

# DeepSeekMoE：在专家混合语言模型中迈向终极专家专业化

Damai Dai<sup>\*1,2</sup>, Chengqi Deng<sup>1</sup>, Chenggang Zhao<sup>\*1,3</sup>, R.X. Xu<sup>1</sup>, Huazuo Gao<sup>1</sup>, Deli Chen<sup>1</sup>, Jiashi Li<sup>1</sup>, Wangding Zeng<sup>1</sup>, Xingkai Yu<sup>\*1,4</sup>, Y. Wu<sup>1</sup>, Zhenda Xie<sup>1</sup>, Y.K. Li<sup>1</sup>, Panpan Huang<sup>1</sup>, Fuli Luo<sup>1</sup>, Chong Ruan<sup>1</sup>, Zhifang Sui<sup>2</sup>, Wenfeng Liang<sup>1</sup>

1 DeepSeek-AI

2 北京大学多媒体信息处理国家重点实验室

3 清华大学交叉信息科学研究所 4 南京大学软件新技术国家重点实验室

[{daidamai, szf}@pku.edu.cn, {wenfeng.liang}@deepseek.com"><https://github.com/deepseek-ai/DeepSeek-MoE>](mailto:{daidamai, szf}@pku.edu.cn)

## Abstract

在大型语言模型时代，专家混合（MoE）是一种很有前途的架构，用于在扩展模型参数时管理计算成本。然而，像 GShard 这样的传统 MoE 架构激活了顶级专家，在确保专家专业化方面面临着挑战，即每个专家都获得不重叠且有针对性的知识。作为回应，我们提出了 DeepSeekMoE 架构，以实现最终的专家专业化。它涉及两个主要策略：（1）将专家细分为不同的专家，并激活灵活组合激活专家；（2）将专家隔离为共享专家，旨在捕获共同知识并减少路由专家中的冗余。从具有 2B 参数的适度规模开始，我们证明 DeepSeekMoE 2B 实现了与 GShard 2.9B 相当的性能，后者具有  $1.5 \times$  专家参数和计算。此外，DeepSeekMoE 2B 在总参数数量相同的情况下几乎接近其密集对应模型的性能，这设定了 MoE 模型的上限。随后，我们将 DeepSeekMoE 扩展到 16B 参数，并表明它实现了与 LLaMA2 7B 相当的性能，而计算量仅为约 40%。此外，我们将 DeepSeekMoE 扩展到 145B 参数的初步努力一致验证了其相对于 GShard 架构的巨大优势，并显示其性能与 DeepSeek 67B 相当，仅使用 28.5%（甚至可能是 18.2%）的计算量。

## 一、简介

最近的研究和实践经验表明，只要有足够的训练数据，通过增加参数和计算预算来扩展语言模型可以产生更强大的模型（Brown 等人，2020 年；Hoffmann 等人，2022 年；OpenAI，2023 年；Touvron 等人，2023a）。然而，必须承认，将模型扩展到极大规模的努力也与极高的计算成本相关。考虑到巨大的成本，专家混合（MoE）架构（Jacobs et al., 1991; Jordan and Jacobs, 1994; Shazeer et al., 2017）已成为一种流行的解决方案。它可以

---

\*在 DeepSeek-AI 实习期间的贡献。

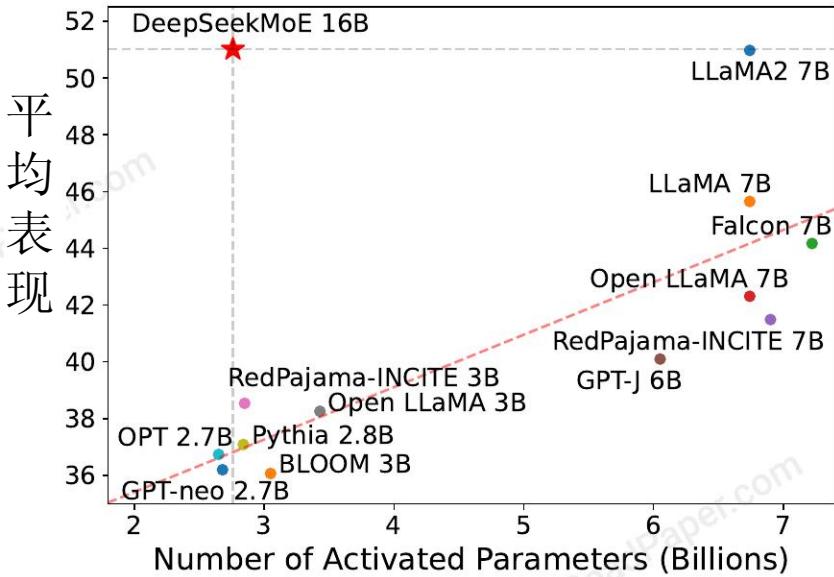


图1 | DeepSeekMoE 16B 与 Open LLM 排行榜上的开源模型之间的比较。红色虚线是根据除 DeepSeekMoE 16B 之外的所有模型的数据点线性拟合的。DeepSeekMoE 16B 始终大幅优于具有相似数量激活参数的模型，并实现了与 LLaMA2 7B 相当的性能，后者的激活参数约为 2.5 倍。

启用参数缩放，同时将计算成本保持在适度的水平。MoE 架构最近在 Transformers 中的应用 (Vaswani 等人, 2017) 已经成功尝试将语言模型扩展到相当大的规模 (Du 等人, 2022; Fedus 等人, 2021; Lepikhin 等人, 2021; Zoph , 2022)，伴随着卓越的表现。这些成就强调了 MoE 语言模型的巨大潜力和前景。

尽管 MoE 架构具有广阔的潜力，但现有的 MoE 架构可能会遇到知识混合和知识冗余的问题，这限制了专家的专业化，即每个专家都获得非重叠和集中的知识。传统的 MoE 架构用 MoE 层替代 Transformer 中的前馈网络 (FFN)。每个 MoE 层由多个专家组成，每个专家在结构上与标准 FFN 相同，每个令牌分配给一名 (Fedus 等人, 2021) 或两名 (Lepikhin 等人, 2021) 专家。这种架构表现出两个潜在问题：(1) 知识混合性：现有的 MoE 实践通常雇用有限数量的专家（例如 8 或 16 名），因此分配给特定专家的代币可能会涵盖不同的知识。因此，指定的专家将打算在其参数中汇集截然不同类型的知识，而这些知识很难同时利用。(2) 知识冗余：分配给不同专家的代币可能需要共同知识。结果，多个专家可能会集中获取各自参数的共享知识，从而导致专家参数的冗余。这些问题共同阻碍了专家对现有 MoE 实践的专业化，使其无法达到 MoE 模型的理论性能上限。

针对上述问题，我们推出了 DeepSeekMoE，这是一种专为最终专家专业化而设计的创新 MoE 架构。我们的架构涉及两个主要策略：(1) 细粒度专家分割：在保持参数数量不变的同时，我们通过分割专家将专家分割成更细的粒度。

FFN 中间隐藏维度。相应地，在保持计算成本恒定的情况下，我们还激活更细粒度的专家，以实现更灵活、适应性更强的激活专家组合。细粒度的专家分割可以将不同的知识更精细地分解，更精确地学习到不同的专家中，每个专家将保留更高的专业水平。此外，组合激活专家的灵活性的增加也有助于更准确、更有针对性的知识获取。（2）共享专家隔离：我们隔离某些专家作为始终激活的共享专家，旨在捕获和巩固不同背景下的共同知识。通过将公共知识压缩到这些共享专家中，将减少其他路由专家之间的冗余。这可以提高参数效率并确保每个路由专家通过专注于独特的方面来保持专业性。DeepSeekMoE 中的这些架构创新提供了训练参数高效的 MoE 语言模型的机会，其中每个专家都是高度专业化的。

从具有 2B 参数的适度规模开始，我们验证了 DeepSeek-MoE 架构的优势。我们对涵盖不同任务的 12 个零样本或少样本基准进行评估。实证结果表明，DeepSeekMoE 2B 大幅超越 GShard 2B (Lepikhin 等人, 2021)，甚至可以与 GShard 2.9B（具有  $1.5 \times$  专家参数和计算量的更大 MoE 模型）相匹配。值得注意的是，我们发现 DeepSeekMoE 2B 在参数数量相同的情况下几乎接近其密集对应模型的性能，这设定了 MoE 语言模型的严格上限。为了追求更深入的见解，我们对 DeepSeekMoE 的专家专业化进行了详细的消融研究和分析。这些研究验证了细粒度专家分割和共享专家隔离的有效性，并为支持 DeepSeekMoE 能够实现高水平专家专业化的论点提供了经验证据。

利用我们的架构，我们随后将模型参数扩展到 16B，并在具有 2T 令牌的大规模语料库上训练 DeepSeekMoE 16B。评估结果显示，仅用约 40% 的计算量，DeepSeekMoE 16B 就可以达到与 DeepSeek 7B (DeepSeek-AI, 2024)（在同一 2T 语料库上训练的密集模型）相当的性能。我们还将 DeepSeekMoE 与开源模型进行了比较，评估表明 DeepSeekMoE 16B 始终大幅优于具有相似激活参数数量的模型，并且实现了与 LLaMA2 7B 相当的性能 (Touvron 等人, 2023b)，后者具有大约是激活参数的 2.5 倍。图 1 显示了 Open LLM 排行榜<sup>1</sup> 的评估结果。此外，我们还进行监督微调 (SFT) 以进行对齐，将模型转换为聊天模型。评估结果显示，DeepSeekMoE Chat 16B 在聊天设置中也实现了与 DeepSeek Chat 7B 和 LLaMA2 SFT 7B 相当的性能。受这些结果的鼓舞，我们进一步进行了初步努力，将 DeepSeekMoE 扩展到 145B。实验结果仍然一致地验证了其相对于 GShard 架构的实质性优势。此外，它的性能与 DeepSeek 67B 相当，仅使用 28.5%（甚至可能是 18.2%）的计算量。

我们的贡献总结如下：

Ø 架构创新。我们介绍 DeepSeekMoE，一种创新的 MoE 架构，旨在实现最终的专家专业化，它采用细粒度专家分割和共享专家隔离两种主要策略。

■ 经验证。我们进行了大量的实验来凭经验验证 DeepSeekMoE 架构的有效性。实验结果验证了高

---

<sup>1</sup>[https://huggingface.co/spaces/HuggingFaceH4/open\\_llm\\_leaderboard](https://huggingface.co/spaces/HuggingFaceH4/open_llm_leaderboard)

DeepSeekMoE 2B 的专家专业化水平，并表明 DeepSeekMoE 2B 几乎可以接近 MoE 模型的上限性能

■ 可扩展性。我们扩展了 DeepSeekMoE 来训练 16B 模型，结果表明，只需大约 40% 的计算量，DeepSeekMoE 16B 就可以达到与 DeepSeek 7B 和 LLaMA2 7B 相当的性能。我们还初步努力将 DeepSeekMoE 扩展到 145B，突出其相对于 GShard 架构的一贯优势，并展示与 DeepSeek 67B 相当的性能。 ■ 与教育部对齐。我们成功地在 DeepSeekMoE 16B 上进行监督微调，以创建对齐的聊天模型，展示了 DeepSeekMoE 16B 的适应性和多功能性。公开发布。本着开放研究的精神，我们向公众发布了 DeepSeekMoE 16B 的模型检查点。值得注意的是，该模型可以部署在具有 40GB 内存的单个 GPU 上，无需量化。

## 2. Preliminaries: Mixture-of-Experts for Transformers

我们首先介绍 Transformer 语言模型中常用的通用 MoE 架构。标准 Transformer 语言模型是通过堆叠标准 Transformer 块的层数来构建的，其中每个块可以表示如下：

$$\mathbf{u}_{1:T}^l = \text{Self-Att}(\mathbf{h}_{1:T}^{l-1}) + \mathbf{h}_{1:T}^{l-1}, \quad (1)$$

$$\mathbf{h}_t^l = \text{FFN}(\mathbf{u}_t^l) + \mathbf{u}_t^l, \quad (2)$$

其中， $T$  表示序列长度， $\text{Self-Att}(\cdot)$  表示自注意力模块， $\text{FFN}(\cdot)$  表示前馈网络（FFN）， $u$  表示 1: 表示第  $\delta$  个注意力模块之后所有 token 的隐藏状态， $h$  表示第  $\delta$  个 Transformer 块之后第  $\delta$  个标记的输出隐藏状态。为了简洁起见，我们在上述公式中省略了层归一化。

构建 MoE 语言模型的典型做法通常是以指定间隔用 MoE 层替换 Transformer 中的 FFN (Dü et al., 2022; Fedus et al., 2021; Lepikhin et al., 2021; Zoph, 2022)。MoE 层由多个专家组成，每个专家在结构上与标准 FFN 相同。然后，每个令牌将分配给一名 (Fedus 等人, 2021) 或两名 (Lepikhin 等人, 2021) 专家。如果将第  $\delta$  FFN 替换为 MoE 层，则其输出隐藏状态  $h$  的计算表示为：

$$\mathbf{h}_t^l = \sum_{i=1}^N \left( g_{i,t} \text{FFN}_i(\mathbf{u}_t^l) \right) + \mathbf{u}_t^l, \quad (3)$$

$$g_{i,t} = \begin{cases} s_{i,t}, & s_{i,t} \in \text{Topk}(\{s_{j,t} | 1 \leq j \leq N\}, K), \\ 0, & \text{otherwise,} \end{cases} \quad (4)$$

$$s_{i,t} = \text{Softmax}_i \left( \mathbf{u}_t^{lT} \mathbf{e}_i^l \right), \quad (5)$$

其中  $\delta$  表示专家总数， $\text{FFN}_\delta(\cdot)$  是第  $\delta$  专家 FFN， $s_{i,t}$  表示的门值第  $\delta$  位专家， $\delta$  表示代币到专家的亲和度， $\text{Topk}(\cdot, \delta)$  表示包含  $\delta$  最高亲和度的集合为第  $\delta$  个令牌和所有  $\delta$  专家计算的得分之间的分数，并且  $e_\delta$  是  $\delta$  中第  $\delta$  专家的质心第 3 层。请注意， $\delta$  是稀疏的，表明只有  $\delta$  门值之外的  $\delta$  为非零。这种稀疏性确保了 MoE 层内的计算效率，即每个令牌将仅分配给 4 名专家并由 4 名专家进行计算。此外，在上述公式中，为了简洁起见，我们省略了层归一化操作。

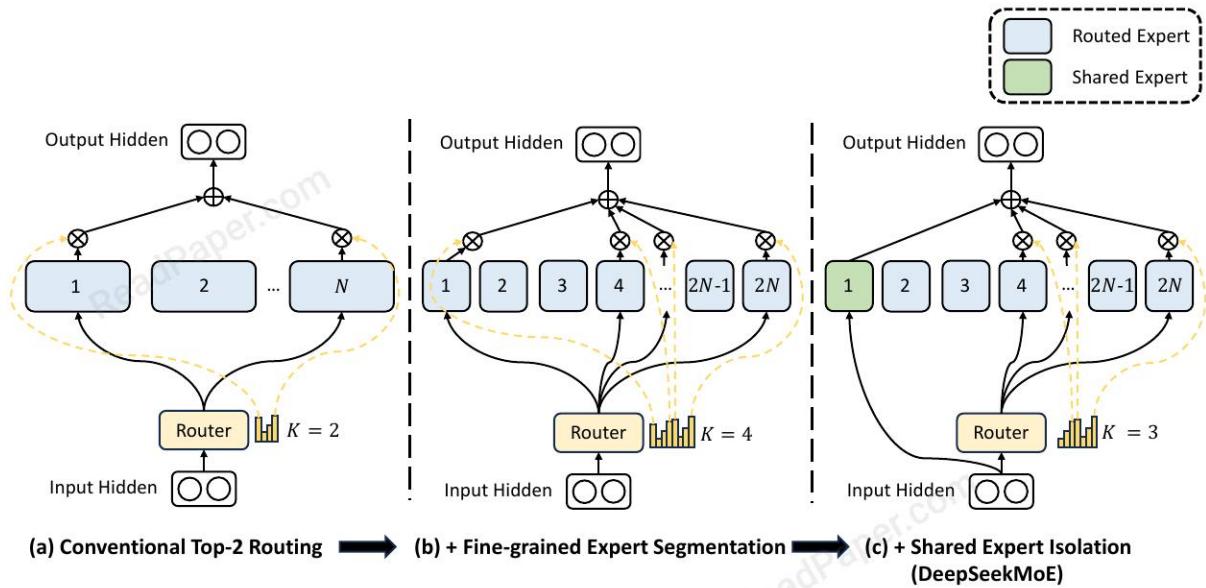


图2| DeepSeekMoE 的插图。子图 (a) 展示了采用传统 top-2 路由策略的 MoE 层。子图 (b) 说明了细粒度的专家分割策略。随后，子图(c)演示了共享专家隔离策略的集成，构成了完整的DeepSeekMoE架构。值得注意的是，在这三种架构中，专家参数的数量和计算成本保持不变。

### 3. DeepSeekMoE Architecture

在第 2 节概述的通用 MoE 架构之上，我们引入了 DeepSeekMoE，它是专门为开发专家专业化的潜力而设计的。如图 2 所示，我们的架构包含两个主要策略：细粒度专家分割和共享专家隔离。这两种策略都是为了提高专家的专业化水平。

#### 3.1. 细粒度专家细分

在专家数量有限的场景下，分配给特定专家的代币将更有可能涵盖不同类型的知识。因此，指定的专家将打算学习其参数中截然不同类型的知识，并且它们很难同时利用。然而，如果每个令牌可以路由给更多的专家，那么不同的知识将有可能分别在不同的专家中分解和学习。在此背景下，每位专家仍然可以保持高水平的专家专业化，有助于专家之间更加集中的知识分配。

为了实现这一目标，在保持专家参数数量和计算成本一致的同时，我们以更细的粒度对专家进行细分。更精细的专家细分使得激活专家的组合更加灵活、适应性更强。具体来说，在图 2(a) 所示的典型 MoE 架构之上，我们通过将 FFN 中间隐藏维度减少到其原始大小的 1 倍，将每个专家 FFN 分割为更小的专家。由于每个专家变得更小，作为回应，我们还将激活的专家数量增加到 5 倍，以保持相同的计算成本，如图 2(b) 所示。随着细粒度

专家分割，MoE层的输出可以表示为：

$$\mathbf{h}_t^l = \sum_{i=1}^{mN} \left( g_{i,t} \text{FFN}_i \left( \mathbf{u}_t^l \right) \right) + \mathbf{u}_t^l, \quad (6)$$

$$g_{i,t} = \begin{cases} s_{i,t}, & s_{i,t} \in \text{Topk}(\{s_{j,t} | 1 \leq j \leq mN\}, mK), \\ 0, & \text{otherwise,} \end{cases} \quad (7)$$

$$s_{i,t} = \text{Softmax}_i \left( \mathbf{u}_t^{lT} \mathbf{e}_i^l \right), \quad (8)$$

其中专家参数总数等于标准FFN中参数数量的乘积，并且表示细粒度专家的总数。通过细粒度的专家分割策略，非零门的数量也将增加到。

从组合的角度来看，细粒度的专家分割策略大大增强了激活专家的组合灵活性。作为说明性示例，我们考虑  $\delta = 16$  的情况。典型的 top-2 路由策略可以产生  $16 \cdot 2 = 120$  种可能的组合。相比之下，如果每个专家被分成 4 个较小的专家，细粒度路由策略可以产生  $64 \cdot 8 = 4,426, 165, 368$  个潜在组合。组合灵活性的激增增强了实现更准确、更有针对性的知识获取的潜力。

### 3.2. 共享专家隔离

使用传统的路由策略，分配给不同专家的令牌可能需要一些常识或信息。结果，多个专家可能会集中获取各自参数的共享知识，从而导致专家参数的冗余。然而，如果有共享专家致力于捕获和巩固不同上下文中的共同知识，那么其他路由专家之间的参数冗余将会得到缓解。这种冗余的减少将有助于建立一个由更多专业专家组成的参数效率更高的模型。

为了实现这一目标，除了细粒度的专家细分策略外，我们还进一步隔离了专家，作为共享专家。无论路由器模块如何，每个令牌都将确定性地分配给这些共享专家。为了保持恒定的计算成本，其他路由专家中激活的专家数量将减少  $\delta / \delta$ ，如图 2(c) 所示。集成共享专家隔离策略后，完整的 DeepSeekMoE 架构中的 MoE 层的公式如下：

$$\mathbf{h}_t^l = \sum_{i=1}^{K_s} \text{FFN}_i \left( \mathbf{u}_t^l \right) + \sum_{i=K_s+1}^{mN} \left( g_{i,t} \text{FFN}_i \left( \mathbf{u}_t^l \right) \right) + \mathbf{u}_t^l, \quad (9)$$

$$g_{i,t} = \begin{cases} s_{i,t}, & s_{i,t} \in \text{Topk}(\{s_{j,t} | K_s + 1 \leq j \leq mN\}, mK - K_s), \\ 0, & \text{otherwise,} \end{cases} \quad (10)$$

$$s_{i,t} = \text{Softmax}_i \left( \mathbf{u}_t^{lT} \mathbf{e}_i^l \right). \quad (11)$$

最后，在 DeepSeekMoE 中，共享专家的数量为  $\delta / \delta$ ，路由专家的总数为  $\delta \delta \wedge \delta / \delta$ ，非零的数量是  $\delta \delta \delta \delta$ 。

值得注意的是，共享专家隔离的原型可以归功于 Rajbhandari 等人。(2022)。关键的区别在于，他们从工程角度推导出该策略，而我们从算法的角度来处理它。

### 3.3. 负载平衡考虑

自动学习的路由策略可能会遇到负载不平衡的问题，这体现了两个显着的缺陷。首先，存在路由崩溃的风险（Shazeer et al., 2017），即模型总是只选择少数专家，导致其他专家无法得到充分的训练。其次，如果专家分布在多个设备上，负载不平衡会加剧计算瓶颈。

专家级平衡损失。为了降低路由崩溃的风险，我们还采用了专家级的余额损失。余额损失的计算如下：

$$\mathcal{L}_{\text{ExpBal}} = \alpha_1 \sum_{i=1}^{N'} f_i P_i, \quad (12)$$

$$f_i = \frac{N'}{K'T} \sum_{t=1}^T \mathbb{1}(\text{Token } t \text{ selects Expert } i), \quad (13)$$

$$P_i = \frac{1}{T} \sum_{t=1}^T s_{i,t}, \quad (14)$$

where  $\alpha_1$  is a hyper-parameter called expert-level balance factor,  $N'$  is equal to  $(mN - K_s)$  and  $K'$  is equal to  $(mK - K_s)$  for brevity.  $\mathbb{1}(\cdot)$  denotes the indicator function.

设备级平衡损失。除了专家级平衡损失之外，我们还引入了设备级平衡损失。当旨在缓解计算瓶颈时，没有必要在专家级别强制执行严格的平衡约束，因为对负载平衡的过多约束会损害模型性能。相反，我们的主要目标是确保跨设备的平衡计算。如果我们将所有路由专家划分为  $\delta$  组  $\{E_1, E_2, \dots, E_{\delta}\}$ ，并将每个组部署在单个设备上，则设备级平衡损失计算如下：

$$\mathcal{L}_{\text{DevBal}} = \alpha_2 \sum_{i=1}^D f'_i P'_i, \quad (15)$$

$$f'_i = \frac{1}{|\mathcal{E}_i|} \sum_{j \in \mathcal{E}_i} f_j, \quad (16)$$

$$P'_i = \sum_{j \in \mathcal{E}_i} P_j, \quad (17)$$

其中  $\delta/2$  是一个称为设备级平衡因子的超参数。在实践中，我们设置一个小的专家级平衡因子来降低路由崩溃的风险，同时设置一个更大的设备级平衡因子来促进跨设备的平衡计算。

## 4. 验证实验

### 4.1. 实验装置

#### 4.1.1. 训练数据和标记化

我们的训练数据是从 DeepSeek-AI 创建的大规模多语言语料库中采样的。该语料库主要集中于英语和汉语，但也包含其他语言。它是去-

来自不同的来源，包括网络文本、数学材料、编码脚本、出版文献和各种其他文本材料。为了验证实验的目的，我们从语料库中采样包含 100B 个标记的子集来训练我们的模型。对于标记化，我们利用 HuggingFace Tokenizer<sup>2</sup> 工具在训练语料库的较小子集上训练字节对编码 (BPE) (Sennrich 等人, 2016) 标记化器。在验证实验中，我们准备了词汇量为 8K 的分词器，当训练更大的模型时，词汇量将会扩大。

#### 4.1.2. 基础设施

我们基于 HAI-LLM (High-Flyer, 2023) 进行实验，这是一种高效、轻量级的训练框架，集成了多种并行策略，包括张量并行 (Korthikanti et al., 2023; Narayanan et al., 2021; Shoeybi 等人, 2019)、ZeRO 数据并行性 (Rajbhandari 等人, 2020)、PipeDream 管道并行性 (Harlap 等人, 2018)，更具体地说，专家并行性 (Lepikhin 等人, 2021) 数据和张量并行。为了优化性能，我们使用 CUDA 和 Triton 开发 GPU 内核 (Tillet 等人, 2019)，用于门控算法并融合不同专家中线性层的计算。

---

所有实验均在配备 NVIDIA A100 或 H800 GPU 的集群上进行。A100 集群中的每个节点都包含 8 个 GPU，通过 NVLink 桥接成对连接。H800 集群每个节点还配备 8 个 GPU，在节点内使用 NVLink 和 NVSwitch 互连。对于 A100 和 H800 集群，都利用 InfiniBand 互连来促进节点之间的通信。

#### 4.1.3. Hyper-Parameters

模型设置。在验证实验中，我们将 Transformer 层数设置为 9，隐藏维度设置为 1280。我们采用多头注意力机制，总共 10 个注意力头，其中每个头的维度为 128。对于初始化，所有可学习参数随机初始化，标准差为 0.006。我们用 MoE 层替换所有 FFN，并确保专家参数的总数等于标准 FFN 的 16 倍。此外，我们将激活的专家参数（包括共享专家参数和激活的路由专家参数）保留为标准 FFN 的 2 倍。在此配置下，每个 MoE 模型总共约有 2B 个参数，激活参数的数量约为 0.3B。

---

训练设置。我们采用 AdamW 优化器 (Loshchilov 和 Hutter, 2019)，超参数设置为  $\delta_1 = 0.9$ ,  $\delta_2 = 0.95$ ,  $\text{weight\_decay} = 0.1$ 。使用预热和逐步衰减策略来安排学习速率。最初，学习率在前 2K 步中从 0 线性增加到最大值。随后，学习率在 80% 的训练步骤中乘以 0.316，在 90% 的训练步骤中再次乘以 0.316。验证实验的最大学习率设置为  $1.08 \times 10 \times 3$ ，梯度裁剪范数设置为 1.0。批量大小设置为 2K，最大序列长度为 2K，每个训练批量包含 4M 个令牌。相应地，训练总步数设置为 25,000，以实现 100B 训练令牌。由于训练数据丰富，我们在训练时不使用 dropout。鉴于模型规模相对较小，所有参数（包括专家参数）都部署在单个 GPU 设备上，以避免计算不平衡。相应地，我们在训练期间不会丢弃任何令牌，并且不会使用设备级

---

<sup>2</sup><https://github.com/huggingface/tokenizers>

平衡损失。为了防止路由崩溃，我们设置专家级平衡因子为 0.01。

为了便于阅读，我们还在附录 A 中提供了不同大小的 DeepSeekMoE 超参数概览表。

#### 4.1.4. 评估基准

我们对涵盖各种类型任务的广泛基准进行评估。我们列出基准如下。

语言建模。对于语言建模，我们在 Pile 的测试集上评估模型 (Gao et al., 2020)，评估指标是交叉熵损失。

语言理解和推理。对于语言理解和推理，我们考虑 HellaSwag (Zellers et al., 2019)、PIQA (Bisk et al., 2020)、ARC-challenge 和 ARC-easy (Clark et al., 2018)。这些任务的评估指标是准确性。

阅读理解。对于阅读理解，我们使用 RACE-high 和 RACE-middle Lai 等人。 (2017)，评价指标是准确性。

代码生成。对于代码生成，我们在 HumanEval (Chen et al., 2021) 和 MBPP (Austin et al., 2021) 上评估模型。评估指标为 Pass@1，表示仅一代尝试的通过率。

**Closed-Book Question Answering.** For closed-book question answering, we consider TriviaQA (Joshi et al., 2017) and NaturalQuestions (Kwiatkowski et al., 2019). The evaluation metric is the Exactly Matching (EM) rate.

## 4.2. 评价

基线。包括 DeepSeekMoE 在内，我们比较了五个模型进行验证实验。Dense 表示总参数为 0.2B 的标准密集 Transformer 语言模型。哈希层 (Roller et al., 2021) 是一种基于 top-1 哈希路由的 MoE 架构，具有 2.0B 总参数和 0.2B 激活参数，与密集基线对齐。Switch Transformer (Fedus et al., 2021) 是另一种著名的基于 top-1 可学习路由的 MoE 架构，其总参数和激活参数与哈希层相同。GShard (Lepikhin 等人, 2021) 采用了 top-2 可学习路由策略，总参数为 2.0B，激活参数为 0.3B，因为与 top-1 路由方法相比，多了一名专家被激活。DeepSeekMoE 有 1 个共享专家和 63 个路由专家，其中每个专家的大小是标准 FFN 的 0.25 倍。包括 DeepSeekMoE 在内，所有比较模型都共享相同的训练语料库和训练超参数。所有比较的 MoE 模型都具有相同数量的总参数，并且 GShard 具有与 DeepSeekMoE 相同数量的激活参数。

结果。我们在表 1 中列出了评估结果。对于所有演示的模型，我们报告了在 100B 代币上训练后的最终评估结果。从表中，我们得出以下观察结果：(1) 在稀疏架构和更多总参数的情况下，哈希层

Metric	# Shot	Dense	Hash Layer	Switch	GShard	DeepSeekMoE
# Total Params	N/A	0.2B	2.0B	2.0B	2.0B	2.0B
# Activated Params	N/A	0.2B	0.2B	0.2B	0.3B	0.3B
FLOPs per 2K Tokens	N/A	2.9T	2.9T	2.9T	4.3T	4.3T
# Training Tokens	N/A	100B	100B	100B	100B	100B
Pile (Loss)	N/A	2.060	1.932	1.881	1.867	<b>1.808</b>
HellaSwag (Acc.)	0-shot	38.8	46.2	49.1	50.5	<b>54.8</b>
PIQA (Acc.)	0-shot	66.8	68.4	70.5	70.6	<b>72.3</b>
ARC-easy (Acc.)	0-shot	41.0	45.3	45.9	43.9	<b>49.4</b>
ARC-challenge (Acc.)	0-shot	26.0	28.2	30.2	31.6	<b>34.3</b>
RACE-middle (Acc.)	5-shot	38.8	38.8	43.6	42.1	<b>44.0</b>
RACE-high (Acc.)	5-shot	29.0	30.0	30.9	30.4	<b>31.7</b>
HumanEval (Pass@1)	0-shot	0.0	1.2	2.4	3.7	<b>4.9</b>
MBPP (Pass@1)	3-shot	0.2	0.6	0.4	0.2	<b>2.2</b>
TriviaQA (EM)	5-shot	4.9	6.5	8.9	10.2	<b>16.6</b>
NaturalQuestions (EM)	5-shot	1.4	1.4	2.5	3.2	<b>5.7</b>

表 1 | 验证实验的评估结果。粗体字体表示最好。与其他 MoE 架构相比，DeepSeekMoE 表现出显著的性能优势。

在相同数量的激活参数下，Switch Transformer 的性能明显优于密集基线。（2）与 Hash Layer 和 Switch Transformer 相比，GShard 的激活参数更多，性能也比 Switch Transformer 稍好。（3）在相同数量的总参数和激活参数的情况下，DeepSeekMoE 比 GShard 表现出压倒性的优势。这些结果展示了我们的 DeepSeekMoE 架构在现有 MoE 架构领域的优越性。

#### 4.3. DeepSeekMoE 与 MoE 模型的上限紧密结合

我们已经证明 DeepSeekMoE 优于密集基线和其他 MoE 架构。为了更准确地了解 DeepSeekMoE 的性能，我们将其与具有更多总参数或激活参数的较大基线进行比较。这些比较使我们能够估计 GShard 或密集基线所需的模型大小，以实现与 DeepSeekMoE 相当的性能。

与 GShard 1.5 的比较。表 2 显示了 DeepSeekMoE 与专家大小为 1.5 倍的更大 GShard 模型之间的比较，这导致专家参数和专家计算量均为 1.5 倍。总体而言，我们观察到 DeepSeekMoE 实现了与 GShard<sup>TM</sup>1.5 相当的性能，这凸显了 DeepSeekMoE 架构固有的显着优势。除了与 GShard-1.5 的比较之外，我们还在附录 B 中展示了与 GShard-1.2 的比较。

此外，我们将 DeepSeekMoE 的总参数数量增加到 13.3B，并将其与总参数数分别为 15.9B 和 19.8B 的 GShard-1.2 和 GShard-1.5 进行比较。我们发现，在更大的规模上，DeepSeekMoE 甚至可以明显优于 GShard<sup>TM</sup>1.5。这些

Metric	# Shot	GShard×1.5	Dense×16	DeepSeekMoE
Relative Expert Size	N/A	1.5	1	0.25
# Experts	N/A	0 + 16	16 + 0	1 + 63
# Activated Experts	N/A	0 + 2	16 + 0	1 + 7
# Total Expert Params	N/A	2.83B	1.89B	1.89B
# Activated Expert Params	N/A	0.35B	1.89B	0.24B
FLOPs per 2K Tokens	N/A	5.8T	24.6T	4.3T
# Training Tokens	N/A	100B	100B	100B
Pile (Loss)	N/A	1.808	1.806	1.808
HellaSwag (Acc.)	0-shot	54.4	55.1	54.8
PIQA (Acc.)	0-shot	71.1	71.9	72.3
ARC-easy (Acc.)	0-shot	47.3	51.9	49.4
ARC-challenge (Acc.)	0-shot	34.1	33.8	34.3
RACE-middle (Acc.)	5-shot	46.4	46.3	44.0
RACE-high (Acc.)	5-shot	32.4	33.0	31.7
HumanEval (Pass@1)	0-shot	3.0	4.3	4.9
MBPP (Pass@1)	3-shot	2.6	2.2	2.2
TriviaQA (EM)	5-shot	15.7	16.5	16.6
NaturalQuestions (EM)	5-shot	4.7	6.3	5.7

表 2 | DeepSeekMoE、更大的 GShard 模型和更大的密集模型之间的比较。在“#专家”行中，“+”表示“共享专家”和“路由专家”。在“#激活的专家”行中，“+”表示“已激活的共享专家”和“已激活的路由专家”。DeepSeekMoE 实现了与包含 1.5 倍专家参数和计算的 GShard 模型相当的性能。此外，DeepSeekMoE 几乎接近具有 16 倍 FFN 参数的密集模型的性能，这在模型容量方面设定了 MoE 模型的上限。

结果也在附录 B 中提供。

与 Dense 的比较 16。表 2 还显示了 DeepSeekMoE 和更大的密集模型之间的比较。为了公平比较，我们不使用广泛使用的注意力和 FFN 参数之间的比率（1:2）。相反，我们配置 16 个共享专家，其中每个专家具有与标准 FFN 相同数量的参数。该架构模仿具有 16 倍标准 FFN 参数的密集模型。从表中我们发现 DeepSeekMoE 几乎接近 Dense 16 的性能，Dense 16 在模型容量方面设定了 MoE 模型的严格上限。这些结果表明，至少在大约 2B 个参数和 100B 个训练令牌的规模上，DeepSeekMoE 的性能与 MoE 模型的理论上限密切相关。此外，我们在附录 B 中提供了与 Dense 4 的额外比较。

#### 4.4. 消融研究

为了证实细粒度专家分割和共享专家隔离策略的有效性，我们对 DeepSeekMoE 进行了消融研究，结果如图 3 所示。为了公平比较，我们确保比较中包含的所有模型都具有

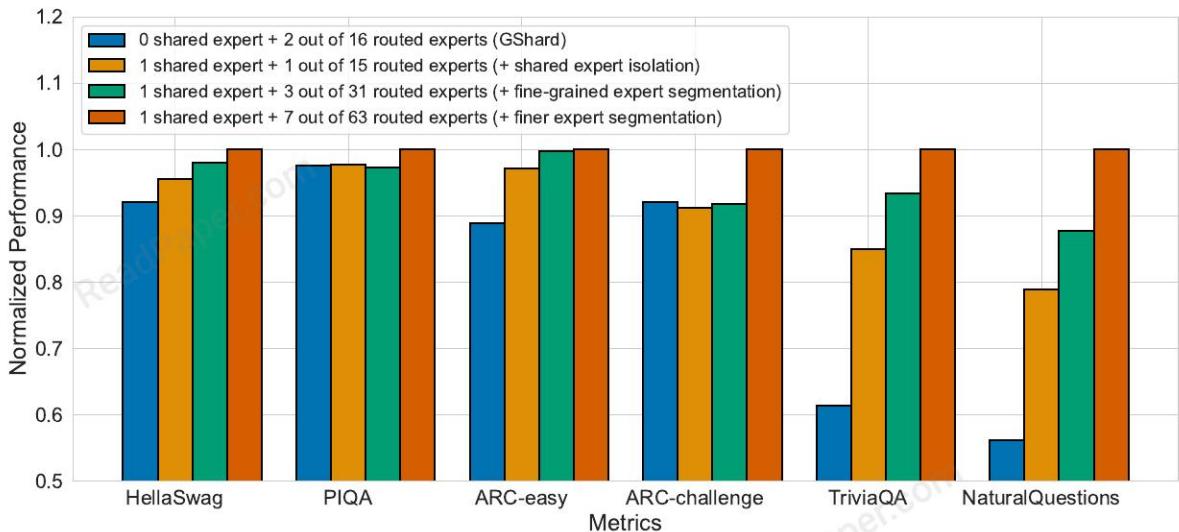


图3 | DeepSeekMoE 的消融研究。为了演示的清晰度，性能按最佳性能标准化。所有比较模型都具有相同数量的参数和激活的参数。我们可以发现，细粒度的专家分割和共享专家隔离都有助于增强整体性能。

总参数和激活参数的数量相同。

共享专家隔离。为了评估共享专家隔离策略的影响，我们基于GShard隔离一位专家作为共享专家。从图3中我们观察到，与GShard相比，共享专家的有意隔离可以提高大多数基准测试的性能。这些结果支持了共享专家隔离策略有助于增强模型性能的主张。

细粒度的专家细分。为了评估细粒度专家分割策略的有效性，我们通过将专家进一步细分为更细粒度来进行更详细的比较。具体来说，我们将每个专家分为2或4个较小的专家，总共32个（1个共享+31个路由）或64个（1个共享+63个路由）专家。图3揭示了一个一致的趋势，即专家分割粒度的不断细化对应于整体模型性能的不断增强。这些发现为细粒度专家细分策略的有效性提供了实证依据。

共享专家与路由专家之间的比率。此外，我们还研究了共享专家和路由专家的最佳比例。基于总共64个专家的最细粒度，并保持总专家数和激活专家数恒定，我们尝试将1、2和4个专家隔离为共享专家。我们发现共享专家和路由专家的不同比例不会显著影响性能，1、2和4个共享专家的Pile损失分别为1.808、1.806和1.811。考虑到1:3的比例会产生稍微更好的Pile损失，因此在扩展DeepSeekMoE时，我们将共享专家和激活的路由专家之间的比例保持为1:3。

#### 4.5. 专家专业化分析

本节我们对DeepSeekMoE 2B的专家专业化进行实证分析。本节中的 DeepSeekMoE 2B 指的是表 1 中报告的模型，即包含 2.0B 个总参数，其中 1 个共享专家和 63 个路由专家中的 7 个被激活。

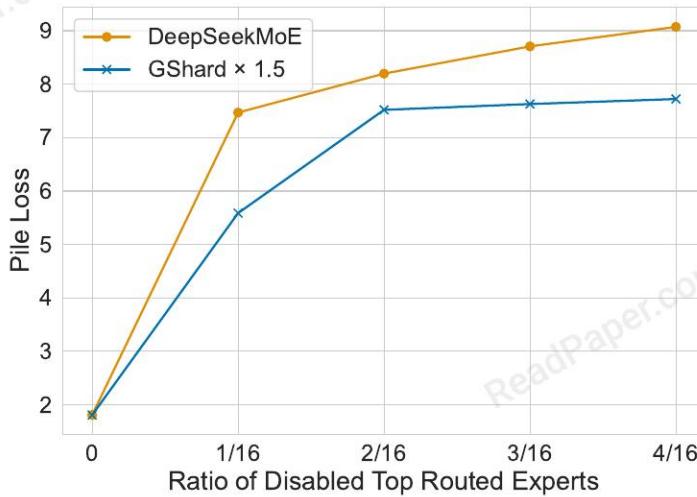


图4|关于不同比例的残疾顶级专家的桩损失。值得注意的是，DeepSeekMoE 对禁用的顶级路由专家的比例表现出更高的敏感性，这表明 DeepSeekMoE 中的路由专家之间的冗余度较低。

DeepSeekMoE 的路由专家冗余度较低。为了评估路由专家之间的冗余度，我们禁用不同比例的顶级路由专家并评估桩损失。具体来说，对于每个 token，我们屏蔽一定比例的路由概率最高的专家，然后从剩余的路由专家中选择前 K 个专家。为了公平起见，我们将 DeepSeekMoE 与 GShard<sup>TM</sup>1.5 进行比较，因为当没有专家被禁用时，它们具有相同的桩损失。如图 4 所示，与 GShard<sup>TM</sup>1.5 相比，DeepSeekMoE 对顶级路由专家的禁用更加敏感。这种敏感性表明 DeepSeekMoE 中的参数冗余程度较低，因为每个路由专家都更不可替代。相比之下，GShard<sup>TM</sup>1.5 在其专家参数之间表现出更大的冗余，因此它可以缓冲顶级路由专家被禁用时的性能下降。

共享专家无法被路由专家取代。为了研究 DeepSeekMoE 中共享专家的作用，我们禁用它并激活另一个路由专家。对 Pile 的评估显示，即使我们保持相同的计算成本，Pile 损失也显着增加，从 1.808 上升到 2.414。这一结果凸显了共享专家的重要作用，并表明共享专家捕获了路由专家未共享的基础和必要知识，使其成为路由专家不可替代的。

DeepSeekMoE 更准确地获取知识。为了验证我们的主张，即组合激活专家的更高灵活性有助于更准确和更有针对性的知识获取，我们研究了 DeepSeekMoE 是否可以用更少的激活专家获取必要的知识。具体来说，我们将激活的路由专家数量从 3 改为 7，并评估由此产生的桩损失。如图 5 所示，即使只有

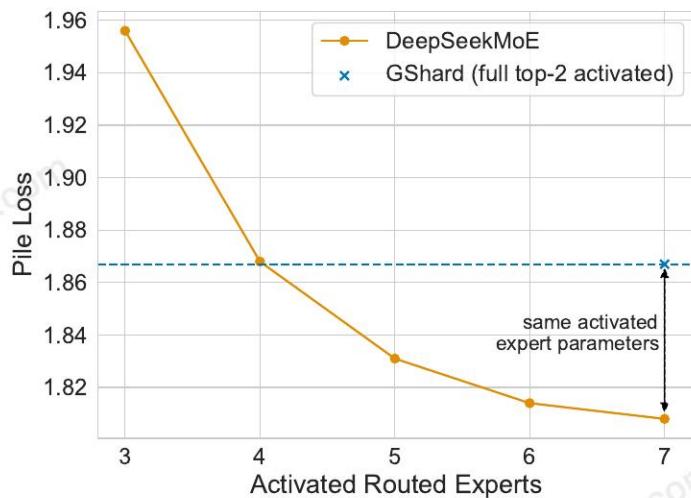


图5 | DeepSeekMoE 中不同数量的激活路由专家的桩损失。仅激活 4 个路由专家，DeepSeekMoE 就实现了与 GShard 相当的桩损失。

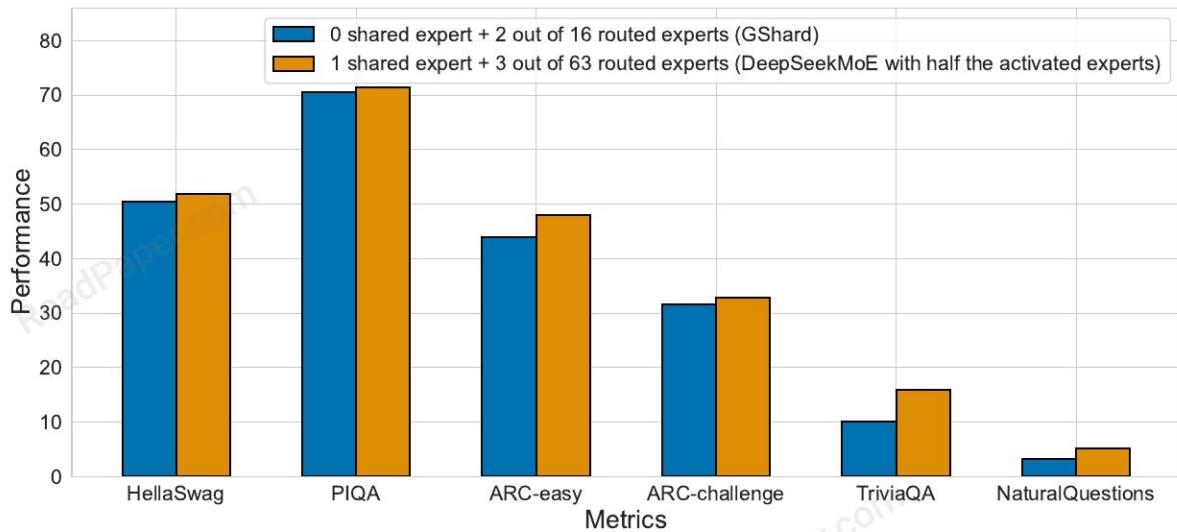


图6 | GShard 和 DeepSeekMoE 与一半激活专家（从头开始训练）的比较。在总专家参数相同且仅激活专家参数一半的情况下，DeepSeekMoE 的性能仍然优于 GShard。

激活 4 个路由专家，DeepSeekMoE 实现了与 GShard 相当的桩损失。这一观察结果支持了 DeepSeekMoE 可以更准确、更有效地获取必要知识的主张。

受这些发现的鼓舞，为了更严格地验证 DeepSeekMoE 的专家专业化和准确的知识获取，我们从头开始训练一个新模型。该模型由 1 个共享专家和 63 个路由专家组成，其中只有 3 个路由专家被激活。图 6 所示的评估结果表明，即使总专家参数相同且激活的专家参数只有一半，DeepSeekMoE 的性能仍然优于 GShard。这凸显了 DeepSeekMoE 利用专家参数的能力。

效率更高，即激活的专家中有效参数的比例远高于GShard。

## 5. 扩展到 DeepSeekMoE 16B

通过 DeepSeekMoE 架构，我们将 MoE 模型扩展到具有 16B 总参数的更大规模，并在 2T 代币上进行训练。我们的结果表明，与 LLaMA2 7B 相比，DeepSeekMoE 16B 只需约 40% 的计算量即可实现卓越的性能。

### 5.1. 实验装置

#### 5.1.1. 训练数据和标记化

我们从第 4.1.1 节中描述的同一语料库中对训练数据进行采样。与验证实验不同，我们使用 2T 令牌采样大量数据，与 LLaMA2 7B 的训练令牌数量保持一致。我们还使用 HuggingFace Tokenizer 工具来训练 BPE 分词器，但 DeepSeekMoE 16B 的词汇量大小设置为 100K。

#### 5.1.2. Hyper-Parameters

模型设置。对于 DeepSeekMoE 16B，我们将 Transformer 层数设置为 28，隐藏维度为 2048。我们采用多头注意力机制，总共 16 个注意力头，其中每个头的维度为 128。至于初始化，所有可学习参数随机初始化，标准差为 0.006。我们用 MoE 层替换除第一层之外的所有 FFN，因为我们观察到第一层的负载平衡状态收敛得特别慢。每个 MoE 层由 2 个共享专家和 64 个路由专家组成，其中每个专家的大小是标准 FFN 的 0.25 倍。每个代币将被路由到这 2 个共享专家以及 64 个路由专家中的 6 个。由于与过小的专家尺寸相关的计算效率可能降低，因此不采用更精细的专家分割粒度。在超过 16B 的较大规模下，仍然可以采用更细的粒度。在我们的配置下，DeepSeekMoE 16B 的总参数约为 16.4B，其中激活的参数数量约为 2.8B。

训练设置。我们采用 AdamW 优化器 (Loshchilov 和 Hutter, 2019)，超参数设置为  $\delta_1 = 0.9$ ,  $\delta_2 = 0.95$ ,  $\text{weight\_decay} = 0.1$ 。学习率也使用预热和逐步衰减策略来安排。最初，学习率在前 2K 步中从 0 线性增加到最大值。随后，学习率在 80% 的训练步骤中乘以 0.316，在 90% 的训练步骤中再次乘以 0.316。DeepSeekMoE 16B 的最大学习率设置为  $4.2 \times 10 \times 4$ ，梯度裁剪范数设置为 1.0。批量大小设置为 4.5K，最大序列长度为 4K，每个训练批量包含 18M 个令牌。相应地，训练总步数设置为 106,449，以实现 2T 训练令牌。由于训练数据丰富，我们在训练时不使用 dropout。我们利用管道并行性将模型的不同层部署在不同的设备上，并且对于每一层，所有专家都将部署在同一设备上。因此，我们在训练期间也不会丢弃任何代币，并且不会采用设备级余额损失。为了防止路由崩溃，我们设置了一个相当小的专家级平衡因子 0.001，因为我们发现在我们的并行化策略下，较高的专家级平衡因子并不能提高计算效率，反而会损害模型性能。

### 5.1.3. 评估基准

除了验证实验中使用的基准之外，我们还结合了其他基准以进行更全面的评估。我们介绍与验证实验中使用的基准的区别如下。

语言建模。对于语言建模，我们还在 Pile 测试集上评估模型 (Gao et al., 2020)。由于 DeepSeekMoE 16B 中使用的分词器与 LLaMA2 7B 中使用的是不同的。为了公平比较，我们使用每字节位数 (BPB) 作为评估指标。

阅读理解。对于阅读理解，我们还考虑了 DROP (Dua et al., 2019)。评估指标是完全匹配 (EM) 率。

数学推理。对于数学推理，我们还结合了 GSM8K (Cobbe 等人, 2021) 和 MATH (Hendrycks 等人, 2021)，使用 EM 作为评估指标。

多科目多项选择。对于多主题多项选择，我们还评估了 MMLU 上的模型 (Hendrycks 等人, 2020)。评价指标是准确性。

消歧义。为了消除歧义，我们还考虑了 Winogrande (Sakaguchi 等人, 2019)，评估指标是准确性。

中国基准。由于 DeepSeekMoE 16B 是在双语语料库上进行预训练的，因此我们还根据四个中文基准对其进行评估。CLUEWSC (Xu et al., 2020) 是中文消歧基准。CEval (Huang et al., 2023) 和 CMMLU (Li et al., 2023) 是两个中文多学科多项选择基准，其形式与 MMLU 类似。CHID (Zheng et al., 2019) 是一个汉语成语完成基准，旨在评估对中国文化的理解。上述中国基准的评估指标是准确性或EM。

打开法学硕士排行榜。我们根据我们的内部评估框架评估所有上述基准。为了公平、方便地将 DeepSeekMoE 16B 与开源模型进行比较，我们另外在 Open LLM 排行榜上对 DeepSeekMoE 16B 进行了评估。Open LLM Leaderboard 是 HuggingFace 支持的公共排行榜，它由六个任务组成：ARC (Clark et al., 2018)、HellaSwag (Zellers et al., 2019)、MMLU (Hendrycks et al., 2020)、TruthfulQA (Lin 等人, 2022)、Winogrande (Sakaguchi 等人, 2019) 和 GSM8K (Cobbe 等人, 2021)。

## 5.2. 评价

### 5.2.1. 与 DeepSeek 7B 的内部比较

我们首先对 DeepSeekMoE 16B 和 DeepSeek 7B (DeepSeek-AI, 2024) 进行内部比较，这是一个具有 6.9B 参数的密集语言模型。为了确保公平性，两个模型都使用 2T 令牌在同一语料库上进行训练。这使得能够准确评估我们的 MoE 架构的有效性，而不受训练数据的影响。

Metric	# Shot	DeepSeek 7B (Dense)	DeepSeekMoE 16B
# Total Params	N/A	6.9B	16.4B
# Activated Params	N/A	6.9B	2.8B
FLOPs per 4K Tokens	N/A	183.5T	74.4T
# Training Tokens	N/A	2T	2T
Pile (BPB)	N/A	0.75	<b>0.74</b>
HellaSwag (Acc.)	0-shot	75.4	<b>77.1</b>
PIQA (Acc.)	0-shot	79.2	<b>80.2</b>
ARC-easy (Acc.)	0-shot	<b>67.9</b>	<b>68.1</b>
ARC-challenge (Acc.)	0-shot	48.1	<b>49.8</b>
RACE-middle (Acc.)	5-shot	<b>63.2</b>	61.9
RACE-high (Acc.)	5-shot	<b>46.5</b>	<b>46.4</b>
DROP (EM)	1-shot	<b>34.9</b>	32.9
GSM8K (EM)	8-shot	17.4	<b>18.8</b>
MATH (EM)	4-shot	3.3	<b>4.3</b>
HumanEval (Pass@1)	0-shot	26.2	<b>26.8</b>
MBPP (Pass@1)	3-shot	<b>39.0</b>	<b>39.2</b>
TriviaQA (EM)	5-shot	59.7	<b>64.8</b>
NaturalQuestions (EM)	5-shot	22.2	<b>25.5</b>
MMLU (Acc.)	5-shot	<b>48.2</b>	45.0
WinoGrande (Acc.)	0-shot	<b>70.5</b>	<b>70.2</b>
CLUEWSC (EM)	5-shot	<b>73.1</b>	72.1
CEval (Acc.)	5-shot	<b>45.0</b>	40.6
CMMU (Acc.)	5-shot	<b>47.2</b>	42.5
CHID (Acc.)	0-shot	<b>89.3</b>	<b>89.4</b>

表 3 | DeepSeek 7B 和 DeepSeekMoE 16B 之间的比较。粗体字体表示最好或接近最好。DeepSeekMoE 16B 仅需要 40.5% 的计算量，即可实现与 DeepSeek 7B 相当的性能。

评估结果如表 3 所示，得到以下观察结果：(1) 总体而言，仅用大约 40% 的计算量，DeepSeekMoE 16B 就达到了与 DeepSeek 7B 相当的性能。(2) DeepSeekMoE 16B 在语言建模和知识密集型任务（例如 Pile、HellaSwag、TriviaQA 和 NaturalQuestions）方面表现出显着的优势。鉴于在 MoE 模型中，FFN 参数比注意力参数重得多，这些结果与变形金刚中的 FFN 表现出知识记忆能力的命题相符 (Dai 等人, 2022a)。(3) 与其他任务上的优异性能相比，DeepSeekMoE 在处理多项选择任务上表现出局限性。这种不足源于 DeepSeekMoE 16B 中的注意力参数有限 (DeepSeekMoE 16B 只有大约 0.5B 个注意力参数，而 DeepSeek 7B 有 2.5B 个注意力参数)。我们早期对 DeepSeek 7B 的研究表明，注意力能力与多项选择任务的表现之间存在正相关关系。例如，配备多查询注意力机制的 DeepSeek 7B MQA (Shazeer, 2019) 在类似 MMLU 的任务中也表现不佳。此外，为了更全面地了解培训过程

DeepSeekMoE 16B，我们还在附录C中提供了DeepSeekMoE 16B和DeepSeek 7B (Dense) 在训练过程中的基准曲线以供参考。

重要的是，由于 DeepSeekMoE 16B 中的参数数量较少，它可以在具有 40GB 内存的 GPU 上实现单设备部署。通过适当的算子优化，它可以实现 7B 密集模型的推理速度近 2.5 倍。

Metric	# Shot	LLaMA2 7B	DeepSeekMoE 16B
# Total Params	N/A	6.7B	16.4B
# Activated Params	N/A	6.7B	2.8B
FLOPs per 4K Tokens	N/A	187.9T	74.4T
# Training Tokens	N/A	2T	2T
Pile (BPB)	N/A	0.76	<b>0.74</b>
HellaSwag (Acc.)	0-shot	75.6	<b>77.1</b>
PIQA (Acc.)	0-shot	78.0	<b>80.2</b>
ARC-easy (Acc.)	0-shot	<b>69.1</b>	68.1
ARC-challenge (Acc.)	0-shot	49.0	<b>49.8</b>
RACE-middle (Acc.)	5-shot	60.7	<b>61.9</b>
RACE-high (Acc.)	5-shot	45.8	<b>46.4</b>
DROP (EM)	1-shot	<b>34.0</b>	32.9
GSM8K (EM)	8-shot	15.5	<b>18.8</b>
MATH (EM)	4-shot	2.6	<b>4.3</b>
HumanEval (Pass@1)	0-shot	14.6	<b>26.8</b>
MBPP (Pass@1)	3-shot	21.8	<b>39.2</b>
TriviaQA (EM)	5-shot	63.8	<b>64.8</b>
NaturalQuestions (EM)	5-shot	<b>25.5</b>	<b>25.5</b>
MMLU (Acc.)	5-shot	<b>45.8</b>	45.0
WinoGrande (Acc.)	0-shot	69.6	<b>70.2</b>
CLUEWSC (EM)	5-shot	64.0	<b>72.1</b>
CEval (Acc.)	5-shot	33.9	<b>40.6</b>
CMMLU (Acc.)	5-shot	32.6	<b>42.5</b>
CHID (Acc.)	0-shot	37.9	<b>89.4</b>

表 4 | LLaMA2 7B 和 DeepSeekMoE 16B 之间的比较。DeepSeekMoE 16B 仅占 39.6% 的计算量，在大多数基准测试中均优于 LLaMA2 7B。

### 5.2.2. 与开源模型的比较

与 LLaMA2 7B 的内部比较。在开源模型领域，我们主要将 DeepSeekMoE 16B 与 LLaMA2 7B (Touvron et al., 2023b) 进行比较，LLaMA2 7B 是一种众所周知的强大开源语言模型，具有 6.7B 参数。DeepSeekMoE 16B 和 LLaMA2 7B 都经过 2T 令牌的预训练。与 LLaMA2 7B 相比，DeepSeekMoE 拥有 245% 的总参数，但只需要 39.6% 的计算量。我们的内部基准测试结果如表 4 所示，得出以下观察结果。（1）在评估的基准中，只有约 40% 的计算量，DeepSeekMoE 16B 在大多数基准上都优于 LLaMA2 7B。（2）DeepSeekMoE 16B 的数学推理和代码生成能力

比 LLaMA2 7B 更强，这归因于我们的预训练语料库中数学和代码相关文本的丰富存在。（3）鉴于我们的预训练语料库中存在中文文本，DeepSeekMoE 16B 在中文基准测试中比 LLaMA2 7B 表现出显着的性能优势。（4）尽管接受的英语文本训练较少，但 DeepSeekMoE 16B 在英语理解或知识密集型基准测试上与 LLaMA2 7B 相比取得了相当或更好的性能，这表明了 DeepSeekMoE 16B 的卓越能力。

Open LLM 排行榜评估。除了内部评估之外，我们还在 Open LLM Leaderboard 上评估 DeepSeekMoE 16B，并将其与其他开源模型进行比较。除了 LLaMA 7B 之外，我们还考虑了更广泛的开源模型，包括 LLaMA 7B (Touvron 等人, 2023a)、Falcon 7B (Almazrouei 等人, 2023)、GPT-J 6B (Wang 和 Komatsuzaki, 2021)、RedPajama-INCITE 7B 和 3B (Together-AI, 2023)、Open LLaMA 7B 和 3B (Geng 和 Liu, 2023)、OPT 2.7B (Zhang 等人, 2022)、Pythia 2.8B (Biderman 等人, 2022)、GPT-neo 2.7B (Black 等人, 2021) 和 BLOOM 3B (Scao 等人, 2022)。如图 1 所示，评估结果表明 DeepSeekMoE 16B 始终大幅优于具有相似激活参数的模型。此外，它的性能与 LLaMA2 7B 相当，后者的激活参数约为 2.5 倍。

## 6. Alignment for DeepSeekMoE 16B

先前的研究表明，MoE 模型通常不会通过微调获得显着收益 (Artetxe 等人, 2022 年; Fedus 等人, 2021 年)。然而，沉等人。(2023) 的研究结果表明 MoE 模型确实可以从指令调整中受益。为了评估 DeepSeekMoE 16B 是否可以从微调中受益，我们进行监督微调以构建基于 DeepSeekMoE 16B 的聊天模型。实验结果表明，DeepSeekMoE Chat 16B 也实现了与 LLaMA2 SFT 7B 和 DeepSeek Chat 7B 相当的性能。

### 6.1. 实验装置

训练数据。为了训练聊天模型，我们对内部精选数据（包括 140 万个训练样本）进行监督微调 (SFT)。该数据集涵盖了广泛的类别，包括数学、代码、写作、问答、推理、总结等。我们的 SFT 训练数据大部分是英文和中文，使得聊天模型具有通用性并适用于双语场景。

超参数。在监督微调期间，我们将批量大小设置为 1024 个示例，并使用 AdamW 优化器进行 8 个时期的训练 (Loshchilov 和 Hutter, 2019)。我们采用最大序列长度为 4K，并尽可能密集地打包训练样例，直到达到序列长度限制。我们不使用 dropout 进行监督微调，只是将学习率设置为  $10^{-5}$ ，没有结合任何学习率调度策略。

**Evaluation Benchmarks.** For the evaluation of the chat models, we employ benchmarks similar to those used in Section 5.1.3, with the following adjustments: (1) We exclude Pile (Gao et al., 2020) since chat models are seldom employed for pure language modeling. (2) We exclude

CHID (Zheng et al., 2019) due to the observed instability of results, hindering the derivation of solid conclusions. (3) We additionally include BBH (Suzgun et al., 2022) to provide a more comprehensive assessment of the reasoning ability of the chat models.

Metric	# Shot	LLaMA2 SFT 7B	DeepSeek Chat 7B	DeepSeekMoE Chat 16B
# Total Params	N/A	6.7B	6.9B	16.4B
# Activated Params	N/A	6.7B	6.9B	2.8B
FLOPs per 4K Tokens	N/A	187.9T	183.5T	74.4T
HellaSwag (Acc.)	0-shot	67.9	71.0	<b>72.2</b>
PIQA (Acc.)	0-shot	76.9	78.4	<b>79.7</b>
ARC-easy (Acc.)	0-shot	69.7	<b>70.2</b>	<b>69.9</b>
ARC-challenge (Acc.)	0-shot	<b>50.8</b>	50.2	50.0
BBH (EM)	3-shot	39.3	<b>43.1</b>	42.2
RACE-middle (Acc.)	5-shot	63.9	<b>66.1</b>	64.8
RACE-high (Acc.)	5-shot	49.6	<b>50.8</b>	<b>50.6</b>
DROP (EM)	1-shot	40.0	<b>41.7</b>	33.8
GSM8K (EM)	0-shot	<b>63.4</b>	62.6	62.2
MATH (EM)	4-shot	13.5	14.7	<b>15.2</b>
HumanEval (Pass@1)	0-shot	35.4	45.1	<b>45.7</b>
MBPP (Pass@1)	3-shot	27.8	39.0	<b>46.2</b>
TriviaQA (EM)	5-shot	60.1	59.5	<b>63.3</b>
NaturalQuestions (EM)	0-shot	<b>35.2</b>	32.7	<b>35.1</b>
MMLU (Acc.)	0-shot	<b>50.0</b>	49.7	47.2
WinoGrande (Acc.)	0-shot	65.1	68.4	<b>69.0</b>
CLUEWSC (EM)	5-shot	48.4	66.2	<b>68.2</b>
CEval (Acc.)	0-shot	35.1	<b>44.7</b>	40.0
CMMLU (Acc.)	0-shot	36.9	<b>51.2</b>	49.3

表 5 | LLaMA2 SFT 7B、DeepSeek Chat 7B 和 DeepSeekMoE Chat 16B 之间的比较，所有这三个模型都在相同的 SFT 数据上进行了微调。与两个 7B 密集模型相比，DeepSeekMoE Chat 16B 在大多数基准测试中仍然实现了相当或更好的性能，而计算量仅为 40%。

## 6. 2. 评价

基线。为了验证对齐后 DeepSeekMoE 16B 的潜力，我们对 LLaMA2 7B、DeepSeek 7B 和 DeepSeekMoE 16B 进行监督微调，其中我们使用完全相同的微调数据以确保公平性。相应地，我们构建了三个聊天模型，包括 LLaMA2 SFT 7B 3、DeepSeek Chat 7B 和 DeepSeekMoE Chat 16B。随后，我们在广泛的下游任务中将 DeepSeekMoE Chat 16B 与其他两种密集聊天模型（大约 2.5 倍的 FLOP）进行比较。

<sup>3</sup>We use LLaMA2 SFT to distinguish from the official LLaMA2 Chat (Touvron et al., 2023b) model.

结果。评估结果如表 5 所示。我们的主要观察结果包括：(1) DeepSeekMoE Chat 16B 虽然消耗了近 40% 的计算量，但在语言理解和推理 (PIQA、ARC、BBH) 方面实现了与 7B 密集模型相当的性能、机器阅读理解 (RACE)、数学 (GSM8K、MATH) 和知识密集型任务 (TriviaQA、NaturalQuestions)。(2) 在代码生成任务上，DeepSeekMoE Chat 16B 显著优于 LLaMA2 SFT 7B，展示了在 HumanEval 和 MBPP 上的显著改进。此外，它还超越了 DeepSeek Chat 7B。(3) 在包括 MMLU、CEval 和 CMMLU 在内的多项选择题回答基准上，DeepSeekMoE Chat 16B 仍然落后于 DeepSeek Chat 7B，这与基本模型的观察结果一致 (第 5.2.1 节)。然而，值得注意的是，经过监督微调后，DeepSeekMoE 16B 和 DeepSeek 7B 之间的性能差距缩小了。(4) 受益于双语语料库的预训练，DeepSeekMoE Chat 16B 在所有中文基准上均明显优于 LLaMA2 SFT 7B。这些结果证明了 DeepSeekMoE 16B 中英文的平衡能力，增强了其在不同场景下的通用性和适用性。总之，对聊天模型的评估凸显了 DeepSeekMoE 16B 从对齐中受益的潜力，并验证了其在仅使用约 40% 的计算量的情况下实现与密集模型相当的性能的一贯优势。

## 7. DeepSeekMoE 145B Ongoing

受到 DeepSeekMoE 16B 出色性能的鼓舞，我们进一步初步努力将 DeepSeekMoE 扩展到 145B。在这项初步研究中，DeepSeekMoE 145B 在 245B 令牌上进行训练，但它表现出了优于 GShard 架构的一致优势，并有望达到或超过 DeepSeek 67B (密集) 的性能。此外，在 DeepSeekMoE 145B 的最终版本和完整训练完成后，我们还计划将其公开。

### 7.1. 实验装置

训练数据和标记化。对于 DeepSeekMoE 145B，我们采用与 DeepSeekMoE 16B 完全相同的训练语料库和标记器，唯一的区别是 DeepSeekMoE 145B 在初始研究中使用 245B 标记进行训练。

模型设置。对于 DeepSeekMoE 145B，我们将 Transformer 层数设置为 62，隐藏维度设置为 4096。我们采用多头注意力机制，总共 32 个注意力头，其中每个头的维度为 128。至于初始化，所有可学习参数随机初始化，标准差为 0.006。与 DeepSeekMoE 16B 一样，我们也用 MoE 层替换除第一层之外的所有 FFN。每个 MoE 层由 4 个共享专家和 128 个路由专家组成，其中每个专家的大小是标准 FFN 的 0.125 倍。每个代币将被路由到这 4 位共享专家以及 128 个路由专家中的 12 位。在此配置下，DeepSeekMoE 145 的总参数约为 144.6B，激活参数的数量约为 22.2B。

训练设置。我们采用 AdamW 优化器 (Loshchilov 和 Hutter, 2019)，超参数设置为  $\delta_1 = 0.9$ ,  $\delta_2 = 0.95$ ,  $\text{weight\_decay} = 0.1$ 。对于 DeepSeekMoE 145B 的初步研究，我们采用了预热和恒定学习率调度程序。最初，学习率在前 2K 步中从 0 线性增加到最大值。

随后，学习率在剩余的训练过程中保持恒定。DeepSeekMoE 145B 的最大学习率设置为  $3.0 \times 10 \times 4$ ，梯度裁剪范数设置为 1.0。批量大小设置为 4.5K，最大序列长度为 4K，每个训练批量包含 18M 个令牌。我们训练 DeepSeekMoE 145B 13,000 步，获得 245B 训练令牌。此外，我们在训练期间不使用 dropout。我们利用管道并行性将模型的不同层部署在不同的设备上，对于每一层，所有路由的专家将统一部署在 4 台设备上（即专家并行性与数据并行性相结合）。由于我们对 DeepSeekMoE 145B 采用专家并行，因此应考虑设备级负载平衡以减少计算瓶颈。作为回应，我们将设备级平衡因子设置为 0.05，以鼓励跨设备平衡计算。另外，我们仍然设置了一个小的专家级平衡因子 0.003，以防止路由崩溃。

评估基准。我们在与 DeepSeekMoE 16B 所用的完全相同的内部基准上评估 DeepSeekMoE 145B（参见第 5.1.3 节）。

## 7.2. 评价

基线。除了 DeepSeekMoE 145B 之外，我们还考虑了另外三个模型进行比较。DeepSeek 67B (Dense) 是一个密集模型，总参数为 67.4B（模型和训练详细信息请参阅 DeepSeek-AI (2024)）。GShard 137B 与 DeepSeekMoE 145B 具有相同的隐藏维度和层数，但遵循 GShard 架构。请注意，为了计算效率，DeepSeekMoE 145B 将每个 Expert 中的中间隐藏维度对齐为 64 的倍数，因此其模型大小比 GShard 137B 大 6%。DeepSeekMoE 142B (半激活) 具有与 DeepSeekMoE 145B 类似的架构，但它仅包含 2 个共享专家，并且 128 个路由专家中只有 6 个被激活。值得注意的是，所有比较模型（包括 DeepSeekMoE 145B）都共享相同的训练语料库。此外，比较中的所有 MoE 模型都是从头开始训练的，并共享相同的训练超参数。

结果。从表 6 中呈现的评估结果，我们得到以下观察结果：(1) 尽管具有可比的总参数和计算量，DeepSeekMoE 145B 显著优于 GShard 137B，再次凸显了 DeepSeekMoE 架构的优势。(2) 总体而言，DeepSeekMoE 145B 仅用 28.5% 的计算量就达到了与 DeepSeek 67B (Dense) 相当的性能。与 DeepSeekMoE 16B 的研究结果一致，DeepSeekMoE 145B 在语言建模和知识密集型任务中表现出显著的优势，但在多项选择任务中存在局限性。(3) 在更大的规模上，DeepSeekMoE 142B (半激活) 的性能并没有落后于 DeepSeekMoE 145B 太多。此外，尽管只有一半的激活专家参数，DeepSeekMoE 142B (半激活) 仍然与 DeepSeek 67B (密集) 的性能相匹配，计算量仅为 18.2%。它还优于 GShard 137B，这与第 4.5 节的结论一致。

## 八、相关工作

专家混合 (MoE) 技术首先由 Jacobs 等人提出。(1991)；Jordan 和 Jacobs (1994) 使用独立的专家模块处理不同的样本。沙泽尔等人。(2017) 将 MoE 引入语言模型训练中，并构建基于 LSTM 的大规模 (Hochreiter 和 Schmidhuber, 1997) MoE 模型。随着 Transformer 成为最流行的架构

Metric	# Shot	DeepSeek 67B (Dense)	GShard 137B	DeepSeekMoE 145B	DeepSeekMoE 142B (Half Activated)
# Total Params	N/A	67.4B	136.5B	144.6B	142.3B
# Activated Params	N/A	67.4B	21.6B	22.2B	12.2B
Relative Expert Size	N/A	N/A	1	0.125	0.125
# Experts	N/A	N/A	0 + 16	4 + 128	2 + 128
# Activated Experts	N/A	N/A	0 + 2	4 + 12	2 + 6
FLOPs per 4K Tokens	N/A	2057.5T	572.7T	585.6T	374.6T
# Training Tokens	N/A	245B	245B	245B	245B
Pile (Loss.)	N/A	1.905	1.961	<b>1.876</b>	1.888
HellaSwag (Acc.)	0-shot	74.8	72.0	<b>75.8</b>	74.9
PIQA (Acc.)	0-shot	79.8	77.6	<b>80.7</b>	80.2
ARC-easy (Acc.)	0-shot	69.0	64.0	<b>69.7</b>	67.9
ARC-challenge (Acc.)	0-shot	<b>50.4</b>	45.8	48.8	49.0
RACE-middle (Acc.)	5-shot	<b>63.2</b>	59.2	62.1	59.5
RACE-high (Acc.)	5-shot	<b>46.9</b>	43.5	45.5	42.6
DROP (EM)	1-shot	<b>27.5</b>	21.6	<b>27.8</b>	28.9
GSM8K (EM)	8-shot	<b>11.8</b>	6.4	<b>12.2</b>	13.8
MATH (EM)	4-shot	2.1	1.6	<b>3.1</b>	2.8
HumanEval (Pass@1)	0-shot	<b>23.8</b>	17.7	19.5	23.2
MBPP (Pass@1)	3-shot	<b>33.6</b>	27.6	<b>33.2</b>	32.0
TriviaQA (EM)	5-shot	57.2	52.5	<b>61.1</b>	59.8
NaturalQuestions (EM)	5-shot	22.6	19.0	<b>25.0</b>	23.5
MMLU (Acc.)	5-shot	<b>45.1</b>	26.3	39.4	37.5
WinoGrande (Acc.)	0-shot	70.7	67.6	<b>71.9</b>	70.8
CLUEWSC (EM)	5-shot	69.1	65.7	<b>71.9</b>	72.6
Ceval (Acc.)	5-shot	<b>40.3</b>	26.2	37.1	32.8
CMMLU (Acc.)	5-shot	<b>40.6</b>	25.4	35.9	31.9
CHID (Acc.)	0-shot	88.5	86.9	<b>90.3</b>	88.3

表 6 | DeepSeek 67B (Dense) 和 MoE 模型在总参数约 140B 规模下的比较。在“#专家”和“#激活专家”行中，“+”分别表示“共享专家”和“路由专家”。粗体字体表示除最后一列之外的最佳或接近最佳性能。DeepSeekMoE 145B，甚至只有一半激活专家参数的 DeepSeekMoE 142B（半激活），其性能大幅优于 GShard 137B。此外，DeepSeekMoE 145B 的计算量为 28.5%，其性能与 DeepSeek 67B 相当。

对于 NLP，许多尝试将 Transformer 中的 FFN 扩展为 MoE 层来构建 MoE 语言模型。GShard (Lepikhin et al., 2021) 和 Switch Transformer (Fedus et al., 2021) 是采用可学习的 top-2 或 top-1 路由策略将 MoE 语言模型扩展到极大的规模的先驱。Hash Layer (Roller et al., 2021) 和 StableMoE (Dai et al., 2022b) 使用固定路由策略来实现更稳定的路由和训练。周等人。(2022) 提出了一种专家选择路由策略，其中每个令牌可以分配给不同数量的专家。Zoph (2022) 重点关注 MoE 模型中训练不稳定和微调困难的问题，

并建议 ST-MoE 克服这些挑战。除了MoE架构和训练策略的研究之外，近年来还出现了大量大规模语言或多模态模型（Du et al., 2022; Lin et al., 2021; Ren et al., 2023; Xu et al., 2023） et al., 2023）基于现有的 MoE 架构。总体而言，之前的MoE模型大多基于传统的top-1或top-2路由策略，为提高专家专业化留下了很大的空间。对此，我们的DeepSeekMoE架构旨在最大限度地提高专家的专业化程度。

## 9. 结论

在本文中，我们介绍了用于 MoE 语言模型的 DeepSeekMoE 架构，其目标是实现最终的专家专业化。通过细粒度的专家分割和共享的专家隔离，与主流的 MoE 架构相比，DeepSeekMoE 实现了显着更高的专家专业化和性能。从适度规模的 2B 参数开始，我们验证了 DeepSeekMoE 的优势，展示了其接近 MoE 模型性能上限的能力。此外，我们提供的经验证据表明 DeepSeekMoE 比 GShard 具有更高水平的专家专业化。

扩展到 16B 总参数的更大规模，我们在 2T 代币上训练 DeepSeekMoE 16B，并展示了其与 DeepSeek 7B 和 LLaMA2 7B 相当的出色性能，计算量仅为约 40%。此外，基于DeepSeekMoE 16B进行监督微调对齐，构建了MoE聊天模型，进一步展示了其适应性和通用性。此外，我们进行了初步探索，将 DeepSeek-MoE 扩展到 145B 参数。我们发现 DeepSeekMoE 145B 仍然保持了相对 GShard 架构的显着优势，并且表现出与 DeepSeek 67B 相当的性能，仅使用 28.5%（甚至可能是 18.2%）的计算量。

出于研究目的，我们向公众发布了 DeepSeekMoE 16B 的模型检查点，该模型检查点可以部署在具有 40GB 内存的单个 GPU 上。我们渴望这项工作为学术界和工业界提供有价值的见解，并为大规模语言模型的加速发展做出贡献。

## References

- E. Almazrouei, H. Alobeidli, A. Alshamsi, A. Cappelli, R. Cojocaru, M. Debbah, E. Goffinet, D. Heslow, J. Launay, Q. Malartic, B. Noune, B. Pannier, and G. Penedo. Falcon-40B: an open large language model with state-of-the-art performance, 2023.
- M. Artetxe, S. Bhosale, N. Goyal, T. Mihaylov, M. Ott, S. Shleifer, X. V. Lin, J. Du, S. Iyer, R. Pasunuru, G. Anantharaman, X. Li, S. Chen, H. Akin, M. Baines, L. Martin, X. Zhou, P. S. Koura, B. O'Horo, J. Wang, L. Zettlemoyer, M. T. Diab, Z. Kozareva, and V. Stoyanov. Efficient large scale language modeling with mixtures of experts. In Y. Goldberg, Z. Kozareva, and Y. Zhang, editors, *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, EMNLP 2022, Abu Dhabi, United Arab Emirates, December 7-11, 2022*, pages 11699–11732. Association for Computational Linguistics, 2022. doi: 10.18653/V1/2022.EMNLP-MAIN.804. URL <https://doi.org/10.18653/v1/2022.emnlp-main.804>.
- J. Austin, A. Odena, M. Nye, M. Bosma, H. Michalewski, D. Dohan, E. Jiang, C. Cai, M. Terry, Q. Le, et al. Program synthesis with large language models. *arXiv preprint arXiv:2108.07732*, 2021.

S. Biderman, H. Schoelkopf, Q. G. Anthony, H. Bradley, K. O'Brien, E. Hallahan, M. A. Khan, S. Purohit, U. S. Prashanth, E. Raff, A. Skowron, L. Sutawika, and O. van der Wal. Pythia: A suite for analyzing large language models across training and scaling. In A. Krause, E. Brunskill, K. Cho, B. Engelhardt, S. Sabato, and J. Scarlett, editors, *International Conference on Machine Learning, ICML 2023, 23-29 July 2023, Honolulu, Hawaii, USA*, volume 202 of *Proceedings of Machine Learning Research*, pages 2397–2430. PMLR, 2023. URL <https://proceedings.mlr.press/v202/biderman23a.html>.

Y. Bisk、R. Zellers、R. L. Bras、J. Gao 和 Y. Choi。PIQA：用自然语言推理物理常识。第三十四届 AAAI 人工智能大会，AAAI 2020，第三十二届人工智能创新应用大会，IAAI 2020，第十届 AAAI 人工智能教育进展研讨会，EAAI 2020，美国纽约州纽约市，2 月 2020 年 7-12 日，第 7432 页至 7439 页。AAAI 出版社，2020。doi:

10.1609/aaai.v34i05.6239. URL <https://doi.org/10.1609/aaai.v34i05.6239>.

S. Black, L. Gao, P. Wang, C. Leahy, and S. Biderman. GPT-Neo: Large Scale Autoregressive Language Modeling with Mesh-Tensorflow, Mar. 2021. URL <https://doi.org/10.5281/zenodo.5297715>. If you use this misc, please cite it using these metadata.

T. B. Brown, B. Mann, N. Ryder, M. Subbiah, J. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, S. Agarwal, A. Herbert-Voss, G. Krueger, T. Henighan, R. Child, A. Ramesh, D. M. Ziegler, J. Wu, C. Winter, C. Hesse, M. Chen, E. Sigler, M. Litwin, S. Gray, B. Chess, J. Clark, C. Berner, S. McCandlish, A. Radford, I. Sutskever, and D. Amodei. Language models are few-shot learners. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020*, 2020. URL <https://proceedings.neurips.cc/paper/2020/hash/1457c0d6bfcb4967418bfb8ac142f64a-Abstract.html>.

M. Chen, J. Tworek, H. Jun, Q. Yuan, H. P. de Oliveira Pinto, J. Kaplan, H. Edwards, Y. Burda, N. Joseph, G. Brockman, A. Ray, R. Puri, G. Krueger, M. Petrov, H. Khlaaf, G. Sastry, P. Mishkin, B. Chan, S. Gray, N. Ryder, M. Pavlov, A. Power, L. Kaiser, M. Bavarian, C. Winter, P. Tillet, F. P. Such, D. Cummings, M. Plappert, F. Chantzis, E. Barnes, A. Herbert-Voss, W. H. Guss, A. Nichol, A. Paino, N. Tezak, J. Tang, I. Babuschkin, S. Balaji, S. Jain, W. Saunders, C. Hesse, A. N. Carr, J. Leike, J. Achiam, V. Misra, E. Morikawa, A. Radford, M. Knight, M. Brundage, M. Murati, K. Mayer, P. Welinder, B. McGrew, D. Amodei, S. McCandlish, I. Sutskever, and W. Zaremba. Evaluating large language models trained on code. *CoRR*, abs/2107.03374, 2021. URL <https://arxiv.org/abs/2107.03374>.

P. Clark, I. Cowhey, O. Etzioni, T. Khot, A. Sabharwal, C. Schoenick, and O. Tafjord. Think you have solved question answering? try arc, the AI2 reasoning challenge. *CoRR*, abs/1803.05457, 2018. URL <http://arxiv.org/abs/1803.05457>

K. Cobbe, V. Kosaraju, M. Bavarian, M. Chen, H. Jun, L. Kaiser, M. Plappert, J. Tworek, J. Hilton, R. Nakano, et al. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*, 2021.

D. Dai, L. Dong, Y. Ha, Z. Sui, B. Chang, 和 F. Wei。预训练 Transformer 中的知识神经元。S. Muresan、P. Nakov 和 A. Villavicencio, 编辑, 计算语言学协会第 60 届年会论文集 (第一卷: 长论文), ACL 2022, 爱尔兰都柏林, 2022 年 5 月 22-27 日, 第 8493 页至 8502 页。计算协会

Linguistics, 2022a. doi: 10.18653/V1/2022.ACL-LONG.581. URL <https://doi.org/10.18653/v1/2022.acl-long.581>.

D. Dai、L. Dong、S. Ma、B. Cheng、Z. Sui、B. Chang 和 F. Wei。Stablemoe: 专家混合的稳定路由策略。S. Muresan、P. Nakov 和 A. Villavicencio, 编辑, 计算语言学协会第 60 届年会论文集(第一卷: 长论文), ACL 2022, 爱尔兰都柏林, 2022 年 5 月 22-27 日, 第 7085 页至 7095 页。计算语言学协会, 2022b。doi: 10.18653/v1/2022.acl-long.489。

URL <https://doi.org/10.18653/v1/2022.acl-long.489>.

DeepSeek-AI. Deepseek llm: Scaling open-source language models with longtermism. *arXiv preprint arXiv:2401.02954*, 2024.

N. Du, Y. Huang, A. M. Dai, S. Tong, D. Lepikhin, Y. Xu, M. Krikun, Y. Zhou, A. W. Yu, O. Firat, B. Zoph, L. Fedus, M. P. Bosma, Z. Zhou, T. Wang, Y. E. Wang, K. Webster, M. Pellat, K. Robinson, K. S. Meier-Hellstern, T. Duke, L. Dixon, K. Zhang, Q. V. Le, Y. Wu, Z. Chen, and C. Cui. Glam: Efficient scaling of language models with mixture-of-experts. In *International Conference on Machine Learning, ICML 2022, 17-23 July 2022, Baltimore, Maryland, USA*, volume 162 of *Proceedings of Machine Learning Research*, pages 5547–5569. PMLR, 2022.

网址 <https://proceedings.mlr.press/v162/du22c.html>.

D. Dua、Y. Wang、P. Dasigi、G. Stanovsky、S. Singh 和 M. Gardner。DROP: 阅读理解基准, 需要对段落进行离散推理。J. Burstein、C. Doran 和 T. Solorio, 编辑, 2019 年计算语言学协会北美分会会议记录: 人类语言技术, NAACL-HLT 2019, 美国明尼苏达州明尼阿波利斯, 6 月 2 日-7, 2019, 第 1 卷(长论文和短论文), 第 2368-2378 页。计算语言学协会, 2019。doi: 10.18653/V1/N19-1246。网址

<https://doi.org/10.18653/v1/n19-1246>

W. Fedus, B. Zoph, and N. Shazeer. Switch transformers: Scaling to trillion parameter models with simple and efficient sparsity. *CoRR*, abs/2101.03961, 2021. URL <https://arxiv.org/abs/2101.03961>.

L. Gao, S. Biderman, S. Black, L. Golding, T. Hoppe, C. Foster, J. Phang, H. He, A. Thite, N. Nabeshima, et al. The Pile: An 800GB dataset of diverse text for language modeling. *arXiv preprint arXiv:2101.00027*, 2020.

X. Geng and H. Liu. Openllama: An open reproduction of llama, May 2023. URL [https://github.com/openlm-research/open\\_llama](https://github.com/openlm-research/open_llama).

A. Harlap, D. Narayanan, A. Phanishayee, V. Seshadri, N. R. Devanur, G. R. Ganger, and P. B. Gibbons. Pipedream: Fast and efficient pipeline parallel DNN training. *CoRR*, abs/1806.03377, 2018. URL <http://arxiv.org/abs/1806.03377>

D. Hendrycks, C. Burns, S. Basart, A. Zou, M. Mazeika, D. Song, and J. Steinhardt. Measuring massive multitask language understanding. *arXiv preprint arXiv:2009.03300*, 2020.

D. Hendrycks, C. Burns, S. Kadavath, A. Arora, S. Basart, E. Tang, D. Song, and J. Steinhardt. Measuring mathematical problem solving with the math dataset, 2021.

High-Flyer. Hai-llm: An efficient and lightweight tool for training large models, 2023. URL <https://www.high-flyer.cn/en/blog/hai-llm>.

S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural Computing*, 9(8):1735–1780, 1997. URL <https://doi.org/10.1162/neco.1997.9.8.1735>

- J. Hoffmann, S. Borgeaud, A. Mensch, E. Buchatskaya, T. Cai, E. Rutherford, D. de Las Casas, L. A. Hendricks, J. Welbl, A. Clark, T. Hennigan, E. Noland, K. Millican, G. van den Driessche, B. Damoc, A. Guy, S. Osindero, K. Simonyan, E. Elsen, J. W. Rae, O. Vinyals, and L. Sifre. Training compute-optimal large language models. *CoRR*, abs/2203.15556, 2022. doi: 10.48550 /arXiv.2203.15556. URL <https://doi.org/10.48550/arXiv.2203.15556>
- Y. Huang, Y. Bai, Z. Zhu, J. Zhang, J. Zhang, T. Su, J. Liu, C. Lv, Y. Zhang, J. Lei, et al. C-Eval: A multi-level multi-discipline chinese evaluation suite for foundation models. *arXiv preprint arXiv:2305.08322*, 2023.
- R. A. Jacobs, M. I. Jordan, S. J. Nowlan, and G. E. Hinton. Adaptive mixtures of local experts. *Neural Computing*, 3(1):79–87, 1991. URL <https://doi.org/10.1162/neco.1991.3.1.79>
- M. I. Jordan and R. A. Jacobs. Hierarchical mixtures of experts and the EM algorithm. *Neural Computing*, 6(2):181–214, 1994. URL <https://doi.org/10.1162/neco.1994.6.2.181>
- M. Joshi, E. Choi, D. Weld, and L. Zettlemoyer. triviaqa: A Large Scale Distantly Supervised Challenge Dataset for Reading Comprehension. *arXiv e-prints*, art. arXiv:1705.03551, 2017.
- V. A. Korthikanti, J. Casper, S. Lym, L. McAfee, M. Andersch, M. Shoeybi, and B. Catanzaro. Reducing activation recomputation in large transformer models. *Proceedings of Machine Learning and Systems*, 5, 2023.
- T. Kwiatkowski, J. Palomaki, O. Redfield, M. Collins, A. Parikh, C. Alberti, D. Epstein, I. Polosukhin, M. Kelcey, J. Devlin, K. Lee, K. N. Toutanova, L. Jones, M.-W. Chang, A. Dai, J. Uszkoreit, Q. Le, and S. Petrov. Natural questions: a benchmark for question answering research. *Transactions of the Association of Computational Linguistics*, 2019.
- G. Lai、Q. Xie、H. Liu、Y. Yang 和 E. H. Hovy。 RACE: 来自考试的大规模阅读理解数据集。 M. Palmer、R. Hwa 和 S. Riedel, 编辑, 《2017 年自然语言处理经验方法会议论文集》, EMNLP 2017, 丹麦哥本哈根, 2017 年 9 月 9-11 日, 第 785 – 794 页。计算协会 Linguistics, 2017. doi: 10.18653/V1/D17-1082. URL <https://doi.org/10.18653/v1/d17-1082>
- D. Lepikhin, H. Lee, Y. Xu, D. Chen, O. Firat, Y. Huang, M. Krikun, N. Shazeer, and Z. Chen. Gshard: Scaling giant models with conditional computation and automatic sharding. In *9th International Conference on Learning Representations, ICLR 2021*. OpenReview.net, 2021. 网址 <https://openreview.net/forum?id=qrwe7XHTmYb>.
- H. Li, Y. Zhang, F. Koto, Y. Yang, H. Zhao, Y. Gong, N. Duan, and T. Baldwin. CMMLU: Measuring massive multitask language understanding in Chinese. *arXiv preprint arXiv:2306.09212*, 2023.
- J. Lin, R. Men, A. Yang, C. Zhou, M. Ding, Y. Zhang, P. Wang, A. Wang, L. Jiang, X. Jia, J. Zhang, J. Zhang, X. Zou, Z. Li, X. Deng, J. Liu, J. Xue, H. Zhou, J. Ma, J. Yu, Y. Li, W. Lin, J. Zhou, J. Tang, and H. Yang. M6: A chinese multimodal pretrainer. *CoRR*, abs/2103.00823, 2021. URL <https://arxiv.org/abs/2103.00823>
- S. Lin, J. Hilton, and O. Evans. Truthfulqa: Measuring how models mimic human falsehoods. In S. Muresan, P. Nakov, and A. Villavicencio, editors, *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, ACL 2022, Dublin,

Ireland, May 22-27, 2022, pages 3214–3252. Association for Computational Linguistics, 2022. doi: 10.18653/V1/2022.ACL-LONG.229. URL <https://doi.org/10.18653/v1/2022.acl-long.229>.

- I. Loshchilov and F. Hutter. Decoupled weight decay regularization. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net, 2019. URL <https://openreview.net/forum?id=Bkg6RiCqY7>.
- D. Narayanan, M. Shoeybi, J. Casper, P. LeGresley, M. Patwary, V. Korthikanti, D. Vainbrand, P. Kashinkunti, J. Bernauer, B. Catanzaro, et al. Efficient large-scale language model training on gpu clusters using megatron-lm. In *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis*, pages 1–15, 2021.
- OpenAI. GPT-4 technical report. *CoRR*, abs/2303.08774, 2023. doi: 10.48550/arXiv.2303.08774. URL <https://doi.org/10.48550/arXiv.2303.08774>.
- S. Rajbhandari、J. Rasley、O. Ruwase 和 Y. He。零：针对训练万亿参数模型的内存优化。 C. Cuicchi、I. Qualters 和 W. T. Kramer 编辑，高性能计算、网络、存储和分析国际会议论文集，SC 2020，虚拟活动/美国佐治亚州亚特兰大，2020 年 11 月 9–19 日，第 20 页。
- IEEE/ACM, 2020. doi: 10.1109/SC41405.2020.00024. URL <https://doi.org/10.1109/SC41405.2020.00024>.
- S. Rajbhandari, C. Li, Z. Yao, M. Zhang, R. Y. Aminabadi, A. A. Awan, J. Rasley, and Y. He. Deepspeed-moe: Advancing mixture-of-experts inference and training to power next-generation AI scale. In K. Chaudhuri, S. Jegelka, L. Song, C. Szepesvári, G. Niu, and S. Sabato, editors, *International Conference on Machine Learning, ICML 2022, 17-23 July 2022, Baltimore, Maryland, USA*, volume 162 of *Proceedings of Machine Learning Research*, pages 18332–18346. PMLR, 2022. URL <https://proceedings.mlr.press/v162/rajbhandari22a.html>.
- X. Ren, P. Zhou, X. Meng, X. Huang, Y. Wang, W. Wang, P. Li, X. Zhang, A. Podolskiy, G. Arshinov, A. Bout, I. Piontkovskaya, J. Wei, X. Jiang, T. Su, Q. Liu, and J. Yao. Pangu- $\Sigma$ : Towards trillion parameter language model with sparse heterogeneous computing. *CoRR*, abs/2303.10845, 2023. URL <https://doi.org/10.48550/arXiv.2303.10845>.
- S. Roller, S. Sukhbaatar, A. Szlam, and J. Weston. Hash layers for large sparse models. *CoRR*, abs/2106.04426, 2021. URL <https://arxiv.org/abs/2106.04426>.
- K. Sakaguchi, R. L. Bras, C. Bhagavatula, and Y. Choi. Winogrande: An adversarial winograd schema challenge at scale, 2019.
- T. L. Scao, A. Fan, C. Akiki, E. Pavlick, S. Ilic, D. Hesslow, R. Castagné, A. S. Luccioni, F. Yvon, M. Gallé, J. Tow, A. M. Rush, S. Biderman, A. Webson, P. S. Ammanamanchi, T. Wang, B. Sagot, N. Muennighoff, A. V. del Moral, O. Ruwase, R. Bawden, S. Bekman, A. McMillan-Major, I. Beltagy, H. Nguyen, L. Saulnier, S. Tan, P. O. Suarez, V. Sanh, H. Laurençon, Y. Jernite, J. Launay, M. Mitchell, C. Raffel, A. Gokaslan, A. Simhi, A. Soroa, A. F. Aji, A. Alfassy, A. Rogers, A. K. Nitzav, C. Xu, C. Mou, C. Emezue, C. Klamm, C. Leong, D. van Strien, D. I. Adelani, and et al. BLOOM: A 176b-parameter open-access multilingual language model. *CoRR*, abs/2211.05100, 2022. doi: 10.48550/ARXIV.2211.05100. URL <https://doi.org/10.48550/arXiv.2211.05100>.

- R. Sennrich, B. Haddow, and A. Birch. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, ACL 2016, August 7-12, 2016, Berlin, Germany, Volume 1: Long Papers*. The Association for Computer Linguistics, 2016. doi: 10.18653/V1/P16-1162. URL <https://doi.org/10.18653/v1/p16-1162>.
- N. Shazeer. Fast transformer decoding: One write-head is all you need. *CoRR*, abs/1911.02150, 2019. URL <http://arxiv.org/abs/1911.02150>
- N. Shazeer, A. Mirhoseini, K. Maziarz, A. Davis, Q. V. Le, G. E. Hinton, and J. Dean. Outrageously large neural networks: The sparsely-gated mixture-of-experts layer. In *5th International Conference on Learning Representations, ICLR 2017*. OpenReview.net, 2017. URL <https://openreview.net/forum?id=B1ckMDqlg>
- S. Shen, L. Hou, Y. Zhou, N. Du, S. Longpre, J. Wei, H. W. Chung, B. Zoph, W. Fedus, X. Chen, T. Vu, Y. Wu, W. Chen, A. Webson, Y. Li, V. Zhao, H. Yu, K. Keutzer, T. Darrell, and D. Zhou. Flan-moe: Scaling instruction-finetuned language models with sparse mixture of experts. *CoRR*, abs/2305.14705, 2023. doi: 10.48550/ARXIV.2305.14705. URL <https://doi.org/10.48550/arXiv.2305.14705>
- M. Shoeybi, M. Patwary, R. Puri, P. LeGresley, J. Casper, and B. Catanzaro. Megatron-lm: Training multi-billion parameter language models using model parallelism. *arXiv preprint arXiv:1909.08053*, 2019.
- M. Suzgun, N. Scales, N. Schärli, S. Gehrmann, Y. Tay, H. W. Chung, A. Chowdhery, Q. V. Le, E. H. Chi, D. Zhou, et al. Challenging big-bench tasks and whether chain-of-thought can solve them. *arXiv preprint arXiv:2210.09261*, 2022.
- P. Tillet、H. T. Kung 和 D. Cox。Triton：一种用于平铺神经网络计算的中间语言和编译器。第三届 ACM SIGPLAN 国际机器学习和编程语言研讨会论文集，MAPL 2019，第 10 - 19 页，美国纽约州纽约市，2019 年。计算机协会。ISBN 9781450367196. doi: 10.1145/3315508.3329973
- 一起-AI。Redpajama-data：重现 llama 训练数据集的开源配方，四月 2023. URL <https://github.com/togethercomputer/RedPajama-Data>
- H. Touvron, T. Lavril, G. Izacard, X. Martinet, M. Lachaux, T. Lacroix, B. Rozière, N. Goyal, E. Hambro, F. Azhar, A. Rodriguez, A. Joulin, E. Grave, and G. Lample. Llama: Open and efficient foundation language models. *CoRR*, abs/2302.13971, 2023a. doi: 10.48550/arXiv.2302.13971. URL <https://doi.org/10.48550/arXiv.2302.13971>
- H. Touvron, L. Martin, K. Stone, P. Albert, A. Almahairi, Y. Babaei, N. Bashlykov, S. Batra, P. Bhargava, S. Bhosale, D. Bikell, L. Blecher, C. Canton-Ferrer, M. Chen, G. Cucurull, D. Esiobu, J. Fernandes, J. Fu, W. Fu, B. Fuller, C. Gao, V. Goswami, N. Goyal, A. Hartshorn, S. Hosseini, R. Hou, H. Inan, M. Kardas, V. Kerkez, M. Khabsa, I. Kloumann, A. Korenev, P. S. Koura, M. Lachaux, T. Lavril, J. Lee, D. Liskovich, Y. Lu, Y. Mao, X. Martinet, T. Mihaylov, P. Mishra, I. Molybog, Y. Nie, A. Poulton, J. Reizenstein, R. Rungta, K. Saladi, A. Schelten, R. Silva, E. M. Smith, R. Subramanian, X. E. Tan, B. Tang, R. Taylor, A. Williams, J. X. Kuan, P. Xu, Z. Yan, I. Zarov, Y. Zhang, A. Fan, M. Kambadur, S. Narang, A. Rodriguez, R. Stojnic, S. Edunov, and T. Scialom. Llama 2: Open foundation and fine-tuned chat models. *CoRR*, abs/2307.09288, 2023b. doi: 10.48550/arXiv.2307.09288. URL <https://doi.org/10.48550/arXiv.2307.09288>

- A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017*, pages 5998–6008, 2017. URL <https://proceedings.neurips.cc/paper/2017/hash/3f5ee243547dee91fb053c1c4a845aa-Abstract.html>.
- B. Wang and A. Komatsuzaki. GPT-J-6B: A 6 Billion Parameter Autoregressive Language Model. <https://github.com/kingoflolz/mesh-transformer-jax>, May 2021.
- L. Xu, H. Hu, X. Zhang, L. Li, C. Cao, Y. Li, Y. Xu, K. Sun, D. Yu, C. Yu, Y. Tian, Q. Dong, W. Liu, B. Shi, Y. Cui, J. Li, J. Zeng, R. Wang, W. Xie, Y. Li, Y. Patterson, Z. Tian, Y. Zhang, H. Zhou, S. Liu, Z. Zhao, Q. Zhao, C. Yue, X. Zhang, Z. Yang, K. Richardson, and Z. Lan. CLUE: A Chinese language understanding evaluation benchmark. In D. Scott, N. Bel, and C. Zong, editors, *Proceedings of the 28th International Conference on Computational Linguistics, COLING 2020*, Barcelona, Spain (Online), December 8–13, 2020, pages 4762–4772. International Committee on Computational Linguistics, 2020. doi: 10.18653/V1/2020.COLING-MAIN.419. URL <https://doi.org/10.18653/v1/2020.coling-main.419>
- F. Xue, Z. Zheng, Y. Fu, J. Ni, Z. Zheng, W. Zhou, and Y. You. Openmoe: Open mixture-of-experts language models. <https://github.com/XueFuzhao/OpenMoE>, 2023.
- R. Zellers、A. Holtzman、Y. Bisk、A. Farhadi 和 Y. Choi。HellaSwag: 机器真的能完成你的句子吗? A. Korhonen、D. R. Traum 和 L. Márquez, 编辑, 计算语言学协会第 57 届会议论文集, ACL 2019, 意大利佛罗伦萨, 2019 年 7 月 28 日至 8 月 2 日, 第 1 卷: 长论文, 页数4791–4800。计算协会  
Linguistics, 2019. doi: 10.18653/v1/p19-1472. URL <https://doi.org/10.18653/v1/p19-1472>
- S. Zhang, S. Roller, N. Goyal, M. Artetxe, M. Chen, S. Chen, C. Dewan, M. Diab, X. Li, X. V. Lin, T. Mihaylov, M. Ott, S. Shleifer, K. Shuster, D. Simig, P. S. Koura, A. Sridhar, T. Wang, and L. Zettlemoyer. Opt: Open pre-trained transformer language models, 2022.
- C. 郑、M. 黄和 A. 孙。Chid: 用于完形填空测试的大规模汉语成语数据集。A. Korhonen、D. R. Traum 和 L. Márquez, 编辑, 计算语言学协会第 57 届会议论文集, ACL 2019, 意大利佛罗伦萨, 2019 年 7 月 28 日至 8 月 2 日, 第 1 卷: 长论文, 页数778—787。计算语言学协会, 2019。  
doi: 10.18653/V1/P19-1075. URL <https://doi.org/10.18653/v1/p19-1075>
- Y. Zhou, T. Lei, H. Liu, N. Du, Y. Huang, V. Zhao, A. M. Dai, Z. Chen, Q. V. Le, and J. Laudon. Mixture-of-experts with expert choice routing. In NeurIPS, 2022. URL [http://papers.nips.cc/paper\\_files/paper/2022/hash/2f00ecd787b432c1d36f3de9800728eb-Abstract-Conference.html](http://papers.nips.cc/paper_files/paper/2022/hash/2f00ecd787b432c1d36f3de9800728eb-Abstract-Conference.html)
- B. Zoph. Designing effective sparse expert models. In *IEEE International Parallel and Distributed Processing Symposium, IPDPS Workshops 2022*, Lyon, France, May 30 - June 3, 2022, page 1044. IEEE, 2022. URL <https://doi.org/10.1109/IPDPSW55747.2022.00171>

## Appendices

### A. 超参数概述

我们在表 7 中概述了各种尺寸的 DeepSeekMoE 的超参数。

# Params	# Layers	Hidden Size	# Attn Heads	# Shared Experts	# Routed Experts	Relative Expert Size	Sequence Length	Batch Size (Sequence)	Learning Rate
2.0B	9	1280	10	1	63 (7 activated)	0.25	2048	2048	1.08e-3
16.4B	28	2048	16	2	64 (6 activated)	0.25	4096	4608	4.2e-4
144.6B	62	4096	32	4	128 (12 activated)	0.125	4096	4608	3.0e-4

表 7 | 各种尺寸的 DeepSeekMoE 超参数概述。相对专家规模是与标准 FFN 进行比较。

### B. Comparing DeepSeekMoE with Larger Models

DeepSeekMoE、GShard 1.2 和 GShard 1.5 之间的比较如表 8 所示。DeepSeekMoE、Dense 4 和 Dense 16 之间的比较如表 9 所示。

Metric	# Shot	GShard×1.2	GShard×1.5	DeepSeekMoE
Relative Expert Size	N/A	1.2	1.5	0.25
# Experts	N/A	0 + 16	0 + 16	1 + 63
# Activated Experts	N/A	0 + 2	0 + 2	1 + 7
# Total Expert Params	N/A	2.3B	2.8B	1.9B
# Activated Expert Params	N/A	0.28B	0.35B	0.24B
# Training Tokens	N/A	100B	100B	100B
Pile (Loss)	N/A	1.824	<b>1.808</b>	<b>1.808</b>
HellaSwag (Acc.)	0-shot	53.7	54.4	<b>54.8</b>
PIQA (Acc.)	0-shot	71.8	71.1	<b>72.3</b>
ARC-easy (Acc.)	0-shot	46.8	47.3	<b>49.4</b>
ARC-challenge (Acc.)	0-shot	31.7	<b>34.1</b>	<b>34.3</b>
RACE-middle (Acc.)	5-shot	43.7	<b>46.4</b>	44.0
RACE-high (Acc.)	5-shot	31.9	<b>32.4</b>	31.7
HumanEval (Pass@1)	0-shot	3.7	3.0	<b>4.9</b>
MBPP (Pass@1)	3-shot	2.4	<b>2.6</b>	2.2
TriviaQA (EM)	5-shot	15.2	15.7	<b>16.6</b>
NaturalQuestions (EM)	5-shot	4.5	4.7	<b>5.7</b>

表 8 | DeepSeekMoE 和更大的 GShard 模型之间的比较。

在 13B 总参数的更大范围内，我们还将 DeepSeekMoE 与 GShard-1.2 和 GShard-1.5 进行比较，结果如表 10 所示。在更大的范围内，DeepSeekMoE 甚至明显优于 GShard-1.5。

Metric	# Shot	Dense×4	Dense×16	DeepSeekMoE
Relative Expert Size	N/A	1	1	0.25
# Experts	N/A	4 + 0	16 + 0	1 + 63
# Activated Experts	N/A	4 + 0	16 + 0	1 + 7
# Total Expert Params	N/A	0.47B	1.89B	1.89B
# Activated Expert Params	N/A	0.47B	1.89B	0.24B
# Training Tokens	N/A	100B	100B	100B
Pile (Loss)	N/A	1.908	<b>1.806</b>	<b>1.808</b>
HellaSwag (Acc.)	0-shot	47.6	<b>55.1</b>	<b>54.8</b>
PIQA (Acc.)	0-shot	70.0	71.9	<b>72.3</b>
ARC-easy (Acc.)	0-shot	43.9	<b>51.9</b>	49.4
ARC-challenge (Acc.)	0-shot	30.5	33.8	<b>34.3</b>
RACE-middle (Acc.)	5-shot	42.4	<b>46.3</b>	44.0
RACE-high (Acc.)	5-shot	30.7	<b>33.0</b>	31.7
HumanEval (Pass@1)	0-shot	1.8	4.3	<b>4.9</b>
MBPP (Pass@1)	3-shot	0.2	<b>2.2</b>	<b>2.2</b>
TriviaQA (EM)	5-shot	9.9	<b>16.5</b>	<b>16.6</b>
NaturalQuestions (EM)	5-shot	3.0	<b>6.3</b>	5.7

表 9 | DeepSeekMoE 和更大的密集基线之间的比较。

Metric	# Shot	GShard×1.2	GShard×1.5	DeepSeekMoE
Relative Expert Size	N/A	1.2	1.5	0.25
# Experts	N/A	0 + 16	0 + 16	1 + 63
# Activated Experts	N/A	0 + 2	0 + 2	1 + 7
# Total Expert Params	N/A	15.9B	19.8B	13.3B
# Activated Expert Params	N/A	2.37B	2.82B	2.05B
# Training Tokens	N/A	100B	100B	100B
HellaSwag (Acc.)	0-shot	66.6	67.7	<b>69.1</b>
PIQA (Acc.)	0-shot	75.6	<b>76.0</b>	<b>75.7</b>
ARC-easy (Acc.)	0-shot	56.8	56.8	<b>58.8</b>
ARC-challenge (Acc.)	0-shot	<b>39.9</b>	37.6	38.5
RACE-middle (Acc.)	5-shot	51.6	50.6	<b>52.4</b>
RACE-high (Acc.)	5-shot	37.4	36.3	<b>38.5</b>
HumanEval (Pass@1)	0-shot	6.1	6.1	<b>9.8</b>
MBPP (Pass@1)	3-shot	7.0	<b>11.6</b>	10.6
TriviaQA (EM)	5-shot	36.5	36.7	<b>38.2</b>
NaturalQuestions (EM)	5-shot	12.6	12.1	<b>13.7</b>

表 10 | DeepSeekMoE 和更大的 GShard 模型在更大规模上的比较。

### C. Training Benchmark Curves of DeepSeekMoE 16B

我们在图 7 中展示了 DeepSeekMoE 16B 和 DeepSeek 7B (Dense) 训练期间的基准曲线以供参考。

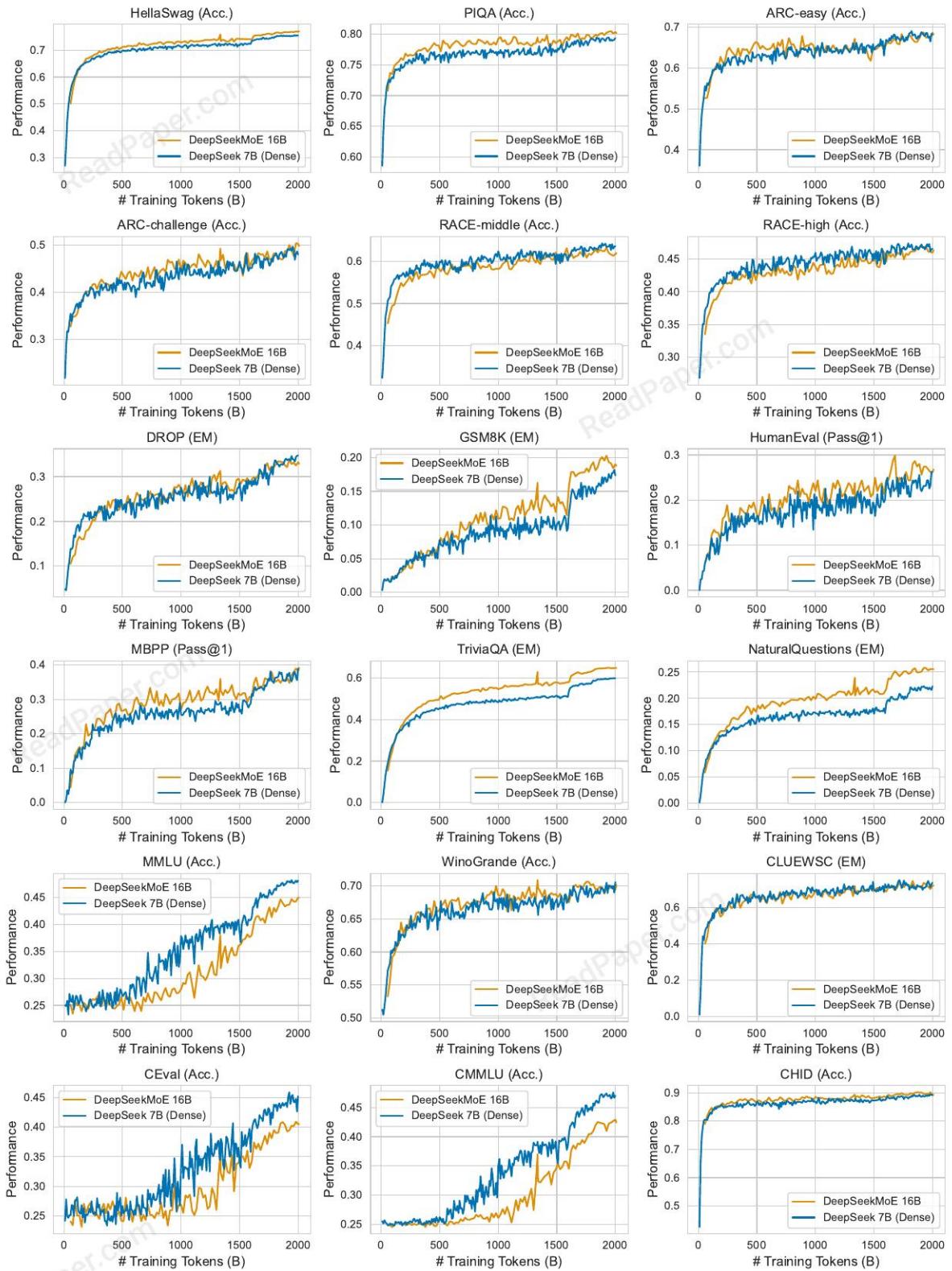


图 7 | DeepSeekMoE 16B 和 DeepSeek 7B (密集) 训练期间的基准曲线。