中文核心期刊　中国科技核心期刊
第三届国家期刊奖百种重点期刊　信息通信领域产学研合作特色期刊

ISSN 1009-6868
CN 34-1228/TN

中兴通讯技术

ZTE TECHNOLOGY JOURNAL

http://tech.zte.com.cn

第 30 卷 · 总第 175 期 · 2024 年 4 月 · 第 2 期

专题：网络大模型

9 771009 686243 04>

《中兴通讯技术》第 9 届编辑委员会成员名单

目　次

中兴通讯技术（ZHONGXING TONGXUN JISHU）
第 30 卷　总第 175 期　2024 年 4 月　第 2 期

中文核心期刊 中国科技核心期刊 第三届国家期刊奖百种重点期刊 信息通信领域产学研合作特色期刊 中国知网、万方数据、重庆维普等数据库收录期刊 1995 年创刊

《中兴通讯技术》2024 年热点专题名称及策划人

1. 下一代多址接入技术
北京交通大学教授 艾渤
北京交通大学教授 陈为

2. 网络大模型
中国电信 IP 领域首席专家 唐宏
中兴通讯无线首席架构师 熊先奎

3. 6G 多天线技术
东南大学首席教授 金石
北京交通大学教授 章嘉懿
东南大学副研究员 韩瑜

4. 6G 无线系统技术
中国信息通信研究院副院长 王志勤
中国移动研究院院长 黄宇红
东南大学教授 王东明

5. 卫星通信技术
哈尔滨工业大学（深圳）教授 张钦宇

6. 数据通信新技术
中国电信研究院教授级高工 解冲锋
中国联通研究院首席科学家 唐雄燕

MAIN CONTENTS

ZTE TECHNOLOGY JOURNAL
Vol. 30 No. 2 Apr. 2024

Special Topic ▶

Network Large Models

Expert Forum ▶

Enterprise View ▶

Research Paper ▶

期刊基本参数：CN 34-1228/TN*1995*b*16*66*zh*P*¥20.00*6500*14*2024-04

网络大模型专题导读

专题策划人

唐宏，中国电信股份有限公司研究院IP领域首席专家，正高级工程师，中国电信科技委常委；长期从事IP网络及其新技术的研发工作；发表论文30余篇，获发明专利100余项。

熊先奎，中兴通讯股份有限公司无线首席架构师、“智算”技术委员会前瞻组组长；长期从事计算系统和体系结构、先进计算范式以及异构计算加速器研究工作；曾主导中兴通讯ATCA先进电信计算平台、服务器存储平台、智能网卡和AI加速器等系统架构设计。

OpenAI公司基于“Scaling Law”哲学，通过堆叠算力，使用模块化易于扩展且擅长捕捉长距离依赖关系的Transformer神经网络，构建了自回归、生成式的千亿以上参数量大模型，在多个人工智能（AI）领域取得了SOTA效果。目前，OpenAI正在通过图像、视频等一系列多模态学习，力图实现“World Model”“World Simulator”，以建立物理世界映射。它的实战成果给全世界AI从业者提供了一种目前看来行之有效且朝向通用人工智能（AGI）发展的方法论和技术路线。

通信网络在向6G演进的过程中，无论是物理层AI和数字孪生，还是网络的智能运维和应用层的内容生成，都离不开AI和大模型。大模型如何与通信网络结合并发挥关键作用？过程中将面临哪些技术挑战？对应的解决方案是什么？这些方案的当前研究进展如何？为此，本期以网络大模型为主题，共收录了9篇文章，针对网络大模型面临的关键挑战与核心问题展开讨论。

AI和大模型都离不开算力，而算力生态系统的构建离不开核心基础软件。本期中，我们很荣幸邀请到清华计算机系郑纬民院士撰写《智能算力核心基础系统软件的现状与展望》作为开篇。文章梳理了智能算力平台中的十大核心基础软件，对这些软件的全球现状进行了详细介绍，并探讨了当前中国算力平台上系统软件栈建设的机遇和挑战。《大语言模型算法演进综述》和《大模型训练技术综述》是两篇综述性文章，分别从算法演进和硬件亲和、训练软硬件系统构建和数据处理过程等角度，对当前大模型核心技术的要点进行了系统介绍。《通信网络与大模型的融合与协同》针对AI与通信的双向协同、网络大模型部署两个方面，深入探讨了通信网络大模型研究的主要进展。《基于存算一体集成芯片的大语言模型专用硬件架构》《低资源集群中的大语言模型分布式推理技术》两篇技术性论文则分别从大模型算力高能效实现、大模型集群通信网络优化等领域进行了有意义的方向性探索。《生成式大模型承载网络架构与关键技术探索》从运营商角度，基于大模型时代基础设施的重要性，对确定性网络和远程直接内存访问（RDMA）传输技术做了介绍和思考。《大语言模型时代的智能运维》《大模型知识管理系统》则从大模型应用层面在网络智能运维应用以及知识索引应用方面进行了分析介绍。

本期的作者来自知名高校、科研机构、企业等，针对网络大模型，从大模型算力、模型训练、关键技术挑战等方面介绍了最新研究成果。希望本期内容能为读者提供有益的启示和参考，并在此对所有作者的大力支持和审稿专家的辛勤指导表示由衷的感谢！

DOI：10.12142/ZTETJ.202402001
收稿日期：2024-03-20

智能算力核心基础系统软件的现状与展望

Status and Prospect of Intelligent Computing Core Basic System Software

郑纬民/ZHENG Weimin，翟季冬/ZHAI Jidong，翟明书/ZHAI Mingshu
（清华大学，中国 北京 100084）
(Tsinghua University, Beijing 100084, China)

DOI：10.12142/ZTETJ.202402002
网络出版地址：http://kns.cnki.net/kcms/detail/34.1228.TN.20240408.1037.005.html
网络出版日期：2024-04-09
收稿日期：2024-02-18

摘要：智能算力对中国人工智能技术的进步具有重要意义。发展智能算力平台，做好核心基础系统软件尤其重要。梳理了智能算力平台中的10个核心基础系统软件，对这些软件的全球现状进行了详细介绍，并探讨了当前中国算力平台上系统软件栈建设的机遇和挑战。

关键词：人工智能；智能算力；大模型；系统软件

Abstract: Intelligent computing power is of great significance to the progress of artificial intelligence technology in China. It is crucial to develop intelligent computing power platforms and optimize foundational system software. Ten foundational systems software within intelligent computing power platforms are analyzed, and a detailed overview of their global status is provided, as well as the opportunities and challenges in the construction of system software stacks on Chinese computing power platforms.

Keywords: artificial intelligence; intelligent computing power; large model; system software

引用格式：郑纬民，翟季冬，翟明书. 智能算力核心基础系统软件的现状与展望 [J]. 中兴通讯技术，2024，30(2)：2-8. DOI：10.12142/ZTETJ.202402002

Citation: ZHENG W M, ZHAI J D, ZHAI M S. Status and prospect of intelligent computing core basic system software [J]. ZTE technology journal, 2024, 30(2): 2-8. DOI: 10.12142/ZTETJ.202402002

随着人工智能（AI）技术的飞速发展，深度学习大模型在文本、图像、视频等数据的理解和生成任务方面展现出强大的能力。这些智能模型的诞生，离不开算力的发展。得益于芯片技术的发展，大模型能力通过不断扩大的规模训练实现了大幅提升。

由于大模型规模急剧扩大，智能算力已成为大模型发展的稀缺资源和关键因素。智能算力规模的提升需要在软硬件技术上取得突破，其中增强核心系统软件尤为重要。

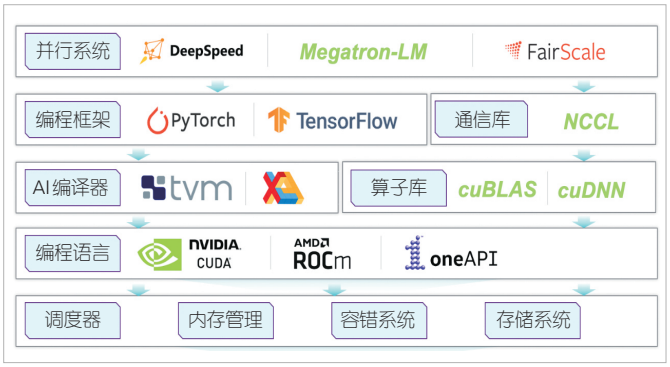
针对智能算力，我们总结出10个核心系统软件，如图1所示。它们在提升智能算力使用效率、降低大模型编程开发难度等方面起到了重要作用。根据功能，这些软件可分为四大类：

1）编程开发软件，包括编程语言和编程框架。它们是开发者和计算系统的交互接口，需要兼顾用户易用性和计算高效性。

2）并行加速软件，包括并行计算系统和通信库。它们使得大模型计算可以充分发挥多机多卡的分布式计算能力。

3）计算加速软件，包括算子库和AI编译器。它们使得大模型的计算负载能在加速卡等硬件上高效执行。

4）基础支撑软件，包括调度、容错、内存分配和存储系统。它们为大模型的易用性、高效性和鲁棒性做出了重要贡献。



▲图1 智能算力的10个核心基础软件

本文中，我们将对这些智能算力核心系统软件做出介绍，并通过回顾全球发展现状，探讨当前中国算力平台上系统软件所面临的机遇和挑战。

1 编程开发软件

大模型的编程开发软件主要有编程语言、编程框架。大模型的开发既需要适应模型的快速迭代，又需要利用各类硬件加速器来运行训练并推理所需的巨量计算。为满足这一需求，用于大模型的编程语言形成了模型开发语言与计算开发语言的分工，后者由前者通过深度学习编程框架来调用。目前主流的模型开发语言为Python语言，在其上实现的现代深度学习编程框架支持自动微分、Python绑定和直观的调试方法等功能，为算法开发者提供了良好的易用性。计算开发语言则由于各异的硬件体系结构呈现出百花齐放的状态。

1.1 编程语言

为了适应各异的硬件体系结构，许多硬件厂商需要使用各自不同的计算开发语言来编写其硬件加速器上的程序。目前，国际上用于大模型的主流硬件是英伟达公司的图形处理器（GPU），主要使用CUDA[1]作为其编程语言。另外，第三方编程语言Triton[2]也被广泛用于编写英伟达GPU上的大模型相关计算。中国具有代表性的硬件公司也采用了不同的编程语言，例如寒武纪的BANGC[3]、华为的AscendC[4]、壁刃的BIRENSUPA[5]、摩尔线程的MUSA[6]。这些编程语言中使用了不同的抽象来描述并行、访存与计算，可面向不同硬件进行针对性优化，但同时这也使不同硬件平台上的软件环境互不兼容。这给开发者灵活利用不同的硬件资源带来了挑战。

为了应对这一挑战，OpenCL[7]、SYCL[8]等编程语言致力于以更广泛的抽象来表达多种不同硬件加速器上的程序。更进一步地，Mojo[9]不仅计划统一多种硬件加速器，还力求统一模型开发语言与计算开发语言，以一套语言完成大模型的全流程开发。但是，统一的语言仍需不同编译器针对具体硬件来逐一实现，例如英特尔公司的oneAPI DPC++[10]就是针对英特尔及英伟达GPU的编译器实现的。下文中我们也将介绍编译器的发展现状。

1.2 编程框架

大模型的复杂度使得直接使用编程语言来开发模型变得十分困难，因此编程框架成为深度学习算法开发者与计算系统交互的界面，对深度学习训练和推理任务的计算性能产生了很大的影响。

Caffe[11]是最早用于深度学习训练任务的框架之一，具有自动微分和GPU支持的功能。它提供了一个带有Python和MATLAB绑定的C++机器学习库，使用内置的应用程序编程接口（API）来定义和训练模型。为了更好地表达各种模型结构和操作，Google开发了TensorFlow[12]框架。TensorFlow将神经网络模型表示为数据流有向无环图（DAG）的框架。TensorFlow已应用于许多领域，包括自然语言处理、计算机视觉、基于物理的AI应用等。然而，在TensorFlow中，用户需要先使用一段代码来描述模型的结构，再使用一段代码来描述训练的过程。这一较为抽象的工作模式给开发和调试工作带来了额外的困难。PyTorch[13]是由Meta开发的神经网络训练和推理框架。相比TensorFlow，PyTorch基于动态计算图，在代码实现上更灵活，调试更容易。作为最用户友好的框架之一，PyTorch在工业界和学术界都被广泛使用。然而，其动态特性使得许多优化（例如内核融合和计算图转换）难以实现。

百度PaddlePaddle[14]和华为MindSpore[15]等新一代框架结合了TensorFlow和PyTorch的特性，通过同时支持静态和动态模式的方法，既保证了易用性，又可以基于编译技术对计算过程进行更多的优化，甚至可以通过自动张量切分等技术实现自动的多卡并行训练。

然而，目前国际上围绕PyTorch平台已形成了丰富的软件生态。例如Megatron-LM[16]、DeepSpeed[17]、HuggingFace[18]等围绕Transformer模型的上层应用软件均基于PyTorch来实现。这些基于PyTorch生态的软件为模型开发和使用者带来了极大的便利，从而进一步鼓励了开发者为这个生态系统开发更多的软件。而Paddle等国产编程框架的用户基数较小，在软件生态上与PyTorch还有着较大的差距。为迎头赶上，厂商需持续带动用户，积极建设中国平台上的开源社区。

2 并行加速软件

近年来，随着大模型规模的扩展，多机多卡协同分布式计算成为大模型的计算范式。并行计算系统及其通信库支撑了大模型的高性能分布式计算。针对大模型训练和推理的计算特征，并行计算系统中的多种并行策略被提出，以充分利用计算资源；通信库则提供了跨机跨卡的通信能力，并力求高效实现这些并行策略所需的复杂通信模式。

2.1 并行计算系统

并行计算系统满足了大模型训练和推理对存储和计算的高需求。大模型的参数量往往超过单个加速卡的内存容量；其训练所需的计算量也远超单卡的计算能力。国际上主流的

并行计算系统包括DeepSpeed[17]、Megatron[16]等。这些系统主要采用了3种并行策略（数据并行、张量并行和流水线并行），并通过它们的灵活组合提高了计算效率。由于英伟达GPU的高计算能力和其通过nvlink[19]高速互连技术实现的高通信能力，Megatron等并行计算系统已经能够在英伟达平台上对GPT、Llama等大模型实现较高的计算效率。

针对中国软硬件的并行计算系统仍有进步空间，目前主要面临两个问题。首先是大模型计算的自动并行难度较大，特别是在涉及3D并行的复杂组合时，需要并行计算系统根据硬件特性选择合适的并行策略组合。目前，主流的自动并行算法面向英伟达硬件特性设计；针对国产硬件的自动并行算法有待提高。另外一个问题是新并行策略的研究。新兴模型和场景，例如混合专家模型、多模态大模型、长序列处理、基于人类反馈的强化学习等，为并行策略的设计带来了新的挑战，需要经典3D并行之外的其他并行策略来支持[20–21]。

综合而言，并行计算系统在处理Transformer类大模型的计算任务上已经相对成熟，尤其在英伟达硬件上表现出色。然而，中国的并行计算系统在易用性和计算效率方面仍有提升空间，需要研究面向国产平台的自动并行算法和新场景下的并行策略，以满足不断发展的科研和工业需求。

2.2 通信库

通信库提供了多卡间的通信能力，是实现并行计算的基础组件。大模型计算对通信库有3点要求：易用、高效、鲁棒。通信库需要根据大模型的训练与推理提供完备的通信模式接口，降低上层使用者的使用难度。通信库提供的通信模式接口应该是高效的，在大规模集群进行大模型训练时，通信往往成为瓶颈，同时通信库需要对大模型的通信模式、底层硬件拓扑做针对性的通信优化。通信库也需要是鲁棒的，在大模型训练与推理中，通信是频繁的，因此系统的鲁棒性很大程度上取决于通信库的鲁棒性。

在国际上，现有的大模型系统常用的通信库是英伟达的NCCL[22]。它提供了常见的集合通信模式，也能在英伟达硬件平台上实现不错的性能，因此受到多数大模型系统的青睐；然而，NCCL通信库在高效性以及鲁棒性上仍有改进空间。国际上的知名公司也针对各硬件平台做出了针对性优化，推出了自己的通信库：如微软的MSCCL[23]、英特尔的OneCCL[24]、AMD的RCCL[25]、Meta的Gloo[26]。

中国算力平台的通信库设计也处在活跃发展期。阿里设计了ACCL[27]，华为设计了HCCL[28]；清华大学的研究团队开发了“八卦炉”系统，在新一代神威超级计算机上探索了针对百万亿参数量大模型训练的通信库设计。综合考虑上层应用的计算负载、底层硬件的性能特征，“八卦炉”系统实现了中国国产平台上的高效通信。

3 计算加速软件

算子库和AI编译器在硬件上高性能实现了大模型所需的基本操作，有效支撑了上述的编程框架、并行计算系统等软件。算子库和AI编译器是大模型训练和推理的基础设施，对于提高计算效率、降低计算能耗具有重要意义。算子库为模型的基本操作奠定了基础，而AI编译器通常为现有算子库无法支持的计算负载生成代码。

3.1 算子库

算子库使得深度学习算法中常用的函数及模型结构的实现更加高效。通过提高底层硬件的利用效率，优化模型计算过程，算子库可以提高大模型的训练和推理效率。不同平台的算子库暴露了相似接口，为上层跨平台深度学习框架的构建提供了便利。

在国际上，最常用的硬件平台为英伟达GPU。英伟达公司在研发高性能GPU的同时，也提供了cutlass[29]、cuBLAS[30]、cuDNN[31]等高性能GPU加速库。这些算子库封装了矩阵乘等常用算子，对上层深度学习框架隐藏了Tensor Core等硬件实现细节，使得各种深度学习框架均可在该平台上实现较优的运行速度。算子库中也提供了基于模板的底层API以简化自定义算子开发。超微公司亦提供了rocBLAS[32]、hipDNN[33]等接口相似的面向超微GPU架构的算子库。

由于近两年美国的芯片限令，中国正在探索硬件的国产化，研发了一系列高性能人工智能芯片。特别是在高性能计算机方面，新一代的神威、天河等超级计算机采用了中国国产的处理器和加速器。这些国产加速器均提供了高性能的算子库，如申威架构的swBLAS、昇腾架构的AOL[34]库等。这些架构为国产硬件与TensorFlow、PyTorch等主流AI框架的兼容提供了基础保障，提升了国产硬件平台的易用性。但由于中国芯片的硬件架构与其他国家的GPU架构有较大差异，不能完全复刻其他国家算子库的开发方式，因此需要使用与国产硬件架构相匹配的算子开发方式以进一步提升算子库性能。

深度学习算子库在大模型的训练和推理中扮演着关键的角色，通过优化底层计算过程，提高了模型性能和效率。国际上已有成熟的深度学习算子库，中国也在通过持续的研发投入不断提升国产硬件的算子库性能。

3.2 AI编译器

AI编译器的主要用途和意义在于自动化地提高模型的执行效率和性能，同时保证优化前后程序的等价性。根据计算图这一通用的程序抽象，AI编译器的主要技术可分为高层次的计算图优化和低层次的算子优化。

目前存在多个被广泛使用的AI编译器系统，例如TVM[35]和XLA[36]。XLA通过静态图编译和优化技术，提供了高性能的执行引擎。它能够将TensorFlow等框架的计算图编译为高度优化的底层代码，利用冗余消除、等价计算图替换和算子融合等技术加速计算。此外，XLA具备内存优化和低精度计算等多种优化，进一步提高硬件加速器的利用效率。TVM提供了一套端到端的编译和优化工具链，旨在加速深度学习模型的推理和训练过程。TVM的核心思想在于针对用户给定的计算逻辑，通过自动调度计算的执行方案，自动化地优化和生成算子的执行代码。这使得TVM能够以相对较小的开发成本为不同硬件平台生成高效的代码，提供了一个跨平台的算子生成工具。

在中国，以PaddlePaddle[14]、Mindspore[15]为代表的通用深度学习框架均集成了大量的AI编译器技术，以提高程序的执行效率。针对中国硬件加速器种类多的现状，目前业界也出现了如InfiniTensor[37]、PowerFusion[38]等一系列新的框架，以提升国产硬件加速器的利用率。InfiniTensor框架探索了基于张量表达式推导的优化技术，尝试在更大的优化空间中发掘新的优化机会。PowerFusion通过细粒度算子拆分的方式，实现了更为高效的算子融合方案。

目前，AI编译器的发展仍然存在较大空间。一方面，针对国产加速器硬件利用率低的问题，加强对国产硬件平台的编译优化支持，提供更广泛的适配能力。另一方面，可以进一步优化编译器的自动优化算法，提高编译器生成的底层代码的效率和性能。此外，加强与中国深度学习框架的集成，提供更便捷的编译和优化工具，也具有重要意义。

4 基础支撑软件

大模型计算具有很高的复杂性，除了上述关键组件，还需要众多基础支撑软件，主要包括调度系统、容错系统、内存分配系统和存储系统。这些系统软件对大模型的易用性和高效性亦有重要意义。

4.1 调度系统

超大规模的模型训练不仅会带来巨大的计算成本，而且需要依赖大型的集群计算系统。在这样的背景下，充分挖掘和利用大规模集群计算能力变得非常关键，而一个高效且稳定的调度系统可以大幅度降低训练成本并显著提升训练效率。具体而言，调度器作为大规模集群系统的核心，主要有以下3个功能：

第一，弹性调度。调度器能够根据大模型训练任务的需要和集群的当前状态，动态分配资源，从而确保训练过程中资源的最优化利用。

第二，资源管理。调度器负责集群各种资源的规划和管理。大模型训练对计算、存储有着极大的需求，因此各种不同的资源，包括CPU、GPU和存储等要进行协同工作。调度器需要确保在有限的资源下最大化利用各个训练任务的性能，避免资源浪费。

第三，队列管理。调度器能够根据设定的优先级对训练任务进行排序，管理多任务的执行队列。调度器结合不同任务的资源需求、优先级高低来进行资源在时间和空间上的分配，从而确保每个训练任务都能获取到足够的资源并及时完成。

在国际上，Kubernetes（K8s）[39]是目前主流大模型训练在大规模集群系统上的调度器。它简化了容器化应用程序的部署、扩展和管理工作，成为了大规模集群管理和调度的主流软件。而在当前大模型兴起的背景下，Kubernetes同样凭借它在大规模系统上的高适用性、GPU等资源的细粒度管理和弹性调度，以及训练任务部署的灵活性等，成为目前大模型训练主要使用的软件。

在中国，华为的ModelArts[20]等平台也提供了高效的人工智能开发环境和强大的集群管理功能。它支持弹性调度，可以灵活地管理包括英伟达GPU和国产加速器在内的多种硬件资源。但是，目前各家调度器仍存在一些尚未解决的问题，例如：大模型训练调度中硬件资源选择和不同并行训练策略的协同、多种训练任务的稳定性，以及效率管理等问题。

总体来说，尽管目前大规模集群调度器已经相对成熟，但为了满足不断发展的大模型训练需求，在大规模系统中更高效地实施调度策略仍然是值得进一步研究的内容。

4.2 容错系统

容错在当今AI领域扮演着至关重要的角色，其主要作用是确保训练和推理过程的稳定性及可靠性。在大规模系统运行过程中，难免出现软硬件故障或异常，有效的容错技术可以确保系统在面对各种异常情况时能够继续正常运行，避免训练终止，从而节约时间和资源，并提高大模型训练的效率。大模型训练的容错技术包括检查点恢复、动态调整学习率、硬件故障处理等，其中最常用的容错技术是基于检查点

的恢复机制。该机制在训练过程中定期保存大模型的中间数据，以便在发生错误或异常时能够快速恢复至上一个检查点。然而，随着当前大模型参数量增大（万亿级），训练所需的集群规模随之扩增（千卡至万卡），训练时间也更长（数星期至数月），训练期间软硬件故障或异常出现的频次增加，高成本的检查点读写开销将不足以支持大规模系统的有效容错。因此，一些研究工作尝试结合并行策略所引入的冗余性，以降低检查点的读写成本，实现低成本容错，为用户提供更加稳定和高效的训练环境。

在国际上，大模型训练的容错技术得到不断地发展和完善，诸如 TensorFlow-Extended[12]、PyTorch（Torch - Elastic）[13]、Horovod[21]、Ray[40]等典型人工智能训练框架均包含了针对分布式训练的容错机制。这些框架大多采用“监控器+任务”的机制对模型训练任务进行监控调度，以快速识别故障或异常的任务，并使用检查点恢复的机制重启异常任务。OpenAI、Google 已经成功在配备成千上万个 GPU 的集群上训练出 ChatGPT、Gemini、Sora 等大模型，足以证明其在大模型训练的大规模分布式容错方面具备丰富经验的。

中国目前在大模型训练容错技术上也具备较为丰富的经验。值得一提的是，“八卦炉”[41]训练框架支持在新一代国产神威超算系统上 10 万节点（57 万个核组、3 700 万核）的大模型训练，该框架同样采用了基于检查点的恢复机制。在如此庞大规模系统上训练时，平均每 1 h 就会出现一次故障。“八卦炉”采用分布式检查点技术将读写检查点的开销控制在 3 min 以内，实现了低成本的有效容错。此外，字节跳动、快手、商汤等公司各自研发的大模型训练框架中均实现了有效的容错机制。中国在大模型训练容错技术方面具备一定的实力和竞争力。

综上所述，大模型训练容错技术是大模型的关键系统软件，可以提高大模型训练和推理过程的效率和可靠性，为人工智能技术的发展和应用奠定更加稳定和可靠的基础。目前全球业界均具备在大规模系统上开展大模型训练的经验，大多采取基于检查点的恢复机制进行容错。随着大模型参数的扩增，训练所需的系统规模随之增加，低成本的大模型容错技术已逐渐成为一项重要的研究主题。结合并行策略、检查点校验算法与硬件容错等方法或将成为降低容错成本的潜在技术方案。

4.3 内存分配系统

大模型的计算需要庞大而复杂的数据支持，这也同时直接对应着硬件内存资源的需求。内存分配因此面临着前所未有的挑战。内存分配指的是在程序执行期间，动态地分配和管理内存资源的过程。这一过程旨在高效利用内存，避免资源浪费，同时保证程序的高效运行。内存管理策略通常包括静态分配和动态分配。静态分配在程序编译时完成，而动态分配则允许程序在运行时根据需要分配内存。例如，以 TensorFlow 为代表的框架，采用的就是静态分配策略，而以 PyTorch 为代表的一系列框架采用的是动态分配策略。相比较来说，静态分配效率高，但框架受限严重，不易用；动态分配效率较低，但框架灵活。当前最为主流的计算框架均基于动态分配。

大模型系统软件的内存分配面临多重挑战。首先是数据规模，大量的数据和模型，需要巨大的内存资源来处理。通常来说，硬件的内存资源决定了模型规模的上限，因此内存资源的利用效率极为重要，也给内存分配带来了挑战。其次是性能，大模型训练需要非常频繁的申请与加速器上内存资源的释放。与此同时，加速器的直接内存的分配时间也各不相同。相比于 CPU 来说，加速器的直接内存需要更多的额外时间。因此，如何设计内存分配器，提高内存分配过程的速度，尽量少地直接分配硬件内存资源，都给大模型应用带来了巨大的挑战。

为了应对这些挑战，研究人员针对大型模型应用开发了一系列内存分配策略和优化技术。通过利用内存池技术预先分配内存块，有效减少了内存碎片和直接分配的时间开销。例如，清华大学的研究团队开发的 swAlloc[42]内存分配器，专门针对神威芯片的特性和大模型应用需求，设计了特殊的内存分配策略。这一策略解决了在神威系统上内存分配效率和速度的核心问题，为新一代神威超级计算机支撑的大规模训练系统“八卦炉”奠定了坚实的基础。

此外，在涉及多台机器的大规模应用场景中，内存资源的高效分配和利用显得尤为关键，同时也蕴含着巨大的优化潜力。特别是面对混合专家（MoE）模型等复杂的新型模型结构时，如 DeepSpeed[17]所采用的 ZERO 优化器以及“八卦炉”[41]使用的 PARO 优化器，均在减少模型内存需求和提升训练效率方面展示了能力。在这种复杂环境下，进一步提高内存资源的使用效率，成为了大模型系统研究中最为关键的研究内容之一。

大模型系统软件的内存分配是确保性能和效率的关键因素。面对大模型系统软件在规模和性能等方面存在的挑战，设计研究合适的内存分配策略和优化技术至关重要。

4.4 存储系统

在大模型训练中，存储系统不仅可以保存数据，而且在

保障训练效率和稳定性方面也发挥着至关重要的作用。随着模型规模的不断扩大，如Meta的Llama-2 70B模型涉及的数据量高达约8 TB，这对存储系统也有着更高的要求。这些数据在训练过程中将被频繁且随机地读取。同时，为了模型的持续迭代和优化，新的数据不断被添加到训练集中。此外，在大模型训练过程中可能会遇到的硬件故障或算法错误要求存储系统必须具备有效的容错机制，例如通过定期创建模型参数的检查点来实现错误恢复，保证训练的连续性和数据的完整性。因此，为了支持高效的大模型训练，存储系统需要综合优化并发读效率、异步写效率和检错纠错等能力。

近年来，为了应对这些挑战，业界已经有了不少研究和进展。例如，Google在训练Gemini模型时，将传统硬盘检查点更换为内存检查点，并设置冗余存储，这样既减少了写入时间，又确保了在部分节点出错时能够恢复完整的参数和优化器状态。中国的MegaScale[43]万卡训练系统引入了两阶段检查点机制，即首先将GPU显存状态传输至CPU内存，随后异步将数据传输至分布式文件系统。这种设计大幅提高了训练效率，同时保持了GPU的计算任务和存储系统之间的非阻塞传输[44-45]。

尽管在存储系统方面已取得一定的进展，但如何在提升效率、保障训练稳定性与优化训练结果之间找到最佳平衡点，依然是一个值得深入研究和探讨的重要课题。

5 结束语

高效的系统软件是发挥底层硬件性能的必要条件。中国智能算力平台的硬件能力已经接近国际领先水平，但是其上的核心基础系统软件仍有较大的进步空间。为了提升中国算力平台竞争力，更好地服务大模型预训练等重要应用场景，清华大学的研究团队在核心基础系统软件层做了深入研究。2021年，清华团队开发了“八卦炉”系统，对上述智能算力软件栈的10个核心基础软件深入优化，成功在新一代神威超级计算机上高效支持了百万亿参数量大模型预训练任务。目前，“八卦炉”系统仍在持续迭代，在更多的国产芯片平台（天数、沐曦、壁仞、寒武纪等）上深度优化核心软件栈，为充分发挥国产算力硬件能力做好软件支持。

参考文献

[1] NVIDIA Corporation. CUDA toolkit [EB/OL]. [2024-02-20]. https://developer.nvidia.com/cuda-toolk it

[2] TILLET P, KUNG H T, COX D. Triton: an intermediate language and compiler for tiled neural network computations [C]// Proceedings of the 3rd ACM SIGPLAN International Workshop on Machine Learning and Programming Languages. ACM, 2019: 10 - 19. DOI: 10.1145/3315508.3329973

[3] 中科寒武纪科技股份有限公司. 寒武纪基础软件平台 [EB/OL]. [2024-02-22]. https://www. cambricon. com/index. php? m=content&c=index&a=lists&catid=71

[4] 华为技术有限公司. Ascend C：面向算子开发场景的编程语言 [EB/OL]. [2024-02-22]. https://www.hiascend.com/zh/ascend-c

[5] 上海壁仞科技股份有限公司. BIRENSUPA™软件开发平台 [EB/OL]. [2024-02-22]. https://www.birentech.com/product_details/1.html

[6] 摩尔线程智能科技（北京）有限责任公司. MUSA SDK [EB/OL]. [2024-02-21]. https://developer.mthreads.com/musa/musa-sdk

[7] STONE J E, GOHARA D, SHI G C. OpenCL: a parallel programming standard for heterogeneous computing systems [J]. Computing in science and engineering, 2010, 12(3): 66-73

[8] The Khronos® SYCL™ Working Group. SYCL™ 2020 Specification (revision 8) [EB/OL]. [2024-02-24]. https://registry. khronos. org/SYCL/specs/sycl-2020/html/sycl-2020.html

[9] Modular Inc. Mojo - the programming language for all AI developers [EB/OL]. [2024-02-22]. https://www. modular. com/max/mojo

[10] Intel Corporation. Intel® oneAPI DPC++ Library. [EB/OL]. [2024-02-22]. https://www. intel. com/content/www/us/en/developer/tools/oneapi/dpc-library.html

[11] JIA Y Q. Caffe [EB/OL]. [2024-02-22]. https://caffe.berkeleyvision.org/

[12] Google Inc. Tensorflow [EB/OL]. [2024-02-22]. https://www.tensorflow.org/

[13] Meta Platform Inc. Pytorch [EB/OL]. [2024-02-23]. https://pytorch.org/

[14] 百度在线网络技术（北京）有限公司. 源于产业实践的开源深度学习平台 [EB/OL]. [2024-02-22]. https://www.paddlepaddle.org.cn/

[15] 华为技术有限公司. 昇思 MindSpore [EB/OL]. [2024-02-22]. https://www.mindspore.cn/

[16] SHOEYBI M, PATWARY M, PURI R, et al. Megatron-lm: training multi-billion parameter language models using model parallelism [EB/OL]. [2024-02-25]. https://arxiv. org/abs/1909.08053

[17] Microsoft Corporation. Deepspeed [EB/OL]. [2024-02-25]. www.deepspeed.ai

[18] Huggingface Inc. Huggingface [EB/OL]. [2024-02-25]. https://huggingface.co/

[19] NVIDIA Corporation. NVLink and NVSwitch [EB/OL]. [2024-02-25]. https://www.nvidia.com/en-us/data-center/nvlink

[20] HE J A, ZHAI J D, ANTUNES T, et al. FasterMoE: modeling and optimizing training of large-scale dynamic pre-trained models [C]//Proceedings of the 27th ACM SIGPLAN Symposium on Principles and Practice of Parallel Programming. ACM, 2022: 120-134. DOI: 10.1145/3503221.3508418

[21] ZHAI M S, HE J A, MA Z X, et al. SmartMoE: efficiently training sparsely-activated models through combining offline and online parallelization [EB/OL]. [2024-02-21]. https://www. usenix. org/conference/atc23/presentation/zhai

[22] NVIDIA Corporation. NVIDIA collective communications library [EB/OL]. [2024-02-25]. https://developer.nvidia.com/nccl

[23] Microsoft Corporation. MSCCL Leaderboard [EB/OL]. [2024-02-25]. https://microsoft.github.io/msccl-leaderboard/

[24] Intel Corporation. Intel® oneAPI collective communications library [EB/OL]. [2024-02-20]. https://www. intel. com/content/www/us/en/developer/tools/oneapi/oneccl.html

[25] AMD Corporation. RCCL Documantation [EB/OL]. [2024-02-20]. https://rocm.docs.amd.com/projects/rccl/en/latest/api.html

[26] Meta Platform Inc. Gloo Documentation [EB/OL]. [2024-02-20]. https://github. com/facebookincubator/gloo/blob/main/docs/readme.md

[27] DONG J B, WANG S C, FENG F, et al. ACCL: architecting highly

scalable distributed training systems with highly efficient collective communication library [J]. IEEE micro, 2021, 41(5): 85–92. DOI: 10.1109/MM.2021.3091475
[28] 华为技术有限公司. 昇思通信 [EB/OL]. [2024–02–20]. https://www. mindspore. cn/docs/zh–CN/r1.10/api_python/mindspore.communication.html
[29] NVIDIA Corporation. cutlass: Fast linear algebra in CUDA C++ [EB/OL]. [2024–02–26]. https://developer. nvidia. com/blog/cutlass–linear–algebra–cuda
[30] NVIDIA Corporation. cuBLAS: Basic linear algebra on NVIDIA GPU [EB/OL]. [2024–02–26]. https://developer.nvidia.com/cublas
[31] NVIDIA Corporation. cuDNN: the NVIDIA CUDA deep neural network library [EB/OL]. [2024–02–20]. https://developer. nvidia.com/cudnn
[32] Advanced Micro Devices. rocBLAS library [EB/OL]. [2024–02–22]. https://github.com/ROCm/rocBLAS
[33] Advanced Micro Devices. hipDNN library [EB/OL]. [2024–02–23]. https://github.com/ROCm/hipDNN
[34] 华为技术有限公司. AI异构计算架构, 昇腾计算服务层, 昇腾算子库AOL [EB/OL]. [2024–02–20]. https://www. hiascend. com/software/cann
[35] The Apache Software Foundation. Apache TVM [EB/OL]. [2024–02–23]. https://tvm.apache.org
[36] OpenXLA. XLA (Accelerated Linear Algebra) [EB/OL]. [2024–02–23]. https://openxla.org/xla
[37] ZHENG L Y, WANG H J, ZHAI J D, et al. EINNET: optimizing tensor programs with derivation–based transformations [EB/OL]. [2024–02–20]. https://www. usenix. org/conference/osdi23/presentation/zheng
[38] MA Z X, WANG H J, XING J Z, et al. PowerFusion: a tensor compiler with explicit data movement description and instruction–level graph IR [EB/OL]. (2023–07–11)[2024–02–20]. https://arxiv.org/abs/2307.04995
[39] The Kubernetes Authors. Production–grade container orchestration [EB/OL]. [2024–02–20]. https://kubernetes.io
[40] 华为技术有限公司. AI开发平台_ModelArts_AI智能开放平台_人工智能平台_机器学习–华为云 [EB/OL]. [2024–02–20]. https://www.huaweicloud.com/product/modelarts.html
[41] The Linux Foundation. Horovod [EB/OL]. [2024–02–22]. https://horovod.ai
[42] Ray–project. Ray [EB/OL]. [2024–02–24]. https://www.ray.io
[43] MA Z X, HE J A, QIU J Z, et al. BaGuaLu: targeting brain scale pretrained models with over 37 million cores [C]//Proceedings of the 27th ACM SIGPLAN Symposium on Principles and Practice of Parallel Programming. ACM, 2022: 192 – 204. DOI: 10.1145/3503221.3508417
[44] 王豪杰, 马子轩, 郑立言, 等. 面向新一代神威超级计算机的高效内存分配器 [J]. 清华大学学报(自然科学版), 2022, 62(5): 943–951. DOI: 10.16511/j.cnki.qhdxxb.2022.22.007
[45] JIANG Z H, LIN H B, ZHONG Y M, et al. MegaScale: scaling large language model training to more than 10, 000 GPUs [EB/OL]. (2024–02–23) [2024–02–25]. https://arxiv. org/abs/2402.15627

作 者 简 介

郑纬民，清华大学计算机系教授、中国工程院院士；长期从事高性能计算机体系结构、并行算法和系统的研究，提出可扩展的存储系统结构及轻量并行的扩展机制，发展了存储系统扩展性理论与方法，在中国率先研制并成功应用集群架构高性能计算机，在国产神威太湖之光上研制的极大规模天气预报应用获得ACM戈登·贝尔奖；曾获国家科技进步奖一等奖1项、二等奖2项，国家技术发明奖二等奖1项，何梁何利基金科学与技术进步奖，以及首届中国存储终身成就奖；发表学术论文400余篇，编写和出版相关教材和专著10部。

翟季冬，清华大学计算机系长聘教授、博士生导师，国家杰出青年科学基金获得者，国家重点研发计划项目负责人，清华大学计算机系高性能所副所长，中国计算机学会（CCF）高性能计算专委副主任、CCF杰出会员，ACM中国高性能计算专家委员会秘书长；主要研究领域包括并行计算、编程模型与编译优化；研究成果获IEEE TPDS 2021最佳论文奖、IEEE CLUSTER 2021最佳论文奖、ACM ICS 2021最佳学生论文奖等；获教育部科技进步奖一等奖、中国计算机学会自然科学奖一等奖、CCF–IEEE CS青年科学家奖；发表论文100余篇，出版专著1部。

翟明书，清华大学计算机系高性能所在读博士研究生；主要研究领域包括高性能计算、机器学习系统；曾担任清华大学学生超算团队队长，获得两次世界冠军；发表多篇论文。

大语言模型算法演进综述

Review of Evolution of Large Language Model Algorithms

朱炫鹏/ZHU Xuanpeng，姚海东/YAO Haidong，刘隽/LIU Jun，熊先奎/XIONG Xiankui

（中兴通讯股份有限公司，中国 深圳 518057）
(ZTE Corporation, Shenzhen 518057, China)

DOI：10.12142/ZTETJ.202402003
网络出版地址：http://kns.cnki.net/kcms/detail/34.1228.TN.20240422.2005.004.html
网络出版日期：2024-04-23
收稿日期：2024-03-02

摘要：基于Transformer架构的大语言模型展现出强大的能力，是人类迈向通用人工智能（AGI）的一个重大进步。大语言模型架构和算法的演进分为提高推理效率、提高模型能力两条技术路线。介绍了两条技术路线主流的技术方案和思路。提高推理效率的方法有分布式推理、计算优化、访存优化、量化等；提高模型能力主要是引入新的架构，如混合专家（MoE）模型、状态空间模型（SSM）等。

关键词：大语言模型；Transformer；注意力

Abstract: The large language model based on the Transformer architecture shows powerful capabilities, and it is a major progress towards artificial general intelligence (AGI). The evolution of large language model architecture and algorithms is divided into two technical paths: improving the inference efficiency and model capability. The mainstream technical solutions and ideas for the two technical routes are described. Methods for improving inference efficiency include distributed inference, computing optimization, memory access optimization, and quantification. To improve model capabilities, new architectures such as mixture of experts (MoE) and state space model (SSM) are introduced.

Keywords: large language model; Transformer; attention

引用格式：朱炫鹏, 姚海东, 刘隽, 等. 大语言模型算法演进综述 [J]. 中兴通讯技术, 2024, 30(2): 9-20. DOI: 10.12142/ZTETJ.202402003
Citation: ZHU X P, YAO H D, LIU J, et al. Review of evolution of large language model algorithms [J]. ZTE technology journal, 2024, 30(2): 9-20. DOI: 10.12142/ZTETJ.202402003

1 大语言模型算法发展概况

OpenAI于2022年、2023年分别发布ChatGPT[1]和GPT-4[2]，其强大的会话能力、多模态能力震惊业界，是人类迈向通用人工智能（AGI）的一个重大进步。ChatGPT和GPT-4能力强大的原因有两个：一是Transformer[3]架构的自注意力机制，可获取任意距离间单词的相关信息；二是大模型、大数据、大算力，规模超过了一定阈值，则会产生涌现能力[4]。

目前各大公司都发布了自己的大语言模型（LLM）。本文中，我们主要介绍大语言模型在两条技术路线上的架构和算法的演进。

1.1 语言模型的发展历程

语言模型的发展经历了统计语言模型、神经语言模型、预训练语言模型和大语言模型4个阶段[5]。其结构从基于统计概率发展到基于神经网络，模型复杂度不断增加，能力也出现了质的提升。

1） 统计语言模型

最初的语言模型是基于统计概率的，即根据语料统计出在某个上下文出现某个词的概率，根据概率选择最合适的词。

2） 神经语言模型

文献[6]首次将神经网络引入语言模型。常见的模型结构有循环神经网络（RNN）[7]、长短期记忆网络（LSTM）[8]等。RNN用隐藏层保存逐个输入的词的信息，但由于梯度消失和梯度爆炸，只能保留短期信息。LSTM使用门控机制，可以选择性地保留长期信息。

3） 预训练语言模型

ELMo[9]用预训练的双向LSTM网络根据上下文动态生成词向量，解决了一词多义问题。双向LSTM网络可以在下游任务上微调，得到更好的效果。基于Transformer的双向编码器表征法（BERT）[10]也采用了预训练+下游任务微调的范式。

4） 大语言模型

预训练语言模型的性能随着规模的增大而提高，成幂律关系[11-12]。OpenAI设计了大型语言模型GPT-3[13]。该模型表现出强大的能力，性能和规模超越了幂律关系，出现了涌现

现象，如上下文学习、思维链推理。

1.2 大语言模型算法演进路线

大语言模型的发展主要有两条技术路线：一是提高推理效率，降低推理成本；二是提高模型能力，迈向AGI。

大语言模型能力强大，有广阔的应用前景，各厂商都在积极部署，提供服务。但是，由于模型规模巨大，算法对硬件不够友好，需要消耗大量的算力、存储、能源。因此，如何降低推理成本、推理延时，是一个亟待解决的问题。大语言模型主要的技术路线有分布式推理、减小模型计算量、减小模型访存量、提升硬件亲和性等。

大语言模型是迈向AGI的重大进步，而Transformer是其中的核心架构，发挥了重大作用。但Transformer也有一定的不足，如计算量大，通过提升规模来提升性能更加困难；上下文窗口长度有限，难以支持超长序列。研究人员通过引入新的结构，解决这些问题，取得了较好的效果。

2 大语言模型架构

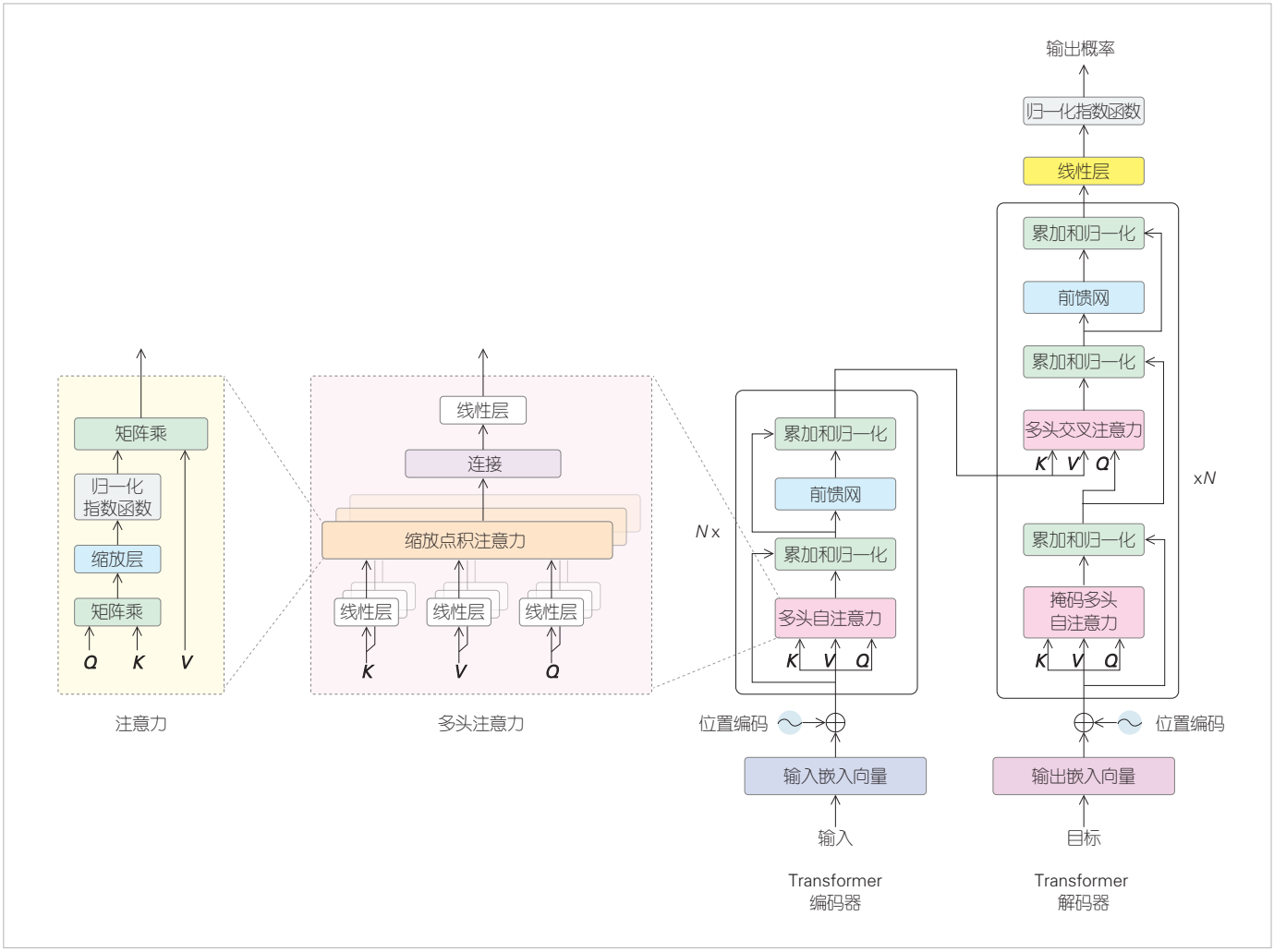
2.1 Transformer

Transformer模块是组成大语言模型的基础单元，由多头注意力、前馈网、Softmax、LayerNorm等部分组成，本节中我们主要介绍Transformer的结构和算法。

2.1.1 注意力机制

注意力机制是针对一个文本序列，计算每个token（符号）与其他tokens之间的相关系数，找出相关度高的tokens，用于生成特征。例如，“这是一只猫，它很可爱。”在这句话里，“猫”与“这”“它”的相关度会比较高。

注意力机制是基于查询-键-值（QKV）计算的。具体算法为：输入$\boldsymbol{Q}$、$\boldsymbol{K}$、$\boldsymbol{V}$，用$\boldsymbol{Q}$和$\boldsymbol{K}^T$做矩阵乘，除以$\sqrt{D_k}$，做Softmax，得到注意力矩阵$\boldsymbol{A}$；$\boldsymbol{A}$和$\boldsymbol{V}$做矩阵乘，得到输出特征。具体的公式如下，结构见图1。



▲图1 Transformer架构[3]

$$\text{Attention}(\boldsymbol{Q},\boldsymbol{K},\boldsymbol{V})=\text{Softmax}(\frac{\boldsymbol{Q}\boldsymbol{K}^{\mathrm{T}}}{\sqrt{D_k}})\boldsymbol{V}=\boldsymbol{A}\boldsymbol{V}, \tag{1}$$

其中，$\boldsymbol{Q}\in\mathrm{R}^{N\times D_k}$，$\boldsymbol{K}\in\mathrm{R}^{M\times D_k}$，$\boldsymbol{V}\in\mathrm{R}^{M\times D_v}$，$N$是$\boldsymbol{Q}$的长度，$M$是$\boldsymbol{K}$的长度，$D_k$是$\boldsymbol{K}$的向量维度，$D_v$是$\boldsymbol{V}$的向量维度。

注意力机制能够并行计算所有tokens间的相关信息，没有距离的限制，与RNN、LSTM相比更具有优势。

2.1.2 多头注意力

多头注意力（MHA）是将$\boldsymbol{Q}$、$\boldsymbol{K}$、$\boldsymbol{V}$转换成多份，每份单独计算注意力，结果合并在一起。每一份称为一个头，多个头可以计算不同领域中的相关关系，增加模型的信息容量和能力，具体如公式（2），结构见图1。

$$\text{MultiHeadAttn}(\boldsymbol{Q},\boldsymbol{K},\boldsymbol{V})=\text{Concat}(\boldsymbol{head}_1,\cdots,\boldsymbol{head}_H)\boldsymbol{W}^O,$$

$$\boldsymbol{head}_i=\text{Attention}(\boldsymbol{Q}\boldsymbol{W}_i^Q,\boldsymbol{K}\boldsymbol{W}_i^K,\boldsymbol{V}\boldsymbol{W}_i^V), \tag{2}$$

其中，$\boldsymbol{Q}$、$\boldsymbol{K}$、$\boldsymbol{V}$的向量维度都是D_m，转换后每一份的维度分别为D_k、D_k、D_v，合并后维度又恢复成D_m。

在Transformer中多头注意力有3种不同的形式：

1）自注意力。多头注意力公式中取$\boldsymbol{Q}=\boldsymbol{K}=\boldsymbol{V}=\boldsymbol{X}$，$\boldsymbol{X}$是Transformer的输入特征，即计算$\boldsymbol{X}$与自己的注意力。

2）掩码自注意力。在自注意力公式中，将注意力矩阵$\boldsymbol{A}$中的某些值改为$-\infty$，避免一些tokens间的关注。

3）交叉注意力。在编码器和解码器之间计算注意力，$\boldsymbol{K}$、$\boldsymbol{V}$来自编码器，$\boldsymbol{Q}$来自解码器。

2.1.3 前馈网

前馈网（FFN）由两个全连接层组成。经过第1个全连接层，特征维度由D_m扩大到D_f；经过第2个全连接层，特征维度由D_f恢复到D_m，具体见公式（3）：

$$\text{FFN}(\boldsymbol{H}')=\text{ReLU}(\boldsymbol{H}'\boldsymbol{W}^1+\boldsymbol{b}^1)\boldsymbol{W}^2+\boldsymbol{b}^2, \tag{3}$$

其中，$\boldsymbol{H}'$是本层输入，$\boldsymbol{W}^1\in\mathrm{R}^{D_m\times D_f}$，$\boldsymbol{W}^2\in\mathrm{R}^{D_f\times D_m}$，$\boldsymbol{b}^1\in\mathrm{R}^{D_f}$，$\boldsymbol{b}^2\in\mathrm{R}^{D_m}$。

2.1.4 残差连接和归一化层

残差连接能够防止梯度消失，归一化层使特征数值维持在均值0、方差1。这样多个Transformer组合成深层网络，可以保持前向、反向数值的稳定，具体见公式（4）—（5）：

$$\boldsymbol{H}'=\text{LayerNorm}(\text{SelfAttention}(\boldsymbol{X})+\boldsymbol{X}), \tag{4}$$

$$\boldsymbol{H}=\text{LayerNorm}(\text{FFN}(\boldsymbol{H}')+\boldsymbol{H}')。 \tag{5}$$

2.1.5 位置编码

Transformer一次性输入序列的所有tokens，不像RNN那样可以根据输入顺序表示token的前后关系。因此，Transformer需要在每个token上累加一个位置编码，来表示token在序列中的位置。

2.1.6 Transformer整体架构

完整的Transformer由编码器和解码器两部分组成，结构如图1。编码器包括多头自注意力、前馈网和其他辅助层。解码器包括掩码多头自注意力、多头交叉注意力、前馈网和其他辅助层。

在设计模型时，我们根据模型的不同功能，可以选择编码器和解码器的不同组合。

1）使用编码器–解码器。使用Transformer的完整结构，输入、输出都是序列。此结构一般用于序列到序列任务，如文本翻译。

2）只使用编码器。输入为序列，输出是序列的表示。此结构一般用于文本分类、序列标记任务，如BERT模型使用的是编码器。

3）只使用解码器。因为只有解码器，没有编码器，要移除解码器中与编码器关联的多头交叉注意力。此结构的输入为序列，输出是一个token（该token再作为输入，继续输出下一个token，直到输出结束）。此结构一般用于序列生成任务。GPT模型使用的是解码器。

2.2 ChatGPT系列模型架构

OpenAI从2018年发布GPT–1[14]到2023年发布GPT–4，模型的能力产生了质的飞跃。模型虽然一直保持Transformer Decoder的总体架构不变，但具体模块有所调整，如GPT–2[15]将LayerNorm移到解码器的输入，而GPT–4引入了混合专家（MoE）层[16]。另外，模型的规模也在数量级地增加，参数量从1.17×10^8增加到1×10^{12}以上。这样的量变引起质变，模型产生了涌现能力。

ChatGPT系列模型的主要创新点和架构如表1所示。

3 大语言模型高效推理

3.1 大语言模型的计算特性

Transformer的结构与CNN有较大差别。CNN的卷积计算数据复用率高。Transformer中的矩阵乘法、矩阵乘向量，数据复用率较低；非线性算子Softmax、LayerNorm计算时要多次遍历数据[17]。这些特点造成Transformer无法充分利用计

▼表1 ChatGPT系列模型的主要创新点和架构

模型名	主要创新点	发布时间/年	上下文序列长度（token）	Transformer层数	多头数量	参数量
GPT-1	• 基于Transformer解码器的单向语言模型 • 无监督预训练+有监督微调模式	2018	512	12	12	1.17×10^{8}
GPT-2	• 多任务预训练，取消微调 • 将LayerNorm移到解码器的输入	2019	1 024	48	48	1.5×10^{9}
GPT-3	• 模型层数增加 • 上下文学习 • 少样本学习、单样本学习、零样本学习	2020	2 048	96	96	1.75×10^{11}
ChatGPT（GPT-3.5）	• RLHF • PPO	2022	4 096	–	–	$>1.75\times10^{11}$
GPT-4	• 引入MoE • 多模态	2023	8 000	–	–	$>1\times10^{12}$

MoE：混合专家　　PPO：近端策略优化　　RLHF：人类反馈强化学习

算硬件的能力，在现有图形处理器（GPU）、加速器上计算效率较低。

3.1.1 模型规模大

大语言模型存在计算量大、存储量大的特点。以GPT-3为例，GPT-3包含96层Transformer，每层有96个注意力头，词向量深度为12 288。整个模型参数量达到1 750亿个，按照INT8数据格式计算，总大小达到175 GB。推理生成一个token需要的计算量达到3 240亿次。

英伟达A100型号GPU的INT8算力为624万亿次运算每秒（TOPS），算力利用率小于10%。A100的显存为80 GB，显然无法装下整个GPT-3模型。因此，对于GPT-3这类大语言模型，推理必须引入分布式技术，将一个模型拆分到多个GPU上，提升推理速度，减小推理延时。

3.1.2 计算强度低

考虑GPU计算深度学习模型的效率，我们需要引入算术强度的概念。算术强度的含义是模型的计算量与内存读写量的比值，具体如公式（6）[17]：

$$\text{Arithmetic Intensity}=\frac{\text{FLOPs}}{\text{MOPs}}, \tag{6}$$

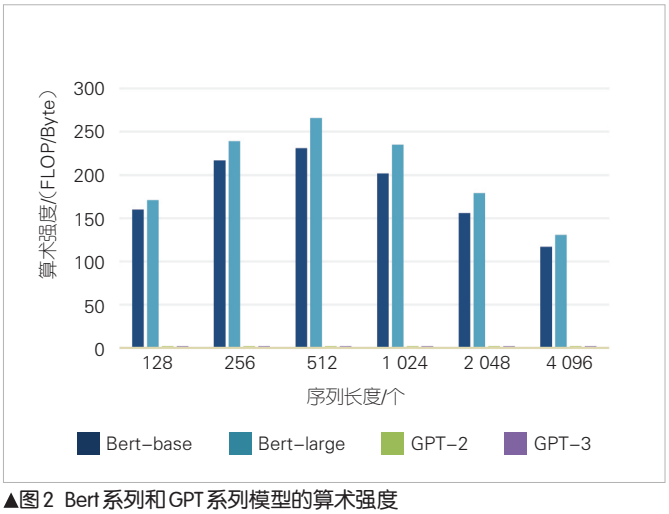
其中，FLOPs表示浮点计算次数，MOPs表示内存访问次数，以一个字节数据为单位统计。算术强度越高，说明模型的计算密集度越高，反之则说明模型的访存密集度越高。对于特定的AI加速器，算术强度有一个最佳值，模型达到这个最佳值，就能够同时最大效率地利用存储带宽与计算资源。如果高于最佳值，模型受限于计算资源，造成存储带宽的浪费；反之，模型受限于存储带宽，则会造成计算资源的浪费。

图2展示了Bert-base、Bert-large、GPT-2、GPT-3在不同序列长度下的推理算术强度。可以看出，由编码器组成的Bert系列模型算术强度较高，达到数百，且随着序列长度变化而变化；由解码器组成的GPT系列模型算术强度很低，只有2，且不随序列长度变化而变化。算术强度为2，意味着读取一个数据只计算2次，计算器件的大量时间浪费在等待数据上，无法充分发挥算力。

表2是Llama2模型[18]（数据类型FP16）在不同Batch（批量）下各层的推理算术强度。可以看出，Llama2的算术强度随着Batch的增大而增大。当Batch为1时，算术强度为1，推理效率很低；当Batch增加到512时，算术强度增加到10.18。再继续增加Batch大小，算术强度基本不变，因为此时激活的大小已超过权重大小，在访存量中占主导地位。

3.1.3 非线性层计算效率低

Transformer中包含LayerNorm、Softmax等非线性运算，



▲图2 Bert系列和GPT系列模型的算术强度

▼表2 Llama2模型各层算术强度

Batch大小	算子	计算量/TFLOPs	访存量/TMOPs	算术强度
1	MHA(线性投影)	42.95	42.46	1.01
	MHA(矩阵乘)	10.74	10.74	1.00
	FFN(线性投影)	57.98	57.99	1.00
	其他	0.02	0.02	1.18
	总计	111.69	111.21	1.00
128	MHA(线性投影)	5 497.56	43.79	125.53
	MHA(矩阵乘)	1 374.39	1 374.56	1.00
	FFN(线性投影)	7 421.70	59.22	125.32
	其他	2.84	2.41	1.18
	总计	14 296.49	1 479.99	9.66
512	MHA(线性投影)	21 990.23	47.82	459.85
	MHA(矩阵乘)	5 497.56	5 498.23	1.00
	FFN(线性投影)	29 686.81	62.95	471.61
	其他	11.37	9.65	1.18
	总计	57 185.98	5 618.65	10.18
4 096	MHA(线性投影)	175 921.86	85.40	2 059.93
	MHA(矩阵乘)	43 980.47	43 985.83	1.00
	FFN(线性投影)	237 494.51	97.71	2 430.59
	其他	91.00	77.19	1.18
	总计	457 487.84	44 246.13	10.34

FFN：前馈网　　TFLOPs：万亿次浮点运算
MHA：多头注意力　　TMOPs：万亿次内存访问

它们也会降低模型的计算效率。Softmax的计算公式为：

$$m = \max(\boldsymbol{X}),\ y_i = \frac{e^{x_i - m}}{\sum_{j=0}^{L-1} e^{x_j - m}}, \tag{7}$$

其中，$\boldsymbol{X}$表示输入向量，L表示输入向量长度。从公式（7）可以看出，Softmax计算会带来这些挑战：计算过程多次完整访问向量$\boldsymbol{X}$，这导致数据长距离共享；每次计算一行向量，这和通常矩阵运算访问数据的方式不一致，这会导致算子融合困难；算术强度低，访存需求大，序列较长时需要反复从片外内存读取数据，这降低了效率[19]。

LayerNorm层包含激活的均值、方差统计计算，也需要多次访问向量$\boldsymbol{X}$，进行非线性计算。这会造成计算效率的降低，和Softmax存在的问题类似。

3.2 大语言模型推理效率提升方法

大语言模型推理效率的评价指标有：首token生成延时、每秒生成token数、每token消耗的芯片毫秒数[20]。提升推理效率的思路主要有3条：

• 增加计算硬件：进行分布式推理，将模型拆分到多个GPU上，提升推理速度，减小推理延时；

• 减小计算量：删除模型中不必要的计算；

• 减小访存量：减少对外部存储器的访问，充分利用缓存；降低数据精度，减小数据大小。

3.2.1 分布式推理

当大语言模型计算量、参数量超过单个GPU能力时，就必须做分布式训练[21]和分布式推理。分布式推理使用的两种典型并行方式为：Tensor并行和Pipeline并行，一般节点内采用Tensor并行[22-24]，节点间采用Pipeline并行[23]。

Tensor并行将每个Transformer层的计算量、存储量平均分布到多个GPU上，能有效降低推理延时，但会增加GPU间数据交互的负担。因此选择Tensor并行度时，需要对比并行计算收益和通信负担，综合评估效率指标。

Pipeline并行将模型的各个Transformer层分配到不同的节点，每个节点负责不同的层。在推理时，各节点上的层进行串行计算。因此，Pipeline并行不能减少推理总延时，只能减小节点存储量。在满足延时指标的前提下，Pipeline并行度不宜过大，只要节点显存足够支持本节点的Transformer层即可。

3.2.2 计算优化

在Transformer的自注意力中，矩阵乘的计算复杂度为序列长度的平方。当序列较长时，计算量会大幅增加。自注意力的非线性算子虽然计算量不大，但计算效率低。这两点都会使延时增大。为了缩短延时，我们可以使用多种方案减少计算步骤，降低计算量。

3.2.2.1 KV Cache

大语言模型由解码器组成，是生成式模型。在推理时，输入一段提示，能够生成一段回答。生成的回答并不是一次生成所有tokens，而是每次推理生成一个token，这个token再和输入的所有tokens拼接在一起，作为下一次推理的输入，生成下一个token。这样反复进行，直到模型输出结束符（EOP）为止。

生成式模型推理的弊端是很明显的。每次输入的所有tokens，都要参与注意力计算。除了最新token，前面的其他tokens的计算与前次推理相同，是重复计算。为了解决这个问题，可使用KV Cache。它可以将每次推理计算出的tokens的特征缓存在显存中，下次推理直接取用，无须重复计算。这样每次推理，输入只有一个最新token，自注意力的计算量可以大幅下降。

大语言模型推理过程属于带宽受限型，KV Cache虽然

减少了大量计算，但也带来了存储和带宽的压力。例如，GPT-3模型参数占用显存大小为350 GB，假设Batch大小为64，输入序列长为512，输出序列长度为32，则KV cache占用显存为164 GB，大约是模型参数占用显存的1/2。KV Cache的规模较大，且对存储带宽要求较高，并会遭遇内存墙问题。因此，又出现了多查询注意力（MQA）[25]和分组查询注意力（GQA）[26]。

3.2.2.2 共享关注头

在原始Transformer的MHA中，QKV分别包含相同数量的头，且一一对应。每个头的QKV内部进行计算，再将结果拼接在一起。

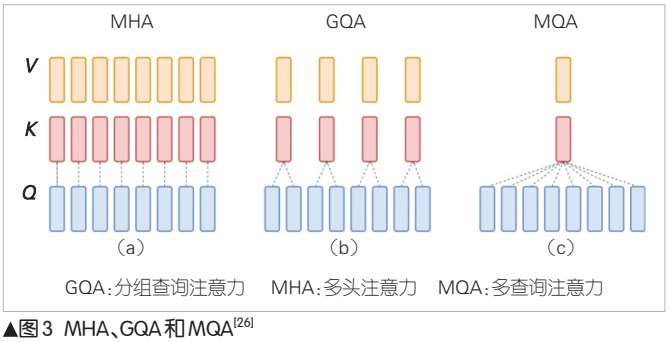
MQA的$\boldsymbol{Q}$仍然保持原来的头数，但$\boldsymbol{K}$和$\boldsymbol{V}$只有一个头，共享给所有$\boldsymbol{Q}$头使用，如图3（c）所示。MQA免除了多个$\boldsymbol{KV}$头的计算和存储，大大减少了存储和访存带宽压力，推理吞吐量可提高30%～40%，而模型性能只有少量损失。

GQA是MHA和MQA的折衷，该方法减少了模型性能损失，获得MQA带来的推理加速好处。具体方法是，不是所有$\boldsymbol{Q}$头共享一组$\boldsymbol{KV}$，而是$\boldsymbol{Q}$头分成多组，每组共享一组$\boldsymbol{KV}$，例如图3（b）是每两个$\boldsymbol{Q}$头共享一组$\boldsymbol{KV}$。

3.2.2.3 推测解码

推测解码[27-28]是2023年新兴的大语言模型推理加速技术，通过增加推理的并行度来提高计算效率，降低延时。其具体方法是为大语言模型配备一个小语言模型，推理时，先由小语言模型“推测”生成几个tokens，然后将这几个tokens放入大语言模型中进行推理验证。如果验证正确，则小语言模型继续“推测”后续token；如果验证错误，大语言模型修正已有token，小语言模型接受修正，继续“推测”后续token。

推测解码之所以能降低延时，一是因为小模型计算速度远超大模型，有数量级的提升；二是因为大模型并行推理几个tokens，只需读取一次参数，计算强度提高，平均每个token的计算延时大幅降低。



▲图3 MHA、GQA和MQA[26]

3.2.2.4 精简Transformer

文献[29]介绍了Transformer的简化，并以信号传播理论及实证研究结果为基础，证明了Transformer中许多组件，如残差连接、Value、投影和LayerNorm，可以在不牺牲训练速度的情况下被删除。在纯自回归解码器和纯BERT编码器模型上的实验表明，简化后Transformer实现了与标准Transformer相当的训练速度和性能，同时训练吞吐量提高了15%，使用的参数减少了15%。

3.2.3 访存优化

3.2.3.1 FlashAttention

FlashAttention[30]通过分块计算Softmax和核函数融合，来降低对显存的访问。

计算Softmax需要遍历两遍全体数据。FlashAttention修改了Softmax算法，将数据分成多个小块，无须遍历即可逐块计算出Softmax的中间结果，当所有块计算完成后，再对中间结果进行一次校正，就可得最终结果。

FlashAttention还将Softmax和前后的矩阵乘等算子融合成一个统一计算设备架构（CUDA）核函数。每块数据在GPU上完成核函数计算后，结果被输出到显存。这个过程充分利用了数据局部性，中间激活保存在GPU缓存，避免了反复读写显存，这可以使计算速度提升7.6倍。

FlashAttention-2[31]是对FlashAttention的改进，它消除了原先频繁的系数更新，减少了对加速设备不擅长的非矩阵乘法运算的需求，提出了在序列长度上的并行化，获得并行加速优势。结合GPU运行特点，在一个注意力计算块内，我们将工作分配在一个CUDA线程block的不同warp上，以减少通信和共享内存读写，提高模型效率。

3.2.3.2 Flash-Decoding

Flash-Decoding[32-33]借鉴了FlashAttention的思路，将并行化维度扩展到KV序列长度。KV序列被分成多个小块，每块内部完成QKV注意力计算，多块之间可并行计算，无须等待Softmax统计全局最大值。

Flash-Decoding不是在运行时实时统计数据最大值，而是在模型设计时离线统计数据的分布，在分布区间内取一个较大的值Φ作为最大值。运行时直接用Φ计算Softmax，可以应付90%以上的情况。如果实际的最大值超过了Φ，就会

暂停Flash-Decoding，回退到原始算法。

3.2.3.3 PagedAttention

KV Cache显著减小了模型的计算量，但也存在一些缺点：一是显存占用大，达到数GB以上；二是大小动态变化，随着序列长度的不同，大小可相差数千倍，且不可预测。这给有效管理KV Cache带来了很大的挑战。研究发现，由于碎片化和过度保留，现有系统浪费了60%～80%的显存。受操作系统中虚拟内存和分页经典思想启发，PagedAttention[34]允许在非连续的内存空间中存储连续的KV Cache，有效提高内存的利用率。同时PagedAttention带来另一个关键优势：高效的内存共享。例如，在并行采样中，多个输出序列是由同一个提示生成的，提示的KV Cache页面可以在输出序列中共享。

3.2.4 量化

量化是深度学习模型通用的压缩方法，将模型的权重/激活数据格式转换为INT8、INT4等整数，以降低计算量和存储量。大语言模型量化按阶段分有3种：训练感知量化（QAT）、训练后量化（PTQ）、微调感知量化（QAFT）[35]。由于大语言模型训练困难，因此QAT用得较少，PTQ、QAFT用得比较多，本文中我们主要讨论PTQ。

3.2.4.1 仅量化权重

大语言模型训练完成后，权重是已知的，而激活则要等到推理时才能知道，且取值范围与输入的序列相关。因此，权重比较容易量化，误差较小；而激活的量化较难，如果遇到离群值，误差会比较大。基于这样的特点，我们可以对模型做混合精度量化，方法有LLM.int8()[36]、GPT量化（GPTQ）[37]、不相干处理量化（QuIP）[38]、激活感知的权重量化（AWQ）[39]、离群值感知的权重量化（OWQ）[40]、稀疏量化表示（SpQR）[41]、细粒度权重量化（FineQuant）[42]。

LLM.int8()认为，激活中的离群值很重要，因此在计算矩阵乘法时，需要对离群值和正常值做不同的处理：

- 取出激活中异常值所在的列以及权重中对应的行，保持FP16格式，计算点乘；
- 剩下的激活正常值，与对应的权重量化到INT8，计算点乘，再反量化成FP16；
- 两者的结果累加，得最终结果。

3.2.4.2 权重和激活都量化

ZeroQuant[43]对权重做按组量化，对激活做按token量化，并逐层使用知识蒸馏缓解精度损失。此方法在BERT和GPT-3模型上精度较高。

针对激活的离群值，SmoothQuant[44]的量化方案是：激活有离群值，权重无离群值，如果把两者平均一下，让它们的数值都落入正常范围，就容易量化了。具体做法是，按通道统计激活的取值范围，如果发现有离群值，则把该通道的数据都除以系数a，权重中对应通道数据都乘以a，这样最终的计算结果不变。

类似的方法还有OliVe[45]、基于重排列的训练后量化（RPTQ）[46]、量化大语言模型（QLLM）[47]、Outlier Suppression+[48]。

4 大语言模型的性能提升

将大语言模型加入新的结构，也可以提升模型性能。典型的结构有MoE和状态空间模型（SSM）[49]。

MoE采用稀疏化策略，即在模型中放置大量专家，每个专家代表一个领域，推理时每次只启用少量专家。这样的话，模型可以在整体上关注更多的领域，而推理时，单个token只关注特定领域，避免了在无关领域上浪费算力。这样既减小了计算量，又提高了模型性能。

SSM不受上下文窗口长度的限制，能够处理超长序列，选择性记录上下文信息，可以在音频、文本等方面展现出良好的性能。SSM在结构上综合了CNN与RNN的特点，计算量、存储量比Transformer更小。

4.1 MoE

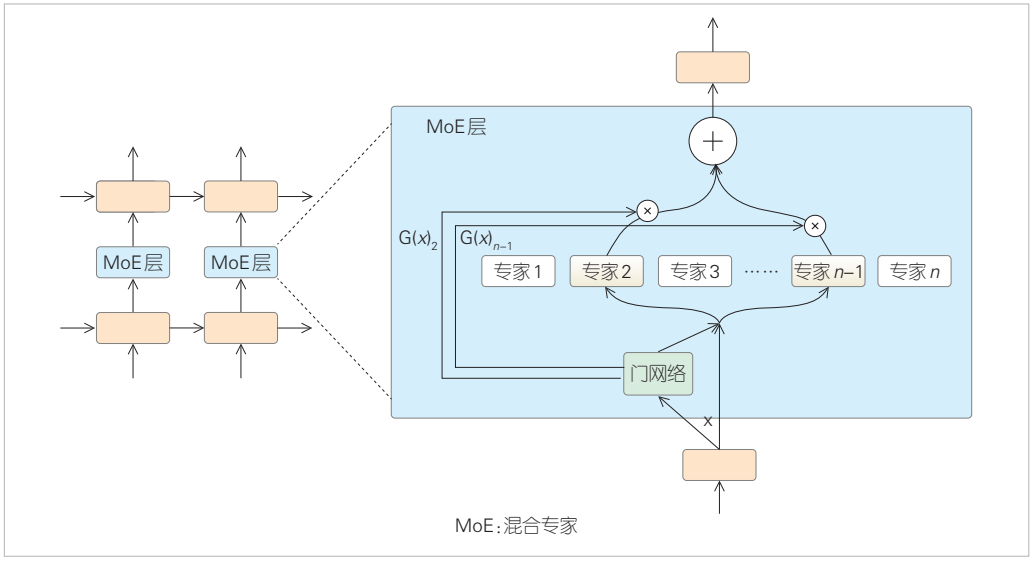
文献[16]首先提出了MoE的概念，认为用单个模型去适应多个场景的样本，会受到很多干扰，因此会导致学习很慢、泛化困难。使用多个模型，且每个模型学习一个场景，就可以得到比较好的效果。在多个模型之前增加一个门网络，就可以决定每个数据应该被哪个模型处理。

文献[50]提出将MoE引入语言模型，并做了两点创新：

- Sparsely-Gated：门网络每次只选择两个专家进行计算，显著降低了计算量；
- token-level：对一个序列中的每个token，各自独立地分别选择专家；

文献[51]的MoE架构如图4，推理的计算步骤如下：

- 输入$\boldsymbol{x}$进入门网络，经过一个全连接层，计算出每个专家的概率；
- 选出概率最高的两个专家，将$\boldsymbol{x}$输入到这两个专家中；
- 两个专家计算的结果，与前面的概率相乘累加，得最终输出。



▲图4 MoE架构[51]

GShard[51]和Switch Transformers[52]在大语言模型中加入MoE，取得了较好效果，下面我们将详细介绍。

4.1.1 GShard

文献[51]认为，增加大语言模型的深度和宽度，计算复杂度则会超线性增加，一旦超过了$O(n)$，大规模训练难以实现。在Transformer中加入MoE，计算复杂度会明显下降到小于$O(n)$，大规模训练则可以实现。因此，文献[51]提出了GShard，并在分布式训练、推理中使用了专家并行，将多个专家分布在不同的计算设备上，降低了每个设备的存储量。

Transformer中加入MoE的方法，将FFN替换为MoE。MoE中有多个专家，每个专家都是一个单独的FFN。具体如公式（8）—（10）：

$$\boldsymbol{G}_{s,E} = \mathrm{GATE}(\boldsymbol{x}_s), \tag{8}$$

$$\mathrm{FFN}_e(\boldsymbol{x}_s) = \boldsymbol{wo}_e \cdot \mathrm{ReLU}(\boldsymbol{wi}_e \cdot \boldsymbol{x}_s), \tag{9}$$

$$\boldsymbol{y}_s = \sum_{e=1}^{E} G_{s,e} \cdot \mathrm{FFN}_e(\boldsymbol{x}_s), \tag{10}$$

其中，$\boldsymbol{x}_s$是MoE的输入，向量$\boldsymbol{G}_{s,E}$是门控网络GATE计算的每个专家的选择概率，其元素为$G_{s,e}, e \in [1, E]$，选择策略是取概率最大的两个专家进行实际计算，其他专家的概率设为0。

MoE中有E个专家FFN_1，…，FFN_E，第e个专家内部的计算为输入全连接层（权重为$\boldsymbol{wi}_e$）、线性整流函数（ReLU）、输出全连接层（权重为$\boldsymbol{wo}_e$）。

$\boldsymbol{y}_s$是MoE的输出，由选中的两个专家的计算结果与概率相乘累加而成。

GShard专家并行如图5所示。图5中分别为标准Transformer编码器、MoE Transformer编码器和多设备分布式的MoE Transformer编码器。MoE Transformer编码器与Transformer编码器的差别是FFN替换成了MoE，MoE放置在单计算设备上，设备需要存储所有专家的权重。多设备分布式的MoE Transformer编码器进一步实现了专家并行，E个专家分别放置在E个计算设备上，每个设备放置一个专家，因此只需存储一个专家的权重。在推理时，每个token通过门网络选择两个专家。如果专家就在自己所在的设备上，则直接进行计算；否则，将token通过All–to–All集合通信接口发送到专家所在的设备，进行计算，结果再通过All–to–All集合通信接口发回到原设备。

4.1.2 Switch Transformers

Switch Transformers也是将FFN替换为MoE，不同的是修改了门网络，每次只选择一个专家计算，称为Switch层。这样有3个好处：

- 减少了路由的计算量；
- 每个专家的批量大小（专家容量）至少可以减半；
- 路由执行简化，通信成本降低。

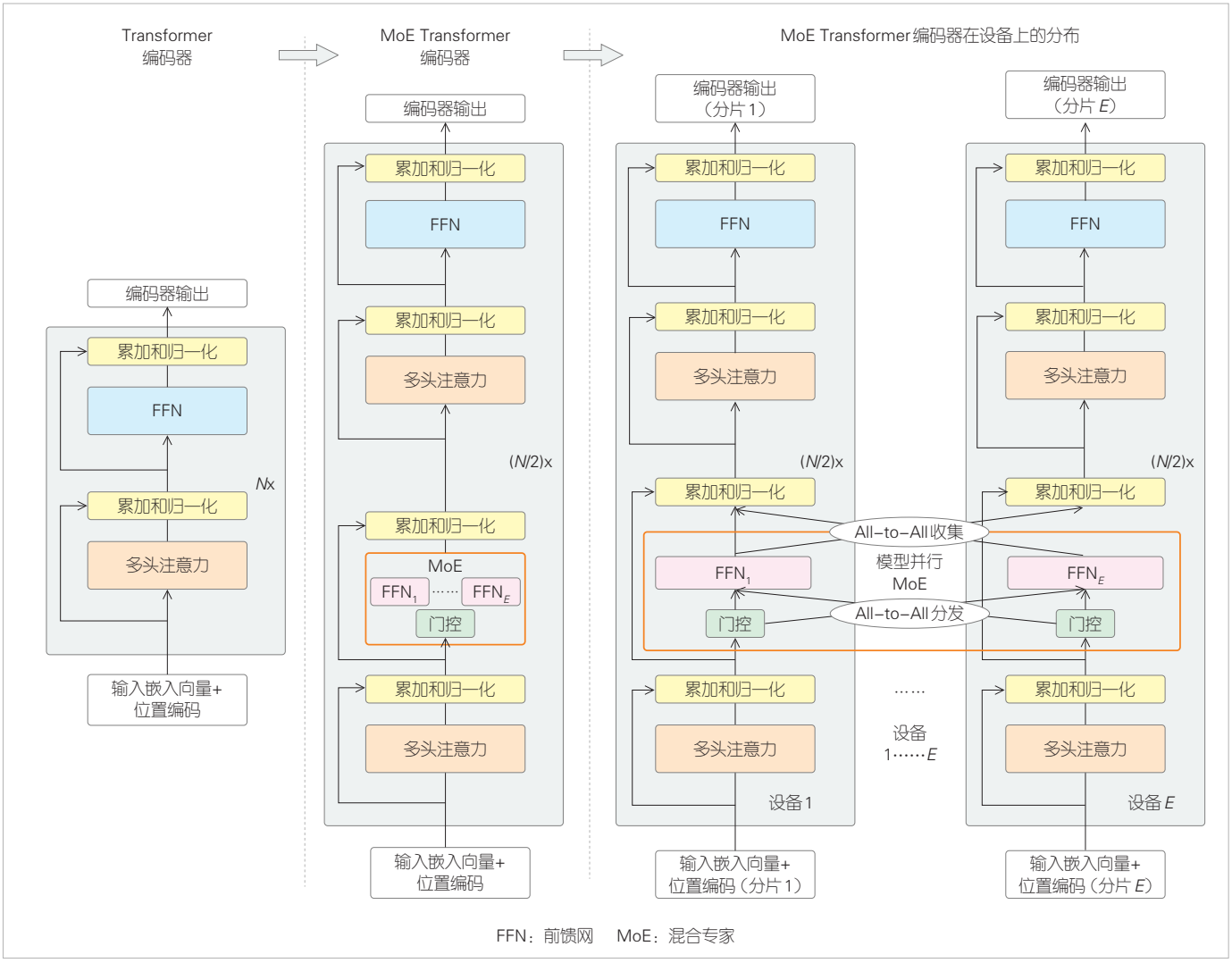
虽然Switch层选择的专家数减小了，但模型的性能却比普通MoE模型更高。这说明在大语言模型场景，设备内存相对稀缺。如果专家接收过多的tokens，会因内存不足而丢弃token，这样会造成模型性能下降。

Switch Transformers使用数据并行、模型并行、专家并行，在较大的集群上进行分布式训练，得到了参数量为3.95×10^{11}、1.571×10^{12}的大语言模型。

4.2 状态空间模型

Transformer的优点是能够并行计算一个序列所有tokens间的相关信息；缺点是序列的长度有限，对于超过上下文窗口长度的序列（如音频、视频）难以计算。SSM可以解决此问题。

SSM是控制论里的概念，其作用是对一个输入连续时间



▲图5 单设备混合专家与多设备GShard 专家并行[51]

信号进行处理，得到一个输出连续时间信号，信号的长度没有限制。调整SSM的参数，可以使输出信号成为输入信号的特征。音频、视频数据本身就是连续时间信号，适合用SSM处理。但文本序列是离散的，且信息密度大，不能直接用SSM处理，因此需要对SSM做离散化及一些调整。

语言模型加入调整后的SSM，有些取得了较好的效果，目前常见的SSM语言模型有线性注意力[53]、S4[54]、H3[55]、Hyena[56]、RetNet[57]、接受权重键值（RWKV）[58]、Mamba[59]等。本文中，我们主要介绍S4和Mamba。

4.2.1 S4模型

S4模型的连续SSM公式为：

$$\boldsymbol{h}'(t) = \boldsymbol{A}\boldsymbol{h}(t) + \boldsymbol{B}\boldsymbol{x}(t), \tag{11}$$

$$\boldsymbol{y}(t) = \boldsymbol{C}\boldsymbol{h}(t), \tag{12}$$

其中，$\boldsymbol{x}(t)$是输入信号，$\boldsymbol{h}(t)$是隐藏状态，$\boldsymbol{h}'(t)$是$\boldsymbol{h}(t)$的变化率，$\boldsymbol{y}(t)$是输出信号，$\boldsymbol{A}$、$\boldsymbol{B}$、$\boldsymbol{C}$是参数矩阵。

SSM离散化后才能用于处理文本序列，具体做法是根据步长Δ将$\boldsymbol{A}$、$\boldsymbol{B}$离散化为$\bar{\boldsymbol{A}}$、$\bar{\boldsymbol{B}}$，得离散化SSM公式为：

$$\boldsymbol{h}_t = \bar{\boldsymbol{A}}\boldsymbol{h}_{t-1} + \bar{\boldsymbol{B}}\boldsymbol{x}_t, \tag{13}$$

$$\boldsymbol{y}_t = \boldsymbol{C}\boldsymbol{h}_t, \tag{14}$$

其中，$\boldsymbol{x}_t$是输入序列中的第t个token，$\boldsymbol{h}_t$是第t个隐藏状态，$\boldsymbol{y}_t$是第t个输出。

以上公式是递归的，序列长度不受限制。如果已知序列长度，可以进行展开，得：

$$\boldsymbol{y}_k = \boldsymbol{C}\bar{\boldsymbol{A}}^k\bar{\boldsymbol{B}}\boldsymbol{x}_0 + \boldsymbol{C}\bar{\boldsymbol{A}}^{k-1}\bar{\boldsymbol{B}}\boldsymbol{x}_1 + \cdots + \boldsymbol{C}\overline{\boldsymbol{AB}}\,\boldsymbol{x}_{k-1} + \boldsymbol{C}\bar{\boldsymbol{B}}\boldsymbol{x}_k。 \tag{15}$$

此式可转换为卷积形式：

$$\bar{K} = (C\bar{B}, C\overline{AB}, ..., C\bar{A}^{k}\bar{B}),$$

$$y = x * \bar{K}。 \quad (16)$$

因此，S4模型在训练时可以使用卷积形式计算SSM，在推理时可以使用递归形式计算SSM。

4.2.2 Mamba模型

Mamba在S4的基础上做了3点改进：

1）对输入信息有选择的处理

序列建模的一个基本问题是把上下文压缩成更小的状态。从这个角度来看，注意力机制虽然效果好，但效率不是很高。因为它不压缩上下文，而是全部存储（也就是KV Cache），这直接导致训练和推理消耗算力大。RNN压缩全部上下文，但压缩率随着序列长度增加而增大，最终会丢失长期信息，因此推理和训练效率高，但性能较差。Mamba的解决办法是，让模型对上下文有选择的处理，丢弃不重要信息，对重要信息进行压缩保留，在计算效率和保留信息两方面取得平衡[59]。

S4的参数矩阵B、C和步长Δ，是所有tokens共享的，每个token计算使用相同的一套B、C、Δ。Mamba的改进点是，用输入序列x（长度为L）经过3个全连接层，生成L套不同的B、C、Δ，每个token计算使用一套，从而实现对信息的选择性处理，关注或忽略特定的内容。

2）硬件感知算法

Mamba在GPU上对SSM算法做了优化，利用扫描而不是卷积来递归计算模型，尽量减少不同级别的GPU内存层次结构之间的IO访问。

3）更简单的架构

Mamba架构是SSM与Transformer的多层感知机（MLP）块的结合，两者相互融合，而不是重叠，形成了一个更加简单的结构，如图6所示。

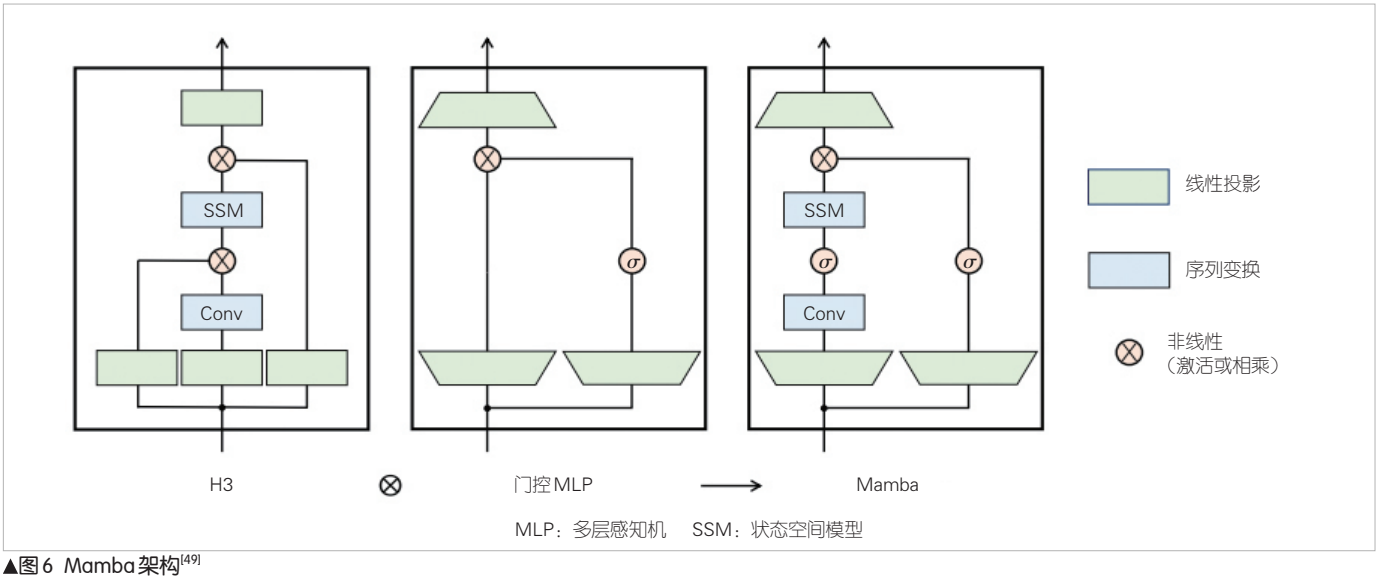
Mamba在合成、音频、基因、语言等多个领域的任务上都表现出良好的性能，并能处理超过1 M长度的序列。

5 结束语

大语言模型是神经网络模型长期发展的成果，特别是Transformer结构的注意力机制与大算力结合，产生了涌现能力，其文本理解、生成、对话能力已接近人类。但Transformer的一些固有特点也导致了其性能的不足，如生成式模式、自注意力造成消耗算力过大，上下文长度有限，硬件亲和性差造成推理性能低等。目前，大语言模型算法的演进是渐进的优化，对Transformer结构做局部的修改，以改善这些不足。未来的演进可能仍然是渐进的，也可能会产生革命性的创新，即提出全新的模式，代替生成式模式和自注意力，增强逻辑推理能力和世界模型认知，更加接近AGI。

参考文献

[1] OpenAI. Introducing ChatGPT [EB/OL]. [2024-03-10]. https://openai.com/blog/chatgpt/
[2] OpenAI. GPT-4 technical report [EB/OL]. [2024-03-10]. DOI: 10.48550/ARXIV.2303.08774
[3] VASWANI A , SHAZEER N, PARMAR N, et al. Attention is all you need [EB/OL]. (2017-06-12) [2023-08-02]. https://arxiv.org/abs/1706.03762
[4] 车万翔, 刘挺. 自然语言处理新范式：基于预训练模型的方法 [J]. 中兴通讯技术, 2022, 27(2): 3-9. DOI: 10.12142/ZTETJ.202202002



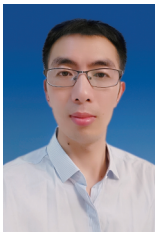
▲图6 Mamba架构[49]

[5] 王海宁. 自然语言处理技术发展 [J]. 中兴通讯技术, 2022, 27(2): 59–64. DOI: 10.12142/ZTETJ.202202009
[6] BENGIL Y, REJEAN D, PASCAL V. A neural probabilistic language model [EB/OL]. (2003–03–01) [2024–03–10]. https://dl. acm. org/doi/10.5555/944919.944966
[7] MIKOLOV T, KARAFIAT M, BURGET L, et al. Khudanpur. Recurrent neural network based language model [EB/OL]. [2024–03–10]. https://www.fit.vut.cz/research/publication/9362/.en
[8] HOCHREITER S, SCHMIDHUBER J. Long short–term memory [J]. Neural computation, 1997, 9(8): 1735–1780. DOI: 10.1162/neco.1997.9.8.1735
[9] PETERS M, NEUMANN M, IYYER M, et al. Deep contextualized word representations [C]//Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics. Association for Computational Linguistics, 2018: 2227–2237. DOI: 10.18653/v1/n18–1202
[10] DEVLIN J, CHANG M, LEE K, et al. BERT: pre–training of deep bidirectional transformers for language understanding [EB/OL]. (2018–11–11)[2024–03–10]. https://arxiv.org/abs/1810.04805
[11] KAPLAN J, MCCANDLISH S, HENIGHAN T, et al. Scaling laws for neural language models [EB/OL]. [2024–03–10]. https://arxiv.org/abs/2001.08361
[12] HOFFMANN J, BORGEAUD S, MENSCH A, et al. Training compute–optimal large language models [EB/OL]. [2024–03–10]. https://arxiv.org/abs/2001.08361
[13] BROWN T B, MANN B, RYDER N, et al. Language models are few–shot learners [C]//Proceedings of the 34th International Conference on Neural Information Processing Systems. ACM, 2020: 1877–1901. DOI: 10.5555/3495724.3495883
[14] RADFOR A, NARASIMHAN K, SALIMANS T, et al. Improving language understanding by generative pre–training [EB/OL]. [2024–03–10]. https://cdn. openai. com/research–covers/language–unsupervised/language_understanding_paper.pdf
[15] RADFORD A, WU J, CHILD R, et al. Language models are unsupervised multitask learners [EB/OL]. (2020–05–28) [2024–03–10]. https://arxiv.org/abs/2005.14165
[16] JACOBS R A, JORDAN M I, NOWLAN S J, et al. Adaptive mixtures of local experts [J]. Neural computation, 1991, 3(1): 79–87. DOI: 10.1162/neco.1991.3.1.79
[17] KIM S, HOOPER C, WATTANAWONG T, et al. Full stack optimization of transformer inference: a survey [EB/OL]. (2023–02–27)[2024–03–10]. https://arxiv.org/abs/2302.14017
[18] TOUVRON H, MARTIN L, STONE K, et al. Llama 2: open foundation and fine–tuned chat models [EB/OL]. (2023–05–18) [2024–03–10]. https://arxiv.org/abs/2307.09288
[19] CHOI J, LI H, KIM B, et al. Accelerating transformer networks through recomposing softmax layers [EB/OL]. [2024–03–10]. https://ieeexplore.ieee.org/document/9975410
[20] POPE R, DOUGLAS H, CHOWDHERY A, et al. Efficiently scaling Transformer inference [EB/OL]. (2022–11–09) [2024–03–10]. https://arxiv.org/abs/2211.05102
[21] 马子轩,翟季冬,韩文弢,等,郑纬民. 高效训练百万亿参数预训练模型的系统挑战和对策 [J]. 中兴通讯技术, 2022, 27(2): 51–58. DOI: 10.12142/ZTETJ.202202008
[22] SHOEYBI M, PATWARY M, PURI R, et al. Megatron–LM training multi–billion parameter language models using model parallelism [EB/OL]. (2019–09–17) [2024–03–13]. https://arxiv.org/abs/1909.08053
[23] NARAYANAN D, SHOEYBI M, CASPER J, et al. Efficient large–scale language model training on GPU Clusters using megatron–LM [EB/OL]. (2021–04–09) [2024–03–10]. https://arxiv. org/abs/2104.04473
[24] KORTHIKANTI V, CASPER J, LYM S, et al. Reducing activation recomputation in large transformer models [EB/OL]. (2022–05–10)[2024–03–10]. https://arxiv.org/abs/2205.05198
[25] SHAZEER N. Fast transformer decoding: one write–head is all you need [EB/OL]. (2019–11–06)[2024–03–10]. https://arxiv.org/abs/1911.02150
[26] AINSLIE J, LEE–THORP J, DE JONG M, et al. GQA: training generalized multi–query transformer models from multi–head checkpoints [C]//Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing. Association for Computational Linguistics, 2023: 4895–4901. DOI: 10.18653/v1/2023.emnlp–main.298
[27] STERN M, SHAZEER N M, USZKOREIT J, et al. Blockwise parallel decoding for deep autoregressive models [EB/OL]. [2023–03–10]. https://arxiv.org/pdf/1811.03115.pdf
[28] LEVIATHAN Y, KALMAN M, MATIAS Y. Fast inference from transformers via speculative decoding [C]//Proceedings of the 40th International Conference on Machine Learning. ACM, 2023: 19274–19286. DOI: 10.5555/3618408.3619203
[29] HE B, HOFMANN T. Simplifying transformer blocks [EB/OL]. (2023–11–03)[2023–03–10]. https://arxiv.org/abs/2311.01906
[30] DAO T, FU Y D, FU Y , et al. Flashattention: fast and memory–efficientexact attention with io–awareness. [EB/OL]. [2023–03–10].https://arxiv. abs/2205.14135
[31] DAO T. Flashattention–2: Faster attention with better parallelism and work partitioning [EB/OL]. (2023–06–17) [2024–03–10]. https://arxiv.org/abs/2307.08691
[32] DAO T, HAZIZA D, MASSA F, et al. Flash–decoding for long–context inference [EB/OL]. (2023–06–17) [2024–03–10]. https://pytorch.org/blog/flash–decoding/
[33] HONG K, DAI G, XU J, et al. FlashDecoding++ : faster large language model inference on GPUs [EB/OL]. (2023–11–02) [2024–03–10]. https://arxiv.org/abs/2311.01282
[34] KWON W, LI Z H, ZHUANG S Y, et al. Efficient memory management for large language model serving with PagedAttention [EB/OL]. (2023–09–12) [2024–03–10]. https://arxiv.org/abs/2309.06180
[35] WAN Z, WANG X, LIU C, et al. Efficient large language models: a survey [EB/OL]. (2023–12–06) [2024–03–10]. https://arxiv.org/abs/2312.03863
[36] DETTERS T, LEWIS M, BELKADA Y. LLM. int8(): 8–bit matrix multiplication for transformers at scale [EB/OL]. (2022–08–15) [2024–03–10]. https://arxiv.org/abs/2208.07339
[37] FRANTAR E, ASHKBOOS S, HOEFLER T, et al. GPTQ: accurate post–training quantization for generative pre–trained transformers [EB/OL]. (2022–10–31) [2024–03–10]. https://arxiv.org/abs/2210.17323
[38] CHEE J, CAI Y, KULESHOV V, et al. QuIP: 2–bit quantization of large language models with guarantees [EB/OL]. (2022–06–25) [2024–03–10]. https://arxiv.org/abs/2307.13304
[39] LIN J, TANG J, TANG H, et al. AWQ: activation–aware weight quantization for LLM compression and acceleration [EB/OL]. (2023–10–13)[2024–03–10]. https://arxiv.org/abs/2306.00978
[40] LEE C, JIN J, KIM T, et al. OWQ: Lessons learned from activation outliers for weight quantization in large language models [EB/OL]. (2023–06–04) [2024–03–10]. https://arxiv. org/abs/2306.02272
[41] DETTERS T, SVIRSCHEVSKI R, EGIAZARIAN V, et al. SpQR: a sparse–quantized representation for near–lossless LLM weight compression [EB/OL]. (2023–06–05) [2024–03–10]. https://arxiv.org/abs/2306.03078
[42] KIM J Y, HENRY R, FAHIM R, et al. FineQuant: unlocking efficiency with fine–grained weight–only quantization for LLMs [EB/OL]. (2023–08–16) [2024–03–10]. https://arxiv. org/abs/2308.09723
[43] YAO Z, AMINABADI Y R, ZHANG M, et al. ZeroQuant: efficient

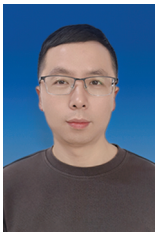
and affordable post-training quantization for large-scale transformers [EB/OL]. (2022-06-04)[2024-03-10]. https://arxiv.org/abs/2206.01861
[44] XIAO G X, LIN J, SEZNEC M, et al. SmoothQuant: accurate and efficient post-training quantization for large language models [EB/OL]. (2022-11-18) [2023-03-10]. https://arxiv. org/abs/2211.10438
[45] GUO C, TANG J, HU W M, et al. OliVe: accelerating large language models via hardware-friendly outlier-victim pair quantization [EB/OL]. (2022-11-18) [2024-03-15]. https://arxiv.org/abs/2304.07493
[46] YUAN Z H, NIU L, LIU J W, et al. RPTQ: reorder-based post-training quantization for large language models [EB/OL]. (2023-04-03)[2024-03-15]. https://arxiv.org/abs/2304.01089
[47] LIU J, GONG RH, WEI X Y, et al. QLLM: accurate and efficient low-bitwidth quantization for large language models [EB/OL]. (2023-12-12)[2024-03-12]. https://arxiv.org/abs/2310.08041
[48] WEI X Y, ZHANG Y C, LI Y H, et al. Outlier suppression+ : accurate quantization of large language models by equivalent and optimal shifting and scaling [EB/OL]. (2023-04-18) [2024-03-12]. https://arxiv.org/abs/2304.09145
[49] GU A, DAO T, ERMON S, et al. Hippo: recurrent memory with optimal polynomial projections [EB/OL]. (2020-08-17) [2024-03-10]. https://arxiv.org/abs/2008.07669
[50] SHAZEER M N, MIRHOSEINO A, ZAZIARZ M, et al. Outrageously large neural networks: the sparsely-gated mixture-of-experts layer [EB/OL]. (2017-01-23) [2024-03-10]. https://arxiv.org/abs/1701.06538
[51] LEPIKHIN D, LEE H J, XU Y Z, et al. GShard: scaling giant models with conditional computation and automatic sharding [EB/OL]. (2020-01-30) [2024-03-10]. https://arxiv. org/abs/2006.16668
[52] FEDUS W, ZOPH B, NOAM M. Switch Transformers: scaling to trillion parameter models with simple and efficient sparsity [EB/OL]. (2021-01-11) [2024-03-10]. https://arxiv. org/abs/2101.03961
[53] KATHAROPOULOS A, VYAS A, PAPPAS N, et al. Transformers are RNNs: fast autoregressive transformers with linear attention [EB/OL]. (2020-06-29) [2024-03-11]. https://arxiv. org/abs/2006.16236
[54] GU A, GOEL K, RE C. Efficiently modeling long sequences with structured state spaces [EB/OL]. (2021-10-31) [2024-03-12]. https://arxiv.org/abs/2111.00396
[55] DAO T, FU Y D, SAAB K K, et al. Hungry hungry hippos: towards language modeling with state space models [EB/OL]. (2022-12-28)[2024-03-12]. https://arxiv.org/abs/2212.14052
[56] POLI M, MASSAROLI S, NGUYEN E, et al. Hyena hierarchy: towards larger convolutional language models [EB/OL]. (2023-02-21)[2024-03-12]. https://arxiv.org/abs/2302.10866
[57] SUN Y T, DONG L, HUANG S H, et al. Retentive network: a successor to transformer for large language models [EB/OL]. (2023-07-17)[2024-03-12]. https://arxiv.org/abs/2307.08621
[58] PENG B, ALCAIDE E, ANTHONY Q, et al. RWKV: reinventing RNNs for the transformer era [EB/OL]. [2024-03-12]. https://aclanthology.org/2023.findings-emnlp.936/
[59] GU A, DAO T. Mamba: linear-time sequence modeling with selective state spaces [EB/OL]. (2023-12-01) [2024-03-12]. https://arxiv.org/abs/2312.00752

作 者 简 介

朱炫鹏，中兴通讯股份有限公司无线资深专家；主要研究方向为深度学习算法、计算机视觉、大语言模型。



姚海东，中兴通讯股份有限公司无线资深专家；主要从事深度学习、大模型网络架构及编译转换技术研究和设计。



刘隽，中兴通讯股份有限公司无线资深专家；主要研究方向包括深度学习算法、AI编译器、AI加速器架构设计和模拟仿真等。

熊先奎，中兴通讯股份有限公司无线首席架构师、“智算”技术委员会前瞻组组长；长期从事计算系统和体系结构、先进计算范式以及异构计算加速器研究工作；曾主导过中兴通讯ATCA先进电信计算平台、服务器存储平台、智能网卡和AI加速器等系统架构设计。

大模型训练技术综述

A Survey on Large Model Training Technologies

田海东/TIAN Haidong，张明政/ZHANG Mingzheng，
常锐/CHANG Rui，童贤慧/TONG Xianhui

（中兴通讯股份有限公司，中国 深圳 518057）
(ZTE Corporation, Shenzhen 518057, China)

DOI：10.12142/ZTETJ.202402004
网络出版地址：http://kns.cnki.net/kcms/detail/34.1228.TN.20240419.0912.002.html
网络出版日期：2024-04-20
收稿日期：2024-03-02

摘要：实现高效训练已成为影响大模型应用普及的关键要素之一。按照数据准备、数据加载、模型初始化及评估、训练并行、模型状态保存的一般训练流程，对大模型高效训练的主要技术进行分析和论述。面对大模型规模的持续增长、数据处理类型的扩展，现有大模型训练技术仍存在较大的优化空间。认为未来大模型训练重点研究方向包括以数据为中心、数据加载智能化和异构加速、网络通信领域定制、训练并行及自动化。

关键词：大模型；数据准备；数据加载；模型初始化；模型评估；训练并行；训练网络；检查点

Abstract: Achieving efficient training has become one of the key factors affecting the popularization of large model applications. The main technologies of efficient training of large models are analyzed and discussed according to the general training process of data preparation, dataloader, model initialization and evaluation, training parallelism, and model state preservation. In the face of the continuous growth of large model scale and the expansion of data processing types, there is still a large room for optimization of existing large model training technologies. In the future, the key research directions of large model training include data-centric, intelligent dataloader and heterogeneous acceleration, customization in the field of network communication, training parallelism and automation.

Keywords: large model; data preparation; dataloader; model initialization; model evaluation; training parallelism; training network; checkpoint

引用格式：田海东, 张明政, 常锐, 等. 大模型训练技术综述 [J]. 中兴通讯技术, 2024, 30(2): 21-28. DOI: 10.12142/ZTETJ.202402004
Citation: TIAN H D, ZHANG M Z, CHANG R, et al. A survey on large model training technologies [J]. ZTE technology journal, 2024, 30(2): 21-28. DOI: 10.12142/ZTETJ.202402004

近年来，深度学习[1-5]领域取得了重大进展，特别是以ChatGPT为代表的大语言模型（下文统称大模型），在人机问答、内容生成领域展示出了人工智能的强大威力，让人们看到了通用人工智能（AGI）的曙光。未来大模型有望成为一项引发新一代工业革命和促进社会发展的变革性技术。然而，大模型的训练对数据准备与预处理、并行训练过程计算访存效率、过程状态保存与故障恢复等方面都有着苛刻的要求。如何实现大模型的高效训练，已成为学术界和工业界共同的研究热点。本文中，我们将按照模型训练的一般流程，并聚焦主要训练过程，对大模型高效训练的主要技术进行分析和论述。

1 数据准备

按照所能处理的数据类型，大模型可分为两类：语言类大模型[6-8]和多模态类大模型。语言类大模型的文本数据来源不仅包括网页、对话文本、书籍等通用数据，还包含多种语言的语料库、科技论文、代码等专用数据。多模态类大模型的数据一般来源于网页和专用数据集，常见形式是图片文本对。语言类大模型训练数据的准备流程一般包括收集、过滤、去重、隐私去除、分词。在转化为向量数据后，相关数据被加载到图形处理器（GPU）中进行训练[9]。而多模态类大模型则需要对图片、视频、语音等非文本的数据对象，先进行适当的解码、缩放、裁剪、归一化处理，再通过特征提取将其转化为向量数据。数据准备的主要方法如下：

1）去重和过滤

由于来源数据良莠不齐，大模型中会不可避免的引入噪声、冗余、无关甚至有害的数据，有些数据还会涉及个人隐私。文献[10]研究发现，反复重复一小部分数据可能会对系统性能造成巨大危害。这是因为重复数据的训练会导致系统从零开始训练和微调性能变差。所以删除重复的数据，可使

系统使用更少的训练步骤来实现相同或更好的准确性。如何定量和定性地理解数据集本身就是一个研究挑战。文献[11]提出了后缀数组子串处理和相似度匹配算法两种可扩展的技术，以检测和删除重复的训练数据。

RefindedWeb[12]对数据集的处理方式更为激进，采用了Bloomfilter和Simhash近似去重，这导致删除率远高于其他方法。RefindedWeb同时证明，只用网页数据并通过严格的过滤、去重和脚本处理等手段，同样可以获取大模型所需的训练语料。

文献[13]则引入数据年龄、质量及毒性危害程度、领域组成等更多维度的量化评价指标，分析对大模型训练的影响，为模型训练数据准备提供了指导方法。

2）建立数据生产体系

从零开始预训练大模型需要高昂的成本。开源的大模型通常不附带开源的数据集。因此，如何高效体系化地收集处理数据显得尤为重要。Ziya2[14]构建了完整的数据生产体系，包括预处理数据、自动评分、基于规则的过滤、消除重复内容和评估数据等子任务。RefinedWeb[12]则提出宏数据优化（MDR）处理数据的思想，侧重于去重和过滤。

3）隐私和安全

一般来说，模型泄露数据的主要原因是过拟合。此时模型会记住数据集中的数据。在大规模数据上进行训练也会存留“记忆”。文献[15]基于此在GPT2上进行攻击验证，认为在数据准备、增加噪声、微调各阶段中都需要有防止数据泄露的措施。删除数据集敏感数据是其中一种方法。知识遗忘作为一种替代方法，也可以用来降低语言模型的隐私风险[16]。文献[15]在执行遗忘时不仅能提供更强的隐私保障，还能保障模型性能不下降，并发现一次性忘记许多样本会导致显著的模型性能下降，而顺序遗忘数据可以缓解这种情况。

2 数据加载

大模型训练需要大量数据、资源和时间，它涉及服务器中所有资源的综合利用，如图1所示。数据加载是指，从存储器中读取数据，根据不同的模型训练要求，对读取的数据设置不同的预处理规则。存储介质、缓存大小、用于获取和预处理数据的CPU线程数量等因素，都会影响到数据加载的性能。

理想情况下，数据加载部分需要稳定地将预处理后的数据发送给GPU，以便GPU能够持续进行数据计算，充分发挥计算能力。但在实际工作中，主流的训练框架如PyTorch、TensorFlow等，在进行数据加载和数据预处理时都存在性能瓶颈。我们将这些瓶颈统称为数据停滞。

目前有多种解决方案都在研究如何解决模型训练过程中的数据停滞问题，主要包括以下几种：

1） 缓存策略

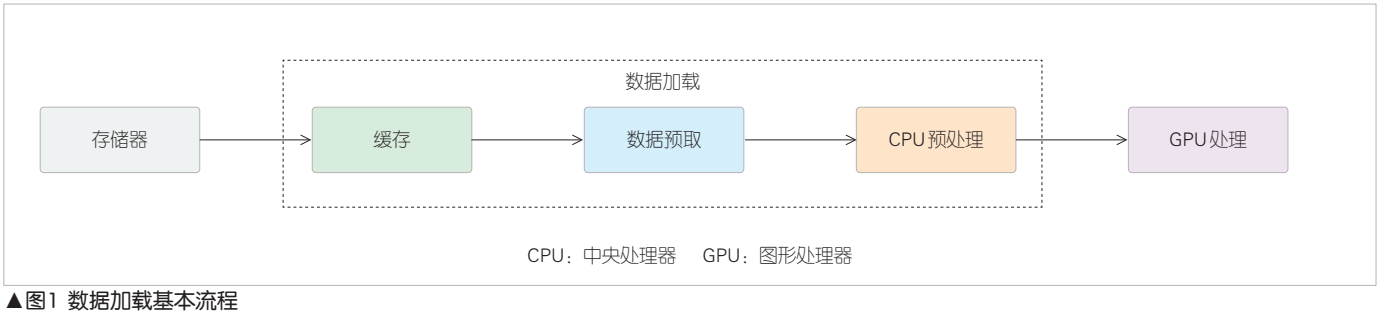
很多训练平台（如PyTorch等）提供了DataLoader，支持提前将数据从存储器读取到内存中，并且通常依赖操作系统的页面缓存技术来缓存重复访问的原始训练数据。通过实验数据分析，数据集的访问模式与操作系统缓存替换策略并不一致。操作系统的页面缓存机制在提升训练效率方面并不明显。

分析大模型训练有其独特的数据访问模式。每个训练周期内系统会以随机方式遍历一次数据集中的所有数据样本。而基于样本重要性、动态打包和多任务处理机制等高速缓存置换算法[17]的研究，则能够有效减少数据停滞时间，提升训练速度。

2） 分布式数据加载

分布式技术可以将训练任务分配到多个计算节点上进行并行计算，以提高训练效率。分布式训练将数据分成多个批次，然后在不同的存储节点上并行加载数据，以减少数据加载等待时间。对于分布式训练过程中存在的同一份数据在多个节点上重复缓存的情况[18]，如果缓存之间缺乏协调，分布式训练就容易受到存储接口的限制。此时，可通过设置数据流策略将存储接口操作与计算操作并行化处理，来进一步提高训练效率。

另外，当参数量很大时，数据通信量也可能会成为模型



▲图1 数据加载基本流程

训练的瓶颈，此时需要通过通信和计算重叠等方式降低通信等待时间[19]。

3） 数据预处理卸载

将数据预取和数据预处理的过程卸载到GPU等设备上，构建高效的数据流水线，可以减少数据加载的等待时间，让训练过程更加流畅。DALI[20]是专门为GPU优化的数据加载库，通过在GPU上运行数据处理管道从而加速数据预处理过程。DALI的性能优于传统的数据流水线，但代价是占用了GPU有限的计算和内存资源。发掘训练模式和数据特征，进行数据压缩，有助于实现GPU内存优化[21]。目前PyTorch等主流深度学习框架都已经支持DALI的使用。

此外，数据流分析工具如DS-Analyzer[22]，可以针对具体的训练场景，精确发现数据流中的性能问题。对数据停滞进行预测和分析，能够为提高训练效率指明具体的改进方向，例如：CoorDL[22]验证了在特定情况下，模型训练能够获得比DALI更高的资源利用率和更好的性能。

3 模型初始化及评估

这一章节中我们主要探讨两个问题：如何做好预训练大模型的初始化准备，以及如何在训练过程中确定更好的模型评估指标，具体包括模型规模的选择、模型超参数的初始化设置、模型的评估等方面。

3.1 模型规模选择

在进行预训练之前，了解大模型的扩展法则可以帮助我们很好地平衡模型大小、数据的规模以及计算量之间的关系。这里我们给出两种大模型的扩展法则，具体说明如下：

1） KM扩展法则

Decoder-only的模型算力有如下关系：

$$C \sim \tau T = 6PD, \tag{1}$$

其中，C表示模型的总计算量，τ表示吞吐量，T表示训练时间，P表示模型的参数量，D表示token数。

2020年OpenAI[23]团队首次通过实验给出了模型性能和模型参数量、数据规模、模型计算量之间的幂律关系。由幂律关系发现，在相同算力下模型的参数量更重要。

2） Chinchilla扩展法则

Google的DeepMind团队提出了Chinchilla扩展法则[24]。他们在一个更大范围（7 000万到160亿参数）的模型和更大范围（50亿到5 000亿tokens）的数据条件下探讨上述KM扩展法则，得出一个系数不同的幂律关系，即Chinchilla扩展法则。Chinchilla扩展法则认为模型大小和数据大小应该以同等的比例增加。

3.2 模型初始化

大模型训练是一个高度实验性的过程，需要承担较高的试错成本。过程中会涉及大量的超参数设置，包括权重初始化、归一化方法、激活函数、位置嵌入、学习率、优化器等。这里我们介绍一些常见的初始化设置。

1） 权重初始化

合理的初始化权重可以帮助模型更快地收敛，使模型拥有更好的性能。最常见的就是高斯噪声初始化。为解决深层Transformer收敛困难的问题，2019年XU Q.[25]认为收敛困难的原因是层归一化（LN）和残差连接相互影响导致梯度消失，提出了利普希茨约束参数初始化（LRI）的参数归一化方式。与此同时，ZHANG B.[26]提出了一种参数初始化方式DS-Init。该方法通过在初始化阶段减少模型参数的方差，来减少残差连接输出的方差，从而缓解反向传播过程中数据通过正则化层时的梯度问题。2020年，HUANG X. S.[27]提出了一种参数初始化策略T-fixup。该策略可以使模型参数在没有预热（WarmUp）和层归一化的情况下仍能够更新收敛。

2） 归一化

训练不稳定是大模型面临的一个难题。LN[28]被广泛应用到Transfromer架构中。LN的位置对于大模型的性能至关重要。2020年，XIONG R.等提出了Pre-LN[29]。相对于一般Tranformer中的LN，研究人员将LN这一阶段提前，解决了学习率WarmUp阶段超参敏感问题，同时优化了收敛过程速度慢等问题，但这也带来了一定的模型性能损失。Chinchilla[24]则采用RMS Norm[30]的方式，取消了传统LN上的均值计算，在训练速度和性能方面都具有优势。

3） 激活函数

为了获得良好性能，前馈网络需要设置合适的激活函数。现有大模型中，GeLU[31]被广泛使用。此外，最新的大模型如PaLM和LaMDA，使用了GeLU的变体SwiGLU[32]和GeGLU[33]，取得了更好的性能。

3.3 模型评估

在模型训练过程中（或对于那些训练好的模型），我们需要评估模型的能力。通常会有很多基准数据集可用于评估模型的逻辑推理、翻译、自然语言推理、问答等方面的能力。这里我们对常用评估方法及指标做一个介绍。

1） 文本对比

一些常规的指标，比如BLEU、ROUGE、METEOR等，

可用来衡量文本的重叠度。PYRAMID[34]可以衡量语义重叠度。对于代码生成模型，pass@k[35]是一个重要衡量指标。对于多输出的复杂模型，KoLA[36]将大模型的评价和认知层面联系起来。KoLA的评测任务由认知层级决定。认知层级包括知识记忆（KM）、知识理解（KU）、知识应用（KA）、知识创造（KC）。

当然，我们也可以从其他方面来评估模型，比如：鲁棒性（NL-Augmenter[37]的语义不变扰动）、基于计数的性别和种族bias[38]、不确定性及公平性等。

2） 自动评估和人类评估

自动评估是指利用一些小模型对大模型的结果进行评估，例如毒性评估相关（ML-based Perspective[39]）、对话系统等。

此外，我们也可以使用单纯的人类评估。人类评估是自然语言处理领域中衡量模型或算法性能的关键方法。然而，人类评估存在不稳定、可重复性低等问题，这可能导致评估结果不够准确。HUSE[40]尝试模仿人类评估的方法，并将人类评估和统计评估相结合，实现多样性评估。

4 训练并行及网络

随着深度学习模型参数和数据规模的增长，传统的单机单卡模式已无法满足大模型的训练要求。因此，我们需要基于单机多卡、多机多卡进行大模型的分布式训练。而利用计算集群，从大量数据中高效地训练出性能优良的大模型是分布式训练的首要目标。为了实现该目标，需要根据硬件资源与数据模型规模的匹配情况，来对计算任务、训练数据和模型进行划分，从而实现分布式训练并行。

4.1 常见的并行策略

这里我们介绍几种常见的并行策略，即数据并行、张量并行和流水线并行。在大模型训练的过程中，通常会将上述并行策略进行组合使用，即混合并行。

1） 数据并行

数据并行是提高训练吞吐量的基本方法之一。它将模型参数和优化器状态复制到多个GPU上，然后将整个训练语料库分配到这些GPU上。这样，每个GPU只需要处理分配给自己的数据，并执行前向和反向传播以获取梯度。在不同GPU上计算出的梯度将进一步聚合以获得整个批量的梯度，然后更新所有GPU上的模型。由于不同GPU上的梯度计算是独立进行的，数据并行机制具有高度可扩展性，因此可以通过增加GPU数量来提高训练吞吐量。以PyTorch为例，数据并行方式有：数据并行（DP）、分布式数据并行（DDP）、完全分片数据并行（FSDP）等。

2） 张量并行

张量并行专注于分解大模型的张量，将一个张量沿着特定维度分成N块。每个GPU保持整个张量的$1/N$，同时不影响整个计算图的正确性。张量并行可以通过跨GPU通信将多个GPU的输出结果聚合成最终结果。最早的张量并行方案由Megatron-LM[41]提出，它是一种高效的一维张量并行实现方法。为了平均分配计算和内存负荷，Colossal-AI[42]把张量沿着两个维度进行切分，这就是二维张量并行。除此之外，Colossal-AI还可以支持更高维度的张量并行。

3） 流水线并行

流水线并行是指将大模型的不同层分配到不同的GPU上，以降低单个GPU的显存消耗。然而，传统流水线并行方式的GPU空泡率较高。针对该问题，Gpipe[43]提出了micro-batch的方式以减少空泡率。PipeDream[44]在Gpipe的基础上使用1F1B的方式优化流水线并行策略，以减少流水的空闲时间和显存。在后续的一些研究中，PipeDream-2BW和PipeDream-Flush[45]等基于原始的PipeDream做了进一步的改进。

4） 序列并行

序列并行是指将序列这个维度划分到不同的GPU上进行并行计算。例如LI S.[46]为解决大模型输入序列长度的限制，将输入序列分割到多个GPU上，并提出环自注意力，将环状通信与自注意力相结合，可以处理超过1.14×10^5的长度序列。

同样地，Megatron-LM[41]在进行张量并行的时候，将LayerNorm和Dropout的输入按长度进行了划分，使得各GPU只需要完成一部分Dropout和LayerNorm操作，并使用选择性激活重计算以减少激活显存。

4.2 其他并行优化策略

1） 基于显存的优化技术ZeRO

ZeRO[47]是由DeepSpeed提出的在数据并行过程中降低显存的一种技术。该技术可以使得单个GPU的显存占用随着GPU的数量增加而线性下降。ZeRO一共提供了3种解决方案：优化器分区、梯度分区和参数分区。前2种方案不会带来新的通信开销，第3种方案会增加50%的通信开销。与此同时，ZeRO提供ZeRO-R在数据并行节点间划分激活，来减少激活的显存开销。

2） 基于模型结构的并行

混合专家（MoE）模型是一种基于模型结构的并行架构。它将大模型拆分成多个小模型（专家模型），在每一轮

迭代中根据样本决定一部分专家用于计算，并引入可训练的门机制保证计算能力的优化。Gshard[48]首次将MoE应用到Transformer上，将间隔的前馈神经网络（FFN）替换成MoE。Switch Transformers[49]是在T5模型上应用MoE设计的，它简化了MoE的路由算法，具有门机制，即每次只推选一个专家。

在后续的研究中，微软的Deepspeed-MoE[50]提出了一种端到端的MoE训练和推理解决方案，有效减少了MoE模型的大小。Faster-MoE[51]提升了MoE模型分布式训练效率，与大模型优化策略（ZeRO、Gshard等）相比获得了显著的性能加速。

3） 自动并行

自动并行的目标是：用户给定一个模型和所使用的机器资源后，系统能够自动地帮用户选择一个较好或者最优的并行策略来实现高效执行。可以说，自动并行是分布式并行的终极目标，它能够减少或避免工程师手动设置分布式并行策略。自动并行分为半自动和全自动两种模式。其中，半自动模式是指用户自己指定张量的切分方式，如Gshard[48]；全自动模型是指由框架自适应选择切分方式，如Flexflow[52]等提到的全自动并行切分方案。

目前许多训练框架如PyTorch、TensorFlow等实现了自动并行。Alpa[53]通过自动搜索intra-op的调度和inter-op的切分方式，几乎兼顾了所有的并行策略，是自动并行的集大成者。

4.3 训练网络

大模型训练集群的各个计算节点之间通过网络进行互联。网络性能直接决定节点间的通信效率，进而影响整个训练集群的吞吐和性能。随着模型规模的持续增大，大模型训练网络面临着多种挑战：大规模扩展、高通量和低延迟等，除了需要带宽增强等基础技术的支撑[54]，还需从互联协议、网络拓扑、在网计算等多个方面进行优化。

1） 互联协议

训练网络互联技术通常分为两类：总线互联协议（包括：NVLink、CXL等）和网络互联协议（包括：RoCE、Infiniband等）。其中，前者用于计算芯片之间短距离、小规模的互联，而后者则用于计算节点之间长距离、大规模的数据通信。随着总线和网络技术的发展，这两类技术已出现逐渐融合的趋势。例如，英伟达的NVLink 4.0已经可以支持256个GPU的互联，CXL在其最新的规范中也明确将支持机架间的互联。

2） 网络拓扑

大模型训练对网络拓扑的扩展性、可靠性和成本等都提出了更高要求。在高性能计算的发展中，Torus无疑占据了重要的位置。相比于Torus结构，胖树网络路由算法更容易实现，网络性能相对更出色。但是胖树网络在扩展至更大规模时需增加网络层数，从而导致链路数随之指数增长，这会大大增加网络成本。Dragonfly由J. JIM等在2008年提出[55]。它的特点是网络直径小、成本较低，在高性能计算方面有着显著优势。然而，面对整体网络节点的增多，Dragonfly等网络结构依然面临网络连线复杂、网络设计成本高、所需全局光纤数多等挑战。

除了上述拓扑结构，MIT和META的rail-only[56]等还提出了定制化拓扑结构。这些拓扑结构专门针对大模型的通信需求进行设计，目的是在提升性能的同时显著降低成本。

3） 在网计算

在网计算通过网络交换侧和端侧设备的协同，利用网络内部的硬件计算引擎，在网络通信过程中实现复杂操作的卸载。基于树状聚合的机制，在网计算可以同时支持多个集合操作。以典型的AllReduce算子为例，传统的通信交互复杂度为$O(\log N)$（N代表网络节点规模），启动在网计算功能后，其交互复杂度变为$O(C)$（C代表网络层级），大大简化了计算节点间的通信交互过程，提升了计算效率。

在训练网络中最知名的在网计算技术是英伟达的可扩展分层聚合和归约协议（SHARP）[57]。Intel提出的switchML[58]系统在其Tofino专用芯片的可编程交换机上，实现了All-Reduce操作，充分利用了交换机的编程能力。

5 训练状态保存

随着参数规模达到千亿级，大模型训练时长会达到数十天，训练过程也可能因各种软硬件故障而中断。这就需要定期保存模型训练的中间状态，包括GPU内存中的模型参数和优化器状态。发生故障时，将最近的检查点载入到GPU内存中，可以实现快速的故障恢复，系统此时只会丢失很短时间内的计算结果。然而，检查点操作过程中序列化、压缩、文件IO的低效所引起的检查点停滞问题，也会阻塞训练任务，浪费GPU计算资源。因此，我们需要对检查点操作进行优化。主要优化方法如下：

1） 异步处理

GPU与CPU处理进行异步设计时，GPU主要完成前向传播、后向传播，CPU完成参数更新和检查点。微软Fiddle团队在CheckFreq[59]中使用动态建模分析，将CPU处理的检查点快照和持久化进行后台异步处理。DeepFreeze[60]设计了VELOC框架用于实现序列化和压缩异步。Gemini[61]给出的交

错流水也是异步的思想。

2） 轻量级、细粒度任务调度

将接口访问任务拆解为轻量级、细粒度子任务可以实现局部并行。例如：DeepFreeze[60]通过建立有向无环图实现分片和序列任务的重新组织调度，进而可以实现分层并行；Gemini[61]使用交错流水的方式进行接口访问任务调度。文献[62]建立分层模型，并使用模拟的方法改进接口传输调度算法。

3） 检查点计划及存储策略

关于检查点的生成频度，Mimose[63]等研究出一种GPU内存在线估算器，可以预测给定激活张量的内存使用率输入，并生成一个检查点计划，有效避免GPU内存溢出问题。

模型训练状态的存储策略也会影响检查点效率。在分布式训练中，可通过副本布局策略化来提升检查点的保存和读取效率。Gemini[61]采用多副本的方式，在本地和远程机器的CPU内存中维护检查点，并通过环状拓扑算法提高本地读取副本的恢复时间。

6 总结与展望

本文按照大模型训练的一般流程，回顾和总结了大模型训练主要阶段的相关技术背景及要点。随着大模型参数规模的不断增大、多模态数据处理类型的扩展，大模型训练的各个阶段都存在较大的优化空间。为进一步提升训练效率，我们认为后续还需重点展开以下几个方面的研究：

1）以数据为中心。数据的质量和数量对大模型的训练结果非常重要，这已经成为学术界和产业界的共识。很多研究人员开始转向以数据为中心的研究，其主要目的是设法提升数据质量，增加数据数量，而不是过多地考虑模型结构。这种转变在大模型领域尤其明显。

2）数据加载智能化和异构加速。根据模型需求动态调整数据加载策略，并结合事务感知或应用感知的缓存预取策略，有助于加快数据加载过程。此外，对于图像、视频、音频等非文本数据，如何结合专用集成电路（ASIC）或现场可编程门阵列（FPGA）等异构加速技术来有效提升数据预处理的效率，也是后续重要的研究方向。

3）网络通信领域定制。针对大模型训练场景特征，融合CXL等低延迟总线技术的发展，网络通信还需在新型网络拓扑、流量工程优化、互联总线协议领域定制等方面进行针对性优化，以更好地适配大模型训练网络的特点。

4）训练并行及自动化。大模型的算法结构和规模正在快速迭代，如何充分利用有限的计算存储网络资源，通过多维度细粒度的并行拆分策略和卸载技术，实现高效的训练并行，是一个需要持续研究的主题。未来，在用户给定模型和机器资源后，能够有效组合多种并行技术，自动帮助用户制定最优的并行策略，是分布式训练并行的终极目标。

致谢

感谢中兴通讯股份有限公司熊先奎、张景涛、姚海东和朱炫鹏对本研究的帮助！

参考文献

[1] AWADA U, ZHANG J K, CHEN S, et al. Machine learning driven latency optimization for Internet of things applications in edge computing [J]. ZTE Communications, 2022, 21(2): 40–52. DOI: 10.12142/ZTECOM.202302007

[2] CAI W B, YANG S L, SUN G, et al. Adaptive load balancing for parameter servers in distributed machine learning over heterogeneous networks [J]. ZTE Communications, 2023, 21(1): 72–80. DOI: 10.12142/ZTECOM.202301009

[3] ZHAO Z P, ZHAO Y L, YAN B Y, et al. Auxiliary fault location on commercial equipment based on supervised machine learning [J]. ZTE Communications, 2022, 20(S1): 7–15. DOI: 10.12142/ZTECOM.2022S1002

[4] 韩炳涛, 刘涛, 唐波. 深度学习的10年回顾与展望 [J]. 中兴通讯技术, 2022, 28(6): 75–84. DOI: 10.12142/ZTETJ.202206013

[5] 张振国, 杨倩倩, 贺诗波. 基于深度学习的图像语义通信系统 [J]. 中兴通讯技术, 2023, 29(2): 54–61. DOI: 10.12142/ZTETJ.202302011

[6] 潘囿丞, 侯永帅, 杨卿, 等. 大规模语言模型的跨云联合训练关键技术 [J]. 中兴通讯技术, 2023, 29(4): 49–56. DOI: 10.12142/ZTETJ.202304010

[7] 曾炜, 苏腾, 王晖, 等. 鹏程·盘古：大规模自回归中文预训练语言模型及应用 [J]. 中兴通讯技术, 2022, 28(2): 33–43. DOI: 10.12142/ZTETJ.202202006

[8] 韩旭, 张正彦, 刘知远. 知识指导的预训练语言模型 [J]. 中兴通讯技术, 2022, 28(2): 10–15. DOI: 10.12142/ZTETJ.202202003

[9] ZHAO W Z, ZHOU K, LI J Y, et al. A survey of large language models [EB/OL]. (2023–03–31) [2024–02–25]. https://arxiv. org/abs/2303.18223

[10] HERNANDEZ D, BROWN T, CONERLY T, et al. Scaling laws and interpretability of learning from repeated data [EB/OL]. (2022–05–21)[2024–02–25]. https://arxiv.org/abs/2205.10487

[11] LEE K, IPPOLITO D, NYSTROM A, et al. Deduplicating training data makes language models better [EB/OL]. [2024–02–25]. http://arxiv.org/abs/2107.06499

[12] PENEDO G, MALARTIC Q, HESSLOW D, et al. The refinedweb dataset for falcon llm: outperforming curated corpora with web data, and web data only [EB/OL]. (2023–07–01) [2024–02–25]. https://arxiv.org/abs/2306.01116

[13] LONGPRE S, YAUNEY G, REIF E, et al. A pretrainer's guide to training data: measuring the effects of data age, domain coverage, quality, &toxicity [EB/OL]. (2023–05–22) [2024–02–25]. https://arxiv.org/abs/2305.13169

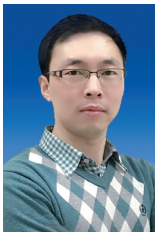
[14] GAN R, WU Z, SUN R, et al. Ziya2: data–centric learning is all LLMs need [EB/OL]. (2023–05–22) [2024–02–25]. https://arxiv.org/abs/2311.03301

[15] CARLINI N, TRAMER F, WALLACE E, et al. Extracting training

data from large language models [EB/OL]. (2020-12-14)[2024-02-25]. https://arxiv.org/abs/2012.07805
[16] JANG J, YOON D, YANG S, et al. Knowledge unlearning for mitigating privacy risks in language models [EB/OL]. (2022-10-04)[2024-02-25]. https://arxiv.org/abs/2210.01504
[17] CHEN W J, HE S B, XU Y W, et al. iCache: an importance-sampling-informed cache for accelerating I/O-bound DNN model training [C]// /Proceedings of 2023 IEEE International Symposium on High-Performance Computer Architecture (HPCA). IEEE, 2023. DOI: 10.1109/HPCA56546.2023.10070964
[18] CHILIMBI T, SUZUE Y, APACIBLE J, et al. Project Adam: building an efficient and scalable deep learning training system [C]//Proceedings of the 11th USENIX conference on Operating Systems Design and Implementation. ACM, 2014: 571-582. DOI: 10.5555/2685048.2685094
[19] HASHEMI S H, JYOTHI S A, CAMPBELL R H. icTac: accelerating distributed deep learning with communication scheduling [EB/OL]. (2018-05-08) [2024-02-25]. https://arxiv. org/abs/1803.03288
[20] NVIDIA. 2020. Fast AI data preprocessing with NVIDIA DALI [EB/OL]. [2024-02-25]. https://devblogs. nvidia. com/fast-ai-data-preprocessing-with-nvidia-dali/
[21] CHEN T Q, XU B, ZHANG C Y, et al. Training deep nets with sublinear memory cost [EB/OL]. [2024-02-25]. http://arxiv. org/abs/1604.06174
[22] MOHAN J, PHANISHAYEE A, RANIWALA A, et al. Analyzing and mitigating data stalls in DNN training [J]. Proceedings of the VLDB endowment, 2021, 14(5): 771-784. DOI: 10.14778/3446095.3446100
[23] KAPLAN J, MCCANDLISH S, HENIGHAN T, et al. Scaling laws for neural language models [EB/OL]. [2024-02-25]. http://arxiv. org/abs/2001.08361
[24] HOFFMANN J, BORGEAUD S, MENSCH A, at al. Training compute-optimal large language models [EB/OL]. (2022-05-29) [2024-02-25]. https://arxiv.org/abs/2203.15556
[25] XU Q, LI C, GONG C, et al. An efficient 2D method for training super-large deep learning models [EB/OL]. (2021-04-12)[2024-02-25]. https://arxiv.org/abs/2104.05343
[26] ZHANG B, TITOV I, SENNRICH R. Improving deep transformer with depth-scaled initialization and merged attention [C]// Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP). Association for Computational Linguistics. DOI: 10.18653/v1/d19-1083
[27] HUANG X S, PÉREZ F, BA J, et al. Improving transformer optimization through better initialization [C]//Proceedings of the 37th International Conference on Machine Learning. ACM, 2020: 4475-4483. DOI: 10.5555/3524938.3525354
[28] BA J L, KIROS J R, HINTON G E. Layer normalization [EB/OL]. (2016-07-21)[2024-02-25]. https://arxiv.org/abs/1607.06450
[29] XIONG R, YANG Y, HE D, et al. On layer normalization in the transformer architecture [EB/OL]. (2020-02-12) [2024-02-25]. https://arxiv.org/abs/2002.04745
[30] ZHANG B, SENNRICH R. Root mean square layernormalization [EB/OL]. (2019-10-16) [2024-02-25]. https://arxiv. org/abs/1910.07467
[31] HENDRYCKS D, GIMPEL K. Gaussian error linear units (GELUs) [EB/OL]. [2024-02-25]. http://arxiv.org/abs/1606.08415
[32] SHAZEER N. GLU variants improve transformer [EB/OL]. (2020-02-12)[2024-02-25]. http://arxiv.org/abs/2002.05202
[33] DAUPHIN Y N, FAN A, AULI M, et al. Language modeling with gated convolutional networks [EB/OL]. (2016-12-23)[2024-02-25]. https://arxiv.org/abs/1612.08083
[34] NENKOVA A, PASSONNEAU R, MCKEOWN K. The pyramid method: incorporating human content selection variation in summarization evaluation [J]. ACM transactions on speech and language processing, 4(2): 4-es. DOI: 10.1145/1233912.1233913
[35] ZAN D G, CHEN B, ZHANG F J, et al. Large language models meet NL2Code: a survey [EB/OL].[2024-02-25]. http://arxiv.org/abs/2212.09420
[36] YU J F, WANG X Z, TU S Q, et al. KoLA: carefully benchmarking world knowledge of large language models [EB/OL]. [2024-02-25]. http://arxiv.org/abs/2306.09296
[37] DHOLE K D, GANGAL V, GEHRMANN S, et al. NL-augmenter: a framework for task-sensitive natural language augmentation [EB/OL]. (2021-12-06) [2024-02-25]. http://arxiv. org/abs/2112.02721
[38] BORDIA S, BOWMAN S R. Identifying and reducing gender bias in word-level language models [EB/OL]. (2019-04-05) [2024-02-25]. http://arxiv.org/abs/1904.03035
[39] LEES A, TRAN V Q, TAY Y, et al. A new generation of perspective API: efficient multilingual character-level transformers [C]//Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining. ACM, 2022: 3197 - 3207. DOI: 10.1145/3534678.3539147
[40] HASHIMOTO T, ZHANG H, LIANG P. Unifying human and statistical evaluation for natural language generation [C]// Proceedings of the 2019 Conference of the North. Association for Computational Linguistics, 2019. DOI: 10.18653/v1/n19-1169
[41] SHOEYBI M, PATWARY M, PURI R, et al. Megatron-LM: training multi-billion parameter language models using GPU model parallelism [EB/OL]. (2019-09-17) [2024-02-25]. http://arxiv.org/abs/1909.08053
[42] LI S G, LIU H X, BIAN Z D. Colossal-AI: a unified deep learning system for large-scale parallel training [EB/OL]. (2021-10-28) [2024-02-25]. https://arxiv.org/abs/2110.14883
[43] HUANG Y, CHENG Y, BAPNA A, et al. Gpipe: efficient training of giant neu-ral networks using pipeline parallelism [EB/OL]. (2019-09-17)[2024-02-25]. https://arxiv.org/abs/1811.06965
[44] HARLAP A, NARAYANAN D, PHANISHAYEE A, et al. Pipedream: fast and efficient pipeline parallel DNN training [EB/OL]. (2018-06-08) [2024-02-25]. https://arxiv. org/abs/1811.06965
[45] NARAYANAN D, PHANISHAYEE A, SHI K, et al. Memory-efficient pipeline-parallel DNN training [EB/OL]. (2019-09-17) [2024-02-25]. https://arxiv.org/abs/2006.09503
[46] LI S G, XUE F Z, BARANWAL C, et al. Sequence parallelism: long sequence training from system perspective [EB/OL]. (2021-05-26)[2024-02-25]. http://arxiv.org/abs/2105.13120
[47] RASLEY J, RAJBHANDARI S, RUWASE O, et al. DeepSpeed: system optimizations enable training deep learning models with over 100 billion parameters [C]//Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining. ACM, 2020. DOI: 10.1145/3394486.3406703
[48] LEPIKHIN D, LEE H, XU Y Z, et al. GShard: scaling giant models with conditional computation and automatic sharding [EB/OL]. (2020-07-30)[2024-02-25]. http://arxiv.org/abs/2006.16668
[49] WILLIAM F, ZOPH B, SHAZEER M. Switch transformers: scaling to trillion parameter models with simple and efficient sparsity [EB/OL]. (2021-01-11) [2024-02-25]. https://arxiv. org/abs/2101.03961

[50] RAJBHANDARI S, LI C L, YAO Z W, et al. DeepSpeed-MoE: advancing mixture-of-experts inference and training to power next-generation AI scale [EB/OL]. [2024-02-25]. http://arxiv.org/abs/2201.05596
[51] HE J A, ZHAI J D, ANTUNES T, et al. FasterMoE: modeling and optimizing training of large-scale dynamic pre-trained models [C]//Proceedings of the 27th ACM SIGPLAN Symposium on Principles and Practice of Parallel Programming. ACM, 2022. DOI: 10.1145/3503221.3508418
[52] JIA Z H, ZAHARIA M, AIKEN A. Beyond data and model parallelism for deep neural networks [EB/OL]. [2024-02-25]. http://arxiv.org/abs/1807.05358
[53] ZHENG L M, LI Z H, ZHANG H, et al. Alpa: automating inter-and intra-operator parallelism for distributed deep learning [EB/OL]. (2022-01-28)[2024-02-25]. https://arxiv.org/abs/2201.12023
[54] 王卫斌, 周建锋, 黄兵. ODICT融合的网络2030 [J]. 中兴通讯技术, 2022, 28(1): 47-56. DOI: 10.12142/ZTETJ.202201011
[55] KIM J, DALLY W J, SCOTT S, et al. Technology-driven, highly-scalable dragonfly topology [EB/OL]. [2024-02-25]. https://pages.cs.wisc.edu/~markhill/restricted/isca08_dragonfly.pdf
[56] WANG W Y, GHOBADI M, SHAKERI K, et al. How to build low-cost networks for large language models (without sacrificing performance)? [EB/OL]. (2023-07-22)[2024-02-25]. https://doi.org/10.48550/arxiv.2307.12169
[57] GRAHAM R L, LEVI L, BURREDY D, et al. Scalable hierarchical aggregation and reduction protocol (SHARP)TM streaming-aggregation hardware design and evaluation[C]//International Conference on High Performance Computing. Springer, 2020: 41-59. DOI: 10.1007/978-3-030-50743-5_3
[58] SAPIO A, CANINI M, HO C Y, et al. Scaling distributed machine learning with In-network aggregation [EB/OL]. [2024-02-25]. http://arxiv.org/abs/1903.06701
[59] MOHAN J, PHANISHAYEE A, CHIDAMBARAM V. CheckFreq: fre-quent, fine-grained DNN checkpointing [EB/OL]. [2024-02-25]. https://www. microsoft. com/en-us/research/uploads/prod/2020/12/checkfreq-fast21.pdf
[60] NICOLAE B, LI J L, WOZNIAK J M, et al. DeepFreeze: towards scalable asynchronous checkpointing of deep learning models [C]//Proceedings of 20th IEEE/ACM International Symposium on Cluster, Cloud and Internet Computing (CCGRID). IEEE, 2020: 172-181. DOI: 10.1109/CCGrid49817.2020.00-76
[61] WANG Z, JIA Z, ZHENG S, et al. GEMINI: fast failure recovery in distributed training with In-memory checkpoints [C]// Proceedings of the 29th Symposium on Operating Systems Principles. ACM, 2023: 364-381. DOI: 10.1145/3600006.3613145
[62] MAURYA A, NICOLAE B, RAFIQUE M M, et al. Towards efficient I/O scheduling for collaborative multi-level checkpointing [C]//Proceedings of 29th International Symposium on Modeling, Analysis, and Simulation of Computer and Telecommunication Systems (MASCOTS). IEEE, 2021: 1-8. DOI: 10.1109/MASCOTS53633.2021.9614284
[63] LIAO J J, LI M Z, SUN Q X, et al. Mimose: an input-aware checkpointing planner for efficient training on GPU [EB/OL]. (2022-09-06)[2024-02-25]. https://arxiv.org/abs/2209.02478

作 者 简 介



田海东，中兴通讯股份有限公司先进计算存储架构师、项目经理；主要从事计算存储系统架构演进、近数据处理、存储体系端侧优化等方向的研究。

张明政，中兴通讯股份有限公司先进计算存储算法工程师；主要从事人工智能深度学习、感算融合等方向的算法研究。

常锐，中兴通讯股份有限公司先进计算存储架构师；主要从事大模型训推系统、数据流支撑平台、安全可信计算等方向的研究。

童贤慧，中兴通讯股份有限公司先进计算存储系统工程师；主要从事数据预处理、安全可信计算、对等系统等方向的研究。

通信网络与大模型的融合与协同

Integration and Collaboration of Communication Networks and Large Models

任天骐/REN Tianqi[1]，李荣鹏/LI Rongpeng[1]，
张宏纲/ZHANG Honggang[2]
（1. 浙江大学，中国 杭州 310007；
2. 之江实验室，中国 杭州 310012）
(1. Zhejiang University, Hangzhou 310007, China;
2. Zhijiang Lab, Hangzhou 310012, China)

DOI：10.12142/ZTETJ.202402005
网络出版地址：http://kns.cnki.net/kcms/detail/34.1228.tn.20240407.1930.004.html
网络出版日期：2024-04-10
收稿日期：2024-02-20

摘要：随着以大模型（LM）为代表的生成式人工智能（AI）的兴起，将LM应用于通信网络的研究引起了学术界和工业界的广泛关注。回顾了目前LM的主流神经网络架构及其能力涌现机理，然后从AI与通信的双向协同、网络大模型部署两方面，深入探讨了通信网络LM研究的主要进展。还分析了网络LM NetGPT将要面临的挑战以及未来的发展方向。考虑到基于AI/机器学习（ML）的通信模型相较于传统模型获得的出色性能，认为将通信网络与LM进行融合并使二者协同工作，能进一步地提升系统的性能。要实现通信网络与LM的融合与协同，本质上是要构建好网络LM，云边协同就提供了一种很好的网络LM部署方案。

关键词：LM；生成式AI；网络智能；NetGPT；模型协同

Abstract: Along with the springing up of generative artificial intelligence (AI), notably epitomized by large models (LM), the incorporation of these LMs within communication networks has attracted extensive attention in both academia and industry. An overview of the dominant deep neural network (DNN) architecture of LMs and its emerging capabilities is introduced. The significant advancements achieved by applying LMs for communication networks from two aspects are discussed, namely, the mutual collaboration between AI and communications, and the deployment of network generative pre-trained transformer (NetGPT). Additionally, the imminent challenges and further work are also discussed. Considering the outstanding performance of AI/machine learning (ML)-based communication models compared to traditional models, it is believed that integrating communication networks with large models and enabling them to work together can further enhance system performance. To realize the integration and collaboration of communication networks and large models, it is essentially necessary to build NetGPT properly. Edge-cloud collaboration provides a good deployment solution for NetGPT.

Keywords: LM; generative AI; network intelligence; NetGPT; model collaboration

引用格式：任天骐, 李荣鹏, 张宏纲. 通信网络与大模型的融合与协同 [J]. 中兴通讯技术, 2024, 30(2): 29-36. DOI: 10.12142/ZTETJ.202402005
Citation: REN T Q, LI R P, ZHANG H G. Integration and collaboration of communication networks and large models [J]. ZTE technology journal, 2024, 30(2): 29-36. DOI: 10.12142/ZTETJ.202402005

随着移动通信网络复杂度的显著增加和通信业务生态的多样化，通信网络面临着越来越多复杂场景的挑战。因此，通信网络既要满足高速、高质量的通信需求，又要向用户提供颇具差异性的业务体验，还要考虑稳定性和安全性，这对通信网络的设计、运营和维护提出了更高的要求。在这样的背景下，人工智能（AI）技术的出现为解决这些问题带来了新希望。现代AI建立在机器学习的基础上，在众多机器学习模型中，Transformer架构[1]因其独特的自注意力机制而脱颖而出，它能够处理长距离依赖关系，这在自然语言处理（NLP）等领域尤为重要。

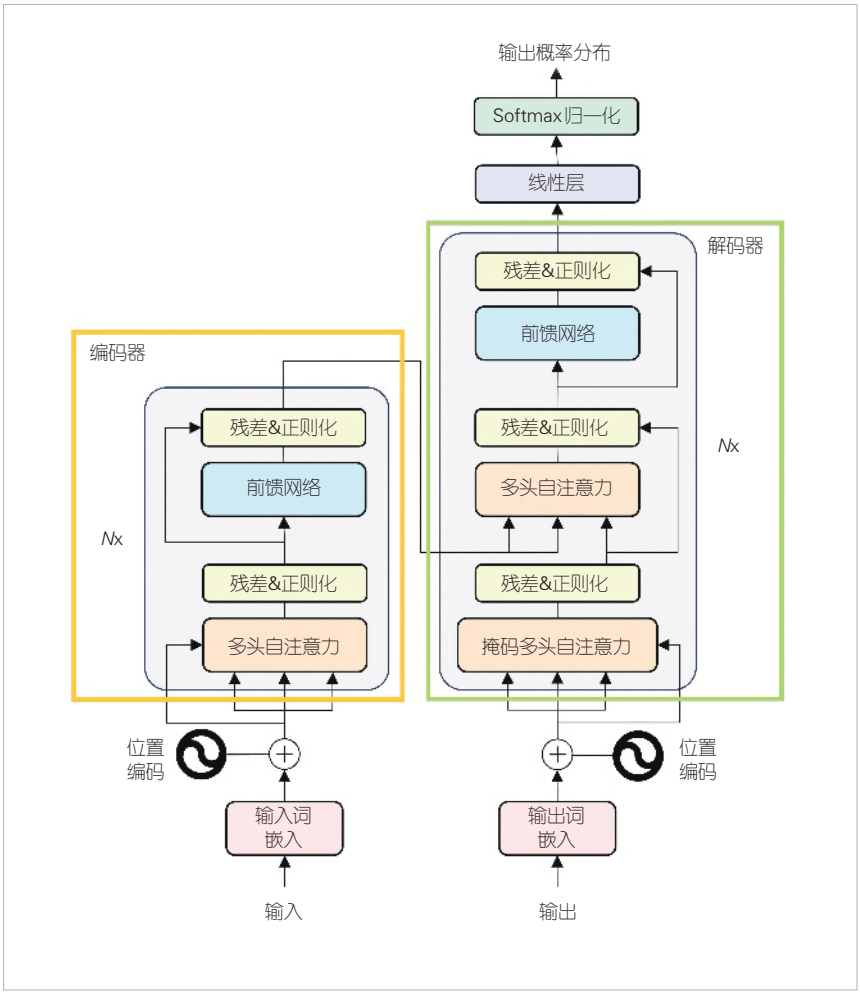
Transformer模型为大型预训练模型的构建提供了基础架构。2018年，谷歌提出的BERT[2]是基于Transformer架构的第一个突破，其革新的双向训练策略极大地提升了模型对文本深层次语义理解能力，在多项语言任务中取得优秀的性能。紧随其后的GPT-2[3]采用了Transformer的解码器结构，通过学习大量的文本数据，能够生成多样且逻辑合理的文本。随着计算资源的提升和优化，以及大模型能力标度率和具体涌现能力的发现，模型的规模正在迅速扩张[4-6]。从BERT模型1.1亿个参数到GPT-3[7]和GPT-4[8]的数百亿乃至

基金项目：国家自然科学基金项目（62071425）；浙江省“领雁”计划项目（2022C01093）；浙江省杰出青年基金项目（LR23F010005）

数万亿个参数。此外，通过预训练和微调，大模型中还融合多模态技术[9-10]，使得大模型在自然语言处理、计算机视觉[11-12]、自动驾驶[13-14]等多个领域展现出强大的潜力。

同时，数据、算力与模型构成了实现AI的三大基石，而6G成为“通、感、算、智、存”集于一体的超级基础设施平台，为融合AI提供了充足的条件，因此基于内生智能的新型网络架构应运而生。内生智能网络不仅要引入AI来构建网络，还需要充分利用网络节点的通信、计算和感知能力，并将通过分布式学习、群智式协同以及云边端一体化算法部署，原生支持各类AI应用，为各行业用户提供实时AI服务和实时计算类新业务[15-16]。

本文中，我们将首先探讨大语言模型（LLM）的基础原理，包括Transformer结构、标度率和涌现能力，以及LLM的预训练与微调过程。进一步地，我们将分析AI，特别是LLM在通信网络中的应用及其带来的双向增益。同时，也将探讨大模型发展面临的问题与挑战，以及如何更好地利用AI技术来优化并实现通信网络的转型。



▲图1 Transformer模型架构[1]

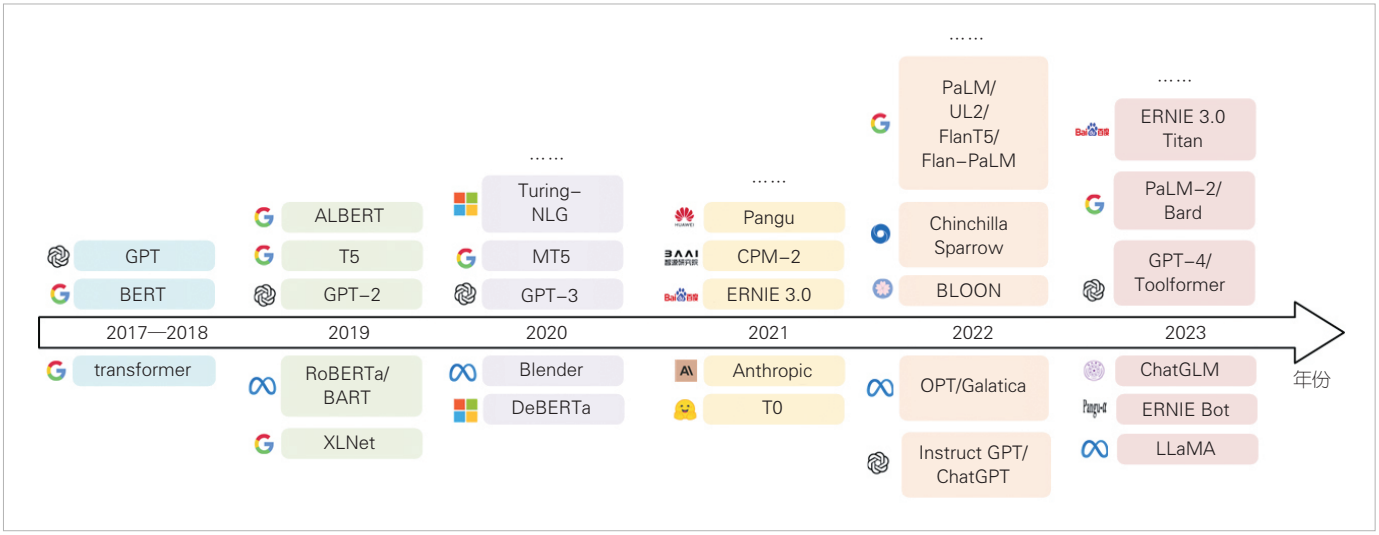
1 大模型的理论与技术

1.1 大模型架构

现有LLM的进步主要得益于Transformer的发展[1]。Transformer模型完全摒弃了传统语言模型广泛使用的循环神经网络（RNN）和长短期记忆网络（LSTM）模型，全面采用自注意力机制来处理序列。

如图1所示，Transformer模型包含编码器和解码器，两者均由N个（原文中N=6）相同的层堆叠而成。编码器负责理解输入文本并构造语义表示，而解码器则使用编码器的输出来生成目标序列。编码器中的每个层由多头自注意力层和全连接前馈网（FFN）两个子层构成，而解码器相比编码器多出一个掩码多头自注意力层。注意力机制的引入使得Transformer模型在处理序列的每个元素时，能够考虑到整个序列的上下文信息，从而在NLP任务中表现出并行化训练和性能优异的特点[18]。例如，Transformers架构通过自注意力机制解决了长距离依赖问题，使模型能够直接关注到序列中任意两个位置之间的关系。同时，Transformer架构允许比RNN更多的并行化，这使得图形处理器（GPU）上的大量数据上有效地预训练非常大的语言模型成为可能。

Transformer模型的提出极大地推动了LLM的发展。LLM的发展历程如图2所示。基于Transformer架构，LLM演化为3种主要架构：仅编码器（encoder-only）、仅解码器（decoder-only）、编码器-解码器（encoder-decoder）。目前，最主流的是仅解码器架构，代表性的LLM有GPT系列[3, 7-8]、LLaMA[18-19]、PaLM[20]等。仅编码器架构模型的代表是BERT系列，包括BERT[2]、RoBERTa[21]和ALBERT[22]等；编码器-解码器架构的代表模型有谷歌的T5模型[23]、Meta AI的BART模型[24]和华为的Pangu大模型等。3种架构各有优劣：仅解码器架构更多关注于从已有的信息扩展出新的内容，适合文本生成和扩展类型的任务，但需要大量的训练数据来提高生成文本的质量和多样性；仅编码器架构能更好地理解输入文本的语义和上下文信息，适合理解和分析类型的任务，缺点是无法直接生成文本输出；编码器-解码器架构能更好



▲图2 大型语言模型发展时间线

地处理输入序列和输出序列之间的关系，适合需要理解输入内容并生成相关响应的任务，如机器翻译、生成式问答等，但模型复杂度较高，训练时间和计算资源消耗较大。

1.2 标度率和涌现能力

大模型的标度率是OpenAI在2020年提出的概念[5]，是AI模型训练过程中的一个重要的经验性发现。在传统的小模型中，其性能往往会随着训练次数的增加而趋于稳定，甚至出现过拟合而导致性能下降。大模型的标度率则揭示了一个不同的现象：随着模型规模、数据集大小以及训练计算量的扩增，模型性能够获得持续提升。具体而言，当不受其他两个因素制约时，模型性能与每个单独的因素呈幂律关系。进一步的研究揭示[6]，当前的LLM实际上训练不足，而为了实现最佳性能，模型规模和训练数据集大小应以大致相同的速度扩增。此外，除了数据集大小，数据质量也被认为是影响模型性能的关键因素。

标度率提出后，可以预见：随着模型参数量的增加，模型在大部分任务中表现出的性能较为稳定。而随着模型规模的持续扩大，研究者发现[5]，对于特定的任务和模型来说，在模型规模小于某个阈值之前，模型基本不具备任务解决能力；但当模型规模大到一定程度时，模型性能显著提高。这被称为大模型的涌现能力。

1.3 大语言模型的预训练、微调与对齐

在大语言模型的预训练阶段，自监督学习发挥着核心的作用。该方法使模型能够在无需人工标注的数据集上学习并理解语言的丰富特征。自监督学习通过构建任务，如掩码语言模型（MLM）或自回归预测，使模型能够从大规模未标注文本中抽取和学习复杂的语言结构和语义信息。这种自监督机制的广泛应用源于其赋予模型从大量的文本数据中学习到通用语言表示的能力，这为模型后续进行特定任务的微调奠定了坚实基础。

预训练完成后，LLM可以获得处理各种任务的通用能力。为了将LLM适配到特定领域的任务，需要对LLM进行微调。LLM的微调，通常采用监督学习的技术路线。由于使用的训练数据通常包含标签或特定任务的指导信息，监督学习能使已经预训练过的模型针对具体的应用进行优化，提高了特定任务上的表现。近期，指令微调作为一种先进的微调策略，允许模型通过理解并执行明确的任务指令来调整其行为，进一步增强了模型对不同任务的适应能力和灵活性。

预训练和微调的策略反映了一种互补性：前者通过自监督学习为模型提供广泛的语言理解能力，而后者则确保模型在前者的基础上针对特定任务实现优化。这种互补性策略极大地提升了模型在多种自然语言处理任务中的泛化能力。

LLM预训练使用了语言建模的目标，但却没有考虑到人类的价值观或偏好，可能产生有害的、误导性的或有偏见的表达，因此需要一些对齐技术来使LLM的行为符合人类期望。为此，InstructGPT[25]利用基于人类反馈的强化学习（RLHF）技术[26]，通过学习奖励模型使LLM适配人类反馈，并将人类纳入训练的循环中来得到对齐良好的LLM[27]。

2 通信网络大模型的研究与发展

6G对网络架构提出了“万物智联，数字孪生”的总体愿景，强调智慧内生是6G网络应当具备的一大特征[15]。这

一特征意味着6G网络将内嵌AI能力，实现架构级智慧内生。6G网络对内能够利用智能来优化网络性能，增强用户体验，自动化网络运营，即使用AI来构建网络；对外能够抽取和封装网络智能，为各行各业用户提供网络和AI结合的通信和计算服务，即网络赋能AI[16]。AI构建网络和网络赋能AI两个概念共同构成了通信网络与大模型融合协同的框架。

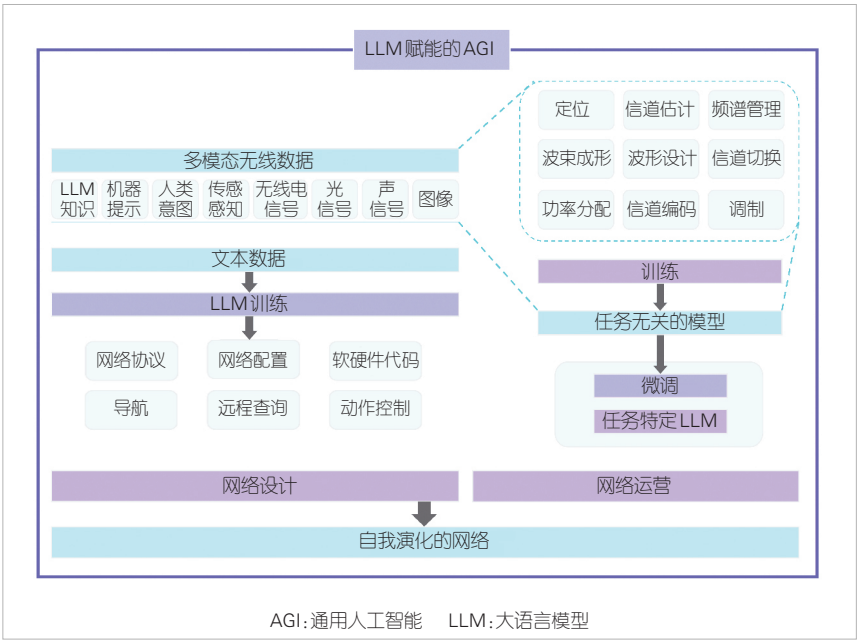
2.1 AI与通信网络的双向协同：构建与赋能

目前，通信网络的AI应用主要涉及机器学习的各个领域，包括监督学习、非监督学习和强化学习等，而生成式AI与通信网络的深度融合还处于起步阶段。这些技术构成了通信网络中机器学习的基础，致力于学习网络参数以优化网络性能。

近年来，LLM作为生成式AI的典型代表，在通信网络中的作用开始受到业界关注。LLM通过在大规模语料库上进行预训练，进而在多种下游任务中微调，从而展现出电信语言的理解能力。L. BARIAH等[28]通过微调LLM来识别第三代合作伙伴项目（3GPP）技术文档中的规范类别，证实了LLM在电信领域的应用价值。此外，LLM还被用于辅助网络运营（NetOps）和增强网络管理，如LLM可以作为NetOps中的常识性知识和推理能力的良好工具[29]。尽管在直接操作网络拓扑方面，LLM依然存在可靠性、可解释性等问题，但S. K. MANI等[30]提出的新框架通过生成自定义代码来解决这些问题，推进了LLM的网络管理实践。

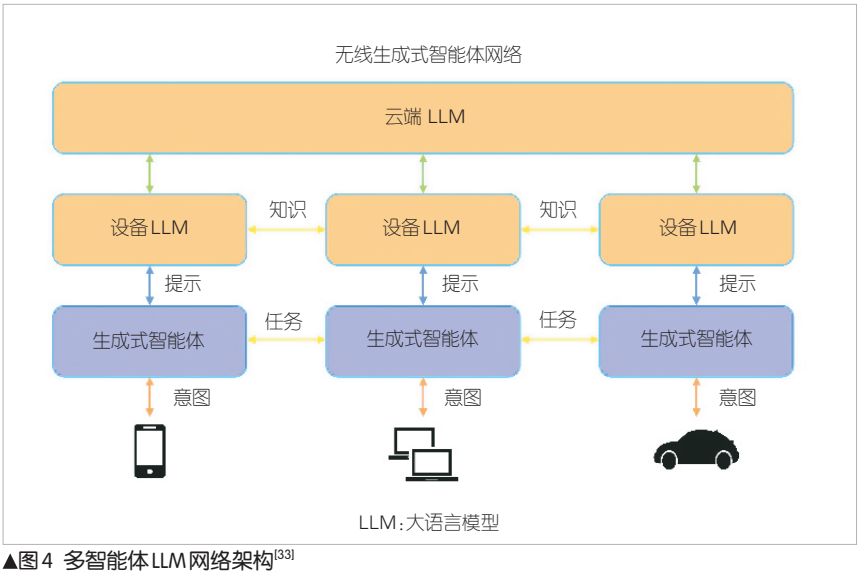
此外，大型生成式AI（GenAI）模型的发展为通信网络带来了新机遇。这些模型通过集成多模态数据，展示出在预训练基础模型、改善无线传感和传输方面的强大能力。L. BARIAH等[31]的研究深入探讨了GenAI模型与电信数据融合的策略，显示出大型GenAI模型在推动网络向自我演化方面的关键作用。图3中给出了通用人工智能（AGI）赋能无线网络的架构。总的来说，尽管LLM和GenAI模型在通信网络中的应用仍面临挑战，但它们在推动电信行业自动化和智能化发展方面的潜力是巨大的[32]。

在网络赋能AI方面，生成式AI正在推动无线设备实现集体智能，这对6G网络中的知识转移计算结构至关重要。该结构的目标在于利用云中的大型生成式AI模型，促进其向分布式集体智能过渡[31]。LLM巨大的计算和存储需求使其难以直接部署在边缘计算环境中。但通过在多个边缘设备上的部署，可以实现多智能体间的协同规划和任务决策。ZOU H.等[33]提出的多智能体LLM网络架构充分展示了这一点，如图4所示，其中无线生成式智能体不仅作为感知环境的传感器，也参与执行决策，这体现了生成式AI、边缘网络和多智能体系统之间创新性的协同效应。



AGI：通用人工智能　LLM：大语言模型

▲图3 AGI赋能的无线网络[31]



LLM：大语言模型

▲图4 多智能体LLM网络架构[33]

2.2 构建网络大模型的实践

在《网络大模型的十大问题》[34]中，网络大模型（NetGPT）被定义为无线网络中部署的大模型，其架构如图5所示。要实现通信网络与大模型的融合与协同，本质上是要构建好网络大模型。

在构建网络大模型的实践方面，WANG Y. C.[35]等调研了如何利用边缘云计算范式构建大规模GenAI系统。边缘云计算是指计算和存储资源靠近数据源或终端设备，将计算功能从传统的云数据中心推向网络边缘。边缘云计算利用了云服务器中强大的计算资源以及边缘服务器中高效的数据管理和通信。相比于云计算和多址边缘计算，边缘云计算在满足计算要求和低延迟要求上展现出优势，同时具有良好的可扩展性和数据安全性。

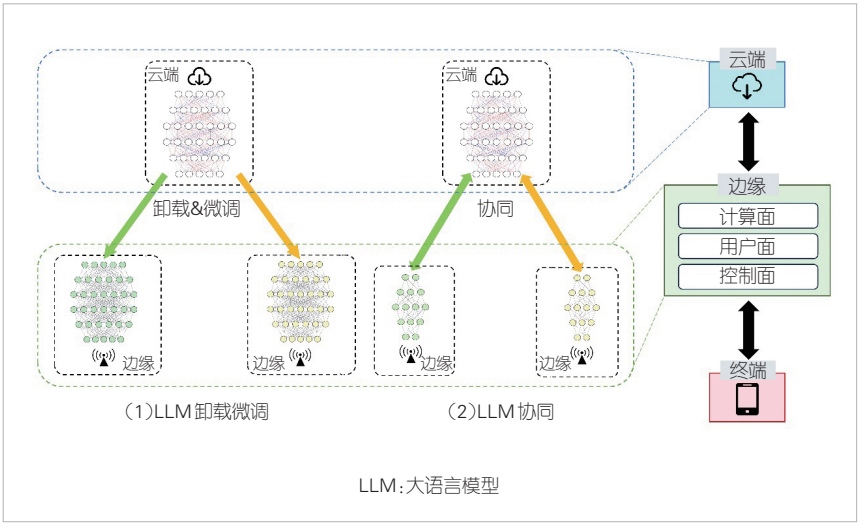
然而，将计算功能推向网络边缘意味着，在边缘端模型需要从云端进行计算卸载，且边缘和云端将缺乏一定的关联性。为了缓解这个问题，CHEN Y.等[36]提出了一种云边协同的部署方案，通过在云端与边缘端部署不同规模的模型协同作业来实现目的。在此架构中，边缘端部署的LLM是轻量级的，专门优化以适应边缘计算的资源限制，并能够利用位置相关信息增强个性化服务，以满足区域特定的需求。相对而言，云端的设备由于其更强大的计算能力和更大的存储空间，部署了完整版本的LLM，负责处理更复杂的全局任务。图6中给出了LLM卸载微调和LLM协同的两种部署方案。

在云边协同的架构中，边缘节点上的LLM负责收集并预处理来自本地区域的请求，包括将简单请求扩展为含有丰富区域特征的完整请求，并执行请求的去重整合。这些处理过的请求随后被发送到云端的LLM，后者利用其强大的计算能力生成高质量、个性化的回答。此过程不仅展现了通信网络在赋能AI方面的作用，还能通过边缘与云端LLM的高效协作，提升AI生成内容的质量和个性化程度；而AI对通信网络的增益则体现在通过边缘节点的LLM实现请求的有效预处理和减少冗余传输，这能够降低通信成本并优化网络时延。

在云端和边缘端部署LLM都需要基于预训练的LLM向通信任务进行迁移。CHEN Y.等[36]在工作中选择并部署了LLaMA-7B模型和GPT-2-base模型，并对部署的LLM进行微调来适应任务需求。在云端部署的LLaMA-7B模型，无法直接生成响应式文本，因此选择基于低秩适应（LoRA）的技术[37]，使用Standford Al - paca数据集[38]进行参数高效的微调。在边缘端部署的GPT-2-base模型，需要附加基于位置的信息来扩展提示，以实现个性化，因此选择self-instruct方法[39]，使用手动编写的位置相关提示与OpenAI的TextDavinci-003模型进行交互，来生成有效的文本样本作为“综合提示”。

云边协同部署网络大模型的工作流程主要集中于协调边缘和云端的一系列网络功能，并优化数据处理和分析的过程。当用户请求生成特定内容时，该架构通过先进的逻辑AI工作流来解析和编排服务，根据用户需求和网络状况的动态变化，选择是在边缘端进行初步处理还是在云端执行深

▲图5 网络大模型NetGPT的3层架构[34]



▲图6 网络大模型部署方案[36]

度处理。在服务部署阶段，逻辑AI工作流将根据服务质量需求映射到相应的物理资源。在融合通信和计算（C&C）资源管理层面，我们不仅需要考虑控制面的无缝连接和用户平面中的信息传输可靠性，还需要在计算平面中有效地协调异构计算资源。此外，该架构还引入新的协议栈以传输AI生成的消息，并实时更新和分发模型，同时考虑引入新的标识符来为实时AI工作流优先分配网络资源。

总体而言，网络大模型实现了C&C资源的深度融合，并通过个性化的云边大模型耦合更新机制促进了云边协同以提高服务质量。此外，通过在边缘处理私有数据，在云端处理大数据的分割机制，达成了计算效率和数据安全的最优平衡。

3 问题与挑战

目前，将AI算法融入通信系统已经有众多应用场景，如AI赋能物理层、AI赋能高层、AI赋能应用层等。此前，研究人员在这些应用场景做出了许多有益的尝试[40]，包括但不限于基于AI的信道估计及反馈[41–42]、基于AI的多输入多输出（MIMO）检测[43–44]、基于AI的资源和功率分配[45]和基于AI的传输层拥塞控制技术[46]等。这些研究都证明，与传统通信模型相比，基于AI/机器学习（ML）的通信模型可以获得更出色的性能。目前的研究大多采用传统的AI算法或神经网络结构，但根据标度率的发现以及大模型在众多领域展示出的卓越性能，我们有理由相信，将大模型应用于这些任务中，将会获得更大的增益。然而网络大模型领域的研究依然面临着一系列基础性问题的挑战。这些挑战主要涉及大模型本身的设计类问题和网络设计如何支撑大模型应用类问题[34]，主要如下：

1）模型协同：在不同规模和部署位置的模型之间实现有效的数据和参数协同是一个主要挑战。此外，不同任务类型对模型推理协同的需求也有所不同。针对跨域任务，L0全网通用大模型需要协同多个L1网络专业大模型进行处理，并提供通用知识；而针对单域任务，L1网络专业大模型需要和多个L2网络小模型进行协同处理，并提供专业知识。总的来说，实现网络内不同规模模型的协同进化，以及明确各自的职责，是解决这一挑战的关键策略。

2）网络架构设计：引入NetGPT优化网络服务需要考虑如何利用NetGPT的自然语言理解能力为应用程序生成专有的网络服务，并处理模型更新导致的计算负担。此外，考虑到当前网络的基于字符串的接口协议可能被基于模型间的协作接口取代，为了保证网络性能的实时性、稳定性和可靠性，需要把NetGPT深度融入6G网络架构，推动网元的智能化。

3）分布式学习与部署：在分布式网络中，考虑到节点计算资源和存储能力的差异，模型需要分布式拆分和自适应调整。在模型学习算法层面，现有的模型并行和数据并行方式存在局限性，因此还需要我们深入探索分布式训练方法。此外，分布式节点间的通信瓶颈是制约模型性能的关键因素，这就需要从算法和网络设计两方面同时入手，进行模型压缩，如剪枝和量化等，在网络内设计高效的节点间通信机制。此外，数据隐私与数据异质性、以及如何降低通信开销，也是需要关注的问题。

4）全生命周期管控和编排：在生命周期管控方面，不仅要选择适当的拆分策略，还要设计高效的更新和维护策略以应对计算开销和时间成本的显著增加。同时，考虑到NetGPT的知识产权保护，还需要建立平衡的协同管理机制。在编排方面，需要对计算任务和网络资源进行合理的识别、编排和反馈，以提高系统性能和资源利用率，实现面向动态需求的NetGPT闭环控制。

4 结束语

大模型作为当前最热门的研究热点，毫无疑问将成为AI与通信融合的关键组成部分，在提高网络中AI的通用性和多任务处理能力等方面发挥重要作用。本文中，我们首先从大模型的架构、标度率和涌现能力以及LLM的训练微调与对齐3个方面回顾了大模型的理论与技术，之后探讨了LLM和生成式AI在通信网络中的应用及其带来的双向增益。接下来，强调了AI与通信网络的双向协同，包括AI构建网络和网络赋能AI的概念，以及构建网络大模型NetGPT的实践。网络大模型作为一种内生智能的新型网络架构展现出巨大潜力，但要成功地部署网络大模型仍然存在一定的挑战。我们期待在该领域能有更多的前瞻性研究工作，为通信网络与大模型的融合与协同带来创新和突破。

致谢

感谢浙江大学陈宇轩和鲁芝琳在本文撰写过程中给予的帮助和支持。

参考文献

[1] VASWANI A, SHAZEER N, PARMAR N, et al. Attention is all you need [C]//Proceedings of the 31st International Conference on Neural Information Processing Systems. ACM, 2017: 6000–6010. DOI: 10.5555/3295222.3295349

[2] DEFLIN J, CHANG M W, LEE K, et al. Bert: pre-training of deep

bidirectional transformers for language understanding [EB/OL]. [2024−03−04]. https://aclanthology.org/N19−1423/
[3] RADFORD A, WU J, CHILD R, et al. Language models are unsupervised multitask learners [EB/OL]. [2024−03−04]. https://cdn. openai. com/better−language−models/language_models_are_unsupervised_multitask_learners.pdf
[4] KAPLAN J, McCANDLISH S, HENIGHAN. Scaling laws for neural language models [EB/OL]. (2020−01−23) [2024−03−04]. https://arxiv.org/abs/2001.08361
[5] WEI J, TAY Y, BOMMASANI R, et al. Emergent abilities of large language models [EB/OL]. (2022−01−15) [2024−03−04]. https://arxiv.org/abs/2206.07682
[6] HOFFMANN J, BORGEAUD S, MENSCH A, et al. Training compute−optimal large language models [EB/OL]. (2022−03−29) [2024−03−04]. https://arxiv.org/abs/2203.15556
[7] BROWN T B, MANN B, RYDER N, et al. Language models are few−shot learners [C]//Proceedings of the 34th International Conference on Neural Information Processing Systems. ACM, 2020: 1877−1901. DOI: 10.5555/3495724.3495883
[8] ACHIAM J, ADLER S, AGARWAL S, et al. GPT−4 technical report [EB/OL]. (2023−03−15) [2024−03−04]. https://arxiv. org/abs/2303.08774
[9] DONG R, HAN C, PENG Y, et al. DreamLLM: synergistic multimodal comprehension and creation [EB/OL]. (2023−09−20) [2024−03−04]. https://arxiv.org/abs/2309.11499
[10] LIN Z Q, YU S, KUANG Z Y, et al. Multimodality helps unimodality: cross−modal few−shot learning with multimodal models [C]//Proceedings of IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). IEEE, 2023: 19325−19337. DOI: 10.1109/cvpr52729.2023.01852
[11] LI J, LI D, XIONG C, et al. Blip: bootstrapping language−image pre−training for unified vision−language understanding and generation [C]//Proceedings of the 39th International Conference on Machine Learning. JMLR, 2022: 12888−12900. DOI: 10.48550/arXiv.2201.12086
[12] LI J, LI D, SAVARESE S, et al. Blip−2: bootstrapping language−image pre−training with frozen image encoders and large language models [C]//Proceedings of the 40th International Conference on Machine Learning. JMLR, 2023: 19730−19742. DOI: 10.5555/3618408.3619222
[13] WANG H W, XIE J, HU C Y, et al. Drivemlm: aligning multi−modal large language models with behavioral planning states for autonomous driving [EB/OL]. (2023−12−14)[2024−03−04]. https://arxiv.org/abs/2312.09245
[14] CUI C, MA Y, CAO X, et al. A survey on multimodal large language models for autonomous driving [C]//Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV). IEEE, 2024: 958−979. DOI: 10.48550/arXiv.2311.12320
[15] IMT−2030(6G)推进组. 6G 总体愿景和潜在关键技术 [EB/OL]. (2022−02−18)[2024−03−02]. https://www.eet−china.com/news/202106090412.html
[16] IMT−2030(6G)推进组. 6G总体网络架构愿景和关键技术展望 [EB/OL]. (2021−09−16) [2024−03−02]. https://cloud. tencent. com/developer/news/857663
[17] ALMMAR J. The illustrated transformer [EB/OL]. (2018−06−27) [2024−03−04]. https://jalammar.github.io/illustrated−transformer/
[18] TOUVRON H, LAVRIL T, IZACARD G, et al. Llama: open and efficient foundation language models [EB/OL]. (2023−02−27) [2024−03−04]. https://arxiv.org/abs/2302.13971
[19] TOUVRON H, MARTIN L, STONE K, et al. Llama 2: open foundation and fine−tuned chat models [EB/OL]. (2023−06−18) [2024−03−04]. https://arxiv.org/abs/2307.09288
[20] CHOWDHERY A, NARANG S, DEVLIN J, et al. Palm: Scaling language modeling with pathways [J]. Journal of machine learning research, 2023, 24(240): 1−113
[21] LIU Y, OTT M, GOYAL N, et al. Roberta: a robustly optimized bert pretraining approach [EB/OL]. (2019−7−26) [2024−03−02]. https://arXiv preprint arXiv:1907.11692, 2019
[22] LAN Z, CHEN M, GOODMAN S, et al. Albert: a lite bert for self−supervised learning of language representations [EB/OL]. (2020−02−09) [2024−03−02]. https://arXiv preprint arXiv: 1909.11942, 2019
[23] RAFFEL C, SHAZEER N, ROBERTS A, et al. Exploring the limits of transfer learning with a unified text−to−text transformer [J]. The journal of machine learning research, 2020, 21(140): 1−67. DOI: 10.48550/arXiv.1910.10683
[24] LEWIS M, LIU Y, GOYAL N, et al. Bart: denoising sequence−to−sequence pre−training for natural language generation, translation, and comprehension [EB/OL]. (2019−10−29) [2024−03−02]. https://arXiv preprint arXiv:1910.13461, 2019.
[25] OUYANG L, WU J, JIANG X, et al. Training language models to follow instructions with human feedback [C]//Advances in neural information processing systems 35 (NeurIPS 2022). Curran Associates, 2022: 27730−27744. DOI: 10.48550/arXiv.2203.02155
[26] CHRISTIANO P F, LEIKE J, BROWN T B, et al. Deep reinforcement learning from human preferences [C]// Proceedings of the 31st International Conference on Neural Information Processing Systems. ACM, 2017: 4302 – 4310. DOI: 10.5555/3294996.3295184
[27] WANG Y, ZHONG W, LI L, et al. Aligning large language models with human: a survey [EB/OL]. (2023−07−24) [2024−03−04]. https://arxiv.org/abs/2307.12966
[28] BARIAH L, ZOU H, ZHAO Q, et al. Understanding telecom language through large language models [EB/OL]. [2024−03−04]. https://arxiv.org/pdf/2306.07933v1.pdf
[29] MIAO Y K, BAI Y, CHEN L, et al. An empirical study of NetOps capability of pre−trained large language models [EB/OL]. (2023−09−11)[2024−03−05]. https://arxiv.org/abs/2309.05557
[30] MANI S K, ZHOU Y J, HSIEH K, et al. Enhancing network management using code generated by large language models [C]//Proceedings of the 22nd ACM Workshop on Hot Topics in Networks. ACM, 2023: 196−204. DOI: 10.1145/3626111.3628183
[31] BARIAH L, ZHAO Q Y, ZOU H, et al. Large generative AI models for telecom: the next big thing? [J]. IEEE communications magazine, 2024: 1−7. DOI: 10.1109/mcom.001.2300364
[32] MAATOUK A, PIOVESAN N, AYED F, et al. Large language models for telecom: Forthcoming impact on the industry [EB/OL]. (2023−08−11) [2024−03−05]. https://arxiv. org/abs/2308.06013
[33] ZOU H, ZHAO Q, BARIAH L, et al. Wireless multi−agent generative AI: from connected intelligence to collective intelligence [EB/OL]. (2023−07−06) [2024−03−05]. https://arxiv.org/abs/2307.02757
[34] TONG W, PENG C, YANG T, et al. Ten issues of NetGPT [EB/OL]. [2024−03−05]. https://arxiv.org/pdf/2311.13106.pdf
[35] WANG Y C, XUE J T, WEI C W, et al. An overview on generative AI at scale with edge−cloud computing [J]. IEEE open journal of the communications society, 2023, (4): 2952−2971. DOI: 10.1109/ojcoms.2023.3320646
[36] CHEN Y, LI R, ZHAO Z, et al. NetGPT: a native−AI network architecture beyond provisioning personalized generative services [EB/OL]. [2024−03−05]. https://ieeexplore. ieee. org/document/10466747
[37] SU J, LU Y, PAN S, et al. LoRA: low−rank adaptation of large language models [EB/OL]. (2021−04−20) [2024−03−02]. https://

arxiv. org/abs/2104.09864
[38] TAORI R, GULRAJANI I. Stanford alpaca: an instruction-following llama model [EB/OL]. [2024-03-02]. https://github.com/tatsu-lab/stanford alpaca
[39] WANG Y, KORDI Y. Self-instruct: aligning language model with self generated instructions [EB/OL]. (2022-12-20) [2024-03-02]. https://arxiv.org/abs/2212.10560
[40] SUN Y, PENG M, ZHOU Y, et al. Application of machine learning in wireless networks: key techniques and open issues [J]. IEEE communications surveys & tutorials, 2019, 21(4): 3072-3108. DOI: 10.1109/COMST.2019.2924243
[41] GAO J, ZHONG C, LI G Y, et al. Deep learning-based channel estimation for massive MIMO with hybrid transceivers [J]. IEEE transactions on wireless communications, 2021, 21(7): 5162-5174. DOI: 10.1109/TWC.2021.3137354
[42] MA X, GAO Z, GAO F, et al. Model-driven deep learning based channel estimation and feedback for millimeter-wave massive hybrid MIMO systems [J]. IEEE journal on selected areas in communications, 2021, 39(8): 2388-2406. DOI: 10.1109/JSAC.2021.3087269
[43] YUN S, MOON S, JEON Y S, et al. Intelligent MIMO detection with momentum-induced unfolded layers [J]. IEEE wireless communications letters, 2024, 13(3): 879-883. DOI: 10.1109/LWC.2023.3348933
[44] HE H, WEN C K, JIN S, et al. Model-driven deep learning for MIMO detection [J]. IEEE transactions on signal processing, 2020, 68: 1702-1715. DOI: 10.1109/TSP.2020.2976585
[45] KARAKS E K, GEMICI Ö F, HOKELEK İ, et al. Work-in-progress: AI based resource and power allocation for NOMA systems [C]//2023 IEEE International Black Sea Conference on Communications and Networking (BlackSeaCom). IEEE, 2023: 402-407. DOI: 10.1109/BlackSeaCom58138.2023.10299756
[46] PILLAI B, CHHABRA G. TCP-CNNLSTM: congestion control scheme for MANET using AI Technologies [C]//2023 Second International Conference on Augmented Intelligence and Sustainable Systems (ICAISS). IEEE, 2023: 63-69. DOI: 10.1109/ICAISS58487.2023.10250756

作 者 简 介

任天骐，浙江大学在读本科生；研究方向为大型语言模型在通信场景中的应用及语义通信。

李荣鹏，浙江大学信息与电子工程学院副教授、博士生导师；主要研究方向为智能通信网络、网络智能、网络切片等；曾入选首批博士后创新人才支持计划，获得浙江省杰出青年基金项目资助，并获吴文俊人工智能优秀青年奖、江苏省科学技术奖一等奖等。

张宏纲，浙江大学兼任教授、博士生导师；长期从事无线通信与网络、人工智能、认知通信、绿色通信、复杂网络等领域的研究；曾获2021年IEEE通信学会杰出论文奖、IEEE Internet of Things Journal（IoT-J）最佳论文奖等；发表论文265篇，拥有IEEE 802.15 UWB国际标准提案16项、国际专利3项。

基于存算一体集成芯片的大语言模型专用硬件架构

Large Language Model Specific Hardware Architecture Based on Integrated Compute-in-Memory Chips

何斯琪/HE Siqi，穆琛/MU Chen，陈迟晓/CHEN Chixiao
（复旦大学，中国 上海 200433）
(Fudan University, Shanghai 200433, China)

DOI：10.12142/ZTETJ.202402006
网络出版地址：http://kns.cnki.net/kcms/detail/34.1228.tn.20240407.1932.006.html
网络出版日期：2024-04-09
收稿日期：2024-02-25

摘要：目前以ChatGPT为代表的人工智能（AI）大模型在参数规模和系统算力需求上呈现指数级的增长趋势。深入研究了大型模型专用硬件架构，详细分析了大模型在部署过程中面临的带宽问题，以及这些问题对当前数据中心的重大影响。提出采用存算一体集成芯片架构的解决方案，旨在缓解数据传输压力，同时提高大模型推理的能量效率。此外，还深入研究了在存算一体架构下轻量化-存内压缩协同设计的可能性，以实现稀疏网络在存算一体硬件上的稠密映射，从而显著提高存储密度和计算能效。

关键词：大语言模型；存算一体；集成芯粒；存内压缩

Abstract: Artificial intelligent (AI) models represented by ChatGPT are showing an exponential growth trend in parameter size and system computing power requirements. The dedicated hardware architecture for large models is studied, and a detailed analysis of the bandwidth bottleneck issues faced by large models during deployment is provided, as well as the significant impact of this challenge on current data centers. To address this issue, a solution of using integrated compute-in-memory chiplets has been proposed, aiming to alleviate data transmission pressure and improve the energy efficiency of large-scale model inference. In addition, the possibility of lightweight in-memory compression collaborative design under the in-memory computing architecture is studied, in order to achieve dense mapping of sparse networks on the integrated in-memory computing architecture hardware, thereby significantly improving storage density and computational energy efficiency.

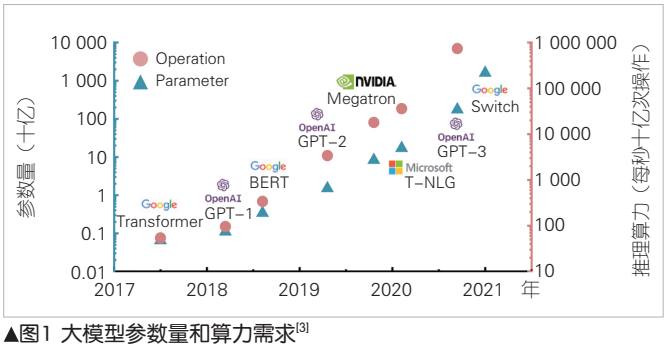
Keywords: large language model; compute-in-memory; chiplet; in-memory compression

引用格式：何斯琪, 穆琛, 陈迟晓. 基于存算一体集成芯片的大语言模型专用硬件架构 [J]. 中兴通讯技术, 2024, 30(2): 37-42. DOI: 10.12142/ZTETJ.202402006

Citation: HE S Q, MU C, CHEN C X. Large language model specific hardware architecture based on integrated compute-in-memory chips [J]. ZTE technology journal, 2024, 30(2): 37-42. DOI: 10.12142/ZTETJ.202402006

近年来，基于注意力机制的大语言模型（LLM）[1]取得了显著成功。与此同时，模型尺寸在迅速增长，如图1所示，每两年模型尺寸增长240倍，而相应的算力需求则增长近750倍。与此同时，硬件每两年3.1倍的发展速度[2]已逼近物理极限，进入了技术发展的瓶颈期。传统的超大规模和超大面积的单芯片系统级芯片（SoC）方案面临着利用率低、良率低、验证复杂度高、设计成本激增等一系列问题，同时集成电路制造已经达到了光刻掩膜版的最大面积上限。因此，大型模型的推理变得异常复杂且成本高昂，这成为当前研究和实际应用中需要解决的关键问题。

为了突破存储单元和计算单元之间的数据搬移的瓶颈，提高计算芯片能效，存算一体的专用芯片架构逐渐成为了神



▲图1 大模型参数量和算力需求[3]

基金项目：国家自然科学基金项目（62322404）；复旦大学-中兴通讯强计算架构研究联合实验室“存算一体架构研究项目”

经网络计算芯片研究和大模型实际部署的重要前进方向。通过电路与架构的协同创新，存算一体架构试图打破存储器和计算器之间的壁垒，实现数据搬移效率的提高或数据搬移次数的减少，从而提高芯片的计算能效。

然而，目前已有的神经网络计算芯片可扩展性欠佳，无法完全适应大模型的推理需求。在上述背景下，处理器领域的巨头已经将目光投向了集成芯粒（Chiplet）这一新兴技术。集成芯粒技术最早由加利福尼亚大学圣塔芭芭拉分校（UCSB）大学的谢源教授于2017年国际计算机辅助设计会议（ICCAD）上提出[4]。与单芯片SoC方案不同，集成芯粒方案先将多个小颗粒芯片独立设计并实现，然后通过先进封装技术重新组装，从而完成系统上的功能集成。美国Intel公司、AMD公司、英伟达公司的服务器/数据中心芯片都已开始广泛采用集成芯粒方案[5-7]。这些方案将高性能计算核心设计为模块化芯片，通过2.5D/3D封装技术、高速片间互联技术和有源基板技术将计算核心芯片模块集成。在不明显增加设计复杂度的前提下，保证芯片的良率，延续了后摩尔时代芯片算力提升。这一趋势为硬件设计提供了更为灵活和高效的解决方案，以适应不断增长的大型模型算力需求。

1 大模型对数据中心的挑战

1.1 集成芯片技术

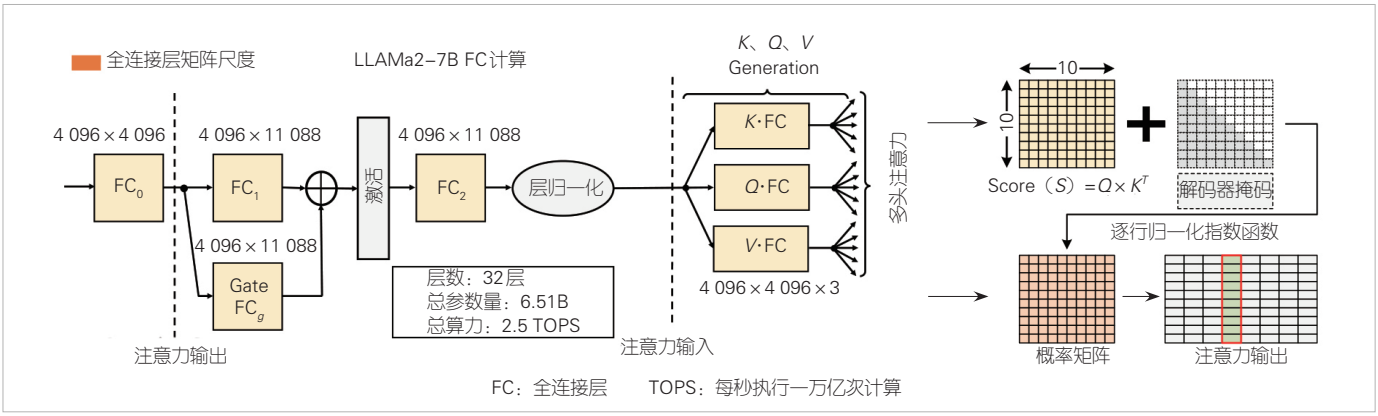
以ChatGPT为代表的人工智能（AI）大模型在参数规模和系统算力需求上呈现出指数级的增长趋势。当前，能够支持大型模型的数据中心和超级计算机普遍采用以xPU+主机内存缓冲器（HBM）集成芯片为核心的高性能处理器芯片系统。如图2所示，这些大算力芯片具备PFLOPS级算力和100 GB级存储性能，例如Nvidia H100图形处理器（GPU）拥有2 PFLOPS（每秒执行1 000万亿次浮点运算）的算力，AMD Instinct MI300拥有383 TFLOPS（每秒执行1万亿次浮点运算）的算力，华为昇腾910 B则具备256 TFLOPS算力等。传统的超大规模和超大面积的单芯片SoC方案已经面临着诸多问题，包括利用率低、良率低、验证复杂度高以及设计成本激增等。同时，集成电路制造已经达到了光刻掩膜版的最大面积上限，而30.48 cm（12英寸）晶圆的掩膜也在光刻机的要求下存在上限，最大芯片设计面积为858 mm^2。在这样的背景下，单芯片SoC的算力进一步扩充空间受到限制，潜在的良率问题和面积限制使得算力的提升变得更加困难。同时，自2023年起，美国进一步加强了针对中国芯片产业的出口限制，对总处理性能和算力密度超过超过规定的芯片实施了更加严格的管制。

▲图2 超算中心总算力和集成芯片数

为了缩小智能计算和处理器芯片技术上的差距，采用微纳架构工艺将多个芯片（粒）集成已经成为克服单芯片制造最大面积极限和芯片电路规模瓶颈的重要手段。不同于单芯片方案，集成芯片方案通过使用先进封装技术将多个小颗粒芯片组件组装在一起，实现了系统上的功能集成。这种方法将大型昂贵的SoC分解为体积更小、良率更高且更具成本效益的单芯片，同时也有助于缩短设计周期，降低成本。集成芯片技术已成为高性能处理器不可或缺的组成部分，而它正朝着3D多层堆叠、更多种类的芯片以及更大规模集成的方向发展。这一发展趋势的目标是进一步满足大型模型对硬件性能的不断增长的需求，适应日益增加的计算和处理任务。

1.2 大模型部署的带宽瓶颈

以图3中展示的拥有70亿参数的大型模型（LLAMa2-7B）为例，该大型模型的每一层多头注意力都包括多个连续前馈（FCL）计算。与此相关的单层参数量达到2.03亿，而32层的参数总量达到65亿，占用整体系数和计算的85%以上，远超过单一互补金属氧化物半导体（CMOS）芯片的片上存储空间。注意力模块的计算存储要求则相对较低，CPU/中等性能网络处理器（NPU）即可完成。在大型模型推理中，如要满足每秒1万个令牌的实时要求，即令牌速率为10 000个/秒，对GPU的带宽需求将达到64 TB/s，而当前的HBM3带宽仅为0.8 GB/s。因此，对于十亿级以上规模的大型模型网络应用场景，现有的GPU/TPU+DRAM分离计算架构难以满足不断增长的模型参数传输带宽需求。



▲图3 LLAMa-7B模型全连接层和注意力模块参数维度示意图

这种情况表明，随着大型模型的不断发展和应用场景的扩大，现有的硬件架构在满足大规模模型计算需求方面面临着巨大的挑战。具体而言，参数量巨大且算力要求高的大模型导致了计算和存储资源高需求的问题，而当前的GPU/TPU+DRAM结构的带宽限制使得数据传输方面的瓶颈日益显现。因此，未来的硬件设计和架构需不断创新，以适应快速增长的大型模型计算需求，提供更高效的数据传输和处理解决方案。

2 存算一体集成芯片的优势

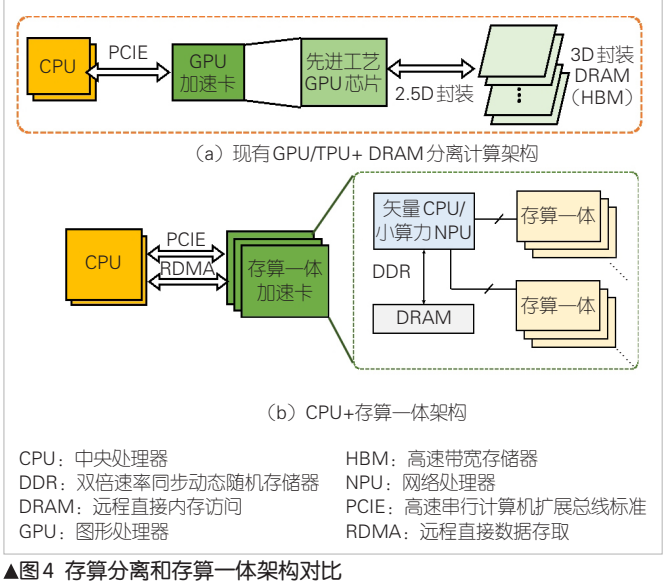
2.1 缓解带宽瓶颈

经典存算一体的设计基于交叉阵列。根据欧姆定律和基尔霍夫定律，输入特征用存储阵列的字线上的电压表示，输出特征会表示为位线上的电流大小，因此能够一次性完成矩阵乘加操作。同时，由于在计算过程中仅进行输入输出的搬运，权重系数一直固定在存储阵列中，所以能够显著减少数据搬移开销。我们发现，如果采用CPU+存算一体的组合的架构，相较于现有的GPU/TPU+DRAM分离计算架构（如图4所示），能够在相同的令牌速率和算力下，实现带宽的显著节约，达到xPU+HBM架构下1 000+倍的水平。举例来说，当采用和第1节相同的令牌速率（10 000个/s）时，存算一体架构仅需32～64 Gbit/s的带宽，就能节省超过1 000倍的带宽。

另一方面，当单颗芯粒的算力达10 TOPS，存储容量达到200 MB时，根据12/14 nm工艺估算，芯粒的计算电路面积约为8 mm^2，存储面积约为300 mm^2，此时实际的算力密度大约为0.032 5 TOPS/mm^2。因此，存算一体集成芯片架构相对于传统的数据中心系统不仅在性能上取得了显著的提升，还在所需的单芯粒接口速度远低于现有管控指标的前提下，为大规模模型的计算提供了更为可行的解决方案。

2.2 存边架构高并行度数据流

以图3所示LLAMa模型为例，我们对大模型全连接层算力和存储容量进行分析，其三层连续的全连接层网络的算力需求为：（4 096×11 008+11 008×4 096+4 096×4 096）×32×10 000 ×2≈68 TOPS；存储容量为：（4 096×11 008+11 008×4 096+4 096×4 096）×32≈3.4 GB，即模型的算力需求与存储容量的比值为目标令牌速率，与网络大小无关。在数据中心中，令牌速率约为1～10 000个/秒，经典的卷积神经网络模型ResNet-50的算力与存储比为4.1×帧率GOPS/25 MB=164×帧率（GOPS/kB)），因此大模型的算力存储比远低于以卷积神经网络（CNN）为主的传统深度神经网络（DNN）模型的算力存储比。传统交叉阵列架构算力存储比为时钟频率×2。为适应大模型的算力存储比，我们提出了存边计算架构（COMB），即将乘加计算逻辑分布在片上权重缓冲静态随机存储器（SRAM）的边缘，算力存储比



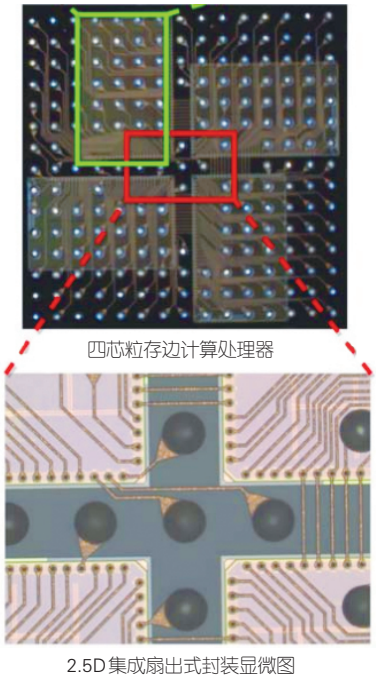
▲图4 存算分离和存算一体架构对比

为时钟频率×2/存储深度。近存计算架构中广泛使用的数据流映射方法完全可以运用在存边计算架构中，权重在计算开始前预先加载在COMB宏中，权重沿输入通道方向切块后，可以展平存入COMB宏的不同列。同时我们可以利用多个COMB Marco电路提高输出通道方向的并行度，完成空间并行计算。

2.3 存算一体技术分类

目前业界已有一些存储颗粒形态的存边计算商业实现方案：海力士（SK Hynix）提出的AiM的每颗DRAM芯粒含有0.5 GB的存储和512 GFLOPS的算力；三星提出的LPDDR5-PIM（存内计算）颗粒的峰值算力可达102.4 GFLOPS。与NPU相比，该设计提升了4.5倍的算力，并节省了72%的功耗。然而，高密度DRAM的工艺专用性强，与CMOS逻辑制造工艺的兼容性较差，且受制于读破坏和电荷泄漏，需要定期刷新存储。

传统嵌入式存储介质SRAM工艺下的微缩比例远远低于逻辑微缩比例。考虑到光刻极限，单芯片的最大SRAM在100 MB量级，且难以发生剧变。因此在过去，集成度一直限制了SRAM存算一体的发展。但随着2.5D/3D堆叠技术的发展，代工厂有望在SRAM上实现更高的集成密度，实现投影面积上等效晶体管密度的提升。如图5所示，我们基于集成扇出（FanOut）工艺，将4颗65 nm SRAM存边计算芯粒集成为一体，实现了SRAM存边计算架构算力和存储容量的显著提升。对于超过4颗芯粒集成的情况，映射方法尚需优化以实现算力随着芯粒数量的线性增长。除此之外，另一种存储颗粒形态的存边计算实现方案是阻变存储器（RRAM）。RRAM是一种能够通过改变二端器件的阻值来存储信息的技术，具有与CMOS工艺兼容性高、非易失、低读取功耗等特点。基于1TnR的RRAM存储器阵列通过三维堆叠技术，能够实现接近DRAM的高密度存储。这一技术趋势为存算一体提供了更为灵活和高效的解决方案。



▲图5 四芯粒2.5D集成Fanout封装[8]

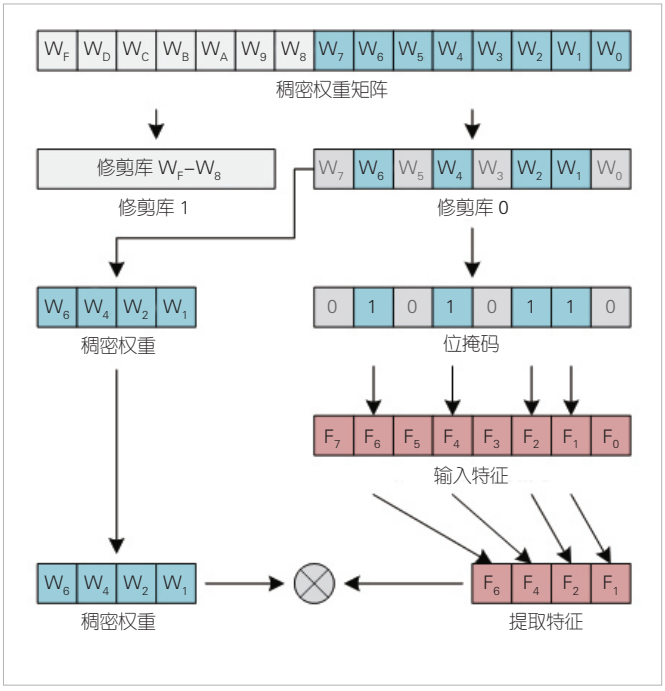
3 轻量化-存内压缩的协同设计

3.1 稀疏网络在存算一体上的部署挑战

随着参数和算力需求的不断增加，大型模型网络的存算一体架构的部署面临更多的挑战。幸运的是，稀疏技术为这一问题提供了一种软硬件协同设计的解决方案。首先，通过对大型模型网络的全连接层进行权重修剪，能够明显减少在生成查询、键和值矩阵时的参数存储需求。其次，大型模型网络所特有的注意力稀疏性进一步减少了自我注意机制的计算和存储需求。然而，在加速稀疏模型的存算一体架构中，仍然存在一些问题。传统的存算一体架构通常以一个交叉杆的形式组织来支持阵列级的计算并行性。在将非结构化剪枝的权重矩阵映射到交叉杆时，存储单元仍然需要保留零值权重，以维持计算的同步性。相较之下，结构化剪枝技术与并行处理更为兼容，但这会降低网络准确性。为了应对这些挑战，我们提出了一种存内稠密权重系数存储方案和基于蝶形网络的存算一体稀疏提取的激活拓扑网络。

3.2 存内稠密权重系数存储

图6展示了模型权重系数稀疏化和稠密存储方案的流程。首先，权重向量被划分为不同的剪枝子组，每个子组具有相同的大小，并按照预定义的稀疏度进行修剪。为了确定稀疏率和剪枝子组的大小，我们在Enwik-8和Text-8任务上使用12层注意力模型。在通过全局修剪对网络进行稀疏化时，我们发现在剪枝子组大小为32、修剪3/4的权重时，网络性能保持不变。因此，我们将剪枝子组大小设置为32，稀疏率设置为75%，以进行稀疏前馈计算。随后，剪枝后的权重被压缩为密集向量和用二进制编码表示的比特掩码，后者可以指示稠密权重的原始位置。最后，根据比特掩码的信息，我们需要从原始输入中提取和路由那些未跳过的输入特征。这一过程实现了对稀疏权重的有效处理。最终的乘积是通过将这两个稠密向量相乘得到的。整个流程的顺序性和稳



▲图6 稀疏–密集计算流程

健性保证了功能的正确性和高效性。

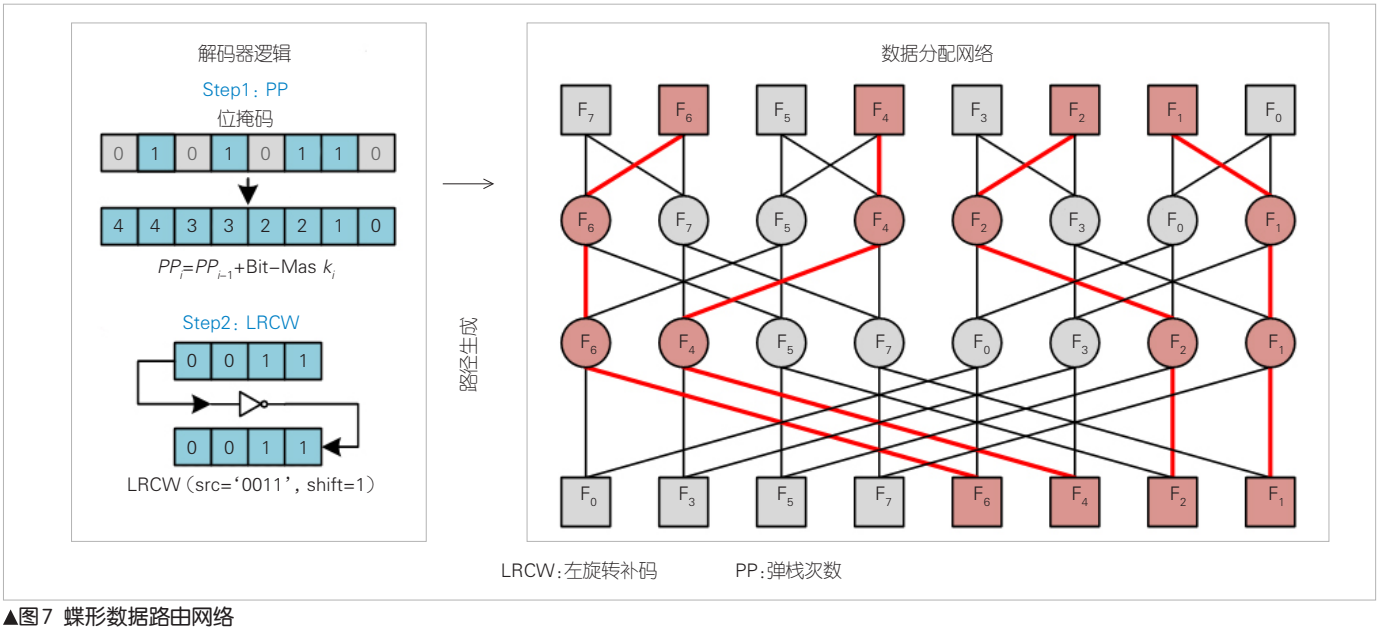
3.3 基于蝶形网络的稀疏提取拓扑

我们运用蝶形网络来提取压缩后的稠密权重所对应的输入激活特征。如图7所示，红色的特征经过蝶形的拓扑网络后，被路由至右侧。这个蝶形网络基于传输管的实现，而传输管的控制信号由解码器实时产生。解码器逻辑接收稠密权重的比特掩码，然后生成控制比特以配置蝶形网络中数据分发的路径。解码机制主要包含两个操作，即前缀pop计数和左旋转补码（LRCW）。前缀pop计数扫描位掩码的序列，并输出当前位置之前1的总数。LRCW是一个标准的左旋转，其唯一不同之处在于移位在任何时候都以补码形式表示。通过这样的操作，我们能够有效地处理比特掩码，从而实现对蝶形网络的灵活配置和输入特征的提取。

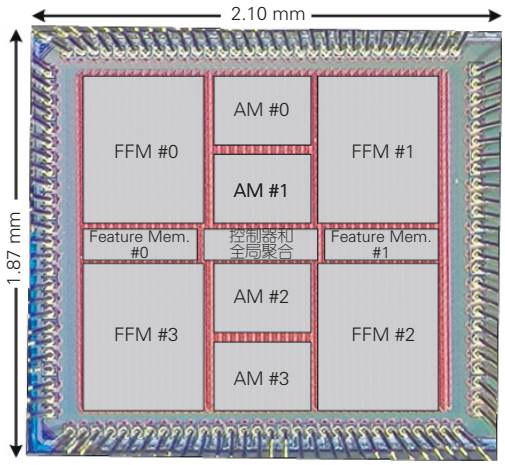
图8显示了采用28 nm CMOS工艺制造的芯片，该芯片工作频率高达320 MHz，总功耗为250.65 mW。考虑到网络稀疏性，该芯片峰值性能为3.3 TOPS。芯片面积3.93 mm^2，面积效率为0.85 TOPS/m^2。该芯片在生成查询、键和值矩阵和整体注意力方面分别实现了高达11.83/25.22 TOPS/W的系统能效。上述轻量化–存内压缩协同设计方案实现了稀疏网络在存算一体硬件上的稠密映射，显著提高存储密度和计算能效。

4 结束语

针对十亿级以上规模的大模型网络应用场景，目前的GPU/TPU+DRAM分离计算架构难以满足不断增长的系数数据传输带宽需求。为了缓解这一问题，存算一体的解决方案，特别是存边计算型的存储颗粒尤为重要，它们有望有效提高带宽。DRAM存算因具有高密度的特点，SRAM和RRAM因其具有高能效特点而备受关注。同时，存内压缩技术的应用可以实现稀疏网络在存算一体硬件上的稠密映射，从而同时提高存储密度和计算的能量效率。因此，在未来的



▲图7 蝶形数据路由网络



工艺节点	28 nm	
精度	INT8	
定制单元晶体管数目	8 T	80 kB
	10 T	64 kB
数字静态随机存储器	96 kB	
芯片面积/mm^2	3.93	
电压/V	0.64~1.03	
频率/MHz	20~320	
功耗/mW	8.27~250.65	
延迟/ms	Enwik8	841
	Text8	896
峰值性能/TOPS	3.33@320 MHz，1.03 V	
能量效率/(TOPS/W)	宏单元	2.52~32.17
	系统	1.96~25.22
面积效率(TOPS/mm^2)	0.85	

AM：注意力单元　FFM：前馈单元

▲图8 芯片照片和汇总表

发展中，矢量计算CPU与存算颗粒的结合有望成为大模型专用的硬件架构。这样的整合能够更好地应对大模型的计算需求，为数据中心芯片带来更为可持续和高效的解决方案。

参考文献

[1] JIAO Y, HAN L, JIN R, et al. 7.2 A 12nm programmable convolution-efficient neural-processing-unit chip achieving 825TOPS [C]// 2020 IEEE International Solid- State Circuits Conference - (ISSCC). IEEE, 2020: 136-140. DOI: 10.1109/ISSCC19947.2020.9062984
[2] DEAN J. 1.1 The deep learning revolution and its implications for computer architecture and chip design [C]//2020 IEEE International Solid- State Circuits Conference - (ISSCC). IEEE, 2020: 8-14. DOI: 10.1109/ISSCC19947.2020.9063049
[3] LIU S W, LI P Z, ZHANG J S, et al. 16.2 A 28nm 53.8TOPS/W 8b sparse transformer accelerator with In-memory butterfly zero skipper for unstructured-pruned NN and CIM-based local-attention-reusable engine [C]//Proceedings of IEEE International Solid- State Circuits Conference (ISSCC). IEEE, 2023: 250-252. DOI: 10.1109/isscc42615.2023.10067360
[4] STOW D, XIE Y, SIDDIQUA T, et al. Cost-effective design of scalable high-performance systems using active and passive interposers [C]//Proceedings of IEEE/ACM International Conference on Computer-Aided Design (ICCAD). IEEE, 2017: 728-735. DOI: 10.1109/iccad.2017.8203849
[5] GOMES W, KHUSHU S, INGERLY B D, et al. 8.1 Lakefield and mobility compute: a 3D stacked 10nm and 22FFL hybrid processor system in 12×12mm2, 1mm package-on-package [C]// 2020 IEEE International Solid- State Circuits Conference - (ISSCC). IEEE, 2020: 144-146. DOI: 10.1109/ISSCC19947.2020.9062957
[6] NAFFZIGER S, LEPAK K, PARASCHOU M, et al. 2.2 AMD chiplet architecture for high-performance server and desktop products [C]//Proceedings of IEEE International Solid- State Circuits Conference - (ISSCC). IEEE, 2020: 44-45. DOI: 10.1109/isscc19947.2020.9063103
[7] SHAO Y S, CLEMONS J, VENKATESAN R, et al. Simba: scaling deep-learning inference with multi-chip-module-based architecture [C]//Proceedings of the 52nd Annual IEEE/ACM International Symposium on Microarchitecture. ACM, 2019: 44-45. DOI: 10.1145/3352460.3358302
[8] ZHU H Z, JIAO B, ZHANG J S, et al. COMB-MCM: computing-on-memory-boundary NN processor with bipolar bitwise sparsity optimization for scalable multi-chiplet-module edge machine learning [C]//2022 IEEE International Solid-State Circuits Conference (ISSCC). IEEE, 2022: 1-3. DOI: 10.1109/ISSCC42614.2022.9731657

作　者　简　介

何斯琪，复旦大学集成芯片与系统全国重点实验室在读硕士研究生；主要研究方向为面向大模型的存算一体SOC研究、深度学习的算法硬件协同设计；发表论文6篇。

穆琛，复旦大学集成芯片与系统全国重点实验室在读博士研究生；主要研究方向为基于易失性、非易失性存储器混合的存算一体SOC研究，通过算法架构电路协同的方式进行功耗及性能优化；发表论文4篇，申请专利2项。

陈迟晓，复旦大学芯片与系统前沿技术研究院副研究员、集成芯片与系统全国重点实验室集成芯片创新中心主任、国家优青、上海市青年科技启明星；研究方向包括人工智能芯片与系统、数模混合集成电路EDA以及先进封装、Chiplet集成；主持多个国家自然科学基金委面上项目；获上海市技术进步奖一等奖；发表论文50余篇，授权中国发明专利9项。

低资源集群中的大语言模型分布式推理技术

Accelerating Distributed Inference of Large Language Models in Low-Resource Clusters

冯文佼/FENG Wenjiao， 李宗航/LI Zonghang，
虞红芳/YU Hongfang
（电子科技大学，中国 成都 611731）
（University of Electronic Science and Technology of China，Chengdu 611731，China）

DOI：10.12142/ZTETJ.202402007
网络出版地址：http://kns.cnki.net/kcms/detail/34.1228.TN.20240404.2315.002.html
网络出版日期：2024-04-08
收稿日期：2024-02-20

摘要：探索了一种并行能力更强、具有更好兼容性的大语言模型（LLM）分布式推理范式。该范式专为弱算力、小显存环境设计。同时面向主机内外差异带宽，设计了基于通信树的高效All-Reduce组通信技术；针对小显存集群，设计了细粒度的显存管理与调度技术。最后，基于这些关键技术，构建了一套针对资源受限场景的LLM推理软件系统，旨在用数量有限的低资源设备，最大化能推理的LLM，同时通过优化通信策略与计算调度加速分布式推理。实验证明，在应用上述技术后，本方案的首词元生成延迟降低34%~61%，每秒生成词元吞吐量提升52%~150%，显存占用降低61%。

关键词：LLM分布式推理范式；资源受限场景；优化通信策略与计算调度

Abstract: A distributed inference paradigm for large language model (LLM) with stronger parallelism and better compatibility is explored, which is designed for weak computing power and small memory environments. Meanwhile, an efficient All-Reduce group communication technique based on communication tree is designed for the different bandwidths inside and outside the host, and a fine-grained memory management and scheduling technique is designed for small memory clusters. Finally, based on these key techniques, a set of LLM inference software system for resource-constrained scenarios is constructed, aiming to maximize the LLMs that can be inferenced with a limited number of low-resource devices, and at the same time accelerating the distributed inference by optimizing the communication strategy and computation scheduling. Experiments demonstrate that after applying the above techniques, the first lexical element generation latency is reduced by 34%~61%, the lexical element generation throughput per second is increased by 52%~150%, and the memory occupation is reduced by 61%.

Keywords: LLM distributed inference paradigm; resource-constrained scenarios; communication and computation scheduling optimization

引用格式：冯文佼, 李宗航, 虞红芳 . 低资源集群中的大语言模型分布式推理技术 [J]. 中兴通讯技术, 2024, 30(2): 43-49. DOI: 10.12142/ZTETJ.202402007
Citation: FENG W J, LI Z H, YU H F. Accelerating distributed inference of large language models in low-resource clusters [J]. ZTE technology journal, 2024, 30(2): 43-49. DOI: 10.12142/ZTETJ.202402007

作为科技革命的核心，人工智能（AI）在计算机视觉和自然语言处理等领域取得了重大进步。OpenAI在2022年底发布的ChatGPT[1]引领了大语言模型（LLM）时代，引发了对人工智能技术潜力的广泛探讨。然而，在全球AI技术竞争日益激烈和国际环境变化的背景下，高性能计算资源变得更加珍贵，尤其是面对AI芯片出口的限制，中国AI技术的独立发展变得迫在眉睫。由于存在技术鸿沟，中国AI硬件在短期内仍然面临着诸如弱算力、小显存和多机低互联带宽的技术挑战。为推动大模型AI产业的发展，中国学术界和工业界提出了在资源受限环境下进行LLM推理的策略。通过整合中低端算力资源，该策略实现超大模型的高效运行，既减少了对国外高端硬件的依赖，也为中小企业和教育机构提供了低成本的推理与部署方案，促进了国产AI计算卡的快速发展。因此，研究低资源环境下的LLM推理优化技术，成为了推动中国AI发展“降本增效”的关键。

现有LLM推理系统，如DeepSpeed[2]和FasterTransformer[3]，主要为强算力、高带宽、大显存的高性能智算中心提供高效的LLM推理能力。但与高性能智算中心相比，

在低资源条件下进行LLM推理仍存在一些不足。1）单张计算卡在算力和显存容量上面临明显限制。例如，NVIDIA的A100和H100这类图形处理器（GPU），在FP16运算性能上比中国的寒武纪思元、燧原邃思领先逾7倍，显存容量超5倍。对于中国的LLM推理应用来说，计算效率和存储能力成为了明显的瓶颈。2）多主机间的通信带宽远小于主机内的高速网络带宽。较大的模型不适合单张计算卡，需要依赖多卡服务器集群以适应显存。这也使我们能够将上述的计算成本和显存分摊到所有计算卡上，但代价是引入计算卡间通信。而在中低端数据中心内，主机间的网络带宽普遍限制在1~25 Gbit/s，与主机内可达100 Gbit/s的显存带宽和互联带宽相距甚远。这使得多机间互联网络的通信效率成为制约分布式推理性能的主要瓶颈。

此外，当前LLM主要采用Transformer架构[4]。它的主要思想是通过自注意力机制获取序列的全局信息，并将这些信息通过网络层进行传递。区别于传统的卷积神经网络（CNN）和循环神经网络（RNN），Transformer架构由于具有多个独立的注意力头，因此不需要按照时间步骤进行计算，具有更强的并行计算能力。为了实现最佳的性能和资源利用率，现在很多研究致力于自动混合并行推理，包括AlpaServe[5]、FlexFlow-Serve[6]和SpotServe[7]等。这些框架能够将自动搜索算法应用于LLM的推理过程，以确定最有效的并行策略。然而，分布式推理面临的主要挑战之一是数据通信产生了额外负担，因为这可能增加总体推理响应时间。尽管现有策略优化了并行计算，但它们往往忽略了针对Transformer架构特有通信需求的优化，这可能导致在推理过程中出现更加明显的延迟。

考虑到Transformer架构固有的内存密集型特性，高效的显存管理仍然是LLM分布式推理中面临的首要挑战。ZeRO-Offload[8]和ZeRO-Infinity[9]支持内存卸载，将GPU的显存压力分担到CPU甚至NVMe内存上，从而打破GPU的显存限制。但此类方法需要所有计算卡间拥有高速连接，因此使用场景将会受到很大的限制。

针对上述挑战，本文提出了适用于低资源集群的LLM分布式推理技术，实现用数量有限的低资源设备，最大化能推理的LLM，同时通过优化通信策略与计算调度来加速推理。

1 问题与动机分析

由于LLM推理对设备算力和显存容量有较高要求，Megatron-LM[10]通过张量并行将模型层，例如注意力、全连接前馈网络（FFN），从内部维度（例如头部、隐藏层）分割成多个部分，并将每个层部署在单独的计算卡上。但这种朴素的张量并行存在一个问题：自注意力的输出必须通过LayerNorm才能输入到FFN中进行计算。LayerNorm的正确性依赖于所有计算卡的自注意力结果，这是因为单卡结果无法确保其准确性。为此，Megatron-LM提出Reduce+LayerNorm+Broadcast算子，即计算卡完成自注意力输出后，先聚合（Reduce）到一卡执行LayerNorm，再将结果广播（Broadcast）回各卡继续多层感知机（MLP）计算。虽然该算子解决了LayerNorm层的并行问题，但它仍依赖单卡执行Reduce、LayerNorm及Broadcast。一方面，这种中心化的计算与通信算子会遭遇单点瓶颈；另一方面，这种算子的适用通信原语局限于Reduce和Broadcast，与诸多经典的All-Reduce通信库及其高效的All-Reduce原语实现（如Ring、Three-Phase Ring等）均不兼容。

由于典型数据中心的分层网络结构限制了跨主机带宽，All-Reduce的性能也会受阻。大规模LLM推理需要多主机合作以满足算力和显存要求。尽管单机多卡间可通过NVLink和高速串行计算机扩展总线标准（PCIe）实现高速通信，但各主机通常按机架分组并连接至架顶式（ToR）交换机。其中，机架内各主机通过1~25 Gbit/s的完整链路平分带宽进行互联，这限制了多主机间All-Reduce的通信效率。相关研究集中于优化模型训练阶段的All-Reduce通信，通过探测网络结构并制定分层聚合策略以适应网络变化，从而解决长期通信不平衡问题，但这并不完全适用于推理阶段。与训练不同，推理尤其是在线推理的持续时间较短，其核心目标是实现低延迟和高吞吐。因此，推理阶段更需针对带宽差异引致的通信瓶颈进行优化。

此外，为了实现用数量有限的低资源设备最大化能推理的LLM，同时考虑到Transformer架构固有内存密集性，高效的显存管理仍然是LLM分布式推理中面临的首要挑战。现有推理系统[3-11]基于高性能智算中心开发了一系列内存卸载技术，例如：通过频繁通信实现了GPU显存负载转移至CPU或NVMe存储，有效突破显存限制。然而，这些推理系统往往沿用了为训练阶段设计的卸载技术[8-9,12-14]，直接应用于资源受限的分布式推理可能不理想。因为这些技术在资源受限环境下可能导致对更多计算卡和高并行度的依赖，增加通信复杂性，并且主机间的低带宽难以支持这种强度的通信。同时，这些技术忽略了生成推理的特殊计算属性，未能利用面向吞吐量的LLM推理计算的结构，并错过了有效调度输入输出（I/O）流量的绝佳机会。这些先前的工作促使我们设计一套适用于低资源集群的LLM分布式推理技术方案。该方案引入了一种高兼容的分布式推理范式，同时特别

关注主机内外带宽差异以及如何最大化LLM推理的潜力。

在本文中，我们研究了一种面向弱算力、小显存的高兼容的分布式推理范式。该范式能支持All-Reduce通信原语。具体而言，我们揭示了在进入非线性层之前，LayerNorm和Broadcast两个操作是可交换的。基于此我们提出了一个创新方案：将传统的Reduce+LayerNorm+Broadcast算子简化为All-Reduce+LayerNorm算子。这一新范式旨在全面支持All-Reduce通信原语，使之能在不同场景中利用多样化的通信库来实现高效的分布式推理。

在中低端数据中心内机架规模下涉及跨主机的All-Reduce通信时，分布式推理低带宽网络会带来明显的性能瓶颈问题。为解决这一问题，我们提出了一种面向主机内外差异带宽的高性能All-Reduce通信算法。具体来说，我们根据主机内和机架内带宽特点的差异性，实现基于通信树的高效All-Reduce组通信库，有效组织分布式推理的中间计算结果聚合与分发，从而减少跨主机通信并充分利用内部高速带宽。

此外，我们还探索了面向LLM、小显存集群的显存管理与调度，旨在低资源环境中实现更大规模与更高效的LLM推理。我们采用了动态调度模型参数的方法，包括及时回收未使用的参数空间以减少显存占用，并预加载即将使用的参数以消除轮次间的等待时间，从而无缝加速推理过程。这种策略通过细粒度控制显存的使用，降低了峰值显存需求，即使在显存有限的硬件条件下也能高效地执行大规模模型推理，在确保推理性能的同时提高硬件资源的使用效率和成本效益。

为了实现上述想法，一些技术难题仍需要解决：

难题1：如何保证本文提出的范式在保持理论计算正确性的同时，与现有张量并行范式具有等价的推理计算效率和资源消耗。

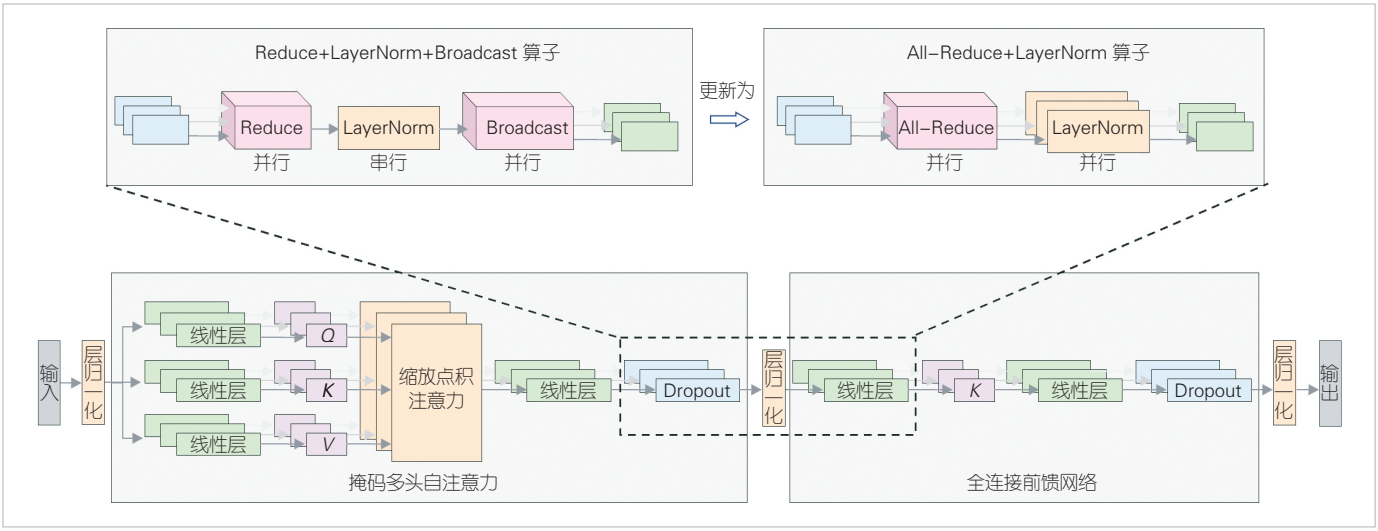
难题2：如何保证本文提出的分层聚合算法在理论计算结果正确的前提下，最大限度地减少由机架内带宽的低带宽网络引起的通信开销。

难题3：基于新型分布式训练范式，如何精准把控推理过程中所需的模型参数在显存中的加载和卸载时机，即明确何时将这些相关数据载入显存以进行有效的计算，以及何时将其从显存中移除以优化资源使用。

2 方案设计

2.1 面向弱算力、小显存的高兼容的分布式推理范式

Megatron-LM采用张量并行来应对大规模模型对高算力和大显存的强依赖，并通过引入Reduce+LayerNorm+Broadcast算子确保计算的准确性。然而，这种中心化的并行范式存在单点瓶颈，并且不支持如信息传递接口（MPI）、NVIDIA集合通信库（NCCL）等主流All-Reduce通信库，影响了其兼容性和效率。为此，我们研究了一种兼容性更好的张量并行范式。该范式能支持All-Reduce通信原语，使之能在不同场景中利用多样化的通信库来实现高效的分布式推理，并且与现有张量并行范式一样具有等价的推理计算效率和显存消耗。具体如图1所示，在进入MLP之前LayerNorm和Broadcast两个操作是可交换的。基于这一关键发现，本文提出将Reduce+LayerNorm+Broadcast算子合并为All-Reduce+LayerNorm算子。接下来，我们将从理论正确性和资



▲图1 面向弱算力、小显存的高兼容分布式推理范式示意图

源消耗两个维度来深入分析这一新算子的性能。

在计算结果理论正确性方面，该推理范式先在单个计算卡上对Reduce后数据进行LayerNorm，再将结果Broadcast给其他计算卡。这与先将Reduce后的结果Broadcast给其他计算卡，再在各计算卡上分别进行LayerNorm，计算得到的结果等价。而在资源消耗方面，All-Reduce+LayerNorm算子不会牺牲显存，因为中间结果Z通过各计算卡并行LayerNorm操作。得到Z之后，LayerNorm产生的临时变量立即被释放。因此，这种方法主要影响峰值显存使用而非总显存。

总的来说，我们提出一种高兼容的分布式推理范式，该范式将Reduce+LayerNorm+Broadcast算子合并为All-Reduce+LayerNorm算子，在确保计算结果理论正确性、资源消耗等价的前提下，统一LLM张量并行范式的通信原语为All-Reduce。一方面，我们可以根据实际环境，灵活使用MPI、Gloo、NCCL、Hovorod、PS等第三方通信库（或自研通信库）来满足个性化的推理需求；另一方面，All-Reduce原语的执行效率比分别执行Reduce和Broadcast原语更高，具有更宽阔的通信优化空间。

2.2 面向主机内外差异带宽的高性能All-Reduce通信算法

当前基于All-Reduce的通信优化研究主要集中在缓解模型训练阶段的长期通信不平衡问题。然而，在追求低延迟和高吞吐的推理阶段，跨主机All-Reduce通信中低带宽网络引发的瓶颈问题更值得关注。如图2所示，数据中心的分层拓扑结构将机器分至机架并连接至ToR交换机，以保障机架内主机间共享完整链路带宽。但带宽通常局限于1~25 Gbit/s，这与主机内多卡间上百Gbit/s的高速通信相距甚远[15]。为此我们开发了一种基于通信树的高效All-Reduce组通信库，以减少跨主机通信并充分利用内部高速带宽。具体来说，对于一次推理任务，当需要进行All-Reduce操作时，通信树的构造过程如下：

首先在各主机内部选出性能最优的计算节点作为本地主节点（LM），用于本地聚合。所有主机中的LM之一被选为全局聚合的全局主控（GM）。通信树的通信按照以下4个步骤完成：

1）每个计算节点将自己的局部计算结果发送到各自的LM（仅限主机内推理流量）；

2）所有的LM将本地聚合结果发送至GM以进行全局聚合（仅主机间流量）；

3）GM完成全局聚合，然后使用反向路由将全局聚合结果传播回LM；

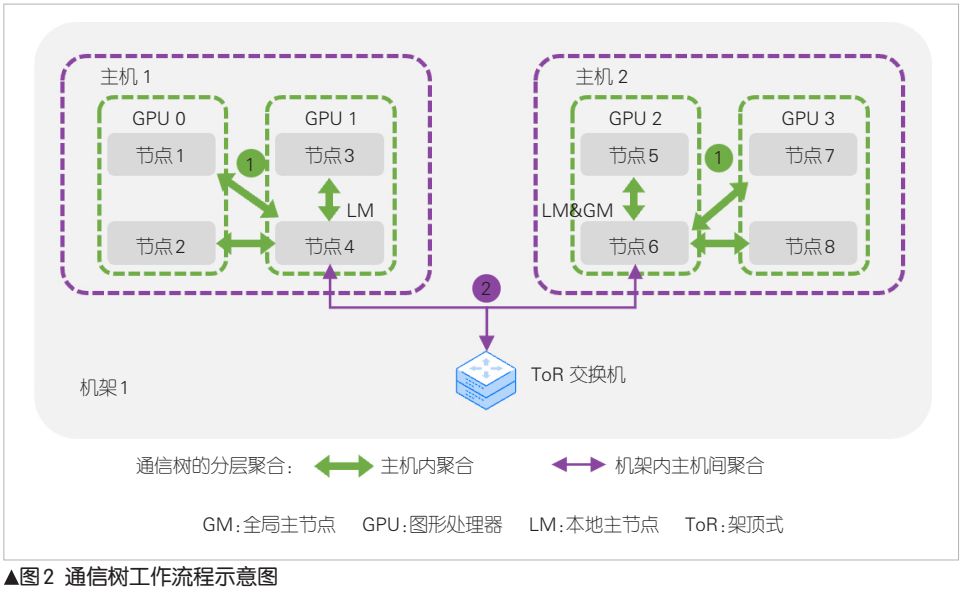
4）LM将全局聚合块扇出到所有计算节点。

与朴素All-Reduce主机间通信次数相比，采用通信树策略后，主机间的All-Reduce通信次数降为仅需主机的数量-1次。可以看到，本方案充分利用了数据中心网络的结构特点，通过优化内部节点的通信路径，有效管理推理中间结果的聚合和分发过程，从而大幅度降低了主机间的通信频率和通信延迟，提高了整体推理效率。

2.3 面向LLM、小显存集群的显存管理与调度

尽管针对LLM的内存卸载技术在高性能计算中心依然有效，但这些主要为训练而设计的技术在资源受限的分布式推理场景中应用可能不理想。它们可能增加对计算资源的依赖，加重通信负担，同时忽略了推理特有的计算需求和优化吞吐量的机会。通常来讲，LLM推理包含预填充和解码两阶段。其中，预填充阶段并行处理输入，解码阶段依赖之前所有tokens信息生成新tokens。为提高效率，现有工作提出将这些信息以键（K）和值（V）的形式缓存于显存中，大大减少了重复计算次数。但随着对长序列推理的需求不断增长，与模型权重和其他激活所需的工作空间相比，KV缓存的显存占用成为主要优化目标。

为解决这个问题，我们开发了一套针对LLM和小显存集群的细粒度显存管理与调度机制，目



▲图2 通信树工作流程示意图

的是在资源受限的环境下，实现更大规模的LLM推理。其中，这一机制包含两个核心模块：最小化显存占用机制和预加载机制。通过将Transformer模型的每个Layer视为独立状态，并将参数分散到不同GPU上，最小化显存占用机制确保每个计算单元仅保留当前必需的参数片段，大幅降低了总体显存需求。同时，预加载机制能在当前计算进行前加载下一步所需的参数，有效消除了推理过程中的等待时间，进一步提升了推理效率。这两个模块的协同工作，使得我们的显存管理策略能够在减少资源消耗的同时保证模型推理的连续性和吞吐。

1）最小化显存占用机制

已知Transformer模型由多个Layer串连而成，我们将每个Layer视为一个独立状态，同时，根据新型分布式推理范式将每个Layer中的参数切分为不同的部分。每个GPU仅维护对应的部分。对于特定的推理计算，用b表示批量大小，s表示输入序列长度，n表示输出序列长度，h表示隐藏维度，L表示Transformer层数，a表示attention heads。考虑有N个GPU执行推理。对于batch X，GPU_n处理Layer 1到Layer L的s_{n1}~s_{n2}参数片段，包括a_{n1}~a_{n2}注意力头和相应的MLP参数切片。针对batch X中的某个prompt生成一个词元过程，具体的显存管理与调度流程如图3所示。

显存管理与计算线程并行运行，前者负责模型参数在显存与内存间的调度，后者执行GPU上的张量并行计算。在推理的每个步骤i中，系统识别Layer(i)作为当前的活动状态。对于GPU_n，其计算线程专注于执行Layer(i)内部特定的s_1~s_2参数切片的计算任务。与此同时，显存管理线程负责从显存中卸载掉之前步骤Layer(i − 1)的s_1~s_2参数切片和对应注意力头的KV缓存。与传统方法相比，显存需完整存储模型参数及KV缓存。本策略确保计算卡在任一时刻仅保留必要的参数，从而在资源受限的环境中实现更大规模的模型推理。

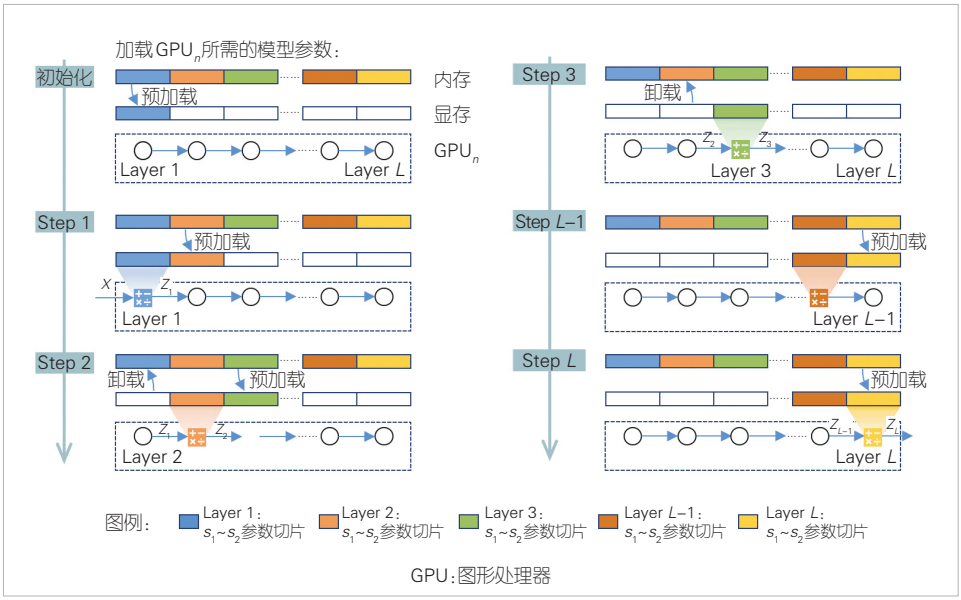
2）预加载机制

然而，上述的串行计算存在一个明显的缺陷：在完成Layer(i)的计算后，推理过程需等待Layer(i + 1)的参数加载至显存，这显著降低了推理效率。因此，我们引入了预加载机制，允许显存管理线程在Layer(i)计算进行时，提前将Layer(i + 1)的s_1~s_2参数切片和对应注意力头的KV缓存从内存预加载至显存。该机制使用少量的显存，消除了计算停滞，保障了推理流程的无缝衔接。

3）支撑的模型范围对比分析

在资源受限条件下，通过上述细粒度的显存管理与调度，这套范式理论上可以支撑多大的模型？考虑fp16中的GPT3-175B模型（L=96，h=12 288），峰值时存储KV缓存的总字节数为$4 \times b \times h \times (s+n)$。模型所需的总显存为350 GB，存储KV缓存所需的总显存为816 GB（其中，b=16，s=512，n=32），总显存需求约为1 166 GB，平均每层需要12 GB显存。在四卡系统中，每卡理论上承担3 GB，即使考虑额外的缓冲和预加载空间，每卡的显存使用也不会超过6 GB。相比之下，传统GPU-only方案每卡需承担290 GB。因此，我们的方法可以推理比GPU-only的解决方案大48倍的模型（每卡承担6 GB与290 GB），超越DeepSpeed Inference的25倍[3]。这表明我们所提方法在模型扩展性方面具有优越性。

总的来说，此方法在保证推理性能的基础上，通过细粒度管理与调度显存，在显存受限的硬件环境中也能高效推理更大规模的模型，极大提高了硬件资源的利用率和成本效益。



▲图3 细粒度的显存管理与调度流程图

3 实验评估

本节我们将围绕两个方面来评估所提方案的性能：1）对首词元生成延迟的降低以及每秒生成词元吞吐量的提升效果；2）量化本系统显存管理与调度机制的优势。

3.1 实验平台设置

在构建系统时，我们选用

PyTorch[16]作为核心框架。在计算方面，我们重新设计了分布式推理的架构，并实现了精细的显存管理及调度策略。在通信层面上，我们依托PyTorch-DDP，打造了一种基于通信树的高效All-Reduce集群通信机制。我们在两台配置有双Intel(R) Xeon(R) E5-2678 v3 CPU、4块NVIDIA RTX 2080TI GPU、128 GB系统内存及44 GB总显存的主机上开展实验。主机间通过1 GB网络带宽互联。实验采用Meta AI发布的LLaMA-3B。表2展示了默认的超参数配置。

3.2 延迟和吞吐量

我们选择基于分布式数据并行（DDP）的原生All-Reduce作为Benchmark，采用参数服务器（PS）架构。worker节点通过采用“星形”拓扑结构进行通信，即多个worker直接与中心服务器进行数据交换。

我们首先对Benchmark和本方案在首词元生成延迟及每秒生成词元吞吐量方面进行了比较测试。其中，首词元生成延迟涵盖模型处理输入并自回归生成下一词元的计算及通信延迟，每秒生成词元吞吐量用每秒可以处理的词元数来衡量。如图4所示，我们测试了不同的输入词元数。相比于Benchmark，本方案的首词元生成延迟降低34%~61%，每秒生成词元吞吐量提升52%~150%。这证明了上述技术的有效性。

▼表2 LLaMA-3B模型默认超参数设置

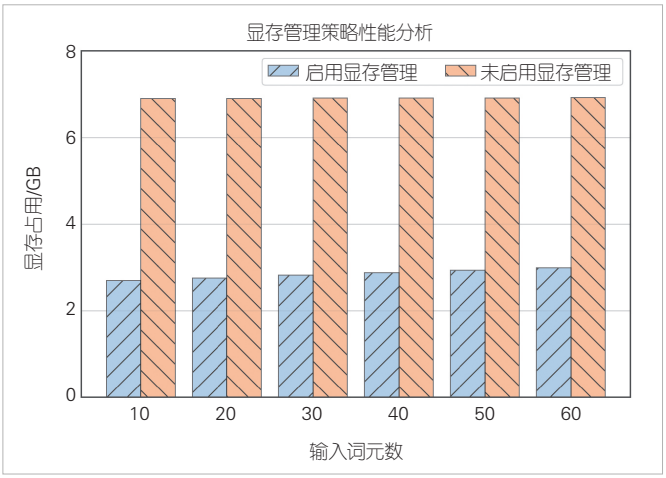
变量名	符号	值
注意力机制中的头数	N_{atten_heads}	32
批量大小	B_{size}	32
隐藏层的维度大小	H_{model}	3 200
模型中的层数	N_{layers}	26
序列长度	seq_len	2 048
词汇表的大小	vocab_size	32 000

3.3 显存占用

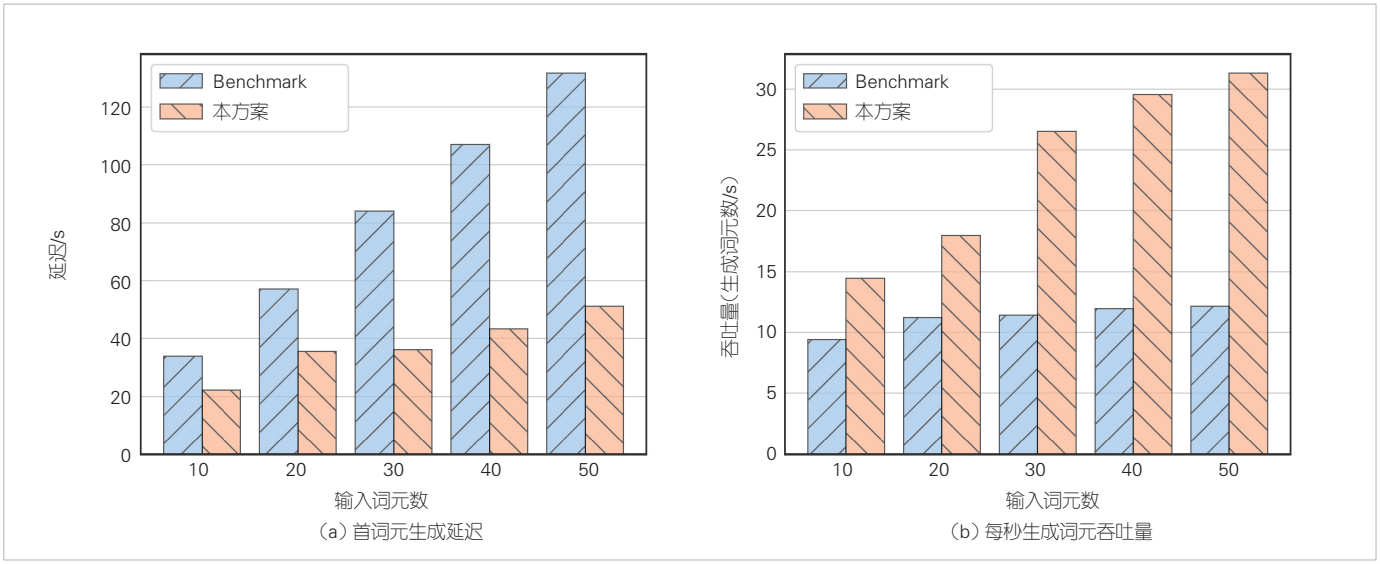
为评估本方案的显存管理与调度效能，我们比较了启用与未启用本显存管理方案时，首词元生成阶段节点的峰值显存占用情况。图5展示当输入词元数量增加时节点显存占用的线性增长趋势。与未启用显存管理相比，本方案的显存占用降低61%。这也验证了2.3节中的分析。

4 结束语

在面对全球竞争和资源限制的挑战下，我们提出了一种适应弱算力及小显存环境的分布式LLM推理架构。同时通



▲图5 显存管理对节点显存占用的影响



▲图4 不同方案下延迟和吞吐量对比

过独创的适应性通信策略和显存管理方案，我们有效克服了带宽和显存限制，构建了一个高效推理框架，使得有限资源下的LLM推理成为可能。此项成果推进了中国AI的自主发展，为中国AI产业的发展和全球技术多样性贡献了重要力量。

致谢

感谢电子科技大学信息与通信工程学院赵舒心和熊彦旭硕士对本研究技术与实验部分的贡献！

参考文献

[1] OpenAI. ChatGPT [EB/OL]. (2022-12-30) [2024-02-25]. https://openai.com/blog/chatgpt
[2] AMINABADI R Y, RAJBHANDARI S, AHMAD AWAN A, et al. DeepSpeed-inference: enabling efficient inference of transformer models at unprecedented scale [C]//Proceedings of SC22: International Conference for High Performance Computing, Networking, Storage and Analysis. IEEE, 2022: 1-15. DOI: 10.1109/SC41404.2022.00051
[3] NVIDIA. FasterTransformer [EB/OL]. (2022-03-20)[2024-02-25]. https://github.com/NVIDIA/FasterTransformer
[4] VASWANI A, SHAZEER N, PARMAR N, et al. Attention is all you need [C]//Proceedings of the 31st International Conference on Neural Information Processing Systems. ACM, 2017: 6000 - 6010. DOI: 10.5555/3295222.3295349
[5] LI Z H, ZHENG L M, ZHONG Y M, et al. AlpaServe: statistical multiplexing with model parallelism for deep learning serving [EB/OL]. [2024-02-25]. http://arxiv.org/abs/2302.11665
[6] Github. FlexFlow [EB/OL]. [2024-02-25]. https://github. com/Flexflow/FlexFlow/tree/inference
[7] MIAO X P, SHI C N, DUAN J F, et al. SpotServe: serving generative large language models on preemptible instances [EB/OL]. [2024-02-25]. http://arxiv.org/abs/2311.15566
[8] REN J, RAJBHANDARI S, AMINABADI R Y, et al. ZeRO-offload: democratizing billion-scale model training [EB/OL]. [2024-02-25]. http://arxiv.org/abs/2101.06840
[9] RAJBHANDARI S, RUWASE O, RASLEY J, et al. ZeRO-infinity: breaking the GPU memory wall for extreme scale deep learning [EB/OL]. [2024-02-25]. http://arxiv.org/abs/2104.07857
[10] SHOEYBI M, PATWARY M, PURI R, et al. Megatron-LM: training multi-billion parameter language models using model parallelism [EB/OL]. [2024-02-25]. http://arxiv. org/abs/1909.08053.pdf
[11] HuggingFace. Hugging face accelerate [EB/OL]. [2024-02-25]. https://huggingface.co/docs/accelerate/index
[12] LI Y J, PHANISHAYEE A, MURRAY D, et al. Harmony: overcoming the hurdles of GPU memory capacity to train massive DNN models on commodity servers [EB/OL]. [2024-02-25]. http://arxiv.org/abs/2202.01306
[13] HUANG C C, JIN G, LI J Y. SwapAdvisor: pushing deep learning beyond the GPU memory limit via smart swapping [C]// Proceedings of the Twenty-Fifth International Conference on Architectural Support for Programming Languages and Operating Systems. ACM, 2020: 1341 - 1355. DOI: 10.1145/3373376.3378530
[14] WANG L N, YE J M, ZHAO Y Y, et al. Superneurons: dynamic GPU memory management for training deep neural networks [C]//Proceedings of the 23rd ACM SIGPLAN Symposium on Principles and Practice of Parallel Programming. ACM, 2018: 41 - 53. DOI: 10.1145/3178487.3178491
[15] LUO L, NELSON J, CEZE L, et al. Parameter hub: a rack-scale parameter server for distributed deep neural network training [C]//Proceedings of the ACM Symposium on Cloud Computing. ACM, 2018: 41-54. DOI: 10.1145/3267809.3267840
[16] PASZKE A, GROSS S, MASSA F, et al. Pytorch: an imperative style, high-performance deep learning library [EB/OL]. (2019-12-03)[2024-02-25]. https://arxiv.org/abs/1912.01703

作 者 简 介

冯文佼，电子科技大学在读硕士研究生；研究方向为分布式机器学习系统及其优化技术、大模型分布式推理优化技术。

李宗航，电子科技大学在读博士研究生、牛津大学和南洋理工大学访问学者；研究方向包括分布式人工智能、联邦学习和大模型分布式计算；相关研究入选中国通信学会2021领先创新科技成果；发表论文20余篇，授权中国发明专利6项，出版学术著作1部。

虞红芳，电子科技大学教授、博士生导师，信息与通信工程学院副院长；长期致力于智慧网络及应用研究；受邀在全球学术会议上做报告10余次，担任3个网络领域全球高水平期刊的副主编；获得2016年教育部自然科学奖二等奖，主持研发的“跨数据中心高性能分布式机器学习系统GeoMX”和“基于轻量级虚拟化的大规模网络创新平台Klonet”分别获中国通信学会2021年未来网络领先创新科技成果奖、2021年网络5.0创新科技成果奖；发表论文100余篇；授权中国发明专利30余项、美国发明专利2项，出版学术专著4本。

生成式大模型承载网络架构与关键技术探索

Network Architecture and Technologies for Large Generative Models

唐宏/TANG Hong，武娟/WU Juan，徐晓青/XU Xiaoqing，张宁/ZHANG Ning

（中国电信股份有限公司研究院，中国 广州 510630）
(Research Institute of China Telecom Company Ltd., Guangzhou 510630, China)

DOI：10.12142/ZTETJ.202402008
网络出版地址：http://kns.cnki.net/kcms/detail/34.1228.TN.20240408.0924.004.html
网络出版日期：2024-04-09
收稿日期：2024-02-20

摘要：生成式大模型训练需要超大规模低时延、高带宽、高可用的网络承载底座。对生成式大模型下高性能网络基础设施的技术发展路线和实现方案进行了研究，认为商用部署时需针对不同训练阶段的工作负载和流量模式，开展定制化网络架构设计和传输协议优化。流控/拥塞控制技术、负载均衡技术、自动化运维技术和面向广域远程直接内存访问（RDMA）的确定性网络传输技术是未来的重点研究方向。

关键词：生成式大模型；RDMA；网络拥塞控制；网络负载均衡

Abstract: The training of large generative models has posed demands for ultra-large-scale, low latency, high bandwidth, and high-availability network infrastructure. The technological development roadmap and implementation schemes of high-performance network infrastructure for large models are investigated. It is believed that the customized network architecture design and transport protocol optimization should be carried out based on workloads and traffic patterns at different training stages during commercial deployment. Flow control/congestion control technologies, load balancing technologies, automated operation and maintenance solutions, and deterministic network transmission technologies for wide-area remote direct memory access (RDMA) are key research directions for the future.

Keywords: large generative model; RDMA; network congestion control; network load balancing

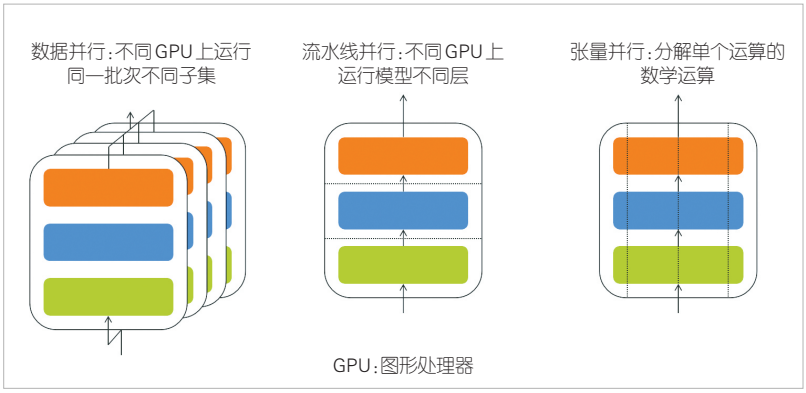
引用格式：唐宏, 武娟, 徐晓青, 等. 生成式大模型承载网络架构与关键技术探索 [J]. 中兴通讯技术, 2024, 30(2): 50-55. DOI: 10.12142/ZTETJ.202402008

Citation: TANG H, WU J, XU X Q, et al. Network architecture and technologies for generative models [J]. ZTE technology journal, 2024, 30(2): 50-55. DOI: 10.12142/ZTETJ.202402008

1 生成式大模型对网络基础设施的挑战

近年来，以ChatGPT、Sora为代表的通用生成式大模型的研究取得了显著进展。生成式大模型的参数规模已实现了从千万级别到万亿级别的飞跃，并朝着十万亿级别前进[1]。由于数据量巨大，需要海量的图形处理器（GPU）做并行计算，而大量的GPU并行计算亟需强大的基础网络支撑。与传统的数据中心网络架构相比，生成式大模型组网呈现以下新需求：

1）超大规模组网[2]。在生成式大模型训练时，数据并行、流水线并行和张量并行同时存在，如图1所示。数据并行和流水线并行所需的“参数面大网”需要跨服务器通信，规模可达十万甚至百万级别的卡数，具有超大规模、高网络容量以及高接入带宽等特点。而实现张量并行的“参数面小网”则通常局限于单个服务器范围内，具有规模小、容量超大以及高接入带宽等特点。



▲图1 3层模型上的并行计算

2）超高带宽。机内通信中GPU间的AllReduce集合通信数据量可达百GB级别。机间GPU通信涉及多种并行模式，产生大量集合通信数据，机间GPU的高速互联对于网络的单端口带宽、节点间的可用链路数量及网络总带宽提出了很高的要求。同时，高速串行计算机扩展总线标准（PCIe）的总线带宽限制了网卡性能的发挥，需适配更高带宽的总线技术以提升机间通信效率。

3）超低时延。对于千亿参数模型来说，通信的端到端耗时占比仅为20%，而对于万亿参数模型，占比增加至50%[1]。传统的流控算法和拥塞控制算法在面对生成式大模型训练网络时，会遇到拥塞头阻、拥塞扩散等挑战。此外，AI训练中流量的特征是“少流”和“大流”，使得传统的等价多路径（ECMP）流量均衡机制因ECMP哈希极化问题造成链路上流量不均而失效。

4）自动化运维。当GPU集群规模达到一定量级后，保障集群系统的稳定高效运行就成为大模型工程化实践中极其重要的环节。与单点GPU故障相比，网络故障会影响数十个甚至更多GPU的连通性。高性能网络的自动化部署、一键式故障定位和业务无感自愈，将决定整个集群的计算稳定性。

随着生成式大模型参数规模的快速增长，传统的数据中心网络架构已经很难满足其训练需求。高性能、高可用的承载网络底座将成为推动其发展的核心基础设施。

2 大模型云网基础架构与关键技术

2.1 网络架构

传统的数据中心拓扑结构为3层的树形拓扑结构。树形结构原理简单，易于部署，但是当面对大模型训练中要求集群内服务器协作完成训练任务的场景时，该结构拓展能力显得不足，服务器间通信受限。

与传统树形网络拓扑中的逐层带宽收敛相比，Fat-Tree网络具有无阻塞和无带宽收敛的特性[3]，目前被主流公有云厂商大规模应用于GPU密集型集群中[4]，如图2所示。单台服务器配备高性能的400 Gbit/s NIC网卡，K台服务器为一组，通过架顶式（ToR）交换机互接。ToR交换机与聚合交换机相连形成一个Pod，实现跨机架的连接。Pod与主干交换机相连，确保中央处理器（CPU）集群中的服务器能够实现any-to-any通信。但是当网络大规模扩展时，受到核心交换机端口数限制，Fat-Tree的横向拓展能力变差[5]。同时，为数以万计的GPU提供非阻塞连接的成本非常昂贵。

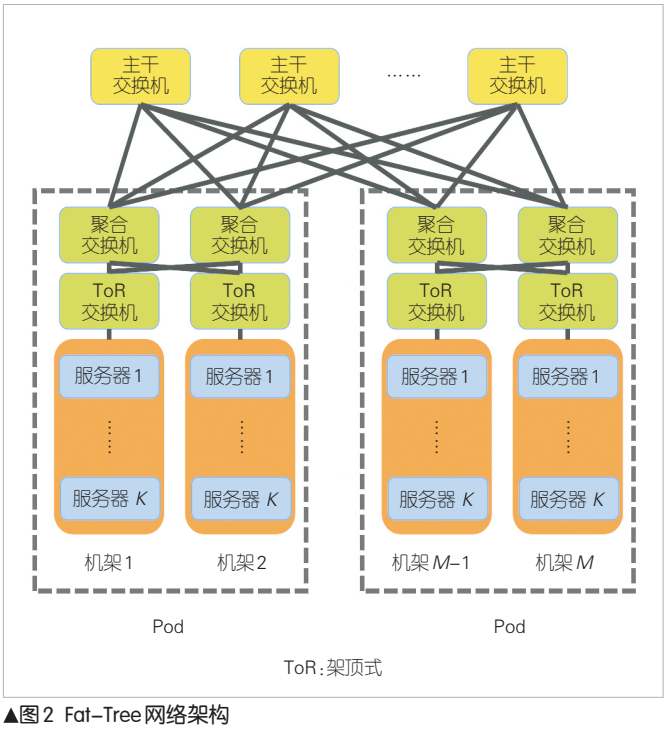
生成式大模型的发展确立了以GPU为中心的集群主导地位。现阶段，GPU间通信采用层次化网络承载：

1）机内网络：利用PCIe总线、NVLink、NVSwitch等技术，实现单台服务器内等多个GPU高带宽短距离互联，为每个GPU提供太比特级的无阻塞any-to-any带宽输入/输出，以便将短程通信流量驻留在高带宽域内。

2）机间网络：利用网卡+交换机模式，实现多个高带宽域互连。服务器间使用远程直接内存访问（RDMA）将数据（中间结果、梯度等）从一个GPU内存传输到另一个GPU内存中（在不同服务器上）。

在生成式大模型训练中，模型参数和数据集分布在集群中的不同GPU上，开展并行训练。训练各阶段的工作负载特征（参数大小、数据集大小和模型架构）不同，流量模型也差异很大。为此，模型设计在优化网络拓扑和提高GPU效率方面发挥着至关重要的作用。在实际网络部署中，需要根据各训练阶段的工作负载模型和流量特征，有针对性地开展网络拓扑设计优化和硬件设备（如交换机）定制。如Google使用了3D环面和光学主干交换机，Meta使用的具有超额订阅主干链路的轨道优化叶交换机。一些高性能计算（HPC）结构还使用蜻蜓拓扑来优化GPU之间的跳数。

目前，大模型集群多部署于同一个地域机房内。随着大模型训练的模型参数规模、数据规模和算力规模的快速发展，单个数据中心机房的硬件设施如电力、液冷、空调等硬件基础设施能力将趋于极限。大模型集群数据中心的超长距



▲图2 Fat-Tree网络架构

广域互联场景需求将逐步增加。但是，与数据中心内部大模型流量相比，广域网承载了多种不同类型的业务，流量特征复杂。虽然流控和拥塞控制等机制使得RDMA在数据中心内部实现了落地部署，但在复杂组网的广域环境下，RDMA远距离直连传输技术并不成熟，在现网中难于规模部署。运维人员需要根据不同的网络环境和流量模型进行RDMA参数设计和调优，这将会面临运维利用率、拥塞、时延等一系列挑战[6]。

相比之下，面向广域RDMA的确定性网络技术（Det-Net）较为成熟，成为近期研究热点。随着灵活以太网（FlexE）、切片分组网（SPN）、时间敏感网络（TSN）、优先级调度队列增强机制、网络演算等各类确定性技术的不断涌现，后续可通过延续优先级流控（PFC）信号、长距离拥塞控制、网络负载均衡等技术实现RDMA的长距离扩展。

2.2 数据传输技术

在生成式大模型训练中，服务器之间需要频繁地进行大量数据的传输和交换。传统的传输控制协议/互联网协议（TCP/IP）在数据传输的过程中需要在用户空间与内核空间之间多次拷贝，降低了数据传输效率。相比之下，RDMA允许应用程序直接访问远程节点的内存，不经过内核，具有高吞吐、低延迟、无CPU占用等优点，可提升模型训练效率，更为适合生成式大模型训练。RDMA从1999年诞生以来，经过20年的发展，技术逐步从高大上的HPC领域走向广阔的通用数据中心领域，广泛应用于大模型训练、高性能计算等场景。

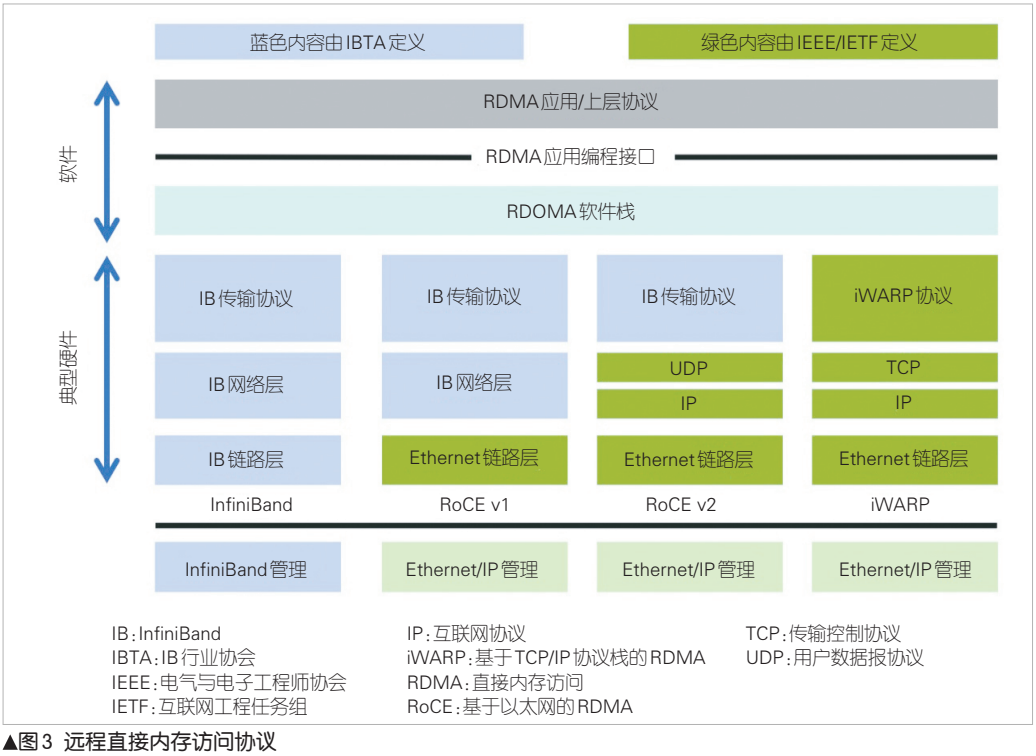
RDMA主要包括3种类型协议：InfiniBand（简称IB）、基于以太网的RDMA（RoCE）以及基于TCP/IP协议栈的RDMA（iWARP）。3种协议都符合RDMA标准，使用相同的上层接口，具体如图3所示。

IB从链路层到传输层定义了一套全新的层次架构，是为高性能计算设计的专用技术。IB在部署时需要专用设备，如IB专用交换机、IB专用网卡、IB专用线缆等，无法与现有的以太网设备兼容。相比于传统以太网，IB具备高带宽、低延迟的数据传输能力以及无损网络的特征，可满足大型数据中心和超级计算中心对高性能网络的需求。但是，IB体系独立封闭，采购维护成本高昂，现阶段主要被用于高性能计算领域，如超级计算机、数据中心和科学研究机构等。

现网中部署着大量基于以太网的产品。为了扩大RDMA的应用范围，IB行业协会（IBTA）组织定义了基于以太网（Ethernet）的RoCE技术标准，允许在不依赖IB专用硬件的情况下使用RDMA。

RoCE通过扩展以太网协议栈，使标准以太网设备支持RDMA操作，实现了高性能远程内存访问与以太网易用性和广泛部署特点的结合。现阶段RoCE有两个主要版本：RoCEv1和RoCEv2。RoCE v1发布于2010年，是基于以太网链路层实现的RDMA协议，但由于它不支持路由，也没有拥塞控制机制，难以在数据中心规模使用。RoCEv2版本是对RoCEv1版本的重大改进，它基于以太网的用户数据报协议（UDP）。RoCEv2支持路由，并且定义了基于显式拥塞通知（ECN）/拥塞通知报文（CNP）的拥塞控制机制。相同场景下，RoCE虽然较IB性能有所降低，但是因其性价比更高，目前已经在一些超大规模数据中心商用部署。

iWARP是国际互联网工程任务组（IETF）提出的基于TCP的RDMA协议。由于TCP是面向连接的可靠协议，这使



▲图3 远程直接内存访问协议

得iWARP在面对有损网络场景时相比于RoCEv2和IB具有更好的可靠性。但是大量的TCP连接会耗费很多的内存资源，另外TCP复杂的流控等机制会导致性能问题，限制了其应用范围，现阶段并未大规模使用。

综上所述，IB在高性能计算领域表现出色，可提供卓越的性能、低延迟和可扩展性，目前在高性能计算领域占据较大优势。相比之下，RoCE则更容易集成到现有以太网基础设施中，并具有较低的成本，是现阶段大模型训练网络的主流方案。

生成式大模型的迅速发展对底层承载网络的性能要求越来越高。业界仍持续开展传输协议的创新。一些云计算和互联网巨头推出新的自研协议。例如，亚马逊提出的可扩展的可靠数据报（SRD）[7]。SRD设计了多路径负载均衡机制，利用尽可能多的不拥塞的网络路径喷洒数据包，在上层消息传递层实现对可靠但乱序的交付数据包进行顺序恢复。通过多路径发送和数据包重排序，提升了网络吞吐能力，降低了传输延迟。

2023年，由众多云计算和网络科技巨头组成超以太联盟，针对IB的封闭生态和原有RoCE的不足，提出了下一代人工智能（AI）和HPC网络的协议：超级以太网传输（UET）[8]。基于IP和以太网进行设计，在基于多路径和数据包喷洒负载均衡、Incast管理机制、高效的速率控制算法、允许乱序数据包传递的应用程序编程接口（API）等方向进行了创新，以减少针对特定网络和负载对拥塞算法的复杂参数调优，支持百万节点的大规模网络扩展。

在2023开源计算（OCP）全球峰会上，谷歌还提出基于新硬件的传输协议Falcon[9]，集成了谷歌多年在网络传输方面的一系列创新技术，包括拥塞控制Swift和保护性负载均衡（PLB）等，推动以太网现代化，以满足下一代大规模AI集群网络的高可靠、高性能、低时延需求。

2.3 拥塞控制技术

RDMA设计目标是高性能和低延迟，它对于底层网络的稳定性和可靠性有极高的要求。网络的可用性决定了整个集群的计算稳定性。RDMA在无损网络状态下可以满速率传输。但一旦发生丢包，将启动“go-back-N”重传机制，放弃已到达的多个包，重新传输N个包，性能急剧下降。然而，现网拓扑复杂度高，流量流向不可预测。因此流控、拥塞控制、负载均衡机制对于RDMA现网落地商用非常重要。

传统的RDMA网络采用基于优先级的流量控制（PFC）流控实现无损以太网[10]。PFC允许交换机在传输数据帧时，对不同数据流设置不同的优先级。一旦交换机的队列超过设定门限，通过逐跳的流量反压，限制发送方的流量速率保障网络无丢包。PFC流控可以在理论上保证不丢包，但PFC以端口级别运行，是一种粗粒度控制机制。规模部署时存在头部阻塞、受害者流、PFC风暴和PFC死锁等问题。

流级别的拥塞控制算法可以缓解PFC缺陷。网络路径上的交换机对流量拥塞情况进行标记。携带拥塞标记信号的报文到达接收者后，再被传回发送者由其根据网络拥塞情况进行调速。在充分利用带宽的前提下，降低网络拥塞程度可以避免频繁触发PFC。现阶段存在多种拥塞控制算法，已大规模部署的有：数据中心量化拥塞通知（DCQCN）[11]、基于延迟的拥塞控制（TIMELY[12]、Swift[13]）和高精度拥塞控制（HPCC）[14]。它们主要区别在于采用的拥塞反馈信号和发送端速率调整方式不同。

DCQCN采用IP报文头中的显示拥塞指示算法（ECN）作为拥塞标记。发送端根据ECN标记情况来推测网络拥塞情况，对源速率进行调整。但ECN标记只携带了有限的信息，拥塞控制调节的颗粒度较粗。目前RDMA网卡商业上直接可用，应用最为广泛。

TIMELY将数据包的往返时延（RTT）作为反馈信号来调整发送端的速率。发送者在主机网卡上对端到端RTT进行测量，基于其变化进行梯度计算，再根据梯度实现基于速率的调速方法。Swift在TIMELY基础上进行改进，将时延进一步区分为网络拥塞和主机拥塞造成的时延，并维持两个拥塞窗口进行调速。目前主要是在Google数据中心使用，依赖于Google的自研网卡。

HPCC利用带内网络遥测收集更详细的链路和端口负载信息，包括时间戳、队列长度、已传输字节数和链路带宽容量等，并以此调整发送端的发送窗口，实现高精度拥塞控制。

面向超大规模的大模型集群网络，上述拥塞控制和流控技术的性能仍然需要提升。现阶段的研发主要集中在两个方向：一方面研究更合适的效能函数来准确评价网络环境的拥塞状态，更准确地探测可用带宽和时延等参数，使发送端获得更准确的数据，从而提高决策的精确性；另一方面，可以进一步细化拥塞窗口的调节方案和不同类型流量调度机制，在兼顾网络的稳定和带宽利用率的同时，保证数据传输质量。

此外，在研的新一代的RDMA网卡将采用更为高效的丢包恢复机制和更好的端到端流控来约束in-flight数据包，不依赖于PFC的RoCEv2网络成为了未来的发展方向。这将有助于把RDMA推广到规模更大、跳数更多的网络中。

2.4 负载均衡技术

RDMA的大规模组网通常采用基于Fat-Tree的Clos架构。基于Fat-Tree的Clos架构的基本理念是使用大量的商用交换机，在服务器之间构造出多个等价路径，交换机对流进行ECMP实现负载均衡，进而形成大规模的无阻塞网络。

与传统数据中心的流量分布不同，大模型训练网络中多为大象流，数量相对较少。这使得传统的ECMP存在哈希极化现象，即多个流可能分配到同个链路上，负载不均造成流冲突。同时，由于ECMP还是一个无状态的局部决策，不关心不同流的大小差异，在流数目不多且大小流长尾严重时，容易造成多条路径中某些路径拥塞而另一些路径空闲，从而造成带宽浪费和影响传输效率。因此，负载均衡是大规模AI集群网络面临的又一挑战。针对ECMP存在问题，有以下两种负载均衡优化方案：

第1种方案是改变流的属性，把流分散到多个等价路径上。在交换机ECMP不变的情况下，改变流的标签，或增加流的熵（将流分割成更小的流或基于报文头的其他位置字段做哈希），让流变得更多从而能通过哈希散开。代表性方法有PLB[15]，在主机端利用拥塞探测感知拥塞的流，对拥塞流的流标签进行更改，引导交换机对流进行重新等价多路径ECMP/加权多路径（WCMP）哈希，从而为拥塞流选择新路径。

第2种方案是基于网络状态的流量调优。通过实时收集网络拓扑和流量等信息，由集中软件定义网络（SDN）控制器为每条流计算出最优路径，或由交换机进行自适应路由选择，包括集中式流量工程[16]、自适应路由技术[17]和网络级负载均衡技术[18]等。

1）集中式流量工程[16]：SDN控制器实时收集网络拓扑和任务放置信息，基于约束最短路径算法为各个流计算最优路径。

2）网络级负载均衡技术[17]：根据大模型训练的流量特征，综合网络拓扑等整网信息，计算出最优的流量转发路径。

3）自适应路由技术[18]：交换机根据出口队列负载评估拥塞情况，为每个数据包选择最不拥塞的端口进行数据传输，以实现负载均衡。由于同一流的不同数据包可能由不同网络路径传输，到达目的节点时可能出现乱序，因此需要网卡侧在RoCE传输层完成对无序数据的转换，再将有序数据传递给应用程序。

此外，目前一些研究[19-20]也指出：原生RoCE协议中规定数据包顺序到达的设计弊端是制约负载均衡的关键因素。未来应从根本上改进传输协议，采用数据包喷洒等技术，使得数据包可以通过多路径顺序传输，然后再利用可编程交换机或智能网卡重新对数据包排序，以充分利用网络的多个可用路径实现负载均衡。

2.5 自动化运维技术

现阶段针对大模型训练的AI集群网络已达万卡规模[21]，未来要扩展到十万和百万卡量级，传统依靠人工的网络运维已经无法满足需求。同时，由于故障无法避免，需要有自动化机制来实现部署和测试，并持续监控网络状态，减少故障和排除故障，实现自动化运维。自动化网络运维技术研究主要集中以下几个方向：

1）端网一体自动化部署和测试：集成多种自动化工具，研发自动选择配置模板，实现全网流控、拥塞控制、负载均衡等关键指标参数的自动化配置和测试验证，缩短大模型训练系统的整体部署时间。

2）信息采集压缩和关键信息提取：传统运维采用简单网络管理协议（SNMP）“拉模式”流量采样方式，采样精度在分钟级别，颗粒度较粗。为实现无损网络，RDMA网络需要全栈巡检和毫秒级的实时监控。除流量信息外，还要采集拓扑、网卡、交换机端口、流控参数等更细颗粒度的信息。Telemetry采用“推模式”流量采集方式，虽然可以实现精细化采集，但在网络中全面开启开销巨大，严重影响网络性能。为此，需要根据网络架构和训练模型，定制化采样精度和信息采集策略，并动态按需调整。为实现高效分析，还需对低频变化信息进行高压缩，提取关键信息，结合带内网络遥测探测流交叉覆盖，实现轻量级近似全链路监控。

3）快速故障定位及自愈：基于多维度的网络指标信息进行故障根因分析，快速定位故障。同时配置故障自愈策略，故障时通过路径切换或节点替换，尽可能地缩短故障恢复时间。例如，利用硬件实现故障感知，无须通过控制面，仅通过数据面的故障通告和故障切换策略；无须等待全网收敛，仅基于端侧智能网卡修改数据包头特定字段的重路由策略等。

3 未来技术展望

大模型的持续发展对算力需求增长迅速，因此高性能高可用的云网基础设施非常必要。打造满足下一代大模型训练需求的低时延、大带宽和高吞吐网络，需要构建新型网络架构，以匹配大模型训练的“大流”“少流”的流量特征；需要研究确定性网络等新技术，满足广域网远距离大模型中心互联需求；需要设计针对大模型通信流量的新型网络协议、拥塞控制和负载均衡机制，实现网络无拥塞和丢包，有效利

用计算资源；需要实现自动化和精细化的运维保障，保障网络持续高效运行。同时，这些技术的系统创新如何实现现网设备和生态结合、如何在工程领域商用落地部署将是一个长期而艰巨的任务，需要产业链各方共同探索解决。

参考文献

[1] 华为. 星河AI网络白皮书 [R]. 2024
[2] 中国移动研究院. 面向AI大模型的智算中心网络演进白皮书 [R]. 2023
[3] AL-FARES M, LOUKISSAS A, VAHDAT A. A scalable, commodity data center network architecture [J]. ACM SIGCOMM computer communication review, 2008, 38(4): 63-74. DOI: 10.1145/1402946.1402967
[4] WANG W, GHOBADI M, SHAKERI K, et al. Optimized network architectures for large language model training with billions of parameters [EB/OL]. [2024-02-25]. https://arxiv. org/pdf/2307.12169v2.pdf
[5] 蒋炜, 钱声攀, 邱奔. 数据中心网络拓扑结构设计策略研究 [J]. 中国电信业, 2021, (S1): 73-78
[6] 赵俊峰, 李芳, 叶晓峰, 等. 面向广域RDMA的确定性网络需求与技术 [J]. 电信科学, 2023, 39(11): 39-51. DOI: 10.11959/j.issn.1000-0801.2023248
[7] SHALEV L, AYOUB H, BSHARA N, et al. A cloud-optimized transport protocol for elastic and scalable HPC [J]. IEEE micro, 2020, 40(6): 67-73. DOI: 10.1109/mm.2020.3016891
[8] Overview and Motivation for the forthcoming utra ethernet consortium specification [EB/OL]. (2023-09-17) [2024-02-24]. https://ultraethernet. org/wp-content/uploads/sites/20/2023/10/23.07.12-UEC-1.0-Overview-FINAL-WITH-LOGO.pdf
[9] Google opens Falcon, a reliable low-latency hardware transport, to the ecosystem [EB/OL]. [2024-02-25]. https://cloud. google.com/blog/topics/systems/introducing-falcon-a-reliable-low-latency-hardware-transport
[10] IEEE DCB. 802.1Qbb – priority-based flow control [EB/OL]. [2024-02-25]. http://www.ieee802.org/1/pages/802.1bb.html
[11] ZHU Y B, ERAN H, FIRESTONE D, et al. Congestion control for large-scale RDMA deployments [J]. ACM SIGCOMM computer communication review, 2015, 45(4): 523-536. DOI: 10.1145/2829988.2787484
[12] MITTAL R, LAM V T, DUKKIPATI N, et al. TIMELY [J]. ACM SIGCOMM computer communication review, 2015, 45(4): 537-550. DOI: 10.1145/2829988.2787510
[13] KUMAR G, DUKKIPATI N, JANG K, et al. Swift: delay is simple and effective for congestion control in the datacenter [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. ACM, 2020: 514-528. DOI: 10.1145/3387514.3406591
[14] LI Y L, MIAO R, LIU H H, et al. HPCC: high precision congestion control [C]//Proceedings of the ACM Special Interest Group on Data Communication. ACM, 2019: 44-58. DOI: 10.1145/3341302.3342085
[15] QURESHI M A, CHENG Y, YIN Q W, et al. PLB: congestion signals are simple and effective for network load balancing [C]// Proceedings of the ACM SIGCOMM 2022 Conference. ACM, 2022: 207 - 218. DOI: 10.1145/3544216.3544226
[16] Meta networking@scale 2023 [EB/OL]. (2023-09-07)[2024-02-26]. https://atscaleconference. com/events/networking-scale-2023/
[17] 华为. HPC无损以太和AI Fabric网络技术白皮书 [R]. 2023
[18] NVIDIA Spectrum-X network platform architecture [EB/OL]. [2024-02-26]. https://nvdam. widen. net/s/h6klwtqv5z/nvidia-spectrum-x-whitepaper-2959968
[19] SONG C H, KHOOI X Z, JOSHI R, et al. Network load balancing with In-network reordering support for RDMA [C]//Proceedings of the ACM SIGCOMM 2023 Conference. ACM, 2023: 816 - 831. DOI: 10.1145/3603269.3604849
[20] Cisco. Cisco silicon one [EB/OL]. [2024-02-26]. https://blogs.cisco.com/sp/building-ai-ml-networks-with-cisco-silicon-one
[21] JIANG Z H, LIN H B, ZHONG Y M, et al. MegaScale: scaling large language model training to more than 10, 000 GPUs [EB/OL]. (2024-02-23) [2024-02-25]. https://arxiv. org/abs/2402.15627

作 者 简 介

唐宏，中国电信股份有限公司研究院IP领域首席专家，正高级工程师，中国电信科技委常委；长期从事IP网络及其新技术的研发工作；发表论文30余篇，获发明专利100余项。

武娟，中国电信股份有限公司研究院正高级工程师；主要从事IP网络、人工智能相关研究工作；已发表多篇论文。

徐晓青，中国电信股份有限公司研究院工程师；主要从事网络设计、网络优化和人工智能相关工作；已发表多篇论文。

张宁，中国电信股份有限公司研究院算法工程师；研究方向包括联邦学习、激励机制、负载均衡等；已发表TMC论文2篇。

大语言模型时代的智能运维

Artificial Intelligence for IT Operations in Era of Large Language Model

裴丹/PEI Dan[1]，张圣林/ZHANG Shenglin[2]，
孙永谦/SUN Yongqian[2]，裴昶华/PEI Changhua[3]
（1. 清华大学，中国 北京 100084；
2. 南开大学，中国 天津 300457；
3. 中国科学院计算机网络信息中心，中国 北京 100190）
(1. Tsinghua University, Beijing 100084, China;
2. Nankai University, Tianjin 300457, China;
3. Computer Network Information Center, Chinese Academy of Sciences, Beijing 100190, China)

DOI：10.12142/ZTETJ.202402009
网络出版地址：http://kns.cnki.net/kcms/detail/34.1228.tn.20240407.1926.002.html
网络出版日期：2024-04-09
收稿日期：2024-03-10

摘要：大语言模型由于其强大的语言能力、代码生成能力、工具编排能力，将是智能运维（AIOps）落地取得突破的重要因素。大模型时代的AIOps架构是多AIOps智能体的人机协同系统。首先列举了AIOps对大语言模型的应用需求，探讨了大语言模型时代的AIOps架构，其次总结了将大语言模型整合到运维工作流程中所面临的挑战，最后结合这些挑战给出了解决思路并倡议以“社区众包，群体智慧”的方式加速落地运维大语言模型。

关键词：大语言模型；智能运维；人机协同；智能体

Abstract: Due to its powerful linguistic capabilities, code generation abilities, and tool orchestration capabilities, the large language model will be an important factor in the breakthrough of artificial intelligence for IT operations (AIOps). The architecture of AIOps in the era of large models is a human-machine collaborative system composed of multiple AIOps intelligent agents. Firstly, the application requirements of AIOps for large language models are listed and the architecture of AIOps in the era of large language models is explored. Secondly, the challenges of integrating large language models into operational workflows are summarized, and solutions to these challenges are proposed. Finally, it advocates for the acceleration of large language model implementation in operations through the "community crowdsourcing and collective intelligence" approach.

Keywords: large language model; AIOps; human-machine synergy; Agent

引用格式：裴丹, 张圣林, 孙永谦, 等. 大语言模型时代的智能运维 [J]. 中兴通讯技术, 2024, 30(2): 56-62. DOI: 10.12142/ZTETJ.202402009
Citation: PEI D, ZHANG S L, SUN Y Q, et al. Artificial intelligence for IT operations in era of large language model [J]. ZTE technology journal, 2024, 30(2): 56-62. DOI: 10.12142/ZTETJ.202402009

近年来，大语言模型（LLM）的出现对自然语言处理、机器学习和人工智能等众多领域产生了革命性的影响。诸如生成式预训练Transformer模型（GPT）系列[1]，在理解和生成自然语言以及执行复杂的文本处理任务上表现出了前所未有的卓越能力。因此，LLM在各行业中得到了广泛应用，并已逐步渗透到智能运维（AIOps）这一前沿领域。

本文系统地研究了AIOps领域对LLM的具体应用需求，深入剖析了大语言模型时代下的AIOps体系架构的发展趋势。同时，本文中，针对将LLM有效整合至运维工作流程所面临的挑战，我们进行了深度探讨，着重强调了群体智慧协同创新对于促进专用于运维场景的大语言模型（OpsLLM）技术研发与快速迭代的重要性。

1 智能运维领域对大语言模型的需求

1.1 AIOps工具更为人性化的交互方式

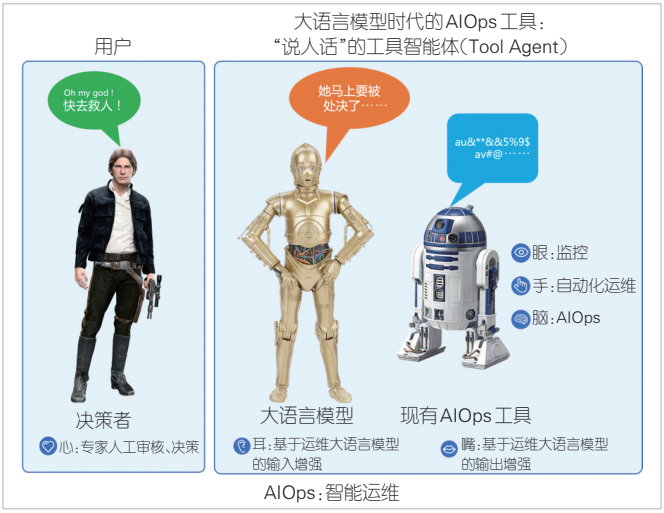
运维环境的复杂性和数据规模化特性，在人工运维阶段给用户和决策者带来的挑战逐渐加剧[2]。运维环境中通常包含多种模态的数据，这会进一步增加分析处理的难度。随着AIOps工具的出现，运维系统逐步具备了数据采集监控（相当于眼睛）、自动化运维（如同手）和智能运维（相当于大脑）的功能。然而，尽管这些工具的功能日益强大，但它们的使用却相对繁琐，通常需要通过特定的界面进行交互，这增加了决策者理解其输出的难度。在LLM时代，已有的运维工具可以通过自然语言与人进行交流，从而使决策者能够

更加直观地理解和应用这些工具的输出信息。

以图1中《星球大战》这一电影为例，LLM在决策者与AIOps工具之间充当翻译者的角色，通过几轮交流，决策者能够做出更加明智和准确的决策。LLM的引入首先将为AIOps工具赋予沟通的能力，使其能够更加高效地与决策者交流，从而实现人性化交互的目标。

AIOps小模型工具经赋能后被称为工具智能体（Tool Agent），具备响应自然语言指令和要求进行工作的能力。工具智能体被定义为现有工具经LLM赋能后的智能体，其功能边界清晰，可接受应用程序编程接口（API）调用或自然语言指令。但仍需明确的是，工具智能体的推理和规划能力（如有）源自工具内置的AIOps与岗位型智能体。这类的Agent本质是在现有的AIOps小模型的基础上进行封装，以供大模型调用。如可以将现有的时序异常检测的算法进行封装，那么大模型通过接口请求该异常检测算法，并将需要查询的时间以及实体通过参数传递给异常检测算法。该算法拉取对应实体的时序数据并进行异常检测，将最后的结果返回给大模型，完成一次异常检测工具智能体的调用。

另一种智能体被称为岗位型智能体（Job Agent）。它充分利用知识、经验、规则、算法，具备了类似运维人员的观察、推理、规划、决策能力，可与运维人员、其他岗位智能体以自然语言交互，与工具型智能体以自然语言或API交互。工具型智能体是现有流程驱动、数据驱动的工具经LLM赋能后的升级，而岗位智能体则以人为本，是模拟一线运维、应用运维、网络运维、存储运维等岗位工程师的智能体。这类的Agent是在上述的工具智能体的基础上，基于现有的LangChain框架，通过“思维链”（CoT）或简单的、规则的配置，将一系列的工具智能体聚合到一起，从而完成一个具体的任务。如在构建一个有线网络故障排障的岗位型智能体时，用户只需要询问该岗位型智能体某个有线网络在某个时间段内是否正常，该智能体便可以通过历史积累的排障手册的规则或者大模型自身的思考能力，决定先去调用数据拉取的工具智能体，然后调用异常检测的工具智能体，并通过对所有的时序曲线或者日志进行异常检测，对所有的异常检测结果输入到根因定位工具智能体中，最后将根因定位的结果返回给大模型。大模型决策时要启动故障消除工具智能体。这些工具智能体以大模型的编排能力为纽带，共同完成了某个特定的岗位或任务。



▲图1 决策者、大语言模型与智能运维工具之间的联系

1.2 智能运维领域对大语言模型的应用需求

有了LLM，AIOps领域的应用将在不同发展阶段扮演不同角色。在近中期应用阶段，LLM可能被定位为助理、教练、顾问和参谋，其主要任务是提供各种建议和指导，而不直接进行决策和处置。这种角色定位能够充分利用LLM的知识和智能，为运维人员提供必要的支持和帮助，同时避免了直接决策和处置可能带来的风险。而在中长期的应用阶段，随着LLM的不断优化和经验积累，其角色可能逐渐演变为内部专家。在这个阶段，LLM具备更多的决策和处置能力，可以参与到实际的运维工作中，对问题进行分析，并提出解决方案，因此在一定程度上还担任着决策的角色。总的来说，AIOps领域对LLM的需求随着发展阶段的不同而有所变化，需要根据错误容忍度和技术挑战的解决难度来合理应用LLM，从而实现AIOps系统的持续改进和优化。下面我们列出了不同阶段AIOps领域对LLM的应用需求。

1）知识检索：在企业环境中蕴藏着大量的结构化知识，这些知识对于运维和故障排除至关重要。为了更有效地利用这些存量知识，LLM时代的AIOps需要通过自然语言的方式进行快速多轮问答，以便在需求出现时迅速获取清晰的排障路径。为了实现这一目标，首要条件是拥有至少60 min的OpsLLM，并且支持检索增强（RAG）技术。这样的模型能够利用其强大的自然语言处理能力，根据问题的特征和上下文信息，快速准确地检索到相关的知识，并以问答形式提供解决方案，从而提高了问题解决的效率和准确性。这种基于OpsLLM和RAG技术的自然语言问答系统，为企业运维团队提供了强大的工具，使他们能够更加高效地应对各种运维挑战，并迅速解决故障。

2）多文档问答：目前许多企业拥有庞大的存量文档资源，包括但不限于运维文档、应急手册、产品手册和排障手册等。这些文档通常以PDF、Word等格式保存，代表着丰富的知识库。以售后技术支持文档为例，以往需要手动存

储、查询和整理常见问题解答（FAQ）。而现在企业可以将成百上千的技术文档上传至私有部署的OpsLLM，再利用OpsLLM的能力，根据文档中的内容和知识，实现快速精准的问答交互。

3）数据注释：过去，监控数据常常以单个字段的形式呈现，其复杂性和抽象性使其难以被人类用户直观理解，进而增加了运维工作的复杂度。随着OpsLLM的引入以及对知识库的调用，情况已经发生了改变。通过将OpsLLM与知识库相结合，监控数据得以转化为自然语言形式，这使得原本冰冷的字段和多模态运维数据能够以更加易于理解的方式呈现，这便形成了所谓的注释型岗位智能体。

4）数据理解：在当今的运维环境中，各种工具智能体如日志工具、告警工具、安全日志工具和指标工具等发挥着至关重要的作用。它们的功能不仅仅是收集和存储运维数据，更重要的是能够对这些数据进行快速、准确的总结和分析。例如，当面对大量日志数据时，工具智能体能够在短时间内对这些数据进行归纳总结，提炼出关键事件，辅助运维人员迅速了解系统的运行状态和可能存在的问题。尽管工具智能体在这方面已经取得了一定的进展，但在运维领域的能力仍然存在不足和改进的空间。

5）脚本解读：在企业运维环境中，存在着大量的存量脚本、结构化查询语言（SQL）查询语句、日志查询以及各种脚本配置等，这些都具有其物理意义和实际运维价值。然而，这些存量查询的脚本往往是以编程语言或者查询语言的形式存在的，对于非技术人员或者新员工而言，理解和应用这些脚本可能存在一定的难度。因此，能否将这些已有的存量查询的脚本翻译成自然语言，并将其中的隐性知识转化为显性知识，成为一个备受关注的问题。这样的转化将极大地提升老员工培训新员工的效率，使得新员工能够更快地掌握运维工作所需的技能和知识。

6）从自然语言到查询（NL2Query）：NL2Query旨在为单个存量工具提供自然语言交互增强的功能，使其具备意图识别、总结等能力，从而进化为工具智能体。这其中涵盖了大量存量AIOps小模型工具、可观测性工具，图数据库和SQL查询等多种形式的查询以及一些代码生成等功能。要实现NL2Query的成功应用，前提条件是需要一个具备良好性能的OpsLLM，以确保对自然语言的准确理解和响应。其次，需要对数据进行标准化处理，以确保查询的一致性和可靠性。此外，还需要标准化工具接口，以便与其他工具和系统进行无缝对接。总体而言，NL2Query是多Agent、人机交互框架中的一个重要组成部分。通过NL2Query技术的应用，不同智能体之间可以使用自然语言进行交流。这实现了接口的统一化，促进了存量工具与总线之间的无缝连接。

7）基于不同运维数据模态的基础模型的工具智能体：基于不同模态运维数据的基础模型开发工具智能体是当前运维领域的一个重要研究方向，涵盖了对指标、日志、调用链、告警数据等多种数据类型的智能分析和处理。针对指标异常检测的落地问题，通常存在着一个落地困境，即虽然算法本身是通用的，但需要针对每个企业的具体指标训练一个定制化模型。这增加了模型训练和部署的复杂性和成本。为了解决这一问题，可以利用Transformer、Diffusion等技术建立一个LLM，以实现零样本学习。

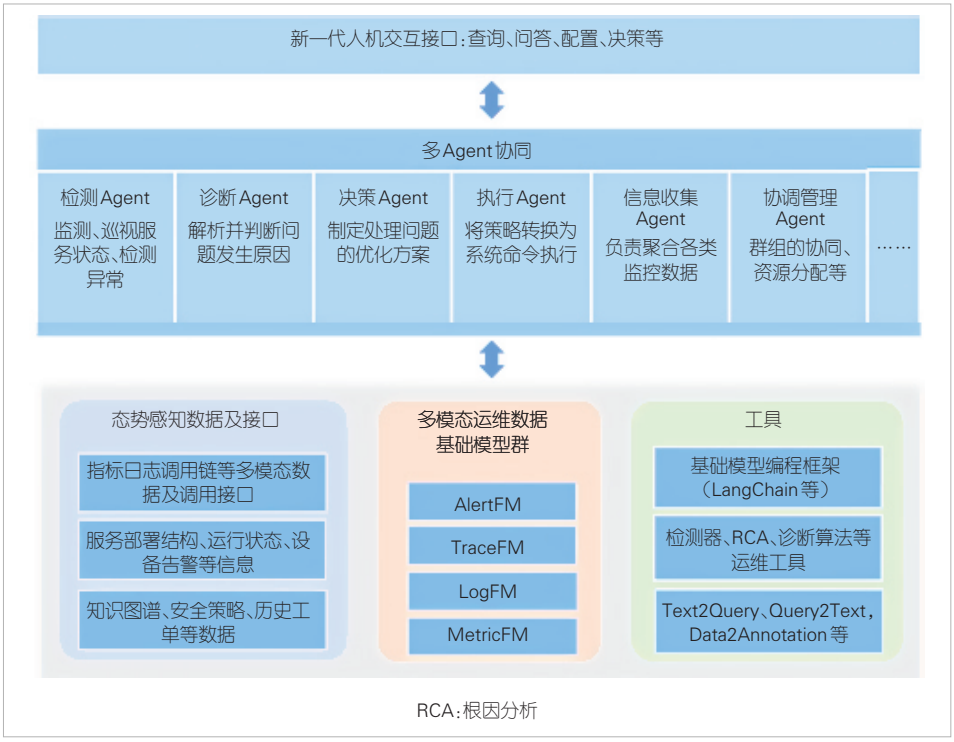
8）多AIOps智能体人机协同，完成复杂运维任务：在运维作战指挥室这一模拟场景中，运维人员聚集在一起，展开各种不同角色或操作工具的讨论与合作，并且每个岗位都配备了专属的LLM数字孪生助手，能够显著提升其工作效率。同时，各种监控工具和AIOps小模型工具也配备了相应的工具型智能体，从而实现了人、岗位智能体、工具智能体之间的自然语言对话和协作。这使得运维应急处置更加高效和智能化。在多AIOps智能体人机协同应用的起步阶段，我们可以将其视为现有的ChatOps运维即时通信聊天室。其中，人的参与占比较高。随着智能体能力的不断提升，其负责的任务占比将逐渐增加，而人的直接参与程度则会逐渐降低，且更加聚焦于最为关键的决策任务。

2 大语言模型时代的AIOps架构

新的AIOps架构将充分利用大语言模型所具备的强大语言理解交互能力和深度的知识学习运用能力，极大地丰富和拓展传统智能运维的功能边界及智能化水平。

如图2所示，新的AIOps架构以多个智能体为主体，形成了一个人机协同的“运维团队”，各智能体犹如具备特定职责的“数字运维专员”，各自承担着信息收集、异常检测、故障诊断、反感决策与执行等核心角色。人与智能体、智能体与智能体之间通过自然语言进行沟通交流。每个智能体的能力通过通识大语言模型结合领域知识、任务要求进一步学习训练获得。如同真实运维人员一样，智能体在执行运维任务时需整合多元信息来源，包括但不限于环境态势感知、基础数据模型以及各类运维工具，并确保在相应权限范围内合理运作。

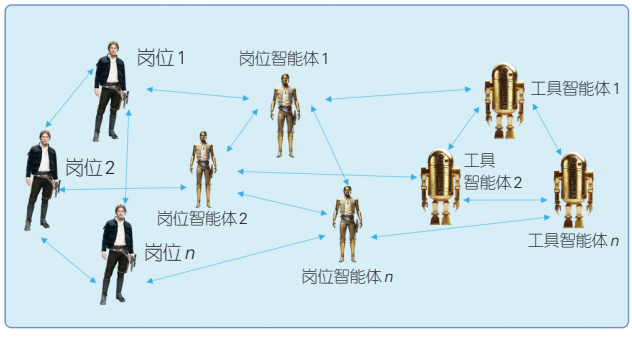
在新型智能运维架构下，人机交互体验将得到前所未有的提升，运维人员所需的技术门槛和工作负担显著减轻，这将有力加快AIOps迈向更高程度无人值守运维的步伐，开启运维自动化的新篇章。



▲图2 多智能体协同的智能运维(AIOps)架构

2.1 多AIOps智能体的人机协同系统

LLM的应用将成为AIOps领域实现突破的关键因素之一。在LLM时代的AIOps架构中，如图3所示，多个智能体形成了一个人机协同系统，其中包括工具智能体（信息收集、检测、诊断等Agent）、岗位智能体（知识培训、决策执行、协调管理等Agent）以及真实运维人员等实体，它们之间通过自然语言进行交互。自然语言充当着运维人员、岗位智能体和工具智能体之间的通用接口，而单聊和群聊窗口则扮演着“服务总线”的角色，负责连接、编排和融合各种小模型工具、结构化知识、人类的经验，以实现人机协同完成运维任务的目标。这种整合了自然语言、智能体和运维人员的协同架构将为AIOps的应用带来新的突破和进展，从而可以促进运维工作的智能化和高效化。



▲图3 多智能运维(AIOps)智能体协同的人机系统

2.2 智能体赋能

针对运维领域的特点，AIOps架构需要对公域和私域进行有效分离。在运维领域，共性多于差异化，如一个运维专家在不同公司间转岗时，尽管需要适应新的工作环境，但依靠通用的运维知识仍能快速展开工作。因此，我们应集中力量处理共性问题。利用人工智能社区最新、最强有力的开源LLM底座，结合多模态的运维知识图谱和混合专家（MoE）模型，可以构建运维通用的LLM。这一举措将为AIOps架构提供更强大的语言理解和决策能力，使其能够更好地适应多样化的运维场景。而在私域方面，由于数据获取困难，算力和语料有限，因此需要简化处理。

正如医疗大模型需要针对影像、核磁、电子计算机断层扫描（CT）等不同数据类型建立对应的基础模型一样[3]，AIOps架构也需要建立针对不同类型运维数据的基础模型，构建面向多模态运维数据的基础模型群。这是因为，每种类型的数据特征各异，如果直接使用LLM处理非文本数据往往效果不佳，因此需要针对不同数据类型设计不同的处理方法。

AIOps架构还应融合已有的自动化运维工具，通过基础模型的编程框架（如LangChain[4]等）进行整合。我们需要确保这些工具的接口尽可能标准化，从而能够清晰描述API，使自然语言描述的需求能够直接转换成接口调用（如生成SQL语句、配置、API调用等）。

3 运维大语言模型落地面临的挑战

尽管通识LLM在许多领域已经展现出了强大的能力，但其无法全面准确地掌握运维领域的专业知识。通识LLM具有广泛的应用前景，但在运维领域仍然存在着许多需要克服的难题。我们既不能过于乐观地期待通识LLM能够立即解决运维领域的所有难题，又要在充分了解其局限性的基础上，积极地探索其在运维领域的应用，持续努力地克服各种挑战，实现OpsLLM的落地应用。

LLM在AIOps领域的落地应用存在如下技术挑战：

1）AIOps系统对错误的容忍度低，因为一旦决策错误将带来灾难性后果。但是，通用LLM容易出现幻觉，错误率较高。

2）运维人员往往要求AIOps系统输出的结果具有可解释性，以便于他们判断结果的准确性并采取相应的运维措施。但通用LLM往往是黑盒模型，可解释性欠佳。

3）AIOps领域的严肃语料数量不足且质量欠佳，但训练或微调OpsLLM往往需要大量高质量语料。

4）大部分企业通常不愿意为AIOps耗费太多计算资源，往往要求OpsLLM具有较低的部署开销。但是，通用LLM的部署、微调和应用往往耗费大量的计算资源。

5）当前，通用LLM呈现百花齐放、日新月异的局面，因此如何选择最优的通用LLM，是OpsLLM亟待解决的挑战。

6）通用LLM往往无法直接处理时间序列、知识图谱、拓扑结构等多模态运维数据，但企业往往积累了海量多模态运维数据亟待OpsLLM处理。

7）企业往往已开发了大量AIOps、自动化运维工具，需要与OpsLLM结合起来，发挥它们的价值。

虽然目前将LLM应用到AIOps领域面临着一些挑战，但前述所有技术挑战都有相应的技术思路可供解决，具体而言有以下6点：

1）为了避免幻觉，增强模型的可解释性，可以采用检索增强（RAG）的方式，增加显式知识的占比，包括思维链、思维树、思维图和知识图谱，并通过“有据可依”的生成策略提供原文引用。

2）解决严肃语料不足的问题可以通过由易到难的课程学习方式进行训练，以逐步提高模型对运维领域的理解和适应能力。

3）针对“私有部署训练和部署开销都要低，私域数据的数量、质量不足”的问题，可以进行模型分层，通过在公域进行预训练、微调和提示工程，训练一个针对运维领域的LLM（即L1层LLM）。在私有部署时，可以避免预训练和微调，而是通过检索方式融合本地知识库，以文档和提示作为便捷的知识工程手段，并通过降低模型的精度来降低私有部署的一系列推理开销。

4）在底座选型时，应尽量与开源LLM的底座解耦，以便更灵活地应对不同需求和场景。

5）对于结构化、多模态和实时数据的处理，可以建立专门的多模态基础模型群，并构建相应的工具智能体，以实现对这些数据的有效处理和分析。

6）对于存量的AIOps小模型工具和自动化运维工具，可以利用工具智能体的方式将其融入到多智能体架构中，以实现更高效、更智能的运维流程。

这些措施将有助于克服技术难题，推动LLM在AIOps领域的发展和应用。

4 大模型时代的AIOps落地建议

针对上述具体的挑战，我们都有相应解决方法和应对手段。从方法论角度看，大模型时代技术日新月异，因此我们更需要更加系统地、有规划地设计大模型时代的AIOps落地路径，少走弯路，用有限的资源获得最大化的落地效能。具体而言，本文中我们总结了大模型时代AIOps的3个重要的设计准则或重要方向。

4.1 训练“懂运维语言”的大语言模型OpsLLM

在运维这一严肃领域，迫切需要训练一个“懂运维”的LLM，而非仅仅是一个通用的LLM。这一模型必须具备真正理解输入文档和上下文的能力，而不是仅提供大致的答案。举例而言，如图4所示，开源的LLM可以被视为训练有素的本科生，他们博闻强记，但如果将这些本科生直接投入运维工作岗位，他们可能无法理解其中的内容，甚至不了解相关术语，从而难以胜任工作。因此需要利用大量与运维相关的语料对模型进行训练、微调和优化，以使其能更好地理解运维上下文，此时的模型可以被视为运维专业研究生。只有这样，它才能在实际的运维场景中发挥作用。这一观点与中文

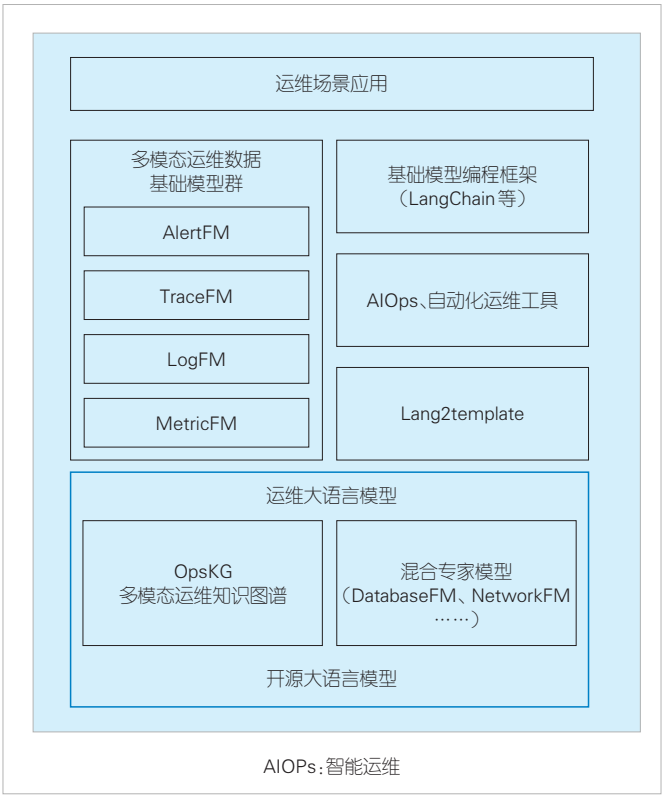
▲图4 运维大语言模型的模型栈

医疗LLM的逻辑类似[5]，强调了为特定领域训练模型的重要性，以确保其在专业场景中的准确性和可靠性。而基于私域运维数据微调，通过检索方式融合本地知识库，以文档和提示作为便捷的知识工程手段的私有部署OpsLLM则可以被视为拥有10年运维工龄的专业运维员工。

OpsLLM是一个综合性的模型，它除了拥有一个基于垂直语料进行预训练、微调或者提示工程的大语言基座模型外，还涵盖了多个关键组成部分[6]。首先，运维大语言模型框架（如图5所示）的中心构成是OpsLLM，即“懂运维”的LLM。OpsLLM的内部结构包含了运维知识图谱、混合专家模型以及开源LLM的底座。在底座部分，尽量采用松耦合的设计，借助流水线工具实现可替换、可迭代、持续演进的特性。此外，针对多模态的运维数据，还需构建多模态基础模型群，例如基于Transformer架构构建的MetricFM、LogFM等基础模型。最后，通过基础模型编程框架，将现有和新的运维工具有机地串联起来，更好地实现运维场景的智能化。

4.2 小步快跑，以用促建

相比于AIOps“全面开花，什么都做”的建设现状，我们更建议AIOps的建设要小步快跑，以用促建，错误容忍度从高到低，循序渐进。将LLM在运维领域的落地应用从岗位助手、岗位培训教练、岗位顾问、岗位参谋逐步转变为内部专家，从提升效率逐渐过渡到参与决策。举例来说，某监控数据采集厂家在应用落地方面采用了一种创新的方法：他们利用ChatGPT搭建了一个“售后工程师GPT助手”。在售后专家工程师与客户进行交流时，他们将客户的问题交给“GPT助手”进行回答，然后再经过售后专家工程师审核修改后传递给客户。这样一来，售后专家工程师的工作效率得到了大幅提升。需要强调的是，这个应用的目的是作为“售后技术支持岗位助手”，旨在帮助售后技术专家提升效率，而不是替代专家进行售后工作。这一实例充分展示了渐进式应用落地策略的可行性和有效性，为AIOps技术的实际应用提供了有益的参考和借鉴。



▲图5 多智能体协同的AIOps架构

4.3 构建开放的运维社区

构建开放社区的方式不仅是LLM快速发展的必经之路，更是推动其不断演进和优化的关键机制。在这种模式下，从各行各业的专家到普通用户，都可以通过参与模型训练、数据标注、模型评估等方式，为LLM的发展贡献自己的智慧和经验。这种集体智慧的汇聚和共享，不仅能够加速模型的学习和优化过程，还能够帮助模型更好地理解各个领域的特定需求和挑战，从而更加精准地应用于实际场景中。

在群体智慧的引导下，LLM在运维领域的应用可以实现以下愿景：

• 运维社区可以积极参与和协作，共同开发和优化针对运维领域的大型语言模型。这些模型将结合运维领域的专业知识和实践经验，具备更高的专业化和适用性，能够更准确地理解和处理运维中的各种复杂情境和问题。

• 通过建立开放的平台和论坛，运维专业人士可以分享自己的经验、技术和最佳实践，共同探讨和应对运维领域的一些挑战和问题。这种知识共享和协作的模式将加速LLM在运维领域的应用和普及，推动运维工作的效率和质量不断提升。

• 借助群体智慧的力量建立丰富多样的运维数据集和场景模拟环境，可以为LLM的训练和优化提供更加充分和真实的数据支持。这将有助于模型更好地理解和模拟运维实践中的各种场景，提高其泛化能力。

• 通过开放的AIOps联盟社区的共同努力，可以不断完善和扩展LLM在运维领域的应用场景和功能，实现从运维监控、故障诊断到自动化运维和智能决策的全面覆盖。这将为运维工作带来革命性的变革，提升运维效率，降低成本，并为企业提供更加可靠和稳定的服务保障。

5 结束语

在LLM时代，AIOps是多AIOps智能体的人机协同系统。这一体系的核心在于运维人员、岗位型智能体和工具型智能体之间的紧密合作和交流。通过人机协同，运维人员可以利用LLM的强大能力，更高效地处理各种复杂的运维任务和问题。与此同时，岗位智能体和工具智能体作为LLM的延伸和应用，为运维人员提供了更多的支持和辅助，进一步提升了整个运维系统的智能化水平。这种多AIOps智能体的人机协同系统不仅能够提高运维效率和质量，还能够适应运维领域日益复杂和多变的需求，为企业的稳定运行和持续发展提供了有力的技术支持。因此，多AIOps智能体的人机协同系统具有重要的理论和实践意义，值得进一步深入研究和应用。

参考文献

[1] RADFORD A, NARASIMHAN K, SALIMANS T, et al. Improving language understanding by generative pre-training [EB/OL]. [2024-03-08]. https://cdn. openai. com/research-covers/language-unsupervised/language_understanding_paper.pdf

[2] 裴丹, 张圣林, 裴昶华. 基于机器学习的智能运维 [J]. 中国计算机学会通讯, 2017, 13(12): 68-72

[3] ZHOU H Y, YU Y Z, WANG C D, et al. A transformer-based representation-learning model with unified processing of multimodal input for clinical diagnostics [J]. Nature biomedical engineering, 2023, (7): 743-755. DOI: 10.1038/s41551-023-01045-x

[4] LangChain [EB/OL].[2024-03-08]. https://www.langchain.com/

[5] ChatMed: a Chinese medical large language model [EB/OL]. [2024-02-04]. https://github.com/michael-wzhu/ChatMed

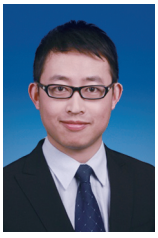
[6] LIU Y H, PEI C H, XU L L, et al. OpsEval: a comprehensive task-oriented AIOps benchmark for large language models [EB/OL]. [2024-03-08]. https://arxiv.org/abs/2310.07637

作 者 简 介

裴丹，清华大学计算机系长聘副教授、博士生导师，ACM/IEEE高级会员；主要研究方向为基于机器学习的智能运维；发表论文200余篇，授权专利30余项。

张圣林，南开大学软件学院副教授、博士生导师；主要研究方向为基于机器学习的智能运维，包括异常检测、故障定位、根因分析等；主持国家自然科学基金项目2项；发表论文40余篇。

孙永谦，南开大学软件学院副教授；主要研究方向为智能运维，包括异常检测、异常定位、告警聚合收敛、故障定位分析等；发表论文30余篇。



裴昶华，中国科学院计算机网络信息中心副研究员；主要研究方向为智能运维、AI for Networking、机器学习；承担国家重点研发计划子课题、国家自然科学基金项目和中国科学院网信专项等项目；发表论文20余篇。

大模型知识管理系统

Large Model Knowledge Management System

周扬/ZHOU Yang，蔡霈涵/CAI Peihan，
董振江/DONG Zhenjiang

（南京邮电大学，中国 南京 210023）
(Nanjing University of Posts and Telecommunications, Nanjing 210023, China)

DOI：10.12142/ZTETJ.202402010
网络出版地址：http://kns.cnki.net/kcms/detail/34.1228.TN.20240422.1818.002.html
网络出版日期：2024-04-23
收稿日期：2024-03-10

摘要：企业知识管理至关重要，而传统企业知识管理系统存在构建成本高、知识利用率低的问题。提出了基于大模型检索增强生成（RAG）技术构建企业知识管理系统的方案。首先介绍了整体方案架构、业务流程与4类知识构建技术，然后重点介绍了检索前处理、知识检索、检索后处理等全流程知识检索技术，并设计了全面的测评框架。经过实践检验，该方案具有知识构建效率高且成本低、意图理解精确、知识检索准确等特点与优势。

关键词：RAG；知识管理系统；大模型；知识工程

Abstract: Enterprise knowledge management is very important, but traditional enterprise knowledge management systems suffer from high construction costs and low knowledge utilization rate. A scheme to build enterprise knowledge management system based on retrieval-augmented generation (RAG) technology is proposed. Firstly, the overall scheme architecture, business processes and four types of knowledge construction technologies are introduced, and then the whole process of knowledge retrieval technology are discussed, such as retrieval pre-processing, knowledge retrieval, retrieval post-processing, and subsequently a comprehensive evaluation framework is designed. The scheme has the characteristics and advantages of high efficiency and low cost of knowledge construction, accurate intention understanding, and accurate knowledge retrieval.

Keywords: RAG; knowledge management system; large model; knowledge engineering

引用格式：周扬，蔡霈涵，董振江. 大模型知识管理系统 [J]. 中兴通讯技术, 2024, 30(2): 63-71. DOI: 10.12142/ZTETJ.202402010
Citation: ZHOU Y, CAI P H, DONG Z J. Large model knowledge management system [J]. ZTE technology journal, 2024, 30(2): 63-71. DOI: 10.12142/ZTETJ.202402010

在当今信息化快速发展的时代，企业面临着前所未有的数据增长和知识爆炸挑战。有效地管理和利用这些知识资源成为企业获得竞争优势、促进创新和提高决策质量的关键因素。这是企业知识管理[1]的重要研究内容，旨在帮助组织系统地收集、整合、共享和分析企业内外的知识和信息，从而最大化知识资产的价值。

传统企业知识管理系统以共享知识库为核心，如ONES Wiki、PingCode Wiki等，旨在搭建共享知识和交流平台。该方案面临多重挑战，主要包括：1）知识库的内容来源于多源异构，大量非结构化知识需要人工处理，转换为关系数据库或知识图谱数据后才能提供服务，但构建效率低、成本高、周期长。2）传统的全文检索技术仅依赖于关键词匹配和倒排索引。随着知识库规模的扩大，用户在庞大的知识库中难以获取所需知识，检索效率低下。3）系统功能单一，用户体验差。系统主要提供固定字段检索或按关键词的全文检索，用户需要在搜索结果中自行提取所需信息，多次搜索仍难以找到合适信息。

随着人工智能技术，尤其是大语言模型（LLM）技术的迅猛发展，企业知识管理的潜力有待进一步挖掘。LLM如ChatGPT、Qwen[2]、Gemini[3]、Gemma[4]等，具有良好的自然语言理解能力，不仅可以处理和分析大量文本数据，还能够生成高质量摘要，回答复杂的查询，甚至推动自动化决策。这些能力有助于大幅提升知识管理的效率和智能化水平。但是大语言模型在生成最终答案时，因自身专业领域知识不足、知识更新不及时以及企事业单位数据无法获取等原因，会出现幻觉而生成不当内容，这在要求内容准确、专业、合规的政企领域成为应用推广的最大障碍。检索增强生成（RAG）应运而生，成为了当前业界解决该问题的核心技术。

基金项目：江苏省重点研发计划项目（BE2023025）

RAG技术概念最早由Meta提出[5]。受限于当时较差的语言模型能力，尽管RAG技术已经在多个知识密集型自然语言处理（NLP）任务上取得了不错效果，但其并未引发更多的关注。在大模型时代，模型的性能取得了巨大的提升，伴随而来的幻觉问题使RAG技术重新进入人们的视野。通过从多数据源中获取外部知识，结合搜索技术和LLM的提示词功能，RAG向大模型提出问题，并把问题在多数据源中进行搜索获取的知识作为背景上下文，将问题和背景上下文信息整合到LLM的提示词中，从而让LLM做出最终的准确回答。

在大模型时代，RAG的发展可分为3个阶段。1）基础RAG（Native RAG）：遵循传统的工作流程包括索引、检索和生成3个模块，也被称为“检索–读取”框架。首先各类知识被分割成离散的块，然后利用embedding模型构建这些块的向量索引；其次，RAG根据查询和索引块的向量相似性识别和检索块；最后，模型根据从检索到的块中获得的上下文信息合成响应。2）高级的RAG：通过丰富的前处理和后处理技术，在信息检索精度和准确率上取得了显著效果。3）模块化的RAG（Modular RAG）：将RAG前、后处理等技术抽离出来并形成模块，进行组合。模块化RAG相比于传统的Native RAG框架，提供了更好的通用性和灵活性。

本文中，我们设计了基于RAG架构的LLM知识管理系统。该系统在充分利用LLM提高知识管理水平的同时，有效缓解了LLM可能产生的幻觉和不当内容问题。

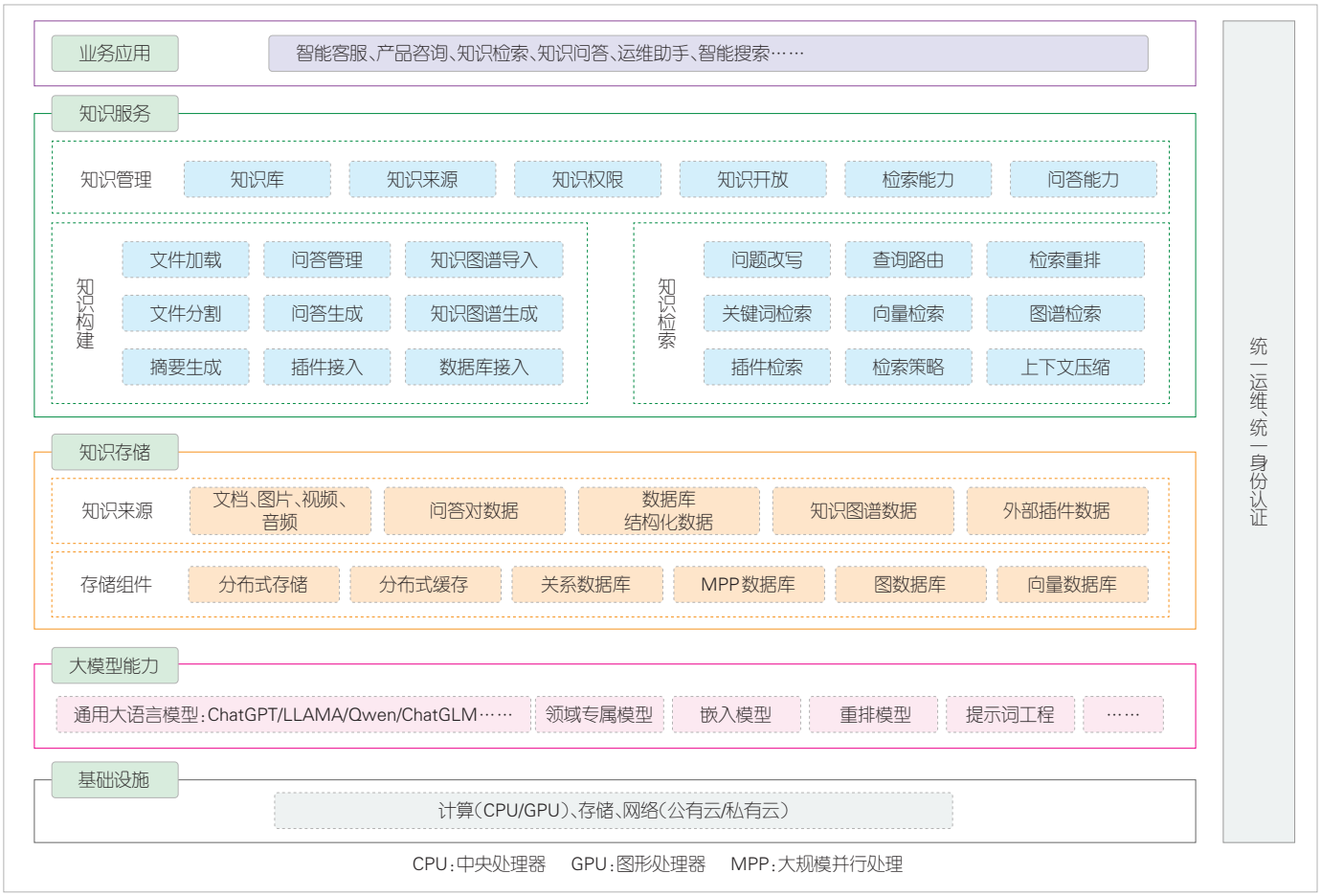
1 系统方案

知识管理系统旨在将人类知识以计算机可理解的形式表示出来，并使计算机能够理解、推理和应用这些知识。这项技术涉及知识表示、知识获取、知识推理、知识存储和知识管理等方面。

1.1 系统架构

如图1所示，LLM知识管理系统架构主要分为以下几个部分：基础设施层、大模型能力层、知识存储层、知识服务层和业务应用层。

基础设施层是构建LLM知识管理系统的底层基础，包



▲图1 大模型知识管理系统架构

括运行系统所需要的计算、存储和网络资源，特别是用于模型部署和推理需要的图形处理器（GPU）资源。部署方式可以是基于公有云服务的部署，也可以是基于企业内部私有云的部署。

在基础设施层之上是大模型能力层，该层包括多种预训练的通用LLM，如ChatGPT、Gemma、LLAMA[6]、Qwen、ChatGLM[7]等，用于理解和生成自然语言，是系统的智能核心。它不仅包括针对特定领域训练或微调以适应特定领域的专属LLM，还包括用于知识构建和知识检索过程中的嵌入模型、重排模型等其他模型，以及使用这些模型的提示词工程。它通过调用基础设施层的计算资源，为整个系统的其他各层提供大模型服务。

知识存储层负责存储和管理企业的知识资产。该层在系统中主要提供知识的存储服务。其中，传统的数据库系统用于存储结构化数据，分布式存储系统用于存储文档、图片、音频和视频等非结构化数据，图数据库用于存储知识图谱数据，外部插件系统用于访问通过外部应用程序编程接口（API）获取的外部知识（例如搜索引擎API等）。另外，向量数据库用于存储基于大模型嵌入技术产生的向量数据。

知识服务层分为3个部分，分别是知识构建、知识检索和知识管理。知识构建主要将来自多种来源的知识数据进行预处理，然后导入到系统，并使用知识存储层的存储组件进行存储。常见的知识数据来源包括非结构化的文档数据、结构化数据库数据、问答（QA）数据、知识图谱数据以及外部API插件数据等。知识检索主要实现根据用户问题获取知识答案的过程。检索的第一步需要对用户问题进行理解和改写，随后采取多种方式进行检索。多种检索方法获得的数据还会经历重排过程，并由大模型最终理解后生成检索结果。知识管理将系统能力统一封装和管理，对业务层提供知识服务能力，同时封装统一的知识开放接口、知识检索能力接口和知识问答能力接口供上层业务层使用。

业务应用层展示了基于LLM知识管理系统构建的常见业务应用。通过知识服务层提供的知识服务，该层提供了以问答方式提供服务的智能客服、面向市场售前人员或客户的产品咨询助手，面向企业提供知识检索和知识问答应用（特别是图书馆图书检索、档案馆档案检索、法律法规条文检索、知识产权专利检索）、复杂系统和场景的运维服务助手，以及基于大模型新一代搜索引擎等应用。

1.2 业务流程

基于LLM的企业知识管理系统的业务流程主要包括知识构建流程、知识检索流程和基于大模型的答案生产流程，如图2所示。知识构建流程包括知识数据预处理、建立索引和知识存储，主要是将企业内部的数据库、知识图谱、文档，外部的Web知识以及构建的QA对进行统一的处理，并存储为企业知识库的统一形式，以完成企业知识数据的处理和构建。知识检索流程包括检索前处理、知识检索、检索后处理、答案生成等步骤。其中，知识检索前处理和检索后处理是可选步骤，在基础知识检索过程中，可能会缺少相关步骤。在知识库构建完成后，用户使用企业知识库进行知识检索。知识检索过程将获取与用户问题相关知识内容的上下文信息。最后，基于知识检索过程获得上下文内容，LLM生成最终答案。

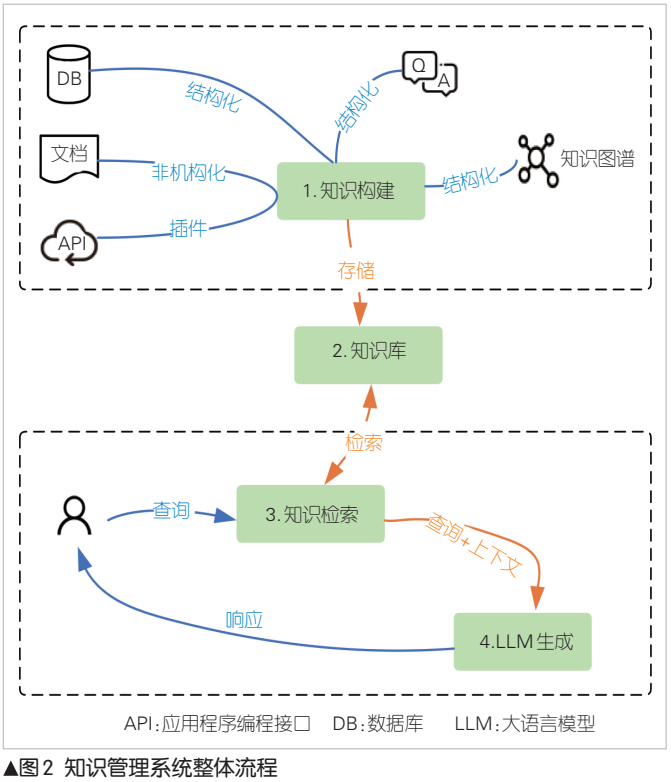
2 关键技术

2.1 知识构建技术

知识构建是企业知识管理系统的核心部分，负责将企业原始知识数据转化为易于存储和检索使用的结构化知识，并将其存入知识库进行管理。构建的知识库的知识质量决定了RAG的最终检索质量。企业的知识来源丰富多样，包括文档、知识图谱、数据库、外部插件等。

2.1.1 文档知识

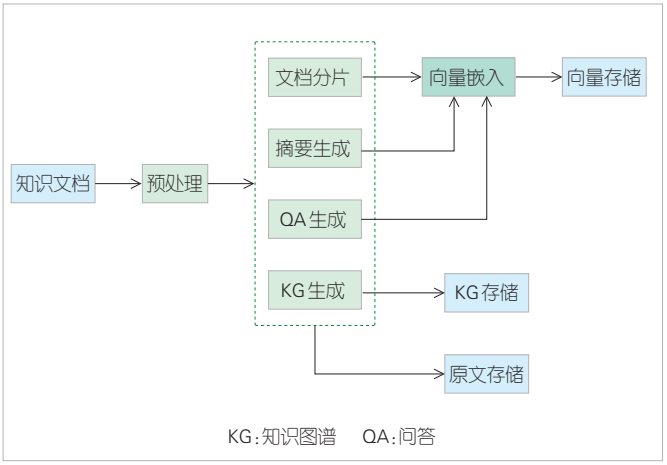
文档型知识是企业知识的主要来源之一。通常，文档型



▲图2 知识管理系统整体流程

知识需要经过预处理、文档切分、向量嵌入等过程，才能完成从原始文档数据到知识库中知识的转变。其中，文档切分算法是一个关键的技术。良好的切分算法应该在满足切片大小的限制的同时，保证每一个切片的语义相对完整。常见的切分算法包括按段落递归切分、按标题切分、按行切分、按固定分隔符切分、按标题切分、按语义切分等。具体的文档知识构建流程如图3所示。

结构化良好的文档，比如word、pdf、html、Markdown等格式文档，通常具有章节结构或标题层次信息。因此，我们可以考虑文档本身的章节或标题层次结构信息，使用按标题切分算法将标题内容和正文内容综合起来进行切片，通过在每一个切片内容的头部添加切片所在的章节或标题信息，使切片内容可以更好地保持原始文档中的语义信息。



▲图3 文档知识构建流程

2.1.2 知识图谱知识

知识图谱通过将知识组织成网络结构的图来表示，它连接了各种实体和它们之间的关系，为知识提供了一种直观的结构化表示。知识图谱构建流程一般包括数据预处理、实体识别、关系提取、属性提取、知识整合和存储等步骤。其中，实体识别和关系提取是较为关键的步骤，传统上可以分别调用专业的小模型，如在实体识别任务上取得SOTA的W2NER[8]、LERERT[9]等模型，在关系提取上取得SOTA的CasRel[10]模型等，来完成相应任务。而在大模型时代，由于LLM具有强大的语义理解能力，可以使用LLM通过预设好的Prompt进行实体识别和关系抽取，形成三元组进行存储。

2.1.3 数据库知识

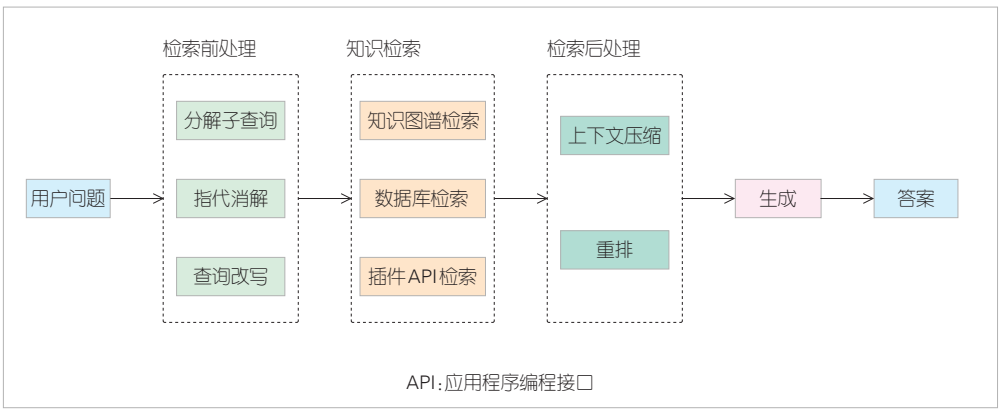
数据库知识指的是存储在传统关系数据库、分析型数据库等数据库中的知识。在信息化建设过程中，企业一般都陆续积累了大量的数据库数据。通过关系数据库理论或数据仓库理论，企业建立了相关的数据库和表，在企业知识管理系统中并不需要重复建设这部分知识数据，但是需要将这部分数据纳入知识管理系统中，以便用户方便地使用已有的数据库知识。

2.2 知识检索技术

知识检索是RAG的核心过程，也是企业知识管理系统的最重要的部分，其目的是确保用户能够快速、准确地找到所需的信息。知识检索包括基础检索流程和复杂检索流程。其中，典型的复杂检索流程包括检索前处理、知识检索、检索后处理、答案生成等步骤。每一个检索步骤都涉及众多的技术细节，而基础检索流程常省略检索前处理和后处理环节。为了提升RAG的准确度，我们采取了多种前后处理技术，并采用了混合知识检索。知识检索具体流程如图4所示。接下来，我们对前处理、知识检索、后处理分别进行介绍。

2.2.1 前处理

在企业知识检索系统中，前处理是指对用户查询进行预处理的一系列技术和方法，旨在优化查询，以提高检索效率和准确性。前处理方法有很多，包括多查询扩展、分解子查询、术语替换、补全历史、指代消融、假设答案、StepBack提示词、查询改写、查询路由等。通过前处理，企业知识检索系统能够更有效地理解用户的查询意图，优化查询以适应复杂的检索环境，从而提供更准确、更相关的检索结果。这里我们将对分解子查询、指代消解、查询改写进行介绍。



▲图4 知识检索流程

2.2.1.1 分解子查询

分解子查询的核心理念在于将一个复杂的原始查询拆分成若干个更小、更易于处理的部分，其中每个部分均代表一个信息独立的子问题。为了实现这一目标，可以采用多查询检索器。该检索器借助LLM，从多个维度自动生成针对给定用户输入的多个查询，进而自动执行提示优化流程。对于生成的每个子查询，多查询检索器都会检索一组与之相关的文档，并最终对所有子查询检索到的文档采取并集操作，从而构建出一个更广泛的潜在相关文档集合。分解子查询可以突破基于向量距离检索方法的某些局限性，从而获取一组更为丰富和多元的检索结果。

2.2.1.2 指代消解

指代消解技术适用于处理用户查询中含有指代词（如“它”“这个公司”等）的情况，有助于提高检索系统对用户查询的准确性。传统通过微调BERT进行指代消解的技术往往只适用于有限、简单的查询语句，在LLM时代，相较于传统的依赖专用小模型进行微调的方法，可以采用Few-shot Prompt并结合思考-行动-观察（CoT）的策略进行指代消解。通过将一些常见的指代消解场景作为Few-shot例子集成到LLM的Prompt中，结合CoT方法，LLM能够分析并处理更复杂的指代消解问题。

2.2.1.3 查询改写

在知识检索系统中，查询改写技术常常可以大幅提高检索准确度。它通过对用户最初的查询进行语言层面的优化与调整，可以增强检索效率并提高结果的精准度。此技术特别适用于处理那些表达模糊不清、含义不明确或结构过于复杂的查询。通过这种方式，系统能够更精确地把握用户的信息需求，并返回更相关的检索结果。

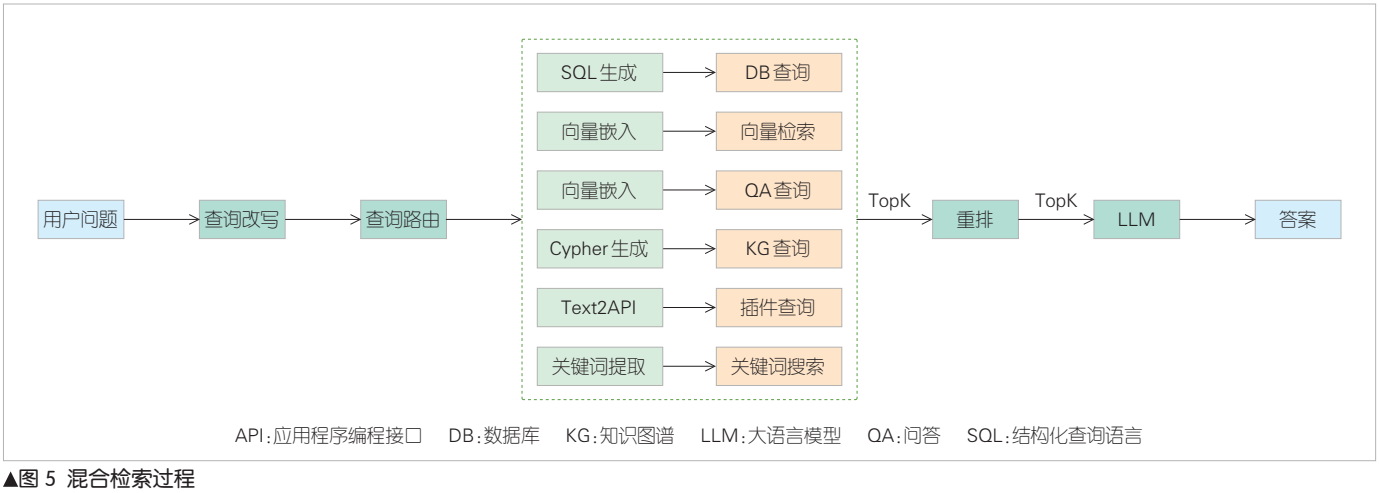
为提高检索系统的准确性，查询改写技术依靠LLM的强大能力，利用精心设计的提示词，让LLM能够有效地改写用户的查询。为了进一步提升查询改写效果，我们还可以引入一个辅助模型“重写器”[11]。这个辅助模型专门负责调整用户查询，使其更好地适应固定检索器和LLM的处理要求。辅助模型重写器可以通过收集领域数据进行有监督的预训练或微调获得。这样，重写器就能更好地满足实际应用场景中的改写需求。

2.2.2 知识检索

在知识检索过程中，通常采用多种检索策略来增强检索的深度，提高检索结果的准确性。如图5所示，混合检索通常将用户查询问题进行改写生成一个或多个查询。经过查询路由模块后，这些查询问题被分发到不同的检索方法流程中。常见的检索方法包括数据库查询、向量检索、QA检索、知识图谱检索、插件检索、关键词检索等。经过多重检索方法检索后每一种检索方法将输出TopK个检索结果。由于不同检索方法生成的检索结果打分标准不同，它们并不能简单地组合在一起进行排序，这时候就需要引入新的重排算法来对这些TopK结果进行组合和重新排序，从而得到最终的TopK，并丢弃其他的候选检索结果。这些最终被选中的TopK将作为上下文和用户查询问题一起交给大模型，让LLM基于上下文内容为用户的提问生成答案。

2.2.2.1 知识图谱检索

知识图谱检索是一种利用知识图谱信息来检索和提供与特定任务相关信息的技术。传统的知识图谱检索较为复杂，一般包括从查询中进行实体识别、关系识别和查询匹配等步



▲图5 混合检索过程

骤。每一个步骤往往都需要专门微调一个小语言模型，而且对于不同的知识图谱，往往需要重新进行微调训练，时间成本较高。在大模型时代，利用大模型出色的语义理解能力和prompt提示词工程，我们仅需要一个大模型就可以较好地对多个知识图谱进行知识检索。

基于大模型的知识图谱检索有两种方式：Text2Cypher和GraphRAG。其中，Text2Cypher将用户问题翻译成图数据库能够识别的Cypher语句，然后调用图数据库接口执行这个生成的Cypher语句以获得执行结果，并将执行结果通过LLM能力生成最终答案。GraphRAG通过构造子图（Sub-Graph）方式来利用知识图谱中的上下文知识以处理用户查询。它首先从用户输入的查询内容中提取实体，然后通过构建与查询相关实体的子图来建立上下文，最后将子图信息作为上下文和用户查询一起送给大模型以给出准确的回答。知识图谱检索工作流程如图6所示。

2.2.2.2 数据库检索

Text2SQL，也称为NL2SQL，是指将自然语言（NL）查询转换为关系型数据库中可执行的SQL查询语言的过程。用户能够以自然语言形式提出查询请求，无须编写SQL语句，从而降低了与数据库交互的复杂性。与知识图谱检索类似，传统的Text2SQL方法也存在流程复杂、组件冗余的情况。同时，采用传统的Text2SQL方法，准确性也难以得到保障。通过引入大模型，我们可以加速整个Text2SQL的流程，并将准确率由原先的60%提升到80%[12]。

Text2SQL进行数据库数据检索主要包括以下步骤：首先利用Schema过滤器筛选与用户输入相关的Schema，然后将筛选的Schema列表与问题一并交给大模型，利用大模型生成SQL语句并执行，最终借助大模型对SQL的执行结果进行分析和总结。

2.2.2.3 插件API检索

插件API检索是指通过API调用外部服务或功能的过程，这被视为LLM与外部世界交互的一种方式。这种交互经常涉及函数调用（Function Calling）。更具体地，它涉及通过API发送请求和接收响应。这些API可能由第三方服务、工具集或自定义实现提供，比如：OpenAI的联网检索和代码解释器就是常见的两种插件检索应用形态。在传统的插件API检索中，面对繁杂的插件API，系统往往难以准确调用正确的插件API。在大模型时代，大模型能够较好地通过API描述，并结合查询，从而较为准确地调用相关API进行检索。

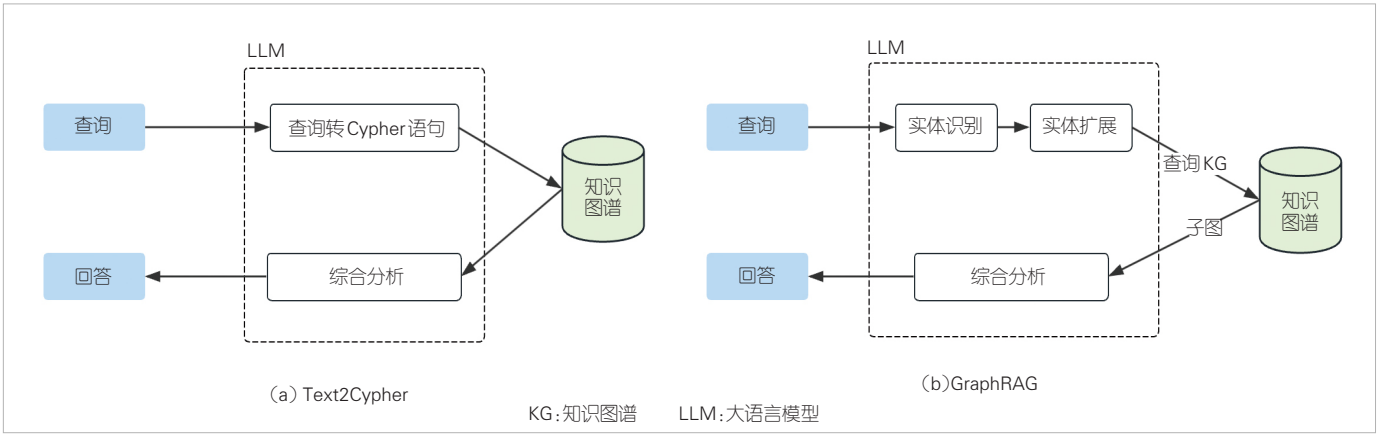
插件API检索的流程首先需要使用API过滤器将用户通过自然语言输入的用户查询进行筛选过滤，从中取出TopK候选相关的API，并将这些API定义和用户查询一起送给大模型进行处理。对于支持Function Call功能的LLM，它会返回函数调用的名称和参数等信息。

2.2.3 后处理

后处理（Postprocessing）阶段负责对检索结果进行进一步的优化和调整，以提高检索系统性能和检索结果质量。这一阶段的核心活动包括但不限于对检索结果进行筛选、压缩和重新排序等操作。进行这些操作的目的是为了精炼并整理出一组最终结果。这些结果随后将被提交给LLM以生成响应答案。在本节中，我们将对上下文压缩、重排技术进行介绍。

2.2.3.1 上下文压缩

通过RAG获得的上下文长度常常达到数千个tokens。当检索步骤所获得的结果内容较多并超出大模型上下文长度时，需要对上下文进行压缩处理以去除冗余信息，查询无关



▲图6 知识图谱检索

噪声，同时保持语义不丢失，进而为LLM提供更有效的上下文信息。常见的上下文压缩方法有内容摘要、关键词提取、LongLLMLingua[13]等。其中，LongLLMLingua通过使用对齐并训练好的小模型来检测移除上下文中不重要的token，并将其转换为人类难以理解但LLM易于理解的形式，有效提升了系统性能。LongLLMLingua的核心思想是将长输入分两步处理：首先使用一个小型编码器模型（通常是BERT等双向编码器）将长输入编码为一个较短的向量表示，然后将编码后的向量连同查询一起输入到一个LLM中（LLM解码器能够识别小型编码器编码后的信息），生成最终的输出。

2.2.3.2 重排

在检索后处理阶段，为确保最相关且最有价值的检索结果能够优先被用作回答查询的上下文输入，我们引入了重新排序（Reranking）机制。重排操作通过对检索阶段获得的检索结果相关性评分进行再次调整，或采用更精细的排序算法，从而实现检索结果的重新排列。重排的关键在于设计高效的打分模型。常见的做法是引入交叉编码器。对于给定查询，交叉编码器将所有检索结果与之进行编码打分，然后按得分递减排列，得分最高者即为最相关检索结果。

为进一步提升重排性能，我们采用了经过训练的专门用于重排的模型，其中Cohere公司的Cohere重排模型和智源的bge-rerank[14]模型因具有代表性而被广泛使用。本文中，我们选用了bge-rerank作为重排器，搭配bge-embedding模型进行文档嵌入，取得了良好效果。重排环节的优化有助于提高上下文的相关性和质量，从而为最终答案生成提供更为可靠的语义支撑。

2.3 答案生成技术

答案生成技术是指，依赖LLM本身的推理能力，结合系统提供的上下文信息进行最终的答案生成。目前，根据开源情况，主流的LLM可以分为以ChatGPT为首的闭源模型和以LLaMA、Qwen为首的开源模型两类。

在闭源大模型中，OpenAI的ChatGPT-4常常在各大评测排行榜中名列前茅，而近期出现的Claude3也显示出了强大的性能。然而，尽管这些模型性能强劲，但由于它们是闭源模型，只提供API调用接口，费用昂贵，不适用于企业知识库中需要频繁调用的场景。此外，企业知识管理系统通常涉及大量的企业内部知识，这对闭源商业模型的隐私保护提出较高要求。

本文所提知识管理系统方案采用了开源大模型。在开源大模型中，比较有名的包括清华大学的ChatGLM、阿里的Qwen以及Meta的LLaMA，具体的参数规模和说明如表1所示。可以看出，ChatGLM-6B受限于参数规模，相较于14B的模型性能略有不足，而LLaMA模型本身只支持英文，即使引入了中文补丁，在中文语境下，Qwen模型性能更胜一筹。综合考虑，我们在方案中选择了Qwen-14B模型。

3 测评框架

为全面评估基于RAG架构的知识管理系统的性能表现，我们需要一个科学全面的测评框架。由S. ES等于2023年9月提出的检索增强生成评估（RAGAs）[15]开源评估框架在业界取得了良好的反响。RAGAs能够快速对RAG系统进行综合评估，所需的输入包括：用户提出的查询问题（Question）、RAG系统生成的答案（Answer）、检索到的与问题相关的上下文文档（Contexts），以及人工标注的参考答案（Ground Truths）。

在获得上述输入信息后，RAGAs基于以下4个评估指标对RAG系统效果进行量化评分：

1）Faithfulness，衡量生成答案与上下文是否保持一致，反映了系统回答的可信赖性。

2）Answer Relevancy，评估生成答案与参考答案的语义相关度，考察答案的准确性。

3）Context Relevancy，测量检索上下文与问题的关联程度，体现上下文选择的恰当性。

4）Context Recall，计算系统检索到的相关上下文数量

▼表1 开源大模型参数规模和说明

模型名称	参数大小/亿	MMLU	CEval	AGIEval	推理显存/GB
ChatGLM-6B[7]	62	36.90	38.90	/	6
LLaMA-7B[6]	70	35.10	27.10	23.90	6
LLaMA-13B	130	46.94	/	33.90	10
Qwen-7B[2]	70	56.70	59.60	/	8
Qwen-14B	140	66.30	72.10	/	13

MMLU：大规模多任务语言理解

占总相关上下文的比例，反映上下文覆盖的完整性。

通过同时关注上述4个维度指标，RAGAs可以综合评估RAG系统在可靠性、准确性和泛化能力等方面的整体水平，为持续优化和改进系统性能提供量化指引。接下来，我们将详细介绍这4个评估指标的具体含义。

Faithfulness是衡量RAG生成的答案Answer与检索到的上下文Context的事实一致性。它是根据Answer和Context计算得出的。Faithfulness的取值范围为0～1之间，且越高越好，计算公式如公式（1）所示：

$$\text{Faithfulness score} = \frac{|\text{Number of claims that can be inferred from given context}|}{|\text{Total number of claims in the generated answer}|}。\quad (1)$$

Answer Relevancy是评估RAG生成的答案（Answer）与用户问题（Question）之间的相关程度。当RAG生成的答案不完整或包含不相关的信息时，系统则将获得较低分数。Answer Relevancy的取值范围为0～1之间，且越高越好，计算公式如公式（2）所示：

$$\text{Answer Relevancy} = \frac{1}{n}\sum_{i=1}^{n}\text{sim}(q,q_i),\quad (2)$$

其中，q为原始问题Question，q_i为提示LLM生成基于该Answer的可能的第i个问题，$\text{sim}(q,q_i)$是计算原始问题q和生成问题q_i的余弦相似度。

Context Relevancy衡量检索到的上下文Context的相关性，根据用户问题Question和检索到的上下文Context计算得到，取值范围在0～1之间，值越高表示相关性越好。理想情况下，检索到的Context应只包含解答Question的信息，计算公式如公式（3）所示：

$$\text{Context Relevancy} = \frac{|S|}{|\text{Total number of sentences in retrived context}|}。\quad (3)$$

Context Recall是衡量检索到的上下文Context与人类提供的真实答案Ground truth的一致性程度。它是根据Ground truth和检索到的Context计算出来的，取值范围在0～1之间，值越高表示性能越好，计算公式如公式（4）所示：

$$\text{Context Recall} = \frac{|\text{GT sentences that can be attributed to context}|}{|\text{Number of sentences in GT}|}。\quad (4)$$

4 结束语

本研究旨在构建一个基于RAG架构的大型企业知识管理系统，以期为企业提供高效的知识检索和利用能力。本文中我们提出了基于RAG构建企业知识管理系统的架构、流程和方法。该系统采用开放的系统架构设计，可基于开源或商业LLM构建，充分保障了企业关注的数据安全；支持多种知识来源，包括文档、知识图谱、数据库和问答等，通过深度挖掘和融合这些异构知识源，形成了全面的专业知识基础。此外，我们设计并实现了完整的知识检索方案，包括检索前处理、知识检索、检索后处理和答案生成等环节，并采用了多种创新技术来提升检索效率和答案质量，介绍了使用RAGAs评估框架对构建的企业知识管理系统进行评估和迭代优化的情况。大量用户反馈和实验评估表明，该系统在准确性、知识覆盖范围、检索效率和用户体验等多个维度均有着优异的表现。

然而，系统中仍存在一些需要进一步改进的问题。首先，当前系统所使用的知识来源仍以文本为主，缺乏将多模态知识融入系统的合理方法。其次，尽管采用了多种文档切分和检索优化手段，但在实际应用场景中还需要针对特定的文档内容设计定制化文档切分算法。最后，系统已经较好地缓解了大模型幻觉的问题，但在企业应用场景下还需要考虑企业合规对齐、数据安全等问题。

在未来，我们希望可以设计更多的垂直领域文档切分算法，采取更有效的embedding和Rerank组合模型，进一步提升RAG技术的检索效率和准确度，同时引入最终回答的合规审查机制，构建一个更高效、更安全的基于RAG的大模型知识管理系统。

参考文献

[1] 牛菁. 大数据赋能企业知识管理创新机理与路径研究: 基于华为案例[J]. 中国新通信, 2023, 25(11): 19-21. DOI: 10.3969/j.issn.1673-4866.2023.11.008
[2] BAI J Z, BAI S, CHU Y F, et al. Qwen technical report [EB/OL]. [2024-02-25]. http://arxiv.org/abs/2309.16609
[3] TEAM G, ANIL R, BORGEAUD S, et al. Gemini: a family of highly capable multimodal models [EB/OL]. [2024-02-25]. http://arxiv.org/abs/2312.11805
[4] TEAM G, MESNARD T, HARDIN C, et al. Gemma: open models based on gemini research and technology [EB/OL]. [2024-02-25]. http://arxiv.org/abs/2403.08295
[5] 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. ACM, 2020: 9459 - 9474. DOI: 10.5555/3495724.3496517
[6] TOUVRON H, LAVRIL T, IZACARD G, et al. LLaMA: open and efficient foundation language models [EB/OL]. [2024-02-25]. http://arxiv.org/abs/2302.13971
[7] ZENG A H, LIU X, DU Z X, et al. GLM-130B: an open bilingual pre-trained model [EB/OL]. [2024-02-25]. http://arxiv. org/abs/2210.02414
[8] LI J Y, FEI H, LIU J, et al. Unified named entity recognition as word-word relation classification [J]. Proceedings of the AAAI

conference on artificial intelligence, 2022, 36(10): 10965–10973. DOI: 10.1609/aaai.v36i10.21344
[9] LIU W, FU X Y, ZHANG Y, et al. Lexicon enhanced Chinese sequence labeling using BERT adapter [C]//Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers). Association for Computational Linguistics, 2021: 5847–5858. DOI: 10.18653/v1/2021.acl-long.454
[10] GAO L Y, MA X G, LIN J, et al. Precise zero-shot dense retrieval without relevance labels [C]//Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). Association for Computational Linguistics, 2023: 1762–1777. DOI: 10.18653/v1/2023. acl-long.99
[11] MA X B, GONG Y Y, HE P C, et al. Query rewriting in retrieval-augmented large language models [C]//Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing. Association for Computational Linguistics, 2023. DOI: 10.18653/v1/2023.emnlp-main.322
[12] GAO D W, WANG H B, LI Y L, et al. Text-to-SQL empowered by large language models: a benchmark evaluation [EB/OL]. [2024-02-25]. http://arxiv.org/abs/2308.15363
[13] JIANG H Q, WU Q H, LUO X F, et al. LongLLMLingua: accelerating and enhancing LLMs in long context scenarios via prompt compression [EB/OL]. [2024-02-25]. http://arxiv.org/abs/2310.06839
[14] XIAO S T, LIU Z, ZHANG P T, et al. C-pack: packaged resources to advance general Chinese embedding [EB/OL]. [2024-02-25]. http://arxiv.org/abs/2309.07597
[15] ES S, JAMES J, ESPINOSA-ANKE L, et al. RAGAS: automated evaluation of retrieval augmented generation [EB/OL]. [2024-02-25]. http://arxiv.org/abs/2309.15217

作 者 简 介

周扬，东南大学和南京邮电大学特聘副教授；研究领域包括人工智能、大数据、物联网等；参与国家级重点项目8项，在人工智能、大数据、物联网等领域具有近20年的大型产品研发与管理经验；申请发明专利20余项。

蔡霈涵，南京邮电大学在读硕士研究生；主要研究方向为人工智能。

董振江，南京邮电大学教授、博士生导师，国务院政府特殊津贴专家，中国人工智能学会常务理事；主要研究方向为人工智能、数据安全与区块链。

SASE关键技术与产业发展研究

Key Technology and Industry Development of Secure Access Service Edge

柴瑶琳/CHAI Yaolin，韩维娜/HAN Weina，
张云畅/ZHANG Yunchang，穆琙博/MU Yubo，韩淑君/HAN Shujun
（中国信息通信研究院，中国 北京 100191）
(China Academy of Information and Communication Technology, Beijing 100191, China)

DOI：10.12142/ZTETJ.202402011
网络出版地址：http://kns.cnki.net/kcms/detail/34.1228.TN.20240423.1324.002.html
网络出版日期：2024-04-23
收稿日期：2024-03-02

摘要：安全访问服务边缘（SASE）有机融合云、网、算和安全，支持随时随处的一站式安全访问，是产业数字化转型的革新性网络安全技术。SASE关键技术包括软件定义广域网（SD-WAN）、零信任网络访问（ZTNA）、云原生网络及网络安全即服务等技术，是网络融合安全架构演进的必然趋势。目前中国SASE产业还处于初期探索阶段，需抓住发展机遇，加强技术产品研发创新和标准体系构建，促进SASE产业高质量发展。

关键词：SASE；SD-WAN；ZTNA；云原生网络；网络安全即服务

Abstract: Secure access service edge (SASE) is a revolutionary network security technology for digital transformation of industry by organically integrating cloud, network, computing, and security to support one-step security access anywhere and anytime. The key technologies of SASE include software defined wide area network (SD-WAN), zero trust network access (ZTNA), cloud native network, and network security as a service, which are the inevitable trend of network convergence security architecture evolution. At present, China's SASE industry is still in the early stage of exploration, so it is necessary to seize the development opportunity, strengthen the innovation of technology product development and standard system construction, and promote the high-quality development of SASE industry.

Keywords: SASE; SD-WAN; ZTNA; cloud native network; network security as a service

引用格式：柴瑶琳，韩维娜，张云畅，等．SASE关键技术与产业发展研究[J]. 中兴通讯技术，2024，30(2)：72-75. DOI：10.12142/ZTETJ.202402011
Citation：CHAI Y L, HAN W N, ZHANG Y C, et al. Key technology and industry development of secure access service edge [J]. ZTE technology journal, 2024, 30(2): 72-75. DOI: 10.12142/ZTETJ.202402011

伴随产业数字化转型进程的加快，云网融合应用场景不断深化，网络安全对产业高质量发展的保障作用也不断凸显。各国高度重视网络安全领域的战略布局和创新研究，抢抓国际新技术主导权。近年来，中国也在不断推进网络安全融合领域的新技术创新应用。安全访问服务边缘（SASE）融合网络和安全创新技术（软件定义广域网、零信任、云原生网络等），以及全面构建云-网-算-安全一体化服务，已成为全球网络安全领域关注的研究焦点。

本文将聚焦SASE热点技术，围绕当前发展现状，重点分析关键技术体系，探讨领域应用挑战并提出未来发展建议。

1 SASE发展现状

1.1 国际SASE战略部署加快

SASE成为各国重塑政府整体网络安全架构的首要选择。2021年12月，加拿大政府在《网络与安全战略》文件中明确提出把SASE作为远程办公场景下替代虚拟专用网络（VPN）的重点技术[1]。2022年6月，美国联邦调查局（FBI）发布的“网络企业重新设计计划”（NERI）最新信息请求显示[2]，FBI对规划架构提出了大量具体的安全要求，如零信任、SASE、强隔离、可见性等。

1.2 全球SASE技术标准体系不断完善

SASE技术发展趋近成熟，国际化组织标准建设进程不断加快。根据Gartner 2021年发布的《Hype Cycle for Emerging Technologies》[3]报告统计，SASE位于全球网络创新技术领域最受关注的前30。目前SASE已跨过早期阶段，进入中期阶段，将在未来2～5年重构网络服务业务模式，推动产业变革性发展。紧跟技术发展，全球SASE标准制定步伐不断加快。2022年，全球城域以太网论坛（MEF）联合微软、Verizon Business等全面推动包括MEF W117 SASE

服务属性和框架等标准的制定。中国SASE标准建设进程在产业各方的合作下也逐步推动。中国通信学会等第三方组织开展了《安全访问服务边缘（SASE）整体方案技术要求》《安全访问服务边缘（SASE）能力成熟度》等标准的编制。

1.3 SASE整体产业发展势头良好

全球SASE产业生态正在形成，供需两方积极行动促进规模化部署。思科、微软、VMware、Palo Alto Networks等一批美国头部企业全面布局SASE市场，融资活跃度较高。此外，电信运营商、网络安全企业、云安全企业、网络设备厂商等产业各方积极加大SASE研发投入，加紧、重点推出SASE产品[4]。同步从需方侧观察，垂直行业包括电信、金融、能源等开始重视采用SASE架构，逐步试点部署。SASE应用正不断从传统网络安全领域扩展到云网/算网融合、边缘安全、物联网等新应用场景。根据国际第三方咨询架构Dell'Oro Group 2022年3月份公布的《网络安全报告》[5]数据显示，2021年SASE网络和安全总支出已超过40亿美元，增长率达到37%。

在新需求、新应用、新威胁的牵引下，以SD-WAN、防火墙、零信任网络访问、云安全访问为主的SASE产品体系不断完善，贯穿基础设施层安全、平台安全、应用安全3个层次以提供一体化网络安全服务。总体上看，SASE整体产业发展稳步向好，市场活跃度不断提升，产品应用不断丰富。

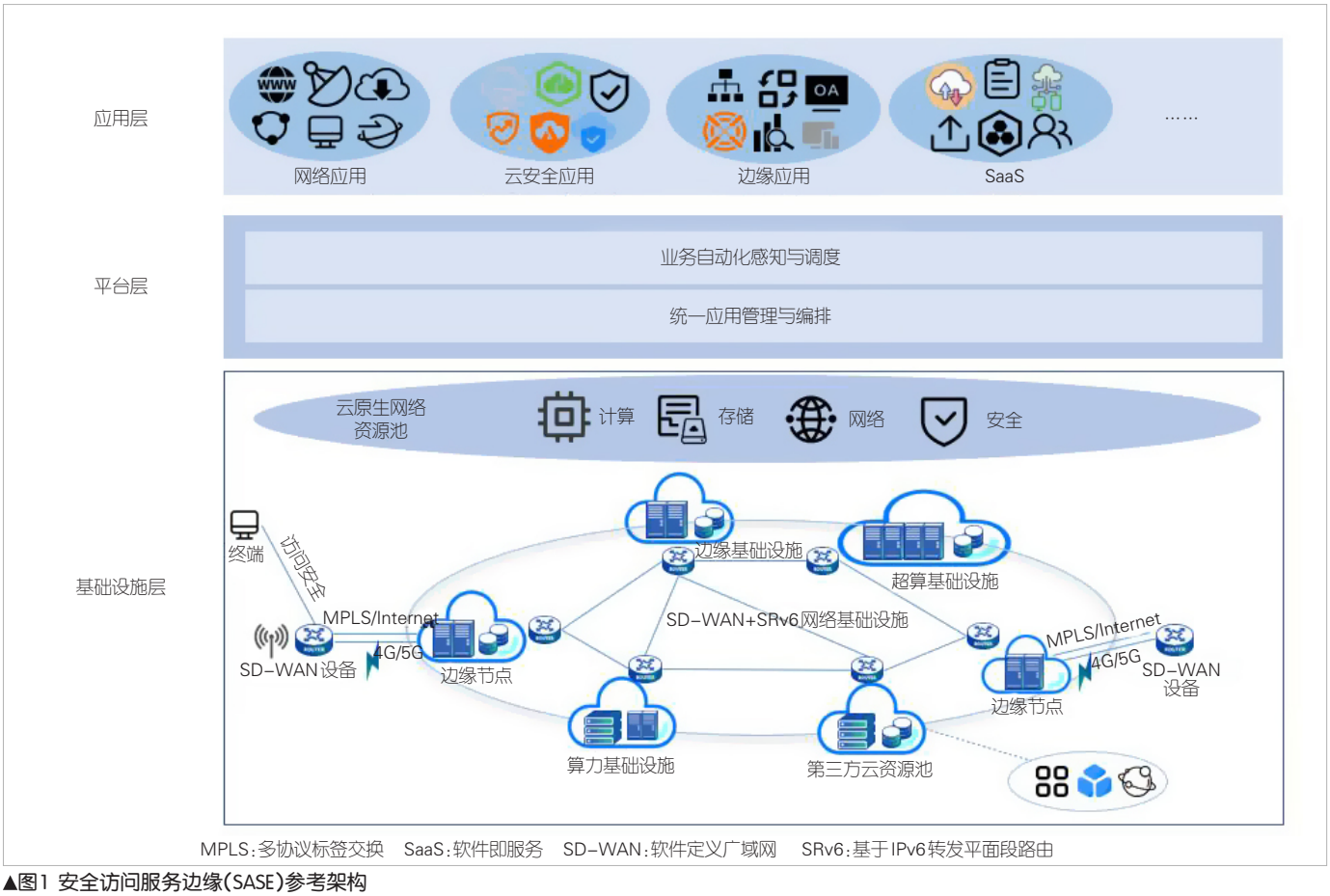
2 SASE参考架构及关键技术

2.1 SASE参考架构

SASE是融合网络能力与安全能力并进行统一管理和交付的创新技术体系，如图1所示，集成了SD-WAN、零信任网络访问、云原生网络、网络安全即服务等多个关键技术能力。SASE目前已经在金融、能源、电信等多场景得到广泛应用，成为数字基础设施建设发展的重要内涵。

从层次结构上看，SASE包括应用层、平台层、基础设施层3个主要部分。

1）SASE应用层：支撑各类应用（包括网络应用、安全应用、边缘应用、SaaS、融合应用等）安全防护要求。



MPLS：多协议标签交换 SaaS：软件即服务 SD-WAN：软件定义广域网 SRv6：基于IPv6转发平面段路由

▲图1 安全访问服务边缘(SASE)参考架构

2）SASE平台层：通过身份安全、网络安全、应用与数据安全功能组件来支撑云、网、算、安全资源的编排安全、运维安全、协同安全等一体化平台安全。

3）基础设施层：作为SASE基础设施资源底座，包括物理基础设施和虚拟化基础设施资源池两层。其中，物理基础设施层包含所有的物理和虚拟的SD-WAN设备、计算节点、边缘节点、公有云、私有云、数据中心、第三方云资源池，虚拟化基础设施资源池将物理基础设施资源统一抽象为计算资源、网络资源、存储资源、安全资源等。

2.2 SASE关键技术

SASE的关键技术包括SD-WAN、零信任网络访问、云原生网络、网络安全即服务等。

1）SD-WAN

SD-WAN作为SASE的网络技术底座，以业务与应用为导向，融合软件定义网络（SDN）、网络功能虚拟化（NFV）、网络编排与探测等多种技术，能够以平台或托管方式提供基础网络连接、广域网加速、安全防御等多种SASE服务。通过对SASE网络的抽象和建模，SD-WAN将上层网络业务和底层网络基础设施具体实现架构进行解耦，通过在SASE平台部署独立的控制面，将网络转发和控制进行分离，从而实现SASE网络集中管控和自动化运维。

2）零信任网络访问

零信任网络访问作为SASE的安全引擎，基于零信任“持续验证、永不信任”的安全理念，通过细颗粒度的身份识别和用户访问行为的上下文信息（包括设备信息、访问时间、访问地点等）来授予动态访问权限，并持续评估访问主体的信任值。零信任网络访问基于统一的数字化身份安全访问体系，赋能SASE形成统一安全访问管控机制，切实保障了无边界、自适应、弹性访问、可持续安全评估的应用体验。

3）云原生网络

云原生网络将SASE网络能力下沉到云中，实现网络资源软硬件解耦、网络资源弹性部署、云网服务一体协同等应用需求。通过k8s开放架构，SASE使用轻量级虚拟化容器技术构建网络功能，支持横跨本地和公有云环境，将防火墙、入侵防御系统（IPS）、网站应用级入侵防御系统（WAF）、用户终端设备（CPE）等网络功能部署在不同容器组（POD）中，作为微服务开发和交付，支持以度量、跟踪和日志记录的方式将每个网络功能POD内部状态外部化等。云原生网络技术全面构建了SASE网络应用新模式，从单体到微服务化转变，实现网络能力插件化、弹性化、自动化。

4）网络安全即服务

网络安全即服务是将SASE网络安全功能（包括防火墙、云访问安全代理、IPS、WAF、SWG等）SaaS化的体现，可提供一站式一体化全流程的安全服务。网络安全即服务重点解决了传统网络安全设备软硬一体、网络安全应用烟囱式、新业务新功能交付低效等主要问题。SASE通过支持网络安全功能云化部署和应用程序编程接口（API）开放化，全面构建网络安全即服务能力体系。网络安全即服务将持续根据SASE实时业务情况来动态创建、响应和变更SASE网络安全能力参数配置，以满足各类资源弹性扩缩、应用轻量化、业务弹性化、服务快速上线的新需求。

3 中国SASE发展面临三大挑战

1）SASE网络和安全统一管理机制尚未形成

产业在推进SASE部署建设时，尚缺乏相应的网络和安全统一管理机制，主要表现在：一是企业对网络安全的重视不足，安全防御意识薄弱，主动建设SASE网络安全平台意愿弱；二是缺乏网络和安全的统一规划，网安业务体系各自为政，亟需SASE相关政策指引；三是SASE作为重要行业关键信息技术基础设施，企业投入资金不足，缺少必要的安全管理保障措施，存在一定的网络安全风险。

2）SASE技术成熟度低且标准体系不完善

SASE技术融合复杂度高，架构部署存在相关技术服务质量参差不齐问题，具体表现在：一是SASE产品服务标准不统一，适用于特定行业特性、需求的SASE应用标准规范缺乏；二是缺乏行业通用SASE平台，亟需探索适应行业特征和发展需求的新型融合架构模式；三是现有网络安全基础设施融合能力不足，SASE产品质量和稳定性有待提高，部分产品在使用过程中可能会出现故障或漏洞，需要及时修复和升级。

3）SASE产业生态仍在初期且发展力量不强

当前，SASE在垂直行业的应用尚未进入大规模部署阶段，SASE产业生态尚未闭环。SASE产业涵盖网络服务提供商、安全服务提供商、设备制造商等，产业生态各方协同性差，供需对接不足。同时，产业仍缺乏融合领域人才培养。

4 SASE产业发展建议

1）加强统筹谋划，形成系统完善的监管机制

一是提升企业网络安全风险认知，推动企业完善自身网络安全管理体系，明确网络安全责权，积极主动增强SASE网络安全平台建设，切实保障关键数据安全；二是针对重点行业关键部门，出台相关政策要求，明确SASE路线图，指

导SASE网络设施和安全设施统一规划、同步建设、分步实施；三是鼓励相关主管部门配套专项资金，并建立健全试点工作机制，推动企业保质保量落实SASE融合试点应用和完善相应保障措施。

2）强化研究新创，建立自主核心的SASE技术体系

围绕SASE的网络体系、平台体系和安全体系，强化核心技术研究，主要包括：一是建立和完善SASE标准体系，指导产业数字化转型工作与SASE关键技术体系的融合应用，持续提升SASE技术与企业业务融合发展水平；二是建设通用技术平台，运用零信任、云原生、SaaS等新一代信息技术，探索构建适应行业特征和发展需求的新型融合架构模式，建设敏捷高效可复用的SASE融合基础设施，提升服务能力；三是优化基础设施，综合采用内置于设备、虚拟化、软件化、可动态加载和配置等方式，实现SASE基础设施升级，增强系统的自身安全属性和灵活部署性，提升安全防护效果。

3）注重培育引导，打造具有竞争力的产业生态

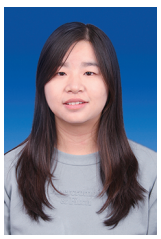
打造SASE产业合作平台，整合电信运营商、网络安全企业、互联网企业等产学研用资源，共同孵化SASE产品方案，实现产业链深度融合。建立SASE行业应用示范标杆，推广SASE优秀案例，全面加深SASE行业实践部署，配套SASE人才培养计划，保障企业SASE网络安全一体化能力建设。

参考文献

[1] 党小东, 柴瑶琳, 穆琋博, 等. 安全访问服务边缘产业发展现状及未来发展趋势 [J]. 信息安全与通信保密, 2023(9): 19-26
[2] Federal Bureau of Investigation. Network enterprise redesign initiative [EB/OL]. (2022-10-01)[2024-02-25]. https://sam.gov/opp/396968fc403b4d838794200355513ae4/view#attachments-links
[3] Gartner.Hype cycle for enterprise networking [EB/OL]. [2024-02-25]. https://blogs. gartner. com/andrew-lerner/2021/10/11/networking-hype-cycle-2021/
[4] 王茜, 陈晨, 井俊丰, 等. 大型企业SASE解决方案及应用实践 [J]. 中兴通讯技术, 2023, 27(2): 45-50. DOI:10.12142/ZTETJ.202301009
[5] Dell'Oro Group. 2021年SASE以37%的增长率改变了市场格局 [EB/OL]. [2024-02-25]. https://mp.weixin.qq.com/s/b689uXP5Vyg0auJVw7Erug

作 者 简 介

柴瑶琳，中国信息通信研究院技术与标准研究所高级项目主管；主要从事SD-WAN、零信任、算网安全等相关技术研究工作；参与20余项行业标准/团体标准研制工作，发表论文10余篇，授权/申请技术专利6项，拥有计算机软件著作权7项，组织发布行业白皮书3个。



韩维娜，中国信息通信研究院技术与标准研究所技术专家；主要从事SD-WAN、网络算力化等相关领域的研究工作；参与多个白皮书撰写、标准研制等工作，目前参编完成著作1本、行业白皮书1个，参与起草行业标准/团体标准5项。

张云畅，中国信息通信研究院技术与标准研究所技术专家；主要从事零信任、算网安全等方面研究工作；参与多个零信任白皮书撰写、标准研制等工作，目前参编完成著作1本、行业白皮书1个，参与研制行业标准/团体标准5项。

穆琋博，中国信息通信研究院技术与标准研究所互联网中心副主任、高级工程师；主要从事算网融合、未来网络、云计算等方面的研究工作；牵头多个技术体系的标准研制，发表论文10余篇，拥有计算机软件著作权12项。

韩淑君，中国信息通信研究院技术与标准研究所高级项目主管；主要研究方向包括算网融合、云计算、人工智能、区块链等；发表论文10余篇，拥有计算机软件著作权3项。

大模型关键技术与应用

Key Technologies and Applications of Large Models

韩炳涛/HAN Bingtao[1,2]，刘涛/LIU Tao[1,2]
（1. 中兴通讯股份有限公司，中国 深圳 518057；
2. 移动网络和移动多媒体技术国家重点实验室，中国 深圳 518055）
(1. ZTE Corporation, Shenzhen 518057, China;
2. The State Key Laboratory of Mobile Network and Mobile Multimedia Technology, Shenzhen 518055, China)

DOI：10.12142/ZTETJ.202402012
网络出版地址：http://kns.cnki.net/kcms/detail/34.1228.TN.20240418.1324.004.html
网络出版日期：2024-04-19
收稿日期：2024-02-18

摘要：介绍了自ChatGPT发布以来，大模型关键技术和应用的主要进展。在大模型设计方面，模型规模不断增加，但已有放缓趋势，更长的上下文以及多模态已经成为主流，计算效率明显提升；在模型训练方面，从单纯追求数据数量逐渐转变为关注数据的多样性和质量，特别是如何使用合成数据训练大模型成为主流探索方向，这是迈向通用人工智能（AGI）的关键；在模型推理方面，模型量化和推理引擎优化极大降低了模型使用成本，诸如投机采样等新兴算法逐渐成熟。在应用层，Agent技术获得了重大进展，在克服大模型固有缺陷方面发挥了不可替代的作用。越来越多的企业开始规划、研发以及使用大模型，企业级大模型应用架构日益成熟完善，并以场景、技术、算法三要素为抓手加速大模型商业价值闭环。

关键词：大模型；模型训练；推理加速；大模型安全；智能体

Abstract: The major advances in key technologies and applications of large models since the release of ChatGPT are presented. In terms of large model design, the model scale is increasing, but it has slowed down. Longer context and multi-mode have become the mainstream, and the computational efficiency has been significantly improved. In terms of model training, the focus has shifted from simply seeking a larger quantity of data to a more focused approach on the diversity and quality of data, especially how to train large models using synthetic data. This is an essential direction towards achieving artificial general intelligence (AGI). In terms of model inference, model quantification and inference engine optimization greatly reduce the cost of model use, and emerging algorithms such as speculative sampling gradually mature. At the application level, Agent technology has made significant progress, playing a critical role in addressing the inherent limitations of large models. More and more enterprises are beginning to plan, develop, and utilize large models, and the enterprise-level large model application architecture is becoming increasingly mature, focusing on scenarios, technologies, and algorithms to accelerate the closing loop of large model commercial value.

Keywords: large model; model training; inference accelerating; large model safety; Agent

引用格式：韩炳涛, 刘涛. 大模型关键技术与应用 [J]. 中兴通讯技术, 2024, 30(2): 76-88. DOI: 10.12142/ZTETJ.202402012
Citation: HAN B T, LIU T. Key technologies and applications of large models [J]. ZTE technology journal, 2024, 30(2): 76-88. DOI: 10.12142/ZTETJ.202402012

2022年底，OpenAI发布了跨时代的ChatGPT应用。这是第一个具有流畅的多轮对话体验、渊博的通识知识，并能够深刻理解人类意图的生成式人工智能（AI）应用。它的成功使大模型成为AI的主旋律，在极短的时间内改变了AI产业的格局。

尽管距离ChatGPT的发布仅过去一年多，但大模型技术已经取得了巨大的进展。随着GPT-4、Gemini、Sora、Claude3、Kimi等一系列大模型的陆续发布，大模型能力已迅速提升，甚至更为强大的通用人工智能（AGI）已初见端倪。本文中，我们试图对ChatGPT发布以来大模型的关键技术做出综述，厘清大模型技术全貌及发展态势，以便读者做出更好的判断和预测。

1 模型设计

目前，主流大模型的结构都是在Transformer基础上不断改进，以实现更强大的语言理解和生成能力、更长的上下文推理能力、更多模态的数据处理能力，以及更高的算力利用效率。

1.1 主流大模型的架构演化

2017年，Google提出Transformer[1]，创造性地将注意力机制作为核心算子来构建语言模型，解决了长短期记忆网络

（LSTM）[2]等循环神经网络计算效率低、训练容易过拟合的问题。此后，OpenAI 和 Google 在 Transformer 基础上，又分别提出了 GPT[3]和 BERT[4]。GPT 采用了 Transformer 的解码器部分，使用从前到后的单向预测模式（类似于补全）。BERT 则采用了 Transformer 的编码器部分，使用上文与下文的双向预测模式（类似于填空）。受益于该模式，BERT 实现了较强的性能，让业界一度认为双向语言模型是更优的选择。

但是，OpenAI 笃定追逐“通用语言模型”，认为从前到后的生成能力可以转化应用于各类语言任务上，因此在后续模型中依然坚持单向预测模式。直到 2020 年，OpenAI 提出了拥有 1 750 亿参数的 GTP-3[5]模型。GTP-3 在对话、知识问答、吟诗作赋等多项任务中展示出的能力均令人印象深刻。

此后，OpenAI 不再公开模型相关的技术细节，研究人员开始把目光聚焦在其他开源模型上。2023 年 Meta 发布 LLaMA 系列模型[6]，进一步优化了 Transfomer 模型架构，并使用更加充分的数据对模型进行训练，获得了不错的性能。此后，许多研究人员相继基于 LLaMA 模型不断地做局部的优化，如 Baichuan、Yi、Mistral、Qwen 等。

1.2 对计算效率的优化

相较于之前 LSTM 等循环神经网络，Transformer 最大的弱点是计算效率低，主要原因是其自注意力机制的计算复杂度与序列长度的平方成正比关系。针对该问题，一条技术路线是放弃 Transformer，设计更高效的模型结构。目前我们认为有可能取代 Transformer 的架构主要包括 Linear Transformer、RWKV 和 Mamba 等。Memba 采用了完全不同的模型架构[7]，彻底抛弃了注意力机制，从状态空间的角度对序列进行建模。相关研究表明，该模型把训练计算复杂度降到线性，且能力与 Transformer 相当。A21 Labs 刚刚发布了首个生产级别的基于 Mamba 的模型 Jamba，模型参数达到了 520 亿，同时提供长达 256 k 的上下文。尽管当前 Memba 在参数规模上与 Transformer 仍然有较大差距，但前景仍被看好。业界认为 Memba 是取代 Transformer 的有力竞争者。

另一条技术路线是继续优化 Transformer 模型结构。例如，在注意力机制方面，代表性的工作包括将多头注意力机制（MHA）改进为多查询注意力机制（MQA）和分组查询注意力机制（GQA），这可以进一步降低计算量。其中，LLaMA2 采用了分组查询注意力机制[8]，Google 最近发布的 Gemini 采用了多查询注意力机制。

除了结构上的改进，优化算子实现也可以提高效率，例如：FlashAttention 通过 Attention 与 Softmax 计算融合，结合 KV Cahche，能显著提高计算速度并降低显存带宽依赖[9]，从而使 Transformer 的计算效率接近线性注意力模型。

1.3 长上下文推理

能够处理的上下文序列长度是语言模型能力的一个关键指标。在许多任务中，如文章阅读理解、代码生成、检索增强生成（RAG）等，都需要模型能够处理很长的上下文。然而，长上下文模型训练与推理，存在计算复杂度高、最大长度约束等问题。

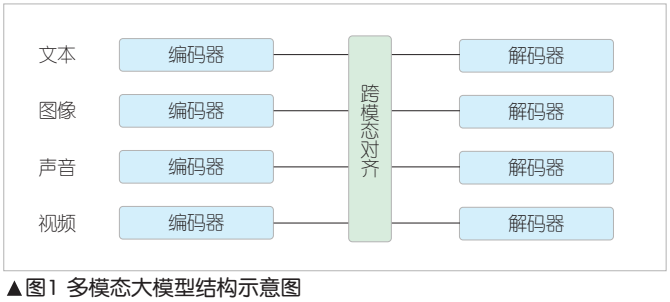
最大长度约束问题，是指模型在实际推理时处理的序列长度超过训练时序列长度，这可能会导致性能的明显下降。针对该问题，有两种解决方案：一种是预训练过程中调整模型设计，可以实现更好的模型外推能力；一种是通过微调和位置嵌入处理，扩大模型的上下文窗口。例如，苏剑林发现如果对注意力矩阵的查询-键-值（QKV）映射矩阵加入 Bias，则使用旋转位置编码（ROPE）的模型可能会获得较好的外推能力。该方法常应用在 Qwen 模型中[10]。此外，研究人员发现，如果对 ROPE 进行插值，并配合简单的微调，可以把上下文窗口的大小从 4 096 扩展到 32 000[11]。

研究人员通过对改进注意力机制，可以降低计算复杂度，其中最具代表性的是 Window Attention 方法。该方法引入了注意力窗口，让每层注意力仅关注序列的局部信息而非全局信息，通过层层堆叠放大模型的上下文感受野。这样就可以把计算复杂度控制在一个明确的范围内，避免随序列长度的无限制增长。该方法常应用在 Mistral 中[12]。此外，还有其他一些替代注意力机制的方法，如上文介绍的线性注意力机制与 Mamba 等模型等。

1.4 多模态能力

多模态大模型可以分为多模态理解大模型和多模态生成大模型。多模态理解大模型输入多模态信息，输出中包含文本信息；多模态生成大模型则相反，其输入中包含文本信息，输出为多模态信息。

自从经典的 BLIP2 模型出现后，多模态大模型设计趋势逐渐稳定，具体如图 1 所示。每种模态均有对应的编码器和



▲图1 多模态大模型结构示意图

解码器。其中，编码器负责接收对应模态信息的输入，并将其编码到语义空间的向量中；解码器则负责从语义空间的向量中解码出对应模态的输出；跨模态对齐单元则负责不同模态语义空间的匹配对齐。这种架构可以很方便地实现多种模态的混合输入和输出。

1.4.1 多模态理解大模型

目前，视觉语言类的多模态理解大模型和多模态生成大模型的发展最为迅速。视觉语言理解大模型（图生文），即对视觉编码器+模态对齐+大语言模型解码器进行组合训练。代表性的工作包括CLIP[13]、BLIP-2[14]、LLaVA[15]和InternLM-XComposer2[16]等。

BLIP-2由预训练好的、冻结参数的视觉模型（CLIP训练的ViT-L/14、EVA-CLIP训练的ViT-g/14）、文本模型（OPT、FlanT5），以及所提出的可训练的Q-Former构成。Q-Former是一个轻量级Transformer，它使用一组可学习的Query向量，从冻结的视觉编码器中提取视觉特征，来对齐文本和语言两个模态的差距，从而把关键的视觉信息传递给大语言模型（LLM）。

LLaVA成功地验证了少量高质量的数据能使模型拥有很强的图文生成能力，其由3部分组成：CLIP预训练模型中的视觉编码器ViT-L/14、一个线性投影层和一个大语言模型LLaMA。LLaVA以图-文对（LAION、CC3M、COCO）数据集为基础，使用ChatGPT/GPT-4来构建指令跟随精调数据集。

零一万物开源Yi-VL多模态大模型[17]也是采用LLaVA架构，使用了Yi-34B-Chat模型，改进了微调训练方法，提高了Yi-VL无缝集成和解释视觉+语言多模态输入的能力。

1.4.2 多模态生成大模型

视觉语言生成大模型（文生图）的解码器部分以扩散模型为主，代表性工作包括Stable Diffusion[18]、DiT[19]、Sora[20]等。

Stable Diffusion的组件和模型组成为：文本编码器，将文本信息转换成数字表示，以捕捉文本中的想法；图像信息创建者，在隐空间中逐步处理扩散信息，以文本嵌入向量和由噪声组成的起始多维数组为输入，输出处理的信息数组；图像解码器，使用处理后的信息数组绘制最终的图像。

Sora是OpenAI发布的文生视频的多模态模型，和Runway、Stable Video Diffusion及PIKA等已有模型相比，其视频生成能力有大幅提升。Sora模型内部分成3个部分：第1部分是变分自动编码器（VAE），包括编码器和解码器两个部分，其作用是对视频进行压缩和解压缩。生成视频的过程是在压缩后的低维隐空间进行计算，相对于直接从原始的像素空间计算，减少了数百倍的计算量。第2部分是基于Transformer的扩散模型，其作用是在隐空间通过迭代降噪过程生成视频。第3个部分是语言大模型作为编码器，其作用是将用户的Prompt编码为一个隐空间的表示，使其在生成视频时内容与文本描述一致，从而使视频内容和用户的Prompt一致。

2 模型训练

GPT开创了生成式预训练方法之后，两阶段训练（任务无关的预训练阶段和任务相关的精调训练阶段）大模型成为主流。预训练阶段的目的是为模型注入大量通识知识，精调训练阶段的目的是提升模型完成特定任务的能力。在这种方式下，仅需少量精调数据即可以让预训练模型具备完成新任务的能力，相比于之前为每个任务端到端完整训练模型，极大节省了训练数据和算力。

随着开源预训练大模型的出现以及应用场景日益复杂，上述两阶段训练方法已不再满足需求，因此出现了更多的训练阶段，如图2所示。由于开源预训练大模型通常使用公开可获得数据训练，专业领域知识不足，使用私域数据对模型再次进行增量预训练可以有效灌注专业知识。大模型在使用过程中需要避免生成各类有害信息，因此在模型完成任务相关的精调训练之后，增加了对齐训练阶段，这样可以使模型输出更加符合人类的价值观。

训练高性能的商用大模型，涉及数据处理、预训练、精调训练、安全等关键技术。

▲图2 大模型多阶段训练过程

2.1 数据多样性与数据质量

数据多样性对于模型能力的全面性来说至关重要，模型学习不同的能力，就需要对应的训练数据。例如GPT-3训练数据的构成既包含Common Crawl这类种类丰富、总量庞大的互联网数据，又包含维基百科、书籍这类的高质量数据。

数据质量是模型能力的决定性因素之一。如今，越来越多的研究人员倾向于使用更少的高质量数据而非海量的低质量数据来训练模型。少而精的高质量数据在大幅降低训练成本的同时，还有可能获得更高的模型性能。

因此，如何评价数据质量成为一个关键问题。2023年底的Ziya2[6]给出了一套非常系统的评估标准，该标准综合了程序评估和人工评估。程序评估涵盖了启发式规则、语言模型、统计指标等方法，并对不同的指标赋予了不同级别。系统化、程序评估与人工评估结合，为数据质量评估方法指明了发展方向。

中兴通讯结合业界研究成果，总结提炼出一套完整的数据管理及处理方法（如图3所示）。在数据管理方面，中兴通讯强调“分级分类”。数据分级是将所有训练数据按质量从低到高分为1—5级，不同质量级别的数据应用于预训练的不同阶段。例如，1级数据为有害数据，严重影响模型性能和安全性，需要从训练数据中予以过滤清除；2级数据为普通网页数据，仅用于模型训练过程中的warm-up阶段；3级数据为高质量网页、书籍、代码数据，用于为模型建立通识知识；4级数据为更高质量的专业书籍、论文、教材、试题数据，用于提升模型的专业性；5级数据为最高质量精调数据，用于大幅提升模型在各种任务评测中的表现。数据分类则是从为模型赋能的角度，将模型能力分为上百个类别，为对应的训练数据建立主题，从而实现数据能力画像。为了实现上述数据分类分级管理，中兴通讯开发一套完整的数据处理框架，如图3所示。

2.2 预训练

生成式预训练是一种让大模型学习世界知识的有效方法。这种方法的主要优势在于：它属于无监督学习方法，无须对数据进行标注就可以得到海量训练数据；只需让模型学习正确预测语料中的词，即可令模型理解语料中所的蕴含知识。

大模型训练算力需求的增长速度远超摩尔定律，因此所需要的并行计算节点数不断增加，算力和显存是大模型训练的主要瓶颈。为了打破设备的算力和显存限制，目前有3个最主要的技术方向：1）以张量并行（TP）、流水线并行（PP），数据并行（DP）构成的3D并行加速，其主要原理是根据硬件集群的特征，如节点数、单节点算力、内存大小、节点间互联带宽及时延等，对计算任务进行不同维度的拆分，以最优化利用硬件资源；2）以ZeRO（零冗余）系列为代表的显存优化技术，其主要原理是尽可能将数据保存到内存从而减少对显存的需求（以增加带宽需求为代价）；3）以Flash Attention为代表的底层算子优化，其主要原理是将多个算子计算进行融合从而最大限度降低“内存墙”对计算效率的影响。

上述算法实现涉及复杂工程优化，因此并行训练框架极为重要。Megatron-LM[21]是由英伟达深度学习应用研究团队开发的大型Transformer语言模型训练框架，其在对TP的支持方面处业界领先。DeepSpeed是微软提出的并行训练框架，其主要优势是支持ZeRO[22]和Checkpoint[23]显存优化技术，对训练显存紧张的场景更友好。FairScale是由Facebook提出的一个用于高性能和大规模训练的PyTorch扩展库，支持全切片数据并行（FSDP），这是扩展大模型训练的推荐方式。

2.3 精调训练

大模型在各类任务中具有泛化能力，然而直接应用预训练模型往往并不能满足所有场景的需求，这就引出了一个关

▲图3 中兴通讯数据处理框架

键技术——精调训练，即针对特定任务或应用场景，在预训练模型的部分或全部参数上进行进一步的学习与优化，提升模型在特定任务上的遵循指令能力、问题解决能力、特定表达方式能力等，从而提高其在该任务上的精度和专业性。

精调训练主要分为全量精调和低资源精调两大类别。其中，低资源精调主要指的是LoRA[24]、prefix-tuning以及P-tuning等方法。这些方法对模型局部进行微调或者冻结一部分参数进行微调。其中，LoRA方法效果最佳。该方法通过低秩近似对预训练模型的部分权重矩阵进行更新，在降低存储成本和计算复杂度的同时，实现模型对目标任务的快速适应。全量精调则在训练过程中会对整个模型的参数进行优化和更新，不仅对计算资源和存储资源要求更高，也更容易出现过拟合、丧失通用性等问题。但全量精调拥有更高的上限，通过适当的训练优化来增强模型的泛化性。全量精调方法能够得到比低资源精调方法更优秀的性能。

由于“对齐税”问题，精调训练在提升特定任务表现的同时会影响模型在其他任务中的总体表现。因此，使用“少而精”的精调训练数据，运用Dropout等技术可以防止过拟合，对平衡大模型的泛化能力和任务适应性尤为重要。

2.4 对齐训练

大语言模型的一些不良行为（例如，不真实的回答、谄媚和欺骗）激发了业界对人工智能对齐领域的深入研究。AI对齐旨在使人工智能系统的行为与人类的意图和价值观相一致，在通往AGI的道路上，AI对齐无疑是安全打开“潘多拉魔盒”的密钥。

对齐训练主要应用强化学习算法，其基本方法是首先基于人类标注数据训练评分模型，然后再基于评分模型运行强化学习算法，通过引导模型获得更高的评分，使其输出更符合人类标准。OpenAI首先提出的基于人类反馈的强化学习（RLHF）就是利用人类的反馈来优化模型的输出，使其更符合人类的偏好和价值观。这不仅能够提升语言模型性能，也能提升安全性和有用性。相比于RLHF涉及多个模型和不同训练阶段的复杂过程，拒绝采样（RS）[25]则更为简洁有效。RS是指让一个模型针对同一个prompt生成K个不同答案，然后利用奖励模型为这K个答案打分，选出最优的答案后，再用最优问答样本对原模型进行监督微调，以增强模型能力。直接偏好优化（DPO）[26]也是RLHF的替代方案之一。DPO利用奖励函数和最优策略之间的映射关系，将约束奖励最大化问题转换为单阶段的策略优化问题。DPO算法因无须拟合奖励模型，且无须在微调期间从LM采样或执行重要的超参数调整，从而实现了稳定性高、性能强且计算量轻等优秀表现，大大简化了实施和训练过程。

中兴通讯的星云大模型在RLHF这一框架的基础上，通过设计质量评优模块并作为对样本自动打分的奖励模型，同时结合RS拒绝采样选取最优样本，经过多次迭代生成更多的高质量代码数据。这使得大模型从这些高质量数据中不断训练优化，从而提升了生成代码的质量。星云大模型的HumanEval@Pass1可以达到83.6分。

2.5 合成数据和自我学习

随着模型规模不断增大，所需的训练数据也更多，最终将会耗尽所有自然产生的数据。因此，基于人工合成数据来训练模型已经成为一个热门研究方向。

借助已有大模型合成精调数据是业界最常用的方式，通过将设计好的指令，并让大模型来获取对应的回答，可以节省大量的人力。WizardLM[27]和WizardCoder[28]根据种子指令分别沿着添加约束、深化、具体化、增加推理步骤、复杂化以及突变等6种演化方向来演化指令，并控制指令的难度和复杂程度，经过多轮不同方向的演化得到大量不同类型的指令，再利用已有大模型生成精调数据，最终取得了不错的精调效果。Magicoder[29]通过在GitHub上随机获取1~15行代码作为种子代码片段，让已有大模型根据提供的种子片段来生成编程问题，这大大丰富了代码精调数据的类型，从而更好地激活大模型的能力。

中兴通讯的星云大模型采用自我学习的方法生成测试用例数据，主要的步骤是通过对同一代码多次生成对应的测试用例，根据编译、覆盖率等情况选出符合要求的高质量数据，再使用这部分高质量数据对模型进行精调训练，从而取得良好效果。在基于HumanEval数据集制作的UTEval数据集上，星云大模型测试用例生成的编译成功率、用例通过率、测试覆盖率指标已经超过GPT4-turbo。

2.6 模型安全性

大模型在提供强大的自然语言处理能力的同时，也带来了安全和隐私方面的挑战。在训练阶段，大模型面临的风险主要源于训练数据。数据可能包含有害内容或设计歧视、侵权、毒性文本。这不仅会影响模型的输出质量，还可能使模型学习并复制这些不当行为。此外，训练数据还可能遭到恶意投毒，即意图故意降低模型的性能或引导模型做出不当行为。

训练阶段引入的常见漏洞包括后门漏洞和数据投毒。针对以上漏洞，可以实施训练语料库治理手段，具体包括：

1）语言识别和解毒：通过自动化工具识别并清除训练

数据中的有毒语言或偏见表达，确保输入数据的质量和安全性。

2）除杂和去标识化：从数据集中移除无关的信息和个人标识信息，减少隐私泄露的风险，并防止模型学习到不必要的个人信息。

上述防御的实现方法可以分为黑盒防御和白盒防御两大类。黑盒防御通常采用基于查询的模型诊断方法，检测模型是否被嵌入了后门中。白盒防御包括通过微调删除后门[32]、模型修剪[33]和通过检查激活来检测后门[34]等方法。例如，Fine-mixing[35]是一种旨在防止在微调模型中出现后门的方法。CUBE防御技术[36]利用了一种称为HDBSCAN的密度聚类算法来准确识别数据集中的簇，来区分包含毒害样本和干净数据的簇。通过利用HDBSCAN的能力，CUBE旨在提供一种有效区分正常数据和有毒数据的方法。

3 模型推理和优化技术

大模型以其强大的理解和生成能力，正在深刻改变我们对人工智能的认知和应用，但其高昂的推理成本也阻碍了技术落地。因此，优化大模型的推理性能成为业界研究的热点。

推理性能优化主要以提高吞吐量和降低时延为目的，关键技术可以划分为：内存管理、算子融合、模型压缩、并行推理、服务调度优化、推理安全及新兴技术。

3.1 内存管理

KV Cache是大模型推理性能优化最常用的技术。该技术在不影响任何计算精度的前提下，通过空间换时间，大幅提升推理性能。Transformer解码器使用自回归产生输出，即每次推理只会预测输出一个token，执行多次后完成全部输出。前后两次的输入只相差一个token，这就存在大量重复计算。KV Cache技术将每个token可复用的$\boldsymbol{K}$和$\boldsymbol{Q}$向量结果保存下来复用，将计算复杂度从$O(n^2)$降低为$O(n)$。

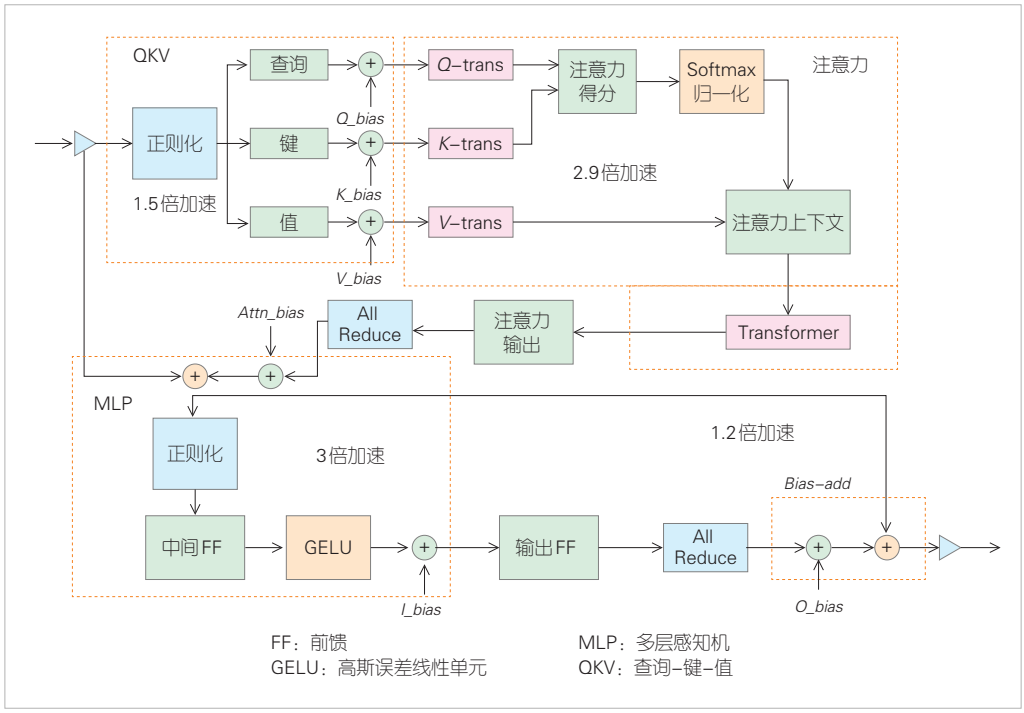
Paged Attention将操作系统中的分页内存管理应用到KV Cache的管理中，这节约了60%～80%的显存，从而支持更大的batch-size，将吞吐率提升了22倍。具体来讲，Paged Attention首先将每个序列的KV Cache分成若干块，每个块包含固定数量token的键和值，然后计算出当前软硬件环境下KV Cache可用的最大空间，并预先申请缓存空间。在推理过程中，通过维护一个逻辑块到物理块的映射表，使多个逻辑块对应一个物理块，并使用引用计数标记物理块被引用的次数，从而实现将地址不连续的物理块串联在一起统一管理。

RadixAttention是一种自动键值缓存重用技术，该技术可以在完成生成请求后不直接丢弃键值缓存，而是在基数树中保留提示和生成结果的键值缓存，从而实现高效的前缀搜索、插入和驱逐。该技术在多轮对话场景可以极大地降低首字时延。

3.2 算子融合

算子融合是深度学习模型推理的一种典型优化技术，旨在通过减少计算过程中的访存次数和统一计算架构（CUDA）Kernel启动耗时，达到提升模型推理性能的目的。

以HuggingFace Transformers库LLaMA-7B模型为例，该模型有30个类型共计2 436个算子，其中aten::slice算子出现频率为388次。大量小算子的执行会降低图形处理器（GPU）利用率，最终影响推理速度。针对Transformer结构



▲图4 Transformer层中的算子融合示意图

特点，算子融合主要分为4类：归一化层和QKV横向融合，自注意力计算融合，残差连接、归一化层、全连接层和激活层融合，偏置加法和残差连接融合。

中兴通讯在vLLM（开源项目名）上实现了针对多查询注意力结构的QKV通用矩阵乘法（GEMM）横向算子融合，以及多层感知机（MLP）中的全连接层+激活融合，性能明显提升，见表1和表2。上述算法的相关代码实现已合入vLLM社区。

由于算子融合一般需要定制化来实现算子CUDA kernel，因此对GPU编程能力要求较高。随着TensorRT、OpenAI Triton、张量虚拟机（TVM）等框架引入编译器技术，算子融合的自动化或半自动化逐步实现，这降低了GPU编程难度，取得了较好的效果。

3.3 模型压缩

模型压缩技术是指在不影响模型精度的情况下，通过缩小模型规模和计算量来提高模型的运行效率。常见的深度学习模型压缩技术包括模型剪枝、知识蒸馏、模型量化和模型分割等。其中，模型量化在这些技术中最具实用性。

SmoothQuant[39]是典型的8 bit LLM量化方法。根据激活量化方式的不同，SmoothQuant提供了3种量化方式：per-tensor static、per-tensor dynamic和per-token dynamic。这3种模型的精度依次提升，计算效率依次降低。SmoothQuant的研究人员观察到，不同的标记在它们的通道上展示出类似的变化，引入了逐通道缩放变换，有效地平滑了幅度，这使得模型更易于量化。激活感知权权重化（AWQ）[38]和生成式预训练Transformer（GPTQ）[37]是典型的仅权重量化的方法，且权重量化的是group粒度。GPTQ提出了一种基于近似二阶信息的新型分层量化技术，使得每个权重的比特宽度减少到3或4位。与未压缩版本相比，该技术几乎没有准确性损失。AWQ[18]的研究人员发现，对于LLM的性能，权重并不是同等重要的，仅保护1%的显著权重可以大大减少量化误差。在此基础上，AWQ采用了激活感知方法，这在处理重要特征时起着关键作用。该方法采用逐通道缩放技术来确定最佳缩放因子，从而在量化所有权重的同时最小化量化误差。

中兴通讯提出了SmoothQuant+[40] 4 bit权重量化训练后量化（PTQ）算法。不同于AWQ对单个层搜索量化参数，SmoothQuant+对整个模型搜索量化参数，并对整个模型进行同一个参数平滑激活，这样能够从模型整体减少量化误差，且搜索效率更高。SmoothQuant+在LLaMA系列模型可以得到比AWQ更好的精度（见表3），同时在性能上也优于AWQ，对应的推理核已开源。

随着大模型上下文长度的增加，KV Cache占用的显存将超过权重和激活，因此对KV Cache进行量化可以显著降低大模型在长上下文推理时资源占用，从而允许系统支撑更多的并发请求数和吞吐率。中兴通讯在生产环境中使用INT4权重量化和KV Cache FP8量化，显存节省了70%，吞吐率提升了2.8倍，推理成本降低75%左右。

3.4 并行推理

当大模型参数量超过单一计算设备所能容纳的上限时，则需要使用分布式并行推理技术。并行推理可以使用模型并行和流水线并行，而模型并行由于可节省显存资源、可降低单用户时延等优势，成为首选的并行方式。

业界最流行的模型并行方案来自Megatron-LM[21]，它的开发者针对Self-Attention和MLP分别设计了简洁高效的模型并行方案。MLP的第1个全连接层为Column Parallel，第2个全连接层为Row Parallel，之后是1次AllReduce规约操作。Self-Attention在计算Query、Key和Value向量时执行Column Parallel（按注意力头个数均分到每个GPU），之后将注意力得分做空间映射时执行Row Parallel，之后是1次AllReduce规约操作。除此之外，LLM模型中的Input Embedding采用

▼表1 StarCoder-15B在A100-40GB上测试查询-键-值融合

批大小V样本数	输入长度/token数	输出长度/token数	注意力基线/s	注意力融合/s	Speedup/%
10	1 024	1 024	27.17	22.80	19
30	1 024	1 024	39.08	37.48	4

▼表2 StarCoder-15B在A100-40GB上测试FC+激活融合

实测的TFLOPs	*B*=1，*M*=1，*K*=6 144	*B*=4，*M*=1，*K*=6 144	*B*=16，*M*=1，*K*=6 144	*B*=64，*M*=1，*K*=6 144	*B*=256，*M*=1，*K*=6 144
基线	0.3	1.2	4.7	17.6	30.3
融合MLP	0.3	1.2	4.9	19.1	47.1
加速率	0.0%	0.0%	4.3%	8.5%	55.4%

FC：全连接层　　MLP：多层感知机
TFLOPs：每秒钟浮点计算万亿次数
注：*B*代表Batchsize；*M*和*K*表示矩阵乘法中的两个维度，*M*恒为1（解码阶段），*K*恒为6 144

▼表3 CodeLLaMA INT4量化在HumanEval上的性能

HumanEval↑	7B/%	13B/%	34B/%
FP16(baseline)	35.98	35.98	51.22
RTN	36.59	33.54	46.34
AWQ	35.98	31.71	50.61
SmoothQuant+	**35.98**	**37.80**	**53.05**

AWQ：激活感知权权重化　　RTN：直接量化
FP16：16 bit原始精度

Row Parallel，Output Embedding采用Column Parallel；Dropout/Layer Norm/Residual Connections等操作都没有做并行拆分。

节点间带宽对模型并行效率有较大影响，高速串行计算机扩展总线标准（PCIE）的理论带宽为32～64 Gbit/s，通常可以满足大模型并行推理需求。模型参数量越大、Batchsize越大，节点间的通信效率就越高，使用模型并行获得的收益越明显。

3.5 服务调度优化

服务调度优化主要考虑的是系统同时为多个用户服务时如何尽可能地提升资源利用率，相关优化主要包括Continuous Batching、Dynamic Batching和异步Tokenize/Detokenize。其中，Continuous Batching和Dynamic Batching主要围绕提高可并发的Batchsize来提高吞吐量，异步Tokenize/Detokenize则通过多线程方式将Tokenize/Detokenize执行与模型推理过程时间交叠，从而实现降低时延目的。

Continuous Batching和Dynamic Batching的思想最早来自文献[41]。Continuous Batching的原理是：将传统batch粒度的任务调度细化为step级别的调度，这解决了不同长短序列无法合并到同一个batch的问题。调度器维护3个队列，分别为Running队列、Waiting队列和Pending队列。队列中的序列状态可以在Running和Waiting/Pending之间转换。在生成每个token后，调度器均会立刻检查所有序列的状态。一旦序列结束，调度器就将该序列由Running队列移除并标记为已完成，同时从Waiting/Pending队列中按先来先服务（FCFS）策略取出一个序列添加至Running队列。

Batching优化技术可有效提升推理吞吐量，目前已在HuggingFace TGI、vLLM、TensorRT-LLM等多个推理框架中实现。

3.6 推理阶段的安全漏洞和防护

推理阶段的安全漏洞不仅可能危及用户的隐私和数据安全，还可能被恶意利用。下文中，我们将详细介绍推理阶段可能遇到的漏洞及其原理，以及如何通过一系列防护措施来提高大模型的安全性。

推理阶段常见的攻击手段包括：1）越狱攻击：通过某些方式使大模型产生退化输出行为，诸如冒犯性输出、违反内容监管输出，或者隐私数据泄漏的输出。2）提示注入攻击：通过注入恶意指令，可以操纵模型的正常输出过程，导致模型产生不适当、有偏见或有害的输出。3）成员推理攻击：通过分辨一条数据是否属于模型的训练集，可以使攻击者获得训练集数据所共有的特征，在训练数据集敏感的应用场景中（例如，生物医学记录和位置跟踪），成功的成员推理攻击会导致严重的隐私泄露和安全威胁。

针对以上漏洞，我们可以实施的防护手段包括：1）越狱攻击防御：基于预处理技术如指令净化、关键词过滤、恶意模型检测等，检测并清理输入或输出中的有害信息，通过语义内容过滤防止大模型生成不受欢迎或不适当的内容，这可以有效减轻潜在的危害。2）提示注入攻击防御：重新设计提示指令是一类预防提示注入攻击的方法，例如re-tokenization[42]、paraphrasing[42]等。re-tokenization是为了打破在提示中注入的恶意指令（如任务忽略文本、特殊字符和虚假响应等）的顺序。paraphrasing可以干扰注入数据的序列，如注入指令及特殊字符插入，减弱提示注入攻击的有效性。还有一些防御方法是基于检测的，侧重于确定给定数据提示的完整性，如困惑度检测就是一种基于提示的检测方法，它向数据提示添加信息或指令。这就会降低质量，并导致困惑度增加。因此，如果数据提示的困惑度超过指定阈值，我们则认为数据提示存在问题。3）成员推理攻击防御：Salem等提出了第一个针对成员推断攻击的有效防御机制[43]，具体方法包括Dropout和模型堆叠。Dropout被定义为随机删除一定比例的神经元连接，可以减轻深度神经网络中的过拟合，这是成员推断攻击实现中的一个重要因素[43]。模型堆叠防御背后的思想是，如果目标模型的不同部分使用不同的数据子集进行训练，那么整体模型有望表现出更低的过拟合倾向。差分隐私[44]也是目前对成员推理攻击最突出的防御手段之一，通过给模型的输出增加噪声，使得攻击者无法根据输出结果在统计上严格区分两个数据集。

3.7 新兴技术

我们将传统优化技术引入大模型推理的同时，也在探索从大模型自回归解码特点出发，通过调整推理执行过程来进一步提升推理性能。并行推测解码作为新兴的推理技术，可以在不损失精度的前提下提高推理速度。

投机采样[45]是一种并行推测解码算法，开创了“小成本生成+大模型验证”的推理技术路线。该算法在已有大模型的基础上，引入一个小模型执行串行解码来提升速度，原大模型执行并行评估采样，保证生成质量，这在保证精度一致性的同时降低了大模型解码的次数，进而提升了推理效率。例如，我们基于HuggingFace Transformers库实现该算法，使用Pythia-6.9B作为基础模型，Pythia-160M作为近似模型，在A100-PCIE-40GB下可以取得3.9倍的推理速度提升。

由于投机采样算法的巨大潜力，有多项工作在其基础上

研究改进。例如，美杜莎头[46]解码无须引入新的模型，而是在原始模型上训练出多个解码头，每个解码头可并行预测多个后续tokens，然后使用基于树状注意力机制并行处理，筛选出合理的后续tokens。前向解码[47]不对原始模型做任何改变，将自回归解码视为求解非线性方程，并采用Jacobi迭代法进行并行解码。

投机采样的推理方式并不适用于所有的应用场景。例如，在文学艺术类的诗词等应用场景，大小模型生成的结果概率分布相差较大；对于代码生成的场景，投机采样比较适合。随着业界研究的深入，投机采样会成为大语言模型推理的必备优化技术。

4 大模型的企业级应用

相较于其他通用技术，AI技术正在以历史罕见的速度高速发展，与人类水平相当的AGI可能在未来10年内出现，相比于此前预计的30～50年要大幅提前。人类社会正在从信息时代加速走向智能时代，旧的商业竞争格局逐步解构，新的机遇不断产生。夯实数字化、加速智能化，以日益强大的人工智能技术强化产品竞争力，提升企业运营效率，将是每个企业把握智能化时代先机重大而紧迫的任务。

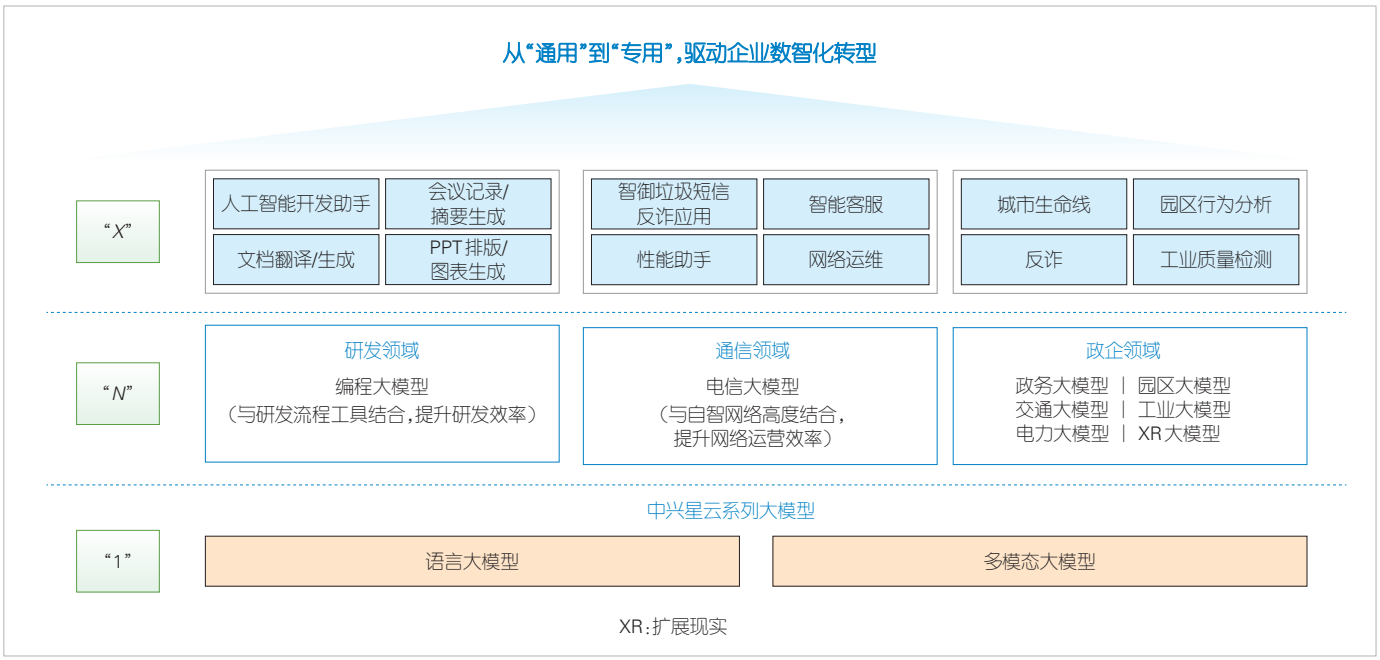
然而，在企业中用好大模型并非易事。首先，大模型技术本身成本高昂，若非有对应的高价值场景，则大模型的应用也难以为继。其次，大模型技术门槛较高，对企业自身的数据治理水平、算力规模、团队能力都有较高的要求。最后，大模型本身仍在快速发展之中，企业应用规划必须具备足够的前瞻性和预测性，才能把握好应用节奏，做到既不掉队，也不过度投资导致大量浪费。

我们认为，大模型的企业级应用包含场景、工程、算法3要素。首先是场景的甄别、价值排序和规划，从价值、技术可行性、资源投入等维度确定“主战场”。然后要有效应对大模型在工程上的复杂性，将算力基础设施、软件平台、框架、工具、能力整合为大模型技术平台，提升开发效率和降低部署成本。最后，紧跟算法发展趋势，将拿来主义和自主创新结合起来，不断推出更强大的模型，赋能各个应用。

4.1 大模型规划

企业需要根据自身的业务特点，梳理出大模型应用的总体规划。结合前文所述的预训练+精调的训练范式，企业适合采用基础模型和领域模型的分层规划。基础模型为预训练大模型，提供通用能力。在其基础上，通过后预训练、精调等技术手段开发出面向特定应用场景的领域模型应运而生。最后，基于这些领域模型面向各场景打造了不同应用。这种分层规划架构，已被多个企业所采用。

以中兴通讯为例，大模型整体规划可以描述为1+*N*+*X*（具体如图5所示）：即1系列基础大模型，确保自主可控和数据资产安全；*N*个领域大模型，通过加入领域KnowHow增量预训练、精调等方式，提高专业性能力；*X*个场景应用，利用大模型，开发出各种场景的应用。



▲图5 中兴通讯星云系列大模型

基础模型的规划以技术为导向。星云系列大模型拥有十亿到千亿不同规模，分别匹配中心云、边缘云、终端部署不同场景下的算力及资源条件；支持图像、表格、文字、代码等多模态的输入输出，可支持大部分企业应用场景；具备前文所述的模型推理优化技术，在保持模型准确率的前提下，可降低70%以上的推理资源需求。

领域模型的规划以价值为导向。例如，研发大模型主要用于企业内部数万名研发人员的研发提效，在文档、代码、测试用例生成等方面能够显著提升效率。目前，中兴通讯3%的代码由AI生成，2024年底预计提升至10%～20%。

这种规划可以将技术和价值紧密结合。基于领域模型打造的应用源源不断提供新的业务和用户数据，这些数据作为训练语料进一步增强领域模型的场景化服务能力。同时，基础模型提供的通用能力则有助于提升领域模型的整体表现。

4.2 大模型应用架构

大模型的研发、部署及应用是极其复杂的工程。合理的架构允许技术组件的解耦和复用，能够有效控制工程的复杂度，因此对大模型应用的降本增效极其重要。

中兴通讯在实践中不断优化迭代的大模型应用架构具体如图6所示。

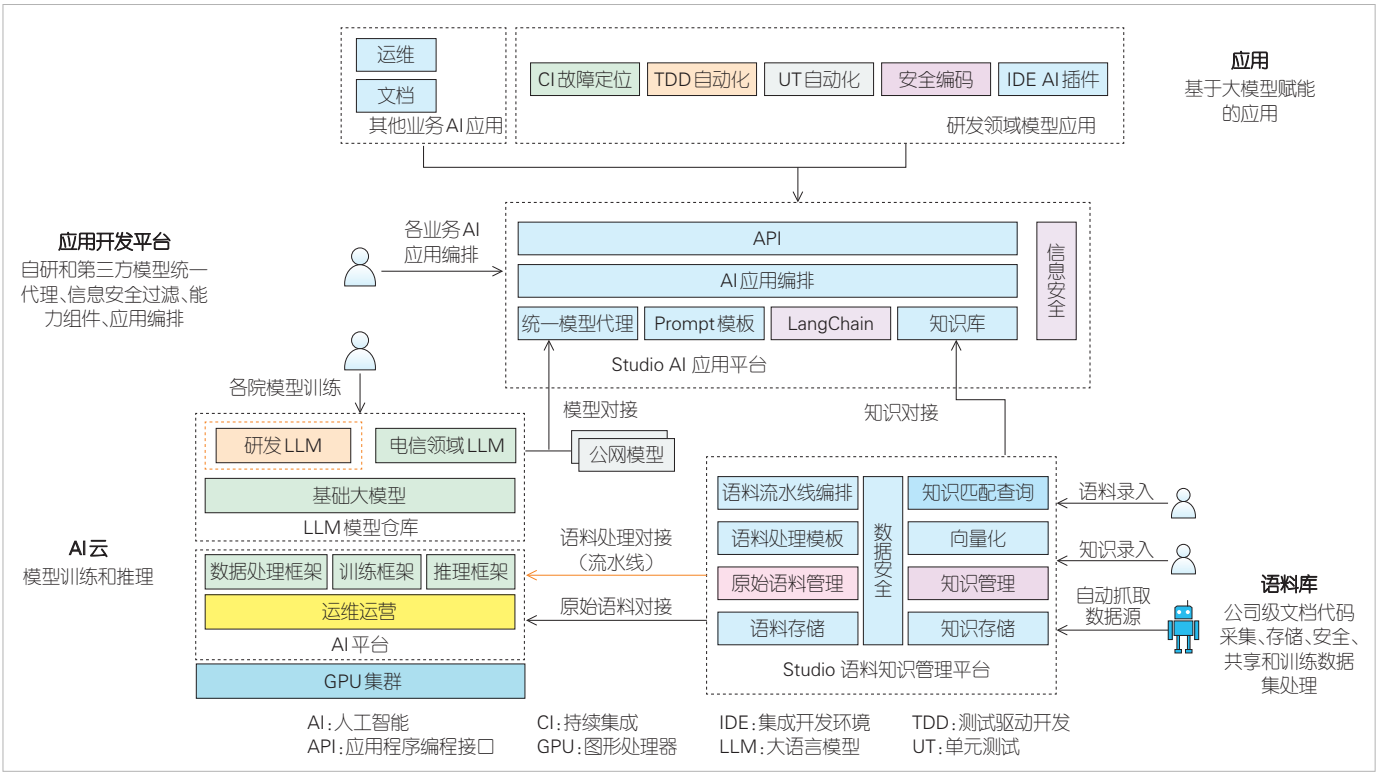
语料库是基于中兴通讯数据智能分析平台（DAIP）大数据平台构建的，能够提供PB级数据存储和处理能力。其主要具备3个功能：1）原始数据采集、存储、处理和共享，实现了互联网数据和公司内部数据的统一；2）基于Spark的训练语料处理，将原始数据转化为可供大模型训练使用的语料；3）基于EBase向量数据库的知识库，将原始语料向量化后存储到数据库中，作为Agent外挂知识库。

AI云基于AiCloud构建，用于管理公司大规模异地、异构算力集群（英伟达、壁仞等），通过算子优化和通信优化，大幅提升基础设施利用效率，同时还提供数据处理、训练、精调、评估、优化端到端的工作流管理，支持高效训练和部署千亿参数以上级别的大模型。前文所述所有的基础大模型、领域大模型都在AI云上训练和部署。

应用开发平台可以实现基础能力组件的集成和编排，实现无代码方式构建Agent。大模型应用平台有超1 000个大模型应用，每日活跃用户万人以上。此外，平台还可以支持模型统一代理、信息安全过滤、Prompt共享等主要功能，这些功能对于提升用户使用体验、保障公司信息安全起到关键作用。

4.3 大模型应用技术

众所周知，大模型的幻觉问题短时间难以解决，因此保



▲图6 中兴通讯企业级大模型应用架构

证大模型在应用中的正确性和可靠性成为一个技术难题。此外，大模型还有知识难以更新、经常无法正确处理数学逻辑推理等问题，这些都制约了大模型的应用。

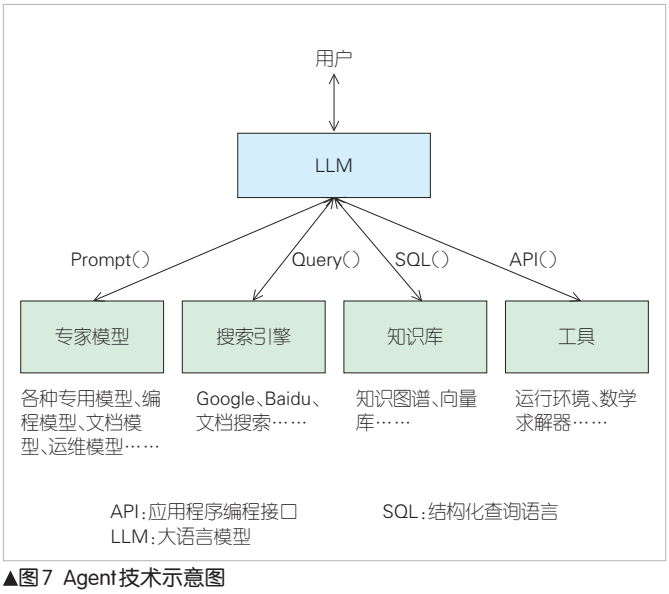
业界逐渐认识到使用单一大模型无法解决上述所有问题，于是尝试使用多个辅助模型、外部知识库、搜索引擎、专业工具等多种手段协同解决大模型实用问题，这些手段最终形成了Agent技术。目前，业界普遍认为Agent是大模型应用技术的发展方向。

4.3.1 Agent技术

Agent并非单一技术，而是一个框架（如图7所示），将大模型与专家模型、搜索引擎、知识库、工具等众多组件集成在一起，通过组件之间的协作，共同完成用户指定的任务。

大模型作为Agent核心组件，需要理解用户意图、拆分任务、流程控制、汇总信息，并生成结果后返回给客户。例如，对于专业性问题，如代码生成，大模型可以根据用户请求生成Prompt与代码模型的交互，并将代码模型生成结果反馈给用户；对于信息查询类问题，大模型根据用户提出的问题转化为Query，并浏览搜索引擎返回的页面链接，再将相关页面的信息总结提炼后返回给客户；对于数学计算等问题，则可以将问题转化为应用程序编程接口（API），调用外部工具完成问题求解。

相对于传统的软件，Agent技术的本质是在控制面用大模型替代固定的程序，从而可以处理新的场景，极大地提升系统灵活性。Agent是智能化时代的软件，是软件发展的下一形态。



▲图7 Agent技术示意图

4.3.2 RAG

在对于需要精确、实时、涉及领域专业知识的任务中，大模型所面临的幻觉频出、信息过时、专业领域深度知识缺乏以及推理能力弱等痛点，成为亟待解决的问题。

为应对上述挑战，RAG技术应时而生，它是Agent的一项关键技术。其核心思想是在生成响应之前，通过信息检索的技术，先从外部数据库中检索出和用户问题相关的信息，然后结合这些信息，LLM生成结果。RAG技术主要包括3个基本步骤：1）索引。索引阶段是构建RAG的准备步骤，类似于ML中的数据清理和预处理步骤。通常，索引阶段包括：收集数据、数据分块、向量嵌入和向量数据库存储。2）检索。检索阶段是构建RAG的主要步骤，由查询向量嵌入和相似度检索组成。3）生成。生成阶段由提示工程构成。

RAG是一项前景广阔的新兴技术，有效提升了大语言模型的生成内容准确率和时效性。RAG技术正在快速发展，不断出现更优秀的应用，如ChatPDF、Lepton Search等应用，RAG为未来的通用人工智能提供更大的可能性。

4.3.3 插件技术

插件技术允许大模型与外部应用协作，例如OpenAI的ChatGPT有数千个插件，从而大幅扩展了大模型的能力和应用领域。

CodeInterpreter使得数据分析变得更加简单，通过与用户的对话就可以处理庞大的数据。CodeInterpreter在本质上是将大语言模型生成的代码放到一个安全的环境中执行，这个环境通常被称为沙盒。在这个沙盒中，开发者可以提前配置所需的依赖库和环境变量，以确保代码能够正常运行。CodeInterpreter的主要功能是执行由大语言模型生成的代码，并返回结果。

Function Calling可以为B端用户开发的APP提供强大的功能，改变交互方式，从图形用户界面（GUI）向自然用户界面（NUI）、语音用户界面（VUI）转变，让客户有更自然的体验。Function Calling的关键技术点是大模型的函数选择能力、函数调用能力、结果解释能力、异常处理能力。

Threading则提供了持久保存且无限长的上下文，帮助用户建立起更全面的客户画像，让客户使用APP时更加得心应手。Threading的关键技术点主要包括更长的上下文理解能力以及向量数据库中检索、总结能力。

中兴通讯基于星云大模型成功实现了上述插件。未来，星云大模型将朝着能力更强、上下文更长、使用体验更好的方向发展，使大模型能够为各行各业赋能。

5 结束语

随着ChatGPT热度的逐渐褪去，对大模型的投资也逐渐趋于理性。大模型如何产生真正的商业价值成为全行业都在思考、探索的问题。一方面，随着大模型规模的不断增加，模型能力在提升的同时，算力成本也在不断飙升，这给大模型长期可持续发展带来了不确定性，因此以实现更低成本算力和更高效率算法为目标的核心技术亟待突破；另一方面，以Agent为代表的应用层技术在解决大模型固有问题并大幅拓展应用边界的潜力还未被完全发掘，业界对Agent的关注度持续上升。大模型机遇与挑战并存，加速发展的趋势在中长期不会改变。

参考文献

[1] VASWANI A, SHAZEER N, PARMAR N, et al. Attention is all you need [C]//Proceedings of the 31st International Conference on Neural Information Processing Systems. ACM, 2017: 6000–6010. DOI: 10.5555/3295222.3295349

[2] HOCHREITER S, SCHMIDHUBER J. Long short–term memory [J]. Neural computation, 1997, 9(8): 1735–1780. DOI: 10.1162/neco.1997.9.8.1735

[3] RADFORD A, NARASIMHAN K. Improving language understanding by generative pre–training [EB/OL]. [2024–03–10]. https://cdn. openai. com/research–covers/language–unsupervised/language_understanding_paper.pdf

[4] DEVLIN J, CHANG M W, LEE K, et al. BERT: pre–training of deep bidirectional transformers for language understanding [EB/OL]. (2018–10–11)[2024–03–10]. https://arxiv.org/abs/1810.04805

[5] BROWN T B, MANN B, RYDER N, et al. Language models are few–shot learners [EB/OL]. (2020–05–28) [2024–03–10]. https://arxiv.org/abs/2005.14165

[6] TOUVRON H, LAVRIL T, IZACARD G, et al. Llama: open and efficient foundation language models [EB/OL]. [2024–03–05]. https://arxiv.org/abs/2302.13971

[7] GU A, DAO T. Mamba: linear–time sequence modeling with selective state spaces [EB/OL]. [2024–03–01]. https://arxiv. org/abs/2312.00752

[8] TOUVRON H, MARTIN L, STONE K, et all. Llama 2: open foundation and fine–tuned chat models [EB/OL]. (2023–07–18) [2024–03–10]. https://arxiv.org/abs/2307.09288

[9] DAO T. FlashAttention–2: faster attention with better parallelism and work partitioning [EB/OL]. (2023–07–17)[2024–03–10]. https://arxiv.org/abs/2307.08691

[10] BAI J, BAI S, CHU Y F, et al. Qwen technical report [EB/OL]. (2023–09–28)[2024–03–11]. https://arxiv.org/abs/2309.16609

[11] XIONG W H, LIU J Y, MOLYBOG I, et al. Effective long–context scaling of foundation models [EB/OL]. (2023–09–27) [2024–03–12]. https://arxiv.org/abs/2309.16039

[12] JIANG A Q, SABLAYROLLES A, MENSCH A, et al. Mistral 7B [EB/OL].[2024–03–12]. https://arxiv.org/abs/2310.06825

[13] RADFORD A, KIM W J, HALLACY C, et al. Learning transferable visual models from natural language supervisior [EB/OL]. (2021–02–26)[2014–03–05]. https://arxiv.org/abs/2103.00020

[14] LI J, LI D, SAVARESE S, et al. Blip–2: Bootstrap–ping language–image pre–training with frozen image encoders and large language models [EB/OL]. (2023–01–30) [2024–03–12]. https://arxiv.org/abs/2301.12597

[15] LIU H, LI C Y, WU Q Y, et al. Visual instruction tuning [EB/OL]. [2024–03–10]. https://arxiv.org/abs/2304.08485

[16] DONG X Y, ZHANG P, ZANG Y H, et al. InternLM–XComposer2: mastering free–form text–image composition and comprehension in vision–language large model [EB/OL]. (2024–01–29)[2024–03–10]. https://arxiv.org/abs/2401.16420

[17] Huggingface models [EB/OL].[2024–03–10]. https://huggingface.co/01–ai/Yi–VL–34B

[18] ROMBACH R, BLATTMANN A, LORENZ D, et al. High–resolution image synthesis with latent diffusion models [EB/OL]. (2021–12–20)[2022–04–13]. https://arxiv.org/abs/2112.10752

[19] PEEBLES W, XIE S N. Scalable diffusion models with transformers [C]//Proceedings of IEEE/CVF International Conference on Computer Vision (ICCV). IEEE, 2023: 4195–4205. DOI: 10.1109/iccv51070.2023.00387

[20] OpenAI. Video generation models as world simulators [EB/OL]. [2024–03–13]. https://openai. com/research/video–generation–models–as–world–simulators

[21] SHOEYBI M, PATWARY M, PURI R, et al. Megatron–LM: training multi–billion parameter language models using model parallelism [EB/OL]. [2024–03–10]. https://arxiv. org/abs/1909.08053

[22] RAJBHANDARI S, RASLEY J, RUWASE O, et al. ZeRO: memory optimizations toward training trillion parameter models [C]// Proceedings of SC20: International Conference for High Performance Computing, Networking, Storage and Analysis. IEEE, 2020: 1–16. DOI: 10.1109/sc41405.2020.00024

[23] CHEN T, XU B, ZHANG C Y, et al. Training deep nets with sublinear memory cost [EB/OL]. [2024–03–10]. https://arxiv. org/abs/1604.06174

[24] HU E J, SHEN Y Y, WALLIS P, et al. Lora: Low–rank adaptation of large language models [EB/OL]. (2021–06–17) [2024–03–10]. https://arxiv.org/abs/2106.09685

[25] TOUVRON H, MARTING L, STONE K, et al. Llama 2: open foundation and fine–tuned chat models [EB/OL]. (2023–07–18) [2024–03–05]. https://arxiv.org/abs/2307.09288

[26] RAFAILOV R, SHARMA A, MITCHELL E, et al. Direct preference optimization: your language model is secretly a reward model [EB/OL]. [2024–03–10]. https://arxiv.org/abs/2305.18290

[27] XU C, SUN Q, ZHENG K, et al. Wizardlm: empowering large language models to follow complex instructions [EB/OL]. (2023–04–24)[2024–03–10]. https://arxiv.org/abs/2304.12244

[28] LUO Z, XU C, ZHAO P, et al. WizardCoder: empowering code large language models with evol–instruct [EB/OL]. [2024–03–10]. https://arxiv.org/abs/2306.08568

[29] WEI Y X, WANG Z, LIU J, et al. Magicoder: source code is all you need [EB/OL]. (2023–12–04)[2024–03–10]. https://arxiv.org/abs/2312.02120

[30] TUFANO M, DRAIN D, SVYATKOVSKIY A, et al. Unit test case generation with transformers and focal context [EB/OL]. (2020–09–11)[2024–03–10]. https://arxiv.org/abs/2009.05617

[31] STEENHOEK B, TUFANO M, SUNDARESAN N, et al. Reinforcement learning from automatic feedback for high–quality unit test generation [EB/OL]. (2023–10–03) [2024–03–09]. https://arxiv.org/abs/2310.02368

[32] SHA Z Y, HE X L, BERRANG P, et al. Fine–tuning is all you need to mitigate backdoor attacks [EB/OL]. (2022–12–18) [2024–03–10]. https://arxiv.org/abs/2212.09067

[33] LIU K, DOLAN–GAVITT B, GARG S. Fine–pruning: defending against backdooring attacks on deep neural networks [EB/OL]. (2018–05–30)[2024–03–10]. https://arxiv.org/abs/1805.12185

[34] CHEN B, CARVALHO W, BARACALDO N, et al. Detecting backdoor attacks on deep neural networks by activation clustering [EB/OL]. (2018–05–30)[2024–04–10]. https://arxiv.org/

abs/1805.12185
[35] ZHANG Z Y, LYU L, MA X J, et al. Fine-mixing: mitigating backdoors in fine-tuned language models [EB/OL]. (2018-05-30)[2024-04-10]. https://arxiv.org/abs/1805.12185
[36] CUI G, YUAN L, HE B, et al. A unified evaluation of textual backdoor learning: frameworks and benchmarks [J]. Advances in neural information processing systems, 2022, 35: 5009-5023
[37] FRANTAR E, ASHKBOOS S, HOEFLER T, et al. GPTQ: accurate post-training quantization for generative pre-trained transformers [EB/OL]. (2022-10-31)[2023-05-22]. https://arxiv.org/abs/2210.17323
[38] LIN J, TANG J, TANG H T, et al. AWQ: activation-aware weight quantization for LLM compression and acceleration [EB/OL]. (2023-05-01)[2023-10-03]. https://arxiv.org/abs/2306.00978
[39] XIAO G X, LIN J, SEZNEC M, et al. SmoothQuant: accurate and efficient post-training quantization for large language models [EB/OL]. (2022-11-18) [2024-02-20]. https://arxiv. org/abs/2211.10438
[40] PAN J Y, WANG C C, ZHENG K F, et al. SmoothQuant+ : accurate and efficient 4-bit post-training weightQuantization for LLM [EB/OL]. (2023-12-06)[2024-02-20]. https://arxiv.org/abs/2312.03788
[41] YU G I, JEONG J S. Orca: A distributed serving system for transformer-based generative models [EB/OL]. [2024-02-20]. https://www.usenix.org/conference/osdi22/presentation/yu
[42] JAIN N, SCHWARTZSCHILD A, WEN Y X, et al. Baseline defenses for adversarial attacks against aligned language models [EB/OL]. [2024-02-22]. https://arxiv.org/abs/2309.00614
[43] SALEM A, ZHANG Y, HUMBERT M, et al. ML-leaks: model and data independent membership inference attacks and defenses on machine learning models [EB/OL].[2024-03-12]. https://arxiv.org/abs/1806.01246
[44] ABADI M, CHU A, GOODFELLOW I, et al. Deep learning with differential privacy [C]//Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security. ACM, 2016: 308-318. DOI: 10.1145/2976749.2978318
[45] KEVUATGAB Y, KALMAN M, MATIAS Y. Fast inference from transformers via speculative decoding [EB/OL]. (2022-11-30) [2024-03-10]. https://arxiv.org/abs/2211.17192
[46] CAI T, LI Y H, GENG Z Y, et al. Medusa: simple framework for accelerating LLM generation with multiple decoding heads [EB/OL]. (2023-01-19) [2024-03-10]. https://arxiv. org/abs/2401.10774
[47] FU Y C, BAILIS P, STOCA P, et al. Breaking the sequential dependency of LLM inference using lookahead decoding [EB/OL]. (2024-02-03) [2024-02-10]. https://arxiv. org/abs/2402.02057

作 者 简 介

韩炳涛，中兴通讯股份有限公司AI首席专家、移动网络和移动多媒体技术国家重点实验室多媒体研究中心副主任、Linux深度学习基金会Adlik项目负责人；研究方向为机器学习平台技术和网络智能化，以及相关核心系统架和AI算法；拥有发明专利多项，出版专著多部。

刘涛，中兴通讯股份有限公司资深算法专家、Adlik开源项目首席架构师、AI预研项目经理；主要研究领域为AI模型并行训练、模型推理优化、高性能计算、异构硬件模型部署等；拥有多项发明专利。

反无人机技术综述：通信技术与人工智能的融合

Overview of Anti-Drone Technology: Integration of Communication Technology and Artificial Intelligence

邱宝华/QIU Baohua
（中国移动通信集团广西有限公司，中国 南宁 530028）
(China Mobile Group Guangxi Company,Nanning 530028, China)

DOI：10.12142/ZTETJ.202402013
网络出版地址：http://kns.cnki.net/kcms/detail/34.1228.TN.20240418.1204.002.html
网络出版日期：2024-04-19
收稿日期：2024-02-18

摘要：对当前反无人机技术的最新进展进行了全面的梳理，内容涵盖了从主动反制策略到被动检测技术的广泛领域。首先，审视了主动干预技术，如电磁干扰、激光系统和无线电频率干扰，以及被动监测技术，包括雷达、光学和热成像等。随后，深入探讨了现代通信技术在反无人机系统中的应用，重点关注了高速数据传输、信号覆盖、高精度定位、动态频率切换、宽频带和多输入多输出（MIMO）技术等方面的支持作用。此外，还详细分析了人工智能技术在提高反无人机效能方面的研究成果和应用算法，指出了目前的成就以及未来发展的潜在方向。最后，展望了反无人机技术的未来发展趋势，包括自主学习、对抗博弈和多智能体协同等方面，并对这些新兴趋势可能带来的挑战及其解决方案进行了前瞻性讨论。本研究可为反无人机技术的未来发展提供一个全面的理论框架参考。

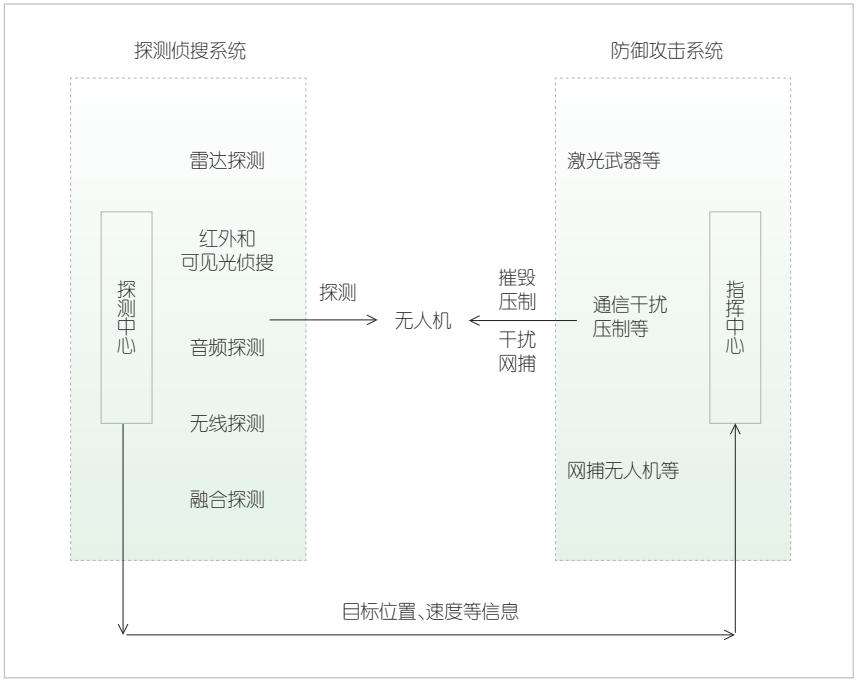
关键词：反无人机技术；智能无人系统；通信技术；人工智能；自主学习；对抗博弈；多智能体协同

Abstract: The latest advancements in anti-drone technology are analyzed, encompassing a wide range of areas from active countermeasures to passive detection methods. The active intervention technologies are discussed including electromagnetic interference, laser systems, and radio frequency jamming, as well as passive surveillance methods like radar, optical, and thermal imaging. Subsequently, the application of modern communication technologies in supporting anti-drone systems is deeply explored, especially high-speed data transmission, signal coverage, high-precision positioning, dynamic frequency switching, broadband, and multiple-input multiple-output(MIMO) technologies. Moreover, the research outcomes and applied algorithms of artificial intelligence in enhancing the efficiency of anti-drone technologies are meticulously analyzed, and their current achievements and potential directions for future development are highlighted. Finally, the future trends in anti-drone technology is forecasted, including autonomous learning, adversarial gaming, and multi-agent collaboration, and discusses the challenges these emerging trends may face along with their solutions. This study can provide a holistic framework for the future development of anti-drone technology.

Keywords: anti-drone technology; intelligent unmanned system; communication technology; artificial intelligence; autonomous learning; adversarial gaming; multi-agent collaboration

引用格式：邱宝华. 反无人机技术综述: 通信技术与人工智能的融合 [J]. 中兴通讯技术, 2024, 30(2): 89-99. DOI: 10.12142/ZTETJ.202402013
Citation：QIU B H. Overview of anti-drone technology: integration of communication technology and artificial intelligence [J]. ZTE technology journal, 2024, 30(2): 89-99. DOI: 10.12142/ZTETJ.202402013

近年来，智能无人系统[1-2]的发展日新月异。在俄乌冲突这一现代战争中，双方都大规模使用了无人机执行多样化任务，包括侦查、打击和情报搜集等。无人机在战场上发挥重要作用，甚至改变了传统的战略和战术。因此，反无人机技术的研究和发展变得尤为重要[3-5]。无人机技术的进步得益于通信技术和人工智能的迅猛发展，而反无人机技术的演进同样依赖于这些关键技术的前进。反无人机技术已经从基础的物理干扰手段，发展到了更为高级的电子和网络反制策略。主动反制技术，包括电磁干扰、激光打击和无线电频率干扰等，能够直接干扰无人机的操作。与此同时，被动检测技术，如雷达系统、光学摄像头和热成像技术，能够在不干预无人机飞行的前提下，监测和识别空中的无人机活动。图1展示了一套当前广泛采用的反无人机系统架构。



▲图1 反无人机系统示意图

1 反无人机技术的现状

为了有效应对无人机技术的挑战，研究人员已经开发了多种反无人机技术。这些技术主要分为两大类：主动反制技术和被动检测技术[6]。

1.1 主动反制技术

主动反制技术通过主动干扰无人机的通信和导航系统，阻止其正常运作或使其被迫着陆。该技术具体包括电磁干扰、激光干扰和无线电频率干扰。

电磁干扰：通过向无人机发送强大的电磁信号，干扰其通信链路，使其无法接收指令或传输数据。电磁干扰可以是广播型、定向型或高能脉冲型，具体应用视情况而定[7-8]。

激光干扰：利用激光束照射无人机，干扰其传感器和摄像头，使图像捕捉和目标识别变得困难[9]。

无线电频率干扰：通过发射与无人机遥控器相同频率的信号，可以迫使无人机返回或着陆，或者使其无法接收操作者的指令[10]。

主动反制技术的优势在于它们能够积极应对无人机的威胁，干扰无人机运行并迫使其离开或着陆。然而，这一技术也存在一些局限性，例如：对操作者和附近设备的潜在干扰，以及对无人机通信和导航系统的依赖。

1.2 被动检测技术

被动检测技术依赖传感器和监测系统，用于检测和识别无人机。相关技术包括雷达系统、光学摄像头和热成像。

雷达系统：利用无线电波来探测无人机的位置和速度，提供关于无人机的精确信息[11]。

光学摄像：使用可见光或红外光来捕捉无人机的图像，有助于监测人员对无人机进行视觉识别[12]。

热成像：利用目标的红外辐射来检测其热量分布，对于在夜间或恶劣天气条件下识别无人机非常有用[13]。

被动检测技术的优势在于它们不会干扰无人机的通信和导航系统，因为它们依赖于传感器和监测设备。然而，它们也有一些限制，如有限的探测范围和受天气条件的影响。

现代通信技术和人工智能在推动无人机技术发展的同时，也在不断促进反无人机技术的发展。在接下来的章节中，我们将探讨通信技术和人工智能如何在反无人机系统中起着核心作用。

2 通信技术在反无人机技术中的关键作用

智能无人系统使得无人机在某些特定任务和情境下可以实现一定程度的自治操作。但对于更复杂的任务和决策来说，即使无人机具有高级的人工智能，也可能需要人的干预。因此，通信技术在反无人机技术中起着关键的作用[14]。

1）高速数据传输

高速数据传输在反无人机技术中的应用是多方面的，主要包括以下几个方面：

（1）实时数据传输：在反无人机系统中，实时性是关键。5G 和 6G 等高速通信技术能够确保雷达数据、光学图像、红外扫描等信息以极低延迟传输到控制中心[15]。这样，操作员可以迅速对无人机的动作做出反应，如改变方向、调整高度等，从而提高拦截的成功率。

（2）多源数据融合：反无人机系统通常需要从多个探测设备收集数据，包括雷达、声纳、摄像头和其他传感器。高速数据传输能力使得这些来自不同源的数据可以迅速合并，以便进行综合分析[6]。这种多源数据融合技术有助于更准确地确定无人机的定位、轨迹和意图，从而为拦截策略

提供支持。

（3）快速决策支持：通过高速数据传输，反无人机系统可以将收集到的数据快速进行分析，以便操作员能够迅速做出决策。这种快速决策支持有助于系统及时识别无人机的类型，评估潜在威胁并制定最佳的拦截策略。

（4）增强的情报共享：高速数据传输还允许反无人机系统与其他安全机构或部队共享情报。这种情报共享可以提高对无人机威胁的整体应对能力，同时也能够在更大的范围内协调拦截行动。

（5）支持高级应用程序：随着技术的发展，反无人机系统将能够支持更多高级应用程序，如机器学习算法、人工智能等。这些高级应用程序需要大量的数据处理和分析，而高速数据传输提供了必要的数据处理速度和带宽。

总之，高速数据传输在反无人机技术中的作用是至关重要的。它不仅提高了系统的实时响应能力，还增强了数据融合、决策支持和情报共享的能力。随着通信技术的不断进步，反无人机系统将变得更加高效和准确。

2）增强的信号覆盖和稳定性

在过去，有限的通信范围可能会限制反无人机系统在某些偏远或具有挑战性的环境中的使用，例如山区、森林或城市深处等。但是，随着信号覆盖的扩展，这些地区不再是“死角”，反无人机系统可以轻松地部署和操作。除了扩展的操作范围，增强的信号稳定性也意味着反无人机系统在各种复杂环境中都可以保持高水平的性能。无论是在高楼大厦之间、大风雨中，还是在其他各种干扰比较强的环境中，现代通信技术都能够确保反无人机系统获得比较稳定、准确的数据流，使其在各种条件下都能有效地执行任务。

增强的信号覆盖和稳定性不仅为反无人机系统提供了更大的工作范围，而且大大提高了对潜在威胁的响应速度和精确度[7]。此外，稳定的通信链接还可以支持远程操作和实时策略调整。这意味着即使在最具挑战性的环境中，操作员仍然可以保持与系统的通信，根据实时数据调整拦截策略或部署其他资源。总之，现代通信技术的信号覆盖和稳定性为反无人机系统提供了更高的灵活性、反应速度和战术效果。

3）高精度的定位技术

高精度定位技术已成为现代通信技术的关键组成部分，为众多行业带来了前所未有的优势。在反无人机技术领域，这一进展尤为关键。通过集成卫星导航系统，如全球定位系统（GPS）和北斗等，反无人机系统的定位精度得到了显著增强[8]。这些先进的导航系统不仅为反无人机系统提供了持续性和稳定性的定位服务，而且还通过多卫星数据融合技术，确保了定位数据的准确性和可靠性。

在复杂的地形环境中，例如城市高楼大厦、山区等地，高精度定位技术能够帮助反无人机系统快速识别并精确锁定目标位置，进而实施有效的干预措施。此外，高精度定位技术还可以与其他传感器和侦测设备，如雷达、红外摄像机等，协同工作，形成一个综合的监控网络。这种多元化的数据输入不仅扩大了系统的探测范围，还提升了对于目标无人机行为分析和预测的能力。

通过对这些数据进行实时分析，反无人机系统能够预判无人机的未来动作，并据此提前部署相应的策略，如电磁干扰、物理拦截等，从而提升拦截的成功率。可以预见，随着技术的不断进步，高精度定位技术将在未来反无人机战术中扮演更加重要的角色。

4）动态频率切换与干扰

动态频率切换是现代通信技术的一项关键进步，它使得设备能够在不同频率间迅速切换，以确保通信的稳定性与安全性[9]。在反无人机技术领域，这一能力成为了一种强大的工具，用于干扰和中断敌方无人机的通信链路。利用动态频率切换技术，反无人机系统能够快速检测并识别目标无人机所使用的通信频率，并立即实施干扰措施。

随着无人机技术的发展，许多无人机已经具备了自动切换备用频率的能力，以应对可能的干扰。然而，动态频率切换技术能够使反无人机系统实时追踪并干扰这些频率的变化，以确保无人机始终处于失联状态。

此外，现代通信技术还提供了更为复杂和多样化的干扰策略。除了传统的信号阻断方法外，反无人机系统还可以发送伪造的控制指令，诱使无人机执行错误的动作，如降落或偏离航线。这种“智能干扰”策略不仅提高了拦截的成功率，还显著降低了误伤无辜设备的风险。

展望未来，我们可以预期动态频率切换与干扰技术将在反无人机战略中扮演更加关键的角色，为防御无人机系统提供更加有效和灵活的手段。

5）宽频带技术

宽频带技术的引入为反无人机技术带来了革命性的变化。与传统的通信技术相比，宽频带技术能够在更广泛的频率范围内进行工作，从而消除了在检测和干扰无人机时的盲区。宽频带技术的应用使得反无人机系统能够同时监视多个频段，从而允许操作员更加准确地确定哪些频率正在被潜在威胁的无人机所使用，并针对性地实施干扰措施[10]。

这种技术不仅显著提高了干扰的成功率，还减少了对其他非目标通信设备的不必要干扰，降低了误伤的风险。

此外，宽频带技术在复杂电磁环境中的适应性也是其另一个重要优势。在现代城市环境中，众多通信设备、信号塔和其他电子设备产生的电磁干扰使得电磁环境极为复杂。宽频带技术使得反无人机系统能够在这样的环境中迅速识别并响应潜在威胁的无人机信号。

宽频带技术的广泛频率覆盖范围、精确的干扰能力以及对复杂电磁环境的适应性，都极大地提升了反无人机技术的能力。这些技术进步为反无人机战略提供了更加有效和灵活的手段，以应对不断演进的无人机威胁。

6）多输入多输出（MIMO）技术

MIMO技术是通信领域近年来的重大突破。它通过使用多个发射天线和接收天线同时传输和接收数据，显著提高了数据传输的速度和可靠性[16]。在反无人机技术领域，MIMO技术也发挥着至关重要的作用，为无人机的检测、跟踪和拦截提供了强有力的技术支持。

MIMO技术的多天线特性使得系统能够从不同角度和位置接收到无人机的信号，并可以利用这些信号差异可以精确计算出无人机的位置。这种方法不仅极大地提高了目标检测的准确性，还使得对无人机的跟踪更加稳定和可靠。此外，当无人机尝试通过改变频率或使用干扰器来规避检测时，MIMO系统的多频道特性能够迅速适应并重新锁定目标。

此外，MIMO技术增强了反无人机系统的信号干扰能力。与传统的单一输入输出系统相比，MIMO技术允许系统同时对多个目标发射干扰信号。这意味着在多架无人机同时出现的情况下，系统仍然能够有效地进行拦截。由于MIMO技术能够提供更强大的信号输出，其干扰效果也更为显著，从而大大增加了无人机被成功拦截的概率。

总的来说，无论是提高定位准确性、增强跟踪稳定性，还是提升干扰效果，MIMO技术都为反无人机技术的发展提供了强有力的技术支撑。

3 人工智能在反无人机技术中的关键角色

人工智能技术在无人机的发展中扮演着至关重要的角色，这一作用在反无人机技术领域同样显著。随着无人机技术的发展，飞行器具备了微小尺寸、高速机动性，以及可能采用的隐蔽或低空飞行轨迹等特性。这些特性使得传统的人工监视和控制手段面临着重大挑战。然而，人工智能技术的应用为这些挑战提供了有效的解决方案。

3.1 人工智能为何重要

在反无人机技术领域，人工智能的重要性体现在以下3个方面：

1）高速数据处理与实时决策

无人机的快速移动和短时间内执行复杂任务的能力要求反无人机系统能够实时跟进和应对。人工智能系统能够快速处理来自雷达、相机和其他传感器的庞大数据流，并立即作出响应决策，如自动跟踪、识别无人机类型及其潜在威胁，并实施相应的防御措施。这一能力是传统手段所难以企及的。

2）模式识别和异常检测

人工智能在模式识别和异常行为检测方面表现出色。通过深度学习，系统可以从过去的数据中学习无人机的飞行模式，并能够识别出与众不同、异常或威胁性的行为。这在识别敌对或非法无人机行为方面至关重要，尤其是在它们试图模仿正常的商业无人机操作或采取隐蔽行动时。

3）自适应与持续学习

无人机技术和用途的不断进化意味着传统的反无人机方法可能很快就会过时。人工智能可以通过不断学习新的无人机特征、战术和干扰技术来适应这种变化，不仅能够根据新的威胁数据更新其模型，还能预测和对抗未来潜在的无人机发展趋势。

这3个原因共同体现了人工智能在处理高速移动目标、复杂数据环境以及不断变化的威胁景观中的关键作用，使其在反无人机技术中变得不可或缺。随着现代人工智能，特别是深度学习和增强学习的发展，反无人机技术的能力得到了大幅度的提升。

3.2 智能算法在反无人机技术中的应用

现有的人工智能技术主要应用于处理和分析多种传感器（如雷达、红外、可见光等）所收集的数据。研究表明，单独使用任一种传感器通常无法有效地探测到无人机，而一个高效的无人机探测系统通常依赖于多种传感器的组合使用。为了提升探测的准确性和效率，实现多传感器数据融合变得至关重要。人工智能算法在这里扮演了一个关键角色，特别是在从大量噪声干扰的数据中提取出有用信息，以及识别潜在的无人机威胁方面。表1详细列出了现有研究中提及的不同传感器收集的信息，以及相应的人工智能算法。

1）雷达探测数据处理

雷达技术在监测和预警海上与陆地目标方面具有重要作用，而在无人机探测领域，其重要性更是显著。雷达探测的基本原理是发射电磁波并接收反射信号，从而获取目标的位置、速度、形状等多维信息。在雷达数据处理中，

关键任务包括检测低空、慢速、小尺寸的目标（即“低慢小”目标），以及有效区分无人机与鸟类等干扰源。为了提高目标检测的准确性，算法的发展至关重要。从传统算法到现代神经网络和深度学习的应用，这一进步极大地提升了雷达探测的性能。

在传统算法的基础上，研究者们通过创新方法取得了显著的性能提升。例如，文献[11]提出了一种基于稀疏字典学习的方法，用于从海杂波中提取有效信息以识别无人机。文献[12]和[13]将多普勒频谱作为图像处理，利用神经网络LeNet[45]和GoogleNet[46]来区分目标和杂波。研究结果显示，LeNet在处理回波方面更加高效，而GoogleNet则在检测概率和虚警率方面表现更佳。文献[18]和[46]使用短时傅里叶变换（STFT）来生成频谱图，并通过主成分分析（PCA）进行降维处理。这些研究将66种类别的无人机通过K最近邻（KNN）、随机森林（RF）、朴素贝叶斯（NB）和支持向量机（SVM）进行分类。结果显示，随机森林在分类精度上最为出色，其次是朴素贝叶斯，而SVM和KNN的精度相对较低。在文献[19]中，研究者通过使用STFT将频谱转化为图像后，再利用深度卷积神经网络（DCNN）对无人机进行分类。文献[20]则直接将DCNN应用于原始微多普勒频谱图上，提出的DCNN模型能够自动学习特征，无须借助任何领域专门知识。

2）红外和可见光探测数据处理

红外和可见光探测是无人机监测的两种关键技术。红外探测捕捉无人机发出的红外波段图像，而可见光探测则获取无人机在可见光波段的图像。这两种方法都广泛应用各类图像处理技术，如降低噪声和抑制背景，以确保对无人机的有效探测和识别。近年来，基于深度学习的技术已成为处理红外和可见光图像的主流趋势。这些方法通常先利用传统图像处理技术确定潜在的目标区域，再通过深度学习进行更精确的目标检测和特征提取。

由于可见光探测不能提供距离信息并且受光照条件影响较大，许多研究正致力于将红外图像与可见光图像，以及雷达数据与可见光传感器信息结合起来，以提高无人机探测的准确性。尽管在反无人机领域，基于深度学习的红外探测技术研究还处于初级阶段，但其已经从其他目标探测领域获得了一定的启发，有望被有效地转化并应用于无人机探测。

文献[21]探索了一种结合可见光和红外图像的方法，通过利用VGG-19网络提取深层特征，并与图像的细节内容结合，实现了融合图像的重构。这项技术不仅对无人机探测有效，也适用于多曝光和多焦点图像融合等其他的一些应用。文献[22]提出了一种基于生成对抗网络（GAN）的红外和可见光图像融合方法，着重于平衡保留红外热辐射信息与增强可见光图像细节之间的关系。而文献[23]研究了雷达与可见光传感器协同监视跟踪低空目标的方法，通过实现量测模型切换和数据的在线更新，以获得更准确的目标信息。

文献[24-25]开发了基于模型的无人机增强技术，结合Faster-RCNN检测器和多域网络跟踪器，并通过利用图像序列中的残差信息来提高跟踪精度。文献[26]创建了基于网络抓取的图像和人工数据集，使用基于VGG和CNN的端到端检测模型进行无人机和鸟类的检测。文献[27]和[28]则分别运用了ResNet-101、Faster-RCNN和单步多框检测器（SSD）模型对无人机和鸟类进行检测和分类，展示了这些模型在训练和测试数据集上的优异性能。文献[29]提出了一种基于CNN的时空语义分割方法，该方法使用U-Net架构来识别图像中的感兴趣区域，并利用ResNet分类网络来确定这些区域是否包含无人机。

3）声音探测数据处理

音频探测技术在无人机监测中扮演着补充角色，它通过捕捉无人机运动时产生的独特声音特征来进行探测。这

▼表1 反无人机采用的不同探测技术及相应的算法

探测技术	人工智能算法(包括信号处理算法)	相关文献
雷达	LeNet、GoogleNet、PCA、KNN、RF、NB、SVM、DCNN等	[11-13,17-20]
红外和可见光	VGG-19、GAN、Faster-RCNN、U-Net、ResNet等	[21-29]
声音	MFCC、LPCC、SVM、GMM、CNN、RNN等	[30-33]
无线	SVM、RF、CNN等	[34-37]
多传感器融合	贝叶斯融合、基于信号方差的融合、最优融合等	[38-44]

CNN：卷积神经网络
DCNN：深度卷积神经网络
GAN：生成对抗网络
GMM：高斯混合模型
KNN：K最近邻
LPCC：线性预测倒谱系数
MFCC：梅尔频率倒谱系数
NB：朴素贝叶斯
PCA：主成分分析
RCNN：区域卷积神经网络
RF：随机森林
RNN：循环神经网络
SVM：支持向量机
VGG：视觉几何组

种技术的主要挑战包括环境噪声的干扰、探测距离的限制，以及公共无人机声音数据集的缺乏。尽管如此，音频探测仍然被视为雷达和可见光探测的有效补充，尤其是在需要区分无人机和其他飞行器的情况下。

在无人机音频探测领域，研究人员通常采用梅尔频率倒谱系数（MFCC）、线性预测倒谱系数（LPCC）等特征提取方法，并结合机器学习算法进行声音数据的分类。以下是一些研究案例：

文献[30]中，研究人员使用了MFCC和LPCC方法来提取特征，并采用多种内核的SVM从包含鸟声、飞机和雷暴声的复杂环境中检测和分类无人机声音。实验结果表明，MFCC方法在性能上优于LPCC。

文献[31]改进了MFCC的流程和参数，并结合了一阶差分和多距离分段采集法。通过训练高斯混合模型（GMM），研究者建立了一个无人机音频的“指纹库”，并实现了84.4%的识别率。

文献[32]探讨了在高噪声环境下无人机声音的检测。为了解决训练数据不足的问题，研究者在无人机声音数据集中添加了多种环境声音数据进行数据扩充，并使用GMM、CNN和循环神经网络（RNN）进行检测。结果显示，RNN在各种背景数据集上表现最佳，而GMM和CNN性能相对较差。

文献[33]提出了一种由高清摄像头和麦克风组成的音频辅助摄像机阵列，通过结合视频和音频数据，使用方向分布直方图处理视频数据，并使用MFCC和SVM处理音频数据，最终采用SVM检测场景中的无人机。这种音视频结合的方法显著提高了检测框架的性能。

这些研究表明，尽管音频探测技术在无人机监测中面临着诸多挑战，但通过适当的特征提取和先进的机器学习算法，仍然可以实现对无人机声音的有效探测和识别。随着技术的进步，音频探测技术有望在未来得到更广泛的应用，并进一步提高无人机监测系统的整体性能。

4）无线探测数据处理

无线探测技术是识别和定位无人机的一种重要手段，它通过监测无人机在通信过程中产生的无线电信号，提取这些信号的频谱特征，并构建无人机特征库，以便对无人机进行检测和定位。无线探测技术的主要方法包括到达时间法（TOA）、到达时间差（TDOA）和无线电测向技术。近年来，随着人工智能技术的发展，SVM、遗传算法、聚类算法和深度学习方法等已被广泛应用于无线电信号的特征提取和分类处理，以实现更准确和高效的无人机探测与定位。

在无线电信号处理领域的研究中，一些技术虽然不是专门为无人机探测设计的，但为该领域提供了有价值的见解。例如，文献[34]中，研究者采用SVM识别无线电信号，并通过多项式搜索算法优化多项式核函数的SVM，使其在识别无线电对地干扰信号方面的准确性和鲁棒性超越传统的遗传算法。文献[35]提出了一种创新的典型频谱方法，用于分析广播频段的频谱数据。该方法通过重复实验提取关键特征，并利用聚类算法构建典型频谱，进而识别干扰源和非法广播，为无线电监测提供了新的视角。

文献[36]采用图像二值化和去噪算法将二维图像转换为二进制格式，再通过六层卷积神经网络（CNN）进行特征提取和分类，有效地检测、跟踪和定位辐射源。文献[37]引入了一种深度门控递归单元卷积网络，专注于无线电信号的特征提取和分类。该方法在对比测试中表现优于SVM和RF，准确率高达90.6%，有效地实现了对31种不同信号的分类。

文献[35]和[36]的方法主要针对单样本分类，而文献[37]的方法适用于多样本情况，显示了深度学习在无线电波形分类中的潜力。这些研究为无人机探测领域提供了新的技术思路和方法参考，有助于推动该领域的技术发展。

5）多传感器融合数据处理

多传感器数据融合能够将来自雷达、红外线、可见光摄像机和声波监测等不同传感器的信息进行整合。融合算法可以通过学习不同传感器的数据表征，优化数据融合过程中的特征提取和决策逻辑,可以在各种环境条件下识别和跟踪目标，即便在视线不佳或天气条件恶劣的情况下也能保持高准确率[38]。特别是在传感器之一被干扰或失效时，融合算法可以重新分配资源，以确保系统的整体性能不受影响。通过这种自我调节的机制，反无人机系统在面对日益复杂的无人机威胁时，可以保持高度的灵活性以及鲁棒性。

最近的研究成果在多传感器数据融合方面展现了创新的方法和显著的潜力。例如，文献[39]针对雷达和红外传感器的数据融合进行了深入研究，分析了5种不同的融合技术：贝叶斯融合、基于信号方差的融合、最优融合、基于误差方差的融合和基于扩展卡尔曼滤波器的融合。研究发现，使用扩展卡尔曼滤波器的融合方法在跟踪性能上优于其他方法。文献[40]探索了激光雷达和红外传感器的融合，以实现高速低空目标的三维定位。在这项工作中，卡尔曼滤波器和扩展卡尔曼滤波器被用于优化状态估计和数据融合过程，且这种融合主要发生在决策级别。文献[41]采用基于导引滤波器的混合多尺度分解方法来融合图像，通过自

适应增强可见光图像，并利用红外图像的像素值进行指数变换，提取红外特征信息，实现了在特征级别的有效融合。文献[42]通过对红外和可见光图像进行特征提取，然后将局部方差偏移、对比和熵作为证据，在特征级别上进行融合。文献[43]将声学、红外摄像机和雷达传感器结合起来进行鸟类监测，通过对这些传感器收集的数据进行预处理，包括对声学传感器数据的特征提取和分类，以及对红外图像进行背景减影、斑点检测、阈值化和噪声抑制，同时对雷达数据采用粒子滤波器进行处理。在数据融合的阶段，文献[43]则采用了两级融合的架构，首先在特征级别上融合了红外和雷达数据，然后在决策层上将这些融合的特征向量与声学数据相结合，并且使用模糊贝叶斯方法[44]进行最终的融合。

这些研究表明，多传感器融合技术不仅能够提高无人机探测系统的性能，而且在不同领域的应用中也显示出巨大的潜力。未来，随着传感器技术的进步和算法的不断发展，多传感器融合技术将在无人机探测和跟踪领域发挥更加重要的作用，并为系统的优化和升级提供新的方向和思路。

基于上述分析，我们看到人工智能算法成功应用于快速数据处理、模式识别以及实时决策等领域。然而，自适应学习和自我进化，在对抗无人机的算法中仍是一片待开发的领域。这些技术的进步对于促进人工智能自我演化至关重要，它们是实现完全自动化、由人工智能驱动的反无人机系统的关键。虽然目前这样的系统大多仍处于概念和规划阶段，但已经有了一些令人鼓舞的进展[14]。

人工智能的发展极大地促进了无人机技术的进步，同时也对反无人机系统提出了更高的技术要求。接下来我们将预测未来的发展趋势，并讨论可能遇到的挑战。

4 未来发展趋势与挑战

为了有效应对无人机技术进步带来的威胁和挑战，反无人机技术需要在以下 3 个核心领域实现重大突破：自主学习、对抗博弈以及多智能体协同。

4.1 自主学习

反无人机技术的发展正呈现出一个明显的趋势，即集中在自主学习领域。随着无人机技术的飞速发展，传统的反无人机系统在适应新策略和变化方面面临越来越大的挑战。为了有效地应对这些挑战，自主学习成为了关键的发展方向。未来的反无人机系统需要利用先进的机器学习算法，应对各种无人机的新威胁。

自主学习的重要性在于系统的实时适应性和智能反应策略。通过从每次遭遇中学习，反无人机系统能够不断优化自身的反应策略，从而提高拦截成功率。特别是，当无人机在遭遇干扰后改变飞行策略时，人工智能算法能够迅速识别这一变化，并灵活地调整预测算法，以适应新的飞行模式。这种自我调整能力对于对抗日益复杂和自主的无人机系统至关重要，因为这些系统可能会实时动态地调整飞行路径，以规避侦测或反制。通过不断的学习和适应，反无人机系统能够提前部署资源，如调整传感器指向、准备干扰设备或调动拦截无人机，从而有效地中和无人机，确保其在进入关键区域或执行潜在任务之前被成功阻截。这一发展趋势为未来的反无人机技术的提升奠定了坚实基础。

然而，随着这一发展趋势，一系列挑战也随之而来，需要精心设计的解决方案来应对。

首先，无人机系统的不断演进增加了系统对新策略和变化的适应难度，这需要更高级别的智能化。解决这一挑战的关键在于优化智能学习算法。传统反无人机系统可能无法快速适应无人机的新策略和变化，缺乏实时智能学习能力。因此，引入深度强化学习[47]等先进机器学习算法变得至关重要。通过大量数据进行模型训练，系统能够持续提升对不断变化的威胁的感知和应对能力，实现系统的智能化进化。

其次，无人机在遭遇干扰后的实时反应性以及飞行策略的动态调整带来了更大的复杂性，传统系统难以满足这种需求。解决这一挑战的途径是引入实时反应性和动态调整的处理机制。通过实时决策算法，反无人机系统能够在毫秒级别内做出智能决策，确保在无人机遭遇干扰时能够迅速而有效地调整应对策略。结合传感器网络，系统可实现对无人机动态调整路径的实时监测和反制，保持对无人机的持续有效拦截。这样的系统能够应对威胁的多样性和动态性，提高反无人机系统的整体应对能力。

最后，跨领域合作与信息共享成为另一个解决方案。多领域的无人机技术发展需要不同领域专家的共同协作和信息共享，以整合各种数据源和技术。解决这一挑战的关键在于建立跨领域的合作机制。联合研发和跨领域团队合作，能够促进信息共享和技术整合，可以推动反无人机系统的综合性发展，提升系统在复杂环境下的适应性和效果。这样的协同性和综合性发展将为未来的反无人机系统提供更为全面、高效的解决方案，使其能够更好地适应不断变化的无人机威胁。

4.2 对抗博弈

在未来反无人机系统的发展中，对抗博弈将成为一个关键的焦点。这一趋势涉及系统与无人机之间智能对抗的模拟和优化，目的是提升系统的适应性和对无人机威胁的整体应对能力。这一发展方向是对无人机技术快速演进和多样化应用的响应，要求系统不仅具备高效的感知能力，还需要通过对抗博弈来应对无人机可能采取的多种战术和策略。

在对抗博弈的框架下，未来的反无人机系统将通过建立智能对抗模型，模拟无人机可能的战术和行为，包括可能的规避策略、干扰手段以及突然变化的飞行路径。利用深度学习和强化学习等先进技术，系统能够从历史数据中学到无人机的行为模式，并实时更新模型以适应新的威胁。这样的模拟将使系统能够更加准确地预测和解读无人机的行为。

基于对抗博弈的模型，反无人机系统可以预测无人机的可能动作，提前识别潜在威胁。这使得系统能够在无人机进入关键区域之前，通过智能的防御措施进行部署，例如：调整传感器指向，准备干扰设备或调动拦截无人机。这样的预测性部署将大大提高系统的反应速度和对抗效果，增强整个反无人机系统的实战能力。

随着无人机技术的快速演进，反无人机系统在对抗博弈方面面临着一系列挑战。

首先，系统需要适应技术的快速发展，应对日益多样化的无人机威胁。为此，建立开放式的软件架构和模块化硬件设计显得尤为重要，以便系统能够轻松升级和适应新技术。

其次，随着对抗模型变得更加复杂，高效的数据处理与分析能力成为关键。在此方面，运用高性能计算平台和优化算法，以及云计算和边缘计算技术[48]，将提高数据分析的效率和响应速度。

最后，深度强化学习虽为系统提供了先进的学习能力，但也带来了对大量数据和计算资源的需求，以及模型可解释性的挑战。解决这一问题的途径在于使用更高效的学习算法，以及采用模型简化和优化技术以改善可解释性。

对于实时更新与适应新威胁的要求，实现自适应学习系统并采用分布式数据收集和处理系统是关键。此外，准确预测无人机的行为对于有效防御至关重要，需要结合多种数据源和智能分析技术来提高预测的准确性。

应对这些挑战的解决方案需要包括持续的技术创新、跨领域协作、模型和算法的标准化、强化防御部署策略，以及人机协同。通过不断探索新的传感技术、人工智能算法和计算平台，系统可以保持技术的领先地位。与军事、网络安全、人工智能等不同领域的专家合作，可以共同解决复杂问题。

此外，制定标准化的模型和算法框架可以提高开发效率和互操作性。采用多层次、多维度的防御策略，包括物理拦截、电子干扰等手段，可以加强系统的防御能力。加入人工智能辅助决策的人机协同系统，将提高系统对复杂环境的适应能力和作战效率。

综上所述，通过采取灵活、创新、协同的策略，反无人机系统将能够有效对抗日益复杂的无人机威胁，确保关键区域的安全与稳定。

4.3 多智能体协同

多智能体协同的应用将成为一个关键趋势，特别是在应对逐渐增加的无人机群威胁方面。这种趋势指向了不同反无人机系统之间必须实施的高效信息共享和协作作战，目的是实现更广泛的监控范围和更精准的任务执行能力。由于无人机技术正迅速发展，尤其是无人机群带来的新型挑战，单个反无人机系统面对这种复杂多目标的威胁时往往显得不够充分。因此，开发能够协同作战的多智能体系统变得至关重要。这类系统能够在各自单元之间实现有效的通信和协调，例如：通过共享有关无人机群的位置、速度和飞行方向等关键信息，各单元可以共同制定出更为有效的拦截和干扰战术。这样的方法不仅提升了单一系统的防御能力，也极大地增强了整体网络防御体系的效能和适应性。

与单智能体系统相比，多智能体系统的主要优势在于各个智能体之间能够互相学习并共享能力，从而极大地提高整体协同决策的效率和有效性。这些智能体能够独立作出决策，同时通过与其他智能体的交互实现知识和经验的互惠共享。例如，一个智能体发现的有效防御策略可以通过网络共享给其他智能体，从而增强整体系统对新挑战的适应能力。协同决策进一步增强了这种系统的效能，特别是在处理诸如多目标拦截和战术规避等复杂战术情况时。此外，未来的研究将致力于提高系统的自适应能力，而这依赖于先进的机器学习算法，使智能体能够快速学习并优化其行动模式。同时，随着智能体间互动和数据共享的增加，保证信息安全和系统稳定运行也成为重要的研究方向，而这需要发展高效的加密通信协议和鲁棒网络架构。综合来看，多智能体系统的发展将专注于增强自主决策、学习能力、协同决策效率、自适应性以及系统的安全性和稳定性，以应对日益复杂的安全挑战。

在当今日益复杂的安全环境中，多智能体协同系统在反无人机领域中应对无人机群等威胁的能力变得至关重要。这一系统面临的主要挑战之一是如何实现高效的信息共享与协作作战。为应对无人机群等复杂威胁，不同反无人机系统之间必须能够有效地共享信息和协调行动。针对这一挑战，可采用的解决方案包括开发高级通信协议和数据共享平台，以确保可以实时、准确地交换信息。此外，运用云计算和边缘计算的相关技术，可以大幅地提高数据处理速度和响应时间，从而加强了多智能体系统的整体协作效率。

此外，自主决策和智能体之间的学习能力也是多智能体协同系统面临的关键挑战。在这方面，每个智能体不仅需要有能力独立作出决策，还要能够从其他智能体那里学习并共享知识和经验。解决这一挑战的方法是利用先进的机器学习和人工智能算法，例如深度强化学习，以增强智能体的自主学习和决策能力。此外，实施智能共享机制可以使一个智能体的学习成果被其他智能体所利用，从而提升整个系统的智能水平和响应能力。为了提高协同决策的效率与有效性，设计高级的决策支持系统和协同算法至关重要，以确保智能体在考虑全局最优解时能够有效地协作。

与此同时，增强多智能体系统在面对环境变化和新型威胁时的自适应能力同样重要。集成的自适应学习机制可以使系统根据环境的变化自动调整其行为和策略。此外，保障信息在多智能体系统间传输的安全性和系统的稳定运行也不容忽视。这需要实施高效的加密通信协议和鲁棒的网络架构设计，以增强系统对外部干扰和内部故障的抵抗能力。总而言之，通过实施这些解决方案，多智能体协同系统能够在提高自主决策和学习能力的同时，保持高效的协同决策，有效应对环境变化，并确保系统的信息安全和稳定性。这些措施将共同提升反无人机系统面对日益复杂和多样化威胁时的总体效能和适应性，为未来安全环境中的挑战提供有效的解决方案。

5 结束语

本文全面探讨了反无人机技术的发展趋势、技术原理，以及通信技术和人工智能在这一领域的关键作用。通过各章节的深入分析，本文得出以下结论：

首先，随着无人机技术的快速进步，反无人机技术的需求变得尤为迫切。尽管无人机为社会带来了诸多便利，但其广泛应用也引发了对安全和隐私的担忧。反无人机技术的研究与开发，旨在应对无人机可能带来的各种潜在威胁，如侵犯隐私、侵犯领空和实施恶意攻击等。

其次，通信技术在反无人机技术中扮演了核心角色。通过优化通信系统，提高数据传输的稳定性和实时响应能力，反无人机技术的运行效率得到了显著提升。随着通信技术的持续进步，反无人机系统的性能和可靠性也将得到进一步加强。

最后，人工智能在无人机目标识别和自主决策方面起到了至关重要的作用。深度学习和计算机视觉技术的应用，使得系统能够精确识别无人机目标，并及时做出智能响应，从而提升了系统的自主性和效率。

未来，反无人机技术的发展将依赖于自主学习、对抗博弈和多智能体协同等关键技术，以更好地适应不断变化的无人机威胁。然而，这一领域也面临着诸多挑战，包括隐身和低速小目标的识别、高机动性无人机的应对、智能无人机和无人机群的出现、法律和道德问题的处理、成本和可持续性问题，以及国际合作的重要性。克服这些挑战需要跨学科的研究和国际合作。

总的来说，反无人机技术的发展离不开通信技术和人工智能的支持，同时也需要应对未来的挑战。只有通过持续的创新和合作，我们才能更好地保障社会安全、保护隐私和维护法律秩序。本论文期望能为反无人机技术领域的研究和实践提供有价值的参考，推动该领域的发展与进步。

参考文献

[1] 张涛, 芦维宁, 李一鹏. 智能无人机综述 [J]. 航空制造技术, 2013, 56(12): 32−35. DOI: 10.3969/j.issn.1671−833X.2013.12.003
[2] ZHANG T, LI Q, ZHANG C S, et al. Current trends in the development of intelligent unmanned autonomous systems [J]. Frontiers of information technology & electronic engineering, 2017, 18(1): 68−85. DOI: 10.1631/FITEE.1601650
[3] 张静, 张科, 王靖宇, 等. 低空反无人机技术现状与发展趋势 [J]. 航空工程进展, 2018, 9(1): 1−8. DOI: 10.16615/j. cnki. 1674−8190.2018.01.001
[4] 蔡亚梅, 姜宇航, 赵霜. 国外反无人机系统发展动态与趋势分析 [J]. 航天电子对抗, 2017, 33(2): 59−64
[5] CHAMOLA V, KOTESH P, AGARWAL A, et al. A comprehensive review of unmanned aerial vehicle attacks and neutralization techniques [J]. Ad hoc networks, 2021, 111: 102324. DOI: 10.1016/j.adhoc.2020.102324
[6] 王妮. 基于多源信息融合的无人机视觉导航技术 [D]. 西安: 西安电子科技大学, 2019
[7] 周新民, 吴佳晖, 贾圣德, 等. 无人机空战决策技术研究进展 [J]. 国防科技, 2021, 42(3): 33−41. DOI: 10.13943/j. issn1671−4547.2021.03.05
[8] 刘宏, 万立健, 陆亚英. 基于北斗卫星导航系统的远距离海洋工程高精度定位技术 [J]. 测绘通报, 2017, (5): 62−66. DOI: CNKI:SUN:

CHTB.0.2017-05-015
[9] YAN S, CAO X Y, LIU Z L, et al. Interference management in 6G space and terrestrial integrated networks: challenges and approaches [J]. Intelligent and converged networks, 2020, 1(3): 271-280. DOI: 10.23919/ICN.2020.0022
[10] GHOSH A, RATASUK R, MONDAL B, et al. LTE-advanced: next-generation wireless broadband technology [J]. IEEE wireless communications, 2010, 17(3): 10-22. DOI: 10.1109/MWC.2010.5490974
[11] DONG Z W, SUN J, SUN J M, et al. Research on sea clutter suppression using sparse dictionary learning [C]//Proceedings of IEEE 8th Joint International Information Technology and Artificial Intelligence Conference (ITAIC). IEEE, 2019: 967-971. DOI: 10.1109/ITAIC.2019.8785824
[12] WANG L, TANG J, LIAO Q M. A study on radar target detection based on deep neural networks [J]. IEEE sensors letters, 2019, 3 (3): 7000504. DOI: 10.1109/LSENS.2019.2896072
[13] CHEN X L, SU N Y, GUAN J, et al. Integrated processing of radar detection and classification for moving target via time-frequency graph and CNN learning [C]//Proceedings of URSI Asia-Pacific Radio Science Conference (AP-RASC). IEEE, 2019: 1-4
[14] 罗俊海, 王芝燕. 无人机探测与对抗技术发展及应用综述 [J]. 控制与决策, 2022, 37(3): 530-544
[15] RAJ V, C A A. Understanding the future communication: 5G to 6G [J]. International research journal on advanced science hub, 2021, 3(Special Issue 6S): 17-23. DOI: 10.47392/irjash.2021.159
[16] SHEKHAR, SINGHAL A, SHARMA R, et al. Study of analysis of multiple input and multiple outputs (mimo) technology in wireless communication [C]//Proceedings of International Conference on Micro-Electronics and Telecommunication Engineering (ICMETE). IEEE, 2016: 658-662. DOI: 10.1109/ICMETE.2016.134
[17] TAN R, LIM H S, SMITS A B, et al. Improved micro-Doppler features extraction using Smoothed-Pseudo Wigner-Ville distribution [C]//Proceedings of IEEE Region 10 Conference (TENCON). IEEE, 2016: 730-733. DOI: 10.1109/TENCON.2016.7848099
[18] SUN Y X, FU H, ABEYWICKRAMA S, et al. Drone classification and localization using micro-doppler signature with low-frequency signal [C]//Proceedings of IEEE International Conference on Communication Systems (ICCS). IEEE, 2018: 413-417. DOI: 10.1109/ICCS.2018.8689237
[19] PARK D, LEE S, PARK S, et al. Radar-spectrogram-based UAV classification using convolutional neural networks [J]. Sensors, 2020, 21(1): 210. DOI: 10.3390/s21010210
[20] LIU Y, LIU J Y. Recognition and classification of rotorcraft by micro-Doppler signatures using deep learning[C]//International Conference on Computational Science. Springer, 2018: 141-152.10.1007/978-3-319-93698-7_11
[21] LI H, WU X J, KITTLER J. Infrared and visible image fusion using a deep learning framework [C]//Proceedings of 24th International Conference on Pattern Recognition (ICPR). IEEE, 2018: 2705-2710. DOI: 10.1109/ICPR.2018.8546006
[22] MA J Y, YU W, LIANG P W, et al. FusionGAN: a generative adversarial network for infrared and visible image fusion [J]. Information fusion, 2019, 48: 11-26. DOI: 10.1016/j.inffus.2018.09.004
[23] 张雅雯, 胡士强. 低空目标的雷达/可见光协同监视跟踪方法研究 [J]. 计算机工程与应用, 2018, 54(6): 234-240
[24] CHEN Y R, AGGARWAL P, CHOI J, et al. A deep learning approach to drone monitoring [C]//Proceedings of Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC). IEEE, 2017: 686-691. DOI: 10.1109/APSIPA.2017.8282120
[25] SCHUMANN A, SOMMER L, KLATTE J, et al. Deep cross-domain flying object classification for robust UAV detection [C]//Proceedings of 14th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS). IEEE, 2017: 1-6. DOI: 10.1109/AVSS.2017.8078558
[26] AKER C, KALKAN S. Using deep networks for drone detection [EB/OL]. [2024-02-26]. https://arxiv.org/abs/1706.05726
[27] NALAMATI M, KAPOOR A, SAQIB M, et al. Drone detection in long-range surveillance videos [C]//Proceedings of 16th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS). IEEE, 2019: 1-6. DOI: 10.1109/AVSS.2019.8909830
[28] MAGOULIANITIS V, ATALOGLOU D, DIMOU A, et al. Does deep super-resolution enhance UAV detection? [C]//Proceedings of 16th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS). IEEE, 2019: 1-6. DOI: 10.1109/AVSS.2019.8909865
[29] CRAYE C, ARDJOUNE S. Spatio-temporal semantic segmentation for drone detection [C]//Proceedings of 16th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS). IEEE, 2019: 1-5. DOI: 10.1109/AVSS.2019.8909854
[30] ANWAR M Z, KALEEM Z, JAMALIPOUR A. Machine learning inspired sound-based amateur drone detection for public safety applications [J]. IEEE transactions on vehicular technology, 2019, 68(3): 2526-2534. DOI: 10.1109/TVT.2019.2893615
[31] 王威, 安腾飞, 欧建平. 无人机被动音频探测和识别技术研究 [J]. 声学技术, 2018, 37(1): 89-93
[32] JEON S, SHIN J W, LEE Y J, et al. Empirical study of drone sound detection in real-life environment with deep neural networks [C]//Proceedings of 25th European Signal Processing Conference (EUSIPCO). IEEE, 2017: 1858-1862
[33] LIU H, WEI Z Q, CHEN Y T, et al. Drone detection based on an audio-assisted camera array [C]//Proceedings of IEEE Third International Conference on Multimedia Big Data (BigMM). IEEE, 2017: 402-406. DOI: 10.1109/BigMM.2017.57
[34] KONG M M, LIU J, ZHANG Z H, et al. Radio ground-to-air interference signals recognition based on support vector machine [C]//Proceedings of IEEE International Conference on Digital Signal Processing (DSP). IEEE, 2015: 987-990. DOI: 10.1109/ICDSP.2015.7252025
[35] YAN H H, ZHOU B, LIU J, et al. Radio signal recognition based on constructing typical spectrum [C]//Proceedings of 2nd IEEE International Conference on Computer and Communications (ICCC). IEEE, 2016: 1889-1894. DOI: 10.1109/CompComm.2016.7925030
[36] ZHANG M, DIAO M, GUO L M. Convolutional neural networks for automatic cognitive radio waveform recognition [J]. IEEE access, 1074, 5: 11074-11082. DOI: 10.1109/ACCESS.2017.2716191
[37] LI R D, HU J H, YANG S Y. Deep gated recurrent unit convolution network for radio signal recognition [C]//Proceedings of IEEE 19th International Conference on Communication Technology (ICCT). IEEE, 2019: 159-163. DOI: 10.1109/ICCT46805.2019.8947225
[38] MUZAMMAL M, TALAT R, SODHRO A H, et al. A multi-sensor data fusion enabled ensemble approach for medical data from body sensor networks [J]. Information fusion, 2020, 53: 155-164. DOI: 10.1016/j.inffus.2019.06.021

[39] RAJESWARI K, ISHWARYA A, VAISHNAVI K K, et al. Performance analysis of data fusion methods for radar and IRST 3D target tracking [C]//Proceedings of International Conference on Wireless Communications, Signal Processing and Networking (WiSPNET). IEEE, 2017: 2570–2574. DOI: 10.1109/WiSPNET.2017.8300227
[40] ILYAS K, ULLAH I. A state estimation and fusion algorithm for high–speed low–altitude targets [C]//Proceedings of 19th International Multi–Topic Conference (INMIC). IEEE, 2016: 1–5. DOI: 10.1109/INMIC.2016.7840127
[41] LUO J Z, RONG C Z, JIA Y X, et al. Fusion of infrared and visible images based on image enhancement and feature extraction [C]// Proceedings of 11th International Conference on Intelligent Human–Machine Systems and Cybernetics (IHMSC). IEEE, 2019: 212–216. DOI: 10.1109/IHMSC.2019.00056
[42] WANG A L, JIANG J N, ZHANG H Y. Multi–sensor image decision level fusion detection algorithm based on D–S evidence theory [C]//Proceedings of Fourth International Conference on Instrumentation and Measurement, Computer, Communication and Control. IEEE, 2014: 620–623. DOI: 10.1109/IMCCC.2014.132
[43] MIRZAEI G, JAMALI M M, ROSS J, et al. Data fusion of acoustics, infrared, and marine radar for avian study [J]. IEEE sensors journal, 2015, 15(11): 6625–6632. DOI: 10.1109/JSEN.2015.2464232
[44] 刘海燕, 陈红林, 史志富, 等. 基于模糊贝叶斯网络的空中目标多传感器融合识别研究 [J]. 电光与控制, 2009, 16(3): 37–41. DOI: 10.3969/j.issn.1671–637X.2009.03.010
[45] LECUN Y, BOTTOU L, BENGIO Y, et al. Gradient–based learning applied to document recognition [J]. Proceedings of the IEEE, 1998, 86(11): 2278–2324. DOI: 10.1109/5.726791
[46] SZEGEDY C, LIU W, JIA Y Q, et al. Going deeper with convolutions [EB/OL]. [2024–02–26]. http://arxiv. org/abs/1409.4842
[47] 刘全, 翟建伟, 章宗长, 等. 深度强化学习综述 [J]. 计算机学报, 2018, 41(1): 1–27. DOI: 10.11897/SP.J.1016.2018.00001
[48] 施巍松, 孙辉, 曹杰, 等. 边缘计算: 万物互联时代新型计算模型 [J]. 计算机研究与发展, 2017, 54(5): 907–924. DOI: 10.7544/issn1000–1239.2017.2016094

作 者 简 介

邱宝华，中国移动通信集团广西有限公司总经理；主要研究方向为智能社会治理。

基于动态通道绑定的更高速无源光网络

Higher Speed PON Based on Dynamic Channel Bonding

张伟良/ZHANG Weiliang[1,2]，王霄雨/WANG Xiaoyu[3]，黄新刚/HUANG Xingang[1,2]
（1. 中兴通讯股份有限公司，中国 深圳 518057；
2. 移动网络和移动多媒体技术国家重点实验室，中国 深圳 518055；
3. 中国电信集团有限公司，中国 北京 100020）
(1. ZTE Corporation, Shenzhen 518057, China;
2. State Key Laboratory of Mobile Network and Mobile Multimedia Technology, Shenzhen 518055, China;
3. China Telecom Corporation Ltd. , Beijing 100020, China)

DOI：10.12142/ZTETJ.202402014
网络出版地址：http://kns.cnki.net/kcms/detail/34.1228.TN.20240404.2315.004.html
网络出版日期：2024-04-08
收稿日期：2024-02-18

摘要：传统无源光网络单通道速率提升成本越来越高。多通道无源光网络可通过动态通道绑定实现速率提升，进一步满足服务多样性需求。分析了现有IEEE和ITU-T无源光网络标准中的动态通道绑定需求和功能实现，以及数据传输过程中存在的带宽效率、数据顺序恢复等问题，提出了一种更高速无源光网络动态通道绑定中数据序列化传输和顺序恢复方法。该方法简化了动态通道绑定处理，避免了带宽效率下降问题。

关键词：更高速无源光网络；动态通道绑定；序列化传输；顺序恢复

Abstract: The cost of single channel rate improvement in traditional passive optical networks (PON) is increasing. Multi-channel PON could achieve higher bandwidth capacity by dynamic channel bonding and could further meet service diversity. The requirement and function of dynamic channel bonding in current IEEE and ITU-T PON standards and the existing problems are analyzed. A transmission method including serialization transmission and order recovery for dynamic channel bonding in higher speed PON is provided. This method simplifies dynamic channel bonding processing and avoids the problem of bandwidth efficiency degradation.

Keywords: higher speed PON; dynamic channel bonding; serialization transmission; order recovery

引用格式：张伟良，王霄雨，黄新刚．基于动态通道绑定的更高速无源光网络 [J]. 中兴通讯技术，2024, 30(2): 100-106. DOI: 10.12142/ZTETJ.202402014
Citation：ZHANG W L, WANG X Y, HUANG X G. Higher speed PON based on dynamic channel bonding [J]. ZTE technology journal, 2024, 30(2): 100-106. DOI: 10.12142/ZTETJ.202402014

1 无源光网络动态通道绑定需求

在10G无源光网络（PON）之后，国际电信联盟电信标准化部门（ITU-T）和电气电子工程师学会（IEEE）分别启动多通道无源光网络和动态通道绑定无源光网络的标准化工作，以提升后10G PON无源光网络的带宽容量和峰值速率。

多通道无源光网络支持多个通道。每个通道由一对上下行波长组成，支持一部分光网络单元（ONU）。各通道彼此独立，并且每个ONU最大仅支持一个通道的带宽容量。动态通道绑定无源光网络，在多通道无源光网络的基础上，各ONU可以支持一个或者多个通道。不同ONU支持的通道及通道数可以不同且可以共存。

IEEE 802.3av标准[1]中的Nx25G-EPON支持25 Gbit/s和50 Gbit/s两种速率。50 Gbit/s速率是通过绑定两个25 Gbit/s速率的通道来实现的。Nx25G-EPON支持动态通道绑定。单通道ONU和由2个通道绑定的ONU可以在同一个光分配网络（ODN）中共存。

ITU-T G.989系列标准[2-3]中的时分波分复用无源光网络（TWDM-PON）可支持4~8个通道，并且每个通道的速率为10 Gbit/s。在后续增补中，ITU-T G.989.1 amd1[4]增加了TWDM-PON超过10G带宽ONU的需求，在ITU-T G.989.3

基金项目：上海市科技计划项目（20511102400）；深圳市战略性发展项目（XMHT20190101034）

amd2[5]中，引入绑定ONU。绑定ONU支持多个通道，且支持动态通道绑定。单通道ONU可以和不同通道数的绑定ONU共存。

ITU-T G.9804.2标准[6]中的更高速无源光网络（HSP）除基于DAW的50G-PON[7]外，还包括TWDM-PON多通道功能。每个通道支持50 Gbit/s速率，并支持动态通道绑定。单通道ONU可以和不同通道数的多通道ONU共存。

2 无源光网络的动态通道绑定实现分析

2.1 Nx25G-EPON动态通道绑定

Nx25G-EPON支持两对上下行波长，分别为：

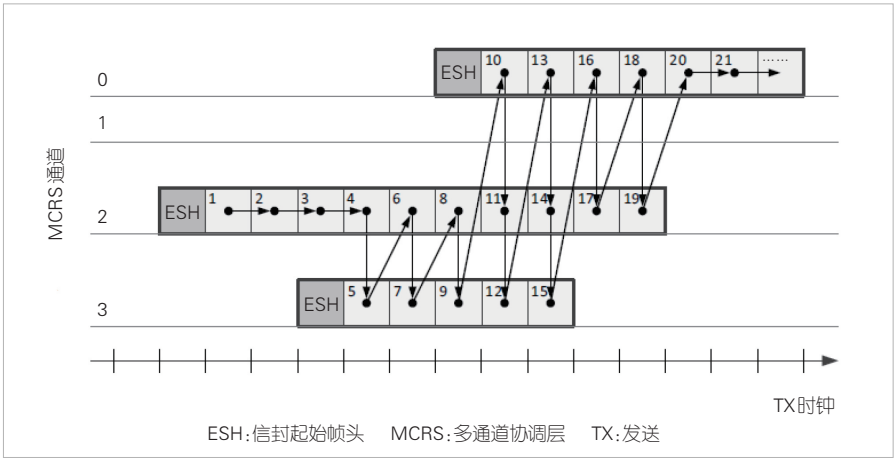
下行波长：1358±2nm，1342±2nm；

上行波长：1 270±10 nm，1 300±10 nm。

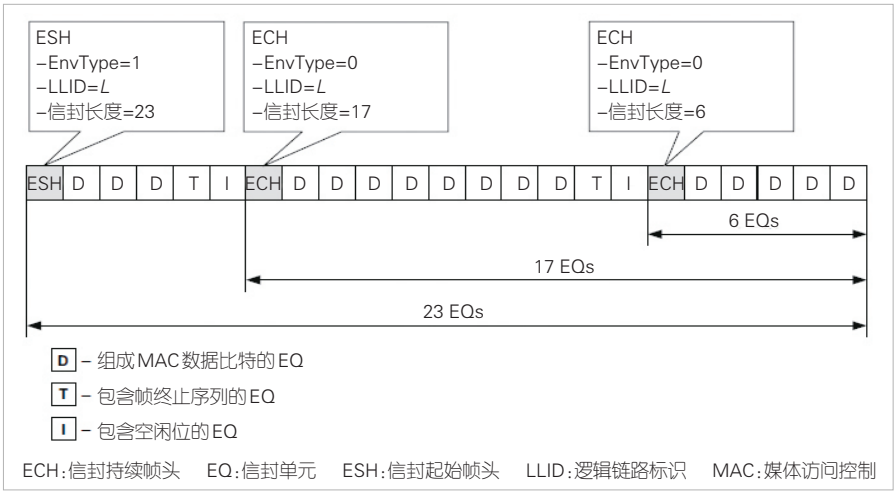
Nx25G-EPON的动态通道绑定机制不限定通道数，把属于同一个逻辑链路标识（LLID）的业务数据帧切分为多个8字节长的信封单元（EQ）。每个EQ在传输机会最早的通道上发送，如果有多个传输机会最早的通道，则选择在通道编号最小的通道上发送，如图1所示。同一个通道上的EQ序列增加8字节的信封起始帧头（ESH）以便封装成信封。

Nx25G-EPON动态通道绑定机制考虑了属于同一个LLID的连续以太网帧发送。第一个以太网帧如果要实现在多通道上传输，就需要在每个通道上增加ESH。后续的每个以太网帧的前导将被替换为信封持续帧头（ECH），不需要增加ESH，具体见图2。ESH和ECH中的信封长度显示，属于同一个LLID以太网帧在连续发送时彼此是相关联的，需要预先收集待发送以太网帧。具体实现过程需要缓存较长时间，存在一定的限制。

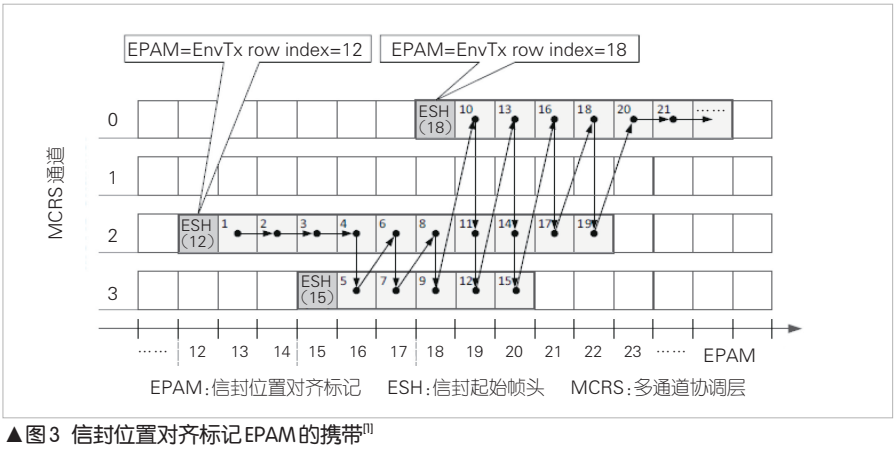
Nx25G-EPON动态通道绑定机制考虑了通道传输偏移，即各通道有不同的传输时延，可能导致各通道上传输数据的相对位置发生变化，进而产生接收乱序。Nx25G-EPON发送端和接收端都设置了32 EQ大小的循环缓存。发送过程中，每个信封的ESH在发送端循环缓存中的位置，被记录在ESH的信封位置对齐标记（EPAM）中，如图3所示。这样可以固定各通道传输数据之间的相对位置关系。



▲图1 Nx25G-EPON动态通道绑定机制数据传输示例[1]



▲图2 同一个LLID的业务数据帧连续发送[1]



▲图3 信封位置对齐标记EPAM的携带[1]

2.2 TWDM-PON动态通道绑定方案

TWDM-PON下行波段和上行波段的分配如表1所示。下行波长和上行波长的具体分配如表2和表3所示。ITU-T G.989.2[2]规定由表2和表3组成下行/上行波长对的前4对波长用于TWDM-PON。如果需要，后4对波长也可以用于TWDM-PON。

在引入动态通道绑定后，ITU-T G.989.3传输汇聚层的标准化工作基本完成。因此，TWDM-PON支持通过传输汇聚层之上的业务层来实现动态通道绑定。每个通道各自激活，各自传输业务。业务层利用多个通道传输业务。

2.3 HSP动态通道绑定

HSP[6]动态通道绑定暂未定义各通道对应的上下行波长，其工作原理如图4所示。HSP数据封装帧包括绑定数据封装（XGEM）帧和非绑定XGEM帧。非绑定XGEM帧和单通道50G-PON的数据传输方式是一样的。对于绑定XGEM帧，业务数据帧被切割为若干4字节数据单元。每个数据单元在时隙最早、编号最小的通道上发送。每个通道上的数据单元序列前会增加一个8字节XGEM帧头。

这里我们以图4为例来解释HSP动态通道绑定数据传输过程。业务数据帧长度为74字节，填充2字节后变为76字节，随后被分割成19个4字节的数据单元。这些数据单元的编号依次为0~18。系统先在通道λ3发送数据单元0~5，然后在通道λ2和λ3交替发送数据单元6和7，最后在通道λ1、λ2和λ3交替发送数据单元8~18。每个通道的数据单元序列前都被增加一个XGEM帧头。在通道λ1上的XGEM帧头中，最后分片标志LF=0表示业务数据帧的结尾不在该通道上，净荷长度指示PLI=16表示该通道包含16个字节业务数据，即4个数据单元，不包含字节填充。在通道λ2上的XGEM帧头中，LF=1表示业务数据帧的结尾在该通道上，PLI=18表示该通道包含18字节业务数据，即5个数据单元。其中，最后一个数据单元有2个字节填充。在通道λ3上的XGEM帧头中，LF=0表示业务数据帧的结尾不在该通道上，PLI=40表示该通道包含40个字节业务数据，即10个数据单元，没有字节填充。

关于通道传输偏移，HSP动态通道绑定机制假设各通道之间的偏移量是固定的，且发送侧和接收侧都知道该偏移量，因此发送侧和接收侧对数据单元顺序有共同认知，但是HSP未给出相应的数据单元顺序保证机制。

HSP动态通道绑定机制只考虑单个业务数据帧在多通道

▼表1 时分波分复用无源光网络下行波段和上行波段[2]

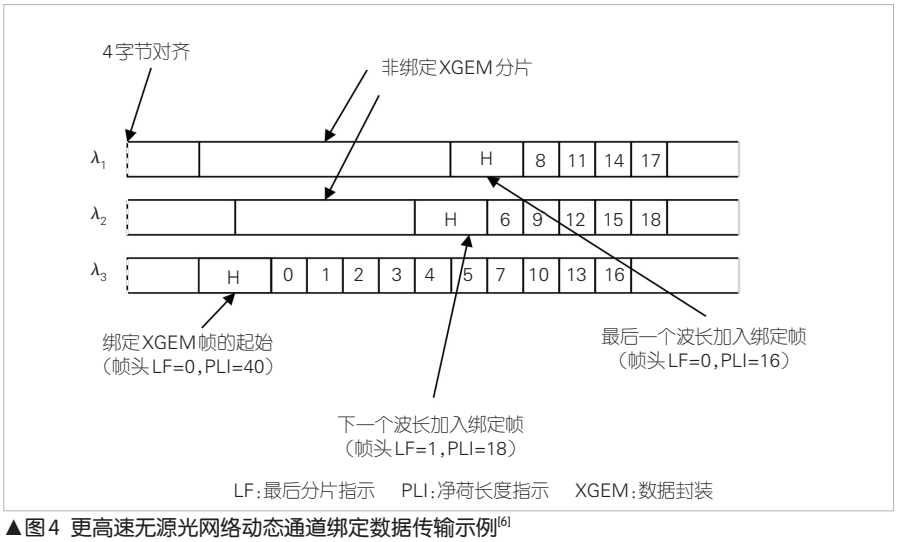
下行波段/nm	上行波段/nm
1 596~1 603	1 524~1 544（宽带选项） 1 528~1 540（收窄选项） 1 532~1 540（窄带选项）

▼表2 时分波分复用无源光网络下行波长定义[2]

通道	中心频率/THz	波长/nm
1	187.8	1 596.34
2	187.7	1 597.19
3	187.6	1 598.04
4	187.5	1 598.89
5	187.4	1 599.75
6	187.3	1 600.60
7	187.2	1 601.46
8	187.1	1 602.31

▼表3 时分波分复用无源光网络上行波长定义[2]

通道	50 GHz通道间隔		100 GHz通道间隔		200 GHz通道间隔	
	频率/THz	波长/nm	频率/THz	波长/nm	频率/THz	波长/nm
1	195.25	1 535.43	195.6	1 532.68	196.1	1 528.77
2	195.20	1 535.82	195.5	1 533.47	195.9	1 530.33
3	195.15	1 536.22	195.4	1 534.25	195.7	1 531.90
4	195.10	1 536.61	195.3	1 535.04	195.5	1 533.47
5	195.05	1 537.00	195.2	1 535.82	195.3	1 535.04
6	195.00	1 537.40	195.1	1 536.61	195.1	1 536.61
7	194.95	1 537.79	195.0	1 537.40	194.9	1 538.19
8	194.90	1 538.19	194.9	1 538.19	194.7	1 538.77



▲图4 更高速无源光网络动态通道绑定数据传输示例[6]

上的传输。每一个业务数据帧在每个通道上的数据单元序列都需要增加一个XGEM帧头。因此，带宽效率会下降。特别是当业务数据帧较短时，带宽浪费更加明显。如表4所示，在短包长情况下，例如包长为100或者200字节时，2通道和4通道的带宽效率与单通道相比下降明显。文献[8]基于抓取的互联网数据包进行数据包长统计分析，其中100字节长度的数据包较为典型。

▼表 4 不同包长在不同通道数情况的带宽效率

通道数	业务数据帧长/字节					
	64	100	200	500	1 000	1 500
1	88.89%	92.59%	96.15%	98.43%	99.21%	99.47%
2	80.00%	86.21%	92.59%	96.90%	98.43%	98.94%
4	66.67%	75.76%	86.21%	93.98%	96.90%	97.91%

3 HSP动态通道绑定的改进研究

HSP灵活通道绑定目前存在的待改进问题包括：

1）带宽效率问题。每个业务数据帧都在多通道上传输，并且每个通道都需要增加XGEM帧头。这导致带宽效率下降。虽然IEEE Nx25G-EPON提供了一种业务数据帧连续传输方法，但是最前面的业务数据帧仍然需要在各通道上增加信封帧头。因此，如果业务数据帧是不连续的，那么每个业务数据帧的传输仍然需要在每个通道上增加帧头。

2）顺序恢复问题。HSP动态通道绑定只是做了一个数据单元顺序恢复的条件假设，并未提供实现机制。虽然IEEE Nx25G-EPON的EPAM机制可以在HSP中重用，例如XGEM帧头中有个18 bit的选项域可以用来定义类似EPAM功能，但是该机制要求每个通道都有帧头。这会对带宽效率产生影响。另外，在标准讨论过程中，有多个功能可能用到选项域，因此需要统筹考虑选项域的重定义。

在现有HSP动态通道绑定机制的基础上，并结合Nx25G-EPON动态通道绑定机制，我们利用HSP自身的技术特征，提出HSP动态通道绑定中的序列化传输和顺序恢复方法。

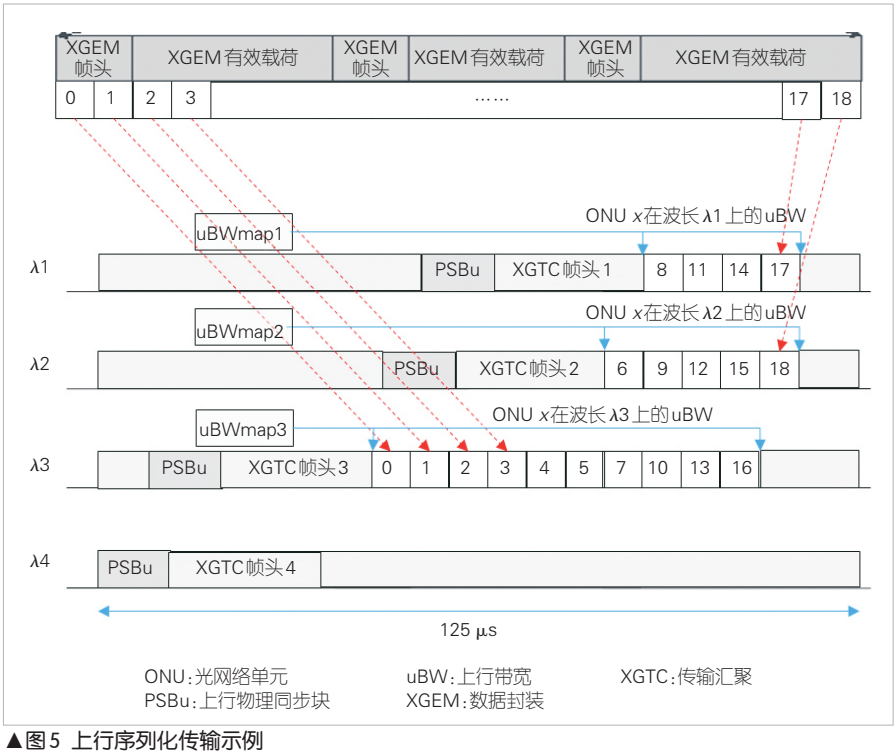
为了描述方便，我们总结了Nx25G-EPON和HSP动态通道绑定中的数据单元最早最小传输规则：业务层数据帧被分割成多个数据单元。各数据单元按照最早最小规则在多个通道上发送，即每个数据单元在传输机会最早且通道编号最小的通道上发送。接收端按照最早最小规则接收数据单元，即总是在有最早时隙和通道编号最小的通道上接收，并对接收到的数据单元进行重组，以恢复业务数据帧。

3.1 序列化传输[9]

在Nx25G-EPON的动态通道绑定机制中，对于属于同一个LLID的连续业务数据帧，除了第一个业务数据帧外，后续的每个业务数据帧的前导都将转化为一个ECH，而不需要在每个通道上增加ESH。实际上，即使是第一个业务数据帧也不需要在每个通道上增加ESH。这是因为当业务数据帧切分为EQ后，EQ序列会按照最早最小规则在多个通道上发送。接收端也按照最早最小规则恢复出同样的一个EQ序列，并恢复为业务数据帧。

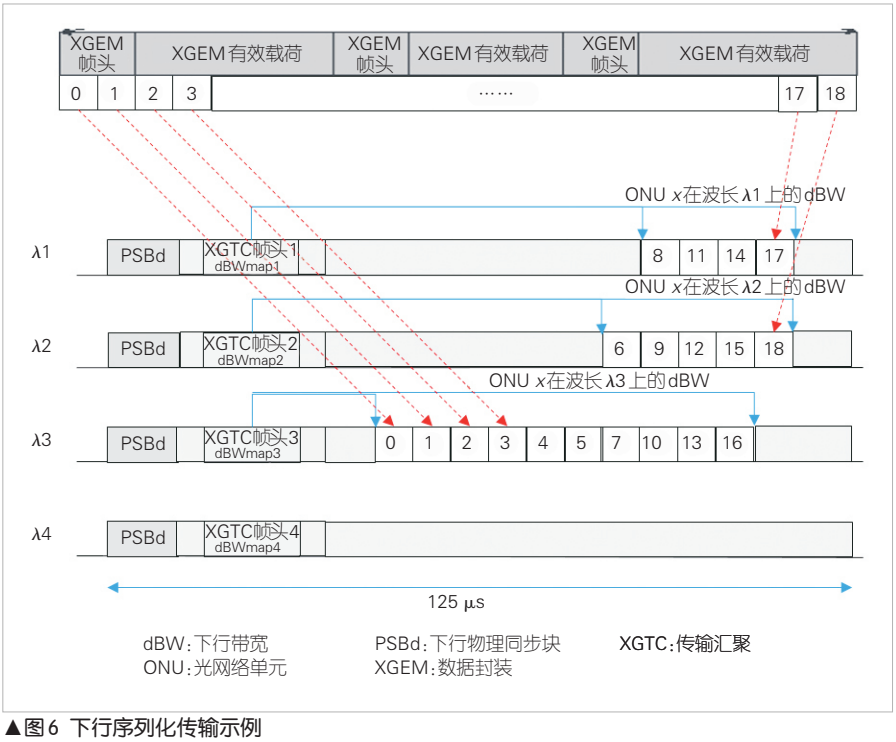
本文提出的序列化传输把业务数据帧封装成XGEM帧，并把属于同一个接收端的连续XGEM帧变成XGEM帧序列。XGEM帧序列被切分成数据单元。数据单元按照最早最小规则在多个通道上发送。接收端按照最早最小规则在多个通道上接收数据单元，并将其恢复成数据单元序列即XGEM帧序列。下面我们分别描述下行序列化传输和上行序列化传输。

图5给出了上行序列化的发送过程。其中，数据单元为4字节，用户数据帧和XGEM帧序列仅仅是示例，不影响上行序列化传输方法的实现。ONU从上行带宽分配结构uBWmap中收集各通道上的上行带宽，收集业务数据帧并



▲图5 上行序列化传输示例

将其封装在XGEM帧中，以形成XGEM帧序列。XGEM帧序列被分割成数据单元。每个数据单元按照最早最小规则在各个通道的上行带宽上发送。当所有这些带宽被数据单元填满后，ONU会在每个通道上构建XGTC帧。数据单元序列作为净荷被封装在XGTC帧中，进一步增加上行物理同步块PSBu和FEC校验。OLT在每个通道上解析PSBu和FEC校验，根据本地保存的uBWmap，获取每个ONU在各个通道上的上行带宽，并根据这些上行带宽按照最早最小原则，在各通道上逐个接收相应ONU发送的数据单元形成数据单元序列，并进一步组装成XGEM帧序列。由于现有标准ITU-T G.9804.2支持上行带宽分配结构，并且上行带宽分配结构是ONU从OLT发送的下行帧中获取的，因此上行序列化发送没有引入额外的开销，其带宽效率和单通道发送方式是一样的。



▲图6 下行序列化传输示例

图6展示了下行序列化发送过程。其中，数据单元为4字节，用户数据帧和XGEM帧序列仅仅是示例，同样不影响下行序列化传输方法的实现。为了支持下行方向的序列化发送方法，需要确定下行带宽的分配方式。文献[10]提出了下行带宽分配方法用于ONU节能。该方法暂时未写入ITU-T G.9804.2。我们参考该文献并引入下行带宽分配。OLT收集各ONU在每个通道上的下行带宽，将其组装成下行带宽分配结构dBWmap并发送。OLT进一步收集业务数据帧并将其封装成XGEM帧。XGEM帧被分割成数据单元。每个数据单元按照最早最小原则在各通道的下行带宽上发送。当所有这些各通道上的带宽填满数据单元后，OLT在每个通道上构建XGTC帧。数据单元序列作为净荷被封装在XGTC帧中，并进一步增加下行物理同步块PSBd和FEC校验。在接收侧，ONU在每个通道上解析PSBd和FEC校验，获取XGTC帧，从XGTC帧头中解析下行带宽分配结构dBWmap，并收集属于自己的各通道上的下行带宽。在这些下行带宽中，ONU会按照最早最小原则从各个通道逐个接收数据单元形成数据单元序列，并进一步组装成XGEM帧序列。由于下行带宽分配结构还不是标准ITU-T G.9804.2中的功能结构，因此，引入下行带宽分配结构会引入额外的开销。参考上行带宽分配结构，一个通道上的下行带宽分配由一个带宽条目指示。一个带宽条目为8字节。这意味着每个通道需额外引入8字节开销。

本文所述序列化传输方法是指，将单通道传输的XGEM帧序列分布到各个通道上传输，不需要在每个通道的数据单元序列前增加XGEM帧头。这使得上行序列化传输的带宽效率和单通道传输一样，避免了动态通道绑定导致的带宽效率下降。对于下行序列化传输的带宽效率，由于下行带宽分配未写入标准，引入下行带宽分配会增加额外的开销。如表5所示，随着连续帧数的增加，带宽效率逐步接近单通道传输效率。当然，如果下行带宽分配将来被写入ITU-T G.9804.2标准，下行序列化传输的带宽效率将和单通道传输一样。

3.2 顺序恢复[11-12]

HSP数据发送有严格的同步机制。下行方向每125 μs传输一个超帧，每个超帧头部均携带PSBd。顺序恢复机制可以利用这一特性。如图7所示，OLT在各通道上同步发送PSBd，且携带相同的超帧号（SFC），作为各通道发送数据的共同参考点。每个数据单元到同步PSBd的距离决定了自身的发送顺序。离同步PSBd近的数据单元发送时间早，离同步PSBd远的数据单元发送时间晚。对于离同步PSBd一样远近的数据单元，根据最早最小原则，在编号更小通道上的数据单元发送时间更早。

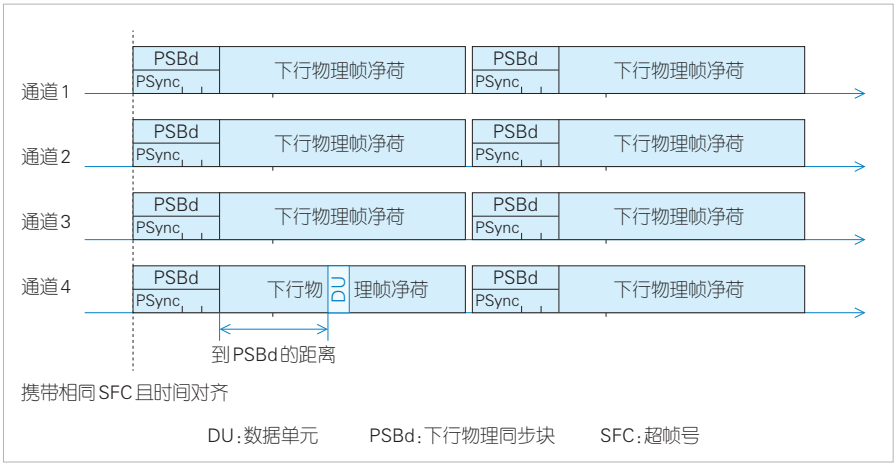
上行PHY帧和下行PHY帧是同步的。根据前文所述，

▼表 5 下行序列化传输、上行序列化传输、Nx25G-EPON在不同连续帧数情况下的带宽效率汇总

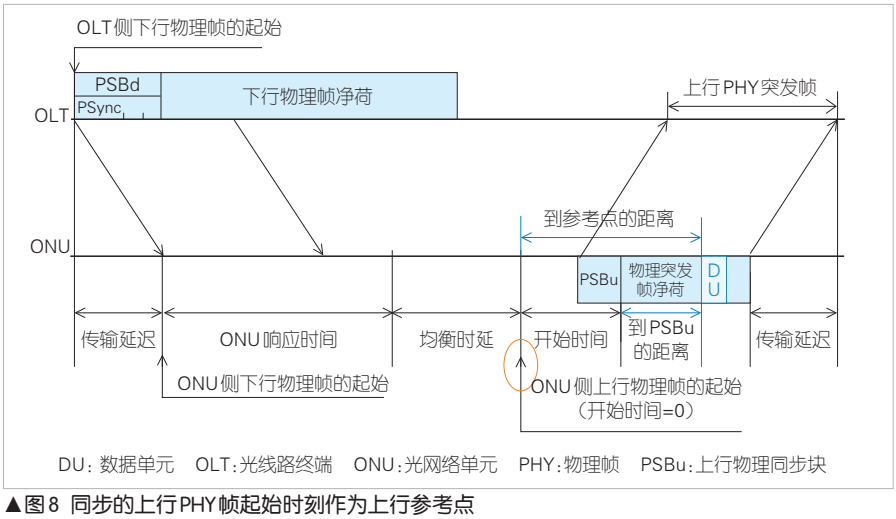
序列化传输方法	通道数	64字节帧长			100字节帧长			200字节帧长			500字节帧长		
		1帧	5帧	10帧	1帧	5帧	10帧	1帧	5帧	10帧	1帧	5帧	10帧
下行序列化	1	80.00%	86.96%	87.91%	86.21%	91.24%	91.91%	92.59%	95.42%	95.79%	96.90%	98.12%	98.27%
	2	72.73%	85.11%	86.96%	80.65%	89.93%	91.24%	89.29%	94.70%	95.42%	95.42%	97.81%	98.12%
	4	61.54%	81.63%	85.11%	71.43%	87.41%	89.93%	83.33%	93.28%	94.70%	92.59%	97.20%	97.81%
上行序列化	n(1～4)		88.89%			92.59%			96.15%			98.43%	
Nx25G-PON	1	88.89%	88.89%	88.89%	92.59%	92.59%	92.59%	96.15%	96.15%	96.15%	98.43%	98.43%	98.43%
	2	80.00%	86.96%	87.91%	86.21%	91.24%	91.91%	92.59%	95.42%	95.79%	96.90%	98.12%	98.27%
	4	66.67%	83.33%	86.02%	75.76%	88.65%	90.58%	86.21%	93.98%	95.06%	93.98%	97.50%	97.96%

由于各通道的下行帧是同步的，所以各通道的上行帧也是同步的。各通道上行PHY帧的开始时刻可以作为上行数据接收顺序的参考点，如图8所示。各个通道上数据单元到上行参考点的距离，包括StartTime加数据单元到PSBu的距离之和。这个距离在上行传输过程中是不变的，可用于表示数据单元的发送顺序。因此，上行方向的发送不需要做修改。

本文所述的顺序恢复方法充分利用了HSP中下行同步特征，不需要在帧结构中增加额外的字段，简化了动态通道绑定机制中的顺序恢复处理，也保留了XGEM帧头中选项域以用于其他功能扩展。



▲图7 更高速无源光网络动态通道绑定的下行参考点即数据单元发送顺序示例



▲图8 同步的上行PHY帧起始时刻作为上行参考点

4 结束语

本文分析了现有无源光网络标准中的动态通道绑定需求、功能，以及HSP动态绑定机制中存在的带宽效率、顺序恢复等问题。结合Nx25G-EPON动态通道绑定机制，利用HSP自身的技术特征，本文提出了HSP动态通道绑定中的序列化传输和顺序恢复方法。该方法简化了动态通道绑定处理，避免了带宽效率下降。此外，针对本文提出的HSP动态通道绑定序列化传输和顺序恢复方法，我们已在在ITU-T标准组织进行提案，并将在HSP标准的后续增补中继续讨论。

参考文献

[1] IEEE. Physical layer specifications and management parameters for 25 Gbit/s and 50 Gbit/s passive optical networks: 802.3ca [S]. 2020

[2] ITU. 40-Gigabit-capable passive optical networks 2 (NG-PON2): physical media dependent (PMD) layer specification: ITU-T G.989.2 [S]. 2019

[3] ITU. 40-Gigabit-capable passive optical networks (NG-PON2): transmission convergence layer specification: ITU-T G.989.3 [S]. 2015

[4] ITU. 40-Gigabit-capable passive optical networks (NG-PON2): general requirements amendment 1: ITU-T G.989.1 amd1 [S]. 2015

[5] ITU. 40-Gigabit-capable passive optical networks (NG-PON2): transmission convergence layer specification amendment 2: ITU-T G.989.3 amd2 [S]. 2018
[6] ITU. Higher speed passive optical networks: common transmission convergence layer specification: ITU-T G.9804.2 [S]. 2021
[7] 张伟良, 黄新刚, 马壮. 基于专用激活波长的低时延50G-PON原理与实现 [J]. 中兴通讯技术, 2022, 27(4): 58-62. DOI: 10.12142/ZTETJ.202204012
[8] LUO Y. Channel bonding analysis [EB/OL]. [2024-02-25]. ftps://fsanftp@store. fsan. org/FSAN/FSAN ARCHIVE/FSAN GROUPS/NGPON/Meeting-Contributions/2019/2019-01 Plano/NG-PON2_Enhancements/HW_Bonding throughput.pdf
[9] ZHANG W L, YUAN L Q. Effective channel bonding framing based on bandwidth map in G.hsp.ComTC [EB/OL]. [2024-02-25]. https://www. itu. int/ifa/t/2017/sg15/exchange/wp1/q2/20-10-20_Phonecall/201020_D32_ZTE_Effective channel bonding framing based on bandwidth map in G.hsp.ComTC.docx
[10] FRANK E. G. HSP: downstream extra features [EB/OL]. [2024-02-25]. https://www. itu. int/ifa/t/2017/sg15/exchange/wp1/q2/20-06-09_Phonecall/200609_D19_FW_DownstreamTopics.docx
[11] ZHANG W L, YUAN L Q. Order recovery for channel bonding in G. hsp. comTC [EB/OL]. [2024-02-25]. https://www. itu. int/ifa/t/2017/sg15/exchange/wp1/q2/20-07-06_Multicall/200706_D55_ZTE_order recovery for channel bonding in G. hsp. comTC_v1.0.docx
[12] ZHANG W L, YUAN L Q. Text proposal of PSBd synchronization for channel bonding in G. hsp. comTC [EB/OL]. [2024-02-25]. https://www. itu. int/ifa/t/2017/sg15/exchange/wp1/q2/20-10-20_Phonecall/201020_D31_ZTE_Text proposal of PSBd synchronization for channel bonding in G.hsp.comTC.docx

作 者 简 介

张伟良，中兴通讯股份有限公司固网团队技术预研资深专家、低时延PON技术负责人；长期从事光接入、家庭网络产品的技术预研、产品规划和标准化工作；主持并参与多项国家“863”项目、省部级重点项目；获得中国专利优秀奖、深圳市专利奖、中国标准创新贡献奖；发表论文10余篇，获得授权专利100余项。

王霄雨，中国电信集团高级项目经理；主要研究方向为接入网络关键技术，包括无源光网络、业务体验感知等。

黄新刚，中兴通讯股份有限公司固网团队技术预研资深专家；长期从事光接入技术研究和标准化工作；主持并参与多项国家“863”项目、省部级重点项目；获得国家科学技术进步奖二等奖一项、电子学会科学技术进步奖一等奖一项、深圳市科技进步奖二等奖一项；获得发明专利10余项。

中兴通讯技术杂志社
促进产学研合作青年专家委员会

刊物相关信息

投稿须知　投稿平台　过刊下载　论文索引与引用指南

中兴通讯技术

(ZHONGXING TONGXUN JISHU)

办刊宗旨：

以人为本，荟萃通信技术领域精英
迎接挑战，把握世界通信技术动态
立即行动，求解通信发展疑难课题
励精图治，促进民族信息产业崛起

双月刊　1995 年创刊
第 30 卷　总第 175 期
2024 年 4 月　第 2 期

主管：安徽出版集团有限责任公司
主办：时代出版传媒股份有限公司
　　深圳航天广宇工业有限公司
出版：安徽科学技术出版社
编辑、发行：中兴通讯技术杂志社

总编辑：王喜瑜
主编：王利
执行主编：黄新明
编辑部主任：卢丹
责任编辑：徐烨
编辑：杨广西、朱莉、任溪溪
设计排版：徐莹
发行：王萍萍
编务：王坤

《中兴通讯技术》编辑部
地址：合肥市金寨路 329 号凯旋大厦 1201 室
邮编：230061
网址：tech.zte.com.cn
投稿平台：tech.zte.com.cn/submission
电子信箱：magazine@zte.com.cn
电话：（0551）65533356

发行方式：自办发行
印刷：合肥添彩包装有限公司
出版日期：2024 年 4 月 30 日
中国标准连续出版物号：ISSN 1009-6868
CN 34-1228/TN
定价：每册 20.00 元