DeepSeekMoE: Towards Ultimate Expert Specialization in Mixture-of-Experts Language Models

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

[1]**DeepSeek-AI**
[2]**National Key Laboratory for Multimedia Information Processing, Peking University**
[3]**Institute for Interdisciplinary Information Sciences, Tsinghua University**
[4]**National Key Laboratory for Novel Software Technology, Nanjing University**
{daidamai, szf}@pku.edu.cn, {wenfeng.liang}@deepseek.com
https://github.com/deepseek-ai/DeepSeek-MoE

Abstract

在大语言模型时代，混合专家（Mixture-of-Experts, MoE）架构在扩展模型参数时管理计算成本方面展现出巨大潜力。然而，传统的 MoE 架构（如 GShard，其从 N 个专家中激活 top-K 个）在确保专家专业化方面面临挑战，即每个专家获取不重叠且专注的知识。为此，我们提出了 **DeepSeekMoE** 架构，旨在实现极致的专家专业化。该架构包含两项核心策略：(1) 将专家细分为 mN 个，并从中激活 mK 个，从而实现激活专家更灵活的组合；(2) 隔离 K_s 个专家作为共享专家，旨在捕捉通用知识并缓解路由专家中的冗余。从 2B 参数的较小规模起步，我们证明了 DeepSeekMoE 2B 的性能可与 GShard 2.9B 相媲美，而后者拥有 1.5× 的专家参数和计算量。此外，DeepSeekMoE 2B 的性能几乎接近具有相同总参数量的稠密模型，后者设定了 MoE 模型的性能上限。随后，我们将 DeepSeekMoE 扩展至 16B 参数，并表明其仅使用约 40% 的计算量即可达到与 LLaMA2 7B 相当的性能。进一步地，我们将 DeepSeekMoE 扩展至 145B 参数的初步尝试持续验证了其相对于 GShard 架构的显著优势，并表明其仅使用 28.5%（甚至可能低至 18.2%）的计算量即可达到与 DeepSeek 67B 相当的性能。

1. Introduction

近期的研究与实践已经验证表明，在训练数据充足的情况下，通过增加参数和计算预算来扩展语言模型，能够显著提升模型性能 (Brown et al., 2020; Hoffmann et al., 2022; OpenAI, 2023; Touvron et al., 2023a)。然而，必须承认的是，将模型扩展至极大规模的努力也伴随着极高的计算成本。鉴于高昂的成本，混合专家（Mixture-of-Experts, MoE）架构 (Jacobs et al., 1991;

*Contribution during internship at DeepSeek-AI.

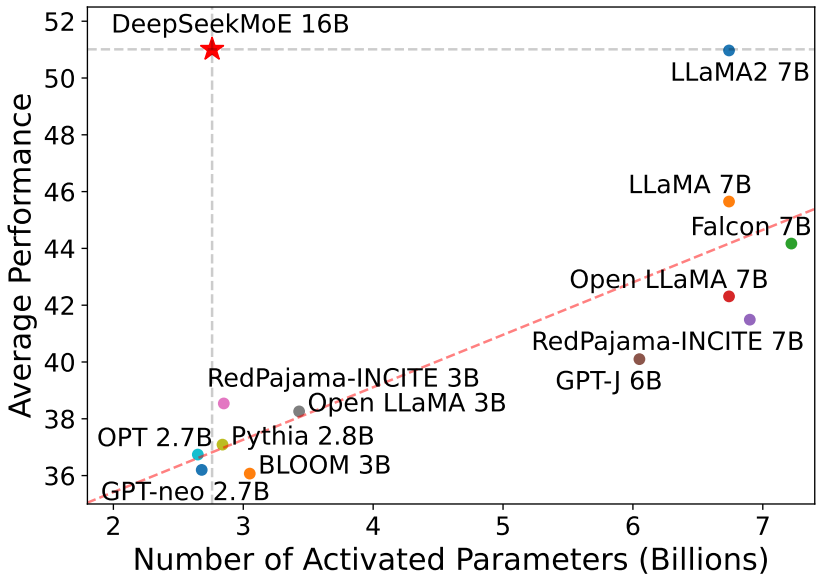

图 1 | DeepSeekMoE 16B 与开源模型在 Open LLM Leaderboard 上的对比。红色虚线为除 DeepSeekMoE 16B 外所有模型数据点的线性拟合结果。DeepSeekMoE 16B 持续大幅优于激活参数量相近的模型，并达到了与激活参数量约为其 2.5 倍的 LLaMA2 7B 相当的性能。

Jordan and Jacobs, 1994; Shazeer et al., 2017) 已成为一种广受欢迎的解决方案。该架构能够在实现参数扩展的同时，将计算成本维持在较低水平。近年来，MoE 架构在 Transformer (Vaswani et al., 2017) 中的应用已成功将语言模型扩展至较大规模 (Du et al., 2022; Fedus et al., 2021; Lepikhin et al., 2021; Zoph, 2022)，并取得了卓越的性能。这些成果凸显了 MoE 语言模型的巨大潜力与前景。

尽管 MoE 架构前景广阔，但现有架构可能面临知识混杂与知识冗余的问题，这限制了专家专业化程度，即每个专家获取不重叠且专注的知识。传统的 MoE 架构通常用 MoE 层替换 Transformer 中的前馈神经网络（Feed-Forward Networks, FFNs）。每个 MoE 层由多个专家组成，每个专家的结构与标准 FFN 相同，且每个 token 会被分配给一个 (Fedus et al., 2021) 或两个 (Lepikhin et al., 2021) 专家。该架构暴露出两个潜在问题：(1) **知识混杂**：现有 MoE 实践通常采用数量有限的专家（例如 8 或 16 个），因此分配给特定专家的 token 很可能涵盖多样化的知识。因此，指定专家将试图在其参数中整合截然不同的知识类型，而这些知识难以同时被有效利用。(2) **知识冗余**：分配给不同专家的 token 可能需要通用知识。因此，多个专家可能会在其各自参数中趋同地学习共享知识，从而导致专家参数冗余。这些问题共同阻碍了现有 MoE 实践中的专家专业化，使其无法达到 MoE 模型的理论性能上限。

针对上述问题，我们引入了 **DeepSeekMoE**，这是一种专为实现极致专家专业化而设计的创新 MoE 架构。我们的架构包含两项核心策略：(1) **细粒度专家分割：**在保持参数量不变的前提下，我们通过分割 FFN 中间隐藏维度将专家划分为更细的粒度。相应地，在保持计算成本不变的情况下，我们也激活更多细粒度专家，以实现激活专家更灵活、适应性更强的组合。细粒度专家分割使得多样化知识能够被更精细地分解，并更精确地学习至不同专家中，从而使每个专

家保持更高的专业化水平。此外，激活专家组合灵活性的提升也有助于实现更准确、更有针对性的知识获取。(2) **共享专家隔离：**我们隔离部分专家作为始终激活的共享专家，旨在捕捉并整合不同上下文中的通用知识。通过将通用知识压缩至这些共享专家中，可以缓解其他路由专家之间的冗余。这有助于提升参数效率，并确保每个路由专家通过专注于独特方面而保持专业化。DeepSeekMoE 中的这些架构创新为训练参数高效且每个专家高度专业化的 MoE 语言模型提供了契机。

从 2B 参数的较小规模起步，我们验证了 DeepSeekMoE 架构的优势。我们在涵盖多样化任务的 12 个零样本或少样本基准上进行了评估。实证结果表明，DeepSeekMoE 2B 大幅超越了 GShard 2B (Lepikhin et al., 2021)，甚至与拥有 1.5× 专家参数和计算量的更大规模 MoE 模型 GShard 2.9B 性能相当。值得注意的是，我们发现 DeepSeekMoE 2B 的性能几乎接近具有等效参数量的稠密模型，后者设定了 MoE 语言模型的严格性能上限。为深入探究，我们对 DeepSeekMoE 的专家专业化进行了详尽的消融实验与分析。这些研究验证了细粒度专家分割与共享专家隔离的有效性，并为 DeepSeekMoE 能够实现高水平专家专业化的论断提供了实证支持。

基于我们的架构，我们随后将模型参数扩展至 16B，并在包含 2T tokens 的大规模语料库上训练了 DeepSeekMoE 16B。评估结果显示，仅使用约 40% 的计算量，DeepSeekMoE 16B 即可达到与在同一 2T 语料库上训练的稠密模型 DeepSeek 7B (DeepSeek-AI, 2024) 相当的性能。我们还将 DeepSeekMoE 与开源模型进行了对比，评估结果表明 DeepSeekMoE 16B 持续大幅优于激活参数量相近的模型，并达到了与激活参数量约为其 2.5 倍的 LLaMA2 7B (Touvron et al., 2023b) 相当的性能。图 1 展示了在 Open LLM Leaderboard[1] 上的评估结果。此外，我们进行了监督微调（Supervised Fine-Tuning, SFT）以实现对齐，将模型转化为对话模型。评估结果显示，在对话设置下，DeepSeekMoE Chat 16B 的性能也与 DeepSeek Chat 7B 和 LLaMA2 SFT 7B 相当。受这些结果的鼓舞，我们进一步开展了将 DeepSeekMoE 扩展至 145B 的初步尝试。实验结果持续验证了其相对于 GShard 架构的显著优势。此外，它仅使用 28.5%（甚至可能低至 18.2%）的计算量即展现出与 DeepSeek 67B 相当的性能。

我们的贡献总结如下：

- **架构创新**。我们引入了 DeepSeekMoE，这是一种旨在实现极致专家专业化的创新 MoE 架构，采用了细粒度专家分割与共享专家隔离两项核心策略。
- **实证验证**。我们进行了大量实验以实证验证 DeepSeekMoE 架构的有效性。实验结果证实了 DeepSeekMoE 2B 的高水平专家专业化，并表明 DeepSeekMoE 2B 几乎能够接近 MoE 模型的性能上限。
- **可扩展性**。我们将 DeepSeekMoE 扩展至 16B 模型进行训练，并表明仅使用约 40% 的计算量，DeepSeekMoE 16B 即可达到与 DeepSeek 7B 和 LLaMA2 7B 相当的性能。我们还开展了将 DeepSeekMoE 扩展至 145B 的初步尝试，凸显了其相对于 GShard 架构的持续优势，并展现出与 DeepSeek 67B 相当的性能。

[1] https://huggingface.co/spaces/HuggingFaceH4/open_llm_leaderboard

- **MoE 对齐**。我们成功对 DeepSeekMoE 16B 进行了监督微调，构建了对齐的对话模型，展现了 DeepSeekMoE 16B 的适应性与多功能性。
- **公开发布**。秉承开放研究的精神，我们向公众发布了 DeepSeekMoE 16B 的模型检查点。值得注意的是，该模型无需量化即可在单张 40GB 显存的 GPU 上部署。

2. Preliminaries: Mixture-of-Experts for Transformers

我们首先介绍 Transformer 语言模型中常用的一种通用 MoE 架构。标准的 Transformer 语言模型通过堆叠 L 层标准 Transformer 块构建，其中每个块可表示为：

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

$$\mathbf{h}_{t}^{l} = \text{FFN}\left(\mathbf{u}_{t}^{l}\right) + \mathbf{u}_{t}^{l}, \tag{2}$$

其中 T 表示序列长度，Self-Att$(\cdot)$ 表示自注意力模块，FFN$(\cdot)$ 表示前馈神经网络（FFN），$\mathbf{u}_{1:T}^{l} \in \mathbb{R}^{T \times d}$ 为经过第 l 个注意力模块后所有 token 的隐藏状态，$\mathbf{h}_{t}^{l} \in \mathbb{R}^{d}$ 为经过第 l 个 Transformer 块后第 t 个 token 的输出隐藏状态。为简洁起见，上述公式中省略了层归一化操作。

构建 MoE 语言模型的典型做法通常是在指定间隔处用 MoE 层替换 Transformer 中的 FFN (Du et al., 2022; Fedus et al., 2021; Lepikhin et al., 2021; Zoph, 2022)。MoE 层由多个专家组成，每个专家的结构与标准 FFN 相同。随后，每个 token 将被分配给一个 (Fedus et al., 2021) 或两个 (Lepikhin et al., 2021) 专家。若将第 l 个 FFN 替换为 MoE 层，其输出隐藏状态 $\mathbf{h}_{t}^{l}$ 的计算可表示为：

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

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

$$s_{i,t} = \text{Softmax}_{i}\left(\mathbf{u}_{t}^{l^{T}} \mathbf{e}_{i}^{l}\right), \tag{5}$$

其中 N 表示专家总数，FFN$_i(\cdot)$ 为第 i 个专家 FFN，$g_{i,t}$ 表示第 i 个专家的门控值，$s_{i,t}$ 表示 token 与专家的亲和力，Topk$(\cdot, K)$ 表示由第 t 个 token 与所有 N 个专家计算出的亲和力得分中前 K 高得分组成的集合，$\mathbf{e}_{i}^{l}$ 为第 l 层中第 i 个专家的质心。注意，$g_{i,t}$ 是稀疏的，表明 N 个门控值中仅有 K 个非零。这种稀疏性保证了 MoE 层内的计算效率，即每个 token 仅会被分配至并在 K 个专家中进行计算。同样，为简洁起见，上述公式中省略了层归一化操作。

3. DeepSeekMoE Architecture

在第二节概述的通用 MoE 架构基础上，我们引入了 DeepSeekMoE，该架构专为挖掘专家专业化潜力而设计。如图 2 所示，我们的架构包含两项核心策略：细粒度专家分割与共享专家隔离。

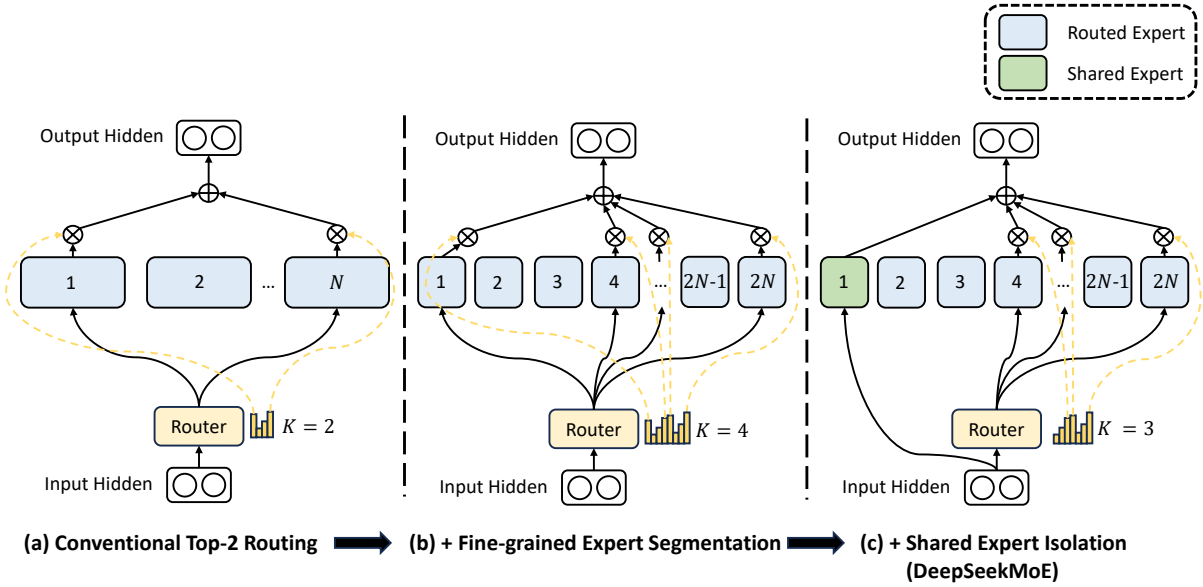

图 2 | DeepSeekMoE 架构示意图。子图 (a) 展示了采用传统 top-2 路由策略的 MoE 层。子图 (b) 说明了细粒度专家分割策略。随后，子图 (c) 演示了共享专家隔离策略的集成，构成了完整的 DeepSeekMoE 架构。值得注意的是，在这三种架构中，专家参数数量与计算成本均保持不变。

这两项策略均旨在提升专家专业化水平。

3.1. Fine-Grained Expert Segmentation

在专家数量有限的场景中，分配给特定专家的 token 更可能涵盖多样化的知识类型。因此，指定专家将试图在其参数中学习截然不同的知识类型，而这些知识难以被同时有效利用。然而，若每个 token 能够被路由至更多专家，多样化知识将有机会被分解并分别学习至不同专家中。在此情境下，每个专家仍能保持高水平的专业化，从而促进专家间知识分布的更加专注。

为实现该目标，我们在保持专家参数总量与计算成本一致的前提下，对专家进行更细粒度的分割。更细的专家分割使得激活专家的组合更加灵活且适应性更强。具体而言，在图 2(a) 所示的典型 MoE 架构基础上，我们通过将 FFN 中间隐藏维度缩减至原始大小的 $\frac{1}{m}$，将每个专家 FFN 分割为 m 个更小的专家。由于每个专家规模变小，相应地，我们将激活专家的数量增加至 m 倍以维持相同的计算成本，如图 2(b) 所示。采用细粒度专家分割后，MoE 层的输出可表示为：

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

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

$$s_{i,t} = \mathrm{Softmax}_i\left(\mathbf{u}_t^{l^T} \mathbf{e}_i^l\right), \tag{8}$$

其中，专家参数的总数等于标准 FFN 参数数量的 N 倍，mN 表示细粒度专家的总数。采用细粒度专家分割策略后，非零门控的数量也将增加至 mK。

从组合学的角度来看，细粒度专家分割策略显著提升了激活专家的组合灵活性。以示例说明，我们考虑 $N = 16$ 的情况。典型的 Top-2 路由策略可产生 $\binom{16}{2} = 120$ 种可能的组合。相比之下，若将每个专家拆分为 4 个更小的专家，细粒度路由策略则可产生 $\binom{64}{8} = 4,426,165,368$ 种潜在组合。组合灵活性的激增提升了实现更精准、更具针对性知识获取的潜力。

3.2. 共享专家隔离

在传统路由策略下，分配给不同专家的 token 可能需要某些公共知识或信息。因此，多个专家可能会在其各自参数中趋同地学习共享知识，从而导致专家参数冗余。然而，若存在专门用于捕捉和整合不同上下文中公共知识的共享专家，则其他路由专家之间的参数冗余将得到缓解。这种冗余的缓解将有助于构建参数效率更高、专家专业化程度更强的模型。

为实现这一目标，除细粒度专家分割策略外，我们进一步隔离出 K_s 个专家作为共享专家。无论路由模块如何工作，每个 token 都将确定性地分配给这些共享专家。为保持计算成本恒定，其他路由专家中激活的专家数量将减少 K_s 个，如图 2(c) 所示。结合共享专家隔离策略后，完整 DeepSeekMoE 架构中的 MoE 层可表述如下：

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

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

$$s_{i,t} = \mathrm{Softmax}_i\left(\mathbf{u}_t^{l^T} \mathbf{e}_i^l\right). \tag{11}$$

最后，在 DeepSeekMoE 中，共享专家的数量为 K_s，路由专家的总数为 $mN - K_s$，非零门控的数量为 $mK - K_s$。

值得注意的是，共享专家隔离的雏形可归功于 Rajbhandari et al. (2022)。关键区别在于，他们是从工程角度推导该策略，而我们是基于算法视角进行探讨。

3.3. 负载均衡考量

自动学习的路由策略可能会遇到负载不均衡问题，该问题主要表现为两个显著缺陷。首先，存在路由崩溃的风险 (Shazeer et al., 2017)，即模型始终仅选择少数专家，导致其他专家无法得到充分训练。其次，若专家分布在多个设备上，负载不均衡会加剧计算瓶颈。

专家级平衡损失。 为缓解路由崩溃的风险，我们还采用了专家级平衡损失。该平衡损失的计算方式如下：

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

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

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

其中，α_1 为称为专家级平衡因子的超参数，为简洁起见，N' 等于 $(mN - K_s)$，K' 等于 $(mK - K_s)$。$\mathbb{1}(\cdot)$ 表示指示函数。

设备级平衡损失。 除专家级平衡损失外，我们还引入了设备级平衡损失。在旨在缓解计算瓶颈时，无需在专家层面施加严格的平衡约束，因为过度的负载均衡约束会损害模型性能。相反，我们的主要目标是确保各设备间的计算负载均衡。若我们将所有路由专家划分为 D 个组 $\{\mathcal{E}_1, \mathcal{E}_2, ..., \mathcal{E}_D\}$，并将每组部署在单个设备上，则设备级平衡损失计算如下：

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

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

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

其中，α_2 为称为设备级平衡因子的超参数。在实际应用中，我们设置较小的专家级平衡因子以缓解路由崩溃风险，同时设置较大的设备级平衡因子以促进设备间的计算负载均衡。

4. 验证实验

4.1. 实验设置

4.1.1. 训练数据与分词

我们的训练数据采样自由 DeepSeek-AI 构建的大规模多语言语料库。该语料库主要以英语和中文为主，但也涵盖其他语言。其数据来源多样，包括网页文本、数学资料、代码脚本、已发表文献以及各种其他文本材料。为了进行验证实验，我们从该语料库中采样了一个包含 100B 词元的子集来训练我们的模型。在分词方面，我们使用 HuggingFace Tokenizer[2] 工具，在训练语料库的一个较小子集上训练字节对编码（BPE） (Sennrich et al., 2016) 分词器。在验证实验中，我

[2]https://github.com/huggingface/tokenizers

们准备了一个词表大小为 8K 的分词器，在训练更大规模的模型时，词表大小将相应扩大。

4.1.2. 基础设施

我们的实验基于 HAI-LLM (High-Flyer, 2023) 进行，这是一个高效且轻量级的训练框架，集成了多种并行策略，包括张量并行 (Korthikanti et al., 2023; Narayanan et al., 2021; Shoeybi et al., 2019)、ZeRO 数据并行 (Rajbhandari et al., 2020)、PipeDream 流水线并行 (Harlap et al., 2018)，以及更具体地，通过结合数据并行和张量并行实现的专家并行 (Lepikhin et al., 2021)。为了优化性能，我们使用 CUDA 和 Triton (Tillet et al., 2019) 开发了 GPU 内核，用于门控算法以及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 集群中的每个节点包含 8 块 GPU，它们通过 NVLink 桥接器两两相连。H800 集群同样采用每节点 8 块 GPU 的配置，节点内通过 NVLink 和 NVSwitch 进行互联。对于 A100 和 H800 集群，均采用 InfiniBand 互连技术以促进节点间的通信。

4.1.3. 超参数

模型设置。 在验证实验中，我们将 Transformer 层数设置为 9，隐藏层维度设置为 1280。我们采用多头注意力机制，共包含 10 个注意力头，每个头的维度为 128。在初始化方面，所有可学习参数均以标准差为 0.006 进行随机初始化。我们将所有前馈神经网络（FFN）替换为混合专家（MoE）层，并确保专家参数的总数为标准 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）保持为标准 FFN 的 2 倍。在此配置下，每个 MoE 模型的总参数量约为 2B，激活参数量约为 0.3B。

训练设置。 我们采用 AdamW 优化器 (Loshchilov and Hutter, 2019)，超参数设置为 $\beta_1 = 0.9$，$\beta_2 = 0.95$，以及 $\text{weight\_decay} = 0.1$。学习率调度采用预热与阶梯衰减策略。初始阶段，学习率在前 2K 步内从 0 线性增加至最大值。随后，在训练步数达到 80% 时，学习率乘以 0.316；在达到 90% 时，再次乘以 0.316。验证实验的最大学习率设置为 1.08×10^{-3}，梯度裁剪范数设置为 1.0。批次大小设置为 2K，最大序列长度为 2K，因此每个训练批次包含 4M 个词元。相应地，总训练步数设置为 25,000 步，以达到 100B 训练词元的目标。由于训练数据充足，我们在训练过程中不使用 dropout。鉴于模型规模相对较小，所有参数（包括专家参数）均部署在单个 GPU 设备上，以避免计算负载不均衡。相应地，我们在训练过程中不丢弃任何词元，也不采用设备级平衡损失。为了防止路由崩溃，我们设置了专家级平衡因子为 0.01。

为便于阅读，我们在附录 A 中还提供了一个关于不同规模 DeepSeekMoE 超参数的概览表。

4.1.4. 评估基准

我们在涵盖多种任务类型的广泛基准测试上进行了评估。基准测试列表如下。

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

语言理解与推理。 对于语言理解与推理任务，我们采用了 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为准确率。

阅读理解。 对于阅读理解任务，我们使用了 RACE-high 和 RACE-middle Lai et al. (2017)，评估指标为准确率。

代码生成。 对于代码生成任务，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，即仅进行一次生成尝试的通过率。

闭卷问答。 对于闭卷问答任务，我们采用了 TriviaQA (Joshi et al., 2017) 和 NaturalQuestions (Kwiatkowski et al., 2019)。评估指标为完全匹配（EM）率。

指标	**提示数**	**Dense**	**Hash Layer**	**Switch**	**GShard**	**DeepSeekMoE**
总参数量	N/A	0.2B	2.0B	2.0B	2.0B	2.0B
激活参数量	N/A	0.2B	0.2B	0.2B	0.3B	0.3B
每 2K Token 的 FLOPs	N/A	2.9T	2.9T	2.9T	4.3T	4.3T
训练 Token 数	N/A	100B	100B	100B	100B	100B
Pile (损失)	N/A	2.060	1.932	1.881	1.867	**1.808**
HellaSwag (准确率)	0-shot	38.8	46.2	49.1	50.5	**54.8**
PIQA (准确率)	0-shot	66.8	68.4	70.5	70.6	**72.3**
ARC-easy (准确率)	0-shot	41.0	45.3	45.9	43.9	**49.4**
ARC-challenge (准确率)	0-shot	26.0	28.2	30.2	31.6	**34.3**
RACE-middle (准确率)	5-shot	38.8	38.8	43.6	42.1	**44.0**
RACE-high (准确率)	5-shot	29.0	30.0	30.9	30.4	**31.7**
HumanEval (Pass@1)	0-shot	0.0	1.2	2.4	3.7	**4.9**
MBPP (Pass@1)	3-shot	0.2	0.6	0.4	0.2	**2.2**
TriviaQA (EM)	5-shot	4.9	6.5	8.9	10.2	**16.6**
NaturalQuestions (EM)	5-shot	1.4	1.4	2.5	3.2	**5.7**

表 1 | 验证实验的评估结果。**粗体**表示最佳结果。与其他 MoE 架构相比，DeepSeekMoE 展现出显著的性能优势。

4.2. 评估

基线模型。 包括 DeepSeekMoE 在内，我们在验证实验中比较了五种模型。**Dense** 表示一个总参数量为 0.2B 的标准稠密 Transformer 语言模型。**Hash Layer** (Roller et al., 2021) 是一种基

于 top-1 哈希路由的 MoE 架构，总参数量为 2.0B，激活参数量为 0.2B，与稠密基线保持一致。**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，其总参数量和激活参数量与 Hash Layer 相同。**GShard** (Lepikhin et al., 2021) 采用 top-2 可学习路由策略，总参数量为 2.0B，激活参数量为 0.3B，因为与 top-1 路由方法相比多激活了一个专家。**DeepSeekMoE** 包含 1 个共享专家和 63 个路由专家，其中每个专家的大小为标准 FFN 的 0.25 倍。包括 DeepSeekMoE 在内，所有对比模型均使用相同的训练语料和训练超参数。所有对比的 MoE 模型具有相同的总参数量，且 GShard 的激活参数量与 DeepSeekMoE 相同。

实验结果。 我们在表 1 中展示了评估结果。对于所有展示的模型，我们报告了在 100B tokens 上训练后的最终评估结果。从表中我们可以得出以下观察结果：（1）凭借稀疏架构和更多的总参数量，Hash Layer 和 Switch Transformer 在激活参数量相同的情况下，性能显著优于稠密基线。（2）与 Hash Layer 和 Switch Transformer 相比，GShard 拥有更多的激活参数量，性能略优于 Switch Transformer。（3）在总参数量和激活参数量相同的情况下，DeepSeekMoE 展现出对 GShard 的压倒性优势。这些结果展示了我们的 DeepSeekMoE 架构在现有 MoE 架构体系中的优越性。

指标	**提示数**	**GShard**×1.5	**Dense**×16	**DeepSeekMoE**
相对专家大小	N/A	1.5	1	0.25
专家数量	N/A	0 + 16	16 + 0	1 + 63
激活专家数量	N/A	0 + 2	16 + 0	1 + 7
专家总参数量	N/A	2.83B	1.89B	1.89B
激活专家参数量	N/A	0.35B	1.89B	0.24B
每 2K Token 的 FLOPs	N/A	5.8T	24.6T	4.3T
训练 Token 数	N/A	100B	100B	100B
Pile (损失)	N/A	1.808	1.806	1.808
HellaSwag (准确率)	0-shot	54.4	55.1	54.8
PIQA (准确率)	0-shot	71.1	71.9	72.3
ARC-easy (准确率)	0-shot	47.3	51.9	49.4
ARC-challenge (准确率)	0-shot	34.1	33.8	34.3
RACE-middle (准确率)	5-shot	46.4	46.3	44.0
RACE-high (准确率)	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、更大规模的 GShard 模型与更大规模的稠密模型之间的对比。在“专家数量”行中，$a + b$ 表示 a 个共享专家和 b 个路由专家。在“激活专家数量”行中，$a + b$ 表示 a 个激活的共享专家和 b 个激活的路由专家。DeepSeekMoE 的性能与专家参数量和计算量为其 1.5 倍的 GShard 模型相当。此外，DeepSeekMoE 的性能几乎接近 FFN 参数量为其 16 倍的稠密模型，后者在模型容量方面为 MoE 模型设定了上限。

4.3. DeepSeekMoE 紧密逼近 MoE 模型的上限

我们已经证明，DeepSeekMoE 的性能优于密集基线模型及其他 MoE 架构。为了更准确地评估 DeepSeekMoE 的性能，我们将其与参数量（总参数量或激活参数量）更大的基线模型进行了比较。这些比较使我们能够估算出 GShard 或密集基线模型达到与 DeepSeekMoE 相当性能所需的模型规模。

与 GShard×1.5 的比较。 表 2 展示了 DeepSeekMoE 与专家规模扩大 1.5 倍的更大 GShard 模型之间的比较，这使得专家参数量和专家计算量均增加了 1.5 倍。总体而言，我们观察到 DeepSeekMoE 取得了与 GShard×1.5 相当的性能，这凸显了 DeepSeekMoE 架构固有的显著优势。除了与 GShard×1.5 的比较外，我们还在附录 B 中展示了与 GShard×1.2 的比较结果。

此外，我们将 DeepSeekMoE 的总参数量增加至 13.3B，并将其与总参数量分别为 15.9B 和 19.8B 的 GShard×1.2 及 GShard×1.5 进行比较。我们发现，在更大规模下，DeepSeekMoE 甚至能显著优于 GShard×1.5。这些结果同样提供在附录 B 中。

与 Dense×16 的比较。 表 2 还展示了 DeepSeekMoE 与更大规模密集模型之间的比较。为了进行公平比较，我们未采用广泛使用的注意力机制与 FFN 参数量之比（1:2）。相反，我们配置了 16 个共享专家，每个专家的参数量与标准 FFN 相同。该架构模拟了一个具有 16 倍标准 FFN 参数量的密集模型。从表中可以看出，DeepSeekMoE 的性能几乎接近 Dense×16，后者在模型容量方面为 MoE 模型设定了严格的上限。这些结果表明，**至少在约 *2B* 参数量和 *100B* 训练 *token* 的规模下，*DeepSeekMoE* 的性能与 *MoE* 模型的理论上限高度吻合**。此外，我们在附录 B 中提供了与 Dense×4 的额外比较结果。

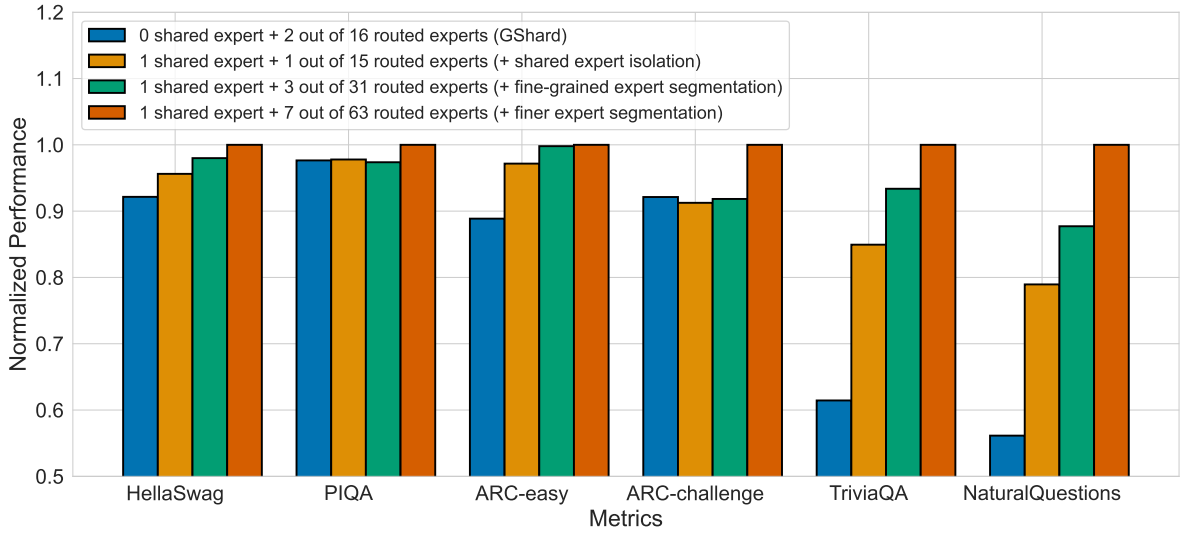

图 3 | DeepSeekMoE 的消融实验。为便于展示，性能已按最佳性能进行归一化处理。所有对比模型的总参数量和激活参数量均相同。我们可以发现，细粒度专家分割与共享专家隔离策略均有助于提升整体性能。

4.4. 消融实验

为了验证细粒度专家分割与共享专家隔离策略的有效性，我们对 DeepSeekMoE 进行了消融实验，并将结果展示于图 3。为确保公平比较，我们保证所有参与对比的模型具有相同的总参数量和激活参数量。

共享专家隔离。 为了评估共享专家隔离策略的影响，我们在 GShard 的基础上隔离出一个专家作为共享专家。从图 3 可以看出，与 GShard 相比，有意隔离一个共享专家在大多数基准测试上均带来了性能提升。这些结果支持了共享专家隔离策略有助于提升模型性能的假设。

细粒度专家分割。 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划分为更细的粒度来进行更详细的比较。具体而言，我们将每个专家分割为 2 个或 4 个更小的专家，从而得到总共 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：专家分割粒度的持续细化对应着整体模型性能的持续提升。这些发现为细粒度专家分割策略的有效性提供了实证支持。

共享专家与路由专家的比例。 此外，我们探讨了共享专家与路由专家的最佳比例。基于包含 64 个专家的最细粒度设置，并在保持总专家数量和激活专家数量不变的前提下，我们尝试分别隔离 1、2 和 4 个专家作为共享专家。我们发现，共享专家与路由专家的不同比例对性能影响并不显著，1、2 和 4 个共享专家分别取得了 1.808、1.806 和 1.811 的 Pile 损失值。考虑到 1:3 的比例能带来略微更优的 Pile 损失值，在扩大 DeepSeekMoE 规模时，我们将共享专家与激活的路由专家的比例保持为 1:3。

4.5. 专家专业化分析

在本节中，我们对 DeepSeekMoE 2B 的专家专业化进行了实证分析。本节中的 DeepSeekMoE 2B 指的是表 1 中报告的模型，即包含 20 亿总参数，其中包含 1 个共享专家，并在 63 个路由专家中激活 7 个。

DeepSeekMoE 在路由专家间表现出更低的冗余度。 为了评估路由专家之间的冗余度，我们禁用了不同比例的最高路由概率专家，并评估 Pile 损失。具体而言，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 top-K 个专家。为保证公平性，我们将 DeepSeekMoE 与 GShard×1.5 进行比较，因为在不禁用任何专家时，两者的 Pile 损失相同。如图 4 所示，与 GShard×1.5 相比，DeepSeekMoE 对最高路由概率专家的禁用更为敏感。这种敏感性表明 DeepSeekMoE 的参数冗余度较低，因为每个路由专家都更具不可替代性。相比之下，GShard×1.5 的专家参数冗余度较高，因此在禁用最高路由概率专家时能够缓解性能下降。

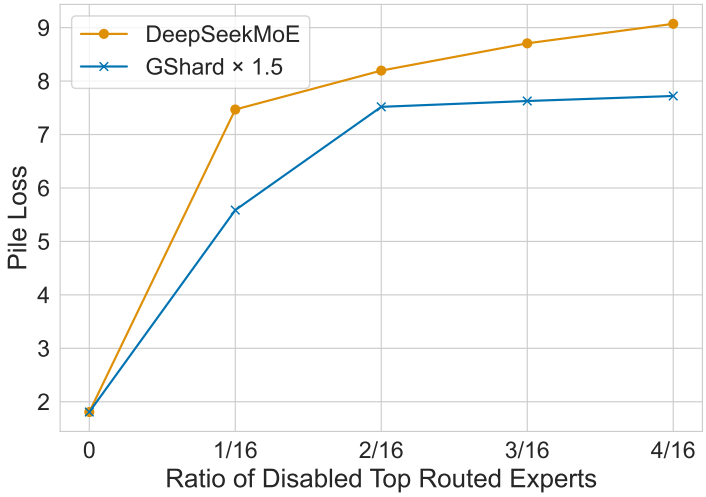

图 4 | 不同比例的最高路由概率专家被禁用时的 Pile 损失。值得注意的是，DeepSeekMoE 对禁用的最高路由概率专家比例表现出更高的敏感性，表明 DeepSeekMoE 中路由专家之间的冗余度较低。

共享专家无法被路由专家替代。 为了探究 DeepSeekMoE 中共享专家的作用，我们将其禁用，并额外激活一个路由专家。Pile 数据集上的评估结果显示，Pile 损失显著上升，从 1.808 增至 2.414，尽管我们保持了相同的计算成本。该结果凸显了共享专家的关键作用，表明共享专家捕获了路由专家未共享的基础且核心的知识，使其无法被路由专家所替代。

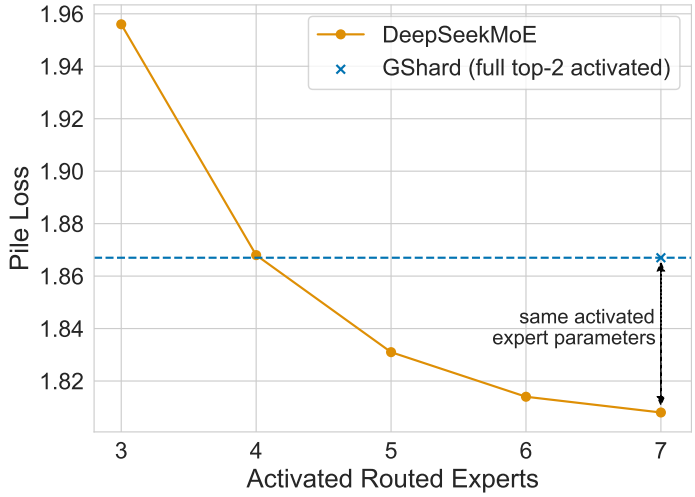

图 5 | DeepSeekMoE 中不同数量激活路由专家对应的 Pile 损失。仅激活 4 个路由专家时，DeepSeekMoE 即可达到与 GShard 相当的 Pile 损失。

DeepSeekMoE 能够更准确地获取知识。 为了验证我们的观点，即组合激活专家时更高的灵活性有助于更准确、更有针对性地获取知识，我们研究了 DeepSeekMoE 是否能够在激活专家数量更少的情况下获取所需知识。具体而言，我们将激活的路由专家数量从 3 个变化到 7 个，并评估相应的 Pile 损失。如图 5 所示，即使仅激活 4 个路由专家，DeepSeekMoE 也能达到与

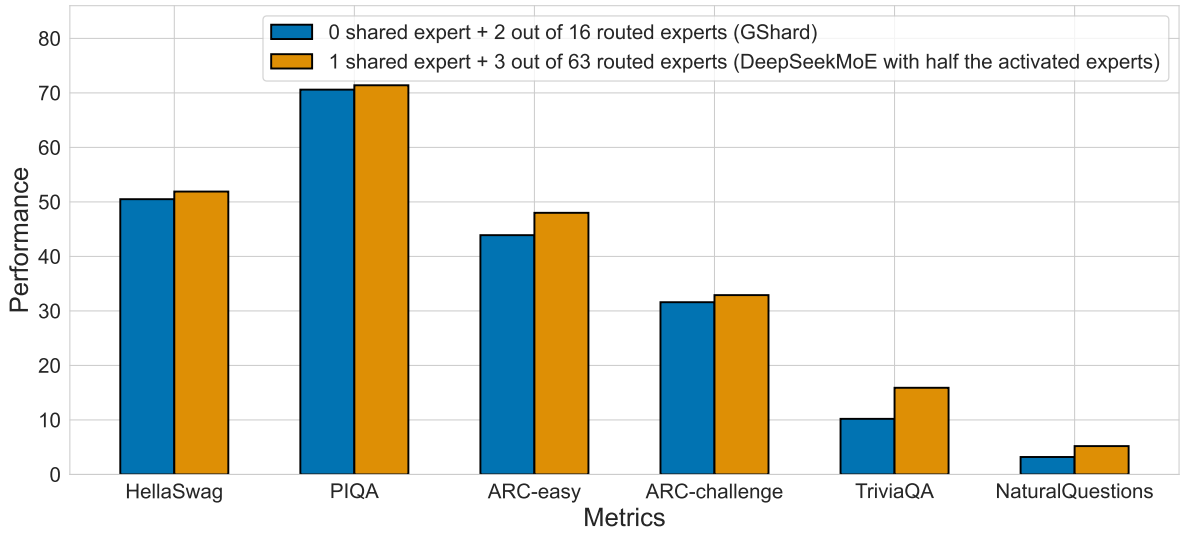

图 6 | 激活专家数量减半时 GShard 与 DeepSeekMoE 的比较（从头训练）。在总专家参数相同且仅激活一半专家参数的情况下，DeepSeekMoE 的性能仍优于 GShard。

GShard 相当的 Pile 损失。这一观察结果支持了 DeepSeekMoE 能够更准确、更高效地获取所需知识的观点。

受这些发现的启发，为了更严谨地验证 DeepSeekMoE 的专家专业化能力和准确的知识获取能力，我们从头训练了一个新模型。该模型包含 1 个共享专家和 63 个路由专家，其中仅激活 3 个路由专家。图 6 所示的评估结果表明，即使在总专家参数相同且仅激活一半专家参数的情况下，DeepSeekMoE 的性能仍优于 GShard。这凸显了 DeepSeekMoE 更高效利用专家参数的能力，即激活专家中有效参数的比例远高于 GShard。

5. 扩展至 DeepSeekMoE 16B

基于 DeepSeekMoE 架构，我们将 MoE 模型扩展至更大规模，总参数量达到 16B，并在 2T 词元上进行训练。实验结果表明，与 LLaMA2 7B 相比，DeepSeekMoE 16B 仅使用约 40% 的计算量便取得了更优越的性能。

5.1. 实验设置

5.1.1. 训练数据与分词

我们按照第 4.1.1 节所述，从相同的语料库中采样训练数据。与验证实验不同，我们采样了更大规模的数据，共计 2T 词元，与 LLaMA2 7B 的训练词元数量保持一致。我们同样使用 HuggingFace Tokenizer 工具训练 BPE 分词器，但将 DeepSeekMoE 16B 的词汇表大小设置为 100K。

5.1.2. 超参数设置

模型设置。 对于 DeepSeekMoE 16B，我们将 Transformer 层数设置为 28，隐藏层维度设置为 2048。我们采用多头注意力机制，共包含 16 个注意力头，每个头的维度为 128。在参数初始化方面，所有可学习参数均以标准差为 0.006 进行随机初始化。我们将除第一层外的所有前馈神经网络（FFN）替换为 MoE 层，因为我们观察到第一层的负载均衡状态收敛速度特别慢。每个 MoE 层包含 2 个共享专家和 64 个路由专家，其中每个专家的规模为标准 FFN 的 0.25 倍。每个词元将被路由至这 2 个共享专家以及 64 个路由专家中的 6 个。由于专家规模过小可能导致计算效率下降，因此未采用更细粒度的专家划分。在超过 16B 的更大规模下，仍可考虑采用更细的粒度。在我们的配置下，DeepSeekMoE 16B 的总参数量约为 16.4B，激活参数量约为 2.8B。

训练设置。 我们采用 AdamW 优化器 (Loshchilov and Hutter, 2019)，超参数设置为 $\beta_1 = 0.9$，$\beta_2 = 0.95$，以及 $\text{weight\_decay} = 0.1$。学习率调度同样采用预热与阶梯衰减（warmup-and-step-decay）策略。初始阶段，学习率在前 2K 步内从 0 线性增加至最大值。随后，在训练步数达到 80% 时，学习率乘以 0.316；在达到 90% 时，再次乘以 0.316。DeepSeekMoE 16B 的最大学习率设置为 4.2×10^{-4}，梯度裁剪范数设置为 1.0。批次大小（batch size）设置为 4.5K，最大序列长度为 4K，因此每个训练批次包含 18M 词元。相应地，总训练步数设置为 106,449 步，以达到 2T 训练词元的目标。由于训练数据充足，我们在训练过程中未使用 dropout。我们利用流水线并行将模型的不同层部署在不同的设备上，且对于每一层，所有专家均部署在同一设备上。因此，我们在训练过程中也不会丢弃任何词元，且未采用设备级平衡损失。为防止路由崩溃，我们将专家级平衡因子设置为较小的 0.001。因为我们发现，在我们的并行策略下，较高的专家级平衡因子无法提升计算效率，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外，我们引入了额外的基准以进行更全面的评估。现将与验证实验所用基准的差异介绍如下。

语言建模。 在语言建模任务中，我们还在 Pile (Gao et al., 2020) 的测试集上对模型进行评估。由于 DeepSeekMoE 16B 使用的分词器与 LLaMA2 7B 不同。为保证公平比较，我们采用每字节比特数（bits per byte, BPB）作为评估指标。

阅读理解。 在阅读理解任务中，我们额外引入了 DROP (Dua et al., 2019) 基准。评估指标为完全匹配（Exactly Matching, EM）率。

数学推理。 在数学推理任务中,我们额外引入了 GSM8K (Cobbe et al., 2021) 和 MATH (Hendrycks et al., 2021) 基准，并以 EM 作为评估指标。

多学科多项选择。 在多项选择题任务中，我们额外在 MMLU (Hendrycks et al., 2020) 上对模型进行评估。评估指标为准确率。

消歧任务。 在消歧任务中，我们额外引入了 WinoGrande (Sakaguchi et al., 2019) 基准，评估指标为准确率。

中文基准。 由于 DeepSeekMoE 16B 在双语语料上进行预训练，我们还在四个中文基准上对其进行了评估。CLUEWSC (Xu et al., 2020) 是一个中文消歧基准。CEval (Huang et al., 2023) 和 CMMLU (Li et al., 2023) 是两个形式与 MMLU 相似的中文多学科多项选择基准。CHID (Zheng et al., 2019) 是一个中文成语补全基准，旨在评估模型对中国文化的理解能力。上述中文基准的评估指标为准确率或 EM。

Open LLM 排行榜。 我们基于内部评估框架对上述所有基准进行了评估。为了公平且便捷地将 DeepSeekMoE 16B 与开源模型进行比较，我们额外在 Open LLM Leaderboard 上对其进行了评估。Open LLM Leaderboard 是由 HuggingFace 支持的公共排行榜，包含六个任务：ARC (Clark et al., 2018)、HellaSwag (Zellers et al., 2019)、MMLU (Hendrycks et al., 2020)、TruthfulQA (Lin et al., 2022)、Winogrande (Sakaguchi et al., 2019) 和 GSM8K (Cobbe et al., 2021)。

5.2. 评估

5.2.1. 与 *DeepSeek 7B* 的内部对比

我们首先对 DeepSeekMoE 16B 与 DeepSeek 7B (DeepSeek-AI, 2024)（一个拥有 6.9B 参数的稠密语言模型）进行了内部对比。为确保公平性，两个模型均在包含 2T 个 token 的相同语料库上进行训练。这使得我们能够准确评估所提 MoE 架构的有效性，且不受训练数据差异的影响。

评估结果如表 3 所示，得出以下观察结果：(1) 总体而言，仅使用约 40% 的计算量，DeepSeekMoE 16B 即达到了与 DeepSeek 7B 相当的性能。(2) DeepSeekMoE 16B 在语言建模以及 Pile、HellaSwag、TriviaQA 和 NaturalQuestions 等知识密集型任务上表现出显著优势。鉴于在 MoE 模型中，FFN 参数量远大于注意力参数，这些结果印证了 Transformer 中的 FFN 具备知识记忆能力的观点 (Dai et al., 2022a)。(3) 与其他任务上的优异表现相比，DeepSeekMoE 在处理多项选择题任务时存在一定局限性。这一不足源于 DeepSeekMoE 16B 中注意力参数相对有限（DeepSeekMoE 16B 仅包含约 0.5B 注意力参数，而 DeepSeek 7B 包含 2.5B 注意力参数）。我们此前对 DeepSeek 7B 的研究表明，注意力容量与多项选择题任务的性能呈正相关。例如，采用多查询注意力机制 (Shazeer, 2019) 的 DeepSeek 7B MQA 在类似 MMLU 的任务上也表现不佳。此外，为了更全面地了解 DeepSeekMoE 16B 的训练过程，我们在附录 C 中提供了 DeepSeekMoE 16B 与 DeepSeek 7B (Dense) 在训练期间的基准测试曲线以供参考。

指标	提示数	DeepSeek 7B (Dense)	DeepSeekMoE 16B
总参数量	N/A	6.9B	16.4B
激活参数量	N/A	6.9B	2.8B
每 4K Token 计算量	N/A	183.5T	74.4T
训练 Token 数	N/A	2T	2T
Pile (BPB)	N/A	0.75	**0.74**
HellaSwag (准确率)	0-shot	75.4	**77.1**
PIQA (准确率)	0-shot	79.2	**80.2**
ARC-easy (准确率)	0-shot	**67.9**	**68.1**
ARC-challenge (准确率)	0-shot	48.1	**49.8**
RACE-middle (准确率)	5-shot	**63.2**	61.9
RACE-high (准确率)	5-shot	**46.5**	**46.4**
DROP (精确匹配)	1-shot	**34.9**	32.9
GSM8K (精确匹配)	8-shot	17.4	**18.8**
MATH (精确匹配)	4-shot	3.3	**4.3**
HumanEval (Pass@1)	0-shot	26.2	**26.8**
MBPP (Pass@1)	3-shot	**39.0**	**39.2**
TriviaQA (精确匹配)	5-shot	59.7	**64.8**
NaturalQuestions (精确匹配)	5-shot	22.2	**25.5**
MMLU (准确率)	5-shot	**48.2**	45.0
WinoGrande (准确率)	0-shot	**70.5**	**70.2**
CLUEWSC (精确匹配)	5-shot	**73.1**	72.1
CEval (准确率)	5-shot	**45.0**	40.6
CMMLU (准确率)	5-shot	**47.2**	42.5
CHID (准确率)	0-shot	**89.3**	**89.4**

表 3 | DeepSeek 7B 与 DeepSeekMoE 16B 的对比。**粗体**表示最优或接近最优。仅使用 40.5% 的计算量，DeepSeekMoE 16B 即达到了与 DeepSeek 7B 相当的性能。

值得注意的是，由于 DeepSeekMoE 16B 参数量适中，它支持在显存为 40GB 的单张 GPU 上进行部署。通过适当的算子优化，其推理速度可达到 7B 稠密模型的近 2.5 倍。

5.2.2. 与开源模型的对比

与 LLaMA2 7B 的内部对比。 在开源模型领域，我们主要将 DeepSeekMoE 16B 与 LLaMA2 7B (Touvron et al., 2023b) 进行对比，后者是一个拥有 67 亿参数且性能强劲的知名开源语言模型。DeepSeekMoE 16B 与 LLaMA2 7B 均在 2T 个 token 上进行了预训练。与 LLaMA2 7B 相比，DeepSeekMoE 的总参数量达到其 245%，但仅需 39.6% 的计算量。我们在内部基准测试上的结果如表 4 所示，由此得出以下观察结论：(1) 在评估的基准测试中，仅使用约 40% 的计算量，DeepSeekMoE 16B 在大多数基准测试上的表现均优于 LLaMA2 7B。(2) DeepSeekMoE 16B 的数学推理和代码生成能力优于 LLaMA2 7B，这归因于我们的预训练语料库中富含数学和代码相关文本。(3) 鉴于我们的预训练语料库中包含中文文本，DeepSeekMoE 16B 在中文基准测试

指标	提示数	LLaMA2 7B	DeepSeekMoE 16B
总参数量	N/A	6.7B	16.4B
激活参数量	N/A	6.7B	2.8B
每 4K Token 计算量	N/A	187.9T	74.4T
训练 Token 数	N/A	2T	2T
Pile (BPB)	N/A	0.76	**0.74**
HellaSwag (准确率)	0-shot	75.6	**77.1**
PIQA (准确率)	0-shot	78.0	**80.2**
ARC-easy (准确率)	0-shot	**69.1**	68.1
ARC-challenge (准确率)	0-shot	49.0	**49.8**
RACE-middle (准确率)	5-shot	60.7	**61.9**
RACE-high (准确率)	5-shot	45.8	**46.4**
DROP (精确匹配)	1-shot	**34.0**	32.9
GSM8K (精确匹配)	8-shot	15.5	**18.8**
MATH (精确匹配)	4-shot	2.6	**4.3**
HumanEval (Pass@1)	0-shot	14.6	**26.8**
MBPP (Pass@1)	3-shot	21.8	**39.2**
TriviaQA (精确匹配)	5-shot	63.8	**64.8**
NaturalQuestions (精确匹配)	5-shot	**25.5**	**25.5**
MMLU (准确率)	5-shot	**45.8**	45.0
WinoGrande (准确率)	0-shot	69.6	**70.2**
CLUEWSC (精确匹配)	5-shot	64.0	**72.1**
CEval (准确率)	5-shot	33.9	**40.6**
CMMLU (准确率)	5-shot	32.6	**42.5**
CHID (准确率)	0-shot	37.9	**89.4**

表 4 | LLaMA2 7B 与 DeepSeekMoE 16B 的对比。仅使用 39.6% 的计算量，DeepSeekMoE 16B 在大多数基准测试上均优于 LLaMA2 7B。

上展现出对 LLaMA2 7B 的显著性能优势。(4) 尽管在英文文本上的训练量较少，DeepSeekMoE 16B 在英文理解或知识密集型基准测试上仍取得了与 LLaMA2 7B 相当或更优的性能，这充分证明了 DeepSeekMoE 16B 的卓越能力。

在 Open LLM Leaderboard 上的评估。 除内部评估外，我们还在 Open LLM Leaderboard 上对 DeepSeekMoE 16B 进行了评估，并将其与其他开源模型进行对比。除 LLaMA2 7B 外，我们还考虑了更广泛的开源模型集合，包括 LLaMA 7B (Touvron et al., 2023a)、Falcon 7B (Almazrouei et al., 2023)、GPT-J 6B (Wang and Komatsuzaki, 2021)、RedPajama-INCITE 7B 和 3B (Together-AI, 2023)、Open LLaMA 7B 和 3B (Geng and Liu, 2023)、OPT 2.7B (Zhang et al., 2022)、Pythia 2.8B (Biderman et al., 2023)、GPT-neo 2.7B (Black et al., 2021) 以及 BLOOM 3B (Scao et al., 2022)。如图 1 所示的评估结果表明，DeepSeekMoE 16B 始终大幅优于激活参数量相近的模型。此外，它与激活参数量约为其 2.5 倍的 LLaMA2 7B 取得了相当的

性能。

6. DeepSeekMoE 16B 的对齐

先前的研究表明，MoE 模型通常无法从微调中获得显著的性能提升 (Artetxe et al., 2022; Fedus et al., 2021)。然而，Shen et al. (2023) 的研究结果表明，MoE 模型确实可以从指令微调中受益。为了评估 DeepSeekMoE 16B 是否能从微调中受益，我们进行了监督微调（SFT），以基于 DeepSeekMoE 16B 构建一个对话模型。实验结果表明，DeepSeekMoE Chat 16B 同样取得了与 LLaMA2 SFT 7B 和 DeepSeek Chat 7B 相当的性能。

6.1. 实验设置

训练数据。 为训练对话模型，我们在内部构建的数据集上进行了监督微调（SFT），该数据集包含 140 万条训练样本。该数据集涵盖数学、代码、写作、问答、推理、摘要等多个广泛类别。我们的 SFT 训练数据主要以英文和中文为主，使得该对话模型具备较强的通用性，并适用于双语场景。

超参数设置。 在监督微调过程中，我们将批次大小（batch size）设置为 1024 个样本，并使用 AdamW 优化器 (Loshchilov and Hutter, 2019) 进行 8 个 epoch 的训练。我们采用 4K 的最大序列长度，并尽可能密集地打包训练样本，直至达到序列长度上限。在监督微调中我们未使用 dropout，仅设置了一个恒定的学习率 10^{-5}，未采用任何学习率调度策略。

评估基准。 对于对话模型的评估，我们采用了与第 5.1.3 节相似的基准测试，并进行了以下调整：(1) 我们排除了 Pile (Gao et al., 2020)，因为对话模型很少用于纯语言建模任务。(2) 由于观察到 CHID (Zheng et al., 2019) 的结果存在不稳定性，难以得出可靠结论，故将其排除。(3) 我们额外加入了 BBH (Suzgun et al., 2022)，以更全面地评估对话模型的推理能力。

6.2. 评估

基线模型。 为了验证 DeepSeekMoE 16B 在对齐后的潜力，我们对 LLaMA2 7B、DeepSeek 7B 和 DeepSeekMoE 16B 进行了监督微调，并使用完全相同的微调数据以确保公平性。相应地，我们构建了三个对话模型，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 与另外两个稠密对话模型（计算量约为其 2.5 倍）进行了比较。

实验结果。 评估结果如表 5 所示。我们的主要观察结果包括：(1) DeepSeekMoE Chat 16B 在仅消耗约 40% 计算量的情况下，在语言理解与推理（PIQA、ARC、BBH）、机器阅读理解（RACE）、

[3]我们使用 LLaMA2 SFT 以区别于官方发布的 LLaMA2 Chat (Touvron et al., 2023b) 模型。

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	**72.2**
PIQA (Acc.)	0-shot	76.9	78.4	**79.7**
ARC-easy (Acc.)	0-shot	69.7	**70.2**	**69.9**
ARC-challenge (Acc.)	0-shot	**50.8**	50.2	50.0
BBH (EM)	3-shot	39.3	**43.1**	42.2
RACE-middle (Acc.)	5-shot	63.9	**66.1**	64.8
RACE-high (Acc.)	5-shot	49.6	**50.8**	**50.6**
DROP (EM)	1-shot	40.0	**41.7**	33.8
GSM8K (EM)	0-shot	**63.4**	62.6	62.2
MATH (EM)	4-shot	13.5	14.7	**15.2**
HumanEval (Pass@1)	0-shot	35.4	45.1	**45.7**
MBPP (Pass@1)	3-shot	27.8	39.0	**46.2**
TriviaQA (EM)	5-shot	60.1	59.5	**63.3**
NaturalQuestions (EM)	0-shot	**35.2**	32.7	**35.1**
MMLU (Acc.)	0-shot	**50.0**	49.7	47.2
WinoGrande (Acc.)	0-shot	65.1	68.4	**69.0**
CLUEWSC (EM)	5-shot	48.4	66.2	**68.2**
CEval (Acc.)	0-shot	35.1	**44.7**	40.0
CMMLU (Acc.)	0-shot	36.9	**51.2**	49.3

表 5 | LLaMA2 SFT 7B、DeepSeek Chat 7B 与 DeepSeekMoE Chat 16B 的对比，这三个模型均在相同的 SFT 数据上进行了微调。与两款 7B 稠密模型相比，DeepSeekMoE Chat 16B 仅使用 40% 的计算量，仍在大多数基准测试上取得了相当或更优的性能。

数学（GSM8K、MATH）以及知识密集型任务（TriviaQA、NaturalQuestions）上均达到了与 7B 稠密模型相当的性能。(2) 在代码生成任务上，DeepSeekMoE Chat 16B 显著优于 LLaMA2 SFT 7B，在 HumanEval 和 MBPP 上表现出显著提升。此外，它也超越了 DeepSeek Chat 7B。(3) 在包括 MMLU、CEval 和 CMMLU 在内的多项选择题问答基准测试中，DeepSeekMoE Chat 16B 仍落后于 DeepSeek Chat 7B，这与基座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 与 DeepSeek 7B 之间的性能差距有所缩小。(4) 得益于双语语料的预训练，DeepSeekMoE Chat 16B 在所有中文基准测试上均显著优于 LLaMA2 SFT 7B。这些结果表明 DeepSeekMoE 16B 在中英文能力上保持均衡，增强了其在多样化场景中的通用性和适用性。综上所述，对话模型的评估凸显了 DeepSeekMoE 16B 从对齐中获益的潜力，并验证了其在仅使用约 40% 计算量的情况下实现与稠密模型相当性能的持续优势。

7. DeepSeekMoE 145B 进行中

受 DeepSeekMoE 16B 卓越性能的鼓舞，我们进一步开展了将 DeepSeekMoE 扩展至 145B 的初步探索。在这项初步研究中，DeepSeekMoE 145B 在 245B tokens 上进行训练，已展现出相对于 GShard 架构的持续优势，并有望达到或超越 DeepSeek 67B（稠密模型）的性能。此外，在 DeepSeekMoE 145B 的最终版本完成并充分训练后，我们也计划将其公开。

7.1. 实验设置

训练数据与分词。 对于 DeepSeekMoE 145B，我们采用了与 DeepSeekMoE 16B 完全相同的训练语料和分词器，唯一的区别在于 DeepSeekMoE 145B 仅在 245B tokens 上进行训练以作初步研究。

模型设置。 对于 DeepSeekMoE 145B，我们将 Transformer 层数设置为 62，隐藏层维度设置为 4096。我们采用多头注意力机制，共包含 32 个注意力头，每个头的维度为 128。在参数初始化方面，所有可学习参数均以标准差为 0.006 进行随机初始化。与 DeepSeekMoE 16B 类似，我们将除第一层外的所有前馈神经网络（FFN）替换为 MoE 层。每个 MoE 层包含 4 个共享专家和 128 个路由专家，其中每个专家的规模为标准 FFN 的 0.125 倍。每个 token 将被路由至这 4 个共享专家以及 128 个路由专家中的 12 个。在此配置下，DeepSeekMoE 145 的总参数量约为 144.6B，激活参数量约为 22.2B。

训练设置。 我们采用 AdamW 优化器 (Loshchilov and Hutter, 2019)，超参数设置为 $\beta_1 = 0.9$、$\beta_2 = 0.95$ 以及 $\text{weight\_decay} = 0.1$。针对 DeepSeekMoE 145B 的初步研究，我们采用预热后恒定的学习率调度策略。初始阶段，学习率在前 2K 步内从 0 线性增加至最大值。随后，在剩余的训练过程中学习率保持恒定。DeepSeekMoE 145B 的最大学习率设置为 3.0×10^{-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，因此每个训练批次包含 18M tokens。我们训练 DeepSeekMoE 145B 共 13,000 步，累计训练 tokens 达到 245B。此外，训练过程中未使用 Dropout。我们利用流水线并行将模型的不同层部署在不同的设备上；对于每一层，所有路由专家将均匀部署在 4 个设备上（即专家并行与数据并行相结合）。由于我们在 DeepSeekMoE 145B 中采用了专家并行，因此需要考虑设备级的负载均衡以缓解计算瓶颈。为此，我们将设备级平衡因子设置为 0.05，以促进设备间的计算均衡。同时，我们仍将专家级平衡因子设置为较小的 0.003，以防止路由崩溃。

评估基准。 我们在与 DeepSeekMoE 16B 完全相同的内部基准上对 DeepSeekMoE 145B 进行评估 (参见第 5.1.3 节)。

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	**1.876**	1.888
HellaSwag (Acc.)	0-shot	74.8	72.0	**75.8**	74.9
PIQA (Acc.)	0-shot	79.8	77.6	**80.7**	80.2
ARC-easy (Acc.)	0-shot	69.0	64.0	**69.7**	67.9
ARC-challenge (Acc.)	0-shot	**50.4**	45.8	48.8	49.0
RACE-middle (Acc.)	5-shot	**63.2**	59.2	62.1	59.5
RACE-high (Acc.)	5-shot	**46.9**	43.5	45.5	42.6
DROP (EM)	1-shot	**27.5**	21.6	**27.8**	28.9
GSM8K (EM)	8-shot	**11.8**	6.4	**12.2**	13.8
MATH (EM)	4-shot	2.1	1.6	**3.1**	2.8
HumanEval (Pass@1)	0-shot	**23.8**	17.7	19.5	23.2
MBPP (Pass@1)	3-shot	**33.6**	27.6	**33.2**	32.0
TriviaQA (EM)	5-shot	57.2	52.5	**61.1**	59.8
NaturalQuestions (EM)	5-shot	22.6	19.0	**25.0**	23.5
MMLU (Acc.)	5-shot	**45.1**	26.3	39.4	37.5
WinoGrande (Acc.)	0-shot	70.7	67.6	**71.9**	70.8
CLUEWSC (EM)	5-shot	69.1	65.7	**71.9**	72.6
CEval (Acc.)	5-shot	**40.3**	26.2	37.1	32.8
CMMLU (Acc.)	5-shot	**40.6**	25.4	35.9	31.9
CHID (Acc.)	0-shot	88.5	86.9	**90.3**	88.3

表 6 | DeepSeek 67B（稠密模型）与总参数量约 140B 的 MoE 模型对比。在 "# Experts" 和 "# Activated Experts" 行中，$a + b$ 分别表示 a 个共享专家和 b 个路由专家。**粗体**表示除最后一列外最优或接近最优的性能。DeepSeekMoE 145B，甚至激活专家参数仅为一半的 DeepSeekMoE 142B（Half Activated），均以较大优势超越了 GShard 137B。此外，DeepSeekMoE 145B 仅使用 28.5% 的计算量，便达到了与 DeepSeek 67B 相当的性能。

7.2. 评估结果

基线模型。 除了 **DeepSeekMoE 145B** 之外，我们还考虑了另外三个模型作为对比。**DeepSeek 67B (Dense)** 是一个包含 67.4B 总参数量的稠密模型（模型与训练细节请参阅 DeepSeek-AI (2024)）。**GShard 137B** 与 DeepSeekMoE 145B 具有相同的隐藏层维度和层数，但采用 GShard 架构。需要注意的是，出于计算效率的考虑，DeepSeekMoE 145B 将每个专家中的中间隐藏维度对齐为 64 的倍数，因此其模型规模比 GShard 137B 大 6%。**DeepSeekMoE 142B (Half**

Activated) 的架构与 DeepSeekMoE 145B 相似，但仅包含 2 个共享专家，且 128 个路由专家中仅有 6 个被激活。值得注意的是，所有对比模型（包括 DeepSeekMoE 145B）均使用相同的训练语料库。此外，对比中的所有 MoE 模型均从头开始训练，并共享相同的训练超参数。

实验结果。 根据表 6 中展示的评估结果，我们得出以下观察结论：(1) 尽管总参数量和计算量相当，DeepSeekMoE 145B 的性能仍显著优于 GShard 137B，再次凸显了 DeepSeekMoE 架构的优势。(2) 总体而言，仅使用 28.5% 的计算量，DeepSeekMoE 145B 便达到了与 DeepSeek 67B (Dense) 相当的性能。与 DeepSeekMoE 16B 的发现一致，DeepSeekMoE 145B 在语言建模和知识密集型任务中表现出显著优势，但在多项选择题任务中存在一定局限。(3) 在更大规模下，DeepSeekMoE 142B (Half Activated) 的性能并未明显落后于 DeepSeekMoE 145B。此外，尽管激活的专家参数仅有一半，DeepSeekMoE 142B (Half Activated) 仍能以仅 18.2% 的计算量匹配 DeepSeek 67B (Dense) 的性能。其性能也优于 GShard 137B，这与第 4.5 节的结论相一致。

8. 相关工作

混合专家（Mixture of Experts, MoE）技术最初由 Jacobs et al. (1991); Jordan and Jacobs (1994) 提出，旨在通过独立的专家模块处理不同的样本。Shazeer et al. (2017) 将 MoE 引入语言模型训练，并构建了基于 LSTM (Hochreiter and Schmidhuber, 1997) 的大规模 MoE 模型。随着 Transformer 成为自然语言处理（NLP）中最主流的架构，许多研究尝试将 Transformer 中的前馈神经网络（FFN）扩展为 MoE 层，以构建 MoE 语言模型。GShard (Lepikhin et al., 2021) 和 Switch Transformer (Fedus et al., 2021) 是早期的开创性工作，它们采用可学习的 Top-2 或 Top-1 路由策略，将 MoE 语言模型扩展至极大规模。Hash Layer (Roller et al., 2021) 和 StableMoE (Dai et al., 2022b) 则采用固定路由策略，以实现更稳定的路由与训练过程。Zhou et al. (2022) 提出了一种专家选择（expert-choice）路由策略，允许每个 token 被分配给不同数量的专家。Zoph (2022) 聚焦于 MoE 模型中训练不稳定和微调困难的问题，并提出了 ST-MoE 以克服这些挑战。除了对 MoE 架构和训练策略的研究外，近年来也涌现出大量基于现有 MoE 架构的大规模语言或多模态模型 (Du et al., 2022; Lin et al., 2021; Ren et al., 2023; Xue et al., 2023)。总体而言，以往的 MoE 模型大多基于传统的 Top-1 或 Top-2 路由策略，在提升专家专业化程度方面仍有较大改进空间。为此，我们的 DeepSeekMoE 架构旨在最大限度地提升专家的专业化程度。

9. 结论

本文介绍了面向 MoE 语言模型的 DeepSeekMoE 架构，旨在实现极致的专家专业化。通过细粒度的专家分割与共享专家隔离，DeepSeekMoE 相较于主流 MoE 架构实现了显著提升的专家专业化程度与模型性能。我们从 2B 参数的较小规模起步，验证了 DeepSeekMoE 的优势，证明其具备逼近 MoE 模型性能上限的能力。此外，我们提供了实证数据，表明 DeepSeekMoE 的专家专业化水平高于 GShard。

在扩展至 16B 总参数规模后，我们在 2T tokens 上训练了 DeepSeekMoE 16B，并展示了其卓越的性能：在仅消耗约 40% 计算量的情况下，其表现可与 DeepSeek 7B 和 LLaMA2 7B 相媲美。此外，我们进行了监督微调以实现对齐，构建了基于 DeepSeekMoE 16B 的 MoE 对话模型，进一步展现了其适应性与多功能性。进一步地，我们对将 DeepSeekMoE 扩展至 145B 参数规模进行了初步探索。我们发现，DeepSeekMoE 145B 相较于 GShard 架构仍保持显著优势，且仅使用 28.5%（甚至可能低至 18.2%）的计算量，即可展现出与 DeepSeek 67B 相当的性能。

出于研究目的，我们向公众开源了 DeepSeekMoE 16B 的模型检查点，该模型可在单张 40GB 显存的 GPU 上部署。我们期望本工作能为学术界与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贡献力量。

参考文献

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, and Y. Choi. PIQA: reasoning about physical commonsense in natural language. In The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference,

IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020, pages 7432–7439. AAAI Press, 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. Hao, Z. Sui, B. Chang, and F. Wei. Knowledge neurons in pretrained transformers. 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 8493–8502. Association for Computational 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. Zheng, Z. Sui, B. Chang, and F. Wei. Stablemoe: Stable routing strategy for mixture of experts. 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 7085–7095. Association for Computational Linguistics, 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. URL https://proceedings.mlr.press/v162/du22c.html.

D. Dua, Y. Wang, P. Dasigi, G. Stanovsky, S. Singh, and M. Gardner. DROP: A reading comprehension benchmark requiring discrete reasoning over paragraphs. In J. Burstein, C. Doran, and T. Solorio, editors, Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers), pages 2368–2378. Association for Computational Linguistics, 2019. doi: 10.18653/V1/N19-1246. URL 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.

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, and E. H. Hovy. RACE: large-scale reading comprehension dataset from examinations. In M. Palmer, R. Hwa, and S. Riedel, editors, Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, EMNLP 2017, Copenhagen, Denmark, September 9-11, 2017, pages 785–794. Association for Computational 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. URL 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, and Y. He. Zero: memory optimizations toward training trillion parameter models. In C. Cuicchi, I. Qualters, and W. T. Kramer, editors, Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis, SC 2020, Virtual Event / Atlanta, Georgia, USA, November 9-19, 2020, page 20. 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-Σ: 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, and D. Cox. Triton: An intermediate language and compiler for tiled neural network computations. In Proceedings of the 3rd ACM SIGPLAN International Workshop on Machine Learning and Programming Languages, MAPL 2019, page 10–19, New York, NY, USA, 2019. Association for Computing Machinery. ISBN 9781450367196. doi: 10.1145/3315508.3329973. URL https://doi.org/10.1145/3315508.3329973.

Together-AI. Redpajama-data: An open source recipe to reproduce llama training dataset, April 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. Bikel, 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/3f5ee243547dee91fbd053c1c4a845aa-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, and Y. Choi. HellaSwag: Can a machine really finish your sentence? In A. Korhonen, D. R. Traum, and L. Màrquez, editors, Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers, pages 4791–4800. Association for Computational 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. Zheng, M. Huang, and A. Sun. Chid: A large-scale chinese idiom dataset for cloze test. In A. Korhonen, D. R. Traum, and L. Màrquez, editors, Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers, pages 778–787. Association for Computational Linguistics, 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.

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.

附录

A. 超参数概览

表 7 展示了不同规模下 DeepSeekMoE 的超参数概览。

# 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 | 不同规模下 DeepSeekMoE 的超参数概览。相对专家规模是相对于标准 FFN 而言的。

B. 与更大规模模型的对比

表 8 展示了 DeepSeekMoE 与 GShard×1.2 及 GShard×1.5 的对比结果。表 9 展示了 DeepSeekMoE 与 Dense×4 及 Dense×16 的对比结果。

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	**1.808**	**1.808**
HellaSwag (Acc.)	0-shot	53.7	54.4	**54.8**
PIQA (Acc.)	0-shot	71.8	71.1	**72.3**
ARC-easy (Acc.)	0-shot	46.8	47.3	**49.4**
ARC-challenge (Acc.)	0-shot	31.7	**34.1**	**34.3**
RACE-middle (Acc.)	5-shot	43.7	**46.4**	44.0
RACE-high (Acc.)	5-shot	31.9	**32.4**	31.7
HumanEval (Pass@1)	0-shot	3.7	3.0	**4.9**
MBPP (Pass@1)	3-shot	2.4	**2.6**	2.2
TriviaQA (EM)	5-shot	15.2	15.7	**16.6**
NaturalQuestions (EM)	5-shot	4.5	4.7	**5.7**

表 8 | DeepSeekMoE 与更大规模 GShard 模型的对比。

在总参数量为 13B 的更大规模下，我们将 DeepSeekMoE 与 GShard×1.2 和 GShard×1.5 进行了对比，结果如表 10 所示。在更大规模下，DeepSeekMoE 甚至显著优于 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	**1.806**	**1.808**
HellaSwag (Acc.)	0-shot	47.6	**55.1**	**54.8**
PIQA (Acc.)	0-shot	70.0	71.9	**72.3**
ARC-easy (Acc.)	0-shot	43.9	**51.9**	49.4
ARC-challenge (Acc.)	0-shot	30.5	33.8	**34.3**
RACE-middle (Acc.)	5-shot	42.4	**46.3**	44.0
RACE-high (Acc.)	5-shot	30.7	**33.0**	31.7
HumanEval (Pass@1)	0-shot	1.8	4.3	**4.9**
MBPP (Pass@1)	3-shot	0.2	**2.2**	**2.2**
TriviaQA (EM)	5-shot	9.9	**16.5**	**16.6**
NaturalQuestions (EM)	5-shot	3.0	**6.3**	5.7

表 9 | DeepSeekMoE 与更大规模稠密基线模型的对比。

指标	提示数	GShard×1.2	GShard×1.5	DeepSeekMoE
相对专家规模	N/A	1.2	1.5	0.25
专家数量	N/A	0 + 16	0 + 16	1 + 63
激活专家数量	N/A	0 + 2	0 + 2	1 + 7
总专家参数量	N/A	15.9B	19.8B	13.3B
激活专家参数量	N/A	2.37B	2.82B	2.05B
训练 Token 数	N/A	100B	100B	100B
HellaSwag (准确率)	0-shot	66.6	67.7	**69.1**
PIQA (准确率)	0-shot	75.6	**76.0**	**75.7**
ARC-easy (准确率)	0-shot	56.8	56.8	**58.8**
ARC-challenge (准确率)	0-shot	**39.9**	37.6	38.5
RACE-middle (准确率)	5-shot	51.6	50.6	**52.4**
RACE-high (准确率)	5-shot	37.4	36.3	**38.5**
HumanEval (Pass@1)	0-shot	6.1	6.1	**9.8**
MBPP (Pass@1)	3-shot	7.0	**11.6**	10.6
TriviaQA (精确匹配率)	5-shot	36.5	36.7	**38.2**
NaturalQuestions (精确匹配率)	5-shot	12.6	12.1	**13.7**

表 10 | 在更大规模下 DeepSeekMoE 与更大 GShard 模型的对比。

C. DeepSeekMoE 16B 的训练基准曲线

我们在图 7 中展示了 DeepSeekMoE 16B 和 DeepSeek 7B (Dense) 训练过程中的基准曲线，以供参考。

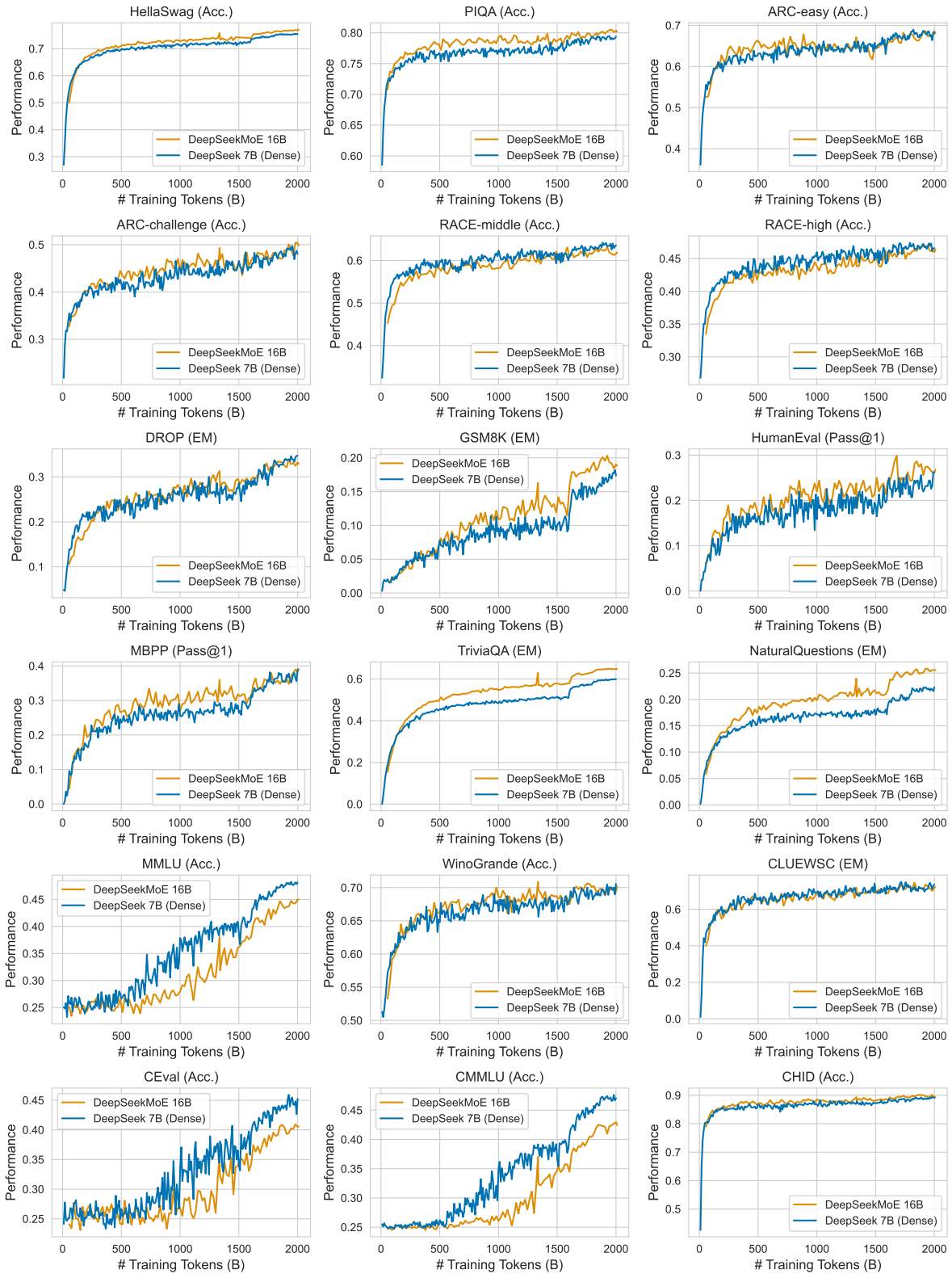

图 7 | DeepSeekMoE 16B 与 DeepSeek 7B (Dense) 训练过程中的基准曲线。