大模型端侧推理综述

1. 摘要

在边缘设备上运行大型语言模型（LLMs）引起了相当大的兴趣，因为它们在**隐私保护、降低延迟和节省带宽**方面具有优势。尽管如此，与功能强大的云计算中心相比，设备本身的有限容量内在地限制了在设备上LLMs的能力。为了弥合基于云的和在设备上的AI之间的差距，移动边缘智能（MEI）提供了一种可行的解决方案，通过在移动网络边缘提供AI能力，与云计算相比具有更好的隐私和延迟表现。MEI**介于在设备上的AI和基于云的AI之间**，具有无线通信能力，并且比终端设备拥有更强大的计算资源。本文提供了一项关于利用MEI进行LLMs的现代综述。我们首先涵盖了LLMs的基础知识，从LLMs和MEI开始，随后是资源高效的LLM技术。然后，我们展示了几个关键应用，以证明在网络边缘部署LLMs的需求，并提出了MEI用于LLMs（MEI4LLM）的架构概览。接着，我们深入探讨了MEI4LLM的各个方面，广泛涵盖了边缘LLM缓存和交付、边缘LLM训练和边缘LLM推理。最后，我们确定了未来的研究方向。我们旨在激发该领域的研究人员利用移动边缘计算，以促进LLMs的部署，使其更接近用户，从而在各种对隐私和延迟敏感的应用中释放LLMs的潜力。

2. 引言

2.1 背景

大语言模型（LLMs）的出现是人工智能（AI）技术的一个里程碑，它使得通用智能成为可能。LLMs不仅在它们构建的任务上表现出色，例如生成文本响应，而且在多模态内容分析、摘要和泛化等任务上也表现出色。例如，GPT-4多模态模型接受图像和文本输入，产生文本输出，在各种专业和学术基准测试中展现出人类级别的性能。除了这些通常被称为基础模型的通用模型外，LLMs还可以被微调以适应特定的行业和应用场景。例如，Google设计的医疗LLM，Med-PaLM M [1]，旨在提供基于丰富的数据模态（包括文本、影像、基因组学等）的高质量答案。**Google DeepMind还开发了机器人2（RT-2）**，这是一个用于控制机器人的视觉-语言-动作AI模型。广泛的用例突显了LLMs对日常生活的深远影响。

由于相关的计算、存储和内存成本，现有的LLMs主要限于云数据中心提供服务。遗憾的是，**基于云的LLM服务带来了固有的缺点，包括数据隐私泄露、高带宽成本和长服务延迟。**用户必须上传他们的数据才能利用云中心的资源来访问LLM服务，这通常会导致显著的通信延迟。此外，上传私人数据对用户隐私构成了严重风险，尤其是在像智能健康这样的隐私敏感应用中。鉴于这些担忧，人们对在设备上部署LLM的兴趣日益增加，这在主要行业参与者之间引发了一场竞争。**例如，Google已经在Pixel 8 Pro智能手机上推出了Gemini Nano，分别拥有18亿和32.5亿参数[3]。**高通计划在Snapdragon驱动的旗舰智能手机和个人电脑上推出Llama 2支持。在设备上部署LLM可以实现对敏感个人数据（如端到端（E2E）加密消息和健康数据）的本地处理。它还为机器人规划和自动驾驶等对延迟敏感的应用提供了低响应时间。这些显著的优势推动了LLMs从云中心向移动设备的持续转移。

2.2 移动边缘智能(MEI)

尽管在设备上的LLM正在成为一个快速增长的领域，但它们在广泛部署方面面临着严重的限制。具体来说，边缘设备上的计算、内存和存储资源的稀缺性大大限制了在设备上LLM的规模。一方面，现有的工业努力集中在小于100亿参数（10 billion parameters）的LLMs上，因为它们对在设备上部署有巨大的资源需求。例如，**Google的Gemini Nano依赖于4位模型，分别有18亿和32.5亿参数，只能支持相对“基本”的功能，如文本摘要、智能回复建议和语法检查。**然而，随着所需功能的复杂性增加，有必要在设备上部署更大规模的LLM，这可能会显著增加LLM在设备上推理的开销。另一方面，在设备上微调为个性化和感知上下文的AI铺平了道路，这是实现卓越AI性能的基础。然而，**现有的在设备上LLM产品没有纳入在设备上训练（微调）功能，因为训练成本通常比AI推理要高得多。**为了解决前述的困境，移动边缘计算提供了一个有希望的解决方案。6G移动网络旨在通过利用边缘计算系统，为广泛的移动设备提供低延迟的AI推理和训练服务，这些系统位于基站等网络端的计算能力上。这引出了一个名为“移动边缘智能（MEI）”的范式。**MEI介于在设备上的AI和基于云的AI之间，具有无线通信和适度规模的计算资源。换句话说，它比边缘设备更强大，但不如云中心那么强大。**由于边缘设备和边缘服务器之间的距离

很短，可以支持大规模的LLMs，同时服务延迟较低。同时，6G边缘可以利用边缘服务器上更强大的内存、能量和计算能力，在不断变化的环境中不断微调LLMs。因此，6G移动边缘预计将在将LLMs推向边缘设备方面发挥重要作用。

2.3 与其他综述的比较

TABLE I: Summary of the related surveys/articles.

Ref.	Description	Scenarios				Perspectives		
		Efficient (on-device) LLM techniques	Edge LLM caching and delivery	Edge LLM training	Edge LLM inference	Storage eff.	Comp. eff.	Comm. eff.
[5]	Reviews recent progress of LLM pre-training, fine-tuning, usage, and capacity evaluation, along with the public resources for deploying LLM.	✔	✗	✗	✗	✗	✔	✗
[6]	Introduces resource-efficient approaches to deploying LLMs, including model architectures, training and inference algorithms, and practical system designs.	✔	✗	✔	✔	✔	✔	✗
[7]	Explores efficient LLM deployment via model-centric methods (model compression, training, inference, and architecture design), data-centric approaches (data selection and prompt engineering), and framework-centric strategies (specific training, inference, and serving frameworks).	✔	✗	✗	✗	✗	✔	✗
[8]	Overviews the deployment of AI-generated content applications in mobile networks, including mobile devices, edge servers, and cloud centers.	✔	✗	✔	✔	✗	✔	✔
[9]	Surveys the current hardware acceleration methods for energy-efficient on-device LLM training and inference.	✔	✗	✗	✗	✗	✔	✗
[10]	Reviews resource-efficient techniques for LLM deployment, covering computational, memory, energy, economic, and network resources based on their applicability across architecture design, pre-training, fine-tuning, inference, and system design.	✔	✗	✗	✔	✔	✔	✔
[11]	Summarizes current research on efficient LLM inference, including compression, fast decoding, and optimization for compiler/system/hardware.	✔	✗	✗	✗	✔	✔	✗
[12]	Reviews the evolution of low-cost on-device LLM training and inference techniques.	✔	✗	✗	✗	✔	✔	✗
[13]	Overviews PEFT algorithms for LLMs, reviews computing-efficient applications and techniques, and introduces system design for PEFT.	✔	✗	✗	✗	✔	✔	✗
Ours	Reviews the state-of-the-art approaches to edge LLM training, inference, caching, and delivery in MEI, with an emphasis on enhancing storage, computing, and communication efficiency of LLM deployment at the network edge.	✔	✔	✔	✔	✔	✔	✔

3. 预备知识——LLMs 和 MEI

3.1 Transformers（略）

3.2 LLMs

LLM架构可以分为三类：仅编码器LLMs、编码器-解码器LLMs和仅解码器LLMs。仅编码器LLMs，如ALBERT，仅由编码器组件组成，通常基于Transformer等高级架构。编码器负责处理输入序列，为每个标记生成上下文化表示。尽管缺少解码器来产生输出序列，但由于其有效的特征提取能力和适应性表示，**仅编码器LLMs在文本分类、句子相似性计算和语言理解等NLP任务上仍然表现出色。**编码器-解码器LLMs，如T5模型，代表了NLP领域的一个关键进步，将编码器和解码器组件整合到其架构中。**编码器**

处理输入序列以生成上下文化表示，而解码器则利用这些表示生成输出序列，通常以序列到序列的方式进行。编码器-解码器LLMs在机器翻译、文本摘要和问答等任务上得到广泛应用，因为它们能够捕获复杂的语言结构和上下文依赖关系。仅解码器LLMs，如著名的GPT系列，是LLMs的一个重要分支。仅解码器LLMs采用自回归解码，这在仅解码器和编码器-解码器LLMs中都被广泛使用，**基于序列中的先前标记生成输出序列。这种架构设计使它们特别适合于模型按顺序生成文本的任务，如语言生成、文本补全和对话响应生成。**

3.3 多模态LLMs

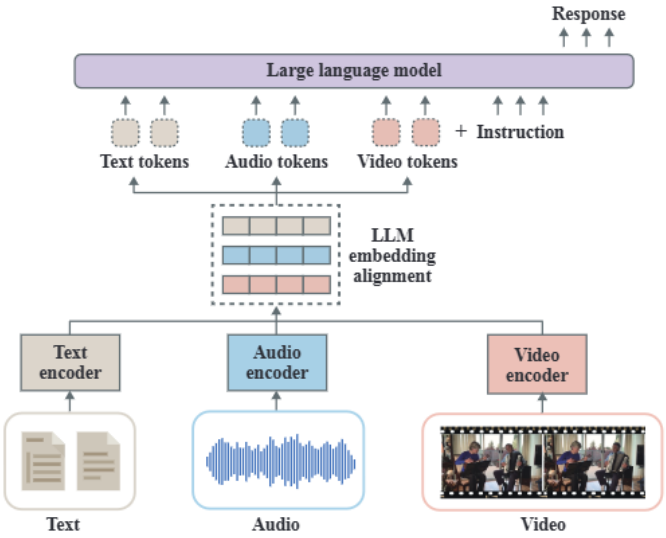


Fig. 2: The structure of multimodal LLM.

多模态感知在追求通用AI的过程中起着关键作用，驱动着处理复杂现实世界数据的必然需求。这需要AI模型能够进行**跨模态信息融合和交互式学习**，提高多个感知领域的训练性能。多模态LLMs继承了LLMs的强大学习能力，通过整合各种模态的基础模型，增强了处理多样化和复杂的多模态任务的能力。LLMs提供了强大的语言生成能力、零样本转移能力和上下文学习能力，而**其他模态的基础模型提供了其他数据类型的有信息的表示。由于不同模态的基础模型各自进行了预训练，构建多模态LLMs的主要挑战在于如何连接这些模型以实现高性能的协同训练/推理。该领域的主要研究集中在通过多模态预训练和多模态指令调整来完善 *模态对齐* 。多模态预训练通过使用多模态数据集训练模型，学习**跨模态的共同表示，**例如XText。在训练过程中，模型通过优化预定义的目标，学习**将来自不同模态的信息相关联，**从而实现跨模态的对齐。这种对齐增强了模型对跨模态关系的理解，导致在各种跨模态任务中的性能提高。多模态指令调整是一种基于预训练模型的微调方法，旨在提高模型在特定任务上的性能。它**将模型与一个或多个与模态相关的任务结合起来，然后使用模态标记的数据对模型进行微调，以提高其与模态特定任务的对齐。**多模态指令调整使模型能够通过**遵循新指令学习增强未见任务的能力，从而提高模型的零样本性能和泛化能力。**【关键词： 跨模态信息融合、 多模态预训练+指令微调——> 模态对齐(共同表示)】

3.4 生成/交互式AI

生成AI（GAI）和交互式AI（IAI），GAI专注于创建包括图像、文本、音乐和视频在内的广泛内容，统称为AI生成内容（AIGC）。另一方面，IAI可以被视为GAI的下一个阶段。IAI在聊天机器人和虚拟助手等应用中响应用户查询，同时使AI代理能够通过用户交互进行适应，从而不断提高准确性。通过利用功能强大的LLMs和GAI的内容生成优势，IAI使AI代理能够模仿人类交互并生成有意义和动态的对话。为了使AI代理能够生成更准确和最新的响应，可以在LLMs中集成**检索增强生成（RAG）**，以增强IAI和GAI。具体来说，LLMs在生成响应时使用输入序列从外部知识源检索相关数据，从而提高内容生成性能。**例如，Google将RAG与Gemini结合使用，增强LLMs生成更准确和上下文相关的响应的能力。**

3.5 工业界LLMs的进展

在医疗保健领域，**Med-PaLM**被设计用于医学图像分析、临床文档处理和患者诊断，帮助医疗专业人员做出准确的诊断和治疗决策。在自动驾驶领域，**DriveMLM**弥合了语言决策和车辆控制命令之间的差距，实现了在现实模拟器中的闭环自动驾驶。最近在设备上的LLMs的进展已经引起了行业的关注。例如，Meta提出了一个名为**MobileLLM**的设备上LLM，利用深度和薄架构、嵌入共享和分组查询注意力机制。尽管如此，**在设备上的LLMs通常与更大模型尺寸的强大LLMs相比表现不佳。**例如，Google为设备上部署而设计的Gemini Nano-1，以4位格式包含仅18亿参数，这些参数从更大的Gemini模型中提取[69]。由于其紧凑的尺寸，当这样的小型LLM的能力不足以满足边缘设备的要求时，这些设备可能仍然需要上传数据以访问大规模的LLMs，即在边缘服务器上。

3.6 移动边缘智能

MEI 作为一个融合了人工智能与移动边缘计算的有前景的范式，彻底改变了移动服务和应用的格局。MEI 的发展源于多种技术进步的融合，包括物联网（IoT）设备的普及、移动网络的部署以及人工智能算法的成熟。这些发展使 MEI 能够克服传统以云为中心的架构的限制，通过在网络边缘提供本地化的人工智能训练/推理和数据处理能力。通过整合人工智能和通信，MEI 框架使移动网络能够提供超越通信的服务，为万物智能奠定了坚实的基础。沿着这条线，"集成人工智能和通信"的用例已经被包括在6G的IMT框架建议中。**在标准化方面**，电信标准化组织3GPP和ITU已经在它们的白皮书中描绘了边缘智能的前景。ITU-3172阐明了基于机器学习应用的延迟敏感要求，在网络边缘托管机器学习功能的必要性。在3GPP针对5G标准化的第18版中，MEI旨在支持分布式学习算法、分割的AI/ML和高效的AI模型分发。

首先，边缘学习，如联邦学习，将在边缘网络中得到全面支持，这使边缘服务器能够聚合来自多个分布式边缘设备模型更新和知识，从而提高AI/ML模型的性能。其次，5G边缘网络中的分割AI/ML可以促进在具有冲突要求的设备上部署AI应用，例如计算密集型、能耗密集型、隐私敏感型和延迟敏感型要求。例如，**在边缘分割推理中，AI模型被分割为子模型，计算密集型和能耗密集型的子模型被卸载到5G边缘服务器（例如基站）。边缘服务器可以执行带有边缘侧子模型的推理，并上传来自边缘设备的中间数据。最后，高效的AI模型下载确保当边缘设备需要适应新的AI任务和环境时，AI模型可以以低延迟被传送到边缘设备。**例如，**自动驾驶车辆在驾驶环境变化时需要在1秒内从5G边缘服务器下载新的AI模型。**为了将基于网络的AI算法整合到5G网络中，MEI框架需要满足边缘服务器和边缘设备之间高速稳定数据链路的请求。这些链路可以为持续上传中间数据/模型更新到边缘服务器提供高且恒定的上行数据速率，并在突发情况下为下载AI模型到边缘设备提供高下行数据速率。**此外，MEI的核心是利用数据源与边缘计算设备的接近性，以实现更接近数据源的智能决策。**

对于边缘AI赋能的IoT应用，微软的"Azure IoT Edge"、谷歌的"Cloud IoT"、亚马逊的"Web Services IoT"和英伟达的"EGX"提供了边缘AI平台，以在从实时视频分析、智能家居到工业IoT的广泛应用中提供实时AI服务。

3.7 MEI4LLM的教训

MEI4LLM是MEI的一个特例。具体来说，MEI4LLM必须具备以下特点：

1. 原生支持模型分割和跨互连边缘节点的并行训练/推理，以便于部署极大规模模型；
2. 无线网络和资源高效LLM训练/推理技术的集成设计，如参数高效微调和令牌（表示）减少，使LLM部署具有成本效益。

本质上，与传统MEI相比，MEI4LLM主要侧重于探索**资源管理和高效AI技术的集成设计**，以支持有限通信计算资源下的LLMs，这将是本综述论文和该领域研究主题的重点。

4. 预备知识——资源高效LLM技术

在边缘设备/服务器上部署LLMs进行训练/推理几个关键挑战：

- 过度的计算开销：据报道，GPT-4生成一个标记的前向传播大约需要560万亿次浮点运算。然而，先进的A100 GPU仅提供每秒195万亿次浮点运算的计算能力。这表明，使用单个A100 GPU生成GPT-4的一个标记大约需要28秒。此外，反向传播通常需要比前向传播更多的计算资源，这意味着在设备上进行训练将更具挑战性。
- 巨大的存储/内存需求：一方面，在边缘设备上缓存LLMs会消耗大量的存储资源。即使是为在设备上部署而设计的LLMs也有数十亿参数，例如，Google的在设备上的Gemini Nano-2有32.5亿参数。另一方面，**在训练期间通常使用的Adam优化器通常需要比推理多12倍的内存资源**，这对于内存有限的移动设备来说是不可接受的。
- 高能耗成本：边缘设备的电池容量有限，限制了在边缘设备上部署LLMs。例如，在小米11智能手机上使用llama.cpp（最轻量级的在设备上LLM引擎之一）运行一个量化为INT4、有130亿参数的LLM，每个标记的能耗大约是56焦耳[88]。这意味着，**如果LLM部署在智能手机上，一个电池容量为3000毫安时、输出电压为3.7伏的智能手机只能生成大约700个标记。**

4.1 高效推理

LLM压缩

- **量化**
  与传统量化策略针对权重和激活都进行量化不同，**LLM量化主要关注权重量化。**原因如下：首先，对激活进行量化会导致LLMs的性能下降更为显著。**其次，使用LLMs生成新标记时，延迟和能源消耗的主要来源通常是由于从内存中加载模型参数。**因此，权重量化允许更有效地从内存中加载量化权重，使推理更加高效，而不会显著降低推理准确性。
- **剪枝**
- **知识蒸馏**

LLMs的快速解码

- **推测性解码**
- **早期退出**
- **混合专家(MoE)**
  混合专家(MoE)可以有效地扩展LLM容量并提高各种下游任务的性能。具体来说，可以将原始的FFN替换为专家网络，该网络由多个FFN专家和一个路由器组成。**在推理期间，路由器将给定的输入标记定向到最适合的FFN专家或专家组。在推理期间只激活和加载LLM的一部分。占用较少存储但需要更多计算资源的非专家权重始终保留在内存中。**相比之下，占用较少计算资源的大型专家权重保存在磁盘上。只有在特定任务需要时才激活并将所需专家权重在磁盘和内存之间传输。
- **上下文稀疏性预测**
  上下文稀疏性预测涉及预测推理计算中需要的少量且输入依赖的注意力头和多层感知器（MLP）参数集。例如，在[99]中，作者提出了Dejavu推理系统，在MLP和注意力块之后插入了轻量级预测器。根据当前块的输入，预测下一块的上下文稀疏性。使用预测的稀疏性，只有下一块中的一小部分MLP参数或注意力头被激活并加载到边缘设备的运行内存中进行推理计算。这种方法减少了计算开销和推理延迟，同时保持了大约相同的推理准确性。例如，使用Dejavu推理系统，OPT-175B的平均准确率在75%的稀疏性下不会降低。此外，与FasterTransformer相比，Dejavu推理系统可以将OPT-175B的推理延迟减少约一半，达到大约75%的稀疏性。
- **并行解码**
  例如Medusa，并在LLM的最后隐藏状态之上引入了额外的解码头。在推理期间，每个额外的解码头可以并行预测其指定位置的多个后续标记。这些预测被组装成候选项，然后与基于树的注意力机制并行处理。在验证了所有候选项之后，接受一个合理的候选项进入下一个解码阶段。此外，

Lookahead Decoding将自回归解码表述为非线性系统，并通过固定点雅可比迭代法求解。在每个推理步骤中，LLM并行生成几个不相交的n-gram，并从n-gram池中并行验证有希望的n-gram，该池缓存了历史生成的n-gram。

- **稀疏注意力**
- **KV缓存优化**
  减少KV缓存大小的一种方法是通过**KV缓存压缩。**例如，在[103]中，作者开发了一种2位量化算法，分别对每个通道的键缓存和每个标记的值缓存进行量化。该算法可以在几乎相同的推理准确性下减少2.6倍的峰值内存使用。此外，为了在压缩过程中保留层特定的信息，作者在[104]中引入了MiniCache。它将连续层中相同位置的高相似性键和值缓存合并为单个缓存，而那些具有重大语义意义的保持不合并。**通过4位量化**，这种方法可以在确保几乎无损模型性能的同时减少41%的内存使用。另一种方法是**KV缓存驱逐**，它采用驱逐策略动态选择KV缓存。例如，在[105]中，作者提出了Scissorhands，它在固定预算内维持KV缓存内存使用。**当缓冲区满时，从缓存中丢弃非影响力的标记。**这种方法可以在不降低性能的情况下将KV缓存的推理内存使用减少5倍。

4.2 高效微调

与推理相比，在设备上进行LLM训练需要显著更高的内存和计算资源。例如，计算LLM OPT-13B[118]的梯度消耗的内存是推理所需内存的12倍。然而，由于LLM微调所需的计算资源远小于全参数训练，因此在设备上部署LLM时广泛采用LLM微调。

参数高效微调（Parameter-efficient Fine-tuning, PEFT）

PEFT已成为LLM微调的突出解决方案，它通过**在微调过程中仅更新少量参数来减轻计算负担。**流行的PEFT技术可以分为三种主要类型，即加性PEFT、选择性PEFT和重参数化PEFT

- **加性PEFT**：为了减轻微调的计算负担，**加性PEFT在LLMs中引入了参数极少的可训练组件，同时保持预训练LLM参数冻结。**加性PEFT可以根据引入组件的局部性进一步分类为三种类型，即适配器调整、提示调整和前缀调整，如下图所示。

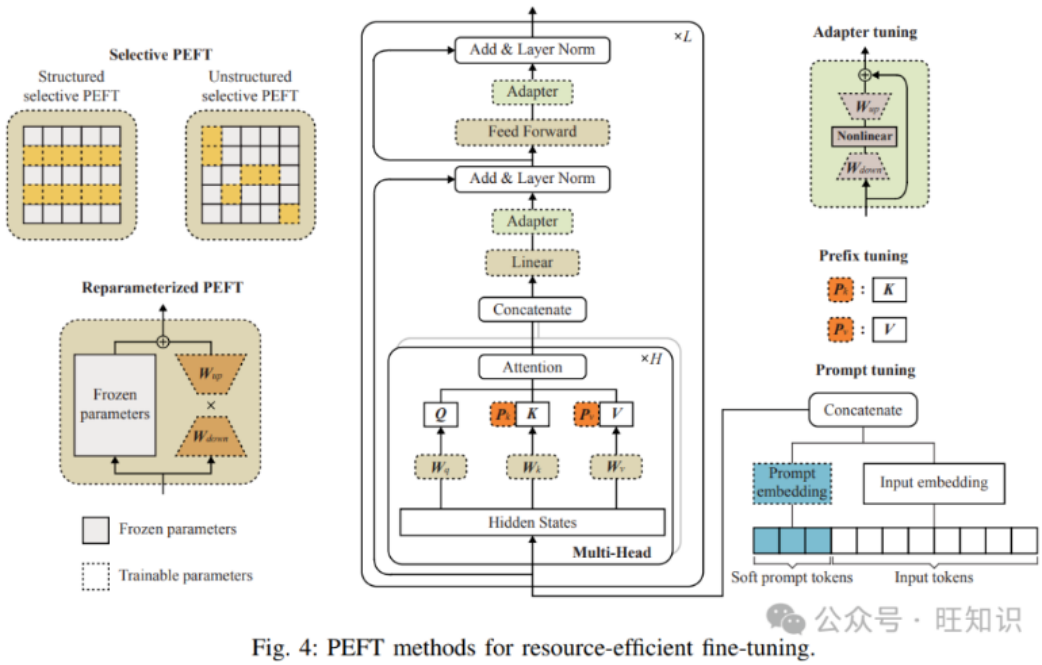


Fig. 4: PEFT methods for resource-efficient fine-tuning.

适配器调整（Adapter Tuning）：适配器调整通过在Transformer层中插入适配器模块并冻结其他参数。在这种方法中，只有在微调期间更新适配器。

提示调整（Prompt Tuning）：在输入标记的开头添加软提示标记进行微调[108]。这种方法利用了LLMs基于前一个标记进行编码和生成的特性。

前缀调整（Prefix Tuning）：在每个Transformer层的多头自注意力（MHA）的键和值中添加可训练的前缀参数。尽管这种方法比提示调整增加了更多的可训练参数，但它允许直接修改LLMs内的表示，使LLMs能够更精确地响应特定任务。

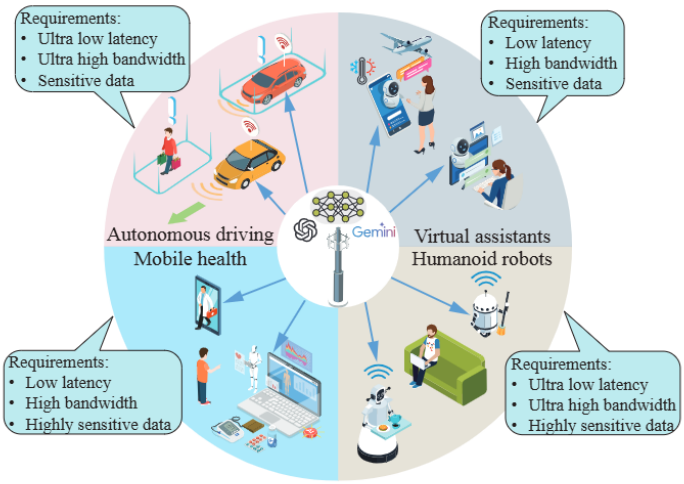
- **选择性PEFT**：尽管加性PEFT可以减少微调的参数数量，但它通过增加更多参数引入了**额外的推理延迟。**为了解决这个问题，选择性PEFT**通过冻结大多数参数并仅更新一个较小的参数子集来保持模型架构。**选择性PEFT可以分为非结构化选择性PEFT和结构化选择性PEFT。
    1. **非结构化选择性PEFT**：非结构化选择性PEFT单独确定可训练参数的选择，这可以提高微调模型的性能。例如，[110]中的作者将PEFT中的可训练参数选择问题重新表述为一个优化问题，并提供了一个二阶近似方法来解决这个问题。**通过选择稀疏的可训练参数，稀疏微调模型的表现优于完全微调模型。**
    2. **结构化选择性PEFT**：结构化选择性PEFT选择规则的参数组合，例如特定模型层，以提高在设备上LLM部署的硬件计算效率。
- **重参数化PEFT**：重参数化PEFT技术**利用低秩矩阵来减少模型微调过程中的可训练参数数量。**在LLMs的重参数化PEFT中最著名的方法之一是LoRA。对于LLM中的预训练权重矩阵，**LoRA引入了两个额外的可训练矩阵，其秩远小于预训练权重矩阵的秩。在微调过程中，预训练权重矩阵被冻结，只有新引入的两个低秩矩阵是可训练的。**这种方法允许有效模型微调，因为更新低秩矩阵所需的计算能力远小于更新预训练权重矩阵。此外，**LoRA不会增加任何额外的推理延迟，因为在推理期间LoRA的微调权重会合并到LLMs的原始权重中。**由于其显著的优势，LoRA启发了众多后续研究工作。**例如，Quantized LoRA (QLoRA)** [141]旨在通过结合量化技术与LoRA，最小化内存消耗，使得具有65B参数的语言模型可以在24小时内使用48 GB GPU进行微调。微调后的LLM在评估任务上达到了99.3%的ChatGPT性能，证明了QLoRA的有效性。

零阶优化（Zeroth-order optimization）

零阶优化是一种新颖的模型训练技术，它**仅通过前向传播来估计梯度更新。**这种方法大大减少了计算负担，因为前向传播，相当于推理，所需的计算资源远少于训练过程中的反向传播。**具体来说，与流行的一阶优化器（如Adam）相比，零阶优化器在训练过程中不需要存储反向传播的中间结果，显著减少了LLM训练中的内存消耗。**例如，在[87]中的作者提出了零阶优化器MeZO，它采用同时扰动随机近似来仅通过前向传播估计模型梯度，并使用估计的梯度来更新模型参数。与使用Adam进行微调相比，使用MeZO进行微调的模型在11个任务中的7个上表现出竞争力，同时只使用了1/12的运行内存，并且只造成了不到1%的准确性降低。此外，为了进一步提高LLM微调的效率，零阶优化技术可以与PEFT方法结合使用，如LoRA和前缀微调[87]。

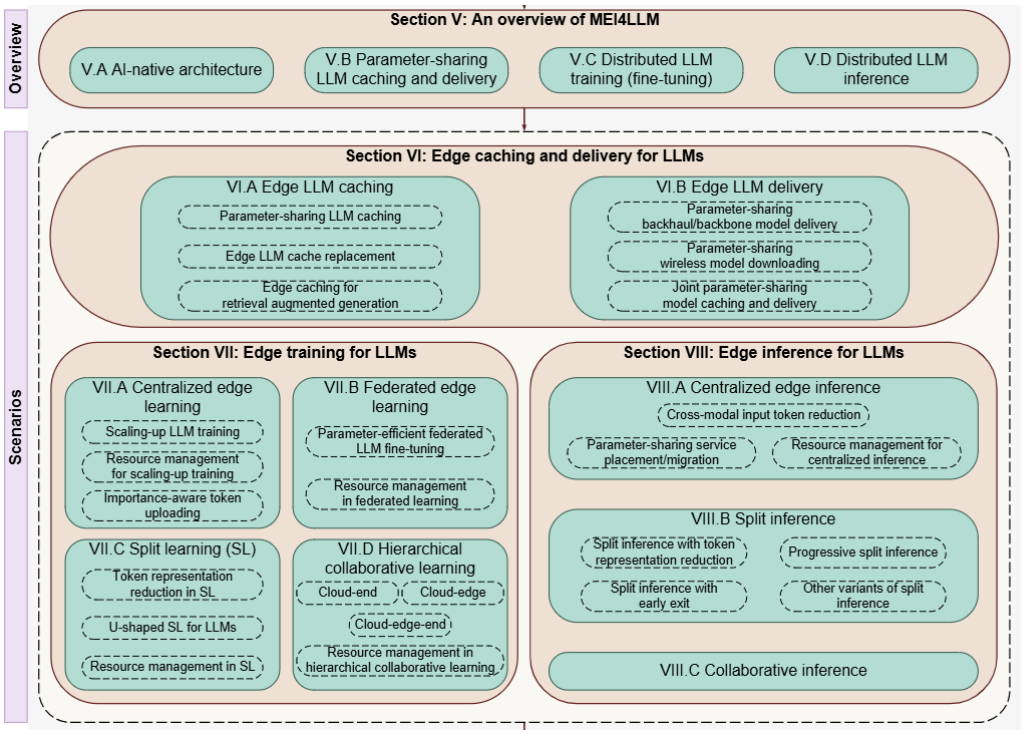
5. 应用场景

下图展示了四个关键的由LLM赋能的应用，同时集中讨论了三个方面：延迟要求、带宽成本和隐私要求。

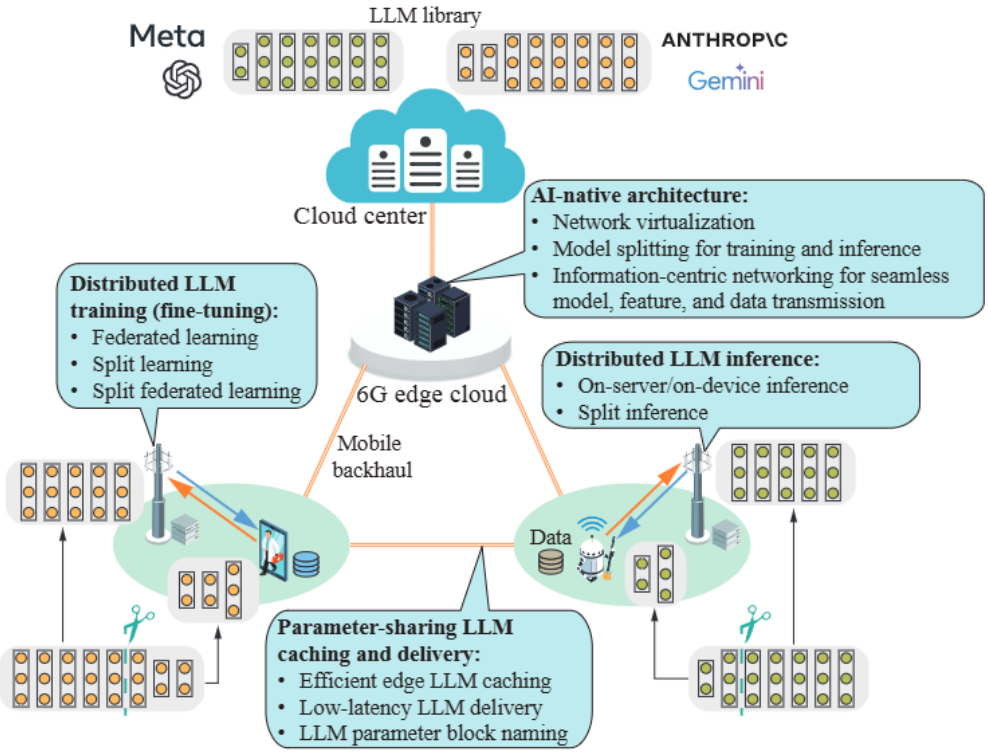


	Mobile health	Humanoid robots	Virtual assistants	Autonomous driving
Latency reqs.	≤400 ms [143]	10-100 ms [79]	200 ms [144]	10 ms [79]
Bandwidth reqs.	10-50 Mbps [145]	80 Mbps-12 Gbps [79]	144 Mbps [146]	80 Mbps-12 Gbps [79]

Fig. 5: Killer LLM-empowered applications demonstrating the need for deploying LLMs at the network edge and the corresponding latency and bandwidth requirements on practical cases.



6. MEI4LLM 概述



AI原生架构

下一代边缘网络将以端到端的方式支持AI服务。6G的目标应该是以最小的通信、计算、存储和能源需求，支持包括LLMs在内的AI，提供卓越的性能。因此，6G通常设想为"面向任务"的架构，而不是最大化吞吐量或最小化延迟，设计目标可以是通过实施最优的分布式计算、特征提取和资源分配方案，在多样化的资源约束下，最小化LLMs输出标记的交叉熵。为了实现这一目标，**网络虚拟化**对于提高资源利用率、灵活性和管理性至关重要。遵循软件定义网络的设计原则，MEI4LLM具有一个中央控制器，协调全网范围内的计算资源和数据传输，具有解耦的控制和数据平面。通过收集全球网络知识，例如LLMs的准确性、各种量化水平、用户对LLM服务的要求、信道条件、用户的电池状态和计算资源的可用性，**控制器在分布式边缘计算系统中分配和协调模型训练/推理和交付**，边缘路由器和服务器之间交换中间破碎数据（即中间激活和反向传播梯度）、模型参数或用户数据。进一步的边缘网络将发展为"神经边缘"[159]，神经网络层分布在边缘节点上，进行协作计算。类似于拥有许多GPU以支持大规模LLMs的云数据中心，MEI4LLM必须具有灵活的**模型分割**，以支持在分布式边缘设备和服务器上进行训练和推理。空中接口和网络设计应该原生支持包括LLMs在内的AI模型的联邦学习、分割学习和分割推理。由于模型训练和推理对数据包错误具有鲁棒性，面向任务的无线传输，例如在切割层的破碎数据，可以通过适当的错误控制进行，以实现最佳效率-可靠性权衡。大规模模型的最佳模型分割、放置和数据路由应该在边缘网络上得到协调支持。最后，可以**实现信息中心网络，以确保跨边缘网络的模型、特征和数据的无缝传输，有效交付LLMs。**在这方面，MEI4LLM应该支持LLM参数块命名和基于名称的传输协议。通过为每个LLM参数块分配名称，MEI4LLM架构中的中央控制器可以将参数请求转发到其缓存位置，减少在网络和最终用户之间交付大规模模型的延迟和带宽消耗。

参数共享的LLM缓存和交付

考虑到边缘设备的有限存储容量和频繁的模型微调，应将感兴趣的LLMs及时地从它们的位置传递到需要它们的地方。此外，考虑到RAG，外部知识源也应缓存在网络边缘，确保一旦LLM应用程序需要，能够及时获取数据/知识。模型/数据交付可以通过有线回传或无线接入网络进行。**LLMs的缓存和交付必须利用参数块可以在各种下游LLMs之间共享**，甚至在同一个LLM内重用[38]这一独特特性，**通过减少重复LLM参数块的缓存和交付成本，提高边缘网络的存储和通信效率。为了实现快速模型交付，MEI4LLM可以构建一个查找表，为LLM参数块分配名称，以便于内容搜索和管理，遵循信息中心网络的原则。**通过

这样做，MEI4LLM范式将LLMs放置在适当的地点，从附近的边缘服务器检索感兴趣的LLMs，并启用LLM参数块到移动用户的路由/多播。

分布式LLM训练（微调）

预计6G MEI系统可以高效地微调LLMs，以适应当地环境。当推理准确度下降或在一定时期后当地环境发生变化时，可以触发边缘LLM微调。例如，由LLMs支持的虚拟助手应**定期微调**，以更好地适应新闻媒体、顶级当地餐厅和热门景点的新趋势，从而提高决策和与用户的交互。由LLMs支持的移动健康应用程序应个性化，以提供更好的预测和健康或健身建议。在下一代移动网络中，边缘训练对LLMs必须回答两个问题：1) **如何保护用户隐私和数据所有权，2) 如何通过边缘节点的协作支持大规模模型训练。**为了增强用户的数据隐私，**联邦学习（FL）和分割学习（SL）**是两种有前景的分布式学习框架，可以在网络边缘实现。具体来说，FL允许边缘设备在本地训练模型，同时只与边缘服务器共享模型参数以进行聚合，从而利用集体智慧而无需共享个人数据。**SL及其变体分割联邦学习（SFL），可以实现设备-服务器共同训练而不共享本地原始数据，**特别适合于边缘设备的大规模LLM微调[161]，因为模型分割允许在不同的边缘节点之间平衡工作负载。

分布式LLM推理

为了适应资源密集型的LLMs，边缘服务器和边缘设备必须以协调的方式执行分布式推理，这取决于通信计算工作负载和隐私要求。边缘推理有几种不同的方式。服务器端推理要求用户将原始数据上传到服务器。这种方法消除了边缘设备的计算负担，但可能违反用户的隐私要求。例如，多模态LLMs可能在家庭环境中收集敏感的音频和视频数据，用户通常不愿意共享。相反，设备端推理保护隐私并消除通信成本，同时将繁重的计算工作负担施加在边缘设备上。分割推理是3GPP 5G技术规范[79]中的一个关键AI推理框架，介于两者之间，**边缘设备和服务器持有AI模型的部分。分割推理涉及将边缘设备的特征上传到边缘服务器进行共同推理。**为了促进LLM推理，MEI4LLM可以根据通信计算资源状态和隐私要求，适当选择这些方案。

【补充： 分割训练 和 推理： 在分割学习中，模型被分割成两部分：一部分在数据拥有者的设备上执行（通常是前半部分），另一部分在服务器或云端执行（通常是后半部分）。这种分割允许数据本地处理，然后仅将中间激活（intermediate activation）传输到云端，避免了原始数据的传输，从而更好地保护数据隐私。 分割推理类似】

7. 边缘缓存和交付LLMs

边缘LLM缓存和交付在LLMs的训练和推理中发挥着不可或缺的作用，是边缘LLM部署的基石。与传统的边缘服务/内容缓存和交付相比，**边缘LLM缓存和交付的主要区别在于利用参数共享性，这在LLMs中非常普遍，目的是提高边缘网络的存储和通信效率。**虽然参数共享性在传统DNNs中也存在，但由于PEFT技术的广泛采用，它在LLMs中更为普遍和重要，需要我们特别设计关注。

TABLE III: Summary of related works for edge LLM caching and delivery.

Scenarios	Techniques	Ref.	Objectives
Edge LLM caching	Parameter-sharing LLM caching	[162]	Proposes TrimCaching framework for LLMs, where shared parameter blocks of LLMs only need to cached once in an edge server for storage efficiency.
Edge LLM delivery	Parameter-sharing wireless model downloading	[163]	Proposes a model multicasting and assembling framework, where shared parameter blocks of models requested by users are multicast to users, and the specific parameter blocks are unicast to each user separately.
		[164]	Compresses model weights in low bitwidth for fast model downloading.

7.1 边缘LLM缓存

边缘模型缓存可以通过提前将AI模型分发到无线边缘服务器来实现低模型下载延迟。与计算卸载的服务放置不同，边缘模型缓存侧重于缓存AI模型以**供最终用户从边缘服务器下载。**AI模型缓存的设计目标是在服务等级协议(QoS)要求内为更多用户提供模型。这种范式使用户能够直接从边缘服务器获取AI模型，而不是访问远程云数据中心，这会产生过度的下载延迟。然而，实施边缘LLM缓存面临几个挑战：

1. **有限的LLM缓存存储容量**：服务提供商旨在将尽可能多的流行LLMs放置在边缘服务器上，以提高缓存命中率并减少用户的模型下载延迟。然而，LLMs的巨大规模对边缘服务器上的存储构成了重大挑战。
2. **高LLM边缘缓存(重新)放置成本**：随着时间的推移，先前缓存的LLMs可能不再符合不断变化的用户请求。为了解决这个问题，服务提供商可能会更换边缘服务器上的LLMs，以更好地适应最新的请求。然而，这些大规模模型的放置导致了相当大的通信开销，并给移动回传网络带来了巨大负担。

解决方案：

- **参数共享LLM缓存**：可以采用参数共享模型缓存来提高网络边缘的存储和传输效率。PEFT（如LoRA）被广泛采用，以使LLMs适应下游任务。在LoRA中，预训练的LLM参数被冻结，只有新引入的参数是可训练的，通常占原始LLM参数的不到1%。在TrimCaching框架中，**只在一个边缘服务器上缓存跨LLMs的共享参数块的一个副本，从而提高存储效率**，如图8所示。

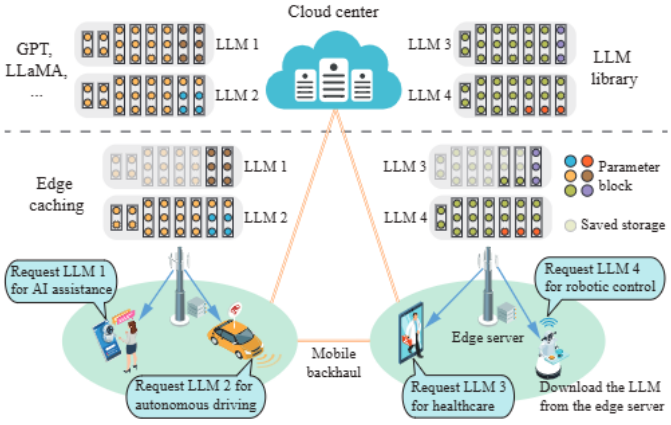


Fig. 8: The TrimCaching mechanism for caching LLMs in wireless edge networks. Popular LLMs are placed on edge servers, where users can download the requested LLMs from the edge network. To enhance storage efficiency, shared parameters across LLMs are cached only once on an edge server.

- **边缘LLM缓存替换**：由于模型的受欢迎程度会随着时间而变化，边缘缓存中的另一个基本研究问题就是LLM替换。通过用新数据替换过时的内容，边缘服务器可以不断地用新内容刷新它们的缓存，以满足不断变化的用户请求[169]。两种最经典的替换策略是基于最近性和频率的策略，它们分别移除最近最少使用(LRU)对象和最少频繁使用(LFU)对象，然后用更新的内容替换它们。然而，**这些策略没有考虑到跨LLMs的共享参数块以及不同边缘节点之间的协作缓存。**提高替换性能的一个方向是，在一定时间后重新进行集中式主动缓存，例如[160]中的方案，但在高用户流动性下，这些方法可能涉及高系统复杂性和高通信成本。另一个方向是开发分布式算法，通常基于马尔可夫决策过程或强化学习，以在不知道其他边缘节点的完整信息的情况下做出替换决策[170]。
- **针对RAG的边缘缓存**：将LLMs与RAG集成，使LLMs能够从外部知识源检索相关信息，对于生成可靠和最新的响应而不重新训练/微调至关重要。然而，由于从远程云检索信息可能耗时，应将最受欢迎的外部知识缓存在网络边缘，以使LLMs能够获取最真实、准确和最新的内容。有趣的是，这个边缘缓存问题自然不同于传统的缓存问题，因为缓存应该通过考虑LLMs的训练状态或内部知识来优化。具体来说，**如果LLMs已经能够记住或容易推断出某些内容，这些知识可以从外部知识源中移除，以节省网络边缘的存储空间，从而提高缓存外部知识源的存储效率。**此外，边缘服务器可以缓存其关联用户经常请求的特定外部知识源，这可以显著提高不同地区用户响应的QoS。另一方面，针对RAG的边缘缓存与即时LLM微调紧密耦合，即我们需要选择哪些数据来更新过时的LLMs，以及哪些数据在边缘服务器上缓存RAG。考虑到这一点，值得研究一个联合LLM微调和知识源缓存问

题，在RAG的背景下，以提高LLMs在延迟约束下的可靠性，考虑到从边缘/云服务器检索外部知识源的延迟以及通过向LLMs提供新数据进行微调的成本。

7.2 边缘LLM交付

从缓存获取模型到最终用户的重要步骤是延迟高效的模型交付。这个过程包括在回传/骨干网络内进行模型路由和通过无线接入链路进行模型下载，面临以下挑战：

1. **过度的回传/骨干交付延迟**：当请求的LLMs不在相关边缘服务器上缓存时，需要在边缘网络内进行LLM路由。然而，与传统AI模型相比，LLMs的模型尺寸显著更大。因此，LLM路由需要在适度的回传流量中在边缘服务器之间进行。
2. **显著的无线下载延迟**：AI模型下载需要在低延迟下完成，以满足最终用户的QoS要求。正如3GPP所设想的，自动驾驶应用要求在1秒内完成AI模型下载。然而，LLMs的巨大模型尺寸阻碍了快速模型下载，使其极难满足严格的服务延迟要求。

解决方案：

- **参数共享回传/骨干模型交付**：为了减少回传/骨干网络内的模型交付成本，可以开发利用LLMs间参数共享的参数高效模型交付。如下图所示。当一个边缘服务器不缓存LLM但缓存其他具有共享参数块的LLM时，只需要传递缺失的LLM参数块，从而减少数据交付成本。例如，对于使用LoRA微调的LLMs，如果共享的骨干已经在边缘服务器上缓存，则只需要传输特定的LoRA参数。此外，在传递参数块时，参数块可以从不同的服务边缘服务器获取。只要所有需要的参数块到达目的地（例如，请求的服务器或用户），就可以组装整个LLM。因此，考虑到多跳回传/骨干通信网络，可以开发缓存感知数据路由，利用重叠参数块的多播来提高网络吞吐量。

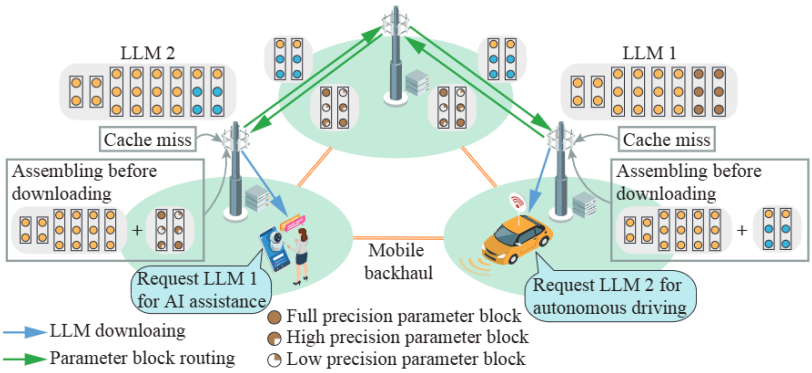


Fig. 10: The parameter-sharing backhaul/backbone LLM delivery framework. In this example, the AI assistant and autonomous driving applications request LLM 1 and LLM 2, respectively, which are not cached on the associated edge server. Therefore, LLM 1 and LLM 2 need to be delivered from another edge server. In this framework, to reduce communication overhead and latency, only the specific LLM parameter blocks need to be transmitted since the shared backbone/parameter blocks are already cached on the associated edge server. The entire LLM can be assembled before downloading, or on edge devices once all the needed parameter blocks have been received. To further reduce delivery latency, parameter-sharing LLM delivery can be combined with various compression techniques, where the compression ratio can be determined by jointly considering model performance and channel/backhaul conditions.

- **参数共享无线模型下载**：为了减少从基站（边缘服务器）到用户的无线模型下载成本，必须考虑参数共享无线模型下载。如下图所示，为了降低下载延迟，关键思想是多播可重用参数块，从而实现及时下载。

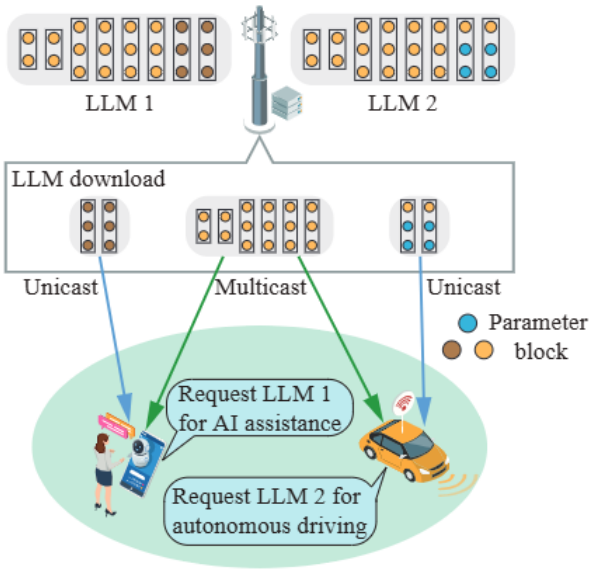


Fig. 11: Parameter-sharing wireless LLM downloading.

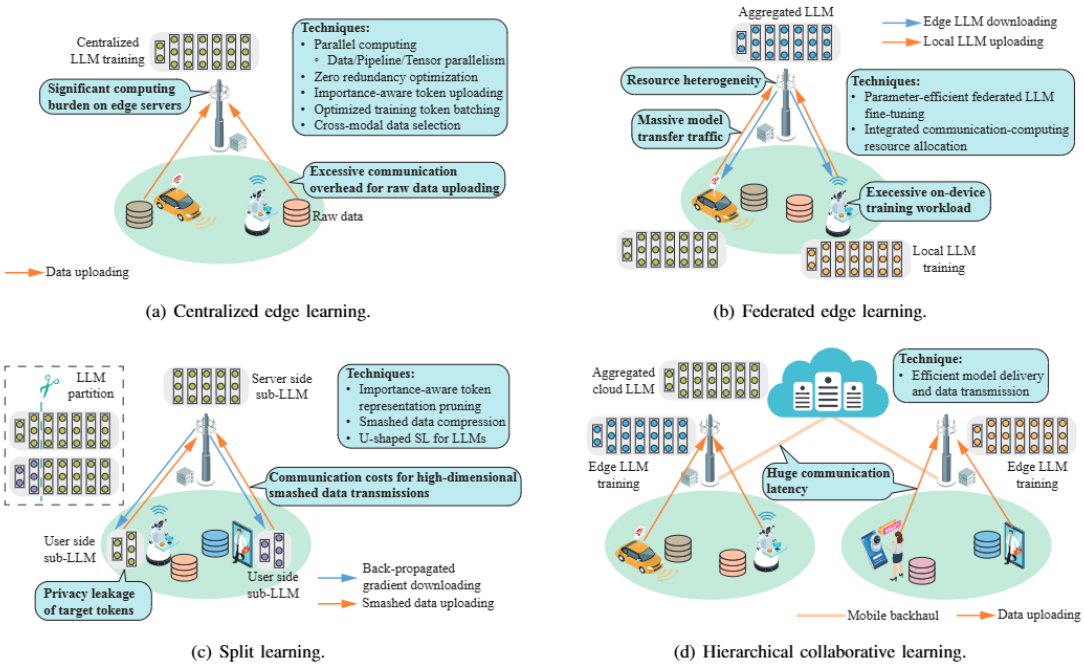
在[163]中，作者提出了一个模型多播和组装框架，利用AI模型间的共享参数，即论文中考虑的Transformers。**该框架在多播共享参数块给多个请求用户的同时，对每个用户单独多播特定的参数块。**然后用户组装下载的参数块以获取所需的LLM。参数共享和下载延迟之间存在权衡。**尽管具有更多共享参数块的模型可以以更低的延迟下载，这也将降低模型在下游任务中的性能。为了减少总模型下载延迟并确保QoS要求，提出的框架旨在在下游推理任务的准确性QoS要求内，最大化不同模型间相同参数块的发生。参数共享LLM交付/下载也可以与各种LLM压缩技术集成。**

- **联合参数共享模型缓存和交付**：参数共享模型缓存和交付在有线网络、无线网络或两者混合中紧密耦合。一方面，模型放置显著影响回传和无线链路上的边缘网络流量。另一方面，无线资源分配和数据路由影响模型放置的最优决策。考虑到跨LLMs的共享参数，这个联合问题与现有的缓存和交付方案有显著不同，因为共享参数的各种内容以及一个模型可以通过从不同源节点获取各种块来恢复。在多跳网络中，可以通过考虑参数共享性和共享模型部分的多播来研究联合缓存和数据路由问题。在缓存辅助的蜂窝网络[174]中，可以联合优化模型放置、前传/回传成本和无线资源分配，以促进快速AI模型下载到边缘设备。

总结： 上述技术旨在1) 只传输模型的一小部分（即任务特定的参数块），2) 在通信网络中存储或多播共享参数块，3) 联合优化LLM能力和外部知识源。所有这些特性都源于利用LLMs中的"可重用知识"来尽可能节省带宽和存储资源。

8. 边缘训练LLMs

边缘训练在网络边缘执行模型训练，以从数据源中提取智能。边缘LLM训练与传统边缘训练的主要区别在于AI模型的大规模，可能太大而无法适应边缘服务器，以及在无线网络上优化PEFT。如图所示，我们通过四类讨论边缘LLM训练：集中式边缘学习、联邦边缘学习、分割学习和分层协作学习。



(a) Centralized edge learning.

(b) Federated edge learning.

(c) Split learning.

(d) Hierarchical collaborative learning.

8.1 集中式边缘学习

在MEI系统中最直接的模型训练方法是从边缘设备收集数据并在边缘服务器上进行模型训练。虽然边缘服务器通常比边缘设备更强大，但集中式边缘学习面临几个关键挑战：

1. **边缘服务器的重大计算负担**：LLM训练或微调需要大量的计算资源和存储/内存容量。例如，以FP32训练Llama-2 7B的经验需求112 GB的GPU内存[195]，这对于边缘服务器来说可能具有挑战性，因为一个强大的H100 GPU只有80 GB的内存。
2. **原始数据上传的过度通信开销**：考虑到多模态原始感知数据，需要将大量数据上传到集中式数据中心。

解决方案：

- **扩展LLM训练**：扩展训练对于集中式边缘训练中的LLM训练至关重要。由于内存和计算限制，单GPU上训练大规模LLMs极具挑战性。因此，在集中式边缘训练的背景下，使用分布式计算和内存资源（如边缘服务器上的多个GPU）进行扩展可以加速训练过程，使在边缘网络中训练LLMs成为可能。扩展LLM训练可以分为两种主要方法：并行训练和GPU内存优化。

  并行训练：考虑到LLMs的极端训练工作量，必须采用并行计算来利用跨边缘节点的资源，将LLM训练分割以减少训练延迟并共享所需的内存空间。三个最突出的并行计算策略是数据并行、流水线并行和张量并行。

  GPU内存优化：有效的GPU内存优化，如Zero Redundancy Optimization (ZeRO)，可以采用以减少边缘LLM训练中分布式GPU的内存使用[179]、[180]、[198]。在[179]中，Rajbhandari等人开发了ZeRO，其中**每个处理器在训练期间只持有优化器状态、梯度和参数的一部分，其余的可以根据需要从其他数据进程中获取，从而减少每个处理器所需的GPU内存空间。**为了进一步减轻GPU内存的压力，在[180]中，作者提出了ZeRO-Offload，它通过利用CPU内存有效地优化GPU内存。**ZeRO-Offload在训练期间在GPU上划分梯度和优化器状态，并将它们卸载到CPU内存中。在反向传播期间，梯度在GPU上计算和平均，每个GPU将其所属的部分平均梯度卸载到CPU内存中。一旦CPU上有了梯度，优化器状态就直接在CPU上更新，然后才收集回GPU。**

- **扩展LLM训练的资源管理**：与基于云的方法不同，资源管理在并行计算LLMs的网络边缘起着重要作用。尽管前述基于云的方法可以应用于相互连接的边缘服务器，但必须仔细考虑边缘服务器的**异构通信和计算能力**，以进行系统优化，这在大规模GPU集群中通常被过度简化。例如，在流水线并行中，将子模型放置在多个边缘服务器上进行LLM训练时，中间的破碎数据应该在边缘服务器之间通过显著异构的有线/无线通信链路交换。在这方面，**模型分割和放置应该基于计算-通信资源约束进行明智的优化。**
- **重要性感知的token上传**：集中式边缘训练LLMs的另一个挑战在于输入标记上传中的通信瓶颈。特别是，多模态LLMs涉及从边缘设备上传大量多模态数据，如文本、音频、高维图像/视频和激光雷达，这可能导致网络拥塞和对延迟敏感应用的不可接受的延迟。为了解决这个问题，可以扩展边缘学习中的重要性感知训练数据传输方案到LLMs的背景中。在[181]中，作者展示了通过为具有更高重要性的训练数据样本分配更多的无线电资源，可以提高训练期间的收敛速度和模型精度。

8.2 联邦边缘学习

在FL中，客户端通过将本地模型更新发送到边缘服务器进行聚合，共同训练全局模型。当FL遇到LLMs时，需要解决一些关键挑战：

1. **设备上的过度训练工作量**：与深度学习加速器的内存带宽（高达7.8 TB/s）相比，嵌入式边缘设备的内存带宽仍然要低得多（高达0.2 TB/s），导致严重的训练时间惩罚。
2. **大量模型传输流量**：从大量边缘设备上传LLMs到边缘服务器进行模型聚合，给电信基础设施带来了沉重的通信负担，对移动用户来说可能非常昂贵。
3. **资源异构性**：在FL中通常观察到落后者问题，其中训练时间由具有最稀缺通信-计算能力的最慢客户端确定。

解决方法：

- **参数高效联邦LLM微调**：为了解决上述挑战，可以采用参数高效联邦LLM微调。**参数高效联邦LLM微调将PEFT集成到FL中，使每个客户端只更新和上传一小部分参数，从而减少通信和计算开销。**
- **FL中LLMs的资源管理**：在LLMs的背景下，优化PEFT在无线网络上的FL仍然是一个重要的未来方向，目前仍处于起步阶段。原理是**通过考虑无线信道条件和模型训练状态来调整可训练参数的比例。**直观地说，更大的可训练参数集可以带来更好的训练性能，同时也会引入更大的通信和计算延迟。在[211]中，基于模型参数在模型收敛之前逐渐稳定的观察，Chen等人提出了一种**自适应参数冻结方案，在训练过程中冻结非同步稳定参数，消除了同步完整模型的需要。**然而，这项工作没有考虑无线网络中的动态信道条件。考虑到LLMs中的PEFT技术，如LoRA，可训练矩阵的秩在很大程度上影响训练准确性和FL中的通信-计算延迟，如[202]所示。因此，**一个重要的研究问题是如何联合优化LoRA中LLMs的秩和无线网络上FL的无线电资源分配。**

8.3 分割学习

虽然FL可以与各种高效微调技术结合用于训练LLMs，但对于轻量级边缘设备来说，它仍然非常资源密集。具体来说，像GPT-3或BERT这样的模型包含数十亿个参数。即使是采用PEFT，边缘设备（如智能手机或IoT设备）也难以在本地执行计算密集型的参数更新[212]。为了解决这些问题，SL可以是边缘网络中LLM训练的有前景的范式，它通过边缘服务器和边缘设备的协作共同训练大规模模型[213]。**SL允许边缘服务器基于模型分割从边缘设备接管主要的训练负载。与FL不同，SL只在边缘设备上放置一个子模型进行训练，从而大大减少了边缘设备的负载。传统的SL涉及边缘服务器和边缘设备之间的顺序交互，这是由于空闲边缘设备的等待时间而成为一个主要瓶颈。SL的变体，包括并行分割学习(PSL)、和SFL，可以应用以实现并行训练，同时利用多个边缘设备的资源。**显然，与FL一样，SL也可以与第三节中的PEFT或其他资源高效技术集成，以进一步减轻边缘设备的负载。例如，边缘设备只需要执行前向传递，冻结客户端参数，大大减少了计算工作量和内存使用。

挑战：

1. **高维破碎数据传输的通信成本**：尽管模型分割利用了分布式计算资源，并减轻了边缘设备的计算负载，但上传破碎层破碎数据的通信开销可能是一个主要瓶颈。考虑到GPT-3 Medium和具有100个数据样本的边缘设备，每个数据样本有1024个标记，一轮训练中破碎层的总破碎数据量大约为400 MB。
2. **目标标记的隐私泄露**：虽然通常很难基于SL中接收到的破碎数据恢复LLMs的原始训练数据，但存在目标标记泄露（即标签泄露）的隐私风险。在SL过程中，一个常用的LLM分割方案是将具有输入模块的子LLM放在边缘设备上，将具有输出模块的子LLM放在边缘服务器上。在这种情况下，边缘设备需要将输入数据的目标标记上传到边缘服务器以进行LLM训练，导致目标标记泄露。

解决方法：

- **SL中的标记表示减少**：首先，可以利用重要性感知的token表示剪枝来消除上传的破碎层中不重要的token表示。**发送最具信息量的破碎数据**，以执行SL，这种方法通过减少与基准相比的通信成本约50%，提高了性能。其次，可以在传输前采用破碎数据压缩。在这方面，**量化**可以被采用以有效减少SL/SFL/PSL中的通信开销。
- **LLMs的U形SL**：在SL框架中，另一个常见问题是真实目标标记的隐私泄露。通常，边缘设备需要将训练样本的目标标记传输到服务器以计算损失函数，导致严重的隐私问题。例如，LLMs的真实目标标记可能是患者的疾病类型和健康建议，这被认为是敏感的个人数据。为了解决这个问题，**U形SL将输入模块的头部神经层或Transformer块和输出模块的尾部层/块留在边缘设备上，而只有中间层/块放在边缘服务器上，从而有效地保护了标签隐私。**
- **SL中LLMs的资源管理**：为了有效和高效地支持SL中的LLMs，需要联合设计SL和无线电资源分配。

8.4 分层协作学习

分层协作学习范式可以促进大规模的LLM训练。与之前的边缘仅范式相比，分层协作学习通过利用云、边缘服务器和边缘设备之间的协同作用，提供了改进的灵活性，以适应不同的任务复杂性和资源可用性。例如，计算密集型的训练任务可以卸载到云服务器，而相对容易的任务可以保留在网络边缘以节省通信延迟/带宽。此外，分层协作学习对于LLMs在global范围内学习全局知识是不可或缺的，其中LLM更新可以在中央云中同步。如下所述，分层协作学习可以分为三类：云-端协作、云-边缘协作和云-边缘-端协作。

1. **云-端协作**：云和边缘设备可以共同训练LLMs。为了减少边缘设备的计算工作量并增强用户隐私，Gao等人在[227]中提出了一个名为DLoRA的分布式PEFT框架。该框架在边缘设备上维护和微调个人PEFT模块，同时在云端存储LLMs的冻结参数。通过交换激活和梯度，边缘设备和云端可以共同训练LLM。此外，为了减少边缘设备的通信开销，作者采用了Kill and Revive机制，动态识别和微调最相关和最重要的PEFT模块。在[191]中，Wang等人提出了一个用于多模态LLMs的云-端协作学习框架，其中云端有一个较大的多模态LLM，而边缘设备有一个较小的多模态LLM。边缘设备，如机器人，将多模态数据上传到云端，以执行KD训练适配器，然后下载到边缘设备。为了减少数据上传中的通信开销，作者采用了不确定性引导的标记采样策略，使边缘设备只上传最具信息量的标记到云端。
2. **云-边缘协作**：云-边缘协作LLM训练使云端和边缘服务器能够共同训练LLMs。一方面，LLMs可以最初在云端进行预训练，然后通过在边缘服务器上进行额外的微调来提高性能。在[8]中，Xu等人提出了一个用于生成性AI模型的云-边缘协作训练和微调框架，其中**模型在云端预训练以学习通用特征，然后使用存储在边缘服务器上的上下文感知数据进行微调以实现定制化。**另一方面，**边缘服务器可以在当地训练LLMs，然后将模型更新发送到云端进行知识共享**，例如FL框架中的模型聚合，这可以提高LLMs的通用性。在[192]中，**多个边缘服务器的边缘模型使用门控神经网络和线性投影连接集成成一个适用于多任务和多模态学习的大规模模型。云端使用云端公共数据集训练这个大规模模型，然后将特定任务的轻量级模型分发回边缘服务器进行个性化微调。**

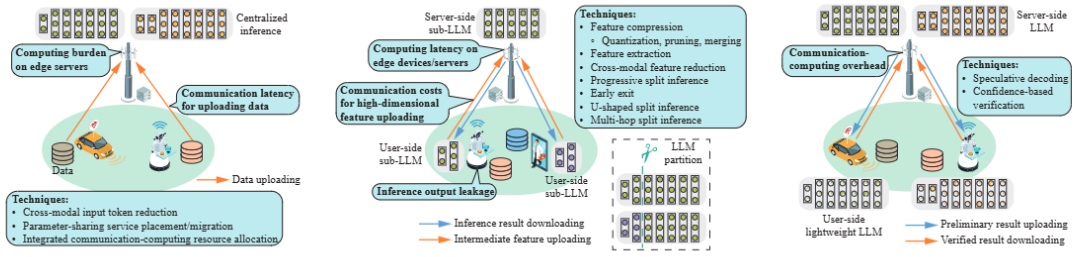
3. **云-边缘-端协作**：具有三级架构的云-边缘-端协作学习带来了更多分布式计算资源和资源管理的灵活性。在[193]中，作者提出了一个分层FL框架，其中边缘服务器聚合它们各自的关联端用户本地训练的模型，云端聚合来自不同边缘服务器的模型。此外，为了减少分层FL框架中的通信成本，作者在[194]中采用了模型量化来压缩模型大小，以实现高效的模型聚合，其中提供了一个收敛上界以优化聚合间隔。分层协作学习范式适用于LLM训练。由于某些LLM应用中的原始数据，如医疗保健，高度敏感，用户可以在保持私有数据在本地设备的同时，通过云端和边缘服务器的帮助参与训练过程[228]。然而，尽管有这些优点，**模型在云端、边缘服务器和边缘设备之间的传递和数据传输必须得到适当处理，因为传递LLMs的模型参数会产生显著的通信延迟，特别是在从云端到边缘服务器的链路上。**
4. **分层协作学习中LLMs的资源管理**：在分层协作训练中，模型聚合间隔（在FL或SFL中）和模型分割必须经过精心设计，以在减少延迟和数据流量成本的同时，最大化LLMs的训练准确性。例如，LLMs中上传参数的比例必须根据PEFT动态调整，考虑到模型训练状态、无线信道条件和主干网络中的互联网延迟。此外，必须仔细设计模型分割和上传的token表示剪枝在云端-边缘-端架构的不同层中。

总结

类似于其他AI模型的边缘训练，边缘LLM训练旨在在有限的通信-计算资源下优化训练准确性。然而，由于LLM训练的资源需求更大，**其基本原理应该是分割训练并"只更新一点点"。**换句话说，这个过程**必须借鉴大规模模型训练的智慧，即在多个边缘设备/服务器上并行训练，并在无线网络上进行PEFT。**这**意味着SL可以是LLM时代的重要边缘学习框架。为了使MEI适应LLMs，关键原则是启用基于边缘网络内动态通信-计算资源的灵活模型分割**，并联合考虑通信-计算资源和LLMs的训练状态来确定LLMs的微调部分。为了使网络优化落地，需要对LLMs的微调，如LoRA，有理论上的理解，即表征训练准确性与可训练参数百分比之间的关系，以确定无线网络上微调的最佳冻结比例。

9. 边缘推理LLMs

边缘LLM推理利用边缘计算为最终用户提供基于训练良好的LLMs的推理服务。边缘推理LLM与传统边缘推理框架的主要区别在于利用LLM的特性，如多模态性、参数共享性和自回归过程，来加速推理过程。边缘LLM推理可以分为三个框架：集中式边缘推理、分割推理和协作推理，如下图所示。



(a) Centralized edge inference. (b) Split inference. (c) Collaborative inference.

9.1 集中式边缘推理

在集中式边缘LLM推理中，边缘设备将其输入标记卸载到边缘服务器进行AI推理。集中式边缘推理涉及从边缘设备上传原始数据到边缘服务器，这与第三节中讨论的集中式边缘学习共享类似的挑战，即**上传数据的通信延迟和边缘服务器上的计算负担（包括过度的内存使用）。**

解决方法：

- **跨模态输入标记减少的LLM推理**：类似于集中式LLM训练，减少输入标记降低了快速边缘推理的数据传输量。通过在推理之前从输入文本序列和图像中移除一些不重要的标记，减少了需要卸载到边缘服务器的数据大小。已经表明，**输入标记修剪对LLMs的推理精度没有显著损失。**例如，在将图像输入到ViT之前，移除与图像内容无关的一些图像块可能不会降低预测结果。换句话说，在LLM推理中可以充分利用跨模态性，以消除LLM推理中的通信-计算冗余。

- **LLM推理的参数共享服务放置/迁移**：在多用户系统中提供服务时，边缘服务器上计算资源的稀缺性，例如内存空间，是一个主要问题。这一特点严重限制了边缘服务器能够支持的边缘LLM推理服务的数量。因此，适当设计集中式边缘推理中的服务放置对于容纳大型模型至关重要。此外，考虑到用户的移动性，服务迁移，即根据用户位置的变化将AI模型从一个地方迁移到另一个地方，也可以被研究。**很多工作并没有利用模型/任务之间的共享参数进行边缘推理。实际上，通过利用这一特性，具有大量共享参数的多个LLM可以被加载到服务器的内存中进行并发推理，从而显著提高通过服务更多用户请求同时进行的推理吞吐量。**这需要为LLM推理设计参数共享服务放置方案，联合考虑边缘网络的内存、存储、计算和频谱约束。
- **集中式LLM推理的资源管理**：为了迎合具有异构通信-计算资源和服务要求的更多用户，需要研究集成的通信-计算资源分配。集中式边缘推理的一个关键原则是**为具有延迟敏感任务的用户分配更多的频谱带宽和计算资源，以满足他们的QoS要求。**在[16]中，Fang等人提出了一种边缘LLM推理方案，用户可以在设计的通信和计算资源分配策略下将LLM推理任务卸载到边缘服务器。通过联合分配频谱带宽和计算资源进行推理任务，可以优化平均端到端延迟和推理精度。**在LLM推理的背景下，无线电资源可以与输入token修剪和参数共享服务放置联合优化。**一方面，可以为具有高维输入标记的用户分配更多的无线信道，以便上传，以实现高推理精度。**优化问题因此可以被制定为在计算-通信延迟约束下，通过联合优化用户的token修剪比例和无线电资源分配**，来最大化精度（或公平性）的问题。另一方面，当考虑参数共享服务放置时，通信延迟和存储/内存效率之间存在权衡。具体来说，尽管将需要相同LLM的多个推理请求卸载到服务器可以增强存储/内存效率，如上所述，数据传输可能会受到更长的通信延迟的影响，对用户的QoS不利。这激发了我们在考虑无线网络上的信道分配和数据路由时，在延迟约束下研究参数共享服务放置/迁移问题。

9.2 分割推理

分割推理是一种通过在边缘设备和边缘服务器上分别放置分割模型来卸载部分计算工作量的技术。最广泛采用的分割推理范式是双分割范式。边缘设备基于原始数据执行设备端子模型的推理，并将中间特征上传到边缘服务器以进行其余神经网络的处理。分割推理特别适合LLM推理。**一方面，与设备上LLM推理相比，分割LLM推理将大部分计算卸载到边缘服务器，从而减少了设备端的工作量，这对于计算密集型的LLM推理至关重要。另一方面，考虑到边缘网络中的LLM应用，如移动健康和自动驾驶，通常涉及高度敏感的个人数据，分割LLM推理有效地缓解了隐私问题，因为边缘设备不需要与边缘服务器共享私有原始数据。**有两种突出的双分割方法用于分割LLM推理。首先，可以将LLMs中最资源密集的模块放置在服务器上，以充分利用其计算能力。**在Transformers中，解码器模块通常需要比编码器模块更多的计算资源。**因此，对于具有编码器-解码器架构的LLMs，如BART[247]，编码器模块通常可以缓存在边缘设备上，而解码器模块可以放置在边缘服务器上。对于基于解码器的LLMs，如GPT系列模型，文本和位置嵌入以及几个底部Transformer块可以放置在边缘设备上，因为它们的工作量相对较轻，而其余的Transformer块可以放置在边缘服务器上。其次，可以放置占用大量存储容量的子模型在边缘服务器上。例如，FFNs占LLMs参数的大约2/3，而FFNs中只有一小部分参数对最终推理结果有重要贡献[248]。因此，这些参数在FFNs中，数据量较小但对推理至关重要，可以放置在边缘设备上以节省存储/内存资源。

分割推理在LLMs中遇到了与SL中讨论的类似的挑战，即高维特征上传的通信成本、边缘设备/服务器上的计算延迟和推理输出泄露。接下来，我们将介绍解决这些挑战的技术。

- **带有标记表示减少的LLMs分割推理**：在考虑分割推理时，有各种方法可以减少LLMs中切割层的标记表示（潜在表示）的体积以上传到边缘服务器。首先，可以通过量化、剪枝和**合并【合并可以通过合并具有相似语义含义的冗余token表示来压缩中间输出的信息。】**压缩Transformers中间层的输出：上传压缩的中间层输出可以大大减少通信开销，从而实现低延迟分割推理。
- **渐进式分割推理**：可以采用[240]中提出的渐进式分割推理机制来消除只要满足所需推理精度就可以不必要地传输中间标记表示。在[240]中，用户可以安排特征的卸载顺序，先将更重要的特征上传到边缘服务器。渐进式特征卸载将一直进行，直到边缘服务器确定上传的特征达到了进行推理的足够目标置信水平。通过在分割LLM推理中使用这种范式，可以减少最终用户和边缘服务器之间的通信

开销，从而节省通信资源。

- **带有早期退出的分割推理**：早期退出技术可以应用于分割推理，以减少边缘设备/服务器上的计算延迟。在这种情况下，当在用户端子LLMs中添加早期退出模块时，可以跳过用户（服务器）端子LLMs中后续层的推理计算，这意味着**不需要上传中间特征以减少延迟**。然而，在采用早期退出方法的LLM推理中，当token在早期退出层输出结果时，后续层的隐藏状态缺失。因此，在带有早期退出技术的分割LLM推理中，如果早期退出层位于用户端子LLMs中，仍然需要将隐藏状态上传到边缘服务器以进行后续层的KV缓存计算。因此，我们需要为早期退出层的隐藏状态上传策略进行设计，例如，根据信道条件的机会性上传隐藏状态。
- **其他分割推理的变体**：尽管分割推理通过在本地设备上保留原始数据来增强用户隐私，但双分割范式仍然允许边缘服务器获得推理结果，这可能是隐私敏感的。为了解决这个问题，可以采用U形或Λ形分割LLM推理范式。

1.

9.3. 协作推理

分割推理交换高维中间特征，导致过度的通信开销。为了克服这个限制，边缘设备和边缘服务器可以以其他模式进行合作，以更小的信息交换量。例如，推测性解码可以采用在LLMs的设备-服务器协作推理中。推测性解码使边缘设备能够运行一个较小的设备上LLM，称为近似模型，同时要求边缘服务器运行一个较大的LLM来验证和纠正边缘设备上传的输出标记。这种方法的主要优点有三个：首先，它使边缘设备能够生成初步结果/决策，这些结果/决策可以用于低延迟推理。其次，平行解码过程加速了边缘服务器上的推理过程。由于LLMs使用自回归解码技术生成标记，通过服务器端LLM单独串行解码，验证长序列的标记要比解码快得多。第三，关于通信开销，输出标记通常比许多情况下切割层中间特征要小。为了进一步节省通信-计算资源，边缘设备可以根据校准的置信分数决定是否将设备上LLM生成的标记上传到边缘服务器进行验证，因为高度自信的输出可能不需要边缘服务器消耗资源进行验证。

总结
边缘设备和服务器通常合作支持LLM推理，以解决隐私和延迟问题。分割或协作推理受到内存和通信-计算延迟等因素的影响。考虑到这些影响因素，边缘LLM推理的优化必须仔细研究，考虑各种LLM技术，如KV缓存优化、多模态特征提取和自回归模型。所有这些特性显著影响LLMs的内存使用、通信开销和推理延迟。因此，边缘LLM推理的设计，如模型分割、早期退出和资源分配，自然不同于传统的边缘推理系统，创造了这一领域丰富的优化挑战和机会。

10. 未来研究方向

10.1 绿色边缘LLM

能耗是公众对LLMs关注的主要问题。据估计，训练GPT-4的能耗相当于1000个美国普通家庭5到6年的能耗。此外，模型推理的能源成本可能更高，因为全球用户频繁的服务请求。因此，一个紧迫的研究问题是设计能源效率高的LLM训练和推理。**边缘LLM可以在三个方面减少LLMs的能耗。首先，在网络边缘提供模型微调或推理消除了将大量数据传输到云中心的需要，从而减少了主干网络中的能源成本。其次，集成通信-计算设计可以共同优化，以提高训练/推理的能源效率。例如，可以通过数据压缩或参数冻结来减少数据通信量，以降低总传输功率，只要达到所需的训练/推理精度即可。最后，边缘LLM可以利用小规模LLM获得初始推理结果，并在推理置信度低时才利用基于云的大规模LLM的力量。这可能减少了由于每次用户请求都调用大规模LLM而产生的云中心的能源消耗。**关于研究问题，绿色边缘LLM具有集成无线通信和计算的设计，必须考虑整体传输和计算能源。考虑到这一点，有两个主要的设计目标，**即减少边缘设备上的能源消耗和减少绿色AI的总体能源消耗。**第一个目标有利于电池受限的IoT和移动设备，使AI服务或训练对客户更加可访问。为了实现这个目标，只要节省能源，就可以将AI训练/推理卸载到边缘服务器。例如，采用PEFT和SL时，可以尽可能多地冻结客户端模型，甚至完全冻结，以最小化边缘设备上的能源消耗。尽管这种方法可能会导致学习收敛速度变慢或增加达到目标训练/推理精度的边缘服务

器的能源消耗，但这种成本可能不太令人关注，因为边缘服务器通常更强大且连接到电源。第二个目标是减少整体系统成本，特别是从移动运营商的角度来看。通过最小化总能源消耗或最大化整体能源效率，网络运营商可以在有限或更低的能源成本下提高以AI为中心的指标。一个有意义的指标可以是“人工智能能效”[257]，即在基本精度要求的约束下，每个能源成本实现的智能（AI精度）量。**在考虑能源消耗的同时优化AI可以消除系统仅使用大量能源来实现微小改进的情况，这从运营商和社会的角度来看都是不合理的。**

10.2 安全边缘LLM

安全边缘LLM是另一个重要的研究领域。虽然LLM安全已经得到了广泛的研究，但边缘LLM安全受到了较少的关注。具体来说，边缘LLM通常涉及LLM的分布式学习，其中出现了新的挑战。尽管联邦学习和分割学习，如前所述，是避免直接访问个人数据的隐私增强方法，但仍存在隐私风险，因为恶意服务器可能会发动攻击以**基于接收到的模型或中间特征恢复原始数据。**在移动边缘，应集成定制的安全组件以确保安全和强大的LLMs。让我们考虑安全边缘LLM的两个方面。第一个方面是防御推理攻击，以保护用户隐私从移动边缘服务器或其他边缘设备的泄露。已经证明，通过仅在训练数据集中插入一些看似良性的句子，可以提示LLMs可能泄露训练过程中其他用户的私人信息，如信用卡信息[260]。在FL和SL中可能也会观察到类似的问题，创造了丰富的研究问题。一种可能的防御机制是将噪声添加到高度敏感的个人数据中，如信用卡信息，基于**差分隐私理论。**此外，可以开发适当的机制来检测来自客户端的此类看似良性的标记/参数/特征。第二个方面是防御数据投毒或后门攻击，通过过滤出旨在改变训练过程的恶意用户，从而维护训练的有效性。这些攻击可能导致LLMs的严重后果，即在考虑医疗保健LLM时输出有害的健康指导。尽管这类攻击已经针对LLMs进行了研究，但仍然缺乏考虑分布式学习的研究工作，特别是FL和SL在边缘设备上的LLMs。考虑到PEFT，攻击者可能只能改变模型的一小部分，例如适配器或提示，以影响训练过程，为设计攻击/防御方案带来了新的挑战/机会。

11. 结论

近年来，语言模型在规模上呈指数级增长，催生了众多具有数十亿参数的LLMs。这一趋势促使我们思考边缘智能如何适应这些庞大的模型。在本文中，我们提倡了从云计算到6G MEI的LLM部署范式转变。我们强调了推动这一范式转变的关键应用，认为云计算很难满足延迟、带宽和隐私要求。与此同时，我们确定了主要源于网络边缘资源限制的挑战。为了应对这些挑战，我们首先提出了一个6G MEI架构用于LLMs，然后阐述了几种方法，以实现在资源受限的移动边缘上高效地进行边缘缓存和交付、边缘训练和边缘推理的LLMs。我们希望本文能激发更多的无线社区研究人员探索在移动边缘部署LLMs，并进一步推进这一新兴领域的发展。