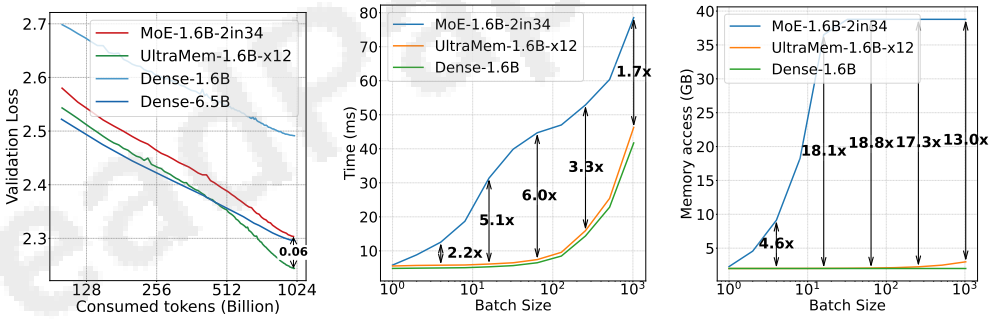
Ultra-Sparse内存网络

Zihao Huang,* **Qiyang Min**,* **Hongzhi Huang**,* **Defa Zhu, Yutao Zeng, Ran Guo, Xun Zhou**
Seed-Foundation-Model Team, ByteDance
{huangzihao.notabot,minqiyang,huanghongzhi.51,zhudefa,
yutao.zeng,guoran.94,zhouxun}@bytedance.com

ABSTRACT

人们普遍认为，变压器模型的性能与对数的参数数量和计算复杂性相关。尽管专家（MOE）的混合物（MOE）将参数计数从computation的复杂性中分解出来，但由于高度记忆成本，它们仍然面临推理的挑战。这项工作介绍了Ultramem，并结合了大规模的超平方内存层来解决这些局限性。我们的方法在保持模型性能的同时显着重新定义了推断潜伏期。我们还提出了这种新体系结构的缩放定律，这表明它不仅可以缩放出良好的缩放属性，而且表现优于MOE。在实验中，我们训练的the tharem thar ultramem有2000万个内存插槽。结果表明，我们的方法可以在高温计算预算中实现最先进的推理速度和模型性能，为数十亿个老虎机或专家铺平了道路。

1 简介



(a) Validation loss (b) Inference time (c) Memory access

图1：我们确保三个模型具有相同的计算，并且MOE和Ultramem具有相同的参数。X轴以对数尺度绘制。在（b）和（c）中，序列长度为1，因为在解码期间，我们只能一次预测一个令牌，而密钥/值的Cachelengths为2048。 A100-SXM-80GB。

由大语言模型（LLMS）驱动的自然语言处理（NLP）的最新进展（Radford等，2019; Brown，2020年）需要以指数级的计算资源来扩展，并在资源有限的环境中提出挑战，例如实时的挑战申请。蟾蜍计算问题，专家的混合物（MOE）（Fedus等，2022; Jiang等，2024）和生产密钥内存（PKM）（Lample等，2019）。 MOE选择性地激活参数，提高训练效率，但由于增加的内存访问而损害了推理时间。PKM保持一致的内存访问，其值嵌入较少，但其性能比MOE更差。

如图1（b）所示，尽管具有相同的计算成本和十二倍的参数，但与密集模型相同，但推理的运行速度慢2至6倍，但随批量大小而变化2至6倍。

* 同等贡献。

如图1（c）所示，这种放缓源于高内存访问需求，突出显示的推理方案效率低下。首要的挑战是如何匹配甚至超过MOE模型的效果，同时保持与DENSENS模型相当的内存访问水平。

在本文中，我们介绍了Ultramem，这是一种建立并扩展PKM概念的体系结构。Ultramem结合了大规模的超平方内存层，可显着提高计算效率，并降低推理潜伏期，同时保持各种基准的模型经济性。该体系结构不仅支持在资源受限环境中的高效语言模型的部署，而且还开辟了新的途径，强制构建甚至更大的模型，而没有以前相关的过高的成本。

总之，我们做出以下贡献：

1。与PKM相比，超大型的增强大大增强，并且在相同的规模上胜过MOE。超过PKM，Ultramem确实拥有在广泛的计算资源上训练大规模mod-els的先决条件，并且已经进行了全面的实验性验证。

2。与MOE相比，在推断期间的内存访问成本明显降低。在共同的推理批次尺寸下，使用SameParameter和计算的MOE的速度可能比MOE快6倍。Ultramem的推理速度几乎与具有等效计算资源的密集模型相同。

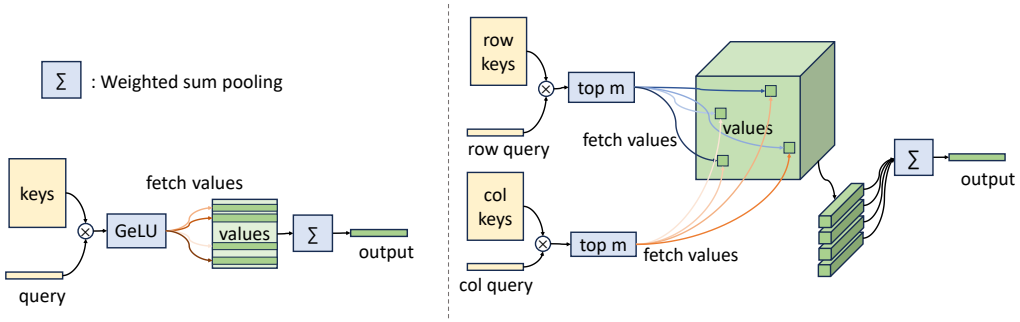
3。我们已经验证了Ultramem的缩放能力。与MOE相似，Ultramem具有强大的能力，并且我们观察到比MOE更强的缩放能力。

2 相关工作

专家的混合物。 Shazeer等。（2017年）提出了Moe和Fedus等。（2022）在大型语言模型中引入了主题，每个令牌每次都会选择一个专家进行推理，从而在不增加计算的情况下对模型参数进行了限制。 Rajbhandari等。（2022）介绍了共享专家的概念，每个代币都利用一些固定的专家以及一些独立的专家。随后的研究重点是改善MOE的门控功能，包括Tokenchoice（Chi等，2022），不可训练的令牌选择（Roller等，2021）和专家选择（Zhouet Al。，2022年），主要解决问题专家失衡的问题。刘等。（2024）; Dai等。（2024）选择将专家切成较小的细分市场，同时激活每个令牌，实现重要的绩效改进的专家。并发研究（Krajewski et al。，2024）仔细研究了粒度的好处和增加专家的数量，并研究了与MOE相关的标准定律。在本文中，我们使用细粒度的MOE作为基线，其中MOE的粒度设置为2。这意味着每个专家的大小是原始Multilayer Perceptron（MLP）的一半，并且两个专家都激活了每个令牌。

大记忆层。 Lample等。（2019年）首先介绍了称为PKM的大型内存层的概念，可以看作将MOE专家切成最小的配置。 Kim＆Jung（2020）提出了一个类似于MOE的共享专家的概念，允许PKM和MLP并行。 Csordás等。（2023）通过删除软轴轴承对PKM进行了轻微的修改。 Peer（HE，2024）改善了PKM中值的激活，以激活小型专家的内在维度为1，从而实现了显着的性能增长。但是，当前对PKMIS的研究仅限于较小的模型，甚至最新的改进版本的PKM仅超过了MOE的情况。另外，当前的PKM不具有适合大型渗透的特征。我们在本文中解决了这些问题。

张量分解将张量分解为一系列小矩阵或张量。在深度学习研究中，这种方法通常用于在训练过程中近似大张量，以节省计算和参数。产品量化（Jegou等，2010）将AVECTOR分解为多个子向量，从而使我们能够使用较小的子向量的数字重建原始矢量，从而减少了模型参数。 Bershatsky等。（2024）在微调阶段初始化了几种术语和一个核心张量，并在训练结束时以塔克分解方式训练这些参数，并以塔克分解的方式重建理论大张量。我们借用这种见识来改善PKM的主要检索。



(a) Multilayer perceptron (b) Large memory layer

图2：多层感知器（MLP）和大存储层（LML）的概述。为了简洁起见，我们省略了内存层的第三个顶级M操作。 MLP通常由两层层和凝胶激活组成。我们将第一个线性层的权重视为键，第二个线性层的特权作为值。 LML使用行和列键来确定索引存储器值的2-Doogical地址，而MLP使用1-D逻辑地址。基于分数较高的指标，“提取值”referstо检索值。

3 ULTRAMEM

3.1 PRELIMINARY

在这里，我们首先根据产品密钥介绍原点大存储层（LML），这是我们提出的方法的基础。首先在先前的工作中探索了基于产品密钥的内存层（PKM）的概念（Lample等，2019）。在他们的方法中，作者将Anexternal存储器模块纳入了语言模型，目的是扩展模型的参数，同时保持相似的计算复杂度。图2（b）中的整体结构图。

存储层通常由两个部分组成：密钥k∈Rn×d k和值v∈Rn×dv。从内存值中，查询向量q∈Rd k从数字键来获得分数来找到最相关的值。得分越高，如果值的影响越好。结果，该过程可以表达为：

$$\mathbf{s} = \sigma(\mathbf{K}\mathbf{q}) \quad \mathbf{o} = \mathbf{V}^{\top}\mathbf{s}, \tag{1}$$

其中s是分数，σ是非线性激活，o是输出。记住世界知识的纪念性内容的注意层，以及MLP层，也遵循上述配方σ在注意力层中的软态和MLP层中的gelu（Geva等，2020）（参见图2（a））。

产品键内存层用n> 10 6扩大内存大小，同时仅激活具有顶部M分数的几个值。在这里，M是控制稀疏性的高参数。尽管值访问了值，但必须完全计算出与值一样大的键，以在公式1之后获得TOP-M激活之前获得得分。为了减轻键的计算复杂性，提出了产品键。借用产品量化的想法，它使用了2D逻辑address（见图2（b）），通常是√n×√n网格，基于内存值检索。特别是，使用2-D逻辑地址（i，j）在物理地址n×i + j.j.j.with antrix中用来索引内存值，然后将其表示为矩阵进一步分解为对行和列分数的麻烦：

$$\mathbf{s}_{row} = \sigma_{\text{TopM}}(\mathbf{K}_{row}q_{row}(\mathbf{x})), \quad \mathbf{s}_{col} = \sigma_{\text{TopM}}(\mathbf{K}_{col}q_{col}(\mathbf{x})), \tag{2}$$

$$\text{s grid =σtopm（s row + sTcol），o = vT×softmax（vec（s grid））， (3)} \quad \texttt{oftMax}(\texttt{vec}(\mathbf{S}_{grid})), \tag{3}$$

其中k行，kcol∈Rn×d k，q行，q col：r di→r d k转换输入隐藏的x∈Rdi to row andcolumn查询，σtopm（·）保留了输入中最大的最大元素并将其余的设置为负数，并且具有无与伦比的矩阵形状的矩阵添加是通过元素广播实现的。应注意，从等式2中删除σtopm不会有任何区别。唯一的

将TOP-M应用于行和列得分的原因是减少S网格上Thelast Top-M操作的计算。作为S行，S col仅具有M激活分数，S网格仅具有M 2个固定剂，而不是n，而不是n，即TOP-M复杂性也从O（n log M）降低（（（√n + m 2））log M）。

请注意，S网格经历了类似于自我发病机构中使用的网格。此外，PKM从自我发项模块中采用了多头机制，其中它利用多个密钥集来检索共享值，我们将h表示为pkmheads的数量。

3.2结构改进

用一袋技巧来改善PKM。我们首先研究了PKM的结构，发现一系列的Minor调整可以稳步改善该模型的性能：

1）我们在方程3中删除了SoftMax的操作，该操作在研究中是良好的（Shen等，2023;Csordás等，2023）。2）我们进行层归一化（LN）（BA等人）。我们发现，逐渐衰减的学习率提供了进一步的好处。4）PKM使用线性层来生成查询，我们在此线性层之前添加了一个因果关系深度卷积的卷积Allealayer（Howard，2017）来增强查询5），以增强查询5）（5）与群体查询相似（Ainslie等人，2023年），我们以两个关键集共享查询。这可以将查询产生的计算成本降低一半，而deformanceImpact。6）通过使D V减半，我们将值的数量增加一倍。在保持催生值参数不变的条件下，我们增加了激活值的多样性，并进一步改善了模型效应。为了使输出与HidDendimension一致，我们在聚合的输出上添加了线性层。

Ultramem总体结构。然后，我们对模型结构和普罗旺斯超专有人进行了更深入的研究。图3显示了基于Apre-Layernorm变形金刚结构的PKM和我们改进的超系统结构。 PKM在带有内存层的更深层中替换MLP或并行操作（Kim＆Jung，2020）。我们注意到PKM的三个缺点：

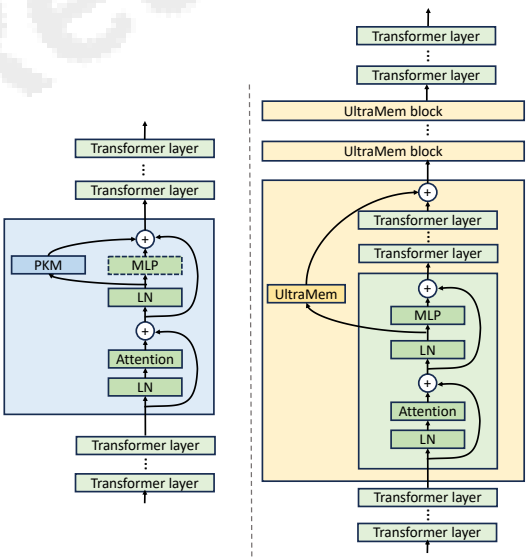


图3：PKM和Ultramem的整体。

1。随着值n显着增加，查询很难找到正确的值。

2。产品键分解引入检索拓扑。例如，让（i，j）成为TOP-1分数的逻辑地址，那么Top-2得分必须是位置I或J列J，这大大限制了Top-M选择的多样性。

3。有不平衡的多GPU计算和通信进行大规模参数训练存在问题，因为无法在单个GPU上放置完整的模型参数。

为了减轻问题1和3，我们将此级记忆层分解为多个以固定间隔分布在ThetransFormer层上的较小的Memory层。此外，这种跳过层的结构使我们能够重叠效果层和变压器层的执行，因为其中的层主要是内存的训练。

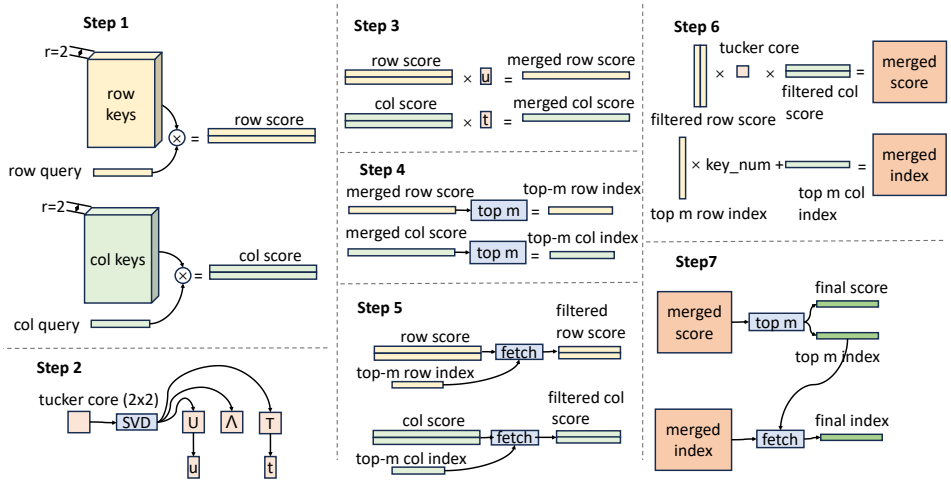


图4：塔克分解的查询键检索（TDQKR）的流量，此处r = 2。“获取”一词是指根据给定索引检索分数的作用量化是一种更精确的检索模块，用于召回超级的价值固定。 TDQKR过程的每个步骤都在维护文章中精心引用以供理解。

塔克分解查询键检索（TDQKR）。我们探索了一个更复杂的多重分析，以减轻问题1和2，其中塔克分解（Malik＆Becker，2018年）代替了产品量化。 TDQKR的整个过程如图4所示。特别是，Tucker分解估算了rank-R矩阵乘法的网格得分：

$$\mathbf{S}_{row} = \mathbf{K}_{row} q_{row}(\mathbf{x}), \qquad \mathbf{S}_{col} = \mathbf{K}_{col} q_{col}(\mathbf{x}), \tag{4}$$

$$\mathbf{S}_{grid} = \sigma_{\text{TopM}}(\mathbf{S}_{row}^{\top} \times \mathbf{C} \times \mathbf{S}_{col}), \tag{5}$$

其中s行，scol∈Rr×n和c∈Rr×r是Tucker Core，这是一个使用兰多姆初始化的可学习参数。为了产生N×R形的行和列得分，Thequery和Key的尺寸被重塑，导致K行，Kcol∈Rr×N×（D K /R）和Q行，Qcol∈Rr×（ D k /r），对应于图4步骤1。

但是，方程5在实践中直接应用，因为Top-M操作无法使用等效的两相顶级技术（如产品量化）来简化。作为备受关系，我们提出了一种近似的Top-M算法来解决此问题。关键是塔克芯的dorank-1近似，因此可以通过以下方式近似：

$$\mathbf{C} \approx \mathbf{u}\mathbf{t}^{\top}, \quad \sigma_{\text{TopM}}(\mathbf{S}_{row}^{\top} \times \mathbf{C} \times \mathbf{S}_{col}) \approx \sigma_{\text{TopM}}((\mathbf{u}^{\top}\mathbf{S}_{row})^{\top} \times (\mathbf{t}^{\top}\mathbf{S}_{col})) \tag{6}$$

u，t∈Rr×1。请注意，（u s s row），（tTs col）∈R1×n是行矢量，然后两阶段的顶部M技术与近似的Objectiveσtopm有关TTs col）），对应图4步骤3。总的来说，我们在行和列得分上进行了近似的TOP-M，过滤Outnon-Top元素，然后我们在S Grid上操作的最终顶部M中使用混凝土物镜， SteelIndex得分精确：

$$\tilde{\mathbf{S}}_{col} = \mathbb{I}_{\text{TopM}}(\mathbf{t}^{\top}\mathbf{S}_{col}) \odot \mathbf{S}_{col} \tag{9}$$

c≈uutT（7）srow = i topm（u s row）⊙s行（8）

s grid =σtopm（s⊺row×c×scol），（10）i topm（·）是二进制值函数，将top-m元素转换为1，否则将与图4＆5相对应的等式8＆9，以及对应于图4 Step6＆7的方程式10。至于等级1的近似，我们利用奇异值分解（SVD）（ABDI，2007年）用u，将tucker核心分解为与图4步骤2相对应的theleading奇异值的左和右奇异向量。

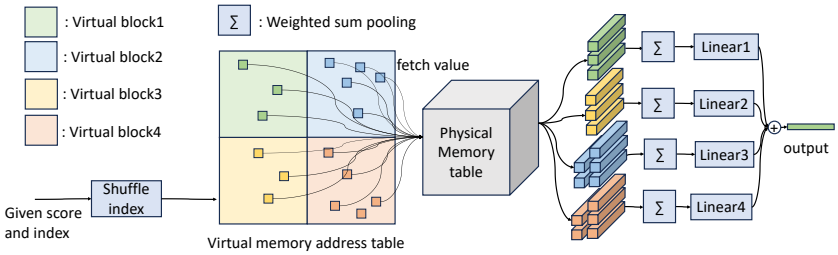


图5：隐式值扩展（IVE）的流，此处e = 4，m = 16。ive通过虚拟扩展内存表来减少内存量并扩大内存大小。每个虚拟块都是物理内存表的参数化。每个虚拟内存地址都对应于分支机构内存地址和投影仪索引。加权和池由虚拟块分组，然后是线性层以产生最终输出。

最后但并非最不重要的一点是，当非最大最大值与最大值一样大时，应考虑近似误差。为了减轻这种情况，通过约束非最大最大特征值在培训期间引入了管理近似值的辅助损失：

$$\text{C =UλTT，（通过SVD）（11）} \tag{11}$$

$$\mathcal{L}_{aux} = \frac{\alpha}{r-1}\sum_{i=2}^{r}\left(\max\left(0,\lambda_i-\tau\right)\right)^2, \tag{12}$$

在此，λ表示C的奇异值以降序为单位，τ是防止C的边缘，以防止C退化为Rank-1矩阵，而α是损耗的系数。

隐式价值扩展（IVE）。尽管使用了稀疏，但由于大量的内存访问，在训练期间保持较大的存储器表仍然是固定的。为了减少内存访问并扩大内存大小，我们将虚拟内存作为隐式值扩展。给定的Avirtual膨胀率E> 1，虚拟内存将内存表扩展到E时间大小。我们将虚拟的内存设计为原始内存值V的多个重新聚体化，我们表示该记忆值。然后，使用E线性投影仪{w p |p∈[1，e]，wp∈Rdv×d'v}，可以将与pth reparamemeterization相对应的和Virtual Memory Blockṽ P定义为：

$$\tilde{\mathbf{V}}_p = \mathbf{V}\mathbf{W}_p. \tag{13}$$

然后，整个虚拟内存是虚拟块的串联ṽ= [ṽT0，ṽT1，。。。，ṽTe]T。注意虚拟值d'v的维度不一定与物理价值d v的维度一致。

$$,\tilde{\mathbf{V}}_1^\top,\ldots,\tilde{\mathbf{V}}_E^\top]^\top.$$

nsion of physical

要应用虚拟内存是有利的，可以将内存表从V替换为ṽ。为了符合虚拟内存大小，密钥大小由√时间。此外，我们建议使用虚拟内存的随机归因，以消除Rowand列评分提出的一些不必要的索引拓扑。具体而言，如果虚拟内存表通过串联结合，则每个记忆值及其扩展将位于同一列中，因此可以同时更频繁地选择。

虚拟内存的天真重新聚集化仍然引入了许多计算，即E·n·D V·D'V和E Times GPU内存访问。一个更好的主意是计算重新聚集量。也就是说，我们将逻辑地址扩展到三胞胎（i，j，p），其中（i，j）是原始的逻辑address，p是虚拟内存块的索引，然后同时进行sum pliming pliming和计算虚拟内存值。因此，等式3被重写为：

$$\hat{\mathbf{s}} = \texttt{Shuffle}(\texttt{vec}(\mathbf{S}_{grid})), \tag{14}$$

$$\mathbf{o} = \tilde{\mathbf{V}}^\top \times \hat{\mathbf{s}} = \sum_p \tilde{\mathbf{V}}_p^\top \times \hat{\mathbf{s}}_p = \sum_p \mathbf{W}_p^\top\left(\mathbf{V}^\top \times \hat{\mathbf{s}}_p\right) \tag{15}$$

其中ŝP表示对应于p-th虚拟内存块的得分。使用等式15，我们可以根据虚拟块索引首先查找和池值，然后转换减少的

物理值直接分为降低虚拟值。此技巧将额外的计算降低frome·n·d v·d'v至e·b·d v·d'v，其中b是批处理中的令牌数量，除线性投影仪外，几乎没有noextra gpu内存访问。图5显示了IVE的流动。

多核分数（MCS）。 PKM对于每个值在维度D V上共享一个分数。从经验上，将多个分数分配给单个值以提高性能。因此，我们将Tucker Core C作为一系列组成核C =重写我c（i）。这允许使用{c（i）} hi = 1生成单个分数映射s aps $\mathbf{S}^{(i)}_{\text{tucker = s row c（i）s col。明显地，}}{}^{(i)}$ Obviously,

$$\text{s tucker =s⊤row（}_{w}(\sum_{i}^{h}\mathbf{C}^{(i)})\mathbf{S}_{col}=\sum_{i}^{h}\mathbf{S}_{row}^{\top}\mathbf{C}^{(i)}\mathbf{S}_{col}=\sum_{i}^{h}\mathbf{S}^{(i)} \tag{16}$$

我们将TOP-M在总分S Tucker上进行，同时应用单个分数s Tucker Ondrever分开值表V = [V（1），。。。，V（h），] v（i）∈Rn×（dv/h），即 $^{(i)}_{tucker}$ on

$$\mathbf{o}=[\hat{\mathbf{s}}^{(1)\top}\mathbf{V}^{(1)},\ldots,\hat{\mathbf{s}}^{(h)\top}\mathbf{V}^{(h)}]^{\top}. \tag{17}$$

当此技术与IVE合并时，我们将物理内存值而不是虚拟记忆值分开，以使等价在等式15中。

改进的初始化。 PKM用高斯分布n（0，1dv）初始化值。由于PKM将SoftMax应用于分数，因此汇总输出的方差为1/D V。我们认为，LMLSH应该被视为类似于MLP的组件，因此应使用类似于MLP的初始化方法。在训练之前，MLP的输出通常遵循高斯分布n（0，12L）（Brown，2020），其中L是层的总数。我们初始化navith n（0，e2MHl），其中m是激活的值编号，h是头号，e是值的expexpansion时间。为了确保Ultramem的输出分布为n（0，12L），我们需要确认Top-M得分的平均值为1，详细信息请参见附录A。

4定量分析为什么超级而不是MOE

增强模型容量而不显着提高计算本体的最有效方法是MOE。该策略采用了一套专门的子模型，称为“专家”，它们共同解决了解决复杂问题的问题。但是，MOE模型对推理程序构成了挑战。

将变压器隐藏的维度视为d，MLP的内部维度为4D，给定批量批次大小为B。将MOE与2inn MoE一起使用（选择2个MOE专家 /令牌）作为Anexample，其中专家的内部维度为2D 。假设选择的专家是完全平衡的，我们可以将单萌层的内存访问作为min（2b，n moe）×2d 2。对于Ultramem，假设值尺寸为D/2，并且每个令牌都激活了TOP-M值，则其内存访问最小值（BM，N）×D/2。随着批处理大小的增加，MOE的内存访问迅速增长，直到需要访问所有专家参数的上限。相比之下，Ultramem的记忆进程非常缓慢，只有当批处理大小为数千万时，才与MOE达到平等。但是，在推论方案中，批处理大小通常不是很大。

图1显示了带有2in34 MOE和×12 1 Ultramem的16亿参数变压器的推理时间和内存访问。有关较大的批量尺寸，请参见附录中的图7。比较tomoe，Ultramem在批次大小为64的最大加速度达到了最大加速度，并且在其他批次尺寸下也显示出显着的加速度。

5个实验

在本节中，我们演示了Ultramem的缩放功能，表明它的表现优于Moe。我们还展示了Ultramem的性能如何随不同的TOP-M值而变化，并且参数数量是如何变化的，并进行了消融研究以测量每个部分的占影响。

1 Ultramem中参数的数量是密集层中参数数的12倍。在此案件中，超量的总参数和总计算与2in34 MOE相同。this

5.1 设置

数据集。培训数据来自Redpajama（计算机，2023年），其中包含1万亿代币。Redpajama代表Llama（Touvron等，2023）数据集的清洁室，完全开放的源版本。验证数据包括C4验证集（Raffel等，2020），该集源自CommonCrawl Web语料库。 C4训练集还将其纳入Redpajama培训数据中。

Tokenizer基于GPT-Neox（Black等，2022）令牌，该代币仪使用字节式编码（BPE）（Sennrich等，2015）算法，词汇大小为50,432。

评估。我们对十个基准数据集的所有模型进行了全面的评估。这些数据集包括MMLU，Trivia-QA，GPQA和ARC，以评估模型的知识性知识； BBH，BOOLQ，HELLASWAG和WINOGRANDE评估推理技能；降低Fortesting阅读理解能力；和测量整体模型性能的Agieval。解码超参数与Llama3的分解超标剂对齐（Dubey等，2024）。划分seeapendix E.

培训细节。我们使用旋转嵌入式（Su等，2024）的标准预符号变压器（Xiong et al。，2020）用于我们的致密模型，该模型具有151m，680m，1.6b和6.5b参数2。稀疏模型，包括Ultramem，PKM和MoE，我们扩展了稀疏参数，从151m，680m和1.6B密集的模型中扩展了十二个。在MOE模型中，使用平衡损失（Fedus等，2022）的重量为0.01，以确保COVERTECTERCTION，两名专家被激活（Jiang et al。，2024），使用平衡损失（Fedus等，2022）。我们略微增加了MOE专家的宽度，以匹配Ultramem的计算和参数成本。在Ultramem模型中，辅助减肥重量为α= 0.001，边缘= 0.15。值的学习率是其他参数的十倍，并且线性衰减。有关模型结构和超参数的详细信息，请参见附录E，以及大规模培训优化，Seeapendix C，D。

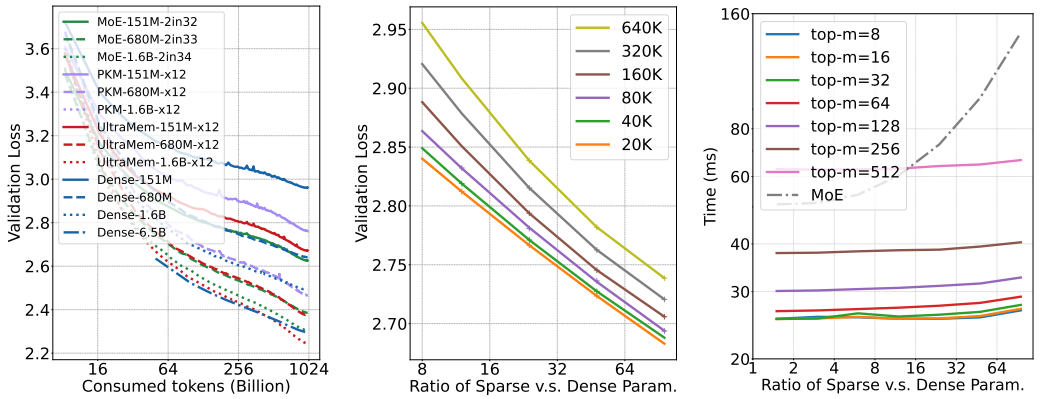
5.2语言建模数据集评估

我们评估了各种尺寸的模型，结果显示在表1 3中，其中插槽是单代币的计算成本，在附录中的图11中提供了显示训练过程中显示变化的曲线。我们观察到，随着模型容量的增加，Ultramem的表现可以超过相同的参数和计算。在1.6B密度模型上，具有12倍的超模型，参数可以匹配6.5B密度模型的性能。

表1：各种模型的性能指标。

Model	Param (B)	FLOPs (G)	Val. loss↓	GPQA↑	TriviaQA↑	BBH cot↑	Hella Swag↑	Wino Grande↑	DROP↑	Avg↑
Dense-151M	0.15	0.30	2.96	19.98	12.67	22.57	35.07	52.49	13.60	26.06
PKM-151M-x12	2.04	0.35	2.76	17.30	24.66	23.14	42.25	51.38	13.10	28.64
MoE-151M-2in32	2.04	0.35	2.63	17.30	33.27	23.24	48.44	55.96	18.57	**33.20**
UltraMem-151M-x12	2.03	0.35	2.67	19.42	28.97	22.65	43.96	50.83	14.08	29.99
Dense-680M	0.68	1.36	2.64	21.09	27.16	24.65	48.83	54.93	22.97	33.27
PKM-680M-x12	8.95	1.50	2.46	20.65	46.31	26.97	57.32	61.72	25.20	39.70
MoE-680M-2in33	8.95	1.50	2.39	20.54	34.19	26.63	62.71	59.98	26.54	38.43
UltraMem-680M-x12	8.93	1.49	2.37	21.99	55.17	26.62	64.15	60.54	25.14	**42.27**
Dense-1.6B	1.61	3.21	2.49	21.76	39.65	26.41	58.6	61.72	22.63	38.46
PKM-1.6B-x12	21.13	3.48	2.34	22.99	48.92	28.98	65.45	63.93	27.55	42.97
MoE-1.6B-2in34	21.36	3.52	2.30	21.32	59.56	29.46	67.34	63.93	28.81	45.07
UltraMem-1.6B-x12	21.41	3.50	2.24	24.66	66.38	30.63	71.52	66.38	29.99	**48.26**
Dense-6.5B	6.44	12.88	2.30	19.98	57.28	31.14	69.73	65.9	33.12	46.19

2不包括令牌词汇嵌入和预测头参数。3此表仅包括评估结果，其中指标随训练而稳步增加。有关Allresults，请参见附录中的表7。 all



(a) Scaling validation loss (b) Loss across varying sparsity (c) Speed across varying sparsity

图6：（a）。 C4验证以不同规模的不同模型的损失。 （b）。具有151m激活参数的缩放曲线。每行代表相同的模型稀疏性；例如，20个识别，将激活每20,000个值中约有一个。随着稀疏参数呈指数增加，损耗逐渐减少。 （c）。 Ultramem和Moewith 1.6B激活参数的推理时间。批处理大小为512，序列长度为1，密钥/值Cachelength是2048。具有固定的激活参数，Ultramem的推理时间几乎保持了恒定的稀疏参数，而MOE的推理时间则显着增加。

5.3值编号和顶部M

在大多数稀疏的LLM（例如MOE和Ultramem）中，在Weensparsity和模型性能上存在明显的正相关。因此，在本节中，我们进行了一系列的缩放实验，通过改变所选的TOP-M和值编号，即稀疏模块的参数，以验证相对于稀疏性的模型性能中的Thechanges。结果如图6（b）所示。

显然，在相同水平的稀疏度下，验证损失随着参数的数量而减少，并且可以保持一定程度的下降。此外，稀疏性越小，即激活参数的比例越大，模型性能越好。但是，此也导致更高的内存访问开销。因此，内存访问和缩放效率之间存在权衡。在最终实验中，我们选择了80K的稀疏性比为TheDefault模型配置。

随着稀疏参数的增加，图6（c）表明，只要激活参数（TOP-M）保持恒定，Ultramem仍保持稳定的推理时间范围的指数增长。不可比，MOE的推理时间在相似的条件下显着上升。此外，图1（b）表明，较小的批量大小，MoE的推理速度降低了更大的比较。

5.4消融

我们基于151m密集的模型进行全面的消融研究。在基线中，ThePKM是与MLP并行运行的版本，使其成为更强的基线。对于此组实验，学习率（LR）设置为1.2E-4，并在500B令牌上进行培训，并评估训练和C4验证集的跨熵损失。我们确保模型的最终版本的参数计数和计算成本本质上是相同级别的。

表2显示了消融结果。我们确定了6个大大改善绩效的变化：

1。将值的数量加倍，同时使其尺寸减半，并同时加倍top-m选择以保持主动参数一致。2。在变压器层上均匀地将单个Ultramem分为多个较小的单元，输出跳过了几个块。此布置将总参数计数，计算成本和稀疏参数激活在或以下级别的级别级别。

表2：改进模型的消融研究

	Train Loss ↓	Valid. Loss ↓	Dense Param.(M)	Sparse Param.(G)	FLOPs (M)
PKM-151M-x10	2.604	2.828	173.01	1.534	346.06
+rm softmax	2.570 -0.034	2.822 -0.006	173.01	1.534	346.06
+half vdim+proj	2.556 -0.014	2.800 -0.022	178.47	1.529	356.98
+share query	2.560 +0.004	2.803 +0.003	173.46	1.529	346.96
+split big mem&skip	2.554 -0.006	2.788 -0.015	161.64	1.536	323.32
+query/key LN	2.553 -0.001	2.789 +0.001	161.64	1.536	323.54
+IVE	2.544 -0.009	2.772 -0.017	172.37	1.536	344.98
+TDQKR	2.538 -0.006	2.764 -0.008	172.37	1.536	344.98
+MCS	2.521 -0.017	2.761 -0.003	172.37	1.536	344.98
+improved init	2.518 -0.003	2.758 -0.003	172.37	1.536	344.98
+value lr decay	2.494 -0.024	2.736 -0.022	172.37	1.536	344.98
+query conv	2.493 -0.001	2.736 -0.000	172.38	1.536	345.02
Total Diff	**-0.111**	**-0.092**	**-0.64**	**+0.002**	**-1.04**

3。塔克分解查询 - 键检索引入了可忽略的其他参数，同时减少了计算，此处r = 2。

4。多核分数显着减少训练损失，并稍微减少验证损失，此处h = 2。

5。隐式值略微增加了参数计数和计算量，但改进很大，在这里e = 4。

6。值参数的LR从其他参数和线性结构的十倍开始，以在训练结束时匹配它们。

除其他更改外，共享查询有助于削减计算成本，而表现较小。如图10所示，使查询/密钥的标准化大大减少了训练困惑和增强稳定性的峰值。（a）。改进的初始化可防止在图10.（b）和（c）中详述的早期至中间训练阶段的得分和输出方差。另外，采用卷积进一步限制了超专有输出中的方差差异（图10.（c））。上面的结果基于增量消融。可以在附录中的表8中找到独立消融的结果，并且与我们的期望保持一致。

除此之外，我们还对IVE，TDQKR和MC进行了另一项消融研究，具有不同的配置，这在表3中记录在表3中。对于IVE，随着E的增加，EVER的增长始终如一地改善了InModel的性能，以及计算成本的显着提高。但是，边际振作随着E的升高而导致我们推荐E = 4。对于TDQKR和MCS，增加R和HDO并没有显着改变计算负载，但是有效性不再显示标记的趋势，因此我们建议使用R = 2和H H = 2和H = 2。

表3：在IVE，TDQKR和MC上消融不同的配置

	IVE				TDQKR				MCS			
	Baseline	E=4	E=9	E=16	Baseline	r=2	r=3	r=4	Baseline	h=2	h=4	h=8
Training loss↓	2.553	-0.009	-0.016	-0.019	2.544	-0.006	-0.0065	-0.0063	2.538	-0.017	-0.017	-0.012
Validation loss↓	2.789	-0.017	-0.025	-0.027	2.772	-0.008	-0.0084	-0.0082	2.764	-0.003	+0.001	+0.006
FLOPs(G)	323.54	+6.6%	+14.9%	+26.4%	344.98	+0.001%	+0.002%	+0.003%	344.98	+0.001%	+0.003%	+0.007%

六，结论

在本文中，我们介绍了Ultramem，与MOE相比，它具有最少的内存访问，因此获得了六倍的优势。同时，在性能方面，超符号通过相同的参数和计算与模型容量的增加相同，表明较高的缩放能力。这项工作为开发更有效和可扩展语言模型提供了一个有希望的方向。

ACKNOWLEDGMENTS

我们对Pingshuo Ma和Wenda Liu表示最深切的感谢，以优化大型Ultramem的早期培训的宝贵援助。我们也感谢Siyan Chen进行的超级推理工作，以及Fan Xia对MOE推理速度的努力。

REFERENCES

赫维·阿卜迪（HervéAbdi）。奇异值分解（SVD）和广义奇异值分解。测量和统计的百科全书，907（912）：44，2007。

Joshua Ainslie, James Lee-Thorp, Michiel de Jong, Yury Zemlyanskiy, Federico Lebrón, and Sumit Sanghai. Gqa: Training generalized multi-query transformer models from multi-head checkpoints. *arXiv preprint arXiv:2305.13245*, 2023.

吉米·雷巴、杰米·瑞恩·基罗斯和杰弗里·E·辛顿。层标准化。 arXiv 预印本arXiv：1607.06450，2016。

Daniel Bershatsky, Daria Cherniuk, Talgat Daulbaev, Aleksandr Mikhalev, and Ivan Oseledets. Lotr: Low tensor rank weight adaptation. *arXiv preprint arXiv:2402.01376*, 2024.

Sid Black, Stella Biderman, Eric Hallahan, Quentin Anthony, Leo Gao, Laurence Golding, Horace He, Connor Leahy, Kyle McDonell, Jason Phang, et al. Gpt-neox-20b: An open-source autoregressive language model. *arXiv preprint arXiv:2204.06745*, 2022.

Tom B Brown. Language models are few-shot learners. *arXiv preprint ArXiv:2005.14165*, 2020.

Zewen Chi，Li Dong，Shaohan Huang，Damai Dai，Shuming MA，Barun Patra，Saksham Singhal，Payal Bajaj，Xia Song，Xian-Ling Mao等。关于稀疏的专家混合物的表示。神经信息处理系统的进展，35：34600–34613，2022。

克里斯托弗·克拉克、肯顿·李、张明伟、汤姆·科维亚特科斯基、迈克尔·柯林斯和克里斯蒂娜·图塔诺娃。 Boolq：探索自然是/否问题的惊人难度。 arXiv 预印本arXiv：1905.10044，2019。

Peter Clark、Isaac Cowhey、Oren Etzioni、Tushar Khot、Ashish Sabharwal、Carissa Schoenick 和 Oyvind Tafjord。您认为您已经解决了问答问题吗？尝试 arc，ai2 推理挑战。arXiv 预印本 arXiv： 1803.05457，2018。

一起电脑。 Redpajama：重现 llama 训练数据集的开源配方，2023.URL https://github.com/togethercomputer/RedPajama-Data。

Róbert Csordás, Kazuki Irie, and Jürgen Schmidhuber. Approximating two-layer feedforward networks for efficient transformers. *arXiv preprint arXiv:2310.10837*, 2023.

戴大麦，邓成奇，赵成刚，徐荣兴，高华作，陈德利，李嘉实，曾旺丁，于兴凯，吴宇，等。 Deepseekmoe：迈向混合专家语言模型的终极专家专业化。 arXiv 预印本 arXiv:2401.06066, 2024。

Dheeru Dua、Yizhong Wang、Pradeep Dasigi、Gabriel Stanovsky、Sameer Singh 和 Matt Gardner。Drop：需要对段落进行离散推理的阅读理解基准。 arXiv 预印本 arXiv：1903.00161，2019。

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

威廉·费杜斯、巴雷特·佐夫和诺姆·沙泽尔。开关变压器：通过简单高效的稀疏性扩展到万亿参数模型。机器学习研究杂志，23（120）：1-39，2022。

Mor Geva, Roei Schuster, Jonathan Berant, and Omer Levy. Transformer feed-forward layers are key-value memories. *arXiv preprint arXiv:2012.14913*, 2020.

Xu Owen He. Mixture of a million experts. *arXiv preprint arXiv:2407.04153*, 2024.

丹·亨德里克斯、科林·伯恩斯、史蒂文·巴沙特、安迪·邹、曼塔斯·马泽卡、道恩·宋和雅各布·斯坦哈特。测量大规模多任务语言理解。 arXiv 预印本arXiv：2009.03300，2020。

AG Howard。 Mobilenets：用于移动视觉应用程序的有效卷积神经网络。ARXIV预印型ARXIV：1704.04861，2017。

Herve Jegou，Matthijs Douze和Cordelia Schmid。最近邻居搜索的产品量化。在模式分析和机器智能上进行的交易，33（1）：117–128，2010。

Albert Q Jiang, Alexandre Sablayrolles, Antoine Roux, Arthur Mensch, Blanche Savary, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Emma Bou Hanna, Florian Bressand, et al. Mixtral of experts. *arXiv preprint arXiv:2401.04088*, 2024.

曼达尔·乔希、恩索尔·崔、丹尼尔·S·韦尔德和卢克·泽特莫耶。 Triviaqa：用于阅读理解的大规模远程监督挑战数据集。 arXiv 预印本 arXiv:1705.03551, 2017。

Gyuwan Kim和Tae-Hwan Jung。验证语言模型的大型产品钥匙内存。 Arxivpreprint Arxiv：2010.03881，2020。

Jakub Krajewski, Jan Ludziejewski, Kamil Adamczewski, Maciej Pióro, Michał Krutul, Szymon Antoniak, Kamil Ciebiera, Krystian Król, Tomasz Odrzygóźdź, Piotr Sankowski, et al. Scaling laws for fine-grained mixture of experts. *arXiv preprint arXiv:2402.07871*, 2024.

Guillaume Lample，Alexandre Sablayrolles，Marc’aurio Ranzato，Ludovic Denoyer和Hervéjégou。带有产品钥匙的大型内存层。神经信息处理系统的进展，2019年32，2019。

Aixin Liu, Bei Feng, Bin Wang, Bingxuan Wang, Bo Liu, Chenggang Zhao, Chengqi Dengr, Chong Ruan, Damai Dai, Daya Guo, et al. Deepseek-v2: A strong, economical, and efficient mixture-of-experts language model. *arXiv preprint arXiv:2405.04434*, 2024.

Osman Asif Malik和Stephen Becker。使用TensorSketch的大型张量的低级Tucker分解。神经信息处理系统的进展，2018年3月31日。

Deepak Narayanan，Mohammad Shoeybi，Jared Casper，Patrick Legresley，Mostofa Patwary，Vijaykorthikanti，Dmitri Vainbrand，Prethvi Kashinkunti，Julie Bernauer，Bryan Catanzaro，Bryan Catanzaro等人对大型大型语言模型培训Gputron-eggutron-tersters tersters。在国际高性能计算，网络，存储和分析的国际会议上，pp。 1-15，2021。

Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. 2019.

Colin Raffel、Noam Shazeer、Adam Roberts、Katherine Lee、Sharan Narang、Michael Matena、YanqiZhou、Wei Li 和 Peter J Liu。使用统一的文本到文本转换器探索迁移学习的局限性。机器学习研究杂志，21（140）：1-67，2020。

Samyam Rajbhandari、李从龙、姚哲伟、张敏佳、Reza Yazdani Aminabadi、Am-mar Ahmad Awan、Jeff Rasley 和 Yuxiong He。 Deepspeed-moe：推进专家混合推理和训练，为下一代人工智能规模提供动力。国际机器学习会议，第 18332–18346 页。 PMLR，2022 年。

David Rein, Betty Li Hou, Asa Cooper Stickland, Jackson Petty, Richard Yuanzhe Pang, Julien Dirani, Julian Michael, and Samuel R Bowman. Gpqa: A graduate-level google-proof q&a benchmark. *arXiv preprint arXiv:2311.12022*, 2023.

斯蒂芬·罗勒、桑巴亚尔·苏赫巴托、杰森·韦斯顿等。大型稀疏模型的哈希层。神经信息处理系统的进展，34：17555–17566，2021。

坂口圭介、罗南·勒·布拉斯、钱德拉·巴加瓦图拉和崔艺珍。 Winogrande：大规模的对抗性 winograd 模式挑战。 ACM 通讯，64(9):99–106,2021。

里科·森里奇、巴里·哈多和亚历山德拉·伯奇。带有子词单元的罕见词的神经机器翻译。 arXiv 预印本 arXiv:1508.07909, 2015。

Noam Shazeer, Azalia Mirhoseini, Krzysztof Maziarz, Andy Davis, Quoc Le, Geoffrey Hinton, and Jeff Dean. Outrageously large neural networks: The sparsely-gated mixture-of-experts layer. *arXiv preprint arXiv:1701.06538*, 2017.

Kai Shen，Junliang Guo，Xu Tan，Siliang Tang，Rui Wang和Jiang Bian。有关变压器中的Relu和Softmax的研究。 Arxiv预印型ARXIV：2302.06461，2023。

Mohammad Shoeybi, Mostofa Patwary, Raul Puri, Patrick LeGresley, Jared Casper, and Bryan Catanzaro. Megatron-lm: Training multi-billion parameter language models using model parallelism. *arXiv preprint arXiv:1909.08053*, 2019.

Jianlin Su, Murtadha Ahmed, Yu Lu, Shengfeng Pan, Wen Bo, and Yunfeng Liu. Roformer: Enhanced transformer with rotary position embedding. *Neurocomputing*, 568:127063, 2024.

Mirac Suzgun、Nathan Scales、Nathanael Schärli、Sebastian Gehrmann、Yi Tay、Hyung Won Chung、Aakanksha Chowdhery、Quoc V Le、Ed H Chi、Denny Zhou 等。具有挑战性的大板凳任务以及思维链是否可以解决它们。 arXiv 预印本 arXiv:2210.09261, 2022。

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023.

熊瑞斌、杨运昌、何迪、郑凯、郑树新、陈星、张惠帅、兰岩岩、王立伟和刘铁岩。关于 Transformer 架构中的层归一化。国际机器学习会议，第 10524–10533 页。 PMLR，2020。

罗温·泽勒斯、阿里·霍尔兹曼、尤纳坦·比斯克、阿里·法哈迪和 Yejin Choi。 Hellaswag：机器真的能完成你的句子吗？ arXiv 预印本 arXiv:1905.07830, 2019。

Wanjun Zhong, Ruixiang Cui, Yiduo Guo, Yaobo Liang, Shuai Lu, Yanlin Wang, Amin Saied, Weizhu Chen, and Nan Duan. Agieval: A human-centric benchmark for evaluating foundation models. *arXiv preprint arXiv:2304.06364*, 2023.

周艳琪、雷涛、刘汉晓、杜楠、黄艳萍、赵文森、戴安德烈、Quoc VLe、James Laudon 等。专家混合与专家选择路由。神经信息处理系统的进展，35：7103–7114，2022 年。

超注性初始化

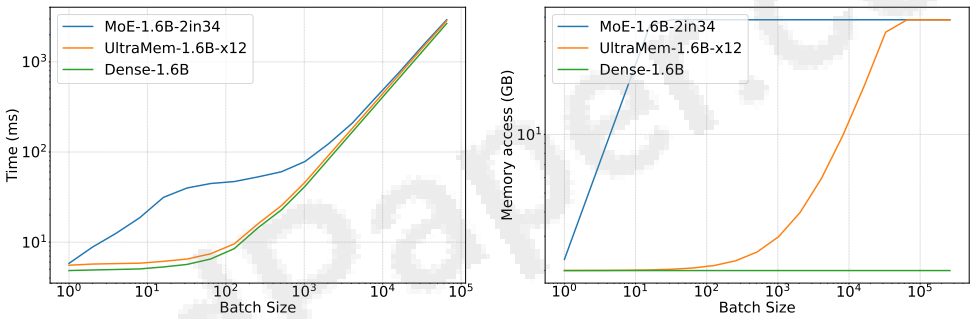
我们用n（0，e2kHl）初始化值，其中k是激活的值编号，h是头号，e是值的扩展时 umber, 间。为了确保Ultramem的输出分布为N（0，12L），以确认Top-M得分的平均值为1。—), We

假设候选分数遵循n（0，1）和k ≪K。我们可以简化问题：给定n标准高斯分布式随机变量x 1，...，x n，随机变量y =平均值（topm（topm（topm（topm）（x 1，...，x n））），找到预期值e（y）。很难获得e（y）的分析溶液，因此我们通过从高斯分布中对m次n点进行采样并计算TOP-M值的平均值来近似E（y）。

然后，我们将查询层标准重量初始化为1/ e（y），键层标准权重为1/ √ dkto确保候选分数的预期为1。 $\sqrt{D_k}$

B推理时间和内存访问

图7显示，与MOE相比，Ultramem的内存访问增长速度要慢得多，仅在批处理大小达到131,072时与MOE进行安装，并且在推理速度上具有优势。



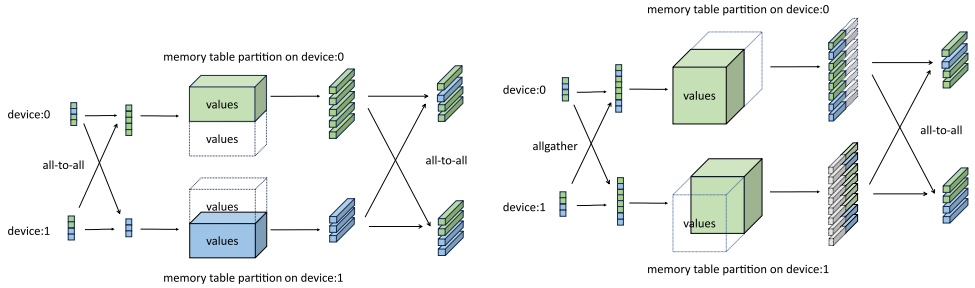
(a) Inference time (b) Memory access

图7：变压器，MOE和Ultramem的推理时间和内存访问。我们确保三个模型具有相同的计算，并且MOE和Ultramem具有相同的参数。 thex轴和y轴都是对数尺度绘制的。序列长度为1，因为在授权期间，我们只能一次预测一个令牌，而密钥/值缓存长度为2048。

C威毒性支持培训效率

随着内存表缩放到数十亿甚至数万亿个参数时，模型并行性变得非常跨多个设备分布模型参数和优化器状态，以确保它们进入设备内存，并且可以在合理的时间范围内进行训练。我们利用Megatron（Shoeybi等，2019; Narayanan等，2021）3D平行性（管道并行性，数据并行性和张量并行性）进行训练。但是，需要有效地对内存表进行多种平行性修改。因为管道并行性无法解决Asingle层的参数超过单个设备的存储容量的方案，而张量并行性将其限制在相对较小的GPU上，因此不足以满足存储器表的记忆要求。因此，我们提出了跨DATA的组合和张量并联基团或其亚组的组合表，以确保有效的分布和可扩展性。

可以将内存表在数字上分区或尺寸分区。附录D中详细详细介绍了Number和尺寸分区的整个过程，以及他们的沟通量分析和有关如何选择适当分区方法的拨号。



(a) Number-wise partitioning (b) Dimension-wise partitioning

图8：数字分区和尺寸分区的过程。图表中省略了加权的沉重步骤。

数字和尺寸的分区。但是，TOP-M的增加将按比例地启动开销。此外，由于加权和汇集后值的大小增加，隐式值的扩展将进一步影响沟通量的fordimension-wise分区。

为了进一步提高绩效，已经实施了几项关键修改：

融合的查找降低操作员：这个新引入的操作员加速了计算，并通过将查找和加权和加权的总和池操作结合到一个更有效的步骤中，从而添加了内存使用情况。

异步执行策略：认识到内存层的跨层利用率的好处，我们采用了异步执行策略。这种战略选择允许与密集的网络操作一起对内存计算进行处理，从而大大影响了整个系统性能。

这些增强表明，我们的平行性策略在大型帧构造中的功效，为对大型模型的更有效培训铺平了道路。

d数字和尺寸的分区详细信息

图8显示了数字和尺寸分区的过程。对于数字分区，我们首先在索引上执行一个将它们分配给相应设备的索引。查找操作后，结果将发送回原始设备，然后进行加权总和池。对于尺寸划分，我们需要在索引上执行一个全聚集操作，以在跨设备上获取无形资点。然后执行查找操作，尺寸划分允许在完成加权总和池后将结果发送回每个设备。

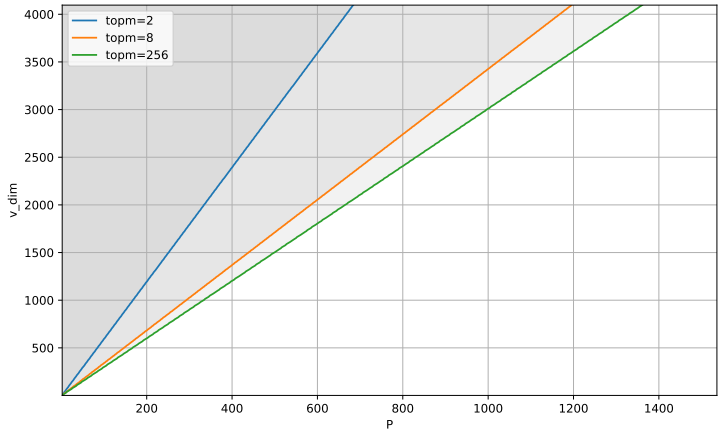


图9：p和v之间的关系数字 /尺寸的沟通量等于1，阴影区域的数字 /尺寸大于1

假设内存表分布在P处理器上，则可以将通信量描述如下：

数字分区的通信量（不考虑索引重复数据删除）：

- All-to-all transmission of indices: $sizeof(int) \times bs \times topm \times (P-1)/P$

· 查找后的全部传输嵌入：sizeOf（bf loat16）×bs×topm×t topm×v dim×（p - 1）/p

尺寸分区的通信量：

- AllGather indices: $sizeof(int) \times bs \times topm \times (P-1)$
- AllGather scores: $sizeof(bfloat16) \times bs \times topm \times (P-1)$

· 嵌入后的全部传输后外观减少：（bfloat16）×vs×vdim×（p - 1）/p的大小

在这里v dim是值维度，BS是批处理尺寸序列序列长度。图9显示了这两种分区方法的通信量P和V DIM之间的处理，帮助我们在固定配置下选择适当的分区方法。

E实验设置

表4显示了所有实验的常见超参数。 “ LR”的学习率，对应于Val-ues 6e-4、2.5E-4、2e-4和1.2e-4的vEN-4和1.2E-4，尺寸为151m，680m，1.6.6b和6.5b（棕色，棕色，棕色，，棕色，分别为棕色，棕色， 2020）。考虑到Ultramem和PKM的插入，Forultramem-151m，它是3：5/6：8/9：11，其中3：5刺激了从laylayer 3中获取超前输入，然后插入了layerer 5的输出，等等，等等，等等。 。对于Ultramem-680m，它是3：7/8：12/13：17/18：22。对于Ultramem-1.6b，是3：7/8：12/13：17/18：22/23：27/28：32。forpkm-151m，是6：6。对于PKM-680M，它是12：12。对于PKM-1.6B，是16:16。 Ultramem和Moe型号的设置型号基于密度参数尺寸对应着密集的对应物。表6显示了用于缩放实验中使用的模型PA-型 - 射手设置。更重要的是，表5显示了Ultra-MEM的共同设置。

Configuration Key	Value
Weight decay	0.1
β_1	0.9
β_2	0.95
LR	6e-4/2.5e-4/2e-4/1.2e-4
LR end ratio	0.1
LR schedule	cosine
LR warmup ratio	0.01
Dropout	0.1
Batch size	2048
Sequence length	2048
Training step	238418

表4：训练超参数

Configuration Key	Value
Tucker rank r	2
Multi-core scoring h	2
Virtual memory expansion E	4
Aux loss weight α	0.001
Aux loss margin τ	0.15

表5：常见的Ultramem配置

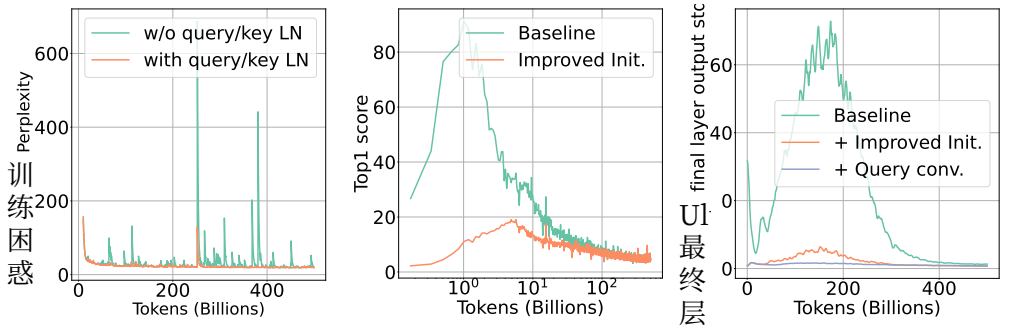
评估数据集。我们使用10个基准来评估各种模型。

1. Knowledge: Massive Multitask Language Understanding (MMLU) (Hendrycks et al., 2020), TriviaQA (Joshi et al., 2017), Graduate-Level Google-Proof Q&A Benchmark (GPQA) (Rein et al., 2023), AI2 Reasoning Challenge (ARC) (Clark et al., 2018).
2. Reasoning: BIG-Bench Hard (BBH) (Suzgun et al., 2022), Boolean Questions (BoolQ) (Clark et al., 2019), HellaSwag (Hella) (Zellers et al., 2019), WinoGrande (Wino) (Sakaguchi et al., 2021).
3. Reading comprehension: Discrete Reasoning Over Paragraphs (DROP) (Dua et al., 2019).
4. Comprehensive ability: AGIEval (Zhong et al., 2023)

Model	Hidden Dim	Inner Dim	Attn Head	Layer	Top-m	Expert	Kdim	Knum	Mem Layer	Param (B)	FLOPs (G)
Dense-151M	1024	4096	16	12	-	-	-	-	-	0.15	0.30
Dense-680M	1536	6144	16	24	-	-	-	-	-	0.68	1.36
Dense-1.6B	2048	8192	16	32	-	-	-	-	-	1.61	3.21
Dense-6.5B	4096	16384	32	32	-	-	-	-	-	6.44	12.88
MoE-151M-2in32	1024	2528	16	12	2	32	-	-	-	2.04	0.35
MoE-680M-2in33	1536	3584	16	24	2	33	-	-	-	8.95	1.50
MoE-1.6B-2in34	2048	4672	16	32	2	34	-	-	-	21.36	3.52
PKM-151M-x12	1024	4096	16	12	16x6	-	512	1347	1	2.04	0.35
PKM-680M-x12	1536	6144	16	24	35x8	-	768	2308	1	8.95	1.50
PKM-1.6B-x12	2048	8192	16	32	42x12	-	896	1792	1	21.44	3.52
UltraMem-151M-x10	2048	8192	16	32	42x2	-	256	1024	3	1.71	0.35
UltraMem-151M-x12	1024	4096	16	12	16x2	-	256	1100	3	2.03	0.35
UltraMem-680M-x12	1536	6144	16	24	35x2	-	384	1632	4	8.93	1.49
UltraMem-1.6B-x12	2048	8192	16	32	42x2	-	448	1792	6	21.41	3.50

表6：模型参数设置。 TOP-M表示MOE中选择的专家编号，是指PKM和Ultramem中选择的ValueNumber Times头号。 KDIM是指PKM和硫酸含量的关键维度。 knum表示键的数量，knum 2是值的数量。

F 更多实验结果



(a) Training perplexity (b) Top1 score (c) UltraMem output standard deviation

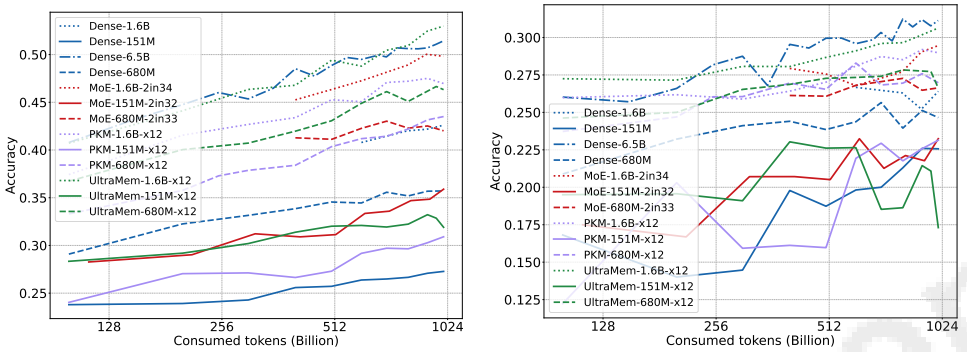
图10：模型培训状态详细信息。“Top1得分”是指检索到的犬的最高分数。“Ultramem Outper STD”表示来自占粉的最后一层的输出的标准偏差。

表7：各种模型的所有性能指标

Model Model	Param (B)	FLOPs (G)	ARC-C↑	GPQA↑	Trivia QA↑	MMLU↑	BBH cot↑	BoolQ↑	Hella Swag↑	Wino Grande↑	AGI Eval↑	DROP↑	Avg↑
Dense-151M	0.15	0.30	25.60	19.98	12.67	26.50	22.57	50.15	35.07	52.49	9.03	13.60	26.77
PKM-151M-x12	2.04	0.35	25.94	17.30	24.66	25.69	23.14	53.48	42.25	51.38	9.65	13.10	28.66
MoE-151M-2in32	2.04	0.35	26.96	17.30	33.27	26.58	23.24	55.96	48.44	55.96	9.34	18.57	**31.56**
UltraMem-151M-x12	2.03	0.35	25.68	19.42	28.97	25.62	22.65	47.74	43.96	50.83	10.00	14.08	28.89
Dense-680M	0.68	1.36	24.06	21.09	27.16	24.64	24.65	46.42	48.83	54.93	9.44	22.97	30.42
PKM-680M-x12	8.95	1.50	25.51	20.65	46.31	25.22	26.98	41.80	57.32	61.72	8.94	25.20	33.97
MoE-680M-2in33	8.95	1.50	25.17	20.54	34.19	24.38	26.63	43.70	62.71	59.98	7.39	26.54	33.13
UltraMem-680M-x12	8.93	1.49	23.72	21.99	55.17	24.97	26.62	48.20	64.15	60.54	8.26	25.14	**35.88**
Dense-1.6B	1.61	3.21	26.30	21.76	39.65	26.19	26.41	51.50	58.6	61.72	9.22	22.63	34.81
PKM-1.6B-x12	21.13	3.48	26.71	22.99	48.92	24.80	28.98	60.06	65.46	63.93	9.51	27.55	37.89
MoE-1.6B-2in34	21.36	3.52	25.43	21.32	59.56	26.18	29.46	42.78	67.34	63.93	6.63	28.81	37.14
UltraMem-1.6B-x12	21.41	3.50	25.94	24.66	66.38	24.67	30.63	59.8	71.52	66.38	8.77	29.99	**40.88**
Dense-6.5B	6.44	12.88	28.16	19.98	57.28	27.68	31.14	68.2	69.73	65.9	9.23	33.12	41.04

表8：模型改进的独立消融研究

	Train Loss ↓	Valid. Loss ↓
PKM-151M-x10	2.604	2.828
+ rm softmax	-0.034	-0.006
+ half vdim+proj	-0.027	-0.02
+ share query	-0.003	-0.002
+ split big mem	-0.003	-0.005
+ query/key LN	-0.002	+0.003
+ IVE	-0.025	-0.023
+ TDQKR	-0.003	-0.007
+ TDQKR + MCS	-0.02	-0.009
+ value lr decay	-0.017	-0.007
+ query conv	-0.005	-0.001



(a) Average accuracy

(b) BBH-cot-3shot accuracy

(c) DROP accuracy

(d) Hellaswag accuracy

(e) Winogrande 5shot accuracy

(f) TriviaQA 5shot accuracy

图11：在整个培训中所有可观察的评估的准确性变化。