混合专家（MoE）架构中的负载均衡机制综合分析报告

第一节 MoE模型中专家负载失衡现象

混合专家（Mixture-of-Experts, MoE）架构作为一种稀疏激活模型，其核心设计理念是通过“分而治之”的策略，在不显著增加计算成本的前提下，大幅扩展模型的参数容量[1]。然而，这一架构的优势与其固有的挑战紧密相连。本章节旨在深入剖析MoE模型中普遍存在的专家负载失衡问题，阐述其现象、根源及其对模型设计理念的根本性挑战。

1.1 MoE的架构愿景：超越计算的规模化扩展

传统的大型语言模型（LLM）通常采用“密集”（dense）架构，即在每次前向传播过程中，模型的所有参数都会被激活和使用。这种模式导致模型的计算成本与其参数量成正比，极大地限制了模型规模的进一步扩展。为了突破这一瓶颈，研究人员提出了稀疏门控混合专家（Sparsely-Gated MoE）架构[5]。

MoE架构的核心思想是在Transformer模型的某些层中，用一个MoE层替换标准的逐点前馈网络（Feed-Forward Network, FFN）层[2]。一个MoE层主要由两部分构成：一组“专家”网络（通常是多个独立的FFN）和一个“门控网络”（Gating Network）或称为“路由器”（Router）[5]。对于输入序列中的每一个令牌（token），门控网络会计算一个分数分布，以决定将该令牌路由到哪些专家进行处理[9]。通过仅选择分数最高的少数几个专家（通常是Top-K，其中K值很小，如1或2）来处理令牌，模型实现了所谓的“条件计算”（conditional computation）[4]。

这种稀疏激活模式是MoE效率的关键所在。它允许模型的总参数量（所有专家参数之和）达到数千亿甚至万亿级别，而单个令牌处理的实际计算量仅与被激活的少数几个专家相关，从而在保持计算成本相对恒定的同时，极大地提升了模型的容量和性能[1]。

1.2 专家利用率的现实：饥饿与集中的专业知识

尽管MoE架构在理论上前景广阔，但在实践中，一个长期存在的关键问题是专家利用率的严重不均衡。这直接回答了研究者们的核心疑虑：在拥有众多专家的MoE模型中，确实会出现某些专家被严重低估，甚至在处理数万个令牌后也难得被激活一次的情况[4]。这并非罕见的意外，而是一个被广泛记录和研究的典型失败模式[12]。

负载失衡的证据在各种研究中都有体现。早期的研究发现，即使在像MNIST这样相对简单的数据集上，原始的MoE训练方法也无法保证所有专家都得到有效利用[11]。更为现代和复杂的模型也面临同样的问题。例如，对DeepSeekMoE模型的深入分析揭示，尽管该模型每层拥有64个专家，且每次计算会激活6个，但模型实际上主要依赖于极少数几个专家。在不同的专业领域中，仅有少数几个专家处理了超过50%的路由决策[17]。这种现象被称为“热专家”（hot experts）问题，即一小部分专家持续接收大量令牌，而其他专家则持续处于“饥饿”状态，接收到的令牌寥寥无几，甚至为零[6]。

这种失衡的极端情况被称为“路由坍塌”（routing collapse）。该术语描述了门控网络在训练过程中收敛到一种退化状态，即对于几乎所有的输入，它都倾向于选择一个固定且极小的专家子集[20]。

MoE架构最大的优势——条件计算，恰恰也是其最大弱点的根源。这一内在矛盾的形成逻辑如下：首先，MoE的目标是为每个令牌激活“最合适”的专家。其次，“最合适”是由门控网络计算出的分数决定的。第三，在训练初期，由于随机初始化或数据分布的细微差异，某些专家可能对特定类型的输入表现出微弱的优势，从而获得稍高的分数。第四，诸如Top-K之类的选择机制本质上是离散的、赢家通吃的。这意味着，那个表现稍好的专家将获得针对该令牌的全部训练信号（梯度更新），而表现稍差的专家则一无所获。最后，这个过程形成了一个正反馈循环，微小的初始差异被迅速放大，最终导致少数专家变得越来越“受欢迎”，而大多数专家则被边缘化，陷入“饥饿”状态。因此，正是提供稀疏性的核心机制，创造了导致负载失衡的条件。

1.3 “富者愈富”的动态：失衡的根本原因

专家负载失衡的背后，是一种被称为“富者愈富”（the rich-get-richer）的自增强反馈循环动力学[25]。这个过程是导致失衡的主要机制。在训练的早期阶段，如果一小部分专家由于随机初始化或早期接触到的数据批次，对某些输入表现出哪怕是微不足道的优势，门控网络就会更倾向于将相关的令牌路由给它们[4]。

这些被频繁选中的专家因此获得了更多的梯度更新，从而更快地学习和优化。随着它们性能的提升，门控网络选择它们的概率会进一步增加，形成一个恶性循环。与此同时，那些在

初期未能获得青睐的专家，由于缺乏训练数据和梯度信号，其性能停滞不前，甚至相对下降，从而更不可能被门控网络选中。这种现象也被描述为优势专家的“压倒性”效应（overpowering）[15]。

此外，MoE的训练过程本身也存在不稳定性。门控网络和所有专家是联合训练的，但从一开始就不清楚为什么它们能够自动分化，学习到各自专门的功能，特别是当所有专家都从相同的权重分布初始化时 [3]。这实质上创造了一种“赛跑”条件，少数几个抢占先机的专家最终会赢得这场比赛，而其他专家则被远远甩在后面。门控网络本身也可能在训练中形成固有的偏见，进一步加剧了问题 [3]。

在这种背景下，“专家专业化”这一理想概念常常被误解。理想情况下，模型中的N个专家应该各自学习到独特且互补的知识领域。然而，在缺乏有效平衡机制的情况下，模型实际达到的状态更应被称为“专家集中化”（expert concentration），这本质上是一种冗余。对DeepSeekMoE的分析提供了有力的证据：单个权重最高的专家的输出，与整个专家组合（ensemble）的输出之间具有极高的相似度，某些层的余弦相似度甚至高达0.95 [17]。这表明，其他被激活的专家对最终结果的贡献微乎其微，或者其功能与主导专家高度重叠。因此，模型并未在所有专家中实现广泛的专业化分工，而是将关键的知识和能力集中到了一个极小的子集中。一个负载严重失衡的MoE模型，其真实的有效容量可能远低于其庞大的参数量所暗示的水平。

第二节 负载失衡引发的连锁负面影响

专家负载失衡并非一个孤立的理论问题，它会引发一系列连锁反应，从底层的系统效率到顶层的模型性能，都会受到严重的不利影响。本章节将详细阐述负载失衡所导致的计算瓶颈、资源浪费和模型质量下降等一系列严重后果。

2.1 系统级效率低下与计算瓶颈

在现代大规模MoE模型的训练和推理中，通常采用专家并行（Expert Parallelism）策略，即将不同的专家分布在不同的计算设备（如GPU）上 [9]。这种并行策略的效率高度依赖于负载的均衡分配。

当负载严重失衡时，会出现“掉队者问题”（straggler problem）。由于一小部分“热专家”处理了绝大多数令牌，承载这些专家的GPU将持续高负荷运转，而托管“冷专家”的大量其他

GPU则处于闲置或等待状态[18]。一个批次（batch）的总处理时间取决于最慢的设备，也就是那个最繁忙的“掉队者”。这导致了巨大的资源浪费，并完全破坏了并行计算带来的效率优势[27]。

此外，负载失衡还会导致通信开销的急剧增加和尾延迟（tail latency）的恶化。MoE层中的all-to-all通信步骤负责将每个令牌从其当前所在的GPU发送到托管其指定专家的GPU。如果负载不均，一个设备可能需要向许多其他设备发送大量令牌，而另一个设备可能只需要接收少量令牌。这种不均衡的通信模式会造成网络拥塞和效率低下的数据传输，从而增加端到端的执行时间，降低系统的整体吞吐量[27]。

2.2 令牌丢弃与容量管理的低效性

为了防止因“热专家”过载而导致的内存溢出和系统崩溃，研究人员引入了“专家容量”（expert capacity）的概念[12]。专家容量为一个专家在单个批次中能够处理的最大令牌数量设置了一个硬性上限[12]。

专家容量的计算通常遵循以下公式：

$$C = \text{round}(N_k \cdot T \cdot \gamma)$$

其中，T是批次中的总令牌数，N是专家总数，k是为每个令牌选择的专家数量（即Top-K中的K），而$\gamma是“容量因子”（capacity factor）12。容量因子γ通常是一个大于1.0的超参数，它为负载的潜在不均衡提供了一个缓冲或“余量”（slack）[12]。

当路由到一个专家的令牌数量超过其容量C时，多余的令牌就会被“丢弃”（dropped）[10]。这些被丢弃的令牌不会被MoE层中的任何专家处理，它们的信息只能通过残差连接（residual connection）传递到下一层。这种令牌丢弃直接损害了模型的学习过程和最终的准确性，因为它意味着模型在处理这些令牌时丢失了重要的计算步骤[10]。

与此相对，那些利用率不足的“冷专家”接收到的令牌数量远少于其容量。为了保持硬件计算所需的数据张量形状规整和静态，这些专家的输入通常会被填充（padded）大量的零或虚拟数据，以达到其容量上限。这种填充操作代表了纯粹的计算资源浪费，因为GPU需要处理这些毫无意义的数据[12]。

容量因子$\gamma的设计和调整，本身就体现了系统效率与模型精度之间的内在矛盾。一个较低的\gamma值（例如接近1.0）可以最大限度地减少填充和计算浪费，从而在负载完美均衡的理想情况下提高系统效率[31]。然而，现实世界的路由 rarely 是完美的。因此，当负载

失衡发生时，较低的

\gamma值将导致更多的令牌被丢弃，直接损害模型精度[12]。反之，一个较高的

\gamma值（例如2.0）提供了巨大的缓冲，能够容纳更多的令牌，减少丢弃率，从而保护模型精度，但这是以在大量未被充分利用的专家上进行无效填充和计算浪费为代价的[30]。因此，调整

\gamma$并非一个简单的技术细节，而是在效率和精度之间进行权衡的直接体现。寻找最优值本身就是一项困难且关键的任务，因为它依赖于对路由不均衡程度的预期[30]。

2.3 模型质量下降与专业化的失败

负载失衡对模型本身质量的损害是根本性的。那些长期处于“饥饿”状态、未经充分训练的专家，在模型中 фактически 成了“死权重”（dead weights）[4]。它们占据了大量的参数空间，却没有对模型的整体能力做出任何贡献。这种现象被称为“统计效率低下”（statistical inefficiency）[12]。在最极端的情况下，一个发生路由坍塌的MoE模型，其性能不会比一个参数量远小于它的密集模型更强大[12]。

更深层次的问题是“同质化表征”（homogeneous representation）和专家多样性的丧失。这是负载失衡在模型层面最严重的后果。当门控网络未能有效地引导专家走向专业化分工时，不同的专家最终可能会学习到相似甚至重叠的功能[25]。这种同质化问题意味着专家之间缺乏多样性，它们的权重表征可能高度相似（一项研究发现相似度可高达99%）[25]。这是一种严重的性能退化，因为它完全违背了“混合专家”架构的设计初衷[25]。

最近的研究还表明，专家利用率不足会直接影响模型的高级能力。当通过剪枝等方式移除专家（这可以看作是专家未被利用的代理）时，模型的指令遵循（instruction-following）能力受到的损害最为显著[34]。这暗示着，这些高级的推理能力可能非常脆弱，并且依赖于整个专家组合的协同作用。

综上所述，负载失衡会引发一个自我加剧的性能下降恶性循环。这个循环始于“富者愈富”的动态（第一节），导致部分专家过载[18]。为了防止系统崩溃，引入了容量限制，这又导致超额的令牌被丢弃[12]。被丢弃的令牌意味着模型在处理这些输入时获得了不完整或降级的学习信号，从而损害了其学习效果和整体性能[10]。与此同时，未被充分利用的专家得不到有效训练，成为“死权重”，无法实现专业化，进一步削弱了模型的有效容量[12]。这个性能下降的模型可能会因此更加依赖于它已知的少数几个“安全”且表现良好的专家，从而反过来加剧了最初的负载不均衡。这样，一个系统层面的问题（负载失衡）导致了一个模型层面的问题

（令牌丢弃和专业化失败），而模型质量的下降又会进一步恶化系统层面的问题。

第三节 训练时平衡策略的演进

为了应对专家负载失衡带来的严峻挑战，研究界开发了一系列在模型训练期间主动进行干预的策略。这些策略从最初基于损失函数的惩罚机制，逐步演进到更为精巧的无损算法和颠覆性的路由架构创新。本章节将对这些关键的平衡策略进行系统性的梳理和分析。

3.1 基础方法：辅助负载均衡损失（**LBL**）

这是解决负载失衡问题的经典方法，属于“有损控制”（Loss-Controlled）的范畴 [20]。其核心思想是在模型的主要损失函数（如语言建模损失）之外，额外增加一个辅助损失项（Auxiliary Load Balancing Loss, LBL），该损失项专门用于惩罚不均衡的令牌分配行为，从而激励门控网络将令牌更均匀地分发给所有专家 [4]。

3.1.1 原理与公式

在GShard、Switch Transformer等开创性的MoE模型中，广泛采用了以下形式的辅助损失函数 [30]：

$$L_{aux} = \alpha \cdot \sum_{i=1}^{N} (f_i \cdot P_i)$$

其中：

- N 是专家总数。
- f_i 是在一个批次（batch）中，被分派到专家 i 的令牌所占的比例。这个项衡量了每个专家的实际负载。
- P_i 是门控网络为专家 i 生成的平均路由概率（或称门控分数）在整个批次上的均值。这个项衡量了路由器对每个专家的选择意图。
- α 是一个超参数，用于控制这个辅助损失在总损失中的权重 [30]。

该损失函数的目标是使每个专家的实际负载 f_i 和路由意图 P_i 都趋向于一个均匀分布（即接

近 1/N)，此时损失值最小 [38]。

3.1.2 LBL的困境：平衡与干扰

尽管LBL在一定程度上能够缓解负载失衡，但它也带来了自身的核心矛盾，即"LBL困境"。

- 权衡的艺术：辅助损失的权重 α 极难调整。如果 α 值过小，辅助损失的惩罚力度不足，无法有效阻止路由坍塌。反之，如果 α 值过大，虽然可以强制实现负载均衡，但往往会损害模型的最终性能 [21]。
- **"干扰梯度"**：这是LBL最主要的副作用。LBL引入的梯度与模型的主要任务（如语言建模）无关。这些"干扰梯度"（interference gradients）可能会与主任务的梯度发生冲突或抵消，从而扰乱正常的学习过程 [21]。为了满足负载均衡的要求，LBL会迫使门控网络做出一种"虚假"的令牌分配，例如将一个明显属于代码领域的令牌分配给一个处理自然语言的专家，这直接阻碍了专家专业化的形成 [30]。

3.1.3 关键改进：全局批次 vs. 微批次 LBL

一个看似微小但影响深远的实现细节是LBL的计算范围。一项关键研究指出了微批次（micro-batch）LBL和全局批次（global-batch）LBL之间的巨大差异 [39]。

- 微批次**LBL**（问题所在）：在标准的数据并行训练中，LBL通常在每个设备处理的微批次内计算。一个微批次通常只包含极少数序列，且这些序列的主题可能非常相似。在这种情况下，LBL会迫使路由器在每个序列内部均匀地分配令牌。这是一个极其严格的约束，它会主动抑制专家专业化。例如，它会强迫一个包含代码的序列中的令牌被均匀地路由到所有专家，包括那些非代码专家，这显然是不合理的 [39]。
- 全局批次**LBL**（解决方案）：这种改进策略通过一个额外的通信步骤，在计算LBL之前，跨所有并行的微批次同步专家的负载统计数据（即 f_i）。这样，LBL的目标就变成了在整个语料库层面上实现负载均衡。它允许在一个批次内部出现专业化的路由行为（例如，一个批次中的所有代码令牌都可以被路由到同一个代码专家），只要从全局来看，所有专家的负载是均衡的。研究证明，这种方法能够显著改善模型的预训练困惑度（perplexity）和下游任务性能，因为它为真正的领域专业化创造了条件 [39]。

3.1.4 辅助稳定性损失：路由器Z-Loss

除了LBL，研究人员还引入了其他辅助损失来提高训练稳定性。其中，路由器Z-Loss（Router Z-Loss）是一个重要的补充。该损失项旨在惩罚门控网络输出的过大logit值。在低精度（如float16）训练中，过大的logit值在经过softmax函数时容易导致数值不稳定和舍入误差。Z-Loss通过对logit施加正则化，有助于稳定训练过程，同时不影响模型质量 [30]。

3.2 范式转移：无损平衡机制

为了彻底摆脱“干扰梯度”的困扰，研究界提出了一类全新的“无损平衡”（Loss-Free Balancing）策略。这些方法的核心目标是在不向模型的总损失函数中添加任何项的情况下控制负载均衡，从而确保训练信号的纯净性 [21]。

3.2.1 动态偏置调整

这是目前最主流的无损平衡策略之一 [21]。

- 机制：在门控网络计算出原始的路由分数 si 后，Top-K选择之前，该方法会为每个专家加上一个独立的偏置项 bi。最终的路由决策基于调整后的分数 si+bi。
- 动态更新：这个偏置项 bi 并非通过梯度下降学习。相反，它根据每个专家近期的负载情况，通过一个简单的算法进行动态更新。如果一个专家近期负载过高，其偏置 bi 就会被调低，使其在下一轮中被选中的概率降低。反之，如果专家负载过低，其偏置 bi 就会被调高 [13]。这构成了一个简单而有效的自适应控制系统。
- 优势：这种方法巧妙地只影响了Top-K专家的选择过程，而不会改变最终用于加权组合专家输出的概率值。因此，它不会产生干扰梯度，使得模型的训练完全聚焦于主任务目标，训练信号更为“纯净” [23]。

3.2.2 基于优化的路由

这类方法引入了更复杂的数学优化技术来追求完美的负载均衡。

- 二元整数规划（**BIP**）：一项研究提出，通过在每一步求解一个微小的二元整数规划问题，来辅助调整Top-K的选择顺序。实验表明，该方法能够从训练的第一个步骤开始就维

持近乎完美的负载均衡状态，并最终获得更低的困惑度 [20]。

- 整数线性规划（**ILP**）：另一项研究则利用整数线性规划来寻找专家在GPU上的最优部署方案。该方法不仅考虑令牌负载，还综合考虑了通信成本和计算时间，并利用了层间路由的亲和性（即被路由到第L层专家A的令牌，在第L+1层很可能被路由到一个可预测的小范围专家集）。通过这种全局优化，可以有效减少设备间的数据传输，实现负载均衡，并显著提升推理速度 [27]。

3.3 重新思考路由器：颠覆性的架构创新

除了调整平衡算法，另一大研究方向是对路由器本身进行结构性创新，从根本上改变令牌分配的规则。

3.3.1 专家选择路由（**Expert Choice Routing**）

这是一种对路由逻辑的优雅而彻底的颠覆 [18]。

- 反向逻辑：传统的路由是“令牌选择专家”。而专家选择路由则反其道而行之，变成了“专家选择令牌”。在该机制下，每个专家都被分配了一个固定的容量（或称“桶大小”），然后每个专家会从当前批次的所有令牌中，挑选出与自己亲和度最高的K个令牌进行处理。
- 核心优势：这种设计从机制上保证了完美的负载均衡，因为每个专家处理的令牌数量是固定的。这完全消除了对任何辅助损失的需求 [18]。
- 次要优势：它自然地实现了每个令牌可由不同数量的专家处理。重要的、复杂的令牌可能会被多个专家同时选中，而简单的令牌可能不会被任何专家选中，这为计算资源的灵活分配提供了可能 [18]。
- 潜在缺陷：一个关键的批评是，在自回归模型（如LLM）中，该机制存在“未来令牌泄露”（future token leakage）的风险。因为专家的选择是全局的（即在位置t的专家可以看到并选择来自未来位置t+n的令牌），这破坏了自回归任务严格的因果关系约束。这可能导致模型在训练时看到“答案”，从而获得虚假的低损失，但在实际生成时表现不佳 [23]。

3.3.2 动态路由（自适应**K**值 ）

这类方法挑战了传统的、固定的Top-K假设 [46]。

- 核心思想：不同的令牌具有不同的处理难度。简单的令牌（如停用词）可能只需要一个专家甚至不需要专家，而涉及复杂推理的令牌则可能需要更多专家的协同工作。动态路由的核心是根据某种对输入难度或路由置信度的度量，为每个令牌动态地调整激活的专家数量 k [46]。
- 实现方式：一些方法利用门控网络的输出置信度来决定 k 的大小 [46]。另一些方法则引入不消耗任何计算资源的“空专家”（null experts），模型通过将令牌路由到这些空专家，来间接实现使用更少数量的真实专家 [52]。更先进的方法则利用强化学习（如PPO算法）来端到端地训练一个轻量级的“分配器”（allocator）模块，由该模块为每个令牌决定最优的
  k 值 [53]。
- 优势：实验表明，这些方法可以在使用更少平均激活专家数的情况下，取得优于固定Top-2路由的性能，从而在保证质量的同时提升了计算效率 [46]。

3.3.3 无丢弃与重分配策略

这类方法的目标是确保批次中的每一个令牌都得到有效处理。

- **DeepSpeed-MoE**的重分配：当一个专家过载时，超出其容量的令牌不会被直接丢弃，而是被重新路由到那些还有空余容量的专家那里 [30]。
- **JetMoE**的无丢弃**MoE**：这是一种更先进的策略，它利用专门为稀疏计算设计的GPU内核（如MegaBlocks的内核），来高效地处理动态和不均衡的负载，从硬件层面确保所有令牌都能被处理，无需丢弃 [30]。

MoE负载均衡策略的演进，反映了机器学习领域一个更广泛的趋势：从“纠正性”设计转向“建构性”设计。LBL是一种典型的“纠正性”或“反应性”方法：它观察到一个坏行为（负载失衡），然后增加一个惩罚项来试图修复它，这好比在一个有缺陷的系统上打补丁。“干扰梯度”[21] 正是这个补丁最主要的副作用。相比之下，无损平衡和专家选择路由等方法则是“建构性”或“前瞻性”的设计。它们从根本上改变了算法（如偏置调整）或架构（如专家-令牌选择的反转），使得坏行为从一开始就更难发生或不可能发生。这反映了该领域的成熟，即从试图用损失函数解决问题，转向通过算法和系统本身的结构来内在地解决问题。

同时，对“平衡”的定义也经历了一个从僵化到灵活的演变过程。微批次LBL要求在极细粒度上实现严格的均匀分配，但这会损害专业化 [39]。全局批次LBL放宽了这一要求，仅追求语料库层面的平衡，允许局部专业化 [39]。专家选择路由则保证了专家层面的完美平衡，但允许

令牌层面出现“不均衡”（即不同令牌获得不同数量的专家服务）[18]。而动态路由则更进一步，认为均匀的负载均衡本身就不是最终目标，真正的目标应该是

高效的资源分配，即根据任务需求动态地调配计算资源（专家）[46]。这揭示了“负载均衡”并非一个单一的概念，研究正在从追求僵化的平衡，转向一种更智能、更具适应性的动态资源管理范式。

最后，ILP/BIP [20] 和自适应算法（如动态偏置）[21] 的引入，标志着运筹学和经典工程控制论的思想正在与深度学习系统设计进行交叉融合。LBL是纯粹的深度学习产物，而动态偏置调整本质上是一个源于控制系统理论的比例-积分（PI）控制器，偏置根据误差（负载偏差）进行调整。ILP/BIP则是运筹学中解决复杂优化和调度问题的经典工具。这一趋势表明，随着LLM系统变得日益复杂，解决方案不再仅仅源于神经网络理论，而是越来越多地借鉴更成熟的工程和数学优化领域的知识来应对系统级的挑战。

第四节 综合、建议与未来展望

经过对MoE负载失衡问题及其解决方案的深入探讨，本章将对各类策略进行综合性的比较分析，为实践者提供具体建议，并展望该领域未来的研究方向和面临的挑战。

4.1 平衡策略的比较分析

各种负载均衡策略并非简单的优劣之分，而是在不同维度上存在复杂的权衡。为了清晰地展示这些权衡，下表对前文讨论的主要策略进行了系统性的比较。

策略	核心机制	平衡保证	主要优势	主要劣势/权衡	关键相关模型/文献
辅助损失 (微批次)	在每个微批次内，向主损失添加一个惩罚项（Laux=α·Σ(fi·Pi)），以鼓励令牌在专家间均匀分配。	无保证。鼓励平衡，但可能被主任务损失压制。	实现简单，是早期MoE模型的标准配置。	引入干扰梯度，严重抑制专家专业化，迫使在序列内部进行不必要的分散。	GShard, Switch Transformer (早期实现) [30]；被[39]批判

辅助损失 (全局批次)	在计算LBL前，跨所有并行设备同步专家负载统计数据，在全局批次（语料库）层面强制均衡。	无保证，但比微批次更有效。	在保持LBL简单性的同时，允许局部（序列内）专业化，显著提升模型性能。	仍然存在干扰梯度问题，需要仔细调整α值。	[39]
无损偏置调整	在路由决策前，为每个专家的原始门控分数加上一个根据其近期负载动态调整的偏置项bi。	强保证。通过反馈控制系统，负载会收敛到均衡状态。	不产生干扰梯度，训练信号纯净；与专家并行天然兼容。	平衡收敛速度依赖于偏置更新率；可能不如LBL对突发失衡的反应迅速。	Loss-Free Balancing, DeepSeek-V2/V3 [21]
专家选择路由	颠倒路由逻辑：由每个专家从批次中选择亲和度最高的Top-K个令牌进行处理。	完美保证。每个专家处理的令牌数量完全相同。	从机制上根除了负载失衡问题，无需任何辅助损失；允许令牌获得可变数量的专家服务。	在自回归任务中存在未来令牌泄露风险，可能破坏因果约束；改变了核心路由逻辑。	Expert Choice [18]
动态K值路由 (自适应)	根据每个令牌的复杂度或路由置信度，动态决定为其激活的专家数量k。	不以均衡为首要目标，而是追求高效计算。	能够为简单任务节省大量计算（FLOPs），将更多资源用于困难任务；比固定Top-K更高效。	实现更复杂，可能需要引入强化学习（PPO）或额外的模块（分配器），增加了训练开销。	Dynamic MoE, Ada-K [46]

从上表可以看出，选择何种策略取决于具体的应用场景和优化目标。辅助损失（特别是全局批次版本）是一种成熟且有效的基线方法，但其干扰梯度问题是其固有缺陷。无损偏置调整提供了一种更为“纯净”的训练范式，旨在不影响主任务学习的前提下实现平衡，代表了对LBL方法的直接改进。专家选择路由则是一种更为激进的架构变革，它以完美的系统平衡性为目标，但可能需要在使用场景上做出权衡（如验证其在自回归任务中的适用性）。最后，动态K值路由则将优化的焦点从“平衡”转向了“效率”，认为计算资源应按需分配，这代表了对MoE资源管理理念的进一步深化。

4.2 对实践者的建议

基于上述分析，可以为正在研究或应用MoE模型的工程师和研究人员提供以下具体建议：

- 追求最佳性能与专业化：对于希望最大化模型性能和专家专业化程度的场景，建议优先考虑全局批次辅助损失（**Global-Batch LBL**）[39]或
  无损偏置调整（**Loss-Free Balancing**）[21]。这两种方法都被证明能更好地保留专家学习特定领域知识的能力，同时有效控制负载。
- 保证系统稳定性与均衡：如果在硬件资源高度受限或对系统延迟有严格要求的环境中，专家选择路由（**Expert Choice Routing**）[18]是一个极具吸引力的选项。它通过设计保证了完美的负载均衡，从而消除了系统中的“掉队者”瓶颈。但应用前必须仔细评估其在具体任务（尤其是自回归生成任务）中是否存在“未来令牌泄露”的负面影响。
- 实现极致计算效率：当目标是最大限度地减少总计算量（FLOPs）时，探索动态**K**值路由（**Dynamic K Routing**）[46]将是最佳选择。这类方法通过避免在简单令牌上进行不必要的计算，实现了对计算资源的智能分配，有望在同等性能下获得更高的推理速度。
- 应避免的策略：强烈建议避免使用朴素的微批次辅助损失（**Micro-Batch LBL**）。如[39]中的研究所示，这种方法对专家专业化的抑制作用非常显著，可能会导致模型性能不及预期。

4.3 开放性挑战与未来研究方向

尽管MoE的负载均衡技术已取得长足进步，但仍有许多开放性问题和充满潜力的研究方向值得探索。

- 动态与自适应系统：当前的研究趋势正朝着更加动态和自适应的系统发展 [55]。未来的路由机制可能会超越对单个令牌的分析，融合更广泛的上下文信息，甚至根据外部环境因素动态调整策略 [51]。
- 推理时负载均衡：目前绝大多数平衡策略都是为训练阶段设计的。一个关键的开放挑战是如何确保模型在推理时，尤其是在面对分布外（out-of-distribution）数据或对抗性输入时，仍能保持负载均衡。预训练阶段学到的路由模式在这些情况下可能会失效，导致性能瓶颈 [30]。
- 自动化与可组合架构：MoE的架构设计仍有巨大的探索空间。未来的研究可能集中在利用神经架构搜索（NAS）等技术来自动发现最优的专家组合、路由机制甚至是层级化的MoE结构（hierarchical MoE），以适应不同任务的需求 [14]。
- 鲁棒性与可解释性：如何使专家学习到的知识更具鲁棒性并易于解释，是一个核心的科学问题。这包括防止专家功能重叠、理解每个专家具体“专长”于什么，以及如何诊断和修复“生病”的专家 [11]。
- 自适应负载均衡算法的深化应用：将更成熟的自适应算法从通用系统设计领域引入

MoE是一个前景广阔的方向[58]。例如，构建能够实时监控专家负载、预测未来令牌分布并动态调整路由策略或专家容量的系统，有望实现更精细和高效的资源管理[59]。

总之，MoE架构中的负载均衡问题已经从一个单纯的系统工程挑战，演变为一个深度融合了算法设计、系统优化和模型理论的交叉学科领域。未来的突破将不仅依赖于更巧妙的损失函数，更将来自于对路由机制的根本性创新和对整个AI系统动态、自适应能力的深刻理解。

Works cited

1. A Comprehensive Survey of Mixture-of-Experts: Algorithms, Theory, and Applications - arXiv, accessed June 15, 2025, https://arxiv.org/html/2503.07137v1
2. A Survey on Mixture of Experts in Large Language Models - arXiv, accessed June 15, 2025, https://arxiv.org/pdf/2407.06204
3. MomentumSMoE: Integrating Momentum into Sparse Mixture of Experts - NIPS, accessed June 15, 2025, https://proceedings.neurips.cc/paper_files/paper/2024/file/32eb183794ef5ef9a3ab1d40a3d2b303-Paper-Conference.pdf
4. What is mixture of experts? | IBM, accessed June 15, 2025, https://www.ibm.com/think/topics/mixture-of-experts
5. A Survey on Mixture of Experts - arXiv, accessed June 15, 2025, https://arxiv.org/pdf/2407.06204?
6. Toward Efficient Inference for Mixture of Experts - University of Pennsylvania, accessed June 15, 2025, https://www.seas.upenn.edu/~leebcc/documents/huang24-neurips.pdf
7. How To Implement Mixture of Experts (MoE) in PyTorch - ApX Machine Learning, accessed June 15, 2025, https://apxml.com/posts/how-to-implement-moe-pytorch
8. What Is Mixture of Experts (MoE)? How It Works, Use Cases & More | DataCamp, accessed June 15, 2025, https://www.datacamp.com/blog/mixture-of-experts-moe
9. Mixture of Experts LLMs: Key Concepts Explained - Neptune.ai, accessed June 15, 2025, https://neptune.ai/blog/mixture-of-experts-llms
10. Accelerating MoE Model Inference with Expert Sharding - arXiv, accessed June 15, 2025, https://arxiv.org/html/2503.08467v1
11. (PDF) Improving Expert Specialization in Mixture of Experts - ResearchGate, accessed June 15, 2025, https://www.researchgate.net/publication/368877500_Improving_Expert_Specialization_in_Mixture_of_Experts
12. Scaling Vision with Sparse Mixture of Experts - NIPS, accessed June 15, 2025, https://papers.neurips.cc/paper_files/paper/2021/file/48237d9f2dea8c74c2a72126cf63d933-Paper.pdf
13. Mixture of experts - Wikipedia, accessed June 15, 2025, https://en.wikipedia.org/wiki/Mixture_of_experts

14. Understanding Mixture of Experts (MoE): A Deep Dive into Scalable AI Architecture, accessed June 15, 2025, https://www.researchgate.net/publication/388828999_Understanding_Mixture_of_Experts_MoE_A_Deep_Dive_into_Scalable_AI_Architecture
15. Mixture-of-Experts: a publications timeline, with serial and distributed implementations, accessed June 15, 2025, https://brunomaga.github.io/Mixture-of-Experts
16. Switch Transformers: Scaling to Trillion Parameter Models with Simple and Efficient Sparsity - Journal of Machine Learning Research, accessed June 15, 2025, https://jmlr.org/papers/volume23/21-0998/21-0998.pdf
17. MoE Lens - An Expert Is All You Need | OpenReview, accessed June 15, 2025, https://openreview.net/forum?id=GS4WXncwSF
18. Mixture-of-Experts with Expert Choice Routing, accessed June 15, 2025, https://papers.neurips.cc/paper_files/paper/2022/file/2f00ecd787b432c1d36f3de9800728eb-Paper-Conference.pdf
19. Mixture-of-Experts with Expert Choice Routing - Google Research, accessed June 15, 2025, https://research.google/blog/mixture-of-experts-with-expert-choice-routing/
20. [2502.15451] Binary-Integer-Programming Based Algorithm for Expert Load Balancing in Mixture-of-Experts Models - arXiv, accessed June 15, 2025, https://arxiv.org/abs/2502.15451
21. Auxiliary-Loss-Free Load Balancing Strategy for Mixture-of-Experts - arXiv, accessed June 15, 2025, https://arxiv.org/html/2408.15664v1
22. [2408.15664] Auxiliary-Loss-Free Load Balancing Strategy for Mixture-of-Experts - arXiv, accessed June 15, 2025, https://arxiv.org/abs/2408.15664
23. Auxiliary-Loss-Free Load Balancing Strategy for Mixture-of-Experts | OpenReview, accessed June 15, 2025, https://openreview.net/forum?id=y1iU5czYpE
24. Auxiliary-Loss-Free Load Balancing Strategy for Mixture-of-Experts - Aili, accessed June 15, 2025, https://aili.app/share/bVRZBuflqvSxlESL6rM0u
25. Diversifying the Mixture-of-Experts Representation for Language Models with Orthogonal Optimizer arXiv:2310.09762v2 [cs.CL] 30, accessed June 15, 2025, https://arxiv.org/pdf/2310.09762
26. Towards Understanding the Mixture-of-Experts Layer in Deep Learning, accessed June 15, 2025, https://proceedings.neurips.cc/paper_files/paper/2022/file/91edff07232fb1b55a505a9e9f6c0ff3-Supplemental-Conference.pdf
27. [2502.06643] MoETuner: Optimized Mixture of Expert Serving with Balanced Expert Placement and Token Routing - arXiv, accessed June 15, 2025, https://arxiv.org/abs/2502.06643
28. MoETuner: Optimized Mixture of Expert Serving with Balanced Expert Placement and Token Routing - arXiv, accessed June 15, 2025, https://arxiv.org/html/2502.06643v1
29. [2105.15082] M6-T: Exploring Sparse Expert Models and Beyond - ar5iv, accessed June 15, 2025, https://ar5iv.labs.arxiv.org/html/2105.15082

30. A Review on the Evolvement of Load Balancing Strategy in MoE LLMs: Pitfalls and Lessons, accessed June 15, 2025, https://huggingface.co/blog/NormalUhr/moe-balance
31. Mixtures of Experts and scaling laws - Nebius, accessed June 15, 2025, https://nebius.com/blog/posts/mixture-of-experts
32. Mixture of Experts package - NVIDIA Docs, accessed June 15, 2025, https://docs.nvidia.com/megatron-core/developer-guide/latest/api-guide/moe.html
33. Advancing Expert Specialization for Better MoE - arXiv, accessed June 15, 2025, https://arxiv.org/html/2505.22323v1
34. [2504.05586] Finding Fantastic Experts in MoEs: A Unified Study for Expert Dropping Strategies and Observations - arXiv, accessed June 15, 2025, https://arxiv.org/abs/2504.05586
35. Load balancing loss in Switch Transformer · Issue #29503 - GitHub, accessed June 15, 2025, https://github.com/huggingface/transformers/issues/29503
36. Incorrect implementation of auxiliary loss · Issue #28255 · huggingface/transformers - GitHub, accessed June 15, 2025, https://github.com/huggingface/transformers/issues/28255
37. AUXILIARY-LOSS-FREE LOAD BALANCING STRAT- EGY FOR MIXTURE-OF-EXPERTS - OpenReview, accessed June 15, 2025, https://openreview.net/pdf/138f19eedd33952236974ad6aac9a9dcd545d462.pdf
38. Switch Transformers: Scaling to Trillion Parameter Models with Simple and Efficient Sparsity - cs.Princeton, accessed June 15, 2025, https://www.cs.princeton.edu/courses/archive/fall22/cos597G/lectures/lec16.pdf
39. arXiv:2501.11873v2 [cs.LG] 4 Feb 2025, accessed June 15, 2025, https://arxiv.org/abs/2501.11873
40. Mixture of Experts in AI: Enhancing Large Language Models - Dragonscale Newsletter, accessed June 15, 2025, https://blog.dragonscale.ai/mixture-of-experts/
41. An implementation of the MoE router z-loss in PyTorch. - GitHub Gist, accessed June 15, 2025, https://gist.github.com/wolfecameron/2305c8c9ccc6d2c2906ba4577d801ccc
42. Implementation of ST-Moe, the latest incarnation of MoE after years of research at Brain, in Pytorch - GitHub, accessed June 15, 2025, https://github.com/lucidrains/st-moe-pytorch
43. Auxiliary-Loss-Free Load Balancing Strategy for Mixture-of-Experts - AIModels.fyi, accessed June 15, 2025, https://www.aimodels.fyi/papers/arxiv/auxiliary-loss-free-load-balancing-strategy-mixture
44. [Literature Review] Auxiliary-Loss-Free Load Balancing Strategy for Mixture-of-Experts, accessed June 15, 2025, https://www.themoonlight.io/en/review/auxiliary-loss-free-load-balancing-strategy-for-mixture-of-experts
45. Auxiliary-Loss-Free Load Balancing Strategy for Mixture-of-Experts - ResearchGate, accessed June 15, 2025,

https://www.researchgate.net/publication/383494436_Auxiliary-Loss-Free_Load_Balancing_Strategy_for_Mixture-of-Experts
46. Harder Tasks Need More Experts: Dynamic Routing in MoE Models - arXiv, accessed June 15, 2025, https://arxiv.org/html/2403.07652v1
47. ZhenweiAn/Dynamic_MoE: Inference Code for Paper "Harder Tasks Need More Experts: Dynamic Routing in MoE Models" - GitHub, accessed June 15, 2025, https://github.com/ZhenweiAn/Dynamic_MoE
48. [2403.07652] Harder Tasks Need More Experts: Dynamic Routing in MoE Models - arXiv, accessed June 15, 2025, https://arxiv.org/abs/2403.07652
49. Harder Tasks Need More Experts: Dynamic Routing in MoE Models - ChatPaper, accessed June 15, 2025, https://chatpaper.com/de/paper/121
50. Harder Task Needs More Experts: Dynamic Routing in MoE Models - ACL Anthology, accessed June 15, 2025, https://aclanthology.org/2024.acl-long.696/
51. DA-MoE: Towards Dynamic Expert Allocation for Mixture-of-Experts Models | Papers With Code, accessed June 15, 2025, https://paperswithcode.com/paper/da-moe-towards-dynamic-expert-allocation-for
52. AdaMOE: Token-Adaptive Routing with Null Experts for Mixture-of-Experts Language Models - ACL Anthology, accessed June 15, 2025, https://aclanthology.org/2024.findings-emnlp.361.pdf
53. Ada-K Routing: Boosting the Efficiency of MoE-based LLMs | OpenReview, accessed June 15, 2025, https://openreview.net/forum?id=9CqkpQExe2
54. GShard: Scaling Giant Models with Conditional Computation and Automatic Sharding, accessed June 15, 2025, https://openreview.net/forum?id=qrwe7XHTmYb
55. Future Directions in Mixture of Experts Research: Towards More Dynamic and Adaptive AI Systems - Modular, accessed June 15, 2025, https://www.modular.com/ai-resources/future-directions-in-mixture-of-experts-research-towards-more-dynamic-and-adaptive-ai-systems
56. MoE at Scale: From Modular Design to Deployment in Large-Scale Machine Learning Systems - Preprints.org, accessed June 15, 2025, https://www.preprints.org/manuscript/202504.1313/v1
57. Addressing Challenges in Mixture of Experts: Load Balancing and Routing Mechanisms - AI Resources - Modular, accessed June 15, 2025, https://www.modular.com/ai-resources/addressing-challenges-in-mixture-of-experts-load-balancing-and-routing-mechanisms
58. Adaptive Load Balancing – System Design | GeeksforGeeks, accessed June 15, 2025, https://www.geeksforgeeks.org/adaptive-load-balancing-system-design/
59. EfficientMoE: Optimizing Mixture-of-Experts Model Training With Adaptive Load Balance, accessed June 15, 2025, https://www.computer.org/csdl/journal/td/2025/04/10876795/247s0GLFJN6
60. EfficientMoE: Optimizing Mixture-of-Experts Model Training With Adaptive Load Balance, accessed June 15, 2025, https://www.researchgate.net/publication/388784031_EfficientMoE_Optimizing_Mixture-of-Experts_Model_Training_with_Adaptive_Load_Balance?_tp=eyJjb250ZX

h0ljp7InBhZ2UiOiJzY2llbnRpZmljQ29udHJpYnV0aW9ucyIsInByZXZpb3VzUGFnZSI6bnVsbCwic3ViUGFnZSI6bnVsbH19