计算机研究与发展
Journal of Computer Research and Development
ISSN 1000-1239,CN 11-1777/TP

《计算机研究与发展》网络首发论文

题目： 面向边缘智能的大模型研究进展
作者： 王睿，张留洋，高志涌，姜彤雲
网络首发日期： 2025-01-27
引用格式： 王睿，张留洋，高志涌，姜彤雲．面向边缘智能的大模型研究进展[J/OL]．计算机研究与发展. https://link.cnki.net/urlid/11.1777.TP.20250127.0921.006

面向边缘智能的大模型研究进展

王睿　张留洋　高志涌　姜彤雲

（北京科技大学计算机与通信工程学院 北京　100083）

（wangrui@ustb.edu.cn）

Research Progress on Large Models for Edge Intelligence

Wang Rui, Zhang Liuyang, Gao Zhiyong and Jiang Tongyun

（*School of Computer and Communication Engineering, University of Science and Technology Beijing, Beijing* 100083）

Abstract With the rapid development of large-scale model technology, these models have exhibited remarkable performance in fields such as natural language processing and computer vision, becoming essential tools for addressing complex issues and drawing significant interest from both the scientific community and the industry. Nonetheless, current cloud-platform-based schemes for training and inference of large models face multiple challenges, including high expenses, restricted scalability, and information security risks. As the scale of model parameters expands continually, the need for low-cost, efficient training and inference methods grows ever more pressing. Carrying out collaborative training and inference of large models on edge devices can dramatically decrease latency and bandwidth demands, concurrently reinforcing data privacy and operational efficiency. This strategy furnishes vital technological support for the economical deployment of large models across a variety of contexts, thereby evolving into one of the prominent research hotspots. This article conducts a thorough investigation of research pertinent to large models in the context of edge intelligence, with an in-depth analysis and discourse primarily focused on two aspects: edge-based training and inference of large models. Ultimately, it outlines the challenges confronted in the progression of large model technologies tailored for edge intelligence and delineates future prospects. The ambition is to stimulate a heightened comprehension and intensified attention from both academic and industrial sectors towards technologies involving large models for edge intelligence, thereby encouraging further scholarly exploration in this thriving domain.

Key words　edge intelligence; large models; federated fine-tuning of large models; edge efficient inference

摘要　随着大模型技术的迅猛发展，大模型在自然语言处理和计算机视觉等领域表现出卓越的性能，成为解决复杂问题的重要工具，并在科研和产业界引发了广泛关注.然而，当前基于云平台的大模型训练和推理方案面临诸多挑战，包括高昂的成本、有限的可扩展性和信息安全风险等.随着模型参数规模的不断扩大，对于低成本、高效训练和推理的需求愈发迫切.在端边侧进行大模型的协同训练和推理，可以显著降低延迟和带宽需求，同时增强数据隐私和操作效率，为大模型在多样化场景中的低成本应用提供关键技术支持，成为当前研究的热点之一.全面调研了面向边缘智能的大模型相关研究，主要从大模型边缘训练和推理 2 个角度对当前相关研究进行了深入分析和讨论.最后，提出了面向边缘智能的大模型技术发展所面临的挑战和未来展望.希望能促进学术界和产业界对面向边缘智能的大模型技术有更深入了解和关注，并能够启发更多的学者开展深入研究.

关键词　边缘智能;大模型;大模型联邦微调;边缘高效推理

中图法分类号　TP393

DOI: 10.7544/issn1000-1239.202440385　　CSTR: 32373.14.issn1000-1239.202440385

自 2022 年底 OpenAI 发布 ChatGPT 以来[1]，众多　　高性能的开源大模型接连发布，掀起了全球范围内前

所未有的大模型浪潮.例如大语言模型 GPT-4[2]、LLaMA[3]、多模态大模型 LLaVA[4]、视觉大模型 SAM[5]，这些大模型相较传统模型具有强大泛化能力，并呈现出了许多传统模型不具备的涌现能力，在自然语言处理和计算机视觉等领域的表现出卓越的性能.这种卓越表现主要体现在的通用性与灵活性上，使得大模型成为解决复杂问题的重要工具，并在教育、医疗、文本生成等各个领域中均展现出非凡潜力，引起了科研界和产业界的广泛关注.

大模型优秀的性能体现在其巨大的参数规模上，但是其参数量具有逐渐提高的趋势，如图 1 所示，为模型的训练和推理带来巨大的挑战.庞大的参数规模使得大模型需要在云端才能完成训练与推理部署，而当前基于云平台的大模型训练和推理方案面临诸多挑战，包括高昂的成本、有限的可扩展性和信息安全风险等.例如 LLaMa2-70B 的预训练在具有 760 个 A100 GPU 节点的超级集群上完成，总计消耗 172 万个 GPU 时[6]，半精度下推理则至少需要 140 GB 的显存容量.随着模型参数规模的不断扩大，对于低成本、高效训练和推理的需求显得愈发迫切.

注：OpenAI 并未公布 GPT-4 的规格，此处采用估计参数量

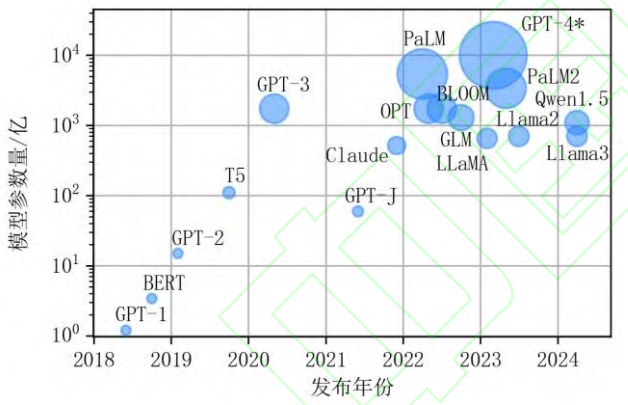


Fig. 1 Trend of parameter growth in large models

图 1 大模型参数量增加趋势

边缘智能在近年来随着物联网、云计算和大数据技术的融合与发展逐渐受到关注[7]，在降低大模型成本上潜力巨大，成为推动大模型实现技术普惠的重要手段.边缘智能指的是在网络的边缘侧，即数据源附近，进行数据处理和分析的能力.与传统的云计算模式相比，边缘智能强调在设备或终端上直接进行计算和决策，从而实现对数据的实时、高效处理.然而，大模型通常需要在云端或高性能计算环境中运行，这可能导致数据传输延迟和带宽限制等问题.而边缘智能则可以在设备端或网络边缘进行数据处理和分析，减少了数据传输的延迟和带宽需求.大模型和边缘智能可以相互协同，实现更高效的数据处理和分析.例如，大模型可以在云端进行全局优化和决策，而边缘智能则可以在设备端进行局部优化和实时响应.这种协同优化可以进一步提高系统的性能和效率.目前国内部分头部手机厂商已经初步实现移动端嵌入大模型，Apple 公司研发部门也提出了有限内存下大模型部署方法[8]，有望将大模型融入其产品中.可以预见大模型逐步进入移动端，甚至 IoT 生态中是未来不可避免的趋势.

有关大模型边缘智能化研究方向可以被分为训练与推理 2 部分.边缘侧的大模型训练，通常指的是大模型的微调训练，该过程主要包含参数高效微调和全微调 2 种策略.参数高效微调旨在通过调整大模型中的一小部分参数，以适应新的任务或数据需求.这一方法通过深入研究不同的训练参数配置或在大模型中嵌入特定模块，以实现高效的大模型微调.而全微调训练则涉及到大模型中所有参数的调整，以全面更新模型.鉴于数据隐私性的重要性和边缘设备的异构性特征，参数高效微调往往与联邦学习相结合，探究在联邦学习框架下如何有效提升不同参数高效微调方法的性能.大模型推理优化技术利用大模型自身特点加速模型推理速度，包括服务优化、通用优化以及设备优化.边缘侧大模型推理方法包括适用于边缘侧通用优化和设备优化，可以被概括为先于部署的模型压缩方法、部署后的模型层面推理优化、以及部署后的系统层面推理优化.此外部分工作致力于提供大模型的边缘侧部署方案，这些工作通常结合上述一种或多种大模型的推理加速方法，为大模型在边缘侧部署提供便利.

本文从边缘智能视角，全面调研了面向边缘智能的大模型相关研究，聚焦于大模型边缘训练和推理 2 个角度，对当前相关研究进行了深入分析和讨论.最后，给出了面向边缘智能的大模型技术发展所面临的挑战和未来展望.

1 大模型边缘训练的研究进展

目前在边缘侧进行大模型训练的研究主要为大模型微调，在传统微调策略中，对大型预训练模型的全部参数进行优化以适应新任务，通常在大模型场景下显得耗时且资源密集.而参数高效微调方法通过仅修改或添加少量额外参数来实现模型的微调，同时保持模型主体的参数固定不变，从而节省资源、加速训练过程并促进模型在边缘场景下的有效应用. 然而，传统的中心化微调方法要求所有数据汇聚至中心节

点，这不仅增加了数据泄露的风险，还可能导致隐私侵犯.而联邦学习[9]作为一种新兴范式，允许数据保留在边缘设备上，仅模型的更新在加密或经过处理后被发送到中央服务器，从而显著降低了隐私泄露的风险，并满足了 GDPR[10]等严格隐私保护法规的要求.然而边缘设备往往受限于有限的计算资源、通信资源和存储资源，大型模型的庞大参数对边缘设备构成了沉重的计算与存储负担，同时也导致了高昂的通信成本.因此在联邦学习中引入参数高效微调方法，成为解决上述问题的有效途径.本章节将聚焦联邦参数高效微调方法，从边缘智能的视角出发，对大型模型的微调技术进行深入探讨和细致分析.

1.1 大模型边缘微调的整体流程

大模型参数量规模的迅速增加使得模型在有限算力资源条件下的训练和微调更具有挑战性，为实现高效大模型预训练，现有研究从内存高效和数据高效 2 方面加速模型训练或降低训练成本.然而模型规模的增大使得训练在费用、时长、能耗、数据量、硬件资源等方面的需求愈发强烈，例如 175B 的 GPT3 模型单次训练费用高达 460 万美元[11]，使用 4 990 亿个 token[12]，预计到 2027 年大模型训练成本最大花费可达 10 亿美元[13]，使得有限资源下的大模型预训练愈发困难.微调作为一种可以将模型适配到特定领域的技术，因其无需重新训练模型参数而受到学术界和工业界广泛关注.现有大模型微调技术研究可被分类为全参数微调和参数高效微调 2 方面，全参数微调通过微调大模型所有参数以取得更好的微调表现[14]，因此通常导致微调开销偏大，部分研究通过优化微调过程中的参数更新步骤来降低内存需求[15-16].参数高效微调旨在减少模型参数更新量，仅需更新模型中的部分参数，同时保持良好的性能，以达到适应下游任务的目的[17].参数高效微调方法的关键在于选择哪些参数进行微调，以及如何设计有效的微调策略来优化这些参数.与全参数微调方法相比，参数高效微调技术具有以下优点：

1） 减少计算资源消耗. 由于只需要更新模型的一小部分参数，因此可以大大减少计算资源的消耗，加快训练速度.

2） 降低过拟合风险. 通过仅调整模型的一部分参数，可以降低过拟合的风险，提高模型的泛化能力.

3） 更好的可扩展性. 参数高效微调技术可以更容易地应用于不同的预训练模型和任务，具有较好的可扩展性.

在大多数现实世界场景中，目前的微调主要以集中式的方法将数据集中在一起进行集中微调.在边缘智能场景下进行大模型微调通常存在以下几个挑战：

1）边缘智能环境下的数据通常包含敏感信息，集中式微调会造成敏感数据外泄风险.如何保证数据的安全和隐私，同时进行有效的模型训练，是一个需要解决的问题.

2）边缘设备的带宽和存储资源往往有限难以满足微调的计算和通信需求，大模型的微调需要大量的计算资源和存储空间，这使得在边缘设备上直接进行微调变得困难.

3）边缘设备可能需要处理多种不同的任务，这些任务可能有不同的数据分布和需求.同时边缘设备可能只有有限的数据可用，这可能导致模型微调时的过拟合或不足.

因此目前针对边缘侧设备进行微调主要采取将联邦学习与参数高效微调技术相结合的方法[18]，通过联邦学习，可以在不同数据源上协同训练模型，这有助于提高模型对不同数据集的适应性和泛化能力，并且使得数据处理过程主要在本地完成，避免了敏感数据的外泄风险，强化了隐私保护，同时使用参数高效微调技术减少了大量数据的传输，从而降低了通信成本，特别是在移动设备或边缘计算场景下尤为重要.联邦高效参数微调技术不仅能够提高模型的性能和泛化能力，还能在保护隐私、降低计算和通信成本、提高响应速度和能源效率方面发挥重要作用.这种结合方法为大模型的微调训练和应用提供了一种更加高效、安全和可持续的途径，特别适合于分布式和资源受限的边缘计算环境，联邦参数高效微调的流程如图 2 所示. 首先，服务端冻结预训练大模型的参数 W_p，依据特定的参数高效微调策略确定可训练全局参数 W_c^t，并进行相应的初始化.随后，服务端将可训练全局参数 W_c^t 分发给各边缘设备.在边缘设备上，基于本地数据集对可训练全局参数 W_c^t 进行更新，得到本地参数 $W_c^{k,t}$，并将这些参数上传至服务端.服务端接收各边缘设备的本地参数后，进行聚合更新，进而优化大模型，得到新的全局参数 W_c^{t+1}. 此过程循环往复，直至模型收敛或达到预定的训练轮次.整个流程的关键阶段涵盖预训练大模型的初始化和参数冻结、边缘设备基于本地数据集的参数微调、以及服务端对参数的聚合与全局模型的优化.这一流程在确保数据隐私和边缘设备资源有效利用的同时，有效提升了模型的性能.

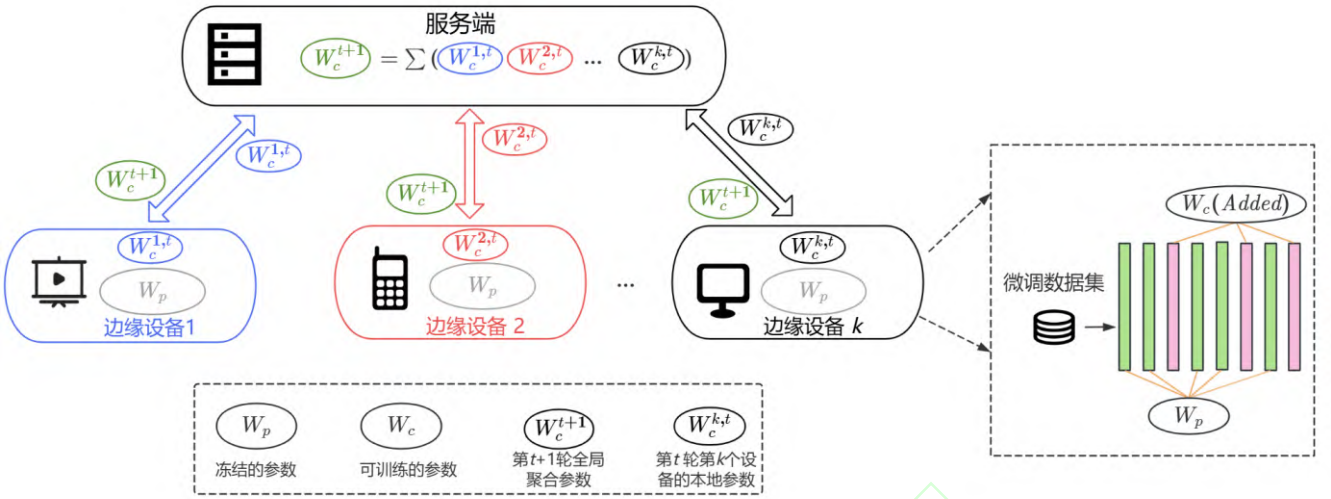


Fig. 2 Federated efficient fine-tuning framework for large model

图 2 大模型联邦参数高效微调框架

1.2 大模型边缘微调方法

参数高效微调技术旨在仅通过调整模型的一小部分参数来适应新的任务或数据，从而提高模型的性能和泛化能力，同时减少计算和存储资源的消耗.该技术现有研究主要包括适配器微调[19-21]、前缀微调[22-23]、提示词微调[24-26]、低秩适配[27-28]等方向，表 1 展示了上述几种方法的更新的参数和近期工作.

Table 1 Comparison of Different Fine-Tuning Methods

表 1 不同微调方法的比较

大模型微调技术	相关工作	更新的参数
适配器微调	文献[19-21]	适配器模块
前缀微调	文献[22-23]	“前缀”向量序列
提示微调	文献[24-26]	提示词向量
低秩适配	文献[27-28]	低秩矩阵
全微调技术	文献[14-16]	大模型所有参数

1）适配器微调

适配器微调技术通过向 LLM 插入瓶颈架构的可训练神经网络模块（即适配器 Adapter），达到有效减少可训练参数量的目的.这些适配器通常只占据原始模型大小的一小部分，但能有效地进行任务适配.它们被插入到预训练模型的每一层或者特定的一些层中，以学习特定下游任务的知识.适配器由 2 个前馈子层构成，第 1 个子层负责将输入维度从原始的 d 投影到一个较小的维度 m，以此来限制适配器模块的参数量.第 2 个子层则将维度 m 重新投影回 d，输出作为适配器模块的结果.

2）前缀微调

前缀微调技术是一种轻量级的微调方法，它通过在输入序列前添加 1 个连续的、特定于任务的向量序列来实现模型的快速适应.前缀微调技术的关键在于向模型的输入中引入一个称为“前缀”的向量序列.这个前缀是由自由参数组成的，不与任何实际的词汇单元对应，它能够为模型提供关于当前处理任务的信息.这种方法的核心优势在于，它只需要训练非常少的额外参数，有时甚至只需 0.1%的参数，就能实现与传统微调相当甚至更优的性能.

3）提示微调

提示微调相关研究在模型的嵌入层加入可训练的提示词向量，将适应下游任务的提示嵌入与文本输入嵌入整合以实现模型适应下游任务并减少参数训练量的目的.该技术主要依赖预训练语言模型的强大表达能力和泛化能力.通过精心设计的提示词，可以激活模型中与任务相关的知识，使模型能够更好地理

解和处理特定任务的数据.

4）低秩适配

低秩适配微调技术是一种高效微调预训练大型语言模型的方法，它主要通过在模型的权重矩阵上添加 1 个低秩矩阵来实现对新任务的快速适应.低秩适配微调技术的核心思想是在预训练模型的权重矩阵中引入 1 个低秩矩阵，这个矩阵可以在微调过程中更新以学习特定任务的信息.这种方法的总体思想和概念与主成分分析（principal component analysis，PCA）和奇异值分解（singular value decomposition，SVD）有关，它们都是利用低维表示来近似高维矩阵或数据集.

大模型联邦参数高效微调方法主要基于上述不同的参数高效微调方法对其联邦化，在联邦过程中会面临着隐私和异构等联邦学习中存在的挑战，现有工作对上述挑战做了初步的探讨和研究.另外部分工作对大模型联邦微调框架的实现进行了研究和开发，旨在构建 1 套完整的大模型联邦微调流程和基准.表 2 展示了现有大模型联邦微调工作，并根据其支持的高效参数微调方法、隐私保护和异构问题等方面进行了对比.

Table 2 Summary of Federated Efficient Fine-tuning Framework for Large Models

表 2　大模型联邦高效微调框架总结

大模型联邦微调框架	支持的参数高效微调方法	隐私保护	异构问题
FedPEAT[29]	适配器调优	√	√
FedPepTAO[30]	提示调优	×	√
SLoRA[31]	低秩适配	×	√
FedPETuning[32]	适配器调优、提示调优	√	×
FederatedScope-LLM[33]	前缀调优、提示调优、低秩适配	√	×
FATE-LLM[34]	适配器调优、提示调优	√	√

在适配器调优方面，通过联邦学习方法对不同客户端的大模型层中适配器进行协同训练，通过传递适配器层进行聚合和分发来降低计算和通信带宽的成本.文献[29]提出了一种将离线调优方法推广到仿真器辅助调优（emulator - assisted tuning，EAT），并将其与参数高效微调相结合，创建参数高效仿真器辅助调优，将其应用扩展到联邦学习中，适配器具有可训练的神经网络参数，为特定任务定制预训练模型，而模拟器提供原始模型的压缩固定参数表示.这种组合不仅通过避免将完整模型传输到移动边缘设备来解决模型隐私问题，而且还显着提高了内存和计算效率.此外，最近也有研究将联邦微调应用在视觉大模型和跨模态大模型的训练阶段，例如 FedDAT[35]提出针对异构多模态联邦学习的微调框架，利用双适配器结构和教师模型组成的双适配器教师模块（dual-adapter teacher，DAT）来处理数据异构性，并通过规范客户端本地更新和应用相互知识蒸馏以实现高效的知识转移，是首个能够高效分布式微调基础模型以适应多种异构视觉-语言任务的方法.

在提示调优方面，文献[30]提出了一种参数高效的自适应优化提示调优方法，利用联邦学习（federated learning，FL）调优大型语言模型.由于在所有提示层中传递整个参数集对应于沉重的通信成本，提出了一种根据每一层的重要性选择适当的提示层的高效方法.同时设计了一种评分方法，根据各层对最终收敛精度的调优影响来识别各层的重要性.PromptFL[36]提出基于提示的训练框架来替换传统模型训练中训练整个共享模型的方法，只更新和传输提示，保留了 CLIP 模型的强大适应性和泛化能力，大幅减少了联邦学习的通信需求并提升了模型性能，保护用户隐私.

在低秩适配调优方面，由于 FL 中最大的挑战之一是在异构客户端分布场景下训练时性能下降，因此文献[31]提出了一种新颖的数据驱动初始化技术克服了 LoRA 在高异构数据场景中的关键限制，它包括 2 个阶段，首先客户使用完全微调技术协作更

新模型找到一个成熟的起点（初始化器）来启动 LoRA 块，然后使用上一阶段学习到的初始化器运行 LoRA 算法.SLoRA 实现了与完全微调相当的性能，具有大约 1%密度的显著稀疏更新，同时将训练时间减少了 90%.

在框架实现方面，由于大模型联邦微调的发展仍处于不成熟阶段，现有研究工作对大模型联邦微调算法的全面实现和基准研究不足，因此文献[32]提出了联邦参数高效微调框架，并为适配器调优、前缀调优和低秩调优方法开发了相应的联邦基准，同时测量了隐私保护能力、性能和资源成本，证明了将预训练大模型与 FL 相结合的潜力，为大模型时代的隐私保护学习提供了 1 个有前途的训练范式；文献[33]提出了 1 个基于 FederatedScope 的大模型联邦微调框架 FederatedScope-LLM，该框架封装了来自不同领域的各种联邦微调数据集的集合，具有可调的数据异构级别和 1 套相应的评估任务，以形成 1 个完整的管道，以基准测试 FL 场景中的联邦微调大模型算法，提供了全面的联邦微调算法，具有较低的通信和计算成本以及通用的编程接口，支持客户端可以或不能访问完整模型的 2 种场景；文献[34]提出了一个工业级大模型联邦微调框架 FATE，该框架支持同构和异构大语言模型的联邦微调训练，通过适配器调优、前缀调优等多种参数高效的微调方法促进 FedLLM 的高效训练，同时采用联邦知识产权保护方式保护大模型的知识产权以及通过隐私保护机制保护训练和推理过程中的数据隐私.

2 大模型边缘推理方法与架构

大模型推理与部署是完成大模型边缘化的重要步骤，在模型的推理过程中，大模型庞大的参数规模产生了巨大的算力、内存、带宽等资源的消耗，令大模型的边缘化过程困难重重.

现有文献综述[37-39]对大模型推理优化的调研并未有效区分边端侧推理与云端推理相关技术，并且更关注大模型的云端推理框架.相比之下，本章节提供了边缘推理视角下的大模型优化加速与部署框架的调研，通过挑选具有代表性且适用于边缘设备推理的最新研究进展，系统性总结近年来大模型推理流程优化的创新工作，并提出相关见解.需要指出的是，我们仅对软件层面的优化工作进行了调研，硬件加速方面的工作已有详细总结[40-42]，虽然本节不包括硬件相关工作，但是这些研究在大模型迈向边缘的道路上同样起到了不可或缺的作用.

2.1 大模型边缘推理的整体流程

在边缘智能的一般范式中，在应用到推理场景之前需要结合多种优化方法对模型进行进一步处理[7]，并利用计算卸载、资源分配、协同等关键技术实现边缘侧优化目标[43].然而模型参数规模的膨胀使得这些技术在边缘侧设备的应用效果越来越不明显，为此需求对大模型进行针对性的优化，以弥补巨大的算力等资源需求与边缘侧低资源设备的间隙.图 3 展示了主流大模型边缘推理的一般流程，在预训练模型正式部署服务之前，大模型边缘化关键技术从多种角度对大模型的资源消耗作出优化，大模型部署框架整合上述技术，并结合目标部署环境提供系统级资源优化能力和友好用户接口.

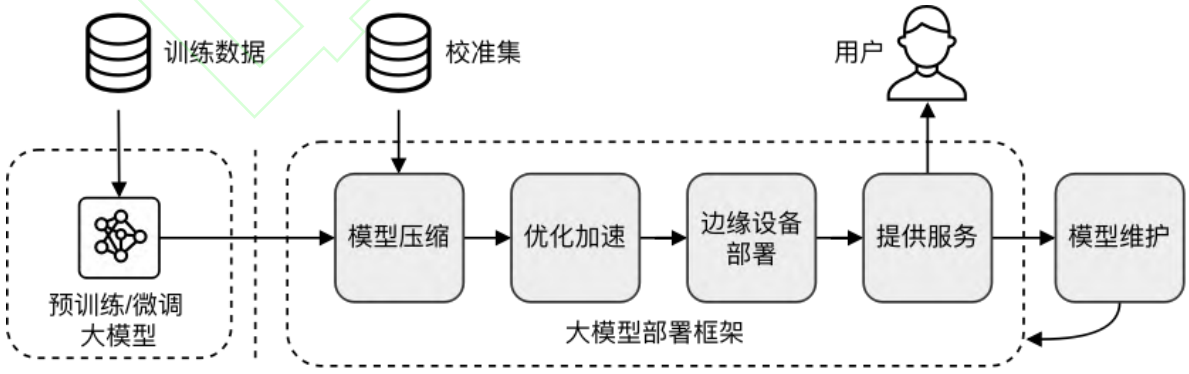


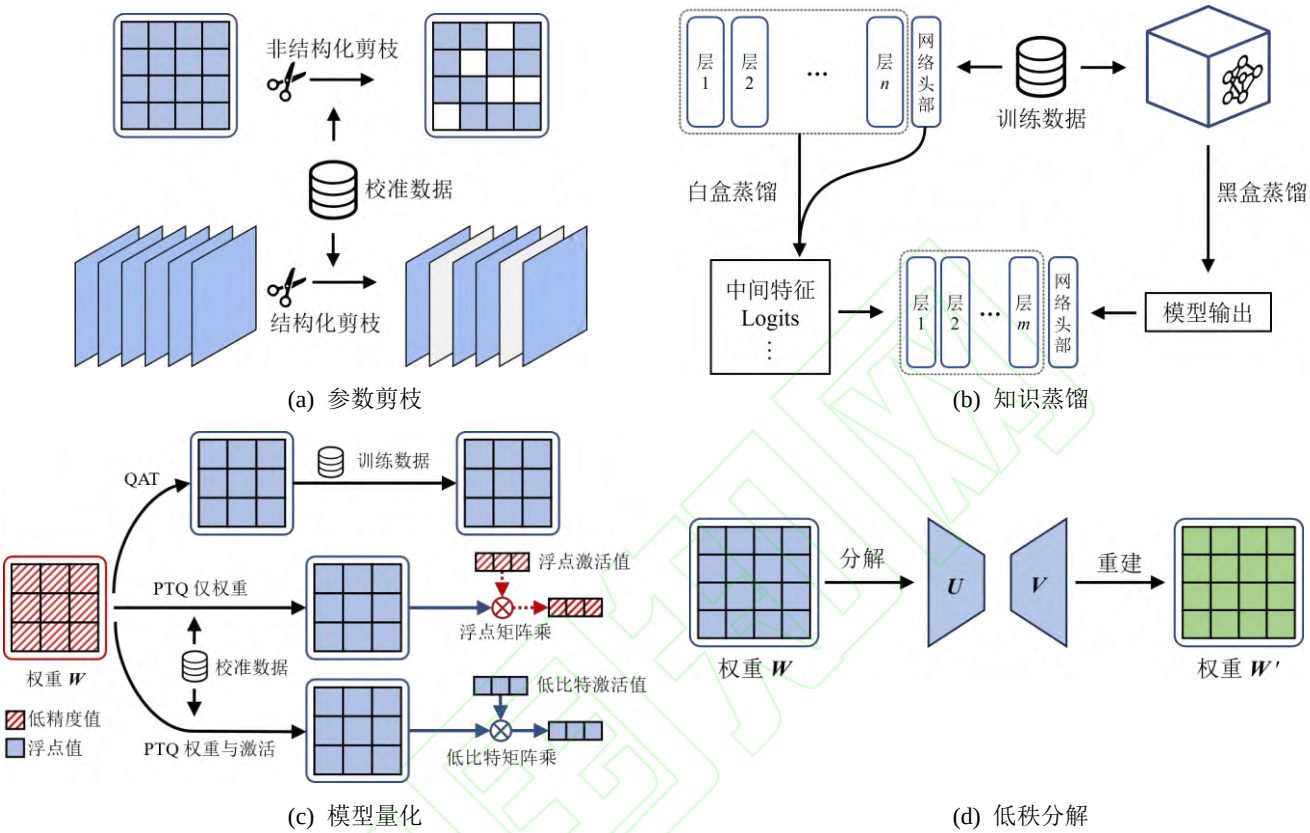
Fig. 3 Process of large model inference for edge

图 3 大模型边缘推理流程

现有主流大模型边缘化关键技术可概括为大模型压缩技术与大模型推理加速技术，我们将在后续与大模型边缘部署框架一同详细分析.

2.2 大模型压缩

一般神经网络的压缩技术可以分为参数剪枝、知识蒸馏、模型量化、低秩分解 4 个方向，目的是减少模型计算与存储等资源的消耗.然而与一般神经网络不同的是，大模型具有架构庞大，算力需求高、访存量多、泛化能力强等特点，使得一般性的模型压缩方法在大模型上效率或效果不佳[44].为了应对这些挑战，许多大模型专用的模型压缩方法被提出，我们在图 4 展示了这些工作不同方向的技术概况，在表 3 详细展示了上述几种方向的分类、优化目标以及近期相关工作.



(a) 参数剪枝　(b) 知识蒸馏

(c) 模型量化　(d) 低秩分解

Fig. 4 Overview of large model compression techniques

图 4 大模型压缩技术概况

Table 3 Classification of Related Work for Large Model Compression

表 3 大模型压缩关键技术相关工作分类

参数剪枝	结构化剪枝	剪除冗余结构，降低模型大小和计算复杂度	文献[45-48]
	非结构化剪枝	实现权重稀疏化，减小模型内存使用量和计算量，依赖特定软硬件加速模型张量运算	文献[49-51]
知识蒸馏	白盒蒸馏	产生特定领域下的小模型，减少模型尺寸和计算量，同时保持模型在特定任务下的性能	文献[52-60]
	黑盒蒸馏	在不访问大模型内部结构的情况下，实现蒸馏过程，产生特定领域的小模型	文献[61-70]
模型量化	训练后量化	降低模型存储大小、节省存储、内存、带宽、计算量，同时保持模型精度	文献[71-81]
	量化感知训练	降低模型量化误差，在降低模型存储、内存、带宽、计算量的前提下，进一步保持模型精度	文献[82-86]
低秩分解	—	减少模型参数量，实现推理加速	文献[87-91]

“—”表示没有更细致的类别划分.

1）参数剪枝

参数剪枝技术通过移除模型的冗余结构或权重压缩模型，按修剪粒度区分，现有大模型剪枝技术可分为结构化剪枝和非结构化剪枝.结构化剪枝移除大模型参数矩阵多个通道或块结构等结构化组成部分，部分研究工作[45-48]多于微调、量化、甚至训练相结合以降低精度损失、提高压缩效果.非结构化剪枝不考虑大模型内部结构，通过神经元级别的删减使模型权重矩阵产生稀疏性，依赖特殊的软硬件实现张量运算加速[92].一般的神经网络剪枝技术在剪枝后利用微调恢复模型的性能，然而由于大模型全参数微调的成本巨大，当前的大模型剪枝通常舍弃微调[50-51]步骤，或者结合参数高效微调[45,48]低成本.由于剪枝无可避免地损失了模型性能，并且大模型参数全量微调对硬件设施算力的要求极高，此类方法应用在大模型上的实用性仍需进一步优化.

2）知识蒸馏

知识蒸馏技术将大模型作为教师模型，利用教师模型的监督信息训练一个小型学生模型，针对大模型的现有研究可分类为白盒蒸馏和黑盒蒸馏 2 种.白盒蒸馏方法同时利用大模型的内部信息和输出训练学生模型，黑盒蒸馏方法假设大模型的内部结构不可见，仅利用教师模型的输出训练学生模.与一般神经网络的知识蒸馏不同，大模型蒸馏更关注知识的转移，而不是架构上的压缩[93].当大模型参数量达到一定程度后会表现出“涌现能力”，即处理复杂任务的表现惊人，利用该特点可以帮助小模型学习应对复杂任务，进而催生了基于思维链（chain-of-thought，CoT）、上下文学习（in-context learning，ICL）、指令遵循（instruction-following，IF）的黑盒蒸馏方法.大模型的知识蒸馏通常用于将某一领域知识提炼到边缘设备可承载的小模型，用于特定的下游任务[55].小型模型的知识储量和表达能力相较大型模型具有较大差距，使用者需要在模型能力与模型尺寸之间做出进一步权衡.

3）模型量化

模型量化方法将权重或激活值的浮点数表示形式转换为更低精度的数值表示形式，在尽量缩减误差的同时充分利用数值表示空间，主流的量化方案包括训练后量化（post-training quantization，PTQ）和量化感知训练（quantization-aware training，QAT）2 种.PTQ 直接转换训练后的模型权重为低精度格式，无需修改模型架构或重新训练，相比 QAT 具有简单高效的优势[94]，而 QAT 将量化过程融入模型的训练过程，使模型适应低精度的存储格式，做到更低的精度损失.QAT 的重训练方法对一般神经网络的精度恢复的通常具有明显效果，但执行大模型的训练成本非常昂贵，因此 PTQ 成为了大模型量化技术的主流[71]，该部分将会在后文展开论述.

4）低秩分解

低秩分解利用模型权重矩阵的低秩特性，将矩阵近似分解为 2 个或多个更小的矩阵，以节省参数量.该技术已被广泛用于大模型高效参数微调[95]，但是最近的工作表明这种技术也可以用于模型压缩[87-89]，且具有出色的压缩效果.例如 TensorGPT[89]使用低秩张量压缩嵌入层，降低了 LLM 的空间复杂度并使其可在边缘设备上使用.LoSparse[88]通过低秩矩阵和稀疏矩阵的和来近似权重矩阵，结合了低秩近似和结构化剪枝的有点，实现了大量内存的节省.

上述 4 种大模型压缩技术为大模型边缘部署提供了极大的便利，其中模型量化中的 PTQ 量化技术因为成本低、精度损失小、效率高而被广泛采用，已经成为大模型边缘部署和应用的重要优化技术.PTQ 量化技术在大模型上的应用包含仅权重量化和权重激活值量化 2 个主流方向，图 4（c）展示了两者的区别.

1）仅权重量化

为了弥补量化带来的误差，当前的大模型量化方案可分为 3 种，分别为离群值分离[77]、2 阶近似补偿[78]、分布平滑[79].这几种方法并不互斥，例如：SpQR[80]对 GPTQ[78]的量化方案提出了进一步的优化策略，分离离群值并采用稀疏矩阵存储，对非离群值权重采用混合精度的双层量化策略，进一步降低了大模型量化后模型性能损失；AWQ[71]基于 LLM 权重重要性不平衡的观点，按照激活值筛选重要权重，并引入平滑因子以减小重要权重的量化误差，最终实现了适用于多种大模型出色量化方案；OWQ[81]理论分析了激活值的离群值对权重量化误差的放大效应，在 AWQ 基础上引入了权重矩阵的混合精度量化方案.

2）权重激活值量化

权重激活值量化同时量化权重和激活值，仅权重量化的优化技术同样也可以用于激活值.例如：ZeroQuant[76]提出了一种细粒度的硬件友好量化方案，对权重和激活值分别采用不同的量化粒度，并采用逐层知识蒸馏的方法缓解量化后精度损失；SmoothQuant[79]通过平滑激活值分布，将激活值量化

的难度转移到模型权重量化上，在此基础上实现了大模型的 W8A8 量化方案；Outlier Suppression+[74]在 Outlier Suppression[73]的基础上，结合离群值非对称分布且主要集中在特定通道的特征，通过通道级转换和缩放以缓解非对称离群值引起的误差；OliVe[75]采用离群值-受害者对量化，考虑到离群值相比正常值重要性更高，过低硬件开销的方法处理局部离群值；QLLM[96]提出了一种自适应通道重组方法，以有效处理激活值中的离群值，并利用校准数据来抵消量化误差；FPTQ[72]设计了一种新颖的 W4A8 后训练量化方法，将 W8A8 和 W4A16 的优势结合起来，并将细粒度的权重量化与逐层激活量化策略相结合，进一步保持模型的原始性能.

2.3 大模型推理加速

大模型推理加速是一系列不修改模型权重情况下优化模型推理效率的算法和技术，其中一些研究由于效果显著已经被广泛应用在模型部署流程中，如 KV（key-value）缓存、推测解码等.根据优化层级的不同，我们将这些研究工作分类为模型层面的优化与系统层面的优化 2 部分，并在表 4 中展示了与大模型推理加速相关的研究分类及相关工作.

Table 4 Classification of Related Work for Large Model Inference Acceleration Technology

表 4　大模型推理加速技术相关工作分类

优化层次	类别	目的	相关工作
推理算法优化	KV 缓存	利用缓存避免注意力的重复计算，牺牲内存提高推理速度	文献[97-102]
	早期退出	提前终止或跳过不必要的计算，降低平均推理时延	文献[103-110]
	高效提示词	压缩或裁剪提示词，在长上下文场景下减少大模型推理计算量和成本	文献[111-115]
	推测解码	避免自回归算法带来的顺序依赖性，提高模型并行能力和推理速度.	文献[116-122]
系统效率优化	算子优化	充分利用硬件加速能力，减少冗余计算	文献[123-129]
	稀疏性加速	减少冗余计算、冗余内存加载	文献[8,130-133]

2.3.1 推理算法优化

该部分包含 KV 缓存、早期退出、高效提示词、推测解码 4 个方向.图 5 展示了 4 种方法的示意图.

1）KV 缓存

尽可能减少键值对的重复计算能够有效提高大模型推理效率，KV 缓存通过在生成过程中缓存这些张量，从而避免每个生成步骤中重新计算过去的 Token 的键值.然而 KV 缓存随着序列和批次大小而线性增长，使得内存或显存资源面临短缺，为此部分研究[97-100]通过约束缓存数量、丢弃不必要的缓存项，以摆脱缓存长度的不可预测性.KV 缓存与量化结合也是一种节省内存的方法，例如：KVQuant[101]将 KV 缓存视为激活值，并应用量化技术进行低精度压缩，实现了超长上下文长度的 LLM 推理.此外高效的内存管理策略对 KV 缓存的效率同样也有很大影响；PagedAttention[102]受虚拟内存和分页机制启发，提出了一种高效的注意力算法.这种方法对 KV 缓存进行分页内存管理，使得非连续存储变得高效，并减少了内部和外部存储的碎片化.

2）早期退出

早期推理是一种条件计算方法，允许不同样本在不同层中提前结束计算，在推理速度和准确性之间取得良好的平衡.在逐 Token 生成的自回归大模型上，现有研究多从 Token 级别提出早退策略[103]，并研究了多种退出条件[105-110].此外部分研究[103-104]更进一步，不同 Token 可以动态地跳过中间特定层，而不仅仅局限于早期层的提前退出.然而早退改变了模型内部结构，因此需要重新训练或微调，这对于边缘侧设备来说可能是难以接受的.

3）高效提示词

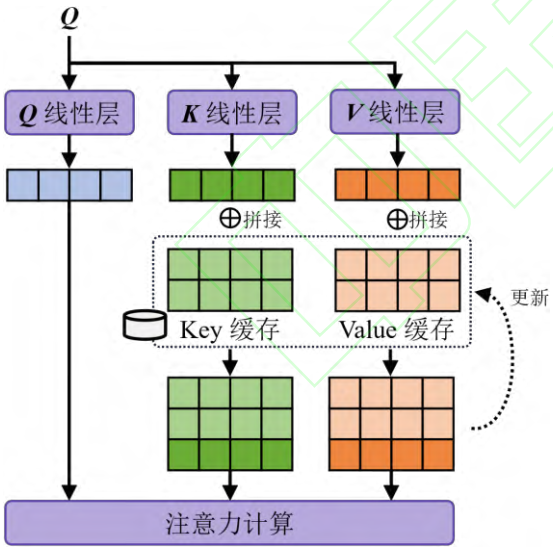
将提示词进行压缩或裁剪可以有效减少大模型推理的计算量和成本，尤其是长上下文场景.近期研究工作中，LLMZip[111]使用 7B 参数量的 Llama 模型作为预测器，并与无损压缩方案结合，取得了较高的文本压缩率.AutoCompressors[112]将预训练的大模型作为压缩器，能够生成长文本的摘要向量，在提高准确性的同时降低推理成本.Selective Context[113]从信息论的角度出发，通过识别和修剪输入上下文中的冗

余内容，使输入更加紧凑，从而提高 LLM 的推理效率.LLMLingua[114]基于小型 LLM 模型，利用压缩与重排实现了在几乎无损的情况下高达 20 倍的压缩率.LongLLMLingua[115]更进一步，提出了基于问题的文档压缩策略，面向长上下文场景下实现提示词高效压缩与推理加速.

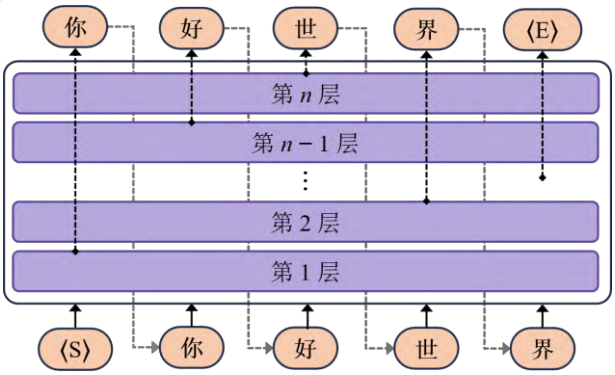
4）推测解码

自回归模型的顺序依赖性[116]使得现有大模型无法充分利用现代加速器能力，导致效率低下，为了摆脱这种依赖性，文献[117]首先提出了推测解码方法，通过使用较小的辅助模型自回归地生成候选序列，较大的主模型通过一次前馈传播判断候选序列中 Token 的正确性并予以纠正.SpecInfer[118]利用多个小型辅助模型，以及一种基于树的推测与 Token 验证机制，大大降低了推理的端到端延迟.Medusa[119] 在 LLM 的最后隐藏状态之上引入了多个头，无需引入辅助模型，能够并行预测多个后续 Token.Lookahead[116]将自回归解码视为求解非线性方程并采用经典雅可比迭代方法进行并行解码，同样也无需辅助模型.EAGLE[120]根据原始模型中间层特征序列预测，使用小型自回归头在特征级别推断下一个特征，通过标记树实现更高的效率.LLMCad[121]将推测解码技术推向边缘侧设备，在物联网设备和智能手机上大幅度提高了 LLM 生成速度.

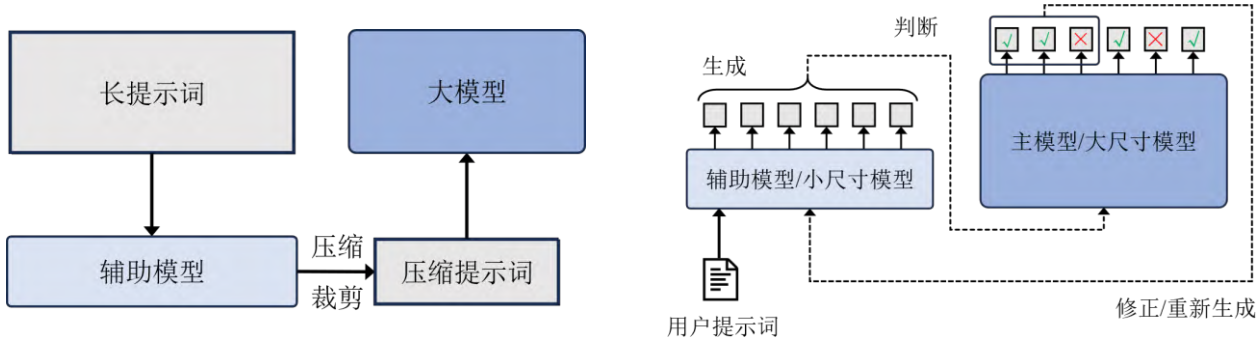
另外，在大模型推理加速领域，从处理大语言模型的策略转向视觉大模型面对的是一个共通的挑战——如何在保证模型性能的同时减少计算资源消耗[134].尽管语言模型和视觉模型在数据处理和模型结构上存在差异，但加速技术的目标一致，即提高实际应用中的推理速度和效率.MuE[135]通过将图像和文本模态在编码器中分解，根据模态灵活跳过不同的层，实现多次早期退出，推动推理效率的同时最小化性能下降.SAMConvex[136]提出一个粗到细的离散优化方法来提高 CT 图像配准的效率，通过计算 SAM 嵌入特征的内积来构建多尺度 6D 成本体积，以此提高模型在特定任务上的执行速度和准确性.MaskCLIP[137]通过优化推理框架，将预训练的 CLIP 模型直接应用于像素级别的预测，而无需专门的注释或复杂的微调过程，实现对未见类别和概念的高效分割.CLIP-Forge[138]采用 2 阶段训练过程，使用未标记的形状数据集和 CLIP 模型，从文本描述中以零样本的方式直接生成 3 维形状，无需在形状-文本配对标签上进行训练，同时采用完全前馈方法，避免了昂贵的推理时间，显著提高了推理阶段的效率.



(a) KV 缓存



(b) 早期退出



(c) 高效提示词　　(d) 推测解码

Fig. 5 Overview of algorithm optimization for large model inference

图 5 大模型推理算法优化概况

2.3.2 系统效率优化

1）算子优化

基于 Transformer[139]架构的大模型核心算子包括注意力算子，算子效率优化通过利用软硬件资源，减少计算量和内存访问，或者利用内核融合等方法减小启动开销等，对于大模型在特定设备上推理具有显著优化效果.对于 GPU 平台的大模型推理，FlashAttention[123]提出了一种利用 GPU 上的高速 SRAM（static random-access memory）的分块注意力算法，使用内核融合避免多次拷贝内存带来的通信开销.FlashAttention-2[124]在原有基础上通过优化 GPU 线程之间的工作分配减少共享内存的读写操作.它进一步通过在线程块和线程束之间分配注意力计算任务，增加了并行度以提高了占用率和效率.FlashDecoding[125]引入了一个沿着键/值序列长度的并行化维度进行规约，即使在小批量大小和长上下文的情况下也能充分利用 GPU.FlashDecoding++[126]引入了一个基于统一最大值的异步 softmax 来消除同步开销以提高了注意力计算效率.它通过双缓冲优化了平面 GEMM（general matrix multiplication）操作，提高了计算利用率并减少了内存延迟.此外 FlashDecoding++实现了一种启发式数据流，能够动态适应硬件资源.对于 CPU 平台的大模型推理，现有研究[122]多设计高度优化的 GEMM 内核，利用低精度运算和 SIMD（single instruction multiple data）指令集的优势加速大模型算子的计算.此外，机器学习编译技术通常将算子融合和优化作为优化目标之一，深度学习编译器[127-129]已被广泛应用在许多大模型部署框架中，对于减少冗余计算，利用边缘硬件环境进行加速具有重要意义.

2）稀疏性加速

近期研究表明，大模型在推理时的激活值具有显著的稀疏性，这为大模型推理效率的优化带来了诸多启发.基于上述观点，文献[130]提出了“上下文稀疏性假设”，使用预测器根据上一层激活值动态预测下一层需要激活的神经元或注意力头，通过舍弃不必要计算达到模型加速效果.除了利用稀疏性减少计算量之外，另一部分研究通过该观点实现高效的内存卸载策略.内存卸载是一种将权重“卸载”到外部存储，在需要时加载部分权重到内存中，使得边缘设备可以运行超过其内存大小的模型.但是频繁的内存交换会导致显著的通信开销，为此高效的内存卸载策略是一个重要的研究方向.FlexGen[131]开发了一种基于线性规划的搜索算法优化吞吐量，以达到最优的卸载策略，并一步将权重和注意力缓存压缩至 4 b，从而显着提高 LLM 推理时最大吞吐量.PowerInfer[132]发现大模型推理表现出高度的局部性，一些被称为“热激活神经元”的神经元被频繁激活.基于这一观察，PowerInfer 设计了神经元感知卸载策略和推理引擎，利用显存和内存存储权重，为显存预加载频繁激活的神经元的权重，而不活跃的神经元的权重则保留在内存中.针对如何在有限内存设备上设计内存卸载策略的问题，LLM in a flash[8]提出了一种基于 DRAM（dynamic random-access memory）和闪存的内存卸载策略，将 LLM 的权重存储在闪存中，而将注意力缓存存储在 DRAM 中，利用滑动窗口缓存过去激活的标记，静态内存预分配以最大限度减少加载延迟，从而在有限的内存设备上实现 LLM 推理.EdgeMoE[133]则开发了是专门为混合专家模型（mixture of experts，MoE）设计的内存卸载策略，利用 MoE 架构的稀疏性，非专家权重存于内存，专家权重仅在激活时从外部存储加载以实现边缘设备的内存节省.

2.4 大模型边缘部署框架

大模型的部署框架通常集合多种模型优化技术，并提供了模型的系统级调度或其他功能.目前主流大模型部署框架[140-143]虽然提供了设备端部署的能力，

但倾向于关注大模型的服务端推理和服务能力.边缘侧的设备通常不具备或具有有限的高性能的神经网络加速芯片，算力和存储相比云端具有显著差异，众多适用于边缘侧的大模型推理引擎和框架因此诞生，并为大模型在边缘侧的开发与应用提供便利.

表 5 展示了现有适用于边缘侧的开源大模型推理框架和引擎，分为通用与专用 2 部分.通用框架指的是通用的边缘侧深度学习推理框架，如 TFLite[146]，TorchExec[147]，MNN[148]，NCNN[149]，这些引擎通常不涉及对大模型架构的专门优化，但是其通用性和灵活性使得它们可以适用于多种模型.另一类推理引擎是专门为大模型推理设计的专用框架，不同于通用的机器学习边缘部署框架[144]，它们通常根据大模型的特点提供专用的加速方案.其中部分框架具有跨架构平台的部署能力，如支持在 Intel，ARM 等芯片架构上运行，而另一些框架则为专门的边缘计算平台设计.此外表格显示模型量化作为一种低成本高成效的优化方案，受到大多数边缘部署框架支持，或提供了量化后模型的推理能力.部分框架，如 MLC-LLM，利用了机器学习编译等技术，进一步减少端侧大模型推理的计算冗余.

Table 5 Summary of Edge Deployment Frameworks for Large Models

表 5　大模型边缘部署框架总结

适用性	框架	特点	量化	多模型支持	跨平台支持
通用	TFLite[146]	在移动设备、嵌入式设备和 loT 设备上运行模型，支持多种开发语言和硬件加速	√	√	√
	TorchExec[147]	PyTorch 平台下边缘部署工具，兼容多种计算平台并具有轻量级运行时	√	√	√
	MNN[148]	轻量级的深度神经网络引擎，对模型格式、算子、设备、操作系统具有广泛的兼容性	√	√	√
	NCNN[149]	适用于移动端的神经网络推理框架，无第三方依赖	√	√	√
专用	MLC-LLM[150]	使用机器学习编译技术加速推理	√	√	√
	llama.cpp[151]	C/C++中 LLM 推理	√	√	√
	llama2.c[152]	纯 C 语言环境执行 Llama 推理			√
	Mllm[153]	适用于移动和边缘设备的多模态推理引擎	√	√	√
	Intel Extension for Transformers[154]	在英特尔平台上提供 LLM 高效推理	√	√	
	InferLLM[155]	轻量级 LLM 推理框架，可部署至移动设备		√	√
	TinyChatEngine[156]	支持多种设备上的多种量化方法	√	√	√
	NanoLLM[157]	为 NVIDIA Jetson 设计的轻量级 LLM 推理引擎		√	

边缘侧的大模型部署框架仍然处于发展阶段，许多框架提供的能力十分有限，适用于边缘侧的大模型的部署框架作为关系大模型能否在边缘侧落地的重要因素，其可用性和多样性仍然有待提高.除上述框架之外，部分工作如 PoweInfer[132]，FlexGen[131]，DeepSparse[145]支持消费级 PC 计算设备上的大语言模型推理，但是这些推理引擎要求设备具备一定的算力水平，其有效性未在边缘侧设备上进行进一步实验.但是由于其对 LLM 推理的提出了多种优化技术与思想，这些工作对推动边缘侧大模型部署仍有较大的参考价值.

3　未来挑战和展望

本文从边缘智能出发，描述了边缘智能下大模型的背景和发展.着重从大模型推理和训练 2 个阶段涉及到的关键技术进行了归纳总结.截至目前，边缘智能下的大模型发展还处于初期阶段，结合边缘智能的特点，目前还存在着以下几个值得关注和讨论的方向：

1）新型大模型架构.目前 Transformer 架构在预训

练大模型中已经占据主导地位，但是其使用的自注意力机制具有平方级别的计算复杂度，使得大模型训练和推理仍然面临成本高、效率低的问题，这一挑战在长文本情景下尤为凸显.为了缓解 Transfromer 架构带来的训练和推理时的资源需求，此前部分研究工作提出了诸多 Transformer 变体，针对注意力机制[158-160]或前馈网络[161-163]进行了大量研究并展现出巨大潜力.另一研究方向则提出了新的模型架构以取代 Transformer，如 Mamba[164]，RWKV[165]，RetNet[166]等.尽管这些架构在性能和效率方面具有不凡的竞争力，但是以这些架构为基础的大模型数量有限，新架构在实践中相比Transformer架构是否具有显著优势，以及模型架构是否存在进一步优化空间仍然有待探索.新型高效架构的探索有望成为大模型领域的重要突破，因此具有较大的研究价值，特别是具有硬件或系统级优化的体系结构，有望让边缘侧大模型的应用成本进一步降低.

2）边缘侧设备资源受限.大模型的训练与推理需要消耗大量计算、内存资源，这种消耗对于边缘设备来说可能是无法承担的.传统的边缘侧深度学习模型部署通常结合轻量化模型结构、模型压缩等技术，但是此类方法仍难以满足大模型在部署到边缘侧时对各种资源的需求.针对此问题，研究更先进的大模型的压缩与加速技术、针对边缘设备的硬件加速器和专门的推理引擎均可以改善大模型在资源受限环境中的微调或推理效率，有望进一步降低大模型在边缘侧部署的成本与压力.同时，现有大模型边缘部署方案通常是将模型完整部署到边缘环境，云边协同作为一种能够平衡利用云端与边端资源的协作模式，如何与现有大模型结合，以实现更高效、更稳定的边缘侧推理方案，同样也是一个前景广阔的研究方向.

3）边缘资源与需求动态性.相比于云计算同质化的计算资源，边缘节点通常具有设备异构性、网络状况多变、存储计算资源差异等特点，使得现有大模型边缘侧微调与推理面临可移植性和效率问题.同时边缘侧场景下用户对模型推理时延、精度等指标的需求也各不相同，使得大模型难以在边缘环境提供稳定一致的服务.通过对边缘侧动态场景进行建模，确立适应动态资源及需求变化的调度机制、协同化策略、自适应算法等方案，是值得进一步关注的问题.

4）大模型联邦微调的异构性问题.复杂动态环境下智能感知中存在“昆虫纲”悖论难题[167-168]，包含感、算、存、传等资源差异性，环境的复杂性与动态性等挑战. 带来感知计算不实时、效果差、难统一等问题. 具体来讲在边缘侧联合众多边缘设备资源来进行大模型联邦微调训练是一种普遍认可的应对大模型定制化需求的可行手段，但在多个边缘设备间对大模型进行联邦微调训练过程中，面对大模型以 GB 为单位的庞大参数，资源受限的边缘设备难以支持其性能需求[83]，同时在客户端间大量的参数传递也给通信带来了巨大压力[169]，导致微调训练效率低下；另一方面，在大模型联邦微调训练中普遍存在的设备异构与数据异构问题更加凸显，严重影响了训练效果和收敛速度[170-171].因此如何在边缘侧资源受限，通信压力大，设备异构、数据异构普遍存在的条件下进行大模型联邦微调训练，还需进行针对性的深入研究.

5）隐私问题.随着模型规模的不断扩大，所需的训练数据量也急剧增加，这导致用户隐私泄露的风险加大.在模型训练过程中，如果不采取适当的保护措施，用户的敏感信息可能会被泄露给攻击者，从而引发严重的隐私安全问题.同时大模型本身也可能成为攻击的目标.攻击者可能会利用模型的漏洞或弱点，对模型进行攻击，从而获取到模型的训练数据、模型结构或推理结果等敏感信息.这种攻击方式不仅会对用户的隐私造成威胁，还会对模型的可用性和可靠性产生严重影响.

6）大模型伦理的规范化.随着大模型逐渐深入边缘侧并应用于各种下游任务，大模型本身的价值观和伦理道德倾向对人类社会的潜在风险愈发显著，同时模型本身的随机性和不可解释性也加剧了大模型伦理问题的不可控性.现有大模型伦理道德规范化手段主要有训练前数据过滤、输出矫正、基于人类反馈的强化学习等，均可以在一定程度上将大模型价值观与人类对齐，但是对齐效果与真实人类社会伦理标准仍然存在巨大差距，未真正实现 AI 与人类普适道德价值的深度对齐[172].大模型的伦理规范问题是全人类社会应当共同应对的挑战，研究更加有效人工智能伦理规范方法和框架，是大模型未来重要的研究方向.

4 总结

边缘智能下的大模型训练和推理具有极大的潜在应用价值，目前的研究还都尚处于初期阶段，许多问题都没有明确的统一和规范，值得我们重点研究.本文首先对边缘智能和大模型的发展以及背景进行了简要回顾，对训练和推理过程中涉及到的关键技术进行了归纳总结，重点从边缘智能角度分析了大模型边缘推理和训练存在的挑战和发展方向.总的来看，在边缘侧进行大模型推理和训练具有极大的应用价值和发展空间，我们未来的研究工作重点将放在动态场景下的大模型推理和训练方面.

作者贡献声明： 王睿提出了论文框架、文献调研路线、指导论文写作并修改论文;张留洋和高志涌负责文献调研、撰写及修改部分论文；姜彤雲补充完善论文.

参 考 文 献

[1] OpenAI. ChatGPT：Optimizing language models for dialogue [EB/OL]. (2022-12-30)[2024-02-10]. https://openai.com/blog/chatgpt/#rf2

[2] Achiam J, Adler S, Agarwal S, et al. GPT-4 technical report[J]. arXiv preprint, arXiv: 2303.08774, 2023

[3] Touvron H, Lavril T, Izacard G, et al. LLaMA: Open and efficient foundation language models[J]. arXiv preprint, arXiv:2302.13971, 2023

[4] Liu Haotian, Li Chunyuan, Wu Qingyang, et al. Visual instruction tuning[J]. arXiv preprint, arXiv:2304.08485, 2023

[5] Kirillov A, Mintun E, Ravi N, etc. Segment anything[J]. arXiv preprint, arXiv: 2304.02643, 2023

[6] Touvron H, Martin L, Stone K,el al. Llama 2: Open foundation and fine-tuned chat models[J]. arXiv preprint, arXiv:2307.09288, 2023

[7] Wang Rui, Qi Jianpeng, Chen Liang, et al. Survey of collaborative inference for edge intelligence[J]. Journal of Computer Research and Development, 2021, 60(2): 398-414 (in Chinese)
（王睿, 齐建鹏, 陈亮, 等. 面向边缘智能的协同推理综述[J]. 计算机研究与发展, 2021, 60(2): 398-414）

[8] Alizadeh K, Mirzadeh I, Belenko D, et al. LLM in a flash: Efficient large language model inference with limited memory[C] //Proc of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). Stroudsburg, PA: ACL, 2024: 12562–12584

[9] Mcmahan H B, Moore E, Ramage D, et al. Communication-efficient learning of deep networks from decentralized data[C] //Proc of the 20th Int Conf on Artificial Intelligence and Statistics PMLR. New York: ACM, 2017: 1273-1282

[10] Custers B, Sears A M, Dechesne F, et al. EU Personal Data Protection in Policy and Practice[M]. The Hague, The Netherlands: TMC Asser Press, 2019

[11] Lambda. OpenAI's GPT-3 language model: A technical overview[EB/OL]. (2020-06-03)[2024-01-08]. https://lambdalabs.com/blog/demystifying-gpt-3#1

[12] Ananthaswamy A. In AI, is bigger always better?[J]. Nature, 2023, 615(7951): 202-205

[13] Brown T, Mann B, Ryder N, et al. Language models are few-shot learners[C] //Proc of the 33rd Int Conf on Neural Information Processing Systems. New York: ACM, 2020: 1877−1901

[14] Lv Kai, Yang Yuqing, Liu Tengxiao, et al. Full parameter fine-tuning for large language models with limited resources[J]. arXiv preprint, arXiv:2306.09782, 2023

[15] Lv Kai, Yan Hang, Guo Qipeng, et al. AdaLomo: Low-memory optimization with adaptive learning rate[J]. arXiv preprint, arXiv:2310.10195, 2023

[16] Malladi S, Gao Tianyu, Nichani E, et al. Fine-tuning language models with just forward passes[J]. arXiv preprint, arXiv:2305.17333, 2023

[17] Ding Ning, Qin Yujia, Yang Guang, et al. Parameter-efficient fine-tuning of large-scale pre-trained language models[J]. Nature Machine Intelligence, 2023, 5(3): 220-235

[18] Chen Chaochao, Feng Xiaohua, Zhou Jun, et al. Federated large language model: A position paper[J]. arXiv preprint, arXiv:2307.08925, 2023

[19] Houlsby N, Giurgiu A, Jastrzebski S, et al. Parameter-efficient transfer learning for NLP[C] //Proc of the 36th Int Conf on Machine Learning PMLR. New York: ACM, 2019: 2790-2799

[20] Hu Zhiqiang, Lan Yihuai, Wang Lei, et al. LLM-adapters: An adapter family for parameter-efficient fine-tuning of large language models[J]. arXiv preprint, arXiv:2304.01933, 2023

[21] Karimi M, Henderson J, Ruder S. Compacter: Efficient low-rank hypercomplex adapter layers[C] //Proc of the 34th Int Conf on Neural Information Processing Systems. New York: ACM, 2021:1022-1035

[22] Li X, Liang P. Prefix-tuning: Optimizing continuous prompts for generation[J]. arXiv preprint, arXiv:2101.00190, 2021

[23] Zhang Renrui, Han Jiaming, Zhou Aojun, et al. Llama-adapter: Efficient fine-tuning of language models with zero-init attention[J]. arXiv preprint, arXiv:2303.16199, 2023

[24] Lester B, Al-Rfou R, Constant N. The power of scale for parameter-efficient prompt tuning[J]. arXiv preprint, arXiv:2104.08691, 2021

[25] Sun Tianxiang, He Zhengfu, Zhu Qin, et al. Multitask pre-training of modular prompt for chinese few-shot learning[C] //Proc of the 61st Annual Meeting of the Association for Computational Linguistics. Stroudsburg, PA: ACL, 2023:11156-11172

[26] Gu Yuxian, Han Xu, Liu Zhiyuan, et al. PPT: Pre-trained prompt tuning for few-shot learning[C] //Proc of the 60th Annual Meeting of the Association for Computational Linguistics. Stroudsburg, PA: ACL, 2022: 8410-8423

[27] Zhang Qingru, Chen Minshuo, Bukharin A, et al. Adaptive budget allocation for parameter-efficient fine-tuning[J]. arXiv preprint, arXiv:2303.10512, 2023

[28] Chen Yukang, Qian Shengju, Tang Haotian, et al. Longlora: Efficient fine-tuning of long-context large language models[J]. arXiv preprint, arXiv:2309.12307, 2023

[29] Chua T J, Yu Wenhan, Zhao Jun, et al. FedPEAT: Convergence of federated learning, parameter-efficient fine tuning, and emulator assisted tuning for artificial intelligence foundation models with mobile edge computing[J]. arXiv preprint, arXiv:2310.17491, 2023

[30] Che Tianshi, Liu Ji, Zhou Yang, et al. Federated learning of large language models with parameter-efficient prompt tuning and adaptive optimization[J]. arXiv preprint, arXiv:2310.15080, 2023

[31] Babakniya S, Elkordy A R, Ezzeldin Y H, et al. SLoRA: Federated parameter efficient fine-tuning of language models[J]. arXiv preprint, arXiv:2308.06522, 2023

[32] Zhang Zhuo, Yang Yuanhang, Dai Yong, et al. FedPETuning: When federated learning meets the parameter-efficient tuning methods of pre-trained language models[C] //Proc of the 61st Annual Meeting of the Association for Computational Linguistics. Stroudsburg, PA: ACL, 2023: 9963-9977

[33] Kuang Weirui, Qian Bingchen, Li Zitao, et al. Federatedscope-llm: A comprehensive package for fine-tuning large language models in federated learning[J]. arXiv preprint, arxiv:2309.00363, 2023

[34] Fan Tao, Kang Yan, Ma Guoqiang, et al. Fate-llm: A industrial grade federated learning framework for large language models[J]. arXiv preprint, arxiv:2310.10049, 2023

[35] Chen Haokun, Zhang Yao, Krompass D, et al. FedDAT: An approach for foundation model finetuning in multi-modal heterogeneous federated Learning[J]. arXiv preprint, arXiv:2308.12305, 2023

[36] Guo Tao, Guo Song, Wang Junxiao, et al. Promptfl: Let federated participants cooperatively learn prompts instead of models-federated learning in age of foundation model[J]. IEEE Transactions on Mobile Computing, 2023, 23(5): 5179-5194

[37] Xu Mengwei, Yin Wangsong, Cai Dongqi, et al. A survey of resource-efficient LLM and multimodal foundation models[J]. arXiv preprint, arXiv:2401.08092, 2024

[38] Wan Zhongwei, Wang Xin, Liu Che, et al. Efficient large language models: A survey[J]. arXiv preprint, arXiv:2312.03863, 2023

[39] Miao Xupeng, Oliaro G, Zhang Zhihao, et al. Towards efficient generative large language model serving: A survey from algorithms to systems[J]. arXiv preprint, arXiv:2312.15234, 2023

[40] Kachris C. A survey on hardware accelerators for large language models[J]. arXiv preprint, arXiv:2401.09890, 2024

[41] Zhong Juan, Liu Zheng, Chen Xi. Transformer-based models and hardware acceleration analysis in autonomous driving: A survey[J]. arXiv preprint, arXiv:2304.10891, 2023

[42] Emani M, Foreman S, Sastry V, et al. A comprehensive performance study of large language models on novel AI accelerators[J]. arXiv preprint, arXiv:2310.04607, 2023

[43] Zhang Xiaodong, Zhang Chaokun, Zhao Jijun. State-of-the-Art survey on edge intelligence [J]. Journal of Computer Research and Development, 2023, 60(12): 2749-2769 (in Chinese)
（张晓东, 张朝昆, 赵继军. 边缘智能研究进展[J]. 计算机研究与发展, 2023, 60(12): 2749-2769）

[44] Zhu Xunyu, Li Jian, Liu Yong, et al. A survey on model compression for large language models[J]. arXiv preprint, arXiv:2308.07633, 2023

[45] Ma Xinyin, Fang Gongfan, Wang Xinchao. LLM-Pruner: On the structural pruning of large language models[J]. arXiv preprint, arXiv:2305.11627, 2023

[46] Xia Mengzhou, Gao Tianyu, Zeng Zhiyuan, et al. Sheared LLaMA: Accelerating language model pre-training via structured pruning[J]. arXiv preprint, arXiv:2310.06694, 2023

[47] Wang Hanrui, Zhang Zhekai, Han Song. SpAtten: Efficient sparse attention architecture with cascade token and head pruning[C] //Proc of the 27th IEEE Int Symp on High-Performance Computer Architecture. Piscataway, NJ: IEEE, 2021: 97-110

[48] Zhang Mingyang, Chen Hao, Shen Chunhua, et al. LoRAPrune: Pruning meets low-rank parameter-efficient fine-tuning[J]. arXiv preprint, arXiv:2305.18403, 2023

[49] Xia Haojun, Zheng Zhen, Li Yuchao, et al. Flash-LLM: Enabling cost-effective and highly-efficient large generative model inference with unstructured sparsity[J]. arXiv preprint, arXiv:2309.10285, 2023

[50] Frantar E, Alistarh D. SparseGPT: Massive language models can be accurately pruned in one-shot[C] //Proc of the 40th Int Conf on Machine Learning PMLR. New York: ACM, 2023: 10323-10337

[51] Sun Mingjie, Liu Zhuang, Bair A, et al. A simple and effective pruning approach for large language models[J]. arXiv preprint, arXiv:2306.11695, 2023

[52] Liang Chen, Zuo Simiao, Zhang Qingru, et al. Less is more: Task-aware

layer-wise distillation for language model compression[C] //Proc of the 40th Int Conf on Machine Learning PMLR. New York: ACM, 2023: 20852-20867
[53] Zhang Chen, Song Dawei, Ye Zheyu, et al. Towards the law of capacity gap in distilling language models[J]. arXiv preprint, arXiv:2311.07052, 2023
[54] Padmanabhan S, Onoe Y, Zhang M, et al. Propagating knowledge updates to LMs through distillation[J]. arXiv preprint, arXiv:2306.09306, 2023
[55] Agarwal R, Vieillard N, Zhou Yongchao, et al. On-policy distillation of language models: Learning from self-generated mistakes[J]. arXiv preprint, arXiv:2306.13649, 2024
[56] Gu Yuxian, Dong Li, Wei Furu, et al. Knowledge distillation of large language models[J]. arXiv preprint, arXiv:2306.08543, 2023
[57] Timiryasov I, Tastet J L. Baby llama: Knowledge distillation from an ensemble of teachers trained on a small dataset with no performance penalty[J]. arXiv preprint, arXiv:2308.02019, 2023
[58] Xiong Yunyang, Varadarajan B, Wu Lemeng, et al. EfficientSAM: Leveraged masked image pretraining for efficient segment anything[J]. arXiv preprint, arXiv:2312.00863, 2023
[59] Yuan Jianlong, Phan M H, Liu Liyang, et al. FAKD: Feature augmented knowledge distillation for semantic segmentation[C] //Proc of the 2024 IEEE/CVF Winter Conf on Applications of Computer Vision. Piscataway, NJ: IEEE, 2024: 595-605
[60] Nasser S A, Gupte N, Sethi A. Reverse knowledge distillation: Training a large model using a small one for retinal image matching on limited data[C] //Proc of the 2024 IEEE/CVF Winter Conf on Applications of Computer Vision. Piscataway, NJ: IEEE, 2024: 7778-7787
[61] Zhu Xuekai, Qi Biqing, Zhang Kaiyan, et al. PaD: Program-aided distillation specializes large models in reasoning[J]. arXiv preprint, arXiv:2305.13888, 2023
[62] Li L H, Hessel J, Yu Youngjae, et al. Symbolic chain-of-thought distillation: Small models can also “think” step-by-step[C] //Proc of the 61st Annual Meeting of the Association for Computational Linguistics. Stroudsburg, PA: ACL, 2023: 2665-2679
[63] Shridhar K, Stolfo A, Sachan M. Distilling reasoning capabilities into smaller language models[C] //Proc of the 61st Annual Meeting of the Association for Computational Linguistics. Stroudsburg, PA: ACL, 2023: 7059-7073
[64] Ho N, Schmid L, Yun S Y. Large language models are reasoning teachers[C] //Proc of the 61st Annual Meeting of the Association for Computational Linguistics. Stroudsburg, PA: ACL, 2023: 14852-14882
[65] Wang Peifeng, Wang Zhengyang, Li Zheng, et al. SCOTT: Self-consistent chain-of-thought distillation[C] //Proc of the 61st Annual Meeting of the Association for Computational Linguistics. Stroudsburg, PA: ACL, 2023: 5546-5558
[66] Hsieh C Y, Li C L, Yeh C K, et al. Distilling step-by-step! Outperforming larger language models with less training data and smaller model sizes[C] //Proc of the 61st Annual Meeting of the Association for Computational Linguistics. Stroudsburg, PA: ACL, 2023: 8003-8017
[67] Chen Zeming, Gao Qiyue, Bosselut A, et al. DISCO: Distilling counterfactuals with large language models[C] //Proc of the 61st Annual Meeting of the Association for Computational Linguistics. Stroudsburg, PA: ACL, 2023: 5514-5528
[68] Jiang Yuxin, Chan C, Chen Mingyang, et al. Lion: Adversarial distillation of proprietary large language models[C] //Proc of the 2023 Conf on Empirical Methods in Natural Language Processing. Stroudsburg, PA: ACL 2023: 3134-3154
[69] Fu Yao, Peng Hao, Ou Litu, et al. Specializing smaller language models towards multi-step reasoning[C] //Proc of the 40th Int Conf on Machine Learning PMLR. New York: ACM, 2023: 10421-10430
[70] Wu Minghao, Waheed A, Zhang Chiyu, et al. LaMini-LM: A diverse herd of distilled models from large-scale instructions[J]. arXiv preprint, arXiv:2304.14402, 2024
[71] Lin Ji, Tang Jiaming, Tang Haotian, et al. AWQ: Activation-aware weight quantization for LLM compression and acceleration[J]. arXiv preprint, arXiv:2306.00978, 2023
[72] Li Qingyuan, Zhang Yifan, Li Liang, et al. FPTQ: Fine-grained post-training quantization for large language models[J]. arXiv preprint, arXiv:2308.15987, 2023
[73] Wei Xiuying, Zhang Yunchen, Zhang Xiangguo, et al. Outlier suppression: Pushing the limit of low-bit transformer language models[C] //Proc of the 36th Int Conf on Neural Information Processing Systems. New York: ACM, 2022:17402-17414
[74] Wei Xiuying, Zhang Yunchen, Li Yuhang, et al. Outlier suppression+: Accurate quantization of large language models by equivalent and effective shifting and scaling[C] //Proc of the 2023 Conf on Empirical Methods in Natural Language Processing. Stroudsburg, PA: ACL, 2023: 1648-1665
[75] Guo Cong, Tang Jiaming, Hu Weiming, et al. OliVe: Accelerating large language models via hardware-friendly outlier-victim pair quantization[C/OL] //Proc of the 50th Annual Int Symp on Computer Architecture. New York:ACM, 2023[2024-9-10].https:/doi.org/10.1145/3579371.3589038
[76] Yao Zhewei, Yazdani A R, Zhang Minjia, et al. ZeroQuant: Efficient and affordable post-training quantization for large-scale transformers [C] //Proc of the 36th Int Conf on Neural Information Processing Systems. New York: ACM, 2022: 27168-27183
[77] Dettmers T, Lewis M, Belkada Y, et al. LLM.int8(): 8-bit matrix multiplication for transformers at scale[C] //Proc of the 36th Int Conf on Neural Information Processing Systems. New York: ACM, 2022: 30318-30332
[78] Frantar E, Ashkboos S, Hoefler T, et al. GPTQ: Accurate quantization for generative pre-trained transformers[C/OL] //Proc of the 11th Int Conf on Learning Representations. OpenReview.net, 2023[2024-09-10]. https://openreview.net/forum?id=tcbBPnfwxS
[79] Xiao Guangxuan, Lin Ji, Seznec M, et al. SmoothQuant: Accurate and efficient post-training quantization for large language models[C] //Proc of the 40th Int Conf on Machine Learning PMLR. New York: ACM, 2023: 38087-38099
[80] Dettmers T, Svirschevski R, Egiazarian V, et al. SpQR: A sparse-quantized representation for near-lossless LLM weight compression[J]. arXiv preprint, arXiv:2306.03078, 2023
[81] Lee Changhun, Jin Jungyu, Kim T, et al. OWQ: Outlier-aware weight quantization for efficient fine-tuning and inference of large language models[C] //Proc of the 38th AAAI Conf on Artificial Intelligence. Palo Alto, CA: AAAI, 2024: 13355-13364
[82] Wang Hongyu, Ma Shuming, Dong Li, et al. BitNet: Scaling 1-bit transformers for large language models[J]. arXiv preprint, arXiv:2310.11453, 2023
[83] Dettmers T, Pagnoni A, Holtzman A, et al. QLoRA: Efficient finetuning of quantized LLMs[C] //Proc of the 37th Advances in Neural Information Processing Systems. Cambridge, MA: MIT Press, 2023: 10088-10115
[84] Kim J, Lee J H, Kim S, et al. Memory-efficient fine-tuning of compressed large language models via sub-4-bit integer quantization[C] //Proc of the 36th Advances in Neural Information Processing Systems. Cambridge, MA: MIT Press,2023: 36187-36207
[85] Liu Zechun, Oguz B, Zhao Changsheng, et al. LLM-QAT: Data-free quantization aware training for large language models[J].arXiv preprint, arXiv:2305.17888, 2023
[86] Liu Xinyu, Wang Tao, Yang Jiaming, et al. MPQ-YOLO: Ultra low mixed-precision quantization of YOLO for edge devices deployment[J]. Neurocomputing, 2024, 574: 127210
[87] Kaushal A, Vaidhya T, Rish I. LORD: Low rank decomposition of monolingual code LLMs for one-shot compression[C/OL] //Proc of the 41st ICML 2024 Workshop on Foundation Models in the Wild. OpenReview.net, 2024[2024-09-10]. https://openreview.net/forum?id=br49PQvuMp
[88] Li Yixiao, Yu Yifan, Zhang Qingru, et al. LoSparse: Structured compression of large language models based on low-rank and sparse approximation[C] //Proc of the 40th Int Conf on Machine Learning. New York: PMLR, 2023: 20336-20350
[89] Xu Mingxue, Xu Yaolei, Mandic D P. TensorGPT: Efficient compression of the embedding layer in LLMs based on the tensor-train decomposition[J]. arXiv preprint, arXiv:2307.00526, 2023
[90] Chang C C, Sung Y Y, Yu Shixing, et al. FLORA: Fine-grained low-rank architecture search for vision transformer[C] //Proc of the 2024 IEEE/CVF Winter Conf on Applications of Computer Vision. Piscataway, NJ: IEEE, 2024: 2482-2491
[91] Benedek N, Wolf L. PRILoRA: Pruned and rank-increasing low-rank adaptation[J]. arXiv preprint, arXiv:2401.11316, 2024
[92] Cheng Hongrong, Zhang Miao, Shi J Q. A survey on deep neural network pruning-taxonomy, comparison, analysis, and recommendations[J]. arXiv preprint, arXiv:2308.06767, 2023
[93] Xu Xiaohan, Li Ming, Tao Chongyang, et al. A survey on knowledge distillation of large language models[J]. arXiv preprint, arXiv:2402.13116, 2024
[94] Zhu Xunyu, Li Jian, Liu Yong, et al. A survey on model compression for large language models[J]. arXiv preprint, arXiv:2308.07633, 2023
[95] Hu E, Shen Yelong, Wallis P, et al. LoRA: Low-rank adaptation of large language models[C/OL] //Proc of the 10th Int Conf on Learning Representations. OpenReview.net, 2022[2024-09-10]. https://openreview.net/forum?id=nZeVKeeFYf9
[96] Liu Jing, Gong Ruihao, Wei Xiuying, et al. QLLM: Accurate and efficient low-bitwidth quantization for large language models[C/OL] //Proc of the 12th Int Conf on Learning Representations. OpenReview.net, 2024[2024-09-10]. https://openreview.net/forum?id=FIplmUWdm3
[97] Xiao Guangxuan, Tian Yuandong, Chen Beidi, et al. Efficient streaming language models with attention sinks[C/OL] //Proc of the 12th Int Conf on Learning Representations. OpenReview.net, 2024[2024-09-10]. https://openreview.net/forum?id=NG7sS51zVF
[98] Liu Zichang, Desai A, Liao Fangshuo, et al. Scissorhands: Exploiting the persistence of importance hypothesis for LLM KV cache compression at test time[C] //Proc of the 37th Advances in Neural Information Processing Systems. Cambridge, MA: MIT Press, 2023: 52342-52364
[99] Zhang Zhenyu, Sheng Ying, Zhou Tianyi, et al. H2O: Heavy-hitter oracle for efficient generative inference of large language models[C] //Proc of the 37th Advances in Neural Information Processing Systems. Cambridge, MA:

MIT Press, 2023: 34661-34710
[100] Ge Suyu, Zhang Yunan, Liu Liyuan, et al. Model tells you what to discard: Adaptive KV cache compression for LLMs[C/OL] //Proc of the 12th Int Conf on Learning Representations. OpenReview.net, 2024[2024-09-10]. https://openreview.net/forum?id=uNrFpDPMyo
[101] Hooper C, Kim S, Mohammadzadeh H, et al. KVQuant: Towards 10 million context length LLM Inference with KV cache quantization[J]. arXiv preprint, arXiv:2401.18079, 2024
[102] Kwon W, Li Zhuohan, Zhuang Siyuan, et al. Efficient memory management for large language model serving with pagedattention[C] //Proc of the 29th Symp on Operating Systems Principles. New York: ACM, 2023: 611-626
[103] Del C L, Del G A, Agarwal S, et al. SkipDecode: Autoregressive skip decoding with batching and caching for efficient LLM inference[J]. arXiv preprint, arXiv:2307.02628, 2023
[104] Zeng Dewen, Du Nan, Wang Tao, et al. Learning to skip for language modeling[J]. arXiv preprint, arXiv:2311.15436, 2023
[105] Schuster T, Fisch A, Gupta J, et al. Confident adaptive language modeling[C] //Proc of the 36th Advances in Neural Information Processing Systems. Cambridge, MA: MIT Press, 2022: 17456-17472
[106] Sun Tianxiang, Liu Xiangyang, Zhu Wei, et al. A simple hash-based early exiting approach for language understanding and generation [J]. arXiv preprint, arXiv:2203.01670, 2022
[107] Liao Kaiyuan, Zhang Yi, Ren Xuancheng, et al. A global past-future early exit method for accelerating inference of pre-trained language models[C] //Proc of the 2021 Conf of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. Stroudsburg, PA: ACL, 2021: 2013-2023
[108] Kong Jun, Wang Jin, Yu L C, et al. Accelerating inference for pretrained language models by unified multi-perspective early exiting[C] //Proc of the 29th Int Conf on Computational Linguistics. Stroudsburg, PA: ACL, 2022: 4677-4686
[109] Zeng Ziqian, Hong Yihuai, Dai Hongliang, et al. ConsistentEE: A consistent and hardness-guided early exiting method for accelerating language models inference[C] //Proc of the 38th AAAI Conf on Artificial Intelligence. Palo Alto, CA: AAAI, 2024: 19506-19514
[110] Bae S, Ko J, Song H, et al. Fast and robust early-exiting framework for autoregressive language models with synchronized parallel decoding[C] //Proc of the 2023 Conf on Empirical Methods in Natural Language Processing. Stroudsburg, PA: ACL, 2023: 5910-5924
[111] Valmeekam C S K, Narayanan K K D, Kalathil D, et al. LLMZip: Lossless text compression using large language models[J]. arXiv preprint, arXiv:2306.04050, 2023
[112] Chevalier A, Wettig A, Ajith A, et al. Adapting language models to compress contexts[C] //Proc of the 2023 Conf on Empirical Methods in Natural Language Processing. Stroudsburg, PA: ACL, 2023: 3829-3846
[113] Li Yucheng, Dong Bo, Guerin F, et al. Compressing context to enhance inference efficiency of large language models[C] //Proc of the 2023 Conf on Empirical Methods in Natural Language Processing. Stroudsburg, PA: ACL, 2023: 6342-6353
[114] Jiang Huiqiang, Wu Qianhui, Lin C Y, et al. LLMLingua: Compressing prompts for accelerated inference of large language models[C] //Proc of the 2023 Conf on Empirical Methods in Natural Language Processing. Stroudsburg, PA: ACL, 2023: 13358-13376
[115] Jiang Huiqiang, Wu Qianhui, Luo Xufang, et al. LongLLMLingua: Accelerating and enhancing LLMs in long context scenarios via prompt compression[C] //Proc of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). Stroudsburg, PA: ACL, 2024: 1658-1677
[116] Fu Yichao, Bailis P, Stoica I, et al. Break the sequential dependency of LLM inference using lookahead decoding[C] //Proc of the 41st Int Conf on Machine Learning. New York: PMLR, 2024: 14060-14079
[117] Leviathan Y, Kalman M, Matias Y. Fast inference from transformers via speculative decoding[C] //Proc of the 40th int Conf on Machine Learning. New York: PMLR, 2023: 19274-19286
[118] Miao Xupeng, Oliaro G, Zhang Zhihao, et al. SpecInfer: Accelerating generative large language model serving with tree-based speculative inference and verification[C] //Proc of the 29th ACM Int Conf on Architectural Support for Programming Languages and Operating Systems, Volume 3. New York: ACM, 2024: 932–949
[119] Cai T, Li Yuhong, Geng Zhengyang, et al. Medusa: Simple LLM inference acceleration framework with multiple decoding heads[C] //Proc of the 41st int Conf on Machine Learning. New York: PMLR, 2024: 5209-5235
[120] Li Yuhui, Wei Fangyun, Zhang Chao, et al. EAGLE: Speculative sampling requires rethinking feature uncertainty[C] //Proc of the 41st int Conf on Machine Learning. New York: PMLR, 2024: 28935-28948
[121] Xu Daliang, Yin Wangsong, Jin Xin, et al. LLMCad: Fast and scalable on-device large language model inference[J]. arXiv preprint, arXiv:2309.04255, 2023
[122] Shen Haihao, Chang Hanwen, Dong Bo, et al. Efficient llm inference on cpus[J]. arXiv preprint, arXiv:2311.00502, 2023
[123] Dao T, Fu Dan, Ermon S, et al. FlashAttention: Fast and memory-efficient exact attention with IO-awareness[C] //Proc of the 36th Advances in Neural Information Processing Systems. Cambridge, MA: MIT Press, 2022: 16344-16359
[124] Dao T. FlashAttention-2: Faster attention with better parallelism and work partitioning[C/OL] //Proc of the 12th Int Conf on Learning Representations. OpenReview.net, 2024[2024-09-10]. https://openreview.net/forum?id=mZn2Xyh9Ec
[125] Dao T, Haziza D, Massa F, et al. Flash-Decoding for long-context inference[EB/OL]. 2023[2024-02-03]. https://pytorch.org/blog/flash-decoding/
[126] Hong Ke, Dai Guohao, Xu Jiaming, et al. FlashDecoding++: Faster large language model inference with asynchronization, flat GEMM optimization, and heuristics[C/OL] //Proc of Machine Learning and Systems. 2024: 148-161 [2024-09-12]. https://proceedings.mlsys.org/paper_files/paper/2024/hash/5321b1dabcd2be188d796c21b733e8c7-Abstract-Conference.html
[127] Lai Ruihang, Shao Junru, Feng Siyuan, et al. Relax: Composable abstractions for end-to-end dynamic machine learning[J]. arXiv preprint, arXiv:2311.02103, 2023
[128] Tillet P, Kung H T, Cox D. Triton: An intermediate language and compiler for tiled neural network computations[C] //Proc of the 3rd ACM SIGPLAN Int Workshop on Machine Learning and Programming Languages. New York: ACM, 2019: 10-19
[129] Feng Siyuan, Hou Bohan, Jin Hongyi, et al. TensorIR: An abstraction for automatic tensorized program optimization[C] //Proc of the 28th ACM Int Conf on Architectural Support for Programming Languages and Operating Systems: Volume 2. New York: ACM, 2023: 804-817
[130] Liu Zichang, Wang Jue, Dao T, et al. Deja Vu: Contextual sparsity for efficient LLMs at inference time[C] //Proc of the 40th Int Conf on Machine Learning. New York: PMLR, 2023: 22137-22176
[131] Sheng Ying, Zheng Lianmin, Yuan Binhang, et al. FlexGen: High-throughput generative inference of large language models with a single GPU[C] //Proc of the 40th Int Conf on Machine Learning. New York: PMLR, 2023: 31094-31116
[132] Song Yixin, Mi Zeyu, Xie Haotong, et al. PowerInfer: Fast large language model serving with a consumer-grade GPU[J]. arXiv preprint, arXiv:2312.12456, 2023
[133] Yi Rongjie, Guo Liwei, Wei Shiyun, et al. EdgeMoE: Fast on-device inference of MoE-based large language models[J]. arXiv preprint, arXiv:2308.14352, 2023
[134] Awais M, Naseer M, Khan S, et al. Foundational models defining a new era in vision: A survey and outlook[J]. arXiv preprint, arXiv:2307.13721, 2023
[135] Tang Shengkun, Wang Yaqing, Kong Zhenglun, et al. You need multiple exiting: Dynamic early exiting for accelerating unified vision language model[C] //Proc of the 44th IEEE/CVF Conf on Computer Vision and Pattern Recognition. Piscataway, NJ: IEEE, 2023: 10781-10791
[136] Li Zi, Tian Lin, Mok C W, et al. Samconvex: Fast discrete optimization for ct registration using self-supervised anatomical embedding and correlation pyramid[G] //Proc of the 26th Medical Image Computing and Computer Assisted Intervention(MICCAI 2023). Berlin: Springer, 2023: 559-569
[137] Zhou Chong, Loy C C, Dai Bo. Extract free dense labels from CLIP[C] //Proc of the 17th Computer Vision(ECCV 2022). Berlin: Springer, 2022: 696-712
[138] Sanghi A, Chu Hang, Lambourn J G, et al. Clip-forge: Towards zero-shot text-to-shape generation[C] //Proc of the 2022 IEEE/CVF Conf on Computer Vision and Pattern Recognition. Piscataway, NJ: IEEE, 2022: 18603-18613
[139] Vaswani A, Shazeer N, Parmar N, et al. Attention is All you Need[C] //Proc of the 31st Advances in Neural Information Processing Systems. Cambridge, MA: MIT Press, 2017: 5999-6009
[140] InternLM. LMDeploy [EB/OL]. 2023[2024-02-04]. https://github.com/InternLM/lmdeploy
[141] Microsoft. DeepSpeed-MII[EB/OL]. 2022[2024-02-04]. https://github.com/microsoft/DeepSpeed-MII
[142] NVIDIA. TensorRT-LLM[EB/OL]. 2023[2024-02-04]. https://github.com/NVIDIA/TensorRT-LLM
[143] vLLM Team. vLLM[EB/OL]. 2023[2024-02-04]. https://github.com/vllm-project/vllm
[144] Lin Ji, Chen Weiming, Lin Yujun, et al. MCUNet: Tiny deep learning on IoT devices[C] //Proc of the 34th Advances in Neural Information Processing Systems. Cambridge, MA: MIT Press, 2020: 11711-11722
[145] Neuralmagic. DeepSparse[EB/OL]. 2021[2024-02-04]. https://github.com/neuralmagic/deepsparse
[146] Li Shuangfeng. TensorFlow lite: On-device machine learning framework[J]. Journal of Computer Research and Development, 2020, 57(9): 1839-1853 (in Chinese)
（李双峰. TensorFlow Lite：端侧机器学习框架[J]. 计算机研究与发展, 2020, 57(9): 1839-1853）
[147] PyTorch Team. PyTorch ExecuTorch[EB/OL]. 2023[2024-05-28]. https://pytorch.org/executorch
[148] Alibaba. MNN[EB/OL]. 2019[2024-06-30]. https://github.com/alibaba/MNN

[149] Tencent. ncnn[EB/OL]. 2017[2024-05-30]. https://github.com/Tencent/ncnn
[150] MLC Team. MLC LLM[EB/OL]. 2023[2024-02-04]. https://github.com/mlc-ai/mlc-llm
[151] Gerganov G. llama.cpp[EB/OL]. 2023[2024-02-04]. https://github.com/ggerganov/llama.cpp
[152] Karpathy A. llama2.c[EB/OL]. 2023[2024-02-04]. https://github.com/karpathy/llama2.c
[153] Mllm Team. mllm[EB/OL]. 2023[2024-02-04]. https://github.com/UbiquitousLearning/mllm
[154] Intel. Intel Extension for Transformers[EB/OL]. 2022[2024-02-04]. https://github.com/intel/intel-extension-for-transformers
[155] Megvii Inc. InferLLM[EB/OL]. 2023[2024-02-04]. https://github.com/MegEngine/InferLLM
[156] MIT Han Lab. TinyChatEngine[EB/OL]. 2023[2024-02-04]. https://github.com/mit-han-lab/TinyChatEngine
[157] NVIDIA. NanoLLM[EB/OL]. 2024[2024-04-28]. https://github.com/dusty-nv/NanoLLM
[158] Shazeer N. Fast transformer decoding: One write-head is all you need[J]. arXiv preprint, arXiv:1911.02150, 2019
[159] Ainslie J, Lee-Thorp J, de Jong M, et al. GQA: Training generalized multi-query transformer models from multi-head checkpoints[C] //Proc of the 2023 Conf on Empirical Methods in Natural Language Processing. Stroudsburg, PA: ACL, 2023: 4895-4901
[160] Choromanski K M, Likhosherstov V, Dohan D, et al. Rethinking attention with performers[C/OL] //Proc of the 9th Int Conf on Learning Representations. OpenReview.net, 2021[2024-09-10]. https://openreview.net/forum?id=Ua6zuk0WRH
[161] Shazeer N. Glu variants improve transformer[J]. arXiv preprint, arXiv:2002.05202, 2020
[162] Lepikhin D, Lee H, Xu Yuanzhong, et al. GShard: Scaling giant models with conditional computation and automatic sharding[C/OL] //Proc of the 9th Int Conf on Learning Representations. OpenReview.net, 2021[2024-09-10]. https://openreview.net/forum?id=qrwe7XHTmYb
[163] Fedus W, Zoph B, Shazeer N. Switch transformers: Scaling to trillion parameter models with simple and efficient sparsity[J]. Journal of Machine Learning Research, 2022, 23(1): 120:5232-120:5270
[164] Gu A, Dao T. Mamba: Linear-time sequence modeling with selective state spaces[J]. arXiv preprint, arXiv:2312.00752, 2023
[165] Peng Bo, Alcaide E, Anthony Q, et al. RWKV: Reinventing RNNs for the transformer era[C] //Proc of the Findings of the Association for Computational Linguistics(EMNLP 2023). Stroudsburg, PA: ACL, 2023: 14048-14077
[166] Sun Yutao, Dong Li, Huang Shaohan, et al. Retentive network: A successor to transformer for large language models[J]. arXiv preprint, arXiv:2307.08621, 2023
[167] Xu Zhiwei, Zeng Chen, Zhao Lu, et al. Domain oriented architecture: An architectural style of intelligent interconnection of all things[J]. Journal of Computer Research and Development, 2019, 56(1): 90-102 (in Chinese)
（徐志伟，曾琛，朝鲁，等. 面向控域的体系结构：一种智能万物互联的体系结构风格 [J]. 计算机研究与发展, 2019, 56(1): 90-102）
[168] Li Guojie. Further understanding of big data[J]. Big Data, 2015, 1(1): 8-16 (in Chinese)
（李国杰. 对大数据的再认识 [J]. 大数据, 2015, 1(1): 8-16）
[169] Woisetschläger H, Isenko A, Wang Shiqiang, et al. Federated fine-tuning of llms on the very edge: The good, the bad, the ugly[C] //Proc of the 8th Workshop on Data Management for End-to-End Machine Learning. New York: ACM, 2024: 39-50
[170] Yang Chengxu, Xu Mengwei, Wang Qipeng, et al. Flash: Heterogeneity-aware federated learning at scale[J]. IEEE Transactions on Mobile Computing, 2024, 23(1): 483-500
[171] Lu Wang, Hu Xixu, Wang Jindong, et al. FedCLIP: Fast generalization and personalization for CLIP in federated learning[J]. IEEE Data Engineering Bulletin, 2023, 46(1): 52-66
[172] Yi Xiaoyuan, Xie Xing. An analysis of the alignment of moral values in the large model[J]. Journal of Computer Research and Development, 2023, 60(9): 1926-1945 (in Chinese)
（矣晓沅，谢幸. 大模型道德价值观对齐问题剖析[J]. 计算机研究与发展, 2023, 60(9): 1926-1945）

Wang Rui, born in 1975. PhD, professor. Senior member of CCF. His main research interests include IoT, edge intelligence, and smart healthcare.
王睿，1975 年生.博士，教授，CCF 高级会员.主要研究方向为物联网、边缘智能和智慧医疗.

Zhang Liuyang, born in 2000. Master candidate. His main research interests include federated learning and large model training.
张留洋，2000 年生.硕士研究生.主要研究方向为联邦学习与大模型训练.

Gao Zhiyong, born in 2000. Master candidate. His main research interests include large model inference and machine learning.
高志涌，2000 年生.硕士研究生.主要研究方向为大模型推理与机器学习.

Jiang Tongyun, born in 2000. Master candidate. Her main research interests include edge intelligence and large model.

姜彤雲，2000 年生.硕士研究生.主要研究方向为边缘智能与大模型.