arXiv:2601.03992v1 [cs.DC] 7 Jan 2026

面向边缘 GPU近存计算系统高效混合专家推理的调度框架

Qi Wu[1]，Chao Fang[1,†]、Jiayuan Chen[2]，Ye Lin[1],Yueqi Zhang[1],Yichuan Bai[1],Yuan Du[1],Li Du[1]南京大学电子科学与工程学院，中国 [2]中国移动研究院，中国 电子邮箱： {qiwu、fantasysee、yelin、zhangyueqi、yicbai}@smail.nju.edu.cn、chenjiayuan@chinamobile.com、 {yuandu、ldu}@nju.edu.cn

摘要—混合专家（MoE）模型通过将模型容量与激活计算解耦，促进了边缘部署，但其庞大的内存占用使得具备近数据处理（NDP）能力的 GPU 系统成为必要，这类系统可将专家卸载至专用处理单元。然而，在此类基于边缘的 GPU近存计算系统上部署混合专家模型面临三项关键挑战：1）由于专家选择和专家并行的不均衡，NDP 单元之间存在严重的负载不均衡；2）NDP 单元执行专家计算时 GPU 利用率不足；3）由于专家激活模式不可预测，预取需要进行大量数据预分析。为应对这些挑战，本文提出了一种高效的推理框架，包含三项关键优化。首先，利用混合专家推理中尚未得到充分探索的张量并行，将大型专家参数分区并分配至多个 NDP 单元，以便同时进行计算，从而适应边缘低批量场景。其次，负载均衡感知调度算法将专家计算分配至 NDP 单元和 GPU，以最大化资源利用率。第三，无需数据集的预取策略主动加载频繁访问的专家，以最小化激活延迟。实验结果表明，与最先进的方法相比，我们的框架使 GPU近存计算系统的端到端延迟平均加速 2.41× 倍，最高加速 2.56× 倍，显著提升了资源受限环境中的混合专家推理效率。

一、引言

对于基于 Transformer 的 [1] 大语言模型（LLM），混合专家（MoE）模型近期已成为一种高效的架构范式，可通过稀疏计算实现模型的大规模扩展。如图 1 所示，混合专家的核心机制是在前馈神经网络（FFN）阶段进行稀疏激活，即针对每个词元 [2],，仅启用一部分选定的参数，称为“专家”。

[3], 这使其有别于会激活所有参数的稠密模型 [4]–[7] 。然而，这种效率需要付出高昂的内存代价，因为专家参数通常从数百亿到数千亿不等 [8], ，远超消费级 GPU 的容量。例如，RTX 5080 是一款现代高端消费级 GPU，配备 16GB 显存 [9], ，不足以容纳许多大型混合专家模型，例如 Qwen3-30B-A3B [10] ，其参数占用空间可超过 60 GB。图 1 所示的内存容量差距，对预算有限的边缘大语言模型部署构成了关键挑战。

†通讯作者。本研究部分受中国国家重点研发计划项目（项目编号：2022YFB4400900）、江苏省战略性新兴产业和关键技术项目（项目编号：BE2023020-3）、江苏省基础研究计划（项目编号：BK20243042）以及南京大学—中国移动通信集团有限公司联合研究院资助。

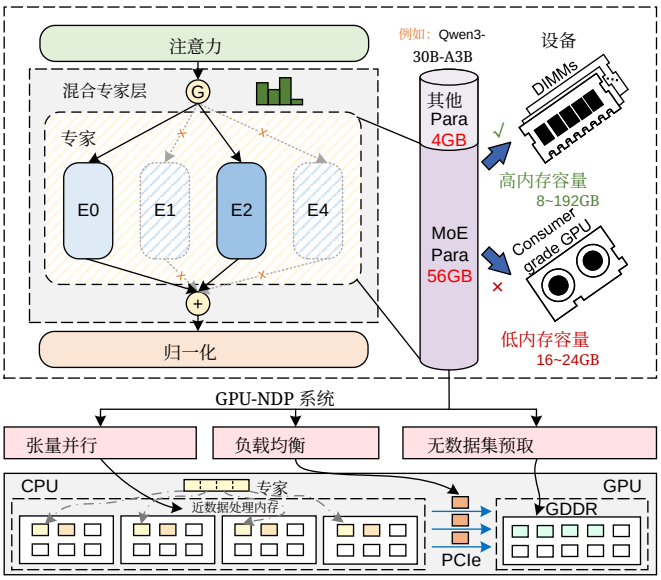


图 1. 消费级 GPU 显存与大型混合专家模型之间日益扩大的内存差距，使 GPU近存计算系统成为一种颇具前景且经济高效的边缘解决方案。我们的调度框架通过张量并行、负载均衡和无数据集预取，支持在此类系统上高效进行混合专家推理。

为解决内存容量这一挑战，当前研究主要沿着三条不同的技术路径展开。其中两条是基于CPU的解决方案。第一种方案利用主机系统内存，即CPU DRAM，作为扩展的GPU内存池，将非活跃专家参数卸载，并在需要时将其传回 [11]–[13]。第二种方案采用异构计算，将部分参数和计算任务交由CPU处理，以降低传输开销 [14]–[16]。尽管这些方法能够通过静态预取 [12] 或多批次策略 [15],降低传输延迟，但仍受限于CPU的内存带宽和计算能力。此外，静态预取需要利用校准数据进行大量预运行，以生成专家使用模式，因此难以适应边缘动态场景。

第三类研究通过将计算能力部署在数据存储附近，探索近数据处理（NDP）系统。 [17],[18]。这些系统可通过将计算能力集成到可插拔的双列直插式内存模块（DIMM）或改进型图形双倍数据速率（GDDR）中，并接入 GPU 内存系统，从而应对内存容量限制，大幅降低 PCIe 总线上的数据传输延迟。然而，当前的

基于 GDDR 的方法会产生高昂的实现成本 [18], ，因此不太适合成本效益高的边缘部署。更重要的是，现有 GPU-NDP 系统的调度策略存在若干局限。首先，多节点系统中的专家并行实现 [17],[18] 会导致单批次推理期间专家分布效率低下。此外，Duplex 等异构调度方案 [18]由于解码阶段各专家的算术强度始终一致，无法实现图形处理器与近数据处理设备之间的计算负载均衡；而 MoNDE [17], 虽然引入了负载均衡策略，却依赖基于历史配置文件的周期性调整，导致其在动态边缘工作负载下效率欠佳。此外，依赖基于校准的专家预取策略 [17] ，在输入分布与校准数据集不同的边缘推理场景中会变得不准确，从而降低效率。

为解决上述局限，本文提出了一种新颖的调度框架，用于加速边缘 GPU-NDP 系统上的混合专家推理，具体面向个人工作站等本地环境，如图 1 所示。该框架专为采用启用 NDP 的 DIMM 配置的边缘部署而设计。与 HBM-PIM 相比，这类配置成本效益更高，集成复杂度也更低，因此更受青睐。此外，我们采用张量并行，将专家参数划分到不同设备上。不同于依赖特定数据集的专家激活热图的现有方法，我们会在初始提示处理阶段（即 预填充）动态观测专家激活模式，并主动将预测会在后续令牌生成阶段（即 解码）激活的专家预取至图形处理器。此外，当图形处理器上没有已激活的专家时，我们的负载均衡机制仅将部分专家参数传回图形处理器，从而能够同时利用图形处理器和 NDP-DIMM 资源进行计算。我们的主要贡献总结如下。

1）提出了一种调度框架，旨在通过集成消费级图形处理器和经济高效的近数据处理双列直插式内存模块，实现 GPU-NDP 系统上的混合专家高效推理。该框架专为边缘部署中常见的单批次推理场景进行了优化。2）

提出了一种动态混合调度策略，通过独特地结合张量并行、负载均衡机制和无数据集专家预取，提升混合专家推理的效率。3）对四种热门混合专家模型进行了全面评估，结果显示，与最先进水平（SOTA）的 MoNDE [17] 相比，端到端延迟最高可加速 2.56× ，并证明了我们的方法在边缘MoE推理中的有效性。

II. 背景与相关工作

A. 边缘*MoE*部署需求

混合专家模型的本地部署源于对离线可用性、强化数据隐私以及无需云依赖的用户个性化需求 [19]–[22]。然而，现代混合专家模型庞大的参数规模

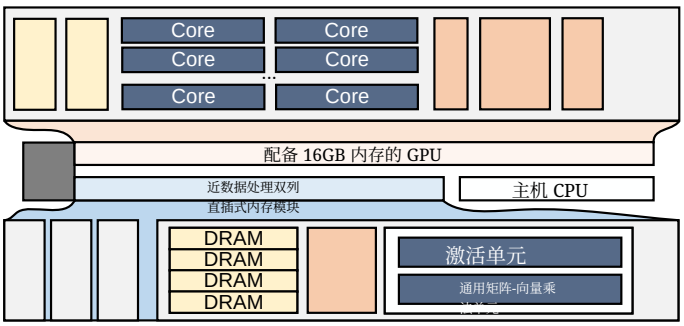


图 2. GPU-NDPDIMM 系统概览。

例如 Qwen3-30B-A3B [10]其规模超过 60GB，这在内存和 I/O 预算有限的情况下，严重阻碍了边缘部署。

实证表明，专家激活模式高度不均衡：只有少数专家会被频繁激活，而大多数专家在大部分词元处理中都处于闲置状态。以往系统 [12], [15] 利用这种稀疏性，通过专家预取和批间权重复用，在大批次间分摊加载开销，从而提高推理效率。然而，边缘场景通常是单用户交互，批大小为 1 [23], ，无法利用批间分摊的机会。此外，传统预取 [12]依赖离线分析校准数据集来确定专家使用模式，因此在动态边缘场景中的效率并不理想。校准数据与实际工作负载不匹配，往往会导致单批次推理过程中传输未使用的专家权重。因此，在低批次边缘推理中，处理器与内存之间的权重流量成为主要瓶颈。这促使我们探索近数据处理方法，将计算移至专家权重所在位置附近，而不是持续传输大型权重矩阵。

B. 面向混合专家模型加速的近数据处理

近数据处理方法 [17], [18], [24]–[36] 通过将计算置于存储阵列附近或内部，解决了数据移动瓶颈。在混合专家推理中，这种方法尤其有效，因为数据传输主要由专家权重而非激活值主导。例如，在单批次场景下，Mixtral-8x7B [37], 每层激活的专家权重为 672 MB，而激活值仅为 0.0078 MB。通过在权重所在位置执行专家计算，近数据处理仅需传输轻量级的激活值。

目前，近数据处理平台涵盖 HBM-PIM 方案，以及基于双列直插式内存模块或低功耗双倍数据速率内存设备的商用模块设计。HBM-PIM 方法在适用的工作负载上表现出色 [26], [27],，但它依赖专用且昂贵的堆叠式内存，因此难以在成本敏感、采用本地部署的系统中推广。相比之下，基于双列直插式内存模块/低功耗双倍数据速率内存的近数据处理利用标准模块，为边缘部署提供了更经济、可扩展的途径 [17], [30]。然而，现有商用方案在边缘MoE推理方面存在显著局限。LP-Spec [30] 采用低功耗双倍数据速率内存近数据处理与神经网络处理器，通过动态调度协调注意力和前馈神经网络的计算，从而加速稠密模型。然而，其调度

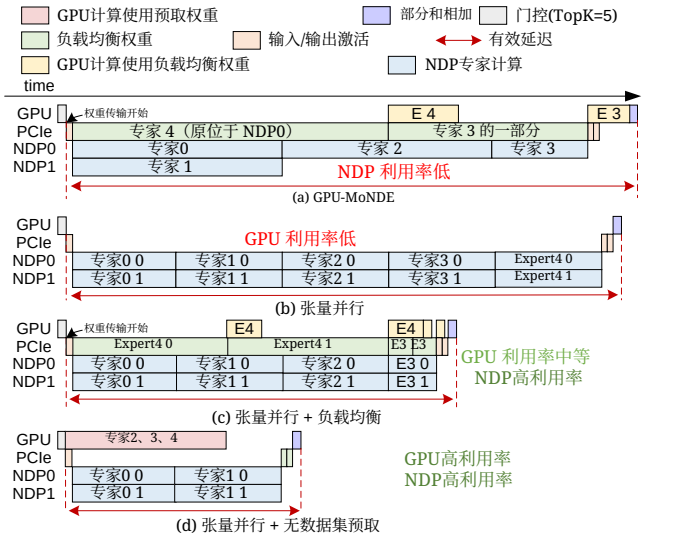


图 3. 解码阶段混合专家工作流调度的比较。所提出的调度框架基于 (b) 张量并行，并辅以 (c) 负载均衡和 (d) 无数据集预取。

仅在令牌剪枝期间触发，并未涵盖混合专家模型。相比之下，MoNDE [17],专门针对混合专家模型，将专家计算卸载到LPDDR近存处理单元，同时将其他计算保留在图形处理器上。它主要考虑单个大内存近数据处理设备，在单批次下扩展到多近存处理器配置时表现不佳，因为设备间工作负载不均衡会显著降低利用率。这些局限凸显了针对多近数据处理器边缘部署场景制定更有效调度策略的必要性，也促使我们提出本框架。

III. 所提出的框架

A. 框架概述

图 2 展示了所提出系统框架的概述。该框架基于一块消费级图形处理器和多个 NDP-DIMM。我们的框架采用基于中央缓冲区的 NDP-DIMM 架构 [18], [24], [28], [38],，其中每个双列直插式内存模块上集成的近存计算核心负责处理本地存储的数据。在这一图形处理器—近存计算双列直插式内存模块系统中，计算工作负载分布在不同的处理单元上。前馈神经网络层中占主导地位的混合专家计算由基于双列直插式内存模块的近存计算单元执行，而注意力机制等其余所有组件则由图形处理器处理。主机 CPU 专用于任务调度与协调。尽管我们的分析基于这一特定硬件配置，但该框架也广泛适用于其他 GPU近存计算系统。为优化混合专家推理，我们提出了三项协同创新：（1）一种张量并行策略，可将专家权重高效分布至各近存计算单元；（2）一种负载均衡调度策略，可根据令牌路由模式进行调整；（3）一种无需数据集的预取策略，无需事先进行数据分析。图 3 展示了这些策略及其相较于先进的 MoNDE [17]所带来的渐进式收益。

B. 张量并行

MoNDE [17]采用的传统专家并行会将完整专家分配到不同的 NDP 上。然而，动态

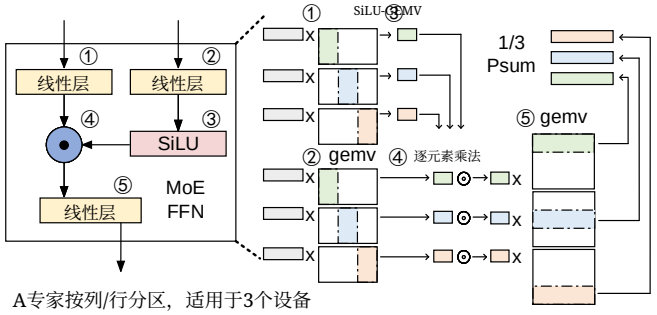


图 4. 采用我们提出的张量并行方法时，混合专家层的计算过程。左侧为混合专家前馈神经网络的常见结构，右侧为将一个专家部署到三个计算设备上的示意图。

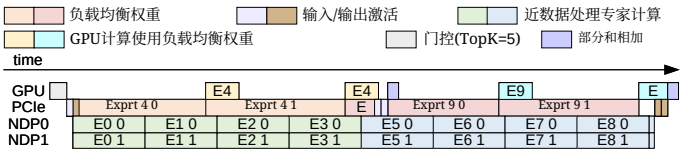


图 5. 我们框架中预填充阶段的混合专家工作流调度。

混合专家模型固有的路由机制无法保证各 NDP 上的专家激活均匀分布，导致资源利用率不理想，如图 3a 所示。在这种情况下，NDP0 部署了四个激活专家，而 NDP1 上只有一个，因此 NDP0 成为关键路径。尽管部分工作负载已卸载到图形处理器，关键路径仍然很长。

为缓解这一低效问题，我们提出在混合专家计算阶段跨 NDP 进行张量并行。由于普遍认为张量并行会带来较高的通信开销，这种策略在以往的混合专家调度研究中基本被忽视。然而，在单批次多NDP设置下，专家并行造成的负载不均衡超过了张量并行的通信成本。如图 3b 所示，张量并行将每个专家的计算均匀划分到所有可用的 NDP 上。因此，无论激活哪个专家，每个 NDP 都会分担一部分计算。图 4 以三个 NDP 设备为例，展示了我们在混合专家层计算中采用的张量并行策略。我们采用两阶段划分方法。前两个线性层中的通用矩阵-向量乘法运算按列划分，而最后一个线性层中的运算则按行划分。这种方法能够在各设备间高效分配计算工作负载和内存需求，从而生成最终的部分和。不过，尽管张量并行在图 3b 中实现了 NDP 的高利用率，但解码阶段的图形处理器仍处于空闲状态，因而还有进一步优化的空间。这一观察表明，有必要在图形处理器与 NDP 之间实现负载均衡，以充分利用双方的计算资源。

C. 负载均衡

为利用近数据处理计算期间闲置的 GPU 资源，我们提出了图 3c 和图 5 所示的负载均衡策略。现有策略（如 MoNDE）[17] 存在两项局限：它们忽视了工作负载分布不均衡的问题

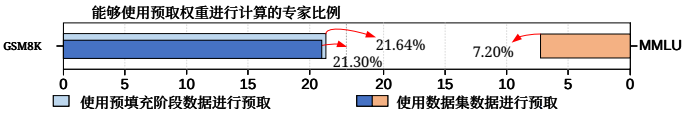


图 6. Qwen3-30B-A3B [10]中预取权重的利用率：比较基于完整校准数据集（GSM8K、MMLU）的分析结果与基于无数据集预填充阶段数据的分析结果。比较使用了从 GSM8K 中随机抽取的 20 个输入。

在激活专家之间，并通过仅令 NDP 计算时间等于权重传输时间来过度简化负载均衡条件，忽略了预填充阶段和解码阶段输入词元数量的差异。此外，他们使用基于历史数据的缩放因子来修正负载均衡方程。

我们基于张量并行重新制定负载均衡条件。关键在于，E_g 值（表示分配给图形处理器的专家数量）应经过确定，使权重传输与图形处理器计算所花费的总时间，与 NDP 计算及输入/输出激活数据传输所花费的总时间相平衡，而不是像 MoNDE [17]那样通过定期缩放因子进行调整。我们还将 T_g 定义为图形处理器专家计算延迟，将 T_n 定义为张量并行后的 NDP-DIMM 专家计算延迟，将T_w 定义为专家传输时间，将 T_a 定义为激活传输时间。 N 表示启用的 NDP-DIMM 数量，S 表示序列长度。由于最后一次传输无法隐藏，因此实际使用的是 $S-1$ 。工作负载参数包括 TopK（每层激活专家的数量），以及 E_n （NDP-DIMM 上的专家数量），其中$E_n + E_g =$ TopK。请注意，图形处理器上的大多数计算都与权重传输重叠，因此， $E_{g'}$ 表示未隐藏的计算时间。为便于分析，将 $E_{g'}$ 定义为 E_g小数部分的 N 分之一，以反映未与 N 路权重传输重叠的图形处理器专家计算所占比例；当 E_g 是 N的整数倍时，则使用 $1/N$ 作为标称值。

解码阶段的负载均衡条件：

$$T_w \cdot E_g + T_g \cdot E_{g'} = (N+1) \cdot T_a + T_n \cdot E_n. \quad (1)$$

预填充阶段的负载均衡条件：

$$T_w \cdot E_g + T_g \cdot E_{g'} + (S-1) \cdot T_a \cdot N = (N+1) \cdot T_a + T_n \cdot E_n. \quad (2)$$

由于解码阶段和预填充阶段具有不同的计算特性，本文提出的负载均衡条件采用不同的形式。公式 (1) 定义了解码阶段的负载均衡条件，而公式 (2) 则考虑了预填充阶段中对序列长度的依赖。利用这些公式，可以计算出负载均衡后的专家 E_g 。如图 3c 所示，负载均衡还可进一步提升混合专家阶段的加速效果。尽管这种方法优化了专家分配，但动态负载均衡所需的频繁权重传输可能受 PCIe 带宽限制，因此我们进一步优化运行时传输开销。

D. 无数据集预取

在需要之前将专家权重预取到 GPU 内存中，可以显著降低 PCIe 传输延迟并

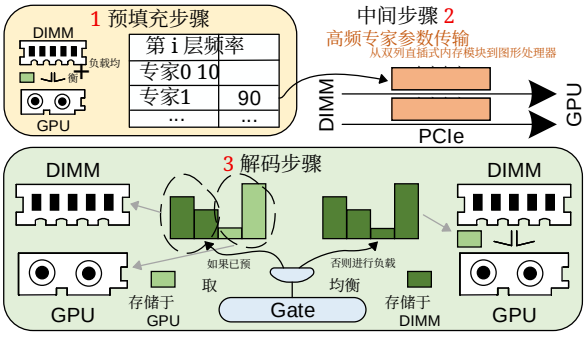


图 7. 无数据集预取策略。该策略分为三个步骤：1 预填充步骤、2 中间步骤和 3 解码步骤。

提高图形处理器利用率。然而，传统预取策略[12],通常依赖于对校准数据集进行预分析，以确定专家的激活模式，这会额外产生数百至数千个推理轮次，并导致预取权重的利用率因数据集而异，如图 6 所示。尽管 [14] 提到活跃专家在预填充和解码阶段具有相似性，但该方法仍使用校准数据集进行初始化，并引入了精度误差。如图 7 所示，我们的策略通过三步流水线消除了这一局限，无需校准数据，便可在实际推理执行过程中动态学习专家激活模式。该流水线分为三个步骤：1预填充、2中间阶段和3解码。其核心创新如图 6 所示，利用预填充阶段观察到的激活模式来指导解码阶段的预取决策，将成本高昂的离线分析转变为轻量级运行时学习，使其能够适应实际工作负载特征。

1预填充步骤：动态信息收集。如图 71 所示，在预填充步骤中，系统处理输入提示的同时，实时收集专家激活统计数据。执行前馈网络计算时，系统采用公式 (2) 中的负载均衡策略，持续监测哪些专家模块被激活，并追踪其频率模式。这些动态收集的激活数据构成即时的任务专属专家表，无需离线校准数据集，并为后续步骤中的预取决策提供直接依据。如图 6 所示，其预取专家利用率甚至高于使用校准数据集时的利用率。

2中间步骤：权重预取。根据预填充步骤中识别出的激活模式，这一中间步骤会有选择地将激活频率最高的专家预取到GPU 内存中，如图72所示。要预取的专家数量取决于可用的GPU 内存，以确保资源利用率达到最佳：在每一层中预取 x 个专家， x 从1开始初始化，并递增，直到GPU无法再容纳更多专家。由于专家预取所需的数据传输量相对于整体推理工作负载而言很小，因此开销极低。同时，其余关键专家也会预先准备好，以便在解码步骤中立即访问。

3解码步骤：自适应计算执行。如图所示——

表 I GPU-NDP 系统配置

GPU配置		NDP-DIMM配置	
频率	2.30 GHz	DDR4类型	3200 MT/s，32GB
GDDR	16GB GDDR7	秩	4 / 双列直插式内存模块
流式多处理器数量	84	存储体组	8 / 秩
张量核心	336	存储体	4 / 存储体组
接口	PCIe 5.0	乘法器	64 / 双列直插式内存模块

表 II 基于混合专家的模型的工作负载特征

模型	E 参数	#E	TopK	隐藏层	中间层	#层
DeepSeek-MoE [41]	15.4B	64	6/2△	2048	1408	27/28*
Qwen3-30B-A3B [10]	29.0B	128	8	2048	768	48
Phi-3.5-MoE [42]	40.3B	16	2	6400	4096	32
Mixtral-8x7B [37]	42.0B	8	2	4096	14336	32

△6/2 表示激活了 6 个路由专家和 2 个共享专家。*27/28 表示 28 层中有 27 层是混合专家层。

图 73 所示，该步骤的执行方式如下：如果激活专家已预取至图形处理器，则在图形处理器上进行计算；否则，未预取的专家将在近数据处理器中执行，如图 3d 所示。然而，尽可能增加在图形处理器上计算的专家数量，并不一定能获得最佳性能。必须满足均衡约束，即图形处理器计算时间应等于近数据处理计算时间与传输时间之和。根据公式（3），可以估算 E_{max}，即能够在图形处理器上计算的专家数量上限。此外，当所有激活专家均未预取时，该框架会无缝切换至标准负载均衡策略来处理计算请求，如图 3c 所示。这种容错备份设计可确保图形处理器和近数据处理双列直插式内存模块的利用率最大化。

可调度至图形处理器的专家参数最大值：

$$T_g \cdot E_{max} = (\text{TopK} - E_{max}) \cdot T_n + (1 + N) \cdot T_a. \quad (3)$$

IV. 实验结果

A. 实验设置

1) 系统： 为评估我们提出的调度框架，我们采用了图 2 所示的图形处理器-近数据处理系统架构。该系统由一块消费级 NVIDIA RTX 5080 图形处理器组成，配备 16GB GDDR显存和 PCIe 5.0 x16，并搭载 Intel i7-14700 处理器，可提供最高 192GB 系统内存，最大带宽为 89.6 GB/s。GPU-NDP 系统最多支持六个 32GB 近数据处理双列直插式内存模块，每个模块的内部带宽为 102.4GB/s，详情见表 I。为评估性能，我们使用 AttAcc! [26] 模拟图形处理器的性能特征，并采用修改版的 Ramulator 2.0 [40] 评估具备计算能力的双列直插式内存模块设备的性能；该模拟工具在图形处理器方面的准确性已得到验证
以及 [40]中的存内处理操作。

2) 模型： 我们从 HuggingFace 仓库中选取了 4 种具有代表性的混合专家模型进行评估：DeepSeek-MoE [41],Qwen3-30B-A3B [10], Phi-3.5-MoE [42],Mixtral-8x7B [37]。这些模型具有多样的架构特征，

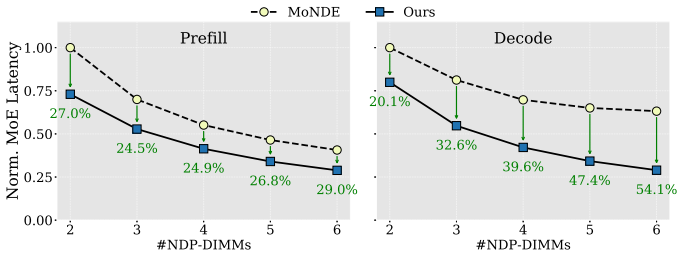


图 8. 预填充和解码阶段的混合专家延迟相对于最先进水平 MoNDE [17]的归一化结果。

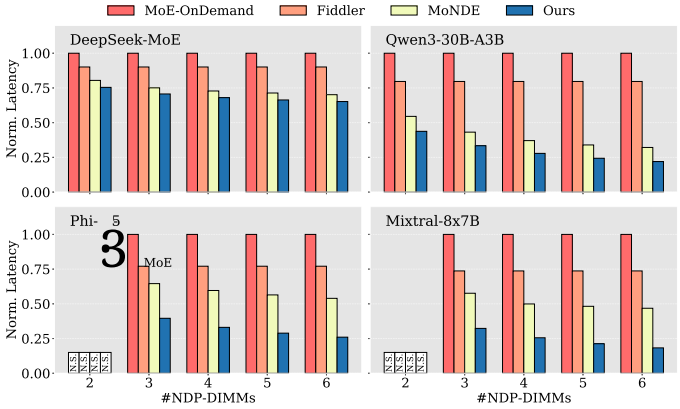


图 9. 不同数量 NDP-DIMM 下的端到端模型性能，相对于 MoE-OnDemand [11]进行归一化。不支持（N.S.）表示该配置无法容纳所有混合专家参数。

如表 II 所示。每层的激活专家数量范围为 2 至 8，每层的专家总数则为 8 至 128。值得注意的是，DeepSeek-MoE [41]包含由所有词元共用的共享专家，这一点使其有别于其他三种模型。鉴于我们关注个人设备的本地部署场景，所有实验均采用批大小为 1。为确保评估条件一致，所有实验的输入和输出序列长度均统一设为 512 个词元。

3) 基线方法： 为全面比较性能，我们评估了 3 种基线方法。1) MoE-OnDemand [11]：一种 GPU 按需专家权重传输系统，其中混合专家参数被卸载到系统内存，激活专家在运行时动态加载到图形处理器中。2) Fiddler [16]：一种最先进水平的 GPU-CPU 异构系统，它将混合专家参数卸载到内存，但在 CPU 而非图形处理器上执行专家计算。3) MoNDE [17]：一种最先进水平的近数据处理系统，使用专用处理单元存储并计算混合专家组件，通过传输激活值而非模型权重来减少数据搬移开销。其部署环境与本文研究的边缘中心场景一致。

B. 独立预填充和解码性能评估

我们通过将混合专家预填充和解码性能与 MoNDE [17]进行对比，分析了二者的性能。图 8 以 Qwen3-30B-A3B 为例，展示了 MoNDE 与我们方法之间的专家计算延迟。结果表明，性能得到了显著提升：在预填充阶段和解码

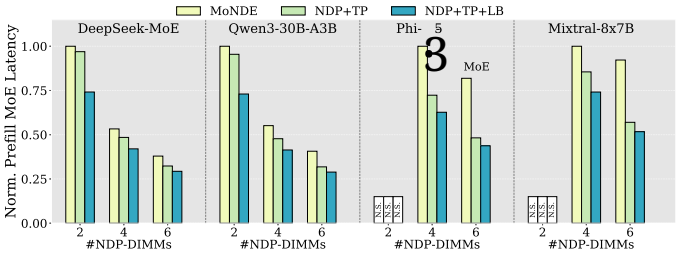


图 1. 张量并行和负载均衡对预填充阶段混合专家计算的影响。N.S. 表示“不支持”，即无法容纳所有混合专家参数。

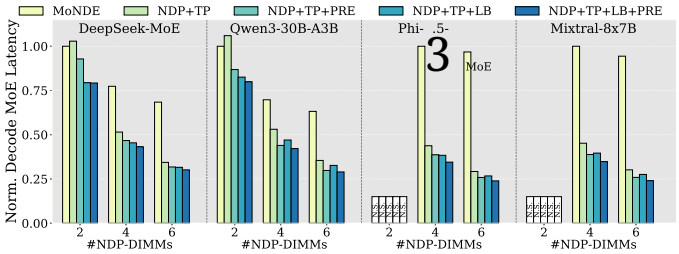


图 11. 张量并行、负载均衡和无数据集预取对解码阶段混合专家计算的影响。N.S. 表示“不支持”，即无法容纳全部混合专家参数。

阶段的平均加速比分别达到 1.{v2} 和 1.{v3}。这一性能提升源于我们更加高效的混合调度策略。MoNDE 根据算术强度循环分配专家工作负载，但无法在所有 NDP 单元上实现最佳利用率。相比之下，我们的混合方法能够在不同场景下实现更加均衡的资源利用率。进一步考察两个阶段可以发现，与预填充阶段相比，解码阶段的性能提升尤为明显。在预填充阶段，尽管 MoNDE 方案下的并发多令牌处理能够使激活专家相对均匀地分布在各个 NDP 单元上，但我们的框架通过让 NDP 更加均衡地参与，仍然能够取得更高的加速比。在解码阶段，由于词元是串行生成的，MoNDE 方案中的 NDP 之间存在显著的负载不均衡。相比之下，我们的方法使所有 NDP 承担相同的负载，从而实现更高的效率。

C. 端到端性能评估

随后，我们评估了所提出框架在不同数量的近数据处理双列直插式内存模块下的端到端性能。图 9 展示了四种混合专家模型的端到端延迟结果，并以 MoE-OnDemand 的性能为基准进行了归一化。我们的框架在所有基线方法上均显著提升了性能。具体而言，与参数卸载方法 MoE-OnDemand [11] 和 Fiddler [16],相比，我们实现的最高加速比分别为 5.49× 和 4.05×；与近数据处理基线 MoNDE [17]相比，加速比为 2.56× 。由于我们的加速主要针对混合专家计算，混合专家层中的参数越多，框架带来的加速效果就越显著。我们的方法之所以有效，是因为整合了张量并行、负载均衡和预取技术，这些技术共同实现了近数据处理双列直插式内存模块间高效的专家分布，同时优化了近数据处理利用率和图形处理器资源分配。

D. 我们所提技术的消融研究

我们开展了消融研究，以量化三项关键技术——张量并行、负载均衡和预取——各自及联合贡献。

我们将这些技术的不同组合与 NDP 基线进行比较，即 MoNDE [17]：仅采用张量并行（NDP+TP）、采用负载均衡的张量并行（NDP+TP+LB）、采用预取的张量并行（NDP+TP+PRE），以及完整的组合（NDP+TP+LB+PRE）。对于所有消融配置，我们都会在每个评估模型的预填充阶段结束后，将每层最有可能用到的 TopK 个专家预取到 GPU 内存中。然而，受 GPU 内存限制，Mixtral-8x7B [37]每层仅能预取一个专家。

如图 10 所示，在预填充混合专家阶段，NDP+TP 和 NDP+TP+LB 策略相较于 MoNDE 分别实现了平均 1.26× 和 1.46× 的加速比。由于预取的专家仅适用于解码混合专家阶段，因此图 10 中未展示与 PRE 相关的策略。在解码混合专家阶段，如图 11 所示，仅采用 NDP+TP 即可实现平均 1.99× 的加速比；与 MoNDE 相比，采用 NDP+TP+LD 和 NDP+TP+PRE 时，平均加速比分别达到 2.19× 和 2.23× 。三种策略的组合（NDP+TP+LB+PRE）在所有评估模型上始终实现最高的平均加速比 2.41×。

如图 11 所示，从解码混合专家阶段的结果中可以观察到两个显著现象。首先，使用 2 个近数据处理双列直插式内存模块时，对于 DeepSeek-MoE 和 QWen3-30B-A3B 模型，NDP+TP 方法的表现略逊于 MoNDE。近数据处理单元的计算能力远低于图形处理器，因此近数据处理资源有限时，张量并行的效果较差；而 MoNDE 能够利用图形处理器与近数据处理的协同作用。其次，与单独使用张量并行相比，混合策略的优势会随着近数据处理双列直插式内存模块数量的增加而减小。随着近数据处理双列直插式内存模块增多，专家计算速度大幅提升，参数传输和图形处理器计算的时间窗口随之缩短，从而降低了负载均衡与预取优化带来的相对收益。

五、结论

本文提出了一种调度框架，利用GPU近存计算系统加速边缘部署场景中的混合专家模型推理。该框架利用张量并行将专家参数划分到多个近数据处理单元上，采用负载均衡感知调度算法优化近数据处理单元和图形处理器的资源利用率，并通过无数据集预取策略主动加载频繁访问的专家。实验结果表明，与最先进的基于近数据处理的MoNDE方法相比，这些协同创新使我们的框架平均降低2.41× 、最高降低2.56× 的端到端延迟，显著提升了资源受限边缘环境中的混合专家推理效率。

参考文献

[1] A. Vaswani 等，“注意力就是你所需要的一切”，《神经信息处理系统进展》（NeurIPS），第30卷，2017年。[2] R. A. Jacobs 等，“自适应局部专家混合”，《神经计算》，第3卷，第1期，第79–87页，1991年。[3] N. Shazeer 等，“规模惊人的神经网络：稀疏门控混合专家层”，载于《国际学习表征会议》（ICLR），2017年。[4] J. Achiam 等，“GPT-4技术报告”，arXiv预印本 arXiv:2303.08774，2023年。[5] H. Touvron 等，“Llama：开放且高效的基础语言模型”，arXiv预印本 arXiv:2302.13971，2023年。[6] S. Zhang 等，“OPT：开放式预训练Transformer语言模型”，arXiv预印本 arXiv:2205.01068，2022年。[7] C. Fang 等，“Anda：利用可变长度分组激活数据格式实现高效大语言模型推理”，载于2025年IEEE高性能计算机体系结构国际研讨会（HPCA）。IEEE，2025年，第1467–1481页。[8] W. Cai 等，“大语言模型中的专家混合综述”，《IEEE知识与数据工程汇刊》（TKDE），2025年。[9] NVIDIA公司。（2025年）GeForce RTX 5080。 [在线]。网址：
h
t
t
p
s://
www.nvidia.com/zh-cn/geforce/graphics-cards/50-series/rtx-5080/
[10] A. Yang 等，“Qwen3技术报告”，arXiv预印本 arXiv:2505.09388，2025年。[11] R. Y. Aminabadi 等，“DeepSpeed-Inference：在前所未有的规模下实现Transformer模型的高效推理”，载于《高性能计算、网络、存储与分析国际会议》（SC），2022年，第1–15页。[12] Z. Fang 等，“Klotski：通过专家感知的多批次流水线实现高效混合专家推理”，载于第30届ACM编程语言与操作系统体系结构支持国际会议（ASPLOS）论文集，第2卷，2025年，第574–588页。[13] A. Eliseev 等，“通过卸载实现专家混合语言模型的快速推理”，arXiv预印本 arXiv:2312.17238，2023年。[14] Y. Zhang 等，“DAOP：面向高效混合专家推理的数据感知卸载与预测性预计算”，载于《欧洲设计、自动化与测试会议》（DATE），2025年，第1–7页。[15] S. Cao 等，“Moe-lightning：在内存受限GPU上实现高吞吐量混合专家推理”，载于第30届ACM编程语言与操作系统体系结构支持国际会议（ASPLOS）论文集，第1卷，2025年，第715–730页。[16] K. Kamahori 等，“Fiddler：用于混合专家模型快速推理的CPU-GPU协同调度”，载于《国际学习表征会议》（ICLR），2025年。[17] T. Kim 等，“MoNDE：面向大规模稀疏模型的近数据专家混合”，载于第61届ACM/IEEE设计自动化会议（DAC）论文集，2024年，第1–6页。[18] S. Yun 等，“Duplex：一种适用于混合专家、分组查询注意力和连续批处理大语言模型的设备”，载于第57届IEEE/ACM国际微体系结构研讨会（MICRO），2024年，第1429–1443页。[19] Y. Ji 等，“协同设计二值化Transformer与硬件加速器，以实现高效的端到端边缘部署”，载于第43届IEEE/ACM计算机辅助设计国际会议（ICCAD）论文集，2024年，第1–9页。[20] S. Hadish 等，“边缘语言模型：技术、挑战与应用综述”，载于2024年第二届基础模型与大语言模型国际会议（FLLM）。IEEE，2024年，第262–271页。[21] L. Huang 等，“一种具有设备端学习能力、适用于极端边缘场景的精度可扩展RISC-V深度神经网络处理器”，载于第29届亚洲及南太平洋设计自动化会议（ASP-DAC），2024年，第927–932页。[22] F. Liu 等，“Spark：通过高效编码实现神经网络的可扩展精度感知加速”，载于2024年IEEE国际高性能计算机体系结构研讨会（HPCA）。IEEE，2024年，第1029–1042页。[23] T. Cai 等，“Medusa：采用多个解码头的简易大语言模型推理加速框架”，载于《国际机器学习会议》（ICML），2024年，第5209–5235页。[24] L. Ke 等，“RecNMP：利用近存处理加速个性化推荐”，载于ACM/IEEE第47届年度国际计算机体系结构研讨会（ISCA）。IEEE，2020年，第790–803页。[25] Y. Kwon 等，“Tensordimm：一种实用的近存处理架构，用于深度学习中的嵌入和张量运算”，载于第 52 届 IEEE/ACM 国际微体系结构研讨会（MICRO）论文集，2019 年，第 740–753 页。[26] J. Park 等，“AttAcc!：释放近存处理的潜力，实现批量基于 Transformer 的生成模型推理”，载于第 29 届 ACM 程序设计语言与操作系统架构支持国际会议（ASPLOS）论文集，第 2 卷，2024 年，第 103–119 页。[27] L. Wu 等，“PIMoE：通过感知节流的任务卸载，推动 NPU-PIM 系统上的高效混合专家 Transformer 部署”，载于第 62 届 ACM/IEEE设计自动化会议（DAC）论文集，2025 年。[28] L. Liu 等，“让每个人都能负担大语言模型推理：利用近数据处理双列直插式内存模块扩展图形处理器内存”，载于 IEEE 高性能计算机体系结构国际研讨会（HPCA），2025 年，第 1751–1765 页。[29] F. Devaux，“真正的存内处理加速器”，载于 2019 年 IEEE Hot Chips 第 31 届研讨会（HCS）。IEEE 计算机学会，2019 年，第 1–24 页。[30] S. He 等，“LP-Spec：通过架构与数据流协同优化，利用低功耗双倍数据速率内存近存处理实现高效的大语言模型移动端推测推理”，载于第 44 届 IEEE/ACM 计算机辅助设计国际会议（ICCAD）论文集，2025 年。[31] Q. Jiang 等，“Ndpage：通过定制页表实现近数据处理架构的高效地址转换”，载于 2025 年欧洲设计、自动化与测试会议（DATE），2025 年，第 1–7 页。[32] S. Liang 等，“Hyqa：面向基于嵌入的问答系统的混合近数据处理平台”，载于 2024 年欧洲设计、自动化与测试会议暨展览会（DATE），2024 年，第 1–6 页。[33] C. Zhang 等，“近存并行索引与合并：实现高效的稀疏矩阵向量乘法间接访问”，载于 2024 年欧洲设计、自动化与测试会议暨展览会（DATE），2024 年，第 1–6 页。[34] Y. Zhao 等，“UM-PIM：具有统一和共享内存空间的基于动态随机存取存储器的近存处理”，载于 ACM/IEEE 第 51 届计算机体系结构国际研讨会（ISCA），2024 年，第 644–659 页。[35] Y. Chen 等，“BRAMAC：用于现场可编程门阵列上乘加运算的 BRAM 内计算架构”，载于 IEEE 第 31 届现场可编程定制计算机国际研讨会（FCCM），2023 年，第 52–62 页。[36] D. Kim 等，“面向人工智能和机器学习的存内处理电路概述”，《IEEE 电路与系统新兴及精选主题期刊》（JETCAS），第 12 卷，第 2 期，第 338–353 页，2022 年。[37] A. Q. Jiang 等，“Mixtral 专家模型”，arXiv 预印本 arXiv:2401 04088，2024 年。[38] J. Cong 等，“AIM：通过可扩展且非侵入式的加速器介入式内存加速计算基因组学”，载于国际存储系统研讨会（MEMSYS）论文集，2017 年，第 3–14 页。[39] 英特尔公司。（2025）intel-core-i7-14700。[在线]。网址：https://www.intel.cn/content/www/cn/zh/products/sku/236781/intel-core-i7-processor-14700-33m-cache-up-to-5-40-ghz/specifications.html [40] H. Luo 等，“Ramulator 2.0：一种现代化、模块化且可扩展的动态随机存取存储器模拟器”，《IEEE 计算机体系结构快报》（RAL），第 23 卷，第 1 期，第 112–116 页，2023 年。[41] D. Dai 等，“DeepSeekMoE：迈向专家混合语言模型中的终极专家专门化”，载于计算语言学协会第 62 届年会（ACL）论文集，2024 年，第 1280–1297 页。[42] E. Haider 等，“Phi-3 安全后训练：通过‘修复—验证’循环使语言模型保持一致”，arXiv 预印本 arXiv:2407.13833，2024 年。