底层物理资源进行管理和控制的功能。作为用户与底层网络资源的接口，一方面，网络操作系统通过相应的接口协议对底层网络设备进行集中化的管理、状态监测、转发决策，以处理和调度下层数据平面的流量；另一方面，网络操作系统通过相应的接口为上层应用提供多层次的可编程功能，从而根据应用场景灵活地制定网络策略。面对复杂、异构的网络环境，研究者们正在通过各种技术实现具备自动化、智能的网络管理的网络操作系统。其中，人工智能作为必不可少的工具已经被应用于网络的感知、挖掘、预测和推理等任务[14]。

伴随着生成式人工智能的热潮，研究者们也开始探索LLM在网络等垂直领域的应用。在电信语言理解上，LLM可以帮助用户理解电信对话文本并提取相关信息，以将用户请求的内容转换为可操作且可解释的网络策略[15]。在流量生成方面，LLM具有很好的“泛化”能力，可以被用来生成具有不同特征的网络流量[16]，从而对网络操作系统的策略进行更好的验证。在网络安全方面，LLM可以在网络防御自动化、网络安全报告生成、威胁情报收集分析、安全代码生成和检测、网络攻击的识别等方面提供帮助[17]，以保护网络操作系统管理的资源免受未经授权的访问、盗窃、损坏或破坏。值得注意的是，若将LLM用于网络攻击，也将对网络操作系统安全造成威胁。在网络协议与数据包分析上，LLM可以更好地理解协议或数据包的规则、状态、通信流、消息结构等复杂协议实体，从而助力网络实现自动化分析[18]。在系统方面，文献[19]利用LLM实现共享网络自动化的协作框架，实现了基于共享意图的6G网络中利益相关者之间的冲突解决和协作。文献[20]介绍了通过LLM构建零接触网络和服务管理配置代理的网络配置生成器，以通过自然语言表达的意图自动生成配置、验证配置并配置网络设备。文献[21]提出以LLM为中心的意图生命周期管理架构，旨在使用自然语言配置和管理网络服务。与上述研究不同，本文提出的系统更加侧重于实现LLM驱动的网络“感知—决策—执行—反馈”闭环系统管理机制。面对网络操作系统智能化、自动化设备的需求，LLM将在更多网络场景发挥作用，以应对不断扩大的网络规模、动态时变的环境、多样化的用户需求和复杂的手动配置[22]。

2　基于LLM的智能网络操作系统

基于上述的趋势、需求和问题分析，本文设计了基于LLM的高可控大网级网络操作系统。该系统通过引入数字孪生技术和先进的网络控制算法，旨在提升网络管理的智能化水平，实现对大规模网络的高效控制与管理。本节将对系统整体进行概述，并对其4个关键层次进行详细说明。

2.1　研究动机

传统NOS运维模式在操作效率、智能化等能力上存在不足，主要体现在：1) 命令行界面（CLI, command line interface）和图形用户界面（GUI, graphical user interface）操作复杂性高，难以对动态网络事件进行快速响应，以及智能化程度不足，缺乏对网络状态的理解；2) 当前运维依赖具备深厚理论知识和丰富实践经验的专业人员，效率受限于专家经验。面对日益增长的网络规模和复杂多样的业务需求，这种人工密集型的规则驱动运维模式已显现出效率和可扩展性低下、响应滞后性等问题。

引入LLM赋能NOS，旨在突破传统NOS局限性。利用LLM的语言理解和生成能力，结合CNOS全局态势感知能力，实现NOS从规则驱动向意图驱动的运维范式转变，提升网络运维的可编程性。此外，LLM能够对海量的日志、性能指标和事件数据进行语义分析，实现潜在故障模式、检测异常行为并推荐优化方案，提供差异化业务服务能力，具备高可用、弹性伸缩、安全可靠等特性。

2.2　系统概述

原CNOS基于SDN架构实现，根据转控分离的思想，由网络控制器、数字孪生层和物理层构成。

基于LLM的智能网络操作系统以CNOS为基础，由物理层、数字孪生层、智能控制层和网络应用层组成，系统架构如图2所示。每层间相互协同工作，共同实现对网络的全面控制与管理。物理层承载着数据传输的任务，该层的数据为数字孪生层提供了基础数据来源，确保数字孪生模型能够准确反映物理网络的实际状态。数字孪生层的数据和分析结果为智能控制层提供了精准的实时网络状态和预测信息，帮助其制定和调整控制策略。智能控制层利用数字孪生层提供的实时数据和预测信息，进行动态资源调度和优化，实现网络的智能化管理。其中，如图2所示箭头指示方向，在下行方向，网络大模型将以意图为导向的自然语言转换成系统指令下发给数字孪生层进行策略验证；在上行方向，网络大模型可归纳网络状态和策略执行结果，其决策和控制指令直接作用于物理层，调节和优化物理网络的运行。网络应用层通过调用智能控制层的控制策略和算法，利用数字孪生层的实时数据和分析结果，进一步提升网络管理的安全性和智能化水平。此外，它还为用户提供了便捷的操作接口，使管理操作更为直观和高效。

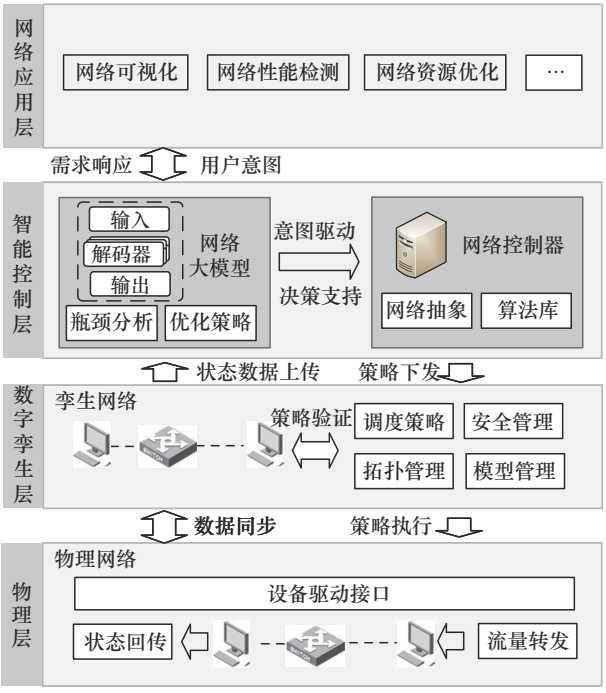

图2　基于LLM的智能网络操作系统架构

2.3　架构设计

本节将分别对这4个关键层进行详细的介绍，以便全面理解每个层级在整个系统架构中的具体作用、功能和相互之间的关系与协作方式，进而了解整个系统的工作原理和优势。

2.3.1　物理层

在智能网络操作系统架构中，物理层是基础，负责承载整个网络的物理连接和数据传输。它包括物理网络单元和节点2个主要组成部分，确保数据在网络中的高效传输和可靠连接。

物理网络单元是网络的实际硬件设备，在数据传输过程中扮演着至关重要的角色。路由器根据转发策略负责数据包的转发，是网络连接的关键设备。节点是网络中的基本连接点，它们负责生成和接收数据，构成了网络的终端用户部分。主要节点包括计算机、服务器、传感器和移动设备。计算机和服务器是网络服务的主要提供者，计算机用于终端用户的日常计算和数据处理，服务器则提供高性能计算和存储资源，处理大量并发请求。

通过物理网络单元和节点的有效结合，物理层不仅确保了网络中数据的高效传输和可靠连接，还构建了设备驱动接口，屏蔽底层异构转发设备的硬件差异，使CNOS能够以统一方式管理和控制各类异构网络设备，为上层的数字孪生层、智能控制层和网络应用层提供了强大的支持。

2.3.2　数字孪生层

数字孪生层是智能网络操作系统架构中的关键部分，利用数字孪生技术为物理网络建立实时虚拟映射，为智能控制提供策略验证环境。数字孪生是一种虚拟模型，精确地模拟了物理网络的结构和行为，提供一个实时的、动态的网络视图，使网络管理更加透明和可控。数字孪生层包括数字孪生模型的创建与维护以及数据的同步与分析，通过对网络设备进行模型抽象，对系统上层提供网络能力的归一化底座。

数字孪生模型是数字孪生层的核心，通过收集物理网络单元和节点的数据，建立与物理网络对应的虚拟模型，并实现低时延的数据同步。创建和维护数字孪生模型的过程如下。

1)数据收集：从物理网络中的各个设备收集状态、性能和配置数据，如网络转发设备的实时负载、传输数据量、接入主机数等运行状态数据。

2)模型创建：基于收集的数据，创建与物理网络结构和行为相对应的虚拟模型。这涉及网络拓扑

的映射、设备属性的仿真以及网络运行状态的虚拟再现。通过高级仿真工具和算法，确保数字孪生模型能够准确反映物理网络的实际情况。

数字孪生层不仅建立、管理虚拟模型，还负责对策略进行验证等，其主要功能如下。

1)策略调度：负责根据实时网络状态，基于预设规则或智能算法，动态地调整物理层的资源分配以优化网络性能、提高资源利用率并满足不同业务的需求。例如，使用预测算法提前预见网络中的异常情况和未来趋势，识别网络中的故障点和异常行为，提供预防性维护和优化建议。在数字孪生层中，策略调度能够在数字空间中模拟并预测不同策略对物理层的实际影响。在验证策略的有效性后再下发给物理层执行，保证系统的安全性和稳定性。

2)安全管理：负责保障数字孪生网络和物理网络的安全，防止未经授权的访问、恶意攻击和数据泄露。通过在数字空间模拟，如分布式拒绝服务攻击（DDoS, distributed denial of service）攻击以及利用系统漏洞等危害系统安全的性能，评估网络安全风险。通过监控物理网络的安全状态，并实时同步到数字孪生体，支持入侵检测以及验证访问控制策略的有效性。

3)拓扑管理：负责维护和更新网络拓扑信息，包括网络设备的连接关系、设备属性、链路状态等。在数字孪生层中，拓扑管理提供了网络拓扑虚拟视图，方便网络管理员进行监控、分析和故障排除，是系统可视化的数据来源。

4)模型管理：负责创建、存储、维护和更新数字孪生体，确保数字孪生模型与物理网络的状态保持一致。持续的数据更新使物理网络和数字孪生模型之间实现实时同步，为网络状态分析和实时决策提供数据支撑。

通过数字孪生模型和数据同步与分析等技术的有机结合，数字孪生层向下能够实现对物理网络的全面监控，向上能够实现策略的仿真验证和网络管理，最终显著提升网络的整体性能和稳定性。

2.3.3　智能控制层

智能控制层是智能网络操作系统架构中的“大脑”，集成了网络管理、控制、监控、分析等核心能力，通过开放框架层对上层应用服务开放。网络大模型作为智能控制层的核心模块，提供智能化的网络管理接口。网络控制器组件继承了SDN控制与转发分离的特点，不仅将底层网络设备抽象为虚拟资源池，还部署快速可插拔的算法，实现业务的快速部署和调整。

基座模型采用LLama架构，为了使模型学习到更深层次的模式和依赖关系，使用Transformer的解码器部分进行N次叠加，如图3所示。

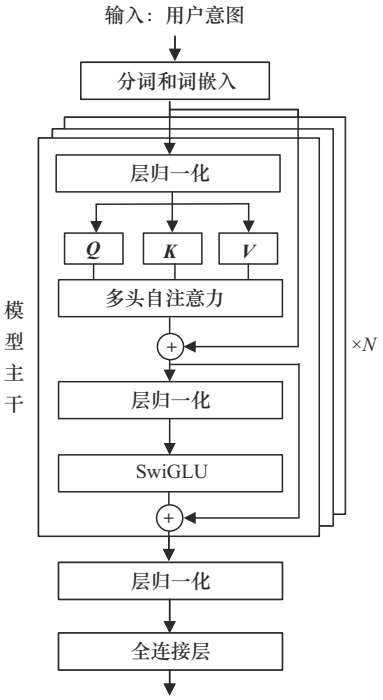

图3　基座模型结构

为了使模型训练过程更加稳定，模型主干采用残差连接，在层归一化中采用RMS归一化函数。针对输入向量$\boldsymbol{a}$，RMS计算式可表示为

$$\mathrm{RMS}(\boldsymbol{a})=\sqrt{\frac{1}{n}\sum_{i=1}^{n}a_i^2} \tag{1}$$

$$\overline{a}_i=\frac{a_i}{\mathrm{RMS}(\boldsymbol{a})} \tag{2}$$

基座模型中使用多头注意力机制，即并行多个自注意力以提升序列数据的处理能力。输入的$\boldsymbol{Q}$、$\boldsymbol{K}$、$\boldsymbol{V}$分别代表Query、Key和Value。$\boldsymbol{Q}$用于寻找输入序列中元素之间的关联性，$\boldsymbol{K}$用于匹配查询的元素，$\boldsymbol{V}$表示该元素实际的信息或内容。当$\boldsymbol{Q}$与$\boldsymbol{K}$的匹配度高时，对应的$\boldsymbol{V}$值就会被更加关注和加权。用SwiGLU激活函数取代ReLU，该激活函数结合了Swish激活函数和门控线性单元（GLU,

gated linear unit)。在位置编码上使用旋转位置嵌入(RoPE, rotary positional embedding),通过绝对位置编码的方式实现相对位置编码。

在从网络应用层到数字孪生层的下行方向上,网络大模型接收用户以意图为导向的自然语言输入,通过理解和解析用户的自然语言命令,将其转换为具体的网络管理操作,由网络控制器向下执行指令或选择算法优化网络性能。这大大简化了网络管理的复杂性,使非专业技术人员也能够轻松管理和操作网络。其主要功能包括以下几点。

1)状态查询:通过整合来自各节点和设备的数据,智能控制层能够实时监控网络的整体性能和各个部分的运行状态,集中管理和协调网络资源,提供全局的网络状态视图,帮助管理员了解整个网络的运行状况。

2)流量调度:智能控制层中部署流量和资源调度模型,具有链路带宽利用率不超限策略、全局带宽利用率最小策略以及基于时延最优策略等;可根据网络状态和任务需求,动态分配和调度网络资源,智能调整带宽分配、流量路径和优先级,优化网络性能,确保端到端的传输时延和网络资源的高利用率。

在从数字孪生层到网络应用层的上行方向上,网络大模型接收网络状态数据。用户可根据需求,一方面接收原始数据,常用于管理员读取指令执行结果;另一方面利用大模型的归纳能力,将网络实时状态以及系统反馈结果进行重点概括,以更友好的形式呈现。

通过网络控制策略和算法与大模型的有机结合,智能控制层实现了对网络资源的全面管理和智能优化,显著提升了网络的整体性能和稳定性。

2.3.4 网络应用层

网络应用层是智能网络操作系统架构的最上层,提供全场景、灵活定制的网络能力,并基于ServiceMesh的微服务架构,采用容器化部署满足网络按需服务的愿景,主要包括网络可视化、网络性能检测和网络资源优化等上层功能。

1)网络可视化通过获取网络拓扑和状态数据,旨在以图形化的方式呈现复杂的网络结构、设备状态以及其他关键的网络信息,使用户可以实时掌握网络运行情况,帮助识别流量瓶颈、异常流量和潜在的安全威胁。

2)网络性能检测通过监控状态数据,分析性能指标,利用数字孪生层提供的数据和智能控制层的网络大模型,实现网络在传输数据时效率和质量的快速评估,提升网络管理效率。

3)网络资源优化旨在通过实时感知网络状态和流量变化,利用智能算法动态调整网络配置,从而消除性能瓶颈。在网络操作系统中,性能优化算法可以根据服务器的负载情况动态调整资源分配,确保关键业务的高效运行和资源的最佳利用。流量预测算法利用历史数据和机器学习模型,通过分析网络流量的周期性和突发性变化,预测未来的网络流量趋势,提供精确的流量预测,为提前进行资源规划和调整提供数据基础,避免网络拥堵和性能下降。例如,针对视频流媒体服务,流量预测可以预测高峰时段的用户访问量,提前调整服务器和网络资源,确保终端节点能够获得稳定的数据流。此外,故障检测算法通过数据分析和模式识别,实时检测异常行为和潜在故障点,提前预警或自动处理。

通过这4层的协同工作,智能网络操作系统架构实现了对大规模网络的全面、智能化管理,显著提升了网络的整体性能和稳定性,满足了现代网络环境对高可控性、灵活性和智能化管理的需求。

3 LLM微调和推理流程

为了增强LLM在网络领域中的实际应用能力,本文设计了一套动态微调和推理流程。微调流程让模型能够不断适应新的操作需求和场景变化;推理流程则保证了系统能够实时、高效地执行用户的自然语言指令。

3.1 LLM微调流程

微调旨在利用特定领域的数据对预训练模型进行再训练,使其在特定应用场景下的表现更为出色。为了使LLM更好地适应所描述的智能网络操作系统架构及其特定的网络操作系统任务,本文设计了一种与系统结合的动态微调流程,通过结合用户的需求记录与网络控制器的操作记录,在微调阶段对模型进行精细调整。如图4所示,该流程通过在大模型使用过程中以及系统管理操作时同步记录输入需求与操作命令,形成微调数据集,并定期利用该数据集进行模型训练,从而不断提升模型的性能和准确性。微调流程具体如下。

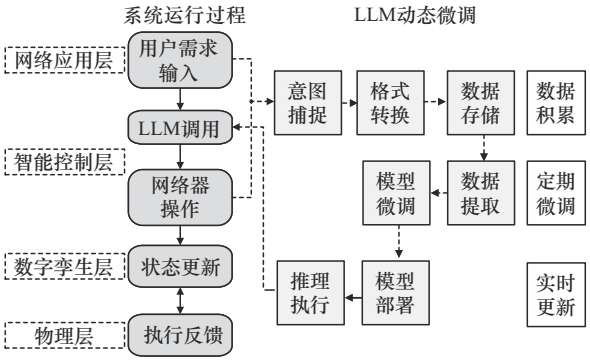

图4　LLM动态微调流程

1)在数据积累阶段，LLM在系统实际使用过程中对用户通过自然语言输入的指令以及网络控制器的操作进行记录。LLM将捕捉到用户的操作意图，并将其转换为内部可处理的格式。这一步的关键在于准确识别用户的意图，以确保后续操作的正确性和有效性。接下来，LLM将相应的操作命令记录下来，包括命令的参数和执行结果。最后，LLM将捕捉到的意图和记录的命令存储在数据库中，记录这些捕捉到的用户意图和相应的操作命令，并将记录的数据存储在数据库中，形成微调数据集。这些步骤确保了LLM能够积累大量的操作数据，为后续的微调提供充足的训练数据。通过这一完整过程，LLM能够自动生成用于微调的数据集。

2)为了确保能够持续适应新的操作需求和使用场景，LLM每隔一段时间会进行一次模型微调，即定期微调。LLM会定期从数据库中提取记录的意图和命令，形成最新的微调数据集。利用这些最新的数据集，对预训练后的LLM进行再训练。这一步骤的核心在于通过微调模型，使LLM能够更好地适应和处理特定网络操作系统任务。通过不断地进行数据提取和模型微调，LLM能够持续优化性能和准确性，以应对新的操作需求和使用场景。

3)在完成模型微调后，系统将微调后的模型更新并部署到实际操作环境中。这一步骤确保了最新的模型能够在实际操作系统中运行，并能够有效地处理用户输入的自然语言指令。

整个微调的设计流程形成了从数据收集、模型训练到实际部署的完整闭环，让LLM在网络操作系统中的应用具有更高的执行效率和准确性。

3.2　LLM推理流程

为了使LLM在实际的网络操作系统中发挥最佳性能，微调完成后，本文将其应用于推理任务。为确保LLM能够准确理解并执行用户的自然语言指令，推理流程被设计为包含多个步骤，如图5所示。推理流程具体如下。

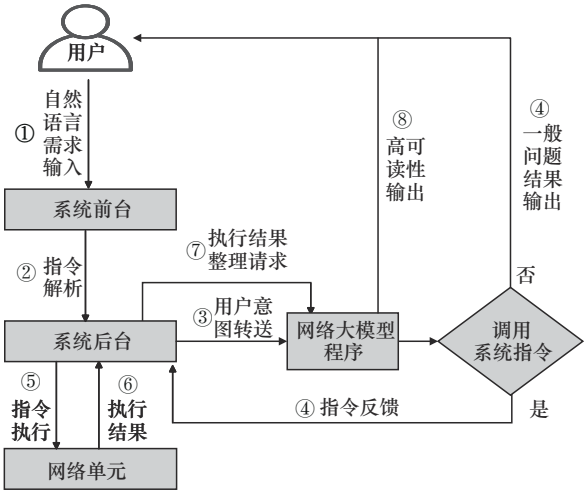

图5　LLM推理流程

用户通过命令行界面或应用程序编程接口以自然语言的形式输入指令请求意图。系统前台接收用户的自然语言输入后，将其传递到系统后台。系统后台解析用户指令，将其转换为网络大模型程序，即LLM可接受的格式，并发送给LLM进行处理。若指令为一般性问题，不需要调用系统指令，则LLM直接生成结果，并返回给用户，无须经过指令执行的处理；若涉及系统指令调用，LLM解析意图并生成系统指令，反馈至系统后台，系统后台接收到LLM生成的系统指令后，调用相应的功能模块执行这些命令；相应的功能模块接收到命令后，执行具体的操作，并生成执行结果。执行结果返回后，系统后台将其传递给LLM进行归纳总结，确保结果具备高可读性，最终将处理后的结果返回给用户，确保用户能直观理解执行效果。

整个流程通过系统前后台的紧密协作与LLM的智能处理，将用户的自然语言指令高效转化为具体的系统操作命令，并确保执行后的结果能够以简洁明了的形式反馈给用户，从而提升操作的精准度和用户体验。

4　实验仿真

本节将通过实验验证所提出的大模型在生成网络操作系统指令方面的性能。实验所用操作系统为

ubuntu 22.04.3 LTS，系统搭载的CPU为Intel(R) Xeon(R) Gold 6226R CPU @ 2.90 GHz，GPU为NVIDIA Tesla V100S PCIe 32 GB。测试集来源于行业内专家整理的多种典型网络操作指令集合，并依据指令的功能特性划分为36类种场景，如表1所示。

这些指令涵盖了网络操作系统中常见的操作，如网络配置、资源管理、流量监控等。实验内容包括2个方面：一是评估微调后的LLM在识别和转换用户指令方面的准确性，即其能否将用户输入的自然语言指令准确地转换成可执行的系统指令；二是评估指令响应时间，证明LLM的应用能有效缩短操作系统管理任务的完成时间，从而提升用户操作的效率。

4.1 LLM识别指令准确度

微调后的LLM的一个关键性能指标是其在识别和转换用户指令方面的准确度。为了验证这一点，本文设计了一组实验，将LLM接收到指令后生成的相应输出与预期的输出进行对比，以计算模型的识别准确度。

ROUGE-L F1分数可以通过结合精确度与召回率，在LLM生成与目标文本完全匹配时给出更准确的分数，能够更全面地反映模型输出与期望输出之间的相似度。因此本文选择ROUGE-L F1分数来评估实验中LLM对指令的识别准确度。

实验过程中，本文先选择了5种具有代表性的场景来进行分析。这些场景包括虚网管理、链路带宽管理、网络监控总览、告警监控与管理以及网络设备管理等，分别包含了对应场景下200条不同的指令集合。为了更全面地评估模型的性能，本文还在涵盖36种场景下的3 000条指令上进行了测试。

图6展示了网络大模型在不同场景下的ROUGE-L F1分数。其中，黑色点代表每个单独样本的得分，灰色实线则表示所有样本的平均分数。虚网管理场景和网络监控总览场景的平均分数分别达到了0.996 1和0.990 1，表明模型对该类结构指令具有稳定的理解与生成能力，整体错误率较低，尽管存在少量离散样本，但总体波动可控；告警监控与管理场景的平均分数几乎接近满分，达到了至少4个9的级别，模型在处理相关指令时展现出极强的泛化能力和指令理解能力；链路带宽管理场景的表现与网络设备管理场景相似，平均分数分别为0.999 2和0.999 3，反映模型在应对该类操作性强的指令时处理效果高度一致，适应能力强，且仅存在极少数偏离点。在全部场景的实验中，模型的平均分数为0.989 0，略低于前述各单一场景，任务类型混合带来的上下文语义复杂性上升导致出现更多偏离均线的样本，但整体仍保持较高的准确性。各场景模型均展现出高准确度，绝大多数样本得分集中在0.98以上，个别样本虽有波动但未出现严重异常，体现出模型良好的稳定性与泛化能力。整体而言，微调后的LLM在处理网络操作管理指令时展现了较高的识别准确度和可靠性，可以有效地辅助用户完成网络系统管理任务，并确保生成指令的正确性。

4.2 LLM生成指令执行成功率

在指令准确度评估基础上，本文进一步引入了指令执行成功率指标，旨在衡量LLM生成指令是否能够在实际系统中被正确解析并成功完成指定操作。在该评估中，本文采用语义一致性优先的判定标准：若生成指令语法正确、系统成功执行，且执行结果与用户意图相符，即视为执行成功；若虽然系统返回了执行反馈，但因关键参数（如设备名称、接口标识、IP地址等）未正确替换，导致功能偏离原始意图，则判定为语义性失败，不计入成功统计。

表1 网络管理场景

核心模块	关键场景
物理管理	物理网络管理，节点管理，网络设备管理，链路带宽管理，路由管理，虚拟网络管理，L2VPN业务管理，确定性业务管理，接入点管理，Srv6 Policy路径管理
模版管理	color模板管理，流分类模板管理，Policy业务模板管理，DSCP模板管理
监控分析	系统监控，网络监控总览，物理网络设备流量监控，物理网络设备利用率监控，物理网络链路流量监控，物理网络链路利用率监控，物理网络链路时延监控，物理网络链路抖动监控，物理网络链路丢包率监控，物理网络故障监控，虚拟网络业务流量监控，虚拟网络接入点流量监控，虚拟网络路径流量监控，虚拟网络路径时延监控，虚拟网络路径抖动监控，虚拟网络路径丢包率监控，网络质量呈现
运维配置	告警监控，告警管理，系统配置管理，日志管理，租户管理

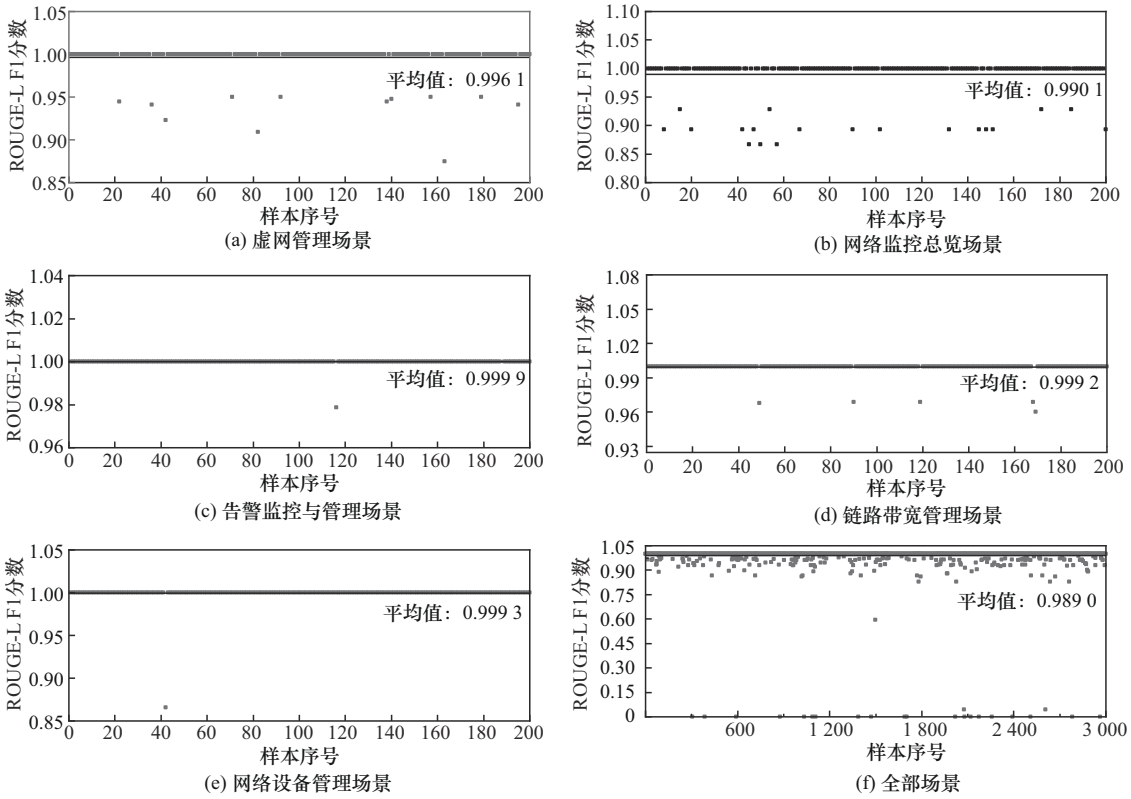

图6　网络大模型在不同场景下的ROUGE-L F1分数

为保持指标适用性与任务代表性，本文在前述5个典型场景的样本下进行成功率测试。为提高测试效率，本文仅对模型生成结果与标准指令存在差异的样本进行了系统执行验证，对于生成结果与标准指令完全一致的样本，基于其结构与语义的全匹配特性，直接判定为执行成功，未重复测试其系统可执行性。图7展示了5个典型任务场景下的执行成功率。整体结果显示，系统在多数任务场景下均具备较高的执行正确率，平均成功率超过95%，说明大模型在多数场景中能够生成符合系统规范并能正确触发目标功能的控制指令。为进一步分析执行失败的样本，本文对不同场景下的错误类型进行了归类。在虚网管理场景中，主要任务为业务查询、路径配置等操作，执行失败原因集中于路径参数未被正确替换，系统无法正确识别对象。在网络监控总览场景中，LLM需根据用户指令生成带时间范围的状态查询请求，失败原因主要表现为时间值参数未被准确解析或替换；时间单位格式不一致，在部分接口中引发系统识别失败。链路带宽管理场景中的执行成功率相对较高，少数失败的样本与虚网管理场景类似，原因为链路路径信息未能正确填充。告警监控与管理和网络设备管理场景中达到最高成功率，皆仅有单个样本出现参数未正确替换的情况。综合分析可见，当前模型绝大部分任务指令生成效果稳定，仅在小部分依赖复杂参数填充或需理解时间等上下文信息的指令中存在一定偏差，整体表现说明模型在当前系统框架下已具备较强的任务适应能力与指令生成实用性。

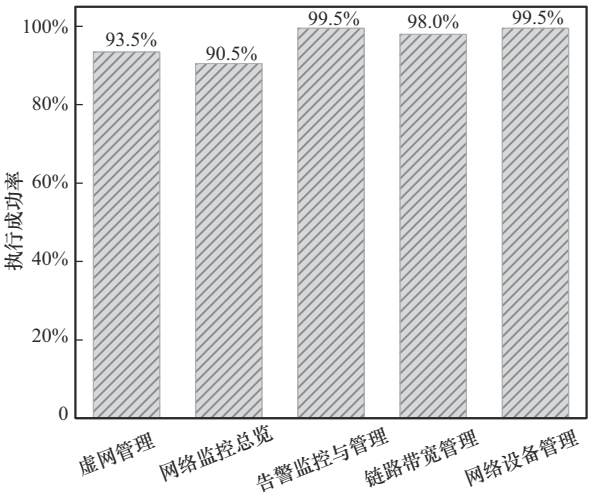

图7　5个典型任务场景下的执行成功率

此外，在部分与标准输出不完全一致的指令中，本文观察到2类常见的格式性偏差：一类为参数间存在空格冗余问题，测试过程中验证发现此类偏差不会影响CNOS对指令的正常解析与执行，可视为可容忍冗余；另一类表现为URL协议类型（http/https）不一致问题，但根据对接的CNOS实际运行机制，此类协议前缀在实际调用阶段会根据系统配置自动适配，不会影响指令的功能执行与最终效果。因此，虽然部分样本在文本结构上未与标准输出完全一致，但在系统具备一定容错能力的前提下，此类非关键性偏差可被有效消化。这表明，即便LLM的生成结果存在轻微格式性偏差，系统依然能够保障功能执行的准确性与完整性。

4.3　基于LLM的网络操作系统响应时间

操作系统的实时性依赖于“指令生成”与“指令执行”2个阶段的流程效率，其中，指令生成指的是LLM解析用户意图并输出可执行指令的耗时，反映了模型的处理效率；指令执行表现为系统接口完成指令解析与功能反馈的耗时，体现了系统后台的处理能力。因此为全面验证所提系统在实际场景中的响应效率，本文将实验划分为2个阶段进行评估。

首先，本文评估了LLM在网络操作系统中生成控制指令的耗时，记录了在测试集下LLM的指令生成时间，结果如图8所示。其中，折线表示不同场景下各个样本响应用户输入意图的指令生成时间，灰色实线则代表该场景指令生成时间的平均值。实验结果显示，模型对于不同场景的任务需求都表现出良好的推理速度，能够较快地完成指令生成。在虚网管理场景中，样本的生成时间稳定，集中在1~2 s，平均生成时间为1.463 5 s，表明模型在处理格式较为标准的指令时具备较高的效率；在网络监控总览场景中，由于涉及更多的数据分析和信息整合，生成时间分布的波动性增大，平均生成时间为1.700 9 s，略高于虚网管理场景；在告警监控与管理、链路带宽管理场景和网络设备管理场景等复杂度更高的场景中，尽管生成时间略有增加，但依然保持在较快范围内，时间分布相对均衡，这些场景通常涉及跨层级的数据交互、实时分析与管理等，因而生成时间相较于前述场景有所上升。总体而言，模型在不同场景下的指令生成时间均维持在合理快速的范围内，表现出较高的处理效率和适应能力。对于包含所有场景的整体测试，指令生成时间为3.563 1 s，虽然较单一场景有所增加，但考虑

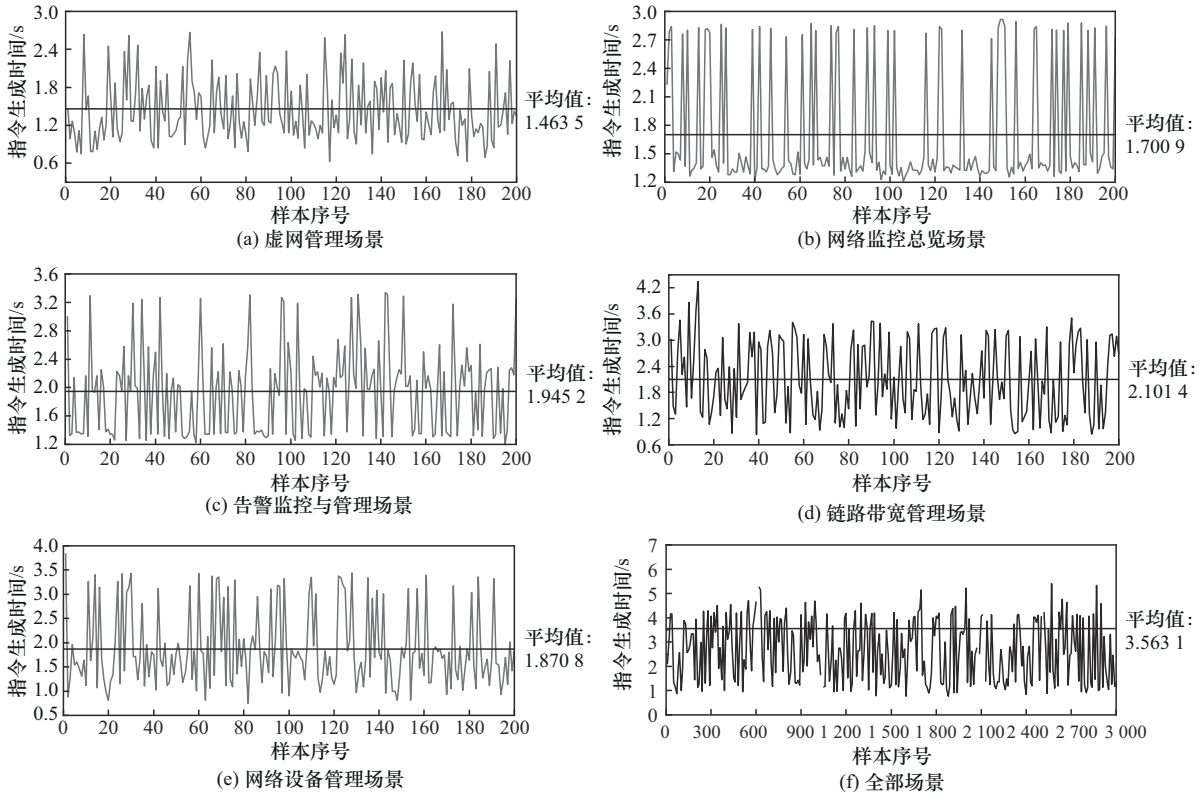

图8　网络大模型在不同场景下的指令生成时间

到测试过程中涉及的多种复杂场景叠加，模型需要在不同任务间进行上下文切换，且与当前主流的LLM相比，如百度自研的轻量级大语言模型ERNIE-Lite-8K-0308，在类似的文本任务中，模型推理时间也通常在3.00~6.26 s，因此3.563 1 s的生成时间符合现有技术水平，进一步验证了其在处理多样化网络管理任务指令时的稳定性和高效性。

其次，针对指令执行时间的实验中，本文从各场景测试样本随机抽取部分数据（共45个样本）进行接口指令执行时间测试。在不同任务场景中对实际CNOS接口的调用结果显示，接口执行时间普遍较短，整体分布范围为9~263 ms，其中大多数接口在100 ms以内即可完成指令解析与反馈，相较于模型生成指令平均耗时，接口执行时间所占比重极小，表明当前系统端到端响应过程的主要耗时集中在大模型推理阶段。该结果进一步验证了LLM输出结果在系统端具备良好的可解析性与执行效率，说明提出的网络操作系统具备对模型生成指令的快速执行能力，能够在实际场景中实现高效联动。

综上所述，微调后的LLM在各类场景中表现出良好的指令生成效率，尽管不同场景下的指令生成时间存在一定的差异，但总体时间维持在较快水平，具备较强的任务适应能力与响应稳定性，在网络管理意图的实时解析中具备可行性；同时，系统接口执行时间显著低于指令生成时间，能够有效支撑指令的快速处理与反馈。两阶段协同形成的端到端处理链路具备良好的整体响应性能，能够满足大多数网络管理任务的实时处理要求。

5　结束语

随着终端设备和网络服务的大量部署，网络环境愈发复杂。当前的网络操作系统存在智能化程度低、业务响应慢、配置复杂等方面的问题，因此，本文提出了一种基于LLM的高可控大网级网络操作系统架构。该系统架构以CNOS为依托，自下而上分为物理层、数字孪生层、智能控制层和网络应用层。通过在智能控制层部署经系统指令微调的LLM，实现了从执行业务指令到理解并执行用户意图的转变。本文设计了LLM动态微调流程和推理流程，为网络智能化管理运维提供了强大支持。实验结果表明，本文所提系统架构能够准确地将用户意图转换成一系列系统控制指令，并将执行结果归纳反馈。经过微调的网络大模型在多个场景中取得了较好的ROUGE-L F1分数，生成指令的实际执行平均成功率超过95%，且针对典型单一场景和全部网络管理场景，指令生成平均时间最低为1.463 5 s，最高为3.563 1 s，执行耗时普遍在100 ms以内，表现出对业务需求的快速响应，能够满足网络运维和管理需求。

未来将进一步探索大模型与小模型的协同调度机制，兼顾生成效果和响应效率，并引入更加规范化的技术框架，以提升系统的集成能力与异构工具之间的互操作性。

参考文献：

[1] WANG Y T, PAN Y H, YAN M, et al. A survey on ChatGPT: AI-generated contents, challenges, and solutions[J]. IEEE Open Journal of the Computer Society, 2023, 4: 280-302.

[2] WU J Y, GAN W S, CHEN Z F, et al. AI-generated content (AIGC): a survey[J]. arXiv Preprint, arXiv: 2304.06632, 2023.

[3] WEI J, WANG X Z, SCHUURMANS D, et al. Chain-of-thought prompting elicits reasoning in large language models[J]. Advances in Neural Information Processing Systems, 2022, 35: 24824-24837.

[4] LONG O, WU J, XU J, et al. Training language models to follow instructions with human feedback[J]. Advances in Neural Information Processing Systems, 2022, 35: 27730-27744.

[5] 黄韬, 刘江, 汪硕, 等. 未来网络技术与发展趋势综述[J]. 通信学报, 2021, 42(1): 130-150.
HUANG T, LIU J, WANG S, et al. Survey of the future network technology and trend[J]. Journal on Communications, 2021, 42(1): 130-150.

[6] 黄韬, 谈沙, 谢人超, 等. 网络操作系统的研究进展与展望[J]. 北京邮电大学学报, 2024, 47(2): 1-10.
HUANG T, TAN S, XIE R C, et al. A review of and prospects for network operating system research[J]. Journal of Beijing University of Posts and Telecommunications, 2024, 47(2): 1-10.

[7] KHALIDI Y. SONiC: the networking switch software that powers the microsoft global cloud[R]. 2017.

[8] CHOI S, BURKOV B, ECKERT A, et al. FBOSS: building switch software at scale[C]//Proceedings of the 2018 Conference of the ACM Special Interest Group on Data Communication. New York: ACM Press, 2018: 342-356.

[9] RAIAAN M A K, MUKTA M S H, FATEMA K, et al. A review on large language models: architectures, applications, taxonomies, open issues and challenges[J]. IEEE Access, 2024, 12: 26839-26874.

[10] WANG J J, HUANG Y C, CHEN C Y, et al. Software testing with large language models: survey, landscape, and vision[J]. IEEE Transactions on Software Engineering, 2024, 50(4): 911-936.

[11] BROWN T, MANN B, RYDER N, et al. Language models are few-shot learners[J]. Advances in Neural Information Processing Systems, 2020, 33: 1877-1901.

[12] ZHANG S S, ROLLER S, GOYAL N, et al. OPT: open pre-trained transformer language models[J]. arxiv Preprint, arxiv: 2205.01068, 2022.

[13] AAKANKSHA C, SHARAN N, JACOB D, et al. PaLM: scaling language modeling with pathways[J]. Journal of Machine Learning Research, 2023, 24: 1-113.
[14] CORONADO E, BEHRAVESH R, SUBRAMANYA T, et al. Zero touch management: a survey of network automation solutions for 5G and 6G networks[J]. IEEE Communications Surveys & Tutorials, 2022, 24(4): 2535-2578.
[15] BARIAH L, ZOU H, ZHAO Q Y, et al. Understanding telecom language through large language models[C]//Proceedings of the GLOBECOM 2023 - 2023 IEEE Global Communications Conference. Piscataway: IEEE Press, 2023: 6542-6547.
[16] KHOLGH D K, KOSTAKOS P. PAC-GPT: a novel approach to generating synthetic network traffic with GPT-3[J]. IEEE Access, 2023, 11: 114936-114951.
[17] GUPTA M, AKIRI C, ARYAL K, et al. From ChatGPT to ThreatGPT: impact of generative AI in cybersecurity and privacy[J]. IEEE Access, 2023, 11: 80218-80245.
[18] SHARMA P, YEGNESWARAN V. PROSPER: extracting protocol specifications using large language models[C]//Proceedings of the 22nd ACM Workshop on Hot Topics in Networks. New York: ACM Press, 2023: 41-47.
[19] CHATZISTEFANIDIS I, LEONE A, NIKAEIN N. Maestro: LLM-driven collaborative automation of intent-based 6G networks[J]. IEEE Networking Letters, 2024, 6(4): 227-231.
[20] LIRA O G, CAICEDO O M, DA FONSECA N L S. Large language models for zero touch network configuration management[J]. IEEE Communications Magazine, 2024: 1-8.
[21] MEKRACHE A, KSENTINI A, VERIKOUKIS C. Intent-based management of next-generation networks: an LLM-centric approach[J]. IEEE Network, 2024, 38(5): 29-36.
[22] HUANG Y D, XU M R, ZHANG X Y, et al. AI-generated network design: a diffusion model-based learning approach[J]. IEEE Network, 2024, 38(3): 202-209.

[作者简介]

霍如（1988– ），女，黑龙江哈尔滨人，博士，北京工业大学副教授，主要研究方向为未来网络、网络智能化、区块链、资源调度等。

沙宗轩（1990– ），男，回族，安徽蚌埠人，北京工业大学博士生，主要研究方向为未来网络、网络人工智能、深度学习、强化学习等。

吕科呈（2001– ），男，广西玉林人，北京工业大学硕士生，主要研究方向为未来网络、车联网、资源调度等。

陈伟（1996– ），男，河北三河人，北京邮电大学博士生，主要研究方向为边缘计算、数字孪生、区块链等。

汪硕（1991– ），男，河南灵宝人，博士，北京邮电大学副教授，主要研究方向为数据中心网络、软件定义网络、网络流量调度等。

黄韬（1980– ），男，重庆人，博士，北京邮电大学教授，主要研究方向为未来网络体系架构、软件定义网络、网络虚拟化等。

F. Richard Yu（1974– ），男，加拿大工程院院士，卡尔顿大学教授，主要研究方向为互联网自主智能、自动驾驶、网络空间安全等。