arXiv:2603.08960v1 [cs.LG] 9 Mar 2026

The qs 不等式：量化推理时混合专家的双重惩罚

Vignesh Adhinarayanan [1] **Nuwan Jayasena** [1]

摘要

混合专家（MoE）模型以较低的训练浮点运算次数实现了较高质量，但这种效率在推理时往往会消失。我们发现了一种双重惩罚，它在结构上使混合专家架构在解码过程中处于不利地位：首先，专家路由会造成微批次碎片化并减少权重复用；其次，大规模驻留专家池会压缩键值缓存可用的高带宽内存（HBM）空间。这一现象被形式化为复用碎片化，会使前馈网络（FFN）陷入带宽受限状态，尤其是在长上下文长度下。

我们**引入 qs 不等式**，这是一项预测性标准，用于识别相较于质量匹配的稠密模型，混合专家何时会在结构上处于不利地位。该标准统一了**稀疏性**（s）（即每个词元激活的参数占比）与**质量等效因子**（q）（即稠密模型要达到混合专家性能所需的规模倍数）。我们在包括 DeepSeek-V3、Qwen3-235B、Grok-1 和 Switch-C 在内的前沿模型上进行评估，结果表明，这种碎片化是一种普遍存在的架构现象。对于 128k上下文下的 DeepSeek-V3，这使质量匹配的稠密基线获得了 4.5 倍的吞吐量优势。至关重要的是，在质量匹配的稠密模型仍可运行的集群规模下，Switch-C 等超大规模架构可能会变得不可行。

我们的结果表明，训练阶段的浮点运算次数效率并不能完整地反映长上下文服务中的推理性能。这些结果还表明，混合专家或许最好被视为一种训练阶段的优化方法，而将其蒸馏为稠密模型，则可能是实现高效推理部署的一条路径。

[1]AMD 研究与高级开发（RAD）。通讯联系人：Vignesh Adhinarayanan <vig- nesh.adhinarayanan@amd.com>, Nuwan Jayasena<nuwan.jayasena@amd.com>。

预印本。*2026*年*3*月*11*日。

1. 引言

混合专家（MoE）架构旨在提高大规模语言模型训练的效率。通过每个词元仅激活一小部分参数，混合专家将模型容量与每令牌计算量解耦，使训练浮点运算次数几乎保持不变，同时总参数数量可扩展至数千亿甚至更多（Shazeer等人，2017；Fedus等人，2022；Lepikhin等人，2020年）。这种稀疏性并非仅有理论上的优势；实践中，在计算预算相同的情况下，混合专家模型的验证损失始终低于稠密模型，因此多个前沿系统都采用了这种架构（Du等人，2022；Rajbhandari等人，2022）。

推理设置提出了一个不同的问题。自回归解码必须满足严格的延迟服务级别目标，在适中的批大小下运行，并为所有先前生成的令牌保留键值（KV）缓存张量。在这种情况下，性能瓶颈常常从算术吞吐量转向内存带宽和数据移动。因此，降低每个词元的浮点运算次数并不一定能降低推理延迟，如果它同时增加每个词元传输的字节数，或使这些字节无法在多个令牌之间摊销。在整个工作中，我们聚焦于图形处理器和TPU等基于DRAM的加速器，其中HBM 流量主导推理性能；虽然定性论证可延伸至以SRAM为中心的设计，但定量分析以高带宽内存为中心。

这一观察揭示了一个潜在的不匹配。混合专家的设计目标明确是减少浮点运算次数，而推理性能往往取决于权重能否在词元之间得到有效复用。为分析这一状态，我们提出复用原则：推理效率取决于每次权重读取能够被多少词元复用，而非避免了多少浮点运算次数。稠密前馈网络（前馈网络）天然符合这一原则，因为它们能将权重读取的开销分摊到整个微批次。混合专家层在现实的推理约束下却无法做到这一点。专家路由会将微批次划分给不同专家，减少每个专家的复用次数，并增加每个词元对应的权重流量。

我们表明，这种效应是结构性的。在推理过程中，批次大小受 KV缓存驻留的限制，尤其是在上下文长度较长时。随着批次大小缩小，混合专家路由会将微批次分散到不同专家，导致每个专家的复用次数趋近于 1，使前馈网络执行进入带宽受限状态，即使实际执行的浮点运算次数仅占稠密模型的很小一部分。我们将这一机制称为复用碎片化，并在第2节对其进行形式化定义。

复用碎片化还会带来一个次级但在实践中意义重大的后果。由于整个专家池必须常驻于服务集群中，混合专家会占用原本可用于键值缓存的高带宽内存（HBM）容量。这会收紧可接受的批次大小，进一步缩小每个专家的子批次，并在一个不断叠加的循环中加剧复用碎片化。我们针对包括 DeepSeek-V3、Qwen3-235B、Grok-1 和 Switch-C 在内的前沿混合专家系统，对这一结构性劣势进行了量化。

我们的核心结论是，训练时FLOP效率并不是推理时经济性的良好代理指标。在现实的服务约束下，尤其是在长上下文场景中，质量匹配的稠密模型能够实现更低的推理成本，因为它们保留了混合专家无法保留的复用能力。这一结论源于复用碎片化，而复用碎片化是结构性的，并非偶发现象。

贡献。

- 我们将**权重复用**而非浮点运算次数，确定为推理效率的关键决定因素（第2节）。
- 我们将**复用碎片化**形式化为推理时专家路由所导致的结构性结果：每个 $R_{\text{混合专家}} \approx Bk/E$ 的复用量按2–3节所述方式缩放。
- 我们推导出 qs **不等式**，将其作为预测边界，用以刻画混合专家推理相较于同等质量稠密模型在结构上处于劣势的条件（见第3节）。
- 我们量化了前沿混合专家模型在容量和吞吐量方面的影响，表明常驻专家池会压缩KV缓存余量，而质量匹配的稠密模型吞吐量优势最高可达5.3倍（见第5–6节）。

这些观察结果促使我们重新审视训练时架构与推理时架构之间的关系；这一主题将在第7节中再次探讨。

2. 混合专家推理的主导原则：复用

本节介绍混合专家在推理时的执行模型，并提出一种延迟分解，表明在现实的推理约束下，稠密模型与混合专家模型之间的性能差异主要源于前馈权重复用。

2.1. 推理时的混合专家前馈执行

我们对比混合专家模型与稠密模型在推理时的执行方式。两种情况下，注意力模块和 KV缓存的访问均保持不变：无论后续前馈计算采用稠密方式还是专家路由方式，每个生成的词元都会读取相同的键值缓存状态以进行注意力计算。因此，推理时的关键区别在于前馈网络（FFN）的执行方式。

稠密 Transformer 在每一层执行一个 FFN，将相同的权重矩阵应用于包含 B 个词元的整个微批次。从硬件的角度看，这相当于每层执行一次大型且摊销充分的通用矩阵乘法，而从内存中读取 FFN 权重的开销则由整个批次共同分摊。

MoE 层将这种执行模式替换为一组 E 个彼此独立的 FFN（专家），每个 FFN 都有自己的权重矩阵。学习型路由器计算词元到专家的分配，并为每个词元选择 k 个专家。在专家并行部署中，这会引入一个分发步骤：将词元嵌入路由到承载所选专家的设备。随后，每个专家对接收到的词元执行其 FFN。生成的专家输出会被散布回原始批次索引，并在所选的 k 个专家之间进行归约，最终得到一个形状与稠密 FFN 输出相同的输出张量。

因此，原始微批次被划分为各个专家本地的微批次，每个专家仅对路由到自身的词元执行其前馈网络。平均而言，每个专家每步处理

$$B_{\text{expert}} = \frac{Bk}{E} \tag{1}$$

个词元。当 E 较大而 B 受限时，B_{expert} 可能只有个位数，甚至接近于 1。这个表达式反映了专家间的平均路由行为，并体现了权重摊销所依赖的有效复用。重要的是，稀疏性并不会消除权重移动：每个专家在处理任何词元之前，仍必须从内存中读取自己的权重矩阵。

由于整个专家池都必须常驻于高带宽内存中，混合专家模型留给键值缓存的空间也比可比的稠密模型更少。随着上下文长度增长，可接受的批次大小会缩小，进一步降低B_{expert}，并加剧摊销损失。

2.2. 推理延迟分解

根据这种执行模型，每令牌解码延迟可以分解为

$$T_{\text{token}} = T_{\text{ffn}} + T_{\text{attn}} + T_{\text{comm,exposed}}. \tag{2}$$

这里， T_{ffn} 表示前馈网络的延迟，T_{attn} 表示注意力计算的开销，包括键值缓存访问；而$T_{\text{comm, exposed}}$ 表示无法通过重叠隐藏的通信开销。

这种分解的两个性质至关重要。首先，稠密模型与混合专家模型的注意力项基本相同：无论前馈网络的结构如何，每个词元执行相同的注意力计算，并且必须访问相同数量的键和值。尽管注意力包含算术运算，但推理时的延迟通常由键值缓存流量主导。其次，虽然通信开销取决于并行化策略，但通常次于前馈网络的执行开销。因此，推理延迟的系统性差异主要源于 T_{ffn}。

2.3. 复用与带宽主导的前馈网络执行

前馈网络延迟由计算耗时和内存耗时中的较大者决定，

$$T_{\text{ffn}} = \max(T_{\text{ffn,comp}}, T_{\text{ffn,hbm}}). \tag{3}$$

在带宽敏感区间，内存项可近似表示为

$$T_{\text{ffn,hbm}} \approx \frac{W_{\text{ffn}}}{R\,\beta_{\text{hbm}}}, \tag{4}$$

其中， W_{ffn} 为前馈网络权重大小， β_{hbm} 为持续内存带宽，R 为复用因子。

对于处理 B_{dense} 个令牌的稠密前馈网络微批次，

$$R_{\text{dense}} \approx B_{\text{dense}}. \tag{5}$$

在专家路由下，复用程度随专家局部微批次大小变化。令 B_{moe} 表示 MoE 层跨所有专家处理的微批次总大小。那么，

$$R_{\text{moe}} \approx \frac{B_{\text{moe}}\,k}{E}. \tag{6}$$

关键在于，两种情况下推理时允许的微批次大小并不相同。由于所有专家权重都必须常驻于高带宽内存中，混合专家模型比质量相当的稠密模型消耗更多高带宽内存容量。可用于键值缓存的内存减少后，可行的最大微批次大小满足

$$B_{\text{moe}} < B_{\text{dense}}. \tag{7}$$

因此，在内存主导区间，混合专家模型会受到双重惩罚。首先，每位专家的复用程度会按路由因子 k/E 降低，其中 k 表示每个词元激活的专家数量。其次，用于分摊权重成本的基准批次大小本身也更小。综合这些影响，

$$R_{\text{moe}} \ll R_{\text{dense}} \tag{8}$$

在现实的推理约束下。

在此区间，混合专家模型每个词元执行的浮点运算次数较少，但前馈网络权重读取所分摊到的词元数也大幅减少，导致每个词元需要传输的前馈网络权重字节数增加，并使执行受带宽限制。我们将这一效应称为复用碎片化。因此，与单独考察稀疏性或浮点运算次数相比，复用因子能更准确地预测推理效率。

3. 混合专家中的稀疏性、复用与质量匹配的推理效率

本节将复用碎片化与质量匹配的比较相结合，推导出一个简单的判据，用于预测混合专家推理相较于稠密模型在结构上何时处于劣势。

3.1. 专家路由下的复用

混合专家模型通常以稀疏性来描述。若某一层包含 E 个专家，且每个词元选择其中 k 个，则每个词元激活的前馈神经网络参数占比为

$$s = \frac{k}{E}. \tag{9}$$

这一指标常用于分析训练时效率。

在推理时，稀疏性体现为每个专家处理的词元数量减少。对于包含 B 个词元的微批次，平均专家本地批次大小为

$$B_{\text{expert}} = Bs. \tag{10}$$

在第2节的执行模型下，这意味着专家路由下的前馈神经网络权重复用与s呈线性关系。因此，我们使用 s 作为混合专家推理中复用的紧凑参数化表示，而不是将其作为加速比的代理。

3.2. 质量匹配的推理比较

将混合专家模型与规模相匹配的稠密模型进行比较，并不能构成有意义的推理时比较。采用混合专家架构，是因为它们在固定训练浮点运算预算下能达到更高的准确率。因此，合适的推理基准应是输出质量与该混合专家模型相匹配的稠密模型。

我们使用质量等效因子q对这一概念进行形式化定义：质量相同的稠密模型具有如下有效前馈网络规模

$$W_{\text{dense}} = q\,W_{\text{active}}, \qquad W_{\text{active}} = s\,W_{\text{moe}}. \tag{11}$$

3.3. 缩放定律与质量等效乘数

稠密模型的参数数量与验证损失之间呈现出可预测的关系。经验缩放定律表明，损失会随着参数数量增加而按浅幂律下降（Kaplan等人，2020年）：

$$L(N) \propto N^{-\alpha_N}, \qquad \alpha_N \approx 0.076. \tag{12}$$

我们考虑在固定评估设置下训练至收敛的模型（不提前停止）。在此情形下，我们忽略任何加性不可约损失项；该项只会重新缩放推导出的乘数，而不会改变定性结论。

设规模为 N_{base} 的稠密基线达到验证损失L_{base}，而混合专家模型在匹配的训练计算量下达到损失 L_{MoE}。根据上述缩放定律，

$$\frac{L_{\text{base}}}{L_{\text{MoE}}} = \left(\frac{N_{\text{dense,qe}}}{N_{\text{base}}}\right)^{\alpha_N}, \tag{13}$$

由此可得

$$N_{\text{dense,qe}} = N_{\text{base}} \left(\frac{L_{\text{base}}}{L_{\text{MoE}}}\right)^{1/\alpha_N}. \tag{14}$$

比值$q \triangleq N_{\text{dense, qe}}/N_{\text{base}}$ 定义了一个质量等效稠密模型倍数。

3.4. 一个具体的参照：DeepSpeed-MoE

DeepSpeed-MoE 提供了一个清晰的计算量匹配比较，可据此推导出稠密模型等效倍数（Rajbhandari 等人，2022）。在计算量匹配的训练中，一个具有 13 亿激活参数并采用混合专家路由的模型，验证损失达到 2.12；而激活规模相同的稠密模型，损失为 2.34。代入这些数值得到

$$\frac{N_{\text{dense,qe}}}{N_{\text{base}}} = \left(\frac{2.34}{2.12}\right)^{1/0.076} \approx 3.7. \tag{15}$$

同样，一个具有3.5亿激活参数的混合专家模型取得了2.29的验证损失，而同等激活规模的稠密模型损失达到2.60。因此，

$$\frac{N_{\text{dense,qe}}}{N_{\text{base}}} = \left(\frac{2.60}{2.29}\right)^{1/0.076} \approx 5.3. \tag{16}$$

表1. 现代混合专家模型的代表性稀疏性 s、质量等效因子 q，以及隐含的 qs。

模型	E	k	$s = k/E$	q	qs
DeepSpeed-MoE	128	1	0.0078	$4-5$	0.03–0.04
GLaM	64	2	0.031	$4-6$	0.12–0.19
GShard	128	2	0.016	$5-6$	0.08–0.10
Switch-C	2048	1	4.9×10^{-4}	~ 3	0.0015
ST-MoE	64	2	0.031	~ 3	0.093

3.5. qs不等式

我们现在将稀疏性与质量等效缩放相结合，以得到混合专家推理效率的一个简单准则。

推理成本由每个词元所需传输的 FFN 权重字节数决定。令 W_{active} 表示 MoE 层中每个词元激活的 FFN 权重占用，其中$W_{\text{active}} = sW_{\text{moe}}$。因此，质量相同的稠密模型的有效前馈网络规模为 $W_{\text{dense, qe}} = q\,W_{\text{active}}$。

采用专家路由时，FFN 权重仅在专家本地批次内复用。对于包含 B 个词元的微批次，每个专家平均处理 $B_{\text{expert}} = Bs$ 个词元，因此激活的 FFN 权重可分摊到 Bs 个词元上，使每词元成本与 $W_{\text{active}}/(Bs)$ 成正比。相比之下，质量匹配的稠密模型将其 FFN 权重分摊到整个批次 B上，使每词元成本与 $q\,W_{\text{active}}/B$ 成正比。

对每词元的 FFN 权重流量取比值，可得

$$\frac{\text{MoE}}{\text{dense (equal quality)}} \approx \frac{1}{qs}. \tag{17}$$

这就得到了 qs 不等式：

$$qs < 1. \tag{18}$$

当 $qs < 1$时，混合专家模型每个词元需要传输的前馈网络权重字节数多于质量相同的稠密模型，因此在推理时面临前馈网络带宽劣势。表1 报告了根据先前混合专家模型推断出的代表性 q、s和 qs 值（Lepikhin等人，2020年；Du等人，2022；Fedus等人，2022；Zoph等人，2022），这些模型均满足 $qs < 1$。

4. 评估方法

本节介绍用于生成论文中所有吞吐量结果的解码阶段推理成本模型和评估方法。该框架在一致的系统和硬件假设下，应用于多个稠密模型和混合专家模型实例。

4.1. 解码阶段成本模型

我们使用一种每词元成本模型来评估解码阶段推理，该模型涵盖了延迟的主要影响因素

在自回归服务下。给定模型配置、系统配置和并行化策略，每词元延迟的计算方式为

$$T_{\text{token}} = \max(T_{\text{compute}}, T_{\text{hbm}}) + T_{\text{comm,exposed}}. \quad (19)$$

该公式与式（2）中的加性分解等价，但在执行瓶颈层面加以表述。计算项汇总了来自投影、注意力和前馈网络（FFN）执行的算术运算量，由闭式浮点运算次数经特定数据类型的峰值吞吐量和并行化因子（张量并行、专家并行和流水线并行）归一化后计算得到。HBM 项汇总了来自前馈网络权重读取和键值（KV）缓存访问的内存流量，再除以持续 HBM 带宽。由于这些组件在执行中存在重叠，每 token 延迟由计算或内存中较慢者决定，因此采用最大值公式。通信成本使用集合通信特定延迟和带宽项进行建模，并被划分为可重叠部分与显露组件。

权重复用被显式建模：稠密前馈网络在整个批次上摊销，而混合专家复用则按Bk/E缩放。该复用因子决定FFN执行是计算受限还是带宽受限。

4.2. 键值缓存建模与内存可行性

我们对键值缓存内存进行显式建模，因为在长上下文下，它占据每 GPU 内存的大部分，并直接决定最大可行批次大小。对于给定的模型和上下文长度，每层每 GPU 的键值缓存占用取决于 KV 布局（完整布局、分组查询注意力、多查询注意力或 MLA 式压缩），以及键值缓存如何分片到各个图形处理器上。

键值缓存并行策略通过沿注意力头或序列维度对缓存进行分区，降低每 GPU 的键值缓存占用。因此，每个图形处理器仅保留总键值缓存状态的一部分，其余部分则分布在并行组中。

内存可行性通过每 GPU 容量约束来保证，

$$\text{budget} = (\text{HBM}_{\text{cap}} - \text{reserve} - \text{misc}) \cdot (1 - \text{safety}), \quad (20)$$

其中已计入所选分区下的所有驻留权重。剩余预算决定每个图形处理器可并发处理的最大序列数，记为$n_{\text{eff, max}}$。所有吞吐量结果均在批次大小为$B = n_{\text{eff, max}}$时评估。

这种容量优先评估是我们比较的核心：在长上下文下，混合专家模型保留更大的驻留权重占用，从而为键值缓存留下更少的内存，因此只能容纳更小的可行批次，这直接降低了前馈神经网络权重复用。

4.3. 通信与重叠建模

通信成本采用环形公式对全归约、全收集和全对全集合通信进行建模，并按照标准集合通信成本模型，显式纳入启动延迟和带宽项（Thakur 等人，2005年；Hoefler 等人，2007年；Patarasuk 与 Yuan，2009）。集合通信类别包括张量并行、键值并行、专家并行和布局转换集合通信。

根据组大小和岛容量，将集合通信映射到节点内或节点间互连网络。岛容量定义为可完全通过节点本地高带宽互连网络（例如 NVSwitch 或同等域）参与集合通信的加速器最大数量。为估算显性通信，每种集合通信类型都与特定阶段的计算窗口（注意力、投影或前馈网络）相关联。只有落在这些窗口内的通信才能与计算或 HBM 流量重叠；其余部分均视为显性通信，并直接计入每 token 延迟。

对于 pass-Q 模式下的上下文并行，为捕捉环形调度固有的顺序传播效应，将集合通信中的环形尾部部分视为不可重叠。

4.4. 并行化搜索与自动调优

吞吐量取决于 KV 可行性、权重复用、集合通信选择和重叠之间的相互作用。因此，我们评估并行化策略的搜索空间，并为每个模型选出吞吐量最高的可行配置。

搜索空间涵盖张量并行、流水线并行、专家并行和数据并行，以及 KV 并行选项（无、KVP（Bhatia 等人，2025）、CP（Yang 等人，2025a））及其相应的消息数量。对于每个候选分区，我们计算可行性（$n_{\text{eff, max}}$）、每 token 延迟、每 GPU 每秒处理的 token 数，以及集群级每秒处理的 token 数。

对稠密模型和混合专家模型，采用相同的搜索空间、可行性约束和选择标准。

4.5. 参考实例化与敏感性

除非另有说明，所有结果均采用一个参考系统，该系统代表一款配备高带宽节点内互连（双向总带宽为 1.8TB/s）的典型高性能平台。为确保我们的结论不受底层系统假设的影响，我们还针对其他硬件和并行维度评估了敏感性，以确认观察到的吞吐量和延迟趋势

具有普适性。主要配置及所考察的这些维度总结于表 2。

我们评估了已发表的架构以及以下模型的混合专家配置：*LLaMA*（Touvron等人，2023）、*Grok-1*（xAI，2024b；a）、*Qwen3*（Yang等人，2025b）、*DeepSeek-V3*（DeepSeek-AI，2024）以及*Switch Transformer*（Switch-C 设置）（Fedus等人，2022）。对于每个模型，我们根据其已发表的规格匹配键值布局和路由参数，并评估上下文长度达到 $L = 16$M 个词元的解码阶段推理。

表 2。 用于评估的参考配置及所探索的维度。

类别	参考配置 tion	探索/变化项
模型	Grok-1、Qwen3-235B、DeepSeek-V3, Switch-C	质量匹配 稠密模型与混合专家模型
上下文	L = 131,072 个令牌	1K 至 16M 个令牌
硬件	180 GB HBM3e, 8.0 TB/s 高带宽内存带宽-宽度，4.5 PFLOPS FP8 / 2.25 PFLOPS BF16 密集计算	±25% 容量和带宽
互连	高带宽 节点内互连 nect	900 GB/s（主），448 GB/s（次级）（单向）
并行度	张量并行=8、专家并行=4、流水线并行=1、KVP=4	张量并行∈ {1, 2, 4, 8}、专家并行∈ {1, 2, 4, 8}、流水线并行∈ {1, 2, 4}，键值∈ {无，KVP，上下文并行}
通信	环形集合通信，阶段感知重叠	启动延迟（α），重叠比例（β）， 上下文并行环形通信尾部

5. 容量与复用结果

我们量化了长上下文下的容量受限批次大小，以及混合专家推理中由此产生的前馈神经网络权重复用。

5.1. 混合专家与稠密模型复用对比

我们将每个混合专家模型与质量匹配的稠密基线进行比较，以分离推理时的复用效应。对于一个总参数量为 X、包含 E 个专家并采用 top-k路由的混合专家模型，激活参数比例为 $s = k/E$。稠密基线定义为

$$X_{\text{dense}} = q\, s\, X, \qquad q = 5. \tag{21}$$

我们设置 $q = 5$，以反映在训练计算量相同的条件下，稠密模型要达到与混合专家相同的验证损失，所需活跃参数量在经验观察中存在的 4–6× 差距，并将其统一应用于所有配对比较。

在推理时，我们计算固定 HBM预算下可行的最大批次大小，分别记为 B_{moe} 和 B_{dense}，并将复用定义为

$$R_{\text{dense}} \approx B_{\text{dense}}, \qquad R_{\text{moe}} \approx B_{\text{moe}} \cdot s. \tag{22}$$

由此得到的复用差距可分解为

$$\frac{R_{\text{dense}}}{R_{\text{moe}}} = \underbrace{\frac{E}{k}}_{\text{routing}} \cdot \underbrace{\frac{B_{\text{dense}}}{B_{\text{moe}}}}_{\text{capacity}}, \tag{23}$$

这将专家路由导致的复用损失与批次容量降低引起的额外损失区分开来。

运行点。 我们报告在 32 个图形处理器（张量并行=8, KVP=4）和 64 个图形处理器（张量并行=8，KVP=8）下的成对结果。所有结果均使用 FP8 权重、BF16 键值存储和 128k上下文。评估方法和内存核算详见第 4节。

5.2. 路由主导复用损失；容量对其起调节作用

表3 报告了符合条件的批次大小、由此推得的复用率，以及公式（23）中的路由-容量分解。

首先，对于 Qwen3-235B-A22B 和 DeepSeek-V3 等细粒度混合专家模型，路由主导复用损失。在 64 个图形处理器下，仅路由因子 $\frac{E}{k}$ 就使 Qwen3 的复用率降低了 $16\times$，使 DeepSeek-V3 的复用率降低了 $32\times$，这表明专家碎片化是复用崩塌的主要原因。

其次，在长上下文下，容量效应会产生乘法叠加，但在这一规模下影响次要。对于 Qwen3，KV 头部空间减少会额外带来 $1.03\times$ 的惩罚；而对于 DeepSeek-V3，则会产生 $1.13\times$的影响。尽管这些惩罚小于路由效应，但它们会与稀疏性导致的复用损失系统性地叠加。

第三，在极端稀疏性区间，仅常驻专家权重就可能使固定集群规模下的混合专家推理无法实现。Switch-C-2048 在 64 个图形处理器上无法接纳哪怕一个 128k上下文序列，这表明，足够细粒度的专家划分可能会超出内存容量，无论路由效率如何。

粗粒度混合专家的运作方式有所不同。对于 Grok-1，激活比例满足 $qs > 1$（在 $q = 5$处），从而使混合专家获得轻微的容量优势（$B_{\text{dense}}/B_{\text{moe}} = 0.90$）。然而，我们预计粗粒度变体的质量差距将会缩小，因为其更大的专家粒度留给稠密模型用于补偿的未激活权重空间更少。在 $q \in \{2, 3\}$ 处的敏感性扫描表明，随着稠密基线在结构上变得更小（$qs < 1$），它重新夺回容量优势（$B_{\text{dense}}/B_{\text{moe}} = 1.24$，在 $q = 2$处）。关键在于，即便在这一最佳情形下，总复用差距仍接近 $5\times$，这证实专家路由碎片化仍然是混合专家推理的根本瓶颈。

表3. 混合专家模型与质量匹配的稠密基线模型之间的复用对比（128k上下文）。

模型	GPUs	q	(E, k)	s	qs	B_{moe}	B_{dense}	R_{moe}	R_{dense}	E/k	$\frac{B_{\text{稠密模型}}}{B_{\text{moe}}}$	$\frac{R_{\text{稠密模型}}}{R_{\text{moe}}}$
Qwen3-235B	64	5	(128, 8)	0.0625	0.313	86	89	5.38	89	16.0	1.03	16.48
DeepSeek-V3	32	5	(256, 8)	0.0313	0.156	100	130	3.12	130	32.0	1.30	41.60
DeepSeek-V3	64	5	(256, 8)	0.0313	0.156	244	267	7.63	267	32.0	1.13	36.16
Grok-1	64	5	(8, 2)	0.2500	1.250	127	114	31.75	114	4.0	0.90	3.60
Grok-1（敏感性）	64	3	(8, 2)	0.2500	0.750	127	143	31.75	143	4.0	1.13	4.52
Grok-1（敏感性）	64	2	(8, 2)	0.2500	0.500	127	158	31.75	158	4.0	1.24	4.96
Switch-C-2048	64	5	(2048, 1)	0.0005	0.002	OOM	—	—	—	2048	—	—

对集群规模的敏感性。随着集群规模缩小，容量压力增大，这是延迟优化部署的典型情形。对于 DeepSeek-V3，将集群从 64 个图形处理器减少到 32 个会提高容量因子 $\frac{B_{\text{dense}}}{B_{\text{moe}}}$，从$1.13\times$ 到 $1.30\times$，而路由稀疏性保持不变，从而使复用差距从 $36\times$ 扩大到 $42\times$。由于该项会乘以公式（23）中的路由因子，细粒度混合专家在较小集群上会承受不成比例的更大复用损失，而这恰恰是最需要低延迟的场景。

6. 吞吐量结果

本节报告了所评估模型在不同上下文长度下的解码阶段吞吐量，并将测得的吞吐量差距与前文确立的容量和复用效应联系起来。所有结果均采用 FP8 权重、BF16 键值存储，以及第 4 节所述的评估方法。为突出相对扩展性和性能下降情况，吞吐量以 1k 上下文长度下的混合专家基线为基准，以百分比表示。

6.1. 吞吐量与上下文长度

表4 报告了 DeepSeek-V3 在 64 个图形处理器上、不同代表性上下文长度下的解码阶段吞吐量。吞吐量以 1k 上下文长度下的混合专家基线为基准（100%），以百分比表示。

随着上下文长度增加，键值缓存的增长会降低两个变体可接受的批次大小，从而限制前馈神经网络复用，并提高键值流量的相对成本。在较短上下文（1k）下，两种模型都能容纳极大的批次大小。然而，即使在算术效率最佳的情况下，稠密基线的表现仍比混合专家变体高出 $2.1\times$（相对吞吐量达到 214%，而混合专家为 100%）。这一结果表明，浮点运算次数减少并不能决定推理速度。

这一差距在 16k 令牌处达到峰值，此时 Dense-5 配置具有 $5.3\times$ 吞吐量优势。在这一区间，两种模型在批次级前馈神经网络权重复用上的差异最为显著，而极端键值缓存限制尚未开始主导。

在长上下文（128k）下，随着键值流量主导执行，两种变体的吞吐量均下降。尽管差距略有收窄，Dense-5 仍保持 $4.5\times$ 优势，这与第5节中的复用分析一致。最后，在极端上下文（4096k–16384k）下，模型之间的架构差异被完全抵消。巨大的键值缓存占用迫使两种模型都退化为单序列执行。随着两种架构的前馈神经网络摊销消失，吞吐量降至 1% 以下并完全趋同。

在所有运行条件下，只要批次级复用达到一定程度，稠密基线的表现就优于混合专家变体。自动调优的分区选择会影响绝对吞吐量，但不会改变这一排序。

表 4. 64 个图形处理器上的解码吞吐量与上下文长度的关系。吞吐量以 1k 上下文长度时混合专家基线的百分比表示。

			相对吞吐量（%）		
上下文	$\mathbf{B}_{\text{moe}}$	$\mathbf{B}_{\text{dense}}$	MoE	Dense-5	加速比
1k	31349	34263	100	214	2.1×
16k	1959	2141	31	163	5.3×
32k	979	1070	18	88	5.0×
128k	244	267	5	23	4.5×
1024k	30	33	0.7	2.9	4.2×
4096k	7	8	0.6	0.7	1.3×
16384k	1	1	0.2	0.2	1.0×

6.2. 归因：为什么稠密模型更快

为解释第 6 节中的吞吐量差距，我们将每个 token 的解码时间分解为 HBM 访问、计算和显式通信。表 5 报告了在 64 个图形处理器上，具有代表性的短上下文长度和长上下文长度下的这一分解结果。所有成本均以相对单位表示，并以混合专家模型在 1k 上下文长度下的计算时间作为归一化基准。

短上下文（1k）。在短上下文下，两种方案都能容纳极大的批次。虽然混合专家相较于稠密基线具有 $10\times$ 算术运算优势（1 个相对单位对比 10 个），但这一理论优势完全被显性通

表 5. 逐 token 解码时间归因（64 个图形处理器）。成本以相对延迟单位报告，并以上下文长度为 1k 时的混合专家计算时间为基准进行归一化。

		相对延迟			
上下文	模型	高带宽内存	计算	通信	差距驱动因素
1k	MoE	3	1	**17**	通信（+17）
	稠密模型	1	10	0	—
128k	MoE	**433**	32	23	高带宽内存（+361）
	稠密模型	72	48	5	—

信所掩盖。通过全对全集合通信将词元分发给专家，其规模与极大的批次大小成正比，导致通信成本高达 17 个相对单位，是混合专家实际计算成本的 17 倍。由于稠密基线无需显性通信，且始终纯粹受计算限制，短上下文下的吞吐量差距完全由通信导致。

长上下文（128k）。随着键值缓存增长限制了批次大小，混合专家模型的执行明显受到高带宽内存限制：每个词元的标准化高带宽内存访问成本为 433 个单位，而稠密基线仅为 72 个单位。高达 361 个单位的高带宽内存访问差距几乎完全解释了吞吐量差异。计算和通信成本被大幅掩盖，这表明在长上下文场景中，浮点运算次数减少并不能转化为性能提升。

6.3. 超越 DeepSeek-V3 的泛化

DeepSeek-V3 的结果表明，长上下文推理的吞吐量取决于复用和内存流量，而非浮点运算次数。现在，我们将考察这一行为能否推广到其他大型混合专家模型。表6 报告了其他模型在 64 个图形处理器上使用 128k 上下文时的吞吐量。

细粒度混合专家：在长上下文中，稠密模型占据优势。对于 Qwen3-235B 等细粒度混合专家，质量匹配的 Dense-5 配置显著优于混合专家变体：尽管微批次大小几乎相同，仍实现了 $4.4\times$ 吞吐量优势，因为每个专家有 B_{moe}个片段。这与 DeepSeek-V3 的结果一致，并证实在长上下文中，路由导致的复用碎片化是影响推理性能的主要因素。每个词元的算术运算量减少，并不意味着吞吐量会更高，因为前馈网络执行一旦受到高带宽内存限制，便无法实现这一点。

粗粒度混合专家：差距越小，反映的稀疏性越弱。Grok-1 处于更粗粒度的稀疏性区间，并表现出更小的吞吐量差距。在名义质量匹配因子下，Dense-5 比混合专家变体快 $1.6\times$ 。这一缩小的差距并不表示混合专家效率有所提升，而是反映了较弱的稀疏性以及额外的容量余量，后者部分抵消了路由导致的复用损失，并且

表6。128k上下文下的吞吐量（64 个图形处理器），包括 Grok-1 对质量匹配因子 q 的敏感性。吞吐量以各模型对应的混合专家基线为参照，按百分比归一化。

			相对吞吐量（%）	
模型	$\mathbf{B}_{\text{moe}}$	$\mathbf{B}_{\text{dense}}$	MoE	稠密模型
Qwen3-235B	86	89	100	436
Grok-1 (q=5)	127	114	100	158
Grok-1 (q=3)	127	143	100	204
Grok-1 (q=2)	127	158	100	228
Switch-C-2048	不可行	–	–	–

由此产生的容量余量部分抵消了路由导致的复用损失。

对质量匹配的敏感性：较低的 q 会扩大 Grok-1 的稠密模型优势。为单独考察质量匹配的影响，我们使用较小的质量匹配因子 q 对 Grok-1 进行评估；对于粗粒度混合专家模型而言，这些取值更为合适。随着 q 降低，稠密模型基线可采用更大的批大小，其吞吐量优势也从 $1.6\times$ 扩大到 $2.0\times$ 和 $2.3\times$。这项敏感性研究表明，Grok-1 的混合专家模型与稠密模型性能看似接近，取决于 q的选择。在更切合实际的质量匹配条件下，稠密模型可能更具优势。

极端容量场景：混合专家模型越不可行，结论就越有力。Switch-C-2048 代表一种极端场景：在 128k 上下文下，仅常驻专家权重就超出可用高带宽内存，使得混合专家推理无法在此集群上进行。

这些结果表明，DeepSeek-V3 的表现并非孤立现象，而是现代混合专家模型中更广泛的一类代表，进一步说明长上下文下的推理效率从根本上受复用和内存限制。

7. 讨论

混合专家在推理时并非总是处于劣势。本节将阐明稀疏性何时有帮助、何时没有帮助，以及这对部署混合专家模型意味着什么。

稀疏性何时占优，何时不占优。当每个专家的子批次较大且执行仍保持计算受限时，混合专家可以具有竞争力。在实践中，当前系统受到显性通信的制约，如第 6.2 节在较短上下文长度下所示，而稠密模型仍保持计算受限。因此，稀疏性的浮点运算次数优势并未转化为更低的延迟。要在这一区间实现混合专家的潜在收益，需要网络协议栈取得进展，包括更低延迟的集

合通信，以及通信与计算之间更有效的重叠。

在另一个极端，当批大小缩小至接近 1 时，混合专家模型每个词元仅激活一小部分参数。在这种情况下，混合专家模型移动的权重字节数少于质量匹配的稠密基线，因此能够实现更高的吞吐量。然而，要出现这种情况，所需的上下文长度极长，约为数千万个词元，远超典型的部署场景。

将混合专家作为训练时的优化手段。虽然混合专家在训练期间能够显著提升效率，但一旦通信和内存流量成为主导因素，这些收益就无法稳定地延续到推理阶段。这表明一种实用的部署策略：主要在训练中使用混合专家，随后将其蒸馏为稠密模型用于推理，从而兼得训练期间稀疏性的优势与稠密执行的推理效率。

8. 结论

混合专家架构通过减少活跃浮点运算次数来提升训练效率，但推理性能取决于内存带宽和权重复用，而非算术吞吐量。我们通过复用原则将这一转变形式化，并表明专家路由会使批次碎片化、降低复用率并增加前馈权重流量；在长上下文场景中，键值缓存驻留会进一步放大这些影响。

我们提出了 qs 不等式，将其作为一条简洁的决策规则，用于预测混合专家相较于质量匹配的稠密模型在何时会处于结构性劣势。在各类前沿混合专家系统中，该规则准确预示了随着上下文长度增加，容量和吞吐量出现的观测性下降。总体而言，这些结果表明，在现实的服务约束下，训练时FLOP效率并不是推理效率的可靠代理指标。

商标声明

AMD、AMD Arrow 标志及其组合均为超威半导体公司的商标。本出版物中使用的其他产品名称仅用于识别目的，可能是其各自公司的商标。

参考资料

Bhatia、N.、More、A.、Borkar、R. 等。Helixparallelism：重新思考面向交互式数百万词元大语言模型解码的分片策略。*arXiv* 预印本 *arXiv:2507.07120*，2025年。

DeepSeek-AI. Deepseek-v3 技术报告。*arXiv* 预印本 *arXiv:2412.19437*，2024。doi: 10.48550/arXiv.2412.19437。Du, N.、Huang, Y.、Dai, A. M.、Tong, S.、Lepikhin, D.、Xu, Y.、Krikun, M.、Le, Q. V.、Wu, Y. 与 Chen, Z. Glam：利用混合专家实现语言模型的高效扩展。载于 Chaudhuri, K. 与 Jegelka, S.（编），第*39*届国际机器学习大会论文集（*ICML 2022*），第1–23页，巴尔的摩，马里兰州，2022。PMLR。Fedus, W.、Zoph, B. 与 Shazeer, N. Switch Transformers：以简单高效的稀疏性扩展至万亿参数模型。 机器学习研究期刊，23(120):1–39，2022。
U
R
Lhttp://jmlr.org/papers/v23/21-0998.html。

Hoefler、T.、Lumsdaine、A. 和 Rehm、W. 非阻塞集体操作的实现与性能分析。在*2007*年美国计算机协会/电气电子工程师学会超级计算会议论文集中，第 1–10 页，2007年。

Kaplan, J., McCandlish, S., Henighan, T., Brown, T. B., Chess, B., Child, R., Gray, S., Radford, A., Wu, J., 和 Amodei, D. 神经语言模型的缩放定律。*arXiv* 预印本 *arXiv:2001.08361*，2020年。网址：https://arxiv.org/abs/2001.08361。

Lepikhin, D., Lee, H., Xu, Y., Chen, D., Firat, O., Huang, Y., Krikun, M., Shazeer, N., 和 Chen, Z. GShard：通过条件计算和自动分片扩展巨型模型。载于 Larochelle, H. 和 Ranzato, M.（编），第*34*届神经信息处理系统会议论文集（*NeurIPS 2020*），第1–21页，加拿大温哥华（线上），2020年。NeurIPS。

Patarasuk, P. 和 Yuan, X. 面向工作站集群的带宽最优全归约算法。《并行与分布式计算杂志》，69(2):117–124，2009。

Rajbhandari, S., Li, C., Yao, Z., Zhang, M., Aminabadi, R. Y., Awan, A., Rasley, J. 和 He, Y. DeepSpeed-MoE：推进混合专家推理和训练，赋能下一代人工智能规模化发展。载于 Chaudhuri, K. 和 Jegelka, S.（编），第 *39* 届国际机器学习大会论文集（*ICML 2022*），第 1–26 页，巴尔的摩，马里兰州，2022。PMLR。

Shazeer, N., Mirhoseini, A., Maziarz, K., Davis, A., Le, Q. V., Hinton, G. 和 Dean, J. 超大规模神经网络：稀疏门控混合专家层。载于 Bengio, Y. 和 LeCun, Y.（编），第 *5* 届国际学习表征会议论文集（*ICLR 2017*），第 1–19 页，法国土伦，2017。OpenReview.net。

Thakur, R., Rabenseifner, R. 和 Gropp, W. 在 MPICH 中优化集合通信操作。《

国际高性能计算应用期刊，19(1):49–66，2005。Touvron, H., Lavril, T., Izacard, G., Martinet, X., Lachaux, M.-A., Lacroix, T., Rozi`ere, B., Goyal, N., Hambro, E., Azhar, F., Rodriguez, A., Joulin, A., Grave, E. 和 Lam- ple, G. Llama：开放且高效的基础语言模型。*arXiv* 预印本 *arXiv:2302.13971*，2023。doi: 10.48550/arXiv.2302.13971。xAI。xai-org/grok-1：Grok 开放发布（代码和加载示例）。GitHub 代码库，2024a。网址 https://github.com/xai-org/grok-1。访问日期：2026- 01-26。xAI。Grok-1 开放发布。xAI 新闻，2024b。网址https://x.ai/news/grok-os。访问日期：2026-01-26。

Yang, A., Yang, J., Ibrahim, A., Xie, X., Tang, B., Sizov, G., Park, J. 和 Huang, J. 面向可扩展百万词元推理的上下文并行。《机器学习与系统会议论文集》，7，2025a。Yang, A. 等。Qwen3 技术报告。*arXiv* 预印本 *arXiv:2505.09388*，2025b。doi: 10.48550/arXiv.2505. 09388。Zoph, B., Bello, I., Kumar, S., Du, N., Huang, Y., Dean, J., Shazeer, N. 和 Fedus, W. ST-MoE：设计高效的专家模型。*arXiv* 预印本 *arXiv:2202.08906*，2022。URL https://arxiv.org/abs/2202.08906。