DeepSeek-V3 技术报告

DeepSeek-AI
research@deepseek.com

摘要

我们提出了 DeepSeek-V3，一个强大的专家混合（MoE，Mixture of Experts）语言模型，总参数为 671B，每个令牌激活 37B。为了实现高效的推理和经济有效的训练，DeepSeek-V3 采用了多头潜注意力（Multi-Head Latent Attention, MLA）和 DeepSeekMoE 架构，这些架构在 DeepSeek-V2 中得到了彻底的验证。此外，DeepSeek-V3 提出了一种负载平衡的无辅助损失策略，并设置了一个多令牌预测训练目标。我们在 14.8 万亿个多样化和高质量的代币上对 DeepSeek-V3 进行了预训练，然后进行了有监督的微调和强化学习阶段，以充分利用其能力。综合评估显示，DeepSeek-V3 优于其他开源模型，并实现了可与领先的闭源模型相媲美的性能。尽管其性能出色，但 DeepSeek-V3 只需要 2.788M H800 GPU 小时就可以进行完整的训练。此外，其训练过程非常稳定。在整个训练过程中，我们没有经历任何不可恢复的损失峰值或执行任何回滚。模型检查点可在中获得 https://github.com/deepseek-ai/DeepSeek-V3。

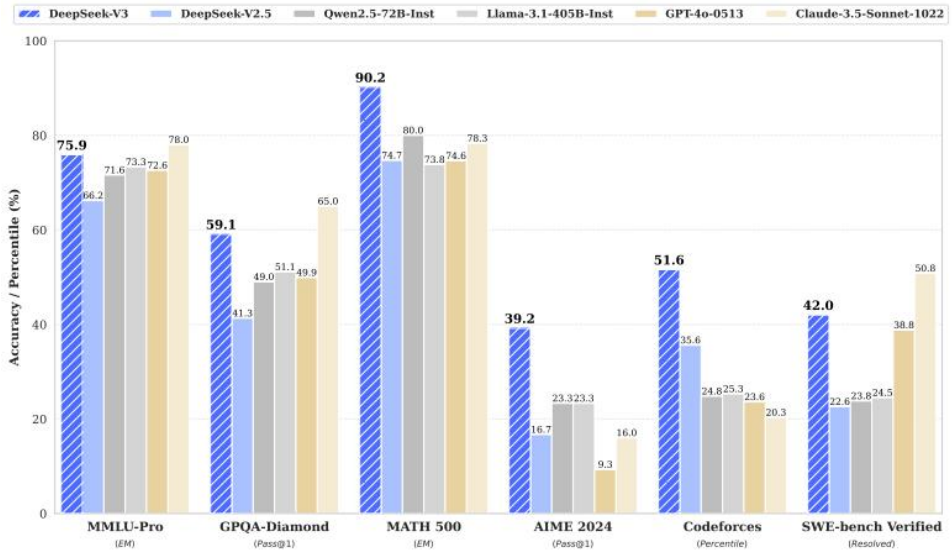


图 1 DeepSeek-V3 及其同行的基准测试性能

目录

1.介绍

近年来，大型语言模型（LLM）经历了快速的迭代和进化(Anthropic, 2024; Google, 2024; OpenAI, 2024a)，逐渐缩小了与人工通用智能（AGI）的差距。除了闭源模型之外，开源模型，包括 DeepSeek-AI 系列 (DeepSeek-AI, 2024a,b,c; Guo et al., 2024)、LLaMA 系列（AI@Meta、2024a、b024；Touvron 等人，2023a、b）、Qwen 系列（Qwen，2023、2024a、b）和 Mistral 系列(Jiang et al., 2023; Mistral, 2024)也在取得重大进展，努力缩小与闭源模型的差距。为了进一步推动开源模型功能的边界，我们扩大了我们的模型，并引入了 DeepSeek-V3，这是一个大型专家混合（MoE）模型，具有 671B 参数，每个令牌有 37B 被激活。

从前瞻性的角度来看，我们始终都在努力追求强大的模型性能和经济成本。因此，在架构方面，DeepSeek-V3 仍然采用多头潜注意力（MLA）（DeepSeek-AI，2024c）进行高效推理，并采用 DeepSeekMoE（Dai et al.，2024）进行具有成本效益的训练。这两种架构已经在 DeepSeekV2（DeepSeek-AI，2024c）中得到了验证，证明了它们在实现高效训练和推理的同时保持鲁棒的模型性能的能力。除了基本架构之外，我们还实现了两个策略来进一步增强模型功能。首先，DeepSeek-V3 提出了一种实现负载平衡的无辅助损失策略（Wang et al.，2024a），目的是最小化因鼓励负载平衡对模型性能的不利影响。其次，DeepSeek-V3 采用了多令牌预测训练目标，我们观察到该目标可以提高评估基准上的整体性能。

为了实现高效的训练，我们支持 FP8 混合精度训练，并对训练框架进行了全面的优化。低精度训练已成为高效训练的一个有前途的解决方案 (Dettmers et al., 2022; Kalamkar et al.,2019; Narang et al., 2017; Peng et al., 2023b)，其演变与硬件能力的进步密切相关(Luo et al., 2024; Micikevicius et al., 2022; Rouhani et al., 2023a)。在这项工作中，我们引入了一个 FP8 混合精度训练框架，并首次在一个非常大规模的模型上验证了其有效性。通过对 FP8 计算和存储的支持，我们实现了加速训练和减少 GPU 内存的使用。在训练框架上，我们设计了一种能够实现高效管道并行性的 DualPipe 算法（双流水线设计和计算-通信重叠），该算法具有较少的管道气泡，并通过计算-通信重叠来隐藏了训练过程中的大部分通信。这种重叠确保了随着模型的进一步扩展，只要我们保持恒定的计算-通信比，我们仍然可以跨节点使用细粒度的专家，同时实现接近零的 all-to-all(全对全)通信开销。此外，我们还开发了高效的跨节点 all-to-all(全对全)通信内核，以充分利用无限带宽（IB）和 NVLink 带宽。此外，我

们精心地优化了内存占用，使得在不使用昂贵的情况下训练 DeepSeek-V3 的张量并行性成为可能。结合这些努力，我们实现了较高的训练效率。

在训练前，我们在 14.8T 高质量和多样化的令牌上训练 DeepSeek-V3。训练前的过程非常稳定。在整个训练过程中，我们没有遇到任何不可恢复的损失峰值或必须回滚。接下来，我们对 DeepSeek-V3 进行了两阶段的上下文长度扩展。在第一阶段，最大上下文长度扩展到 32K，在第二阶段，它进一步扩展到 128K。在此之后，我们在 DeepSeek-V3 的基础模型上进行了后训练，包括监督微调（SFT）和强化学习（RL），以使其与人类的偏好保持一致，并进一步释放其潜力。在训练后阶段，我们从 DeepSeekR1 系列模型中提取出推理能力，同时仔细保持模型精度之间的平衡和生成长度。

我们将根据一系列全面的基准测试来评估 DeepSeek-V3。尽管其经济的训练成本，综合评估显示，DeepSeek-V3-Base 已经成为目前可用的最强的开源基础模型，特别是在代码和数学方面。它的聊天版本也优于其他开源模型，并在一系列的标准和开放式的基准测试上，实现了可与领先的闭源模型相媲美的性能，包括 GPT-4o 和 Claude-3.5-Sonnet。

Training Costs	Pre-Training	Context Extension	Post-Training	Total
in H800 GPU Hours	2664K	119K	5K	2788K
in USD	$5.328M	$0.238M	$0.01M	$5.576M

表 1 DeepSeek-V3 的训练成本，假设 H800 的租金为每 GPU 每小时 2 美元。

最后，我们再次强调了 DeepSeek-V3 的经济训练成本，如表 1 所示，通过我们优化的算法、框架和硬件的协同设计来实现。在训练前阶段，对每万亿代币进行训练 DeepSeek-V3 只需要 180K H800 GPU 小时，即在我们拥有 2048 H800 GPU 的集群上训练 3.7 天。因此，我们的预训练阶段在不到两个月的时间内完成，花费 2664K GPU 小时。结合 119K 的 GPU 小时和 5K 的训练后的 GPU 小时，DeepSeek-V3 的完整训练只需要 2.788M 的 GPU 小时。假设 H800 GPU 的租金为每 GPU 每小时 2 美元，我们的训练总成本仅为 55.76 万美元。请注意，上述成本仅包括 DeepSeek-V3 的官方训练，不包括与先前关于架构、算法或数据的研究和消融实验相关的成本。

我们的主要贡献包括：

架构：创新的负载平衡策略和训练目标

在 DeepSeek-V2 的高效架构之上，我们开创了一种负载平衡的无辅助损失策略，它最大限度地减少了鼓励负载平衡导致的性能下降。

我们研究了一个多令牌预测（MTP）目标，并证明了它对模型性能的好处。它也可以用于推理加速的推测解码。

预训练：达到最终的训练效率

我们设计了一个 FP8 混合精度训练框架，并首次在一个超大规模的模型上验证了 FP8 训练的可行性和有效性。

通过算法、框架和硬件的共同设计，克服了跨节点模块化训练中的通信瓶颈，实现了接近完全的计算通信重叠。这大大提高了我们的训练效率，并降低了训练成本，使我们能够在没有额外开销的情况下进一步扩大模型的规模。

以仅 2.664M H800 GPU 小时的经济成本，我们在 14.8T 代币上完成了 DeepSeek-V3 的预训练，产生了目前最强大的开源基础模型。预训练后的后续训练阶段只需要 0.1M 的 GPU 小时。

后训练：来自 DeepSeek-R1 的知识提炼

我们引入了一种创新的方法，从长思维链（CoT）模型中提取推理能力，特别是从

DeepSeek-R1 系列模型中提取到标准的 LLM，特别是 DeepSeek-V3。我们的管道优雅地结合了 DeepSeek-R1 对 DeepSeek-V3 的验证和反射模式，显著提高了其推理性能。同时，我们还保持了对 DeepSeek-V3 的输出风格和长度的控制。

核心评价结果总结

知识： (1)在 MMLU（大规模多任务语言理解）、MMLU-Pro（大规模多任务语言理解增强）和 GPQA（理工科博士生测试）等教育基准上，DeepSeek-V3 的性能优于所有其他开源机型，在 MMLU 上达到 88.5，在 MMLU-Pro 上达到 75.9，在 GPQA 上达到 59.1。它的性能可与 GPT-4o 和 Claude-Sonnet-3.5,等领先的闭源模型相媲美，缩小了该领域的开源模型和闭源模型之间的差距。(2)对于事实性基准测试，DeepSeek-V3 在 SimpleQA 和中文简体 QA 上的开源模型中都展示了优越的性能。虽然它在英语事实知识（SimpleQA）方面落后于 GPT-4o 和 Claude-Sonnet-3.5,，但它在中国事实知识（中国简体 QA）方面超过了这些模型，突出了它在中国事实知识方面的优势。

代码、数学和推理： (1) DeepSeek-V3 在所有非长思维链（CoT）开源和闭源模型的数学相关基准测试上取得了最先进的性能。值得注意的是，它甚至在特定的基准测试上优于 o1-preview，比如 MATH-500（美国数学竞赛），展示了其健壮的数学推理能力。(2)在编码相关任务上，DeepSeek-V3 成为编码竞争基准的最佳模型，如 LiveCodeBench（世界级编程竞赛），巩固了其在该领域的领先模型的地位。对于与工程相关的任务，虽然 DeepSeek-V3 的性能略低于 Claude-Sonnet-3.5，但它仍然显著地超过了所有其他模型，显示了它在不同技术基准上的竞争力。

在本文的其余部分中，我们首先详细介绍了我们的 DeepSeek-V3 模型体系结构（第 2 节）。随后，我们介绍了我们的基础设施，包括我们的计算集群、训练框架、对 FP8 训练的支持、推理部署策略，以及我们对未来硬件设计的建议。接下来，我们将描述我们的训练前过程，包括训练数据的构建、超参数设置、长上下文扩展技术、相关的评估，以及一些讨论（第 4 节）。之后，我们将讨论我们在训练后方面的努力，包括监督微调（SFT）、强化学习（RL）、相应的评估和讨论（第 5 节）。最后，我们总结了这项工作，讨论了 DeepSeek-V3 现有的局限性，并提出了未来研究的潜在方向（第 6 节）。

2.架构

我们首先介绍了 DeepSeek-V3 的基本架构，其特点是采用多头潜注意力（MLA）（DeepSeek-AI，2024c）进行高效推理和 DeepSeekMoE（Dai et al.，2024）进行经济训练。然后，我们提出了一个多令牌预测（MTP）训练目标，我们观察到它提高在评估基准上的总体性能。对于其他没有明确提到的次要细节，DeepSeek-V3 遵循 DeepSeekV2 的设置（DeepSeek-AI，2024c）。

2.1.基本架构

DeepSeek-V3 的基本架构仍然在 Transformer（Vaswani et al.，2017）框架内。为了高效的推理和经济的训练，DeepSeek-V3 也采用了 MLA 和 DeepSeekMoE，这已经被 DeepSeek-V2 彻底验证了。与 DeepSeek-V2 相比，一个例外是我们另外引入了一个无损耗的负载平衡针对 DeepSeekMoE 的策略（Wang et al.，2024a），以减轻由于努力确保负载平衡而导致的性能下降。图 2 说明了 DeepSeek-V3 的基本架构，我们将在本节中简要回顾 MLA 和 DeepSeekMoE 的详细信息。

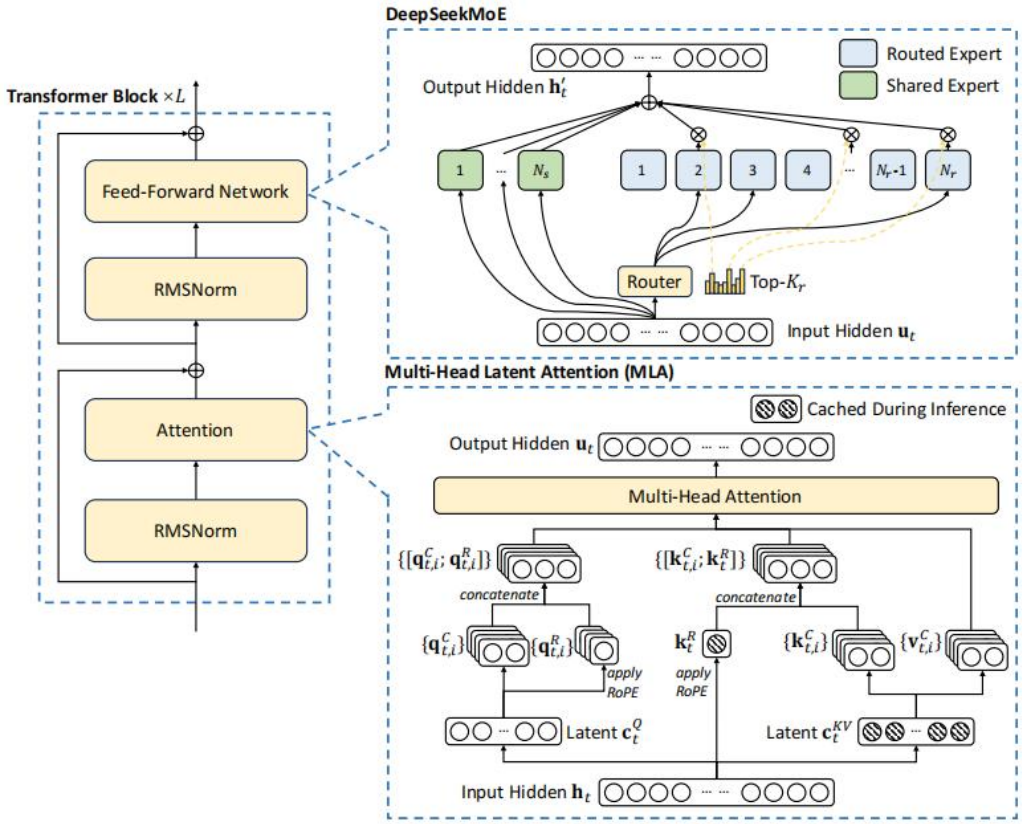


图 2 DeepSeek-V3 的基本架构示意图

在 DeepSeek-V2 之后，我们采用 MLA 和 DeepSeekMoE 来进行有效的推理和经济的训练。

2.1.1.多头潜注意力

值得注意的是，DeepSeek-V3 采用了 MLA 架构。设 表示嵌入维数， ℎ 表示注意头数， ℎ 表示每个头部的维数，h ∈R 表示给定注意层上第 个标记的注意输入。MLA 的核心是对注意键和值的低秩联合压缩，以减少推理过程中的键值（KV）缓存：

$$\boxed{\mathbf{c}_t^{KV}} = W^{DKV}\mathbf{h}_t, \tag{1}$$

$$[\mathbf{k}_{t,1}^C; \mathbf{k}_{t,2}^C; \ldots; \mathbf{k}_{t,n_h}^C] = \mathbf{k}_t^C = W^{UK}\mathbf{c}_t^{KV}, \tag{2}$$

$$\boxed{\mathbf{k}_t^R} = \mathrm{RoPE}(W^{KR}\mathbf{h}_t), \tag{3}$$

$$\mathbf{k}_{t,i} = [\mathbf{k}_{t,i}^C; \mathbf{k}_t^R], \tag{4}$$

$$[\mathbf{v}_{t,1}^C; \mathbf{v}_{t,2}^C; \ldots; \mathbf{v}_{t,n_h}^C] = \mathbf{v}_t^C = W^{UV}\mathbf{c}_t^{KV}, \tag{5}$$

其中 c ∈R 是键的压缩潜在向量，值； （≪ ℎ ℎ ）表示 KV 压缩维数； ∈R × 表示下投影矩阵； ， ∈R ℎ ℎ × 是键和值的上投影矩阵，分别； ∈R ??ℎ × 是用于产生携带旋转位置嵌入（RoPE）的解耦键的矩阵（Su et al., 2024)）；RoPE（·）表示应用 RoPE 矩阵的操作；[·；·]表示连接。需要注意的是，对于 MLA，只有蓝盒向量（即 c 和 k ）需要在生成过程中缓存，这导致显著降低 KV 缓存，同时保持与标准多头注意（MHA）相当的性能（Vaswani et al., 2017）。

对于注意力查询，我们还进行了低秩（最大的不相关的向量的个数）压缩，这可以减少训练过程中的激活记忆：

$$\mathbf{c}_t^Q = W^{DQ}\mathbf{h}_t, \tag{6}$$

$$[\mathbf{q}_{t,1}^C; \mathbf{q}_{t,2}^C; \ldots; \mathbf{q}_{t,n_h}^C] = \mathbf{q}_t^C = W^{UQ}\mathbf{c}_t^Q, \tag{7}$$

$$[\mathbf{q}_{t,1}^R; \mathbf{q}_{t,2}^R; \ldots; \mathbf{q}_{t,n_h}^R] = \mathbf{q}_t^R = \mathrm{RoPE}(W^{QR}\mathbf{c}_t^Q), \tag{8}$$

$$\mathbf{q}_{t,i} = [\mathbf{q}_{t,i}^C; \mathbf{q}_{t,i}^R], \tag{9}$$

其中 c ∈R ′为查询的压缩潜在向量； ′（≪ ℎ ℎ ）表示查询压缩维数；∈R ′× ， ∈R ℎ ℎ × ′分别为查询的下投影矩阵和上投影矩阵；∈R ℎ ℎ × ′是产生携带 RoPE 的解耦查询的矩阵。

最终，将注意查询（q ， ）、键（k ， ）和值（v ， ）组合在一起，得到最终的注意输出 u ：

$$\mathbf{o}_{t,i} = \sum_{j=1}^{t} \mathrm{Softmax}_j\left(\frac{\mathbf{q}_{t,i}^T \mathbf{k}_{j,i}}{\sqrt{d_h + d_h^R}}\right)\mathbf{v}_{j,i}^C, \tag{10}$$

$$\mathbf{u}_t = W^O[\mathbf{o}_{t,1}; \mathbf{o}_{t,2}; \ldots; \mathbf{o}_{t,n_h}], \tag{11}$$

其中， ∈R × ℎ ℎ 为输出投影矩阵。

2.1.2.采用辅助无损耗负载平衡器

DeepSeekMoE 的基本体系结构。对于前馈网络（FFNs），DeepSeek-V3 采用了 DeepSeekMoE 架构 (Dai et al., 2024)。与 GShard （Lepikhin et al.， 2021）相比，DeepSeekMoE 使用了更细粒度的专家，并将一些专家隔离为共享专家。让 u 表示第 个令牌的 FFN 输入，我们计算 FFN 输出 h ‘如下：

$$\mathbf{h}_t' = \mathbf{u}_t + \sum_{i=1}^{N_s} \mathrm{FFN}_i^{(s)}(\mathbf{u}_t) + \sum_{i=1}^{N_r} g_{i,t}\,\mathrm{FFN}_i^{(r)}(\mathbf{u}_t), \tag{12}$$

$$g_{i,t} = \frac{g_{i,t}'}{\sum_{j=1}^{N_r} g_{j,t}'}, \tag{13}$$

$$g_{i,t}' = \begin{cases} s_{i,t}, & s_{i,t} \in \mathrm{Topk}(\{s_{j,t} | 1 \leqslant j \leqslant N_r\}, K_r), \\ 0, & \text{otherwise}, \end{cases} \tag{14}$$

$$s_{i,t} = \mathrm{Sigmoid}\left(\mathbf{u}_t{}^T \mathbf{e}_i\right), \tag{15}$$

其中 和 分别表示共享专家和路由专家的数量；FFN ( )(·) 和 FFN ( )(·) 分别表示第 个共享专家和第 个路由专家； 表示激活的路由专家数量； ，是第 个专家的门控值； ，是第 个 token 与专家的亲和度；e 是第 个路由专家的质心向量；Topk(·,) 表示对于第 个 token 和所有路由专家计算出的亲和度分数中最高的 个分数的集合。与 DeepSeek-V2 略有不同，DeepSeek-V3 使用 S 型函数来计算亲和度分数，并对所有选定的亲和度分数进行归一化处理以产生门控值。

无辅助损失的负载均衡。对于 MoE 模型，不平衡的专家负载会导致路由崩溃 (Shazeer et al., 2017) 并在具有专家并行性的场景中降低计算效率。传统的解决方案通常依赖于辅助损失 (Fedus et al., 2021;Lepikhin et al., 2021) 来避免负载不平衡。然而，过大的辅助损失会损害模型性能 (Wang et al., 2024a)。为了在负载平衡和模型性能之间取得更好的权衡，我们开创了一种无辅助损失的负载平衡策略 (Wang et al., 2024a)，以确保负载平衡。具体来说，我们为每个专家引入一个偏置项 ，并将其添加到相应的亲和分数 ，中，以确定前 K 个路由：

$$g'_{i,t} = \begin{cases} s_{i,t}, & s_{i,t} + b_i \in \text{Topk}(\{s_{j,t} + b_j | 1 \leqslant j \leqslant N_r\}, K_r), \\ 0, & \text{otherwise.} \end{cases} \tag{16}$$

请注意，偏差项仅用于路由。门控值将与 FFN 输出相乘，仍然来自原始的亲和评分 ， 。在训练过程中，我们不断监控每个训练步骤的专家负荷。在每一步结束时，如果对应的专家过载，我们将减少 的偏差项，如果对应的专家负荷不足，我们将增加 ，其中 是一个称为偏差更新速度的超参数。通过动态调整，DeepSeek-V3 在训练过程中保持专家负载平衡，比通过纯辅助损失鼓励负载平衡的模型取得更好的性能。

互补序列辅助损失。虽然 DeepSeek-V3 主要依赖于辅助无损失策略来进行负载平衡，但为了防止任何单个序列内的极端失衡，我们也采用了互补的序列平衡损失：

$$\mathcal{L}_{\text{Bal}} = \alpha \sum_{i=1}^{N_r} f_i P_i, \tag{17}$$

$$f_i = \frac{N_r}{K_r T} \sum_{t=1}^{T} \mathbb{1}\left(s_{i,t} \in \text{Topk}(\{s_{j,t} | 1 \leqslant j \leqslant N_r\}, K_r)\right), \tag{18}$$

$$s'_{i,t} = \frac{s_{i,t}}{\sum_{j=1}^{N_r} s_{j,t}}, \tag{19}$$

$$P_i = \frac{1}{T} \sum_{t=1}^{T} s'_{i,t}, \tag{20}$$

其中平衡因子 是一个超参数，为 DeepSeek-V3 赋值一个极小的值；II（·）表示指示函数； 表示序列中的标记数。顺序上的平衡损失鼓励每个序列上的专家负载得到平衡。

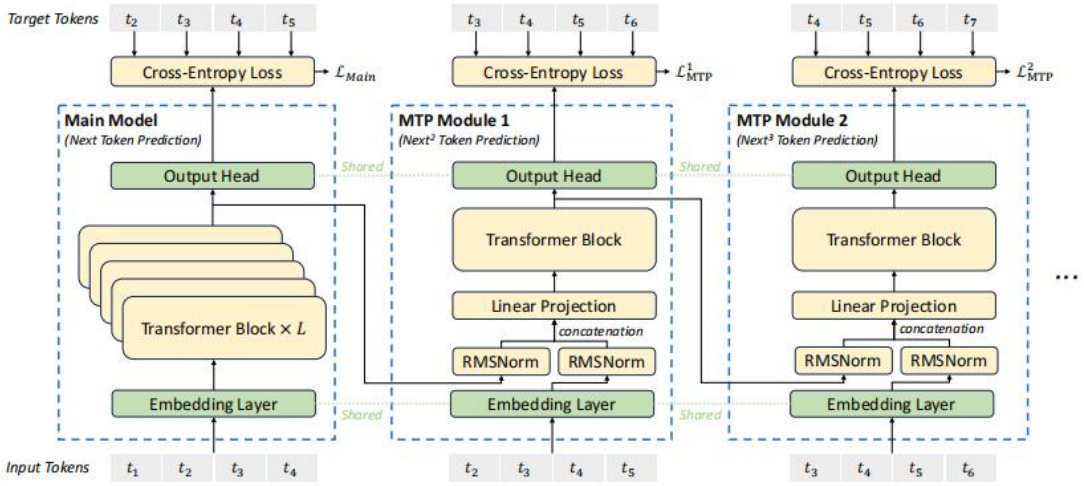


图 3 是我们的多令牌预测（MTP）实现的

说明我们在每个深度保持对每个标记的预测的完整的因果链。

节点限制的路由。与 DeepSeek-V2 使用的受设备限制的路由一样，DeepSeek-V3 也使用了一种受限制的路由机制来限制训练期间的通信成本。简而言之，我们确保每个标记将被发送到最多的 节点，这些节点是根据分布在每个节点上的专家的最高 亲和力得分之和来选择的。在此约束下，我们的 MoE 训练框架几乎可以实现完全的计算-通信重叠。

没有删除令牌。由于有效的负载平衡策略，DeepSeek-V3 在全面训练过程中保持了良好的负载平衡。因此，DeepSeek-V3 在训练过程中不会丢失任何令牌。此外，我们还实现了特

定的部署策略来确保推理负载平衡，因此 DeepSeek-V3 在推理过程中也不会删除令牌。

2.2.多令牌预测

受 Gloeckle 等人（2024）的启发，我们为 DeepSeek-V3 研究并设置了一个多令牌预测（MTP）目标，它将预测范围扩展到每个位置的多个未来令牌。一方面，MTP 目标使训练信号密集，并可能提高数据效率。另一方面，MTP 可能使模型能够预先规划其表示方式，以便更好地预测未来的令牌。图 3 说明了我们的 MTP 的实现。与 Gloeckle 等人（2024）使用独立的输出头并行预测 额外的令牌不同，我们依次预测额外的令牌，并在每个预测深度保持完整的因果链。我们将在本节中详细介绍 MTP 实现的细节。

MTP 模块。具体来说，我们的 MTP 实现使用 个顺序模块来预测 个额外的令牌。第 个 MTP 模块由一个共享的嵌入层 Emb（·）、一个共享的输出头前端（·）、一个 Transformer bl 块 TRM （·）和一个投影矩阵 ∈R ×2 组成。对于第 个输入标记 ，在第 个预测深度，我们首先将（ −1）第 −1∈R 的表示和（ + ）标记 （ + ）∈R 的嵌入与线性投影相结合：

$$\mathbf{h}_i'^k = M_k[\mathrm{RMSNorm}(\mathbf{h}_i^{k-1}); \mathrm{RMSNorm}(\mathrm{Emb}(t_{i+k}))], \tag{21}$$

其中， [·; ·] 表示连接。特别是当 =1 时，h −1 是指主模型给出的表示。注意，对于每个 MTP 模块，它的嵌入层与主模型共享。组合的 h‘ 作为第 个深度的变压器块的输入，以产生当前深度 h 的输出表示：

$$\mathbf{h}_{1:T-k}^k = \mathrm{TRM}_k(\mathbf{h}_{1:T-k}'^k), \tag{22}$$

其中， 表示输入序列长度， ： 表示切片操作（包括左右边界）。最后，以 h 作为输入，共享输出头将计算第 -附加预测令牌 +1+ ∈R 的概率分布，其中 是词汇表大小：

$$P_{i+k+1}^k = \mathrm{OutHead}(\mathbf{h}_i^k). \tag{23}$$

输出头 OutHead（·）将表示线性映射到日志，然后应用 Softmax（·）函数来计算第 个附加令牌的预测概率。此外，对于每个 MTP 模块，它的输出头与主模型共享。我们维持预测的因果链的原则与 EAGLE (Li et al., 2024b) 的相似，但其主要目标是推测性解码(Leviathan et al., 2023; Xia et al., 2023)，而我们利用 MTP 来改进训练。

MTP 训练目标。对于每个预测深度，我们计算一个交叉熵损失 LMTP ：

$$\mathcal{L}_{\mathrm{MTP}}^k = \mathrm{CrossEntropy}(P_{2+k:T+1}^k, t_{2+k:T+1}) = -\frac{1}{T}\sum_{i=2+k}^{T+1} \log P_i^k[t_i], \tag{24}$$

其中 为输入序列长度， 为第 位的地面真值标记， [ ]为 对应的预测概率，由第 位 MTP 模块给出。最后，我们计算所有深度的 MTP 损失的平均值，并将其乘以一个加权因子 ，得到总的 MTP 损失 LMTP，这作为 DeepSeek-V3 的额外训练目标：

$$\mathcal{L}_{\mathrm{MTP}} = \frac{\lambda}{D}\sum_{k=1}^{D} \mathcal{L}_{\mathrm{MTP}}^k. \tag{25}$$

推理中的 MTP。我们的 MTP 策略主要是为了提高主模型的性能，因此在推理过程中，我们可以直接丢弃 MTP 模块，主模型可以独立正常运行。此外，我们还可以重新利用这些 MTP 模块用于推测解码，以进一步提高生成延迟。

3.基础设施

3.1.计算集群

DeepSeek-V3 是在一个配备了 2048 个 NVIDIA H800 GPU 的集群上进行训练的。H800 集群中的每个节点都包含 8 个由 NVLink 和 NVSwitch 连接的 GPU。跨不同节点，利用 iniBand（IB）互连来促进通信。

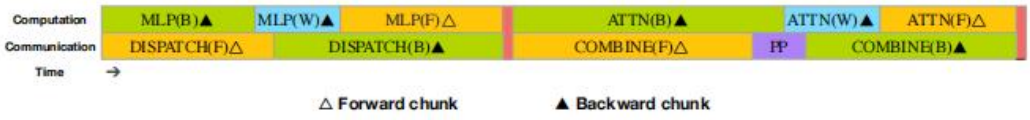


图 4 对于一对单独的正向和向后块的重叠策略（Transformer 块的边界没有对齐）橙色表示向前，绿色表示“向后表示输入”，蓝色表示“向后表示权重”，紫色表示 PP 通信，红色表示障碍。所有人对和 PP 通信都可以完全隐藏。

3.2.训练框架

DeepSeek-V3 的训练是由 HAI-LLM 框架支持的，这是一个由我们的工程师从头开始制作的高效和轻量级的训练框架。总的来说，DeepSeek-V3 应用了 16 路管道并行（PP）(Qi et al., 2023a)、64 路专家并行（EP）(Lepikhin et al., 2021)，以及 ZeRO-1 数据并行（DP）(Rajbhandari et al., 2020)。

为了促进 DeepSeek-V3 的高效训练，我们实施了细致的工程优化。首先，我们设计了一种能够实现高效管道并行性的 DualPipe 算法。与现有的 PP 方法相比，DualPipe 的管道气泡更少。更重要的是，它重叠了跨正向和向后过程的计算和通信阶段，从而解决了跨节点专家并行性所带来的沉重通信开销的挑战。其次，我们开发了高效的跨节点全对通信内核，以充分利用 IB 和 NVLink 带宽，并节省专门用于通信的流式多处理器（SMs）。最后，我们在训练过程中精心优化了内存占用，从而使我们能够在不使用昂贵的张量并行性（TP）的情况下训练 DeepSeek-V3。

3.2.1.双管道和计算-通信重叠

对于 DeepSeek-V3，跨节点专家并行性引入的通信开销导致了无效的计算-通信比约为 1:1。为了解决这一挑战，我们设计了一种创新的管道并行算法 DualPipe，它不仅通过有效地重叠正向和向后计算通信阶段来加速模型训练，而且还减少了管道气泡。

DualPipe 的关键思想是在一对单独的前向和向后方块中重叠计算和通信。具体来说，我们将每个块划分为四个组件：注意力、all-to-all(全对全)的调度、多层感知器（MLP）和 all-to-all(全对全)的组合。特别是，对于一个向后的块，注意力和多层感知器（MLP）被进一步分为两部分，向后输入和向后输入权重，如 ZeroBubble（Qi et al.，2023b）。此外，我们还有一个 PP 通信组件。如图 4 所示，对于一对正向和向后的数据块，我们重新排列这些组件，并手动调整专门用于通信的 GPU SMs 与计算的比例。在这种重叠的策略中，我们可以确保在执行期间，all-to-all(全对全)人和 PP 通信都可以完全隐藏。鉴于有效的重叠策

略，完整的 DualPipe 调度如图 5 所示。它采用了双向管道调度，同时从管道的两端提供微批次，并且很大一部分通信可以完全重叠。这种重叠还确保了随着模型的进一步扩展，只要我们保持恒定的计算-通信比，我们仍然可以跨节点使用细粒度专家，同时实现接近零的 all-to-all(全对全)通信开销。

图 5 在两个方向上 8 PP 等级和 20 微批次的双管道调度。

相反方向的微批次与向前方向的微批次是对称的，因此为了说明简单起见，我们省略了它们的批 ID。两个被共享的黑色边框包围的单元格具有相互重叠的计算和通信。

Method	Bubble	Parameter	Activation
1F1B	$(PP-1)(F+B)$	1×	PP
ZB1P	$(PP-1)(F+B-2W)$	1×	PP
DualPipe (Ours)	$(\frac{PP}{2}-1)(F\&B+B-3W)$	2×	$PP+1$

表 2 在不同的管道并行方法之间的管道气泡和内存使用情况的比较。

表示正向块的执行时间， 表示全向后块的执行时间， 表示“反向权重”块的执行时间，

& 表示两个相互重叠的正向块和向后块的执行时间。

此外，即使在没有重大通信负担的更一般的情况下，DualPipe 仍然显示出效率优势。在表 2 中，我们总结了不同 PP 方法之间的管道气泡和内存使用情况。如表所示，与 ZB1P（Qi et al.，2023b）和 1F1B（Harlap et al.，2018）相比，DualPipe 显著减少了管道气泡，而仅将 的激活记忆峰值提高了 1 倍。虽然 DualPipe 需要保留两个模型参数的副本，但这并不会显著增加内存消耗，因为我们在训练过程中使用了一个较大的 EP 大小。与嵌合体（Li and Hoefler，2021）相比，DualPipe 只要求管道阶段和微批次由二分，而不需要微批次由管道阶段二分。此外，对于 DualPipe，气泡和激活记忆都不会随着微批次数量的增加而增加。

3.2.2.跨节点全能通信的高效实现

为了确保 DualPipe 具有足够的计算性能，我们定制了高效的跨节点全对所有通信内核（包括调度和组合），以节省专门用于通信的 SMs 的数量。内核的实现与 MoE 门控算法和集群的网络拓扑共同设计。具体来说，在我们的集群中，跨节点的 GPU 与 IB 完全互联，节点内的通信通过 NVLink 进行处理。NVLink 提供了 160 GB/s 的带宽，大约是 IB（50 GB/s）的 3.2 倍。为了有效地利用 IB 和 NVLink 的不同带宽，我们将每个令牌最多 4 个节点，从而减少了 IB 流量。对于每个令牌，当它做出路由决策时，它将首先通过 IB 传输到其目标节点上具有相同节点内索引的 GPU。.一旦它到达目标节点，我们将努力确保它通过 NVLink 立即转发给托管其目标专家的特定 GPU，而不会被随后到达的令牌所阻止。这样，通过 IB 和 NVLink 的通信就完全重叠了，每个令牌可以有效地选择每个节点平均 3.2 个专家，而不会产生来自 NVLink 的额外开销。这意味着，尽管 DeepSeek-V3 在实践中只选择了 8 个路由专家，但它可以扩大这个数字到最多 13 个专家（4 个节点×3.2 个专家/节点），同时保持相同的通信成本。总的来说，在这种通信策略下，只有 20 个 SMs 就足以充分利用 IB 和 NVLink 的带宽。

详细地说，我们采用了扭曲专门化技术（Bauer et al.，2014），并将 20 个 SMs 划分为

10 个通信通道。在调度过程中，(1) IB 发送，(2)IB 到 NVLink 转发，(3) NVLink 接收处理。分配给每个通信任务的扭曲数将根据所有 SMs 中的实际工作负载进行动态调整。类似地，在合并过程中，(1) NVLink 发送，(2) NVLink-to-IB 转发和积累，(3) IB 接收和积累也通过动态调整的扭曲来处理。此外，调度和组合核都与计算流重叠，因此我们也考虑了它们对其他 SM 计算内核的影响。具体来说，我们使用定制的 PTX（并行线程执行）指令，并自动调整通信块的大小，这大大减少了 L2 缓存的使用和对其他 SMs 的干扰。

3.2.3.大大节省内存和最小的开销

为了减少训练期间的内存占用，我们采用了以下技术。

RMSNorm 和 MLA 上投影的重新计算。在反向传播过程中，我们重新计算所有的 RMSNorm 操作和 MLA 向上投影，从而消除了持续存储其输出激活的需要。该策略的开销较小，从而显著减少了存储激活的内存需求。

CPU 中的指数移动平均值。在训练过程中，我们保留了模型参数的指数移动平均值（EMA），用于早期估计学习速率衰减后的模型性能。EMA 参数存储在 CPU 内存中，并在每个训练步骤后进行异步更新。这种方法允许我们维护 EMA 参数，而不会产生额外的内存或时间开销。

用于多标记预测的共享嵌入和输出头。使用 DualPipe 策略，我们在相同的 PP 级别上部署模型的最浅层（包括嵌入层）和最深层（包括输出头）。这种安排使得 MTP 模块和主模型之间的参数和梯度、共享的嵌入和输出头的物理共享成为可能。这种物理共享机制进一步提高了我们的内存效率。

3.3.FP8 训练

受低精度训练的最新进展启发(Dettmers et al., 2022; Noune et al., 2022;Peng et al., 2023b)，我们提出了一个细粒度的混合精度框架，利用 FP8 数据格式训练 DeepSeek-V3。虽然低精度训练有很大的希望，但它往往受到激活、权重和梯度中存在异常值的限制(Fishman et al., 2024; He et al.; Sun et al., 2024)。虽然在推理量化方面取得了重大进展(Frantar et al., 2022; Xiao et al., 2023)，但证明低精度技术在大规模语言模型中成功应用的研究相对较少预训练 (Fishman et al., 2024)。为了解决这一挑战并有效地扩展 FP8 格式的动态范围，我们引入了一种细粒度量化策略：使用 1 个× 元素进行块分组或使用 × 元素进行块分组。在我们的提高精度的积累过程中，相关的去量化开销在很大程度上被减轻了，这是实现精确的 FP8 通用矩阵乘法（GEMM）的一个关键方面。此外，为了进一步减少 MoE 训练中的内存和通信开销，我们在 FP8 中缓存和调度激活，同时在 BF16 中存储低精度优化器状态。我们在类似于 DeepSeek-V2-Lite 和 DeepSeekV2 的两个模型尺度上验证了所提出的 FP8 混合精度框架，对大约 1 万亿个令牌进行了训练（见附录 B.1 中的更多细节）。值得注意的是，与 BF16 基线相比，我们的 fp8 训练模型的相对损失误差始终保持在低于 0.25%，这一水平在可接受的训练随机性的范围内。

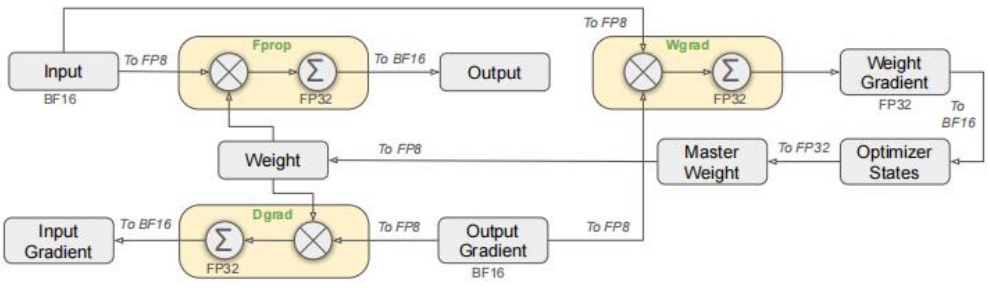


图 6 采用 FP8 数据格式的整体混合精度框架。为了澄清，只说明了线性算子。

3.3.1.混合精度框架

基于在低精度训练中广泛采用的技术（Kalamkar et al., 2019;Narang et al., 2017），我们提出了一个 FP8 训练的混合精度框架。在这个框架中，大多数计算密度操作是在 FP8 中进行的，而一些关键操作在原始数据格式中策略性地维护，以平衡训练效率和数值稳定性。整个框架如图 6 所示。

首先，为了加速模型训练，大部分的核心计算内核，即 GEMM 操作，都是以 FP8 精度实现的。这些 GEMM 操作接受 FP8 张量作为输入，并在 BF16 或 FP32 中产生输出。如图 6 所示，与线性操作符相关的三个 GEMMs，即 Fprop（前通过）、Dgrad（激活后通过）和 Wgrad（权重后通过），都在 FP8 中执行。与原 BF16 方法相比，该设计方法的计算速度提高了一倍。此外，FP8 Wgrad GEMM 允许将激活存储在 FP8 中，以便用于反向传递。这大大减少了内存消耗。

尽管 FP8 格式具有效率优势，但由于某些操作员对低精度计算的敏感性，它们仍然需要更高的精度。此外，一些低成本的运营商也可以利用更高的精度，而对总体训练成本的开销可以忽略不计。因此，经过仔细的调查，我们保持了以下组件的原始精度（如 BF16 或 FP32）：嵌入模块、输出头、MoE 门控模块、归一化操作符和注意操作符。这些高精度的定向保留确保了 DeepSeek-V3 的稳定的训练动态。为了进一步保证数值的稳定性，我们以更高的精度存储主权重、权重梯度和优化器状态。然而这些高精度的组件产生了一些内存开销，它们的影响可以通过在我们的分布式训练系统中跨多个 DP 级别的高效共享来最小化。

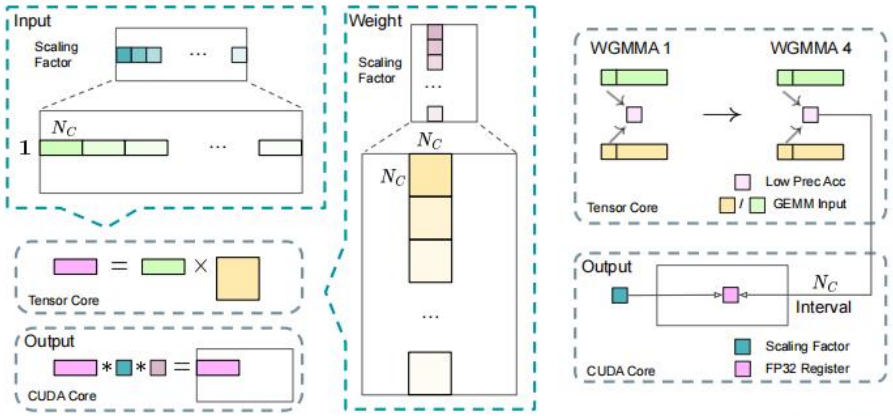


图 7 (a)我们提出了一种细粒度的量化方法来减轻由特征异常值引起的量化误差；为了简单起见，只说明了 Fprop。(b)结合我们的量化策略，我们通过在 = 128 元素 MMA 的间隔内提高 FP8 GEMM 的精度，以实现高精度积累。

3.3.2.通过量化和乘法提高了精度

基于我们的混合精度 FP8 框架，我们引入了几种策略来提高低精度训练精度，重点关注量化方法和乘法过程。

细粒度量化。在低精度的训练框架中，由于 FP8 格式的动态范围有限，溢出和欠流是常见的挑战，这受到其减少的指数位的限制。作为一种标准实践，通过将输入张量的最大绝对值缩放到 FP8 的最大可表示值，输入分布与 FP8 格式的可表示范围对齐（Narang et al.，2017）。该方法使低精度训练对激活异常值高度敏感，从而严重降低量化的精度。为了解

决这个问题，我们提出了一种细粒度的量化方法，它可以在更细粒度的级别上应用缩放。如图 7 (a)所示，(1)对于激活，我们在 1x128 平铺的基础上分组和缩放元素（即每 128 个通道的每个令牌）；(2)对于权重，我们以 128x128 块的基础对元素进行分组和缩放（即每 128 个输出通道的每 128 个输入通道）。这种方法通过根据较小的元素群来调整规模，确保了量化过程可以更好地适应异常值。在附录 B.2 中，我们进一步讨论了当我们以与权重量化相同的方式在块的基础上分组和规模激活时的训练不稳定性。

我们的方法中的一个关键修改是沿着 GEMM 操作的内维引入每组缩放因子。在标准的 FP8 GEMM 中并不直接支持此功能。然而，结合我们精确的 FP32 积累策略，它可以被有效地实现。

值得注意的是，我们的细粒度量化策略与微尺度格式的想法高度一致 (Rouhani et al., 2023b),，而 NVIDIA 下一代图形处理器的张量核心（Blackwell series）已经宣布支持具有更小量化粒度的微尺度格式（NVIDIA，2024a）。我们希望我们的设计可以作为未来工作的参考，以跟上最新的 GPU 架构。

增加积累精度。低精度 GEMM 操作经常存在下流问题，其精度很大程度上依赖于高精度积累，这通常在 FP32 精度中执行(Kalamkar et al., 2019; Narang et al., 2017)。然而，我们观察到 FP8 GEMM 在 NVIDIA H800GPU 上的积累精度仅被限制在 14 位左右，这明显低于 FP32 的积累精度。当内部维数 K 很大时，这个问题将变得更加明显（Wortsman et al.，2023），这是大规模模型训练中的典型场景，批量大小和模型宽度增加。以使用 K = 4096 的两个随机矩阵的 GEMM 运算为例，在我们的初步测试中，张量核中有限的累积精度导致的最大相对误差接近 2%。尽管存在这些问题，有限的积累精度仍然是少数 FP8 框架中的默认选项（NVIDIA，2024b），严重限制了训练精度。

为了解决这个问题，我们采用了推广到 CUDA Cores 以提高精度的策略(Thakkar et al., 2023)。这个过程如图 7 (b)所示具体地说，在张量核上执行 MMA（矩阵乘累加）的过程中，使用有限的位宽累积中间结果。一旦达到 的时间间隔，这些部分结果将被复制到 CUDA Cores 上的 FP32 寄存器中，在那里执行全精度的 FP32 积累。如前所述，我们的细粒度量化应用于沿内维 k 的每组尺度因子。这些尺度因子可以有效地乘以在 CUDA Cores 上作为去量化过程，以最小的额外计算代价。

值得注意的是，这种修改降低了单个疣组的 WGMMA（Warpgroup-level Matrix Multiply-Accumulate)）指令问题率。但是，在 H800 体系结构中，两个 WGMMA 通常同时存在：一个执行升级操作，另一个能够执行 MMA 操作。这种设计使两个操作能够重叠，保持张量核的高利用率。基于我们的实验，设置 = 128 元素，相当于 4 个 WGMMAs，代表了可以在不引入大量开销的情况下显著提高精度的最小积累间隔。

指数上的尾部。与之前工作采用的混合 FP8 格式(NVIDIA, 2024b; Peng et al., 2023b; Sun et al., 2019b)，在 Fprop 中使用 E4M3（4 位指数和 3 位尾数）和 Dgrad 和 E5M2（5 位指数和 2 位尾数），我们在所有张量上采用 E4M3 格式以获得更高的精度。我们将这种方法的可行性归因于我们的细粒度量化策略，即平铺和块级缩放。通过在较小的元素组上进行操作，我们的方法有效地在这些分组的元素之间共享指数位，减轻了有限的动态范围的影响。

在线量化。延迟量化用于张量量化框架工作(NVIDIA, 2024b; Peng et al., 2023b)，它在之前的迭代中保持最大绝对值的历史来推断当前值。为了确保准确的比例和简化框架，我们在线计算每个 1x128 激活块或 128x128 权重块的最大绝对值。在此基础上，我们推导出缩放因子，然后将激活或权重在线量化为 FP8 格式。

3.3.3.低精度的存储和通信

结合我们的 FP8 训练框架，我们通过将缓存的激活和优化器状态压缩为较低精度的格

式，进一步减少了内存消耗和通信开销。

低精度优化器状态。我们采用 BF16 数据格式，而不是 FP32 来跟踪 AdamW (Loshchilov and Hutter, 2017)优化器中的第一和第二力矩，而不会引起明显的性能下降。然而，主权重（由优化器存储）和梯度（用于批大小积累）仍然保留在 FP32 中，以确保在整个训练过程中的数值稳定性。

低精度激活。如图 6 所示，Wgrad 操作是在 FP8 中执行的。为了减少内存消耗，为线性操作符以 FP8 格式的向后传递缓存激活是一种自然的选择。然而，在低成本的高精度训练中，我们特别考虑了一些操作人员：

(1)注意操作符后的线性输入。这些激活也被用于注意操作符的向后传递，这使得它对精度很敏感。我们专门为这些激活采用了定制的 E5M6 数据格式。此外，这些激活将从 1x128 的量化块转换为 128x1 的反向传递块。为了避免引入额外的量化误差，所有的比例因子都是圆比例的，即积分幂为 2。

(2)教育部中 SwiGLU 操作符的输入。为了进一步降低内存成本，我们缓存了 SwiGLU 操作符的输入，并在反向传递中重新计算其输出。这些激活也通过我们的细粒度量化方法存储在 FP8 中，在内存效率和计算精度之间取得了平衡。

低精密通信。通信带宽是 MoE 模型训练的关键瓶颈。为了缓解这一挑战，我们将 MoE 向上投影到 FP8 之前的激活进行量化，然后应用调度组件，该组件与 MoE 向上投影中的 FP8 Fprop 兼容。就像注意算子后的线性输入一样，这种激活的比例因子是 2 的积分幂。类似的策略也应用于 MoE 下投影前的激活梯度。对于前向和后向组合组件，我们在 BF16 中保留了它们，以保持在训练管道的关键部分的训练精度。

3.4.推理和部署

我们在 H800 集群上部署了 DeepSeek-V3，其中每个节点内的 GPU 使用 NVLink 相互连接，集群内的所有 GPU 通过 IB 完全相互连接。为了同时确保在线服务的服务水平目标（SLO）和高吞吐量，我们采用了以下部署策略，将预填充和解码阶段分开。

3.4.1.预填充

预填充阶段的最小部署单元由 4 个节点和 32 个 GPU 组成。注意部分采用具有序列并行（SP）的 4 路张量并行（TP4），并结合 8 路数据并行（DP8）。它的小 TP 大小为 4，限制了 TP 通信的开销。对于 MoE 部分，我们使用了 32 路专家并行性（EP32），它确保了每个专家处理足够大的批处理规模，从而提高了计算效率。对于 MoEall-to-all(全对全)通信，我们使用与训练相同的方法：首先通过 IB 跨节点传递令牌，然后通过 NVLink 在节点内 GPU 之间进行转发。特别地，我们对浅层中密集的多层感知器（MLP）使用单向张量并行性，以节省 TP 通信。

为了在 MoE 部分的不同专家之间实现负载平衡，我们需要确保每个 GPU 处理的令牌数量大致相同。为此，我们引入了冗余专家的部署策略，该策略复制高负载专家并冗余地部署它们。根据在在线部署期间收集的统计数据检测到高负载专家，并定期进行调整（例如，每 10 分钟一次）。在确定冗余专家集之后，我们根据观察到的负载仔细地重新排列节点内 GPU 之间的专家，在不增加跨节点全对通信开销的情况下尽可能平衡跨 GPU 的负载。对于 DeepSeek-V3 的部署，我们为预填充阶段设置了 32 名冗余专家。对于每个 GPU，除了它原来托管的 8 个专家之外，它还将托管一个额外的冗余专家。

此外，在预处理阶段，为了提高吞吐量和隐藏 all-to-all(全对全)和 TP 通信的开销，我们同时处理两个计算工作量相似的微批，将一个微批的注意力和 MoE 与另一个微批的调度和组合重叠。

最后，我们正在为专家探索一种动态冗余策略，其中每个 GPU 拥有更多的专家（例如，16 个专家），但在每个推理步骤中只有 9 个专家将被激活。在每一层的 all-to-all(全对全)操作开始之前，我们动态地计算全局最优路由方案。考虑到在预填充阶段所涉及的大量计算，计算这种路由方案的开销几乎可以忽略不计。

3.4.2.解码

在解码过程中，我们将共享的专家视为一个路由的专家。从这个角度来看，每个令牌将在路由过程中选择 9 个专家，其中共享的专家被视为一个始终被选择的重载专家。解码阶段的最小部署单元由 40 个节点和 320 个 GPU 组成。注意部分采用 TP4 和 SP，与 DP80 结合，而 MoE 部分使用 EP320。对于 MoE 部分，每个 GPU 只托管一个专家，并且 64 个 GPU 负责托管冗余的专家和共享的专家。调度和组合部件的全对全的通信是通过 IB 上的直接点对点传输来实现的，以实现低延迟。此外，我们利用 IBGDA（NVIDIA，2022）技术来进一步最小化延迟并提高通信效率。

与预填充类似，我们根据在线服务中的统计专家负载，会在一定的时间间隔内定期确定冗余专家集。然而，我们不需要重新安排专家，因为每个 GPU 只有一个专家。我们也在探索解码的动态冗余策略。然而，这需要更仔细的算法优化，计算全局最优路由方案，并与调度核融合，以减少开销。

此外，为了提高吞吐量和隐藏 all-to-all(全对全)通信的开销，我们还在探索在解码阶段同时处理具有相似计算工作负载的两个微批。与预填充不同，注意力在解码阶段消耗了更多的时间。因此，我们将一个微批的注意力与另一个微批的 dispatch+MoE+combine 相重叠。在解码阶段，每个专家的批处理大小相对较小（通常在 256 个令牌范围内），而且瓶颈是内存访问，而不是计算。由于 MoE 部分只需要加载一个专家的参数，因此内存访问开销最小，因此使用更少的 sm 不会显著影响总体性能。因此，为了避免影响注意力部分的计算速度，我们只能分配一小部分 SMs 来 dispatch+MoE+combine。

3.5.关于硬件设计的建议

基于我们实现的全能通信和 FP8 训练方案的基础，我们向人工智能硬件供应商提出以下芯片设计建议。

3.5.1.通信硬件

在 DeepSeek-V3 中，我们实现计算和通信的重叠来隐藏计算过程中的通信延迟。与串行计算和通信相比，这大大降低了对通信带宽的依赖性。然而，当前的通信实现依赖于昂贵的 SMs（例如，我们为此在 H800 GPU 中可用的 132 个 SMs 中分配了 20 个），这将限制计算吞吐量。此外，使用 SMs 进行通信会导致显著的低效，因为张量核仍然完全没有得到充分利用。

目前，SMs 主要为全能通信执行以下任务：

转发数据 在 IB（InfiniBand）和 NVLink 域之间，同时聚合来自单个 GPU 的同一节点内的多个 GPU 的 IB 流量。

传输数据 在 RDMA 缓冲区（已注册的 GPU 内存区域）和输入/输出缓冲区之间。

执行减少操作 用于 all-to-all(全对全)组合。

管理细粒度的内存布局 在跨 IB 和 NVLink 域向多个专家传输分块数据期间。

我们希望看到未来的供应商开发硬件，将从有价值的计算单元 SM 中卸载这些通信任务，作为 GPU 协议处理器或网络协议处理器，如 NVIDIA SHARP Graham et al. (2016)。此外，为了降低应用程序编程的复杂性，我们的目标是将该硬件从计算单元的角度统一 IB

（扩展）和 NVLink（扩展）网络。有了这个统一的接口，计算单元可以简单地通过提交基于简单原语的通信请求，在整个 IB-NVLink 统一域中完成读、写、多播和减少等操作。

3.5.2.计算硬件

在张量核中的 FP8 GEMM 积累精度较高。

在当前的 NVIDIA Hopper 架构的张量核心实现中，FP8 GEMM（一般矩阵乘法）采用定点累积，根据加法前的最大指数通过右移来对齐尾数积。我们的实验表明，在符号填充右移后，它只使用每个尾数乘积的最高 14 位，并截断超过这个范围的位。然而，例如，为了从 32 个 FP8×FP8 乘法的积累中获得精确的 FP32 结果，至少需要 34 位的精度。因此，我们建议未来的芯片设计提高张量核的积累精度，以支持全精度的积累，或者根据训练和推理算法的精度要求选择合适的积累位宽。这种方法确保误差保持在可接受的范围内，同时保持计算效率.

支持块和分块量化。目前的 GPU 只支持每张量量化，缺乏对细粒度量化的原生支持，比如我们的小块量化和细块量化。在当前的实现中，当达到 间隔时，部分结果将从张量核复制到 CUDA 核中，再乘以缩放因子，并添加到 CUDA 核上的 FP32 寄存器中。虽然与我们精确的 FP32 积累策略相结合，去量化开销显著减轻，但张量核和 CUDA 核之间的频繁的数据移动仍然限制了计算效率。因此，我们建议未来的芯片通过使张量核能够接收缩放因子并实现具有组缩放的 MMA 来支持细粒度量化。这样，整个部分和的积累和去量化就可以直接在张量核内完成，直到产生最终结果为止，避免了频繁的数据移动。

支持在线量化。目前的实现难以有效地支持在线量化，尽管它的有效性在我们的研究中得到了证明。在现有的过程中，我们需要从 HBM（高带宽内存）中读取 128 个 BF16 激活值（之前计算的输出）进行量化，然后将量化的 FP8 值写入 HBM，以便对 MMA 再次读取。为了解决这种低效率的问题，我们建议未来的芯片将 FP8 转换和 TMA（张量内存加速器）访问集成到一个融合操作中，这样就可以在激活从全局内存转移到共享内存的过程中完成量化，避免频繁的内存读写。我们还建议支持加速扭曲铸造指令，这进一步促进了层标准化和 FP8 铸造的更好融合。或者，可以采用近内存计算方法，将计算逻辑放置在 HBM 附近。在这种情况下，当 BF16 元件从 HBM 读入 GPU 时，它们可以直接被强制转换到 FP8 中，从而减少了大约 50%的芯片外内存访问。

支持已转置的 GEMM 操作。当前的架构使得将矩阵转换与 GEMM 操作融合变得很麻烦。在我们的工作流程中，正向传递过程中的激活被量化为 1x128 FP8 贴图并进行存储。在反向传递过程中，矩阵需要被读出、去量化、转置、重新量化成 128x1 的方块，并存储在 HBM 中。为了减少内存操作，我们建议未来的芯片在 MMA 操作之前能够从共享内存中直接转位读取矩阵，以实现训练和推理所需的精度。结合 FP8 格式转换和 TMA 访问的融合，这种增强将显著简化量化工作流程。

4.预训练

4.1.数据建设

与 DeepSeek-V2 相比，我们通过提高数学样本和编程样本的比例来优化训练前的语料库，同时扩大了英语和汉语之外的多语言覆盖范围。此外，我们的数据处理管道也进行了改进，以在保持语料库多样性的同时最小化冗余。受 Ding 等人（2024）的启发，我们实现了针对数据完整性的文档打包方法，但在训练过程中没有纳入跨样本注意屏蔽。最后，DeepSeek-V3 的训练语料库在我们的标记器中包含了 14.8T 的高质量和多样化的标记。

在 DeepSeekCoder-V2（DeepSeek-AI，2024a）的训练过程中，我们观察到中间填充（FIM）策略并不影响下一个标记预测能力，同时使模型能够基于上下文线索准确预测中间文本。为了与 DeepSeekCoder-V2 相一致，我们还在 DeepSeek-V3 的预训练中加入了 FIM 策略。具体来说，我们使用前缀后缀中间（PSM）框架来结构数据如下：

$$<|\text{fim\_begin}|>f_{\text{pre}}<|\text{fim\_hole}|>f_{\text{suf}}<|\text{fim\_end}|>f_{\text{middle}}<|\text{eos\_token}|>.$$

该结构作为预包装过程的一部分应用于文档级别。FIM 策略的应用为 0.1，与 PSM 框架一致。

DeepSeek-V3 的令牌生成器使用了字节级 BPE(Shibata et al., 1999)，扩展词汇表为 128K 令牌。我们的标记器的预标记器和训练数据被修改，以优化多语言压缩效率。此外，与 DeepSeek-V2 相比，新的预标记器引入了结合标点符号和换行符的令牌。然而，当模型处理没有终端断行的多行提示时，这个技巧可能会引入令牌边界偏差（Lundberg，2023,2023），特别是对于少镜头的评估提示。为了解决这个问题，我们在训练过程中随机分割了一定比例的组合标记，这使模型暴露在更广泛的特殊情况下，并减轻了这种偏差。

4.2.超参数

模型超参数。我们将 Transformer 层数设置为 61，隐藏维度设置为 7168。所有可学习参数均为随机初始化，标准偏差为 0.006。在 MLA 中，我们将注意力头数 ℎ 设置为 128，人均维度 ℎ 设置为 128。KV 压缩维度 设置为 512，查询压缩维度 ‘设置为 1536。对于解耦的查询和键，我们将人均维度 ℎ 设置为 64。除前三层外，我们将所有的 ffn 都替换为 MoE 层。每个 MoE 层由 1 个共享专家和 256 个路由专家组成，其中每个专家的中间隐藏维度为 2048。在路由的专家中，每个令牌将激活 8 个专家，并确保每个令牌最多被发送到 4 个节点。多令牌预测深度 设置为 1，即除了精确的下一个令牌外，每个令牌还将预测一个额外的令牌。作为 DeepSeek-V2，DeepSeek-V3 还在压缩的潜在向量之后使用额外的 RMSNorm 层，并在宽度瓶颈处增加额外的缩放因子。在此配置下，DeepSeek-V3 总共包含 671B 个参数，其中每个令牌有 37 个 B 被激活。

训练超参数。我们使用 AdamW 优化器(Loshchilov and Hutter, 2017)，超参数设置为 1=0.9， 2=0.95， weight_decay = 0.1。我们在训练前将最大序列长度设置为 4K，并在 14.8T 标记上进行预训练 DeepSeek-V3。对于学习速率调度，我们首先在前 2K 步中将其从 0 线性增加到 2.2×10−4。然后，我们保持 2.2×10−4 的恒定学习率，直到模型消耗了 10T 的训练令牌。随后，我们在 4.3T 标记中，根据余弦衰减曲线，将学习速率逐渐衰减到 2.2×10−5。在最后 500B 令牌的训练过程中，我们在前 333B 令牌中保持 2.2×10−5 的恒定学习速率，并在剩下的 167B 令牌中切换到另一个 7.3×10−6 的恒定学习速率。梯度剪切范数被设置为 1.0。我们采用批大小调度策略，在前 469B 令牌的训练中，批大小从 3072 逐渐增加到 15360，然后在剩余的训练中保留 15360。我们利用管道并行性在不同的 GPU 上部署模型的不同层，对于每一层，路由专家将统一部署在属于 8 个节点的 64 个 GPU 上。对于受节点限制的路由，每个令牌最多将被发送到 4 个节点（即， = 4）。对于无辅助损失的负载平衡，我们将前 14.3T 令牌的偏置更新速度 设置为 0.001，将剩下的 500B 令牌设置为 0.0。对于平衡损失，我们将 设置为 0.0001，只是为了避免任何单一序列内的极端不平衡。前 10 个 T 令牌的 MTP 损失重量 设置为 0.3，其余的 4.8T 令牌设置为 0.1。

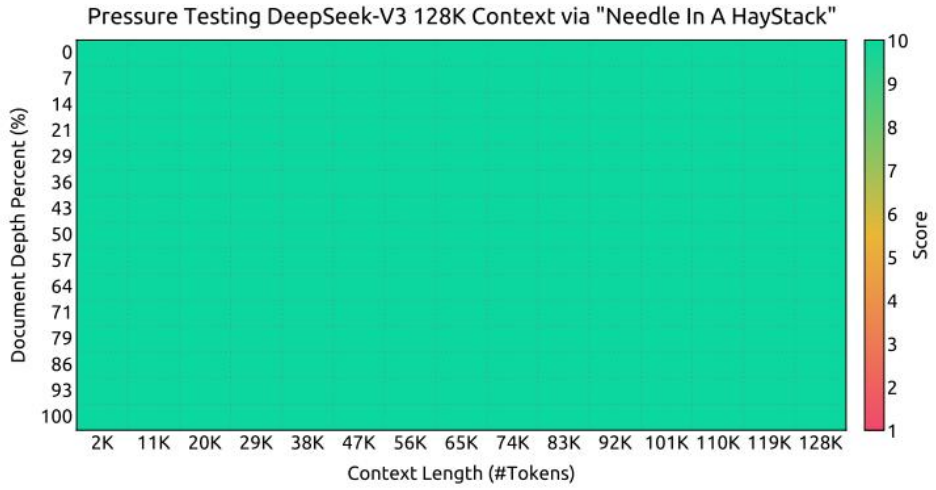


图 8 “干草堆针”（NIAH）测试的|评估结果。DeepSeek-V3 在高达 128K 的所有上下文窗口长度上都表现良好。

4.3.长上下文扩展

我们采用了类似于 DeepSeek-V2（DeepSeek-AI，2024c）的方法，以在 DeepSeek-V3 中实现长上下文能力。在训练前阶段之后，我们应用 YaRN（Peng et al.，2023a）进行上下文扩展，并执行两个额外的训练阶段，每个阶段包括 1000 步，逐步将上下文窗口从 4K 扩展到 32K，然后扩展到 128K。YaRN 配置与 DeepSeek-V2 中使用的配置一致，只应用于解耦的共享密钥 k 。超参数在两个阶段中保持相同，与比例 = 40， = 1， = 32，和比例因子v = 0.1 ln +1。在第一阶段，序列长度设置为 32K，批处理大小为 1920。在第二阶段，序列长度增加到 128K，批量大小减少到 480。这两个阶段的学习速率都设置为 7.3×10−6，与训练前阶段的最终学习速率相匹配。

通过这种两阶段的扩展训练，DeepSeek-V3 能够处理高达 128K 长度的输入，同时保持强大的性能。图 8 说明了 DeepSeek-V3，在经过监督微调后，在“干草堆中的针”（NIAH）测试中取得了显著的性能，证明了在高达 128K 的上下文窗口长度上显示出一致的鲁棒性。

4.4.评估

4.4.1.评价基准

DeepSeek-V3 的基础模型在以英语和汉语为主的多语言语料库上进行了预训练，因此我们在一系列主要以英语和汉语为主的基准测试以及多语言基准测试上评估其性能。我们的评估是基于我们在 HAI-LLM 框架中集成的内部评估框架。被考虑过的基准被分类和列出如下，其中有下划线的基准是中文的，有双下划线的基准是多语言的：

多主题多项选择 数据集包括 MMLU (Hendrycks et al., 2020)、 MMLU Redux (Gema et al., 2024)、 MMLU-Pro (Wang et al., 2024b)、MMMLU (OpenAI, 2024b)、C-Eval(Huang et al., 2023) 和 CMMLU (Li et al., 2023)。

语言理解和推理 数据集包括 HellaSwag (Zellers et al., 2019)、PIQA (Bisk et al., 2020)、ARC (Clark et al., 2018) 和 BigBench Hard (BBH) (Suzgun et al., 2022)。

闭门问答 数据集包括 TriviaQA (Joshi et al., 2017) 和 NaturalQuestions (Kwiatkowski et al., 2019)。

阅读理解 数据集包括 RACE Lai et al. (2017), DROP (Dua et al., 2019)、C3 (Sun et al., 2019a) 和 CMRC (Cui et al., 2019)。

参考消除歧义 数据集包括 CLUEWSC (Xu et al., 2020) 和 WinoGrande Sakaguchi et al. (2019)。

语言模型化 数据集包括 Pile (Gao et al., 2020)。

中文理解和文化 数据集包括 CCPM (Li et al., 2021)。

数学 数据集包括 GSM8K (Cobbe et al., 2021)、 MATH (Hendrycks et al., 2021)、MGSM(Shi et al., 2023) 和 CMath (Wei et al., 2023)。

编程 数据集包括 HumanEval (Chen et al., 2021)、 LiveCodeBench-Base (0801-1101) (Jain et al., 2024)、MBPP (Austin et al., 2021) 和 CRUXEval (Gu et al., 2024)。

标准化考试 数据集包括 AGIEval (Zhong et al., 2023)。请注意，AGIEval 包括英语和中文两个子集。

根据我们之前的工作（DeepSeek-AI，2024b，c），我们采用基于困惑的评估数据集包括 HellaSwag, PIQA, WinoGrande, RACE-Middle, RACE-High,MMLU, MMLU-Redux, MMLU-Pro, MMMLU, ARC-Easy, ARC-Challenge, C-Eval, CMMLU,C3, and CCPM,，并采用基于生成的评估 TriviaQA、, NaturalQuestions, DROP,MATH, GSM8K, MGSM, HumanEval, MBPP, LiveCodeBench-Base, CRUXEval, BBH, AGIEval,CLUEWSC, CMRC 和 CMath.。此外，我们对桩测试执行基于语言建模的评估，并使用比特/字节（BPB）作为度量，以保证使用不同标记器的模型之间的公平比较。

	Benchmark (Metric)	# Shots	DeepSeek-V2 Base	Qwen2.5 72B Base	LLaMA-3.1 405B Base	DeepSeek-V3 Base
	Architecture	-	MoE	Dense	Dense	MoE
	# Activated Params	-	21B	72B	405B	37B
	# Total Params	-	236B	72B	405B	671B
English	Pile-test (BPB)	-	0.606	0.638	**0.542**	0.548
	BBH (EM)	3-shot	78.8	79.8	82.9	**87.5**
	MMLU (EM)	5-shot	78.4	85.0	84.4	**87.1**
	MMLU-Redux (EM)	5-shot	75.6	83.2	81.3	**86.2**
	MMLU-Pro (EM)	5-shot	51.4	58.3	52.8	**64.4**
	DROP (F1)	3-shot	80.4	80.6	86.0	**89.0**
	ARC-Easy (EM)	25-shot	97.6	98.4	98.4	**98.9**
	ARC-Challenge (EM)	25-shot	92.2	94.5	**95.3**	**95.3**
	HellaSwag (EM)	10-shot	87.1	84.8	**89.2**	**88.9**
	PIQA (EM)	0-shot	83.9	82.6	**85.9**	84.7
	WinoGrande (EM)	5-shot	**86.3**	82.3	85.2	84.9
	RACE-Middle (EM)	5-shot	73.1	68.1	**74.2**	67.1
	RACE-High (EM)	5-shot	52.6	50.3	**56.8**	51.3
	TriviaQA (EM)	5-shot	80.0	71.9	**82.7**	**82.9**
	NaturalQuestions (EM)	5-shot	38.6	33.2	**41.5**	40.0
	AGIEval (EM)	0-shot	57.5	75.8	60.6	**79.6**
Code	HumanEval (Pass@1)	0-shot	43.3	53.0	54.9	**65.2**
	MBPP (Pass@1)	3-shot	65.0	72.6	68.4	**75.4**
	LiveCodeBench-Base (Pass@1)	3-shot	11.6	12.9	15.5	**19.4**
	CRUXEval-I (EM)	2-shot	52.5	59.1	58.5	**67.3**
	CRUXEval-O (EM)	2-shot	49.8	59.9	59.9	**69.8**
Math	GSM8K (EM)	8-shot	81.6	88.3	83.5	**89.3**
	MATH (EM)	4-shot	43.4	54.4	49.0	**61.6**
	MGSM (EM)	8-shot	63.6	76.2	69.9	**79.8**
	CMath (EM)	3-shot	78.7	84.5	77.3	**90.7**
Chinese	CLUEWSC (EM)	5-shot	82.0	82.5	**83.0**	**82.7**
	C-Eval (EM)	5-shot	81.4	89.2	72.5	**90.1**
	CMMLU (EM)	5-shot	84.0	**89.5**	73.7	88.8
	CMRC (EM)	1-shot	**77.4**	75.8	76.0	76.3
	C3 (EM)	0-shot	77.4	76.7	**79.7**	78.6
	CCPM (EM)	0-shot	**93.0**	88.5	78.6	92.0
Multilingual	MMMLU-non-English (EM)	5-shot	64.0	74.8	73.8	**79.4**

表 3 DeepSeek-V3-Base 与其他具有代表性的开源基模型的|比较。所有模型都在我们的内部框架中进行评估，并共享相同的评估设置。差距不超过 0.3 的分数被认为是在同一水平上得到的。DeepSeekV3-Base 在大多数基准测试上的性能最好，特别是在数学和代码任务上。

4.4.2.评价结果

在表 3 中，我们将 DeepSeek-V3 的基本模型与最先进的开源基本模型进行了比较，包括 DeepSeek-V2-Base（DeepSeek-AI，2024c）（我们之前的版本）、Qwen2.5 72B Base（Qwen，2024b）和 LLaMA-3.1 405B 基础（AI@Meta，2024b）。我们使用我们的内部评估框架来评估所有这些模型，并确保它们共享相同的评估设置。请注意，由于我们的评估框架在过去几个月中发生的变化，DeepSeek-V2-Base 的性能与我们之前报告的结果略有不同。总的来说，DeepSeek-V3-Base 综合优于 DeepSeek-V2-Base 和 Qwen2.5 72B，在大多数基准上超过 LLaMA-3.1 405B，基本上成为最强的开源模型。

从更详细的角度来看，我们将 DeepSeek-V3-Base 与其他开源基础模型进行比较。(1)与 DeepSeek-V2-Base 相比，由于我们的模型架构的改进、模型大小和训练令牌的扩展，以及数据质量的提高，DeepSeek-V3-Base 实现了预期的显著更好的性能。(2)与最先进的中国开源模型 Qse 相比，只有一半的激活参数，DeepSeek-V3-Base 也显示了显著的优势，特别是在英语、多语言、代码和数学基准方面。在中国的基准测试方面，除了中国的多学科多项选择任务 CMMLU 外，DeepSeek-V3-Base 的表现也优于 Qwen2.5 72B。(3)与 LLaMA-3.1 405B Base 相比，是最大的开源模型，具有 11 倍的激活参数，DeepSeek-V3-Base 在多语言、代码和数学基准上也表现出更好的性能。至于英语和中文基准，DeepSeek-V3-Base 表现出具有竞争力或更好的性能，在 BBH、MMLU 系列、DROP、C-Eval、CMMLU 和 CMMLU 系列上尤其出色。

由于我们高效的架构和全面的工程优化，DeepSeekV3 实现了极高的训练效率。在我们的训练框架和基础设施下，在每万亿代币上训练 DeepSeek-V3 只需要 180K H800 GPU 小时，这比训练 72B 或 405B 密集模型要便宜得多。

Benchmark (Metric)	# Shots	Small MoE Baseline	Small MoE w/ MTP	Large MoE Baseline	Large MoE w/ MTP
# Activated Params (Inference)	-	2.4B	2.4B	20.9B	20.9B
# Total Params (Inference)	-	15.7B	15.7B	228.7B	228.7B
# Training Tokens	-	1.33T	1.33T	540B	540B
Pile-test (BPB)	-	**0.729**	**0.729**	0.658	**0.657**
BBH (EM)	3-shot	39.0	**41.4**	70.0	**70.7**
MMLU (EM)	5-shot	50.0	**53.3**	**67.5**	66.6
DROP (F1)	1-shot	39.2	**41.3**	68.5	**70.6**
TriviaQA (EM)	5-shot	56.9	**57.7**	**67.0**	**67.3**
NaturalQuestions (EM)	5-shot	**22.7**	22.3	27.2	**28.5**
HumanEval (Pass@1)	0-shot	20.7	**26.8**	44.5	**53.7**
MBPP (Pass@1)	3-shot	35.8	**36.8**	61.6	**62.2**
GSM8K (EM)	8-shot	25.4	**31.4**	72.3	**74.0**
MATH (EM)	4-shot	10.7	**12.6**	38.6	**39.8**

表 4 MTP 策略的消融结果。MTP 策略在大多数评估基准上持续地提高了模型的性能。

4.5.讨论

4.5.1.多令牌预测的消融研究

在表 4 中，我们显示了 MTP 策略的消融结果。具体来说，我们在不同尺度上的两个基线模型之上验证了 MTP 策略。在小尺度下，我们训练了一个在 1.33T 令牌上包含 15.7B 总参数的基线 MoE 模型。在大规模上，我们在 540B 代币上训练一个包含 228.7B 总参数的基线 MoE 模型。在此基础上，为了保持训练数据和其他的体系结构不变，我们在它们上添加了一个 1 深度的 MTP 模块，并使用 MTP 策略训练两个模型进行比较。注意，在推理过程中，我们直接丢弃了 MTP 模块，因此比较模型的推理成本完全相同。从表中，我们可以观察到 MTP 策略在大多数评估基准上持续提高了模型性能。

Benchmark (Metric)	# Shots	Small MoE Aux-Loss-Based	Small MoE Aux-Loss-Free	Large MoE Aux-Loss-Based	Large MoE Aux-Loss-Free
# Activated Params	-	2.4B	2.4B	20.9B	20.9B
# Total Params	-	15.7B	15.7B	228.7B	228.7B
# Training Tokens	-	1.33T	1.33T	578B	578B
Pile-test (BPB)	-	0.727	**0.724**	0.656	**0.652**
BBH (EM)	3-shot	37.3	**39.3**	66.7	**67.9**
MMLU (EM)	5-shot	51.0	**51.8**	**68.3**	67.2
DROP (F1)	1-shot	38.1	**39.0**	**67.1**	**67.1**
TriviaQA (EM)	5-shot	**58.3**	**58.5**	66.7	**67.7**
NaturalQuestions (EM)	5-shot	**23.2**	**23.4**	27.1	**28.1**
HumanEval (Pass@1)	0-shot	22.0	**22.6**	40.2	**46.3**
MBPP (Pass@1)	3-shot	**36.6**	35.8	59.2	**61.2**
GSM8K (EM)	8-shot	27.1	**29.6**	70.7	**74.5**
MATH (EM)	4-shot	**10.9**	**11.1**	37.2	**39.6**

表 5 辅助无损耗平衡策略的消融结果。与纯基于辅助损失的方法相比，无辅助损失策略在大部分评估基准上始终能获得更好的模型性能。

4.5.2.辅助无损耗平衡策略的消融研究

在表 5 中，我们展示了辅助无损失平衡策略的消融结果。我们在不同尺度上的两个基线模型上验证了这一策略。在小尺度下，我们训练了一个在 1.33T 令牌上包含 15.7B 总参数的基线 MoE 模型。在大规模上，我们在 578B 代币上训练一个包含 228.7B 总参数的基线 MoE 模型。这两个基线模型都纯粹使用辅助损失来促进负载平衡，并使用具有 top-K 亲和归一化的 s 型门控函数。它们控制辅助损耗强度的超参数分别与 DeepSeek-V2-Lite 和 DeepSeek-V2 相同。在这两个基线模型的基础上，为了保持训练数据和其他的体系结构不变，我们消除了所有的辅助损失，并引入了无辅助损失的平衡策略进行比较。从表中，我们可以观察到，无辅助损失策略在大多数评估基准上始终获得更好的模型性能。

4.5.3.批量负载平衡与顺序明智负载平衡

无辅助损失平衡和顺序级辅助损失之间的关键区别在于它们的平衡范围：批处理平衡和顺序级平衡。与顺序辅助损失相比，批处理平衡施加了更灵活的约束，因为它不强制每个序列强制域内平衡。这种灵活性使专家能够更好地专攻不同的领域。为了验证这一点，我们记录并分析了桩测试集中不同领域的 16B 辅助损耗基线和 16B 无辅助损耗模型的专家负载。如图 9 所示，我们观察到无辅助损失模型如预期的那样显示了更大的专家专业化模式。

为了进一步研究这种灵活性和模型性能优势之间的相关性，我们另外设计并验证了一个批级辅助损失，该损失鼓励在每个训练批上而不是在每个序列上的负载平衡。实验结果表明，当达到相似的批量负载平衡水平时，批辅助损耗也可以达到与无辅助损耗方法相似的模型性能。具体来说，在我们使用 1B MoE 模型的实验中，验证损失分别为：2.258（使用顺序辅助损失）、2.253（使用无辅助损失方法）和 2.253(使用批处理辅助损耗）。我们还在 3B MoE 模型上观察到类似的结果：使用顺序辅助损失的模型获得了 2.085 的验证损失，而使用无辅助损失方法或批量辅助损失的模型获得了相同的验证损失 2.080。

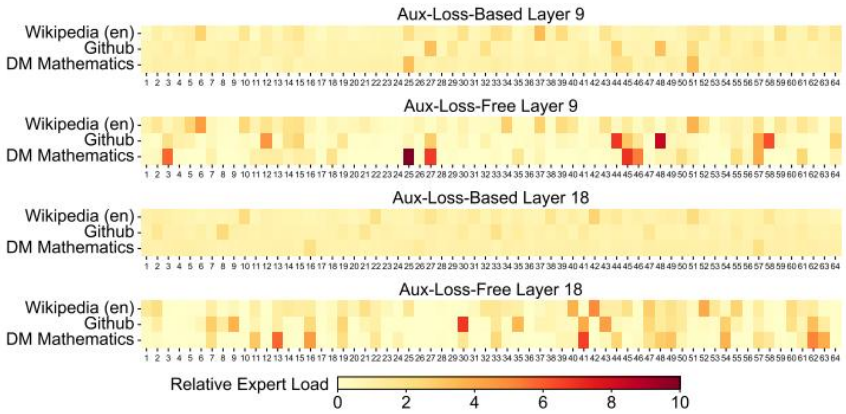


图 9 在桩测试集中的三个领域上的无辅助损耗和基于辅助损耗的模型的专家负载。无辅助损失模型比基于辅助损失的模型显示出更大的专家专业化模式。相对专家负荷是指实际专家负荷与理论上平衡的专家负荷之间的比值。由于空间限制，我们只以两层的结果为例，所有层的结果见附录 C。

此外，尽管批处理负载平衡方法显示出一致的性能优势，但它们在效率方面也面临着两个潜在的挑战： (1)特定序列或小批次内的负载不平衡，(2)推理过程中域移导致的负载不平衡。第一个挑战自然是由我们的训练框架来解决的，该框架使用了大规模的专家并行和数据并行，这保证了每个微批量的大规模。对于第二个挑战，我们还设计并实现了一个具有冗余专家部署的高效推理框架，如第 3.4 节所述，以克服它。

5.后训练

5.1.监督微调

我们管理指令调优数据集，以包含跨越多个域的 150 万个实例，每个域根据其特定需求定制不同的数据创建方法。

推理数据。对于与推理相关的数据集，包括那些关注于数学、代码竞争问题和逻辑难题的数据集，我们通过利用内部的 DeepSeek-R1 模型来生成数据。具体来说，虽然 DeepSeek-R1 生成的数据显示出了很强的准确性，但它也存在诸如过度思考、格式化不良和长度过长等问题。我们的目标是平衡 DeepSeek-R1 生成的推理数据的高精度和定期格式化的推理数据的清晰度和简洁性。

为了建立我们的方法，我们首先开发一个针对特定领域定制的专家模型，如代码、数学或一般推理，使用组合监督微调（SFT）和强化学习（RL）训练管道。这个专家模型可以作为最终模型的数据生成器。训练过程包括生成两种不同类型的监督微调（SFT）样本为每个实例：第一对问题的格式<问题，原始响应>，而第二个包含一个系统提示与问题和 R1 响应<系统提示的格式，问题，R1 响应>。

系统提示符经过精心设计，以包括指导模型产生富含反射和验证机制的响应的指令。在强化学习（RL）阶段，该模型利用高温采样来生成响应，从而集成了从 R1 生成的和原始数据中获得的模式，即使在没有明确的系统提示的情况下。经过数百个强化学习（RL）步骤之后，中间强化学习（RL）模型学习合并 R1 模式，从而战略性地提高整体性能。

在完成 RL 训练阶段后，我们实施拒绝抽样，为最终的模型管理高质量的监督微调（SFT）数据，其中专家模型被用作数据生成源。这种方法确保了最终的训练数据保留了 DeepSeek-R1 的优势，同时产生了简洁和有效的响应。

非推理数据。对于非推理的数据，如创造性写作、角色扮演和简单的问题回答，我们使用

DeepSeek-V2.5 来生成响应，并招募人工注释者来验证数据的准确性和正确性。

监督微调（SFT）设置。我们使用监督微调（SFT）数据集对两个时代进行了微调的 DeepSeek-V3-Base，使用余弦衰减学习速率调度，从 5×10−6 开始，并逐渐减少到 1×10−6。在训练过程中，每个单一的序列都从多个样本中打包。然而，我们采用了一个样本掩蔽策略，以确保这些例子保持孤立和相互不可见。

5.2.强化学习

5.2.1.奖励模式

我们在强化学习（RL）过程中使用了一个基于规则的奖励模型（RM）和一个基于模型的奖励模型（RM）。

基于规则的奖励模型（RM）。对于可以使用特定规则进行验证的问题，我们采用基于规则的奖励系统来确定反馈。例如，某些数学问题有确定性的结果，我们要求模型在指定的格式内（例如，在一个方框中）提供最终的答案，允许我们应用规则来验证正确性。类似地，对于 LeetCode 问题，我们可以使用编译器来基于测试用例生成反馈。通过尽可能地利用基于规则的验证，我们确保了更高级别的可靠性，因为这种方法可以抵抗操作或利用。

基于模型的奖励模型（RM）。对于具有自由形式的地面真相答案的问题，我们依赖于奖励模型来确定回答是否与预期的地面真相相符。相反，对于没有明确理由的问题，比如那些涉及创意写作的问题，奖励模型的任务是基于问题和相应的答案作为输入。奖励模型是从 DeepSeek-V3 SFT 检查点训练出来的。为了提高其可靠性，我们构建了偏好数据，不仅提供最终的奖励，还包括导致奖励的思维链。这种方法有助于减轻在特定任务中进行奖励性黑客攻击的风险。

5.2.2.组相对策略优化

与 DeepSeek-V2（DeepSeek-AI，2024c）类似，我们采用了组相对策略优化（GRPO）（Shao et al.，2024），该模型放弃了通常与策略模型具有相同大小的批评模型，而是从组分数中估计基线。具体来说，对于每个问题 ，GRPO 从旧的策略模型 中抽取一组输出{ 1， 2，…， }，然后通过最大化以下目标来优化策略模型 ：

$$\mathcal{J}_{GRPO}(\theta) = \mathbb{E}[q \sim P(Q), \{o_i\}_{i=1}^{G} \sim \pi_{\theta_{old}}(O|q)]$$
$$\frac{1}{G}\sum_{i=1}^{G}\left(\min\left(\frac{\pi_\theta(o_i|q)}{\pi_{\theta_{old}}(o_i|q)}A_i, \text{clip}\left(\frac{\pi_\theta(o_i|q)}{\pi_{\theta_{old}}(o_i|q)}, 1-\varepsilon, 1+\varepsilon\right)A_i\right) - \beta\mathbb{D}_{KL}\left(\pi_\theta||\pi_{ref}\right)\right), \quad (26)$$

$$\mathbb{D}_{KL}\left(\pi_\theta||\pi_{ref}\right) = \frac{\pi_{ref}(o_i|q)}{\pi_\theta(o_i|q)} - \log\frac{\pi_{ref}(o_i|q)}{\pi_\theta(o_i|q)} - 1, \quad (27)$$

其中 和 为超参数； 为参考模型； 是优势，来自于每组内输出对应的奖励{ 1， 2，…， }：

$$A_i = \frac{r_i - \text{mean}(\{r_1, r_2, \cdots, r_G\})}{\text{std}(\{r_1, r_2, \cdots, r_G\})}. \quad (28)$$

在强化学习（RL）过程中，我们整合了来自不同领域的提示，如编码、数学、写作、角色扮演和问题回答。这种方法不仅使模型与人类的偏好更加一致，而且还提高了基准测试中的性能，特别是在可用的监督微调（SFT）数据有限的情况下。

5.3.评估

5.3.1.评估设置

评价基准。除了基准我们用于基础模型测试，我们进一步评估指导模型 IFEval (Zhou et al., 2023), FRAMES (Krishna et al.,2024), LongBench v2 (Bai et al., 2024), GPQA (Rein et al., 2023), SimpleQA (OpenAI, 2024c), C SimpleQA (He et al., 2024), SWE-Bench Verified (OpenAI, 2024d), Aider 1, LiveCodeBench (Jainet al., 2024) (questions from August 2024 to November 2024)，世界级编程竞赛，中国国家高中数学奥林匹克（CNMO 2024），美国数学竞赛（AIME 2024）（MAA，2024）。

比较基线。我们根据几个强基线对我们的聊天模型进行全面评估，包括 DeepSeeke-V2-0506、DeepSeek-V2.5-0905、Qwen2.5 72B 指令、LLaMA-3.1 405B 指令、Claude-Sonnet-3.5-1022 和 GPT-4o-0513。对于 DeepSeek-V2 模型系列，我们选择了最具代表性的变体进行比较。对于闭源模型，评估是通过它们各自的 API 来执行的。

详细的评估配置。对于包括 MMLU、DROP、GPQA 和 SimpleQA 在内的标准基准测试，我们采用了来自简单 simple-evals 框架的评估提示。我们在零镜头设置下对 MMLU-Redux 使用零 eval 提示格式（Lin，2024）。对于其他数据集，我们按照数据集创建者提供的默认提示来遵循它们的原始评估协议。对于代码和数学基准测试，HumanEval-Mul 数据集总共包括 8 种主流编程语言（Python、Java、Cpp、C#、JavaScript、TypeScript、PHP 和 Bash）。我们使用思维链 CoT 和非思维链 CoT 方法来评估 LiveCodeBench 上的模型性能，其中的数据收集于 2024 年 8 月至 2024 年 11 月。编码力数据集是用竞争对手的百分比来衡量的。到 2024 年 11 月。编码力数据集是用竞争对手的百分比来衡量的。验证的 SWE-Bench 使用无代理框架进行评估（Xia et al.，2024）。我们使用“差异”格式来评估与助手相关的基准测试。对于数学评估，AIME 和 CNMO 2024 以温度为 0.7 进行评估，结果平均超过 16 次运行，而 MATH-500 采用贪婪解码。我们允许所有模型为每个基准测试输出最多 8192 个令牌。

	Benchmark (Metric)	DeepSeek V2-0506	DeepSeek V2.5-0905	Qwen2.5 72B-Inst.	LLaMA-3.1 405B-Inst.	Claude-3.5-Sonnet-1022	GPT-4o 0513	DeepSeek V3
	Architecture	MoE	MoE	Dense	Dense	-	-	MoE
	# Activated Params	21B	21B	72B	405B	-	-	37B
	# Total Params	236B	236B	72B	405B	-	-	671B
English	MMLU (EM)	78.2	80.6	85.3	**88.6**	**88.3**	87.2	**88.5**
	MMLU-Redux (EM)	77.9	80.3	85.6	86.2	**88.9**	88.0	**89.1**
	MMLU-Pro (EM)	58.5	66.2	71.6	73.3	**78.0**	72.6	75.9
	DROP (3-shot F1)	83.0	87.8	76.7	88.7	88.3	83.7	**91.6**
	IF-Eval (Prompt Strict)	57.7	80.6	84.1	86.0	**86.5**	84.3	86.1
	GPQA-Diamond (Pass@1)	35.3	41.3	49.0	51.1	**65.0**	49.9	59.1
	SimpleQA (Correct)	9.0	10.2	9.1	17.1	28.4	**38.2**	24.9
	FRAMES (Acc.)	66.9	65.4	69.8	70.0	72.5	**80.5**	73.3
	LongBench v2 (Acc.)	31.6	35.4	39.4	36.1	41.0	48.1	**48.7**
Code	HumanEval-Mul (Pass@1)	69.3	77.4	77.3	77.2	81.7	80.5	**82.6**
	LiveCodeBench (Pass@1-COT)	18.8	29.2	31.1	28.4	36.3	33.4	**40.5**
	LiveCodeBench (Pass@1)	20.3	28.4	28.7	30.1	32.8	34.2	**37.6**
	Codeforces (Percentile)	17.5	35.6	24.8	25.3	20.3	23.6	**51.6**
	SWE Verified (Resolved)	-	22.6	23.8	24.5	**50.8**	38.8	42.0
	Aider-Edit (Acc.)	60.3	71.6	65.4	63.9	**84.2**	72.9	79.7
	Aider-Polyglot (Acc.)	-	18.2	7.6	5.8	45.3	16.0	**49.6**
Math	AIME 2024 (Pass@1)	4.6	16.7	23.3	23.3	16.0	9.3	**39.2**
	MATH-500 (EM)	56.3	74.7	80.0	73.8	78.3	74.6	**90.2**
	CNMO 2024 (Pass@1)	2.8	10.8	15.9	6.8	13.1	10.8	**43.2**
Chinese	CLUEWSC (EM)	89.9	90.4	**91.4**	84.7	85.4	87.9	90.9
	C-Eval (EM)	78.6	79.5	86.1	61.5	76.7	76.0	**86.5**
	C-SimpleQA (Correct)	48.5	54.1	48.4	50.4	51.3	59.3	**64.8**

表 6 DeepSeek-V3 与其他代表性聊天模型的比较。所有模型都在限制输出长度为 8K 的配置中进行评估。包含少于 1000 个样本的基准测试使用不同的温度设置进行多次测试，以获得稳健的最终结果。DeepSeek-V3 是性能最好的开源模型，并且与前沿的闭源模型相比

也表现出了具有竞争力的性能。

5.3.2.标准评价

表 6 给出了评估结果，显示了 DeepSeek-V3 是性能最好的开源模型。此外，它还可以与 GPT-4o 和 Claude-3.5-Sonnet 等前沿闭源模型进行竞争。

英语基准。MMLU 是一个被广泛认可的基准测试，设计用于评估跨不同知识领域和任务的大型语言模型的性能。DeepSeek-V3 表现出具有竞争力的性能，与 LLaMA- 3.1-405B、GPT-4o 和 Claude-Sonnet 3.5 等顶级模型相当，同时显著优于 Qwen2.5 72B。此外，DeepSeek-V3 在 MMLU-Pro 方面表现出色，这是一个更具挑战性的教育知识基准，它仅后于 Claude-Sonnet 3.5。在 MMLU-Redux 上，一个改进版的 MMLU 与修正后的标签，DeepSeek-V3 超过了它的同行。此外，在理工科博士生测试（GPQA-Dimond）上，DeepSeSeek-V3 取得了显著的成绩，排名落后于 Claude-Sonnet 3.5，并以显著优势优于所有其他竞争对手。

在长上下文理解的基准测试中，如 DROP、LongBench v2 和框架，DeepSeek-V3 继续展示其作为顶级模型的地位。它在 DROP 的 3 个镜头设置中获得了令人印象深刻的 91.6 F1 分数，超过了这个类别中的所有其他模型。在框架上，一个需要问答超过 100k 标记上下文的基准，DeepSeekV3 仅后于 GPT-4o，同时显著优于所有其他模型。这证明了 DeepSeek-V3 在处理极长时间上下文任务方面的强大能力。DeepSeek-V3 的长上下文性能被其在 LongBench v2 上的最佳类性能进一步验证，这是一个在 DeepSeek V3 发布前几周发布的数据集。在事实知识基准上，SimpleQA，DeepSeek-V3 落后于 GPT-4o 和 Claude-Sonnet 3.5，主要是因为其设计重点和资源分配。DeepSeek-V3 分配了更多的训练令牌来学习中文知识，这导致了在 C-SimpleQA 上的卓越表现。在遵循指令的基准测试中，DeepSeek-V3 的性能显著优于它的前身 DeepSeek-V2 系列，突出了其理解和坚持用户定义的格式约束的改进能力。

代码和数学基准。对于 LLM 来说，编码是一项具有挑战性和实际意义的任务，包括以工程为重点的任务，如软件工作台验证和辅助程序，以及算法任务，如人类环境和现场编码台。在工程任务中，DeepSeek-V3 落后于 Claude-Sonnet-3.5-1022，但其性能明显优于开源模型。开源的 DeepSeek-V3 有望促进编码相关工程任务的进步。通过提供对其强大功能的访问，DeepSeek-V3 可以推动软件工程和算法开发等领域的创新和改进，授权开发人员和研究人员推动开源模型在编码任务中可以实现的边界。在算法任务中，DeepSeek-V3 表现出了优越的性能，在 HumanEval-Mul 和直播台等基准上优于所有基线。这一成功可以归因于其先进的知识蒸馏技术，它有效地提高了其代码生成和在算法集中的任务中解决问题的能力。

在数学基准测试上，DeepSeek-V3 展示了卓越的性能，显著超过了基线，并为非 o1 类模型设置了一个新的最先进的状态。具体来说，在 AIME、MATH-500 和 CNMO 2024 上，DeepSeek-V3 的绝对分数优于 Qwen2.5 72B 约 10%，这对于这些具有挑战性的基准是一个相当大的差距。这种显著的能力突出了来自 DeepSeek-R1 的蒸馏技术的有效性，该技术已被证明对非类似 o1 的模型非常有益。

Model	Arena-Hard	AlpacaEval 2.0
DeepSeek-V2.5-0905	76.2	50.5
Qwen2.5-72B-Instruct	81.2	49.1
LLaMA-3.1 405B	69.3	40.5
GPT-4o-0513	80.4	51.1
Claude-Sonnet-3.5-1022	85.2	52.0
DeepSeek-V3	**85.5**	**70.0**

表 7 英语开放式对话评估。对于太空空间 2.0，我们使用长度控制的胜率作为度量。

中文基准。Qwen 和 DeepSeek 是两个具有代表性的模型系列，对中文和英语都有强大的支持。在事实基准中文简单体 QA 中，DeepSeekV3 超过 Qwen2.5-72B 16.4 分，尽管 Qwen2.5 在一个更大的 18T 标记上接受训练，比 DeepSeek-V3 预先训练的 14.8T 标记多出 20%。

对于中国教育知识评价的代表性基准 C-Eval 和线索 (Chinese Winograd Schema Challenge),，DeepSeek-V3 和 Qwen2.5-72B 表现出相似的表现水平，表明这两种模型在挑战汉语推理和教育任务方面都进行了很好的优化。

5.3.3.开放的评估

除了标准的基准测试之外，我们还使用 LLM 作为法官来评估我们在开放式生成任务上的模型，结果如表 7 所示。具体来说，我们坚持 AlpacaEval 2.0 (Dubois et al., 2024)和 Arena-Hard (Li et al.,2024a)的原始配置，它们利用 GPT-4-Turbo-1106 作为两两比较的法官。在 Arena-Hard，DeepSeseek-V3 与基线 GPT-4-0314 的胜率超过了 86%，表现可与 Claude-Sonnet-3.5-1022 等顶级机型媲美。这强调了 DeepSeek-V3 的强大能力，特别是在处理复杂的提示方面，包括编码和调试任务。此外，DeepSeek-V3 实现了一个突破性的里程碑，作为第一个在 Arena-Hard 基准测试上超过 85%的开源模型。这一成就显著地弥补了开源模型和封闭源码模型之间的性能差距，为开源模型在具有挑战性的领域中可以完成的工作设置了一个新的标准。

类似地，DeepSeek-V3 在 AlpacaEval 2.0 上展示了卓越的性能，性能优于闭源码和开源模型。这证明了它在编写任务和处理直接的问答场景方面非常熟练。值得注意的是，它比 DeepSeek-V2.5-0905 高了 20%，突出了在处理简单任务方面的实质性改进，并展示了其进步的有效性。

5.3.4.DeepSeek-V3 作为一个生成性奖励模型

我们比较了 DeepSeek-V3 与最先进的模型，即 GPT-4o 和 Claude-3.5 的判断能力。表 8 显示了这些模型在 RewardBench（Lambert et al.，2024）中的性能。DeepSeek-V3 的性能可以与 GPT-4o-0806 和 Claude-3.5-Sonnet-1022 的最佳版本相媲美，同时也超过了其他版本。此外，投票技术还可以增强 DeepDeepSeek-V3 的判断能力。因此，我们采用 DeepSeekV3 和投票，对开放式问题提供自我反馈，从而改进对齐过程的有效性和鲁棒性。

Model	Chat	Chat-Hard	Safety	Reasoning	Average
GPT-4o-0513	96.6	70.4	86.7	84.9	84.7
GPT-4o-0806	96.1	76.1	88.1	86.6	86.7
GPT-4o-1120	95.8	71.3	86.2	85.2	84.6
Claude-3.5-sonnet-0620	96.4	74.0	81.6	84.7	84.2
Claude-3.5-sonnet-1022	96.4	79.7	91.1	87.6	88.7
DeepSeek-V3	96.9	79.8	87.0	84.3	87.0
DeepSeek-V3 (maj@6)	96.9	82.6	89.5	89.2	89.6

表 8 GPT-4o、Claude-3.5-sonnet 和 DeepSeek-V3 在备用工作台上的|性能。

Model	LiveCodeBench-CoT		MATH-500	
	Pass@1	Length	Pass@1	Length
DeepSeek-V2.5 Baseline	31.1	718	74.6	769
DeepSeek-V2.5 +R1 Distill	37.4	783	83.2	1510

表 9 从 DeepSeek-R1 中提取蒸馏的贡献。LiveCodeBench-CoT 和 MATH-500 的评价设置与表 6 相同。

5.4.讨论

5.4.1.从 DeepSeek-R1 中提取的蒸馏液

我们消除了基于 DeepSeek-V2.5 的 DeepSeek-R1 的蒸馏贡献。基线是在较短的思维链 CoT 数据上进行训练的，而其竞争对手则使用由上述专家检查点生成的数据。

表 9 展示了蒸馏数据的有效性，显示了 LiveCodeBench 和 MATH-500 基准测试的显著改进。我们的实验揭示了一个有趣的权衡：蒸馏导致更好的性能，但也大大增加了平均响应长度。为了保持模型精度和计算效率之间的平衡，我们仔细选择了蒸馏中 DeepSeek-V3 的最佳设置。

我们的研究表明，从推理模型中提取知识为训练后的优化提供了一个很有前途的方向。虽然我们目前的工作集中于从数学和编码领域中提取数据，但这种方法显示了在不同任务领域中更广泛应用的潜力。在这些特定领域的有效性表明，长思维链 CoT 蒸馏对提高其他需要复杂推理的认知任务中的模型表现有价值。在不同的领域中进一步探索这种方法仍然是未来研究的一个重要方向。

5.4.2.自我奖励

奖励在强化学习（RL）中起着关键的作用，指导着优化过程。在通过外部工具进行验证很简单的领域中，例如一些编码或数学场景，强化学习（RL）显示出了非凡的有效性。然而，在更一般的场景下，通过硬编码构建一个反馈机制是不切实际的。在 DeepSeek-V3 的开发过程中，对于这些更广泛的背景，我们采用了宪法规定的人工智能方法（Bai et al.，2022），利用 DeepSeek-V3 本身的投票评估结果作为反馈源。该方法产生了显著的对齐效果，显著提高了 DeepSeek-V3 在主观评价中的性能。通过整合额外的宪法输入，DeepSeek-V3 可以向宪法的方向进行优化。我们认为，这种结合了补充信息和 LLM 作为反馈源的范式是至关重要的。LLM 作为一个多功能处理器，能够将非结构化信息从不同的场景转换为奖励，最终促进 LLM 的自我改进。除了自我奖励之外，我们还致力于发现其他通用的和可扩展的奖励方法，以在一般场景中持续地推进模型的能力。

5.4.3.多令牌预测评估

DeepSeek-V3 不是只预测下一个单一的令牌，而是通过 MTP 技术预测下一个 2 个令牌。结合推测解码框架(Leviathan et al.,2023; Xia et al., 2023)，可以显著加快模型的解码速度。一个自然的问题是关于额外预测令牌的接受率。根据我们的评估，在不同的生成主题中，第二代令牌预测的接受率在 85%到 90%之间，显示了一致的可靠性。这种高接受率使 DeepSeek-V3 能够实现一个显著提高的解码速度，提供 1.8 倍的 TPS（每秒的令牌）。

6.结论、局限性和未来的发展方向

在本文中，我们引入了 DeepSeek-V3，一个大型的 MoE 语言模型，具有 671B 总参数和 37B 激活参数，在 14.8T 标记上进行训练。除了 MLA 和 DeepSeekMoE 架构之外，它还开创了一种无辅助损失的负载平衡策略，并设置了一个多令牌预测训练目标，以获得更强的性能。由于 FP8 训练的支持和细致的工程优化，DeepSeek-V3 的训练具有成本效益。训练后也成功地从 DeepSeek-R1 系列模型中提取了推理能力。综合评估表明，DeepSeek-V3

已经成为目前可用的最强大的开源模型，并实现了可与 GPT-4o 和 Claude-3.5-Sonnet 等领先的闭源码模型相媲美的性能。尽管它表现强劲，但它也保持了经济的训练成本。它只需要 2.788M H800 GPU 小时就可以完成全面的训练，包括训练前、环境长度扩展和训练后。

在承认其强大的性能和成本效益的同时，我们也认识到 DeepSeek-V3 有一些局限性，特别是在部署方面。首先，为了确保有效的推断，DeepSeek-V3 的推荐部署单元相对较大，这可能会给小型团队带来负担。其次，尽管我们对 DeepSeekV3 的部署策略已经实现了比 DeepSeek-V2 的 2 倍以上的端到端生成速度，但仍有进一步增强的潜力。幸运的是，随着更先进的硬件的开发，这些限制有望很自然地得到解决。

DeepSeek 一贯坚持具有长期主义的开源模型的路径，目标是稳步接近 AGI（人工通用智能）的最终目标。在未来，我们计划对以下方向的研究进行战略性投资。

我们将不断地研究和改进我们的模型架构，旨在进一步提高训练和推理效率，努力获得对无限上下文长度的有效支持。此外，我们将尝试突破变压器的体系结构限制，从而突破其建模能力的边界。

我们将不断迭代我们的训练数据的数量和质量，并探索额外的训练信号源的合并，旨在推动数据在更全面的维度范围内的扩展。

我们将不断地探索和迭代我们的模型的深度思维能力，旨在通过扩大他们的推理的长度和深度来提高他们的智力和解决问题的能力。

我们将探索更全面和多维的模型评估方法，以防止在研究过程中优化一组固定的基准的趋势，这可能会产生对模型能力的误导性印象，并影响我们的基础评估。

参考

AI@Meta. Llama 3 model card, 2024a. URL https://github.com/meta-llama/llama3/blob/main/MODEL_CARD.md.

AI@Meta. Llama 3.1 model card, 2024b. URL https://github.com/meta-llama/llama-models/blob/main/models/llama3_1/MODEL_CARD.md.

Anthropic. Claude 3.5 sonnet, 2024. URL https://www.anthropic.com/news/claude-3-5-sonnet.

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.

Y. Bai, S. Kadavath, S. Kundu, A. Askell, J. Kernion, A. Jones, A. Chen, A. Goldie, A. Mirhoseini,C. McKinnon, et al. Constitutional AI: Harmlessness from AI feedback. arXiv preprintarXiv:2212.08073, 2022.

Y. Bai, S. Tu, J. Zhang, H. Peng, X. Wang, X. Lv, S. Cao, J. Xu, L. Hou, Y. Dong, J. Tang, and J. Li. LongBench v2: Towards deeper understanding and reasoning on realistic long-context multitasks. arXiv preprint arXiv:2412.15204, 2024.

M. Bauer, S. Treichler, and A. Aiken. Singe: leveraging warp specialization for high performance on GPUs. In Proceedings of the 19th ACM SIGPLAN Symposium on Principles and Practice of Parallel Programming, PPoPP '14, page 119–130, New York, NY, USA, 2014. Association for Computing Machinery. ISBN 9781450326568. doi: 10.1145/2555243.2555258. URL https://doi.org/10.1145/2555243.2555258.

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.

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.

Y. Cui, T. Liu, W. Che, L. Xiao, Z. Chen, W. Ma, S. Wang, and G. Hu. A span-extraction dataset for Chinese machine reading comprehension. In K. Inui, J. Jiang, V. Ng, and X. Wan, editors, Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), pages 5883–5889, Hong Kong, China, Nov. 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-1600. URL https://aclanthology.org/D19-1600.

D. Dai, C. Deng, C. Zhao, R. X. Xu, H. Gao, D. Chen, J. Li, W. Zeng, X. Yu, Y. Wu, Z. Xie, Y. K. Li, P. Huang, F. Luo, C. Ruan, Z. Sui, and W. Liang. Deepseekmoe: Towards ultimate expert specialization in mixture-of-experts language models. CoRR, abs/2401.06066, 2024. URL https://doi.org/10.48550/arXiv.2401.06066.

DeepSeek-AI. Deepseek-coder-v2: Breaking the barrier of closed-source models in code intelligence. CoRR, abs/2406.11931, 2024a. URL https://doi.org/10.48550/arXiv.2406.11931.

DeepSeek-AI. Deepseek LLM: scaling open-source language models with longtermism. CoRR,abs/2401.02954, 2024b. URL https://doi.org/10.48550/arXiv.2401.02954.

DeepSeek-AI. Deepseek-v2: A strong, economical, and efficient mixture-of-experts language model. CoRR, abs/2405.04434, 2024c. URL https://doi.org/10.48550/arXiv.2405.04434.

T. Dettmers, M. Lewis, Y. Belkada, and L. Zettlemoyer. Gpt3. int8 (): 8-bit matrix multiplication for transformers at scale. Advances in Neural Information Processing Systems, 35:30318–30332, 2022.

H. Ding, Z. Wang, G. Paolini, V. Kumar, A. Deoras, D. Roth, and S. Soatto. Fewer

truncations improve language modeling. arXiv preprint arXiv:2404.10830, 2024.
D. Dua, Y. Wang, P. Dasigi, G. Stanovsky, S. Singh, and M. Gardner. DROP: A reading compre hension 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.
Y. Dubois, B. Galambosi, P. Liang, and T. B. Hashimoto. Length-controlled alpacaeval: A simple way to debias automatic evaluators. arXiv preprint arXiv:2404.04475, 2024.
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.
M. Fishman, B. Chmiel, R. Banner, and D. Soudry. Scaling FP8 training to trillion-token llms. arXiv preprint arXiv:2409.12517, 2024.
E. Frantar, S. Ashkboos, T. Hoefler, and D. Alistarh. Gptq: Accurate post-training quantization for generative pre-trained transformers. arXiv preprint arXiv:2210.17323, 2022.
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.
A. P. Gema, J. O. J. Leang, G. Hong, A. Devoto, A. C. M. Mancino, R. Saxena, X. He, Y. Zhao, X. Du, M. R. G. Madani, C. Barale, R. McHardy, J. Harris, J. Kaddour, E. van Krieken, and P. Minervini. Are we done with mmlu? CoRR, abs/2406.04127, 2024. URL https://doi.org/10.48550/arXiv.2406.04127.
F. Gloeckle, B. Y. Idrissi, B. Rozière, D. Lopez-Paz, and G. Synnaeve. Better & faster large language models via multi-token prediction. In Forty-first International Conference on Machine Learning, ICML 2024, Vienna, Austria, July 21-27, 2024. OpenReview.net, 2024. URL https://openreview.net/forum?id=pEWAcejiU2.
Google. Our next-generation model: Gemini 1.5, 2024. URL https://blog.google/technology/ai/google-gemini-next-generation-model-february-2024.
R. L. Graham, D. Bureddy, P. Lui, H. Rosenstock, G. Shainer, G. Bloch, D. Goldenerg, M. Dubman,S. Kotchubievsky, V. Koushnir, et al. Scalable hierarchical aggregation protocol (SHArP): Ahardware architecture for efficient data reduction. In 2016 First International Workshop on Communication Optimizations in HPC (COMHPC), pages 1–10. IEEE, 2016.
A. Gu, B. Rozière, H. Leather, A. Solar-Lezama, G. Synnaeve, and S. I. Wang. Cruxeval: A benchmark for code reasoning, understanding and execution, 2024.
D. Guo, Q. Zhu, D. Yang, Z. Xie, K. Dong, W. Zhang, G. Chen, X. Bi, Y. Wu, Y. K. Li, F. Luo, Y. Xiong, and W. Liang. Deepseek-coder: When the large language model meets programming - the rise of code intelligence. CoRR, abs/2401.14196, 2024. URL https://doi.org/10.48550/arXiv.2401.14196.
A. Harlap, D. Narayanan, A. Phanishayee, V. Seshadri, N. Devanur, G. Ganger, and P. Gibbons. Pipedream: Fast and efficient pipeline parallel dnn training, 2018. URL

https://arxiv.org/abs/1806.03377.
B. He, L. Noci, D. Paliotta, I. Schlag, and T. Hofmann. Understanding and minimising out lier features in transformer training. In The Thirty-eighth Annual Conference on Neural Information Processing Systems.
Y. He, S. Li, J. Liu, Y. Tan, W. Wang, H. Huang, X. Bu, H. Guo, C. Hu, B. Zheng, et al. Chinese simpleqa: A chinese factuality evaluation for large language models. arXiv preprintarXiv:2411.07140, 2024.
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. arXiv preprint arXiv:2103.03874,2021.
Y. Huang, Y. Bai, Z. Zhu, J. Zhang, J. Zhang, T. Su, J. Liu, C. Lv, Y. Zhang, J. Lei, et al. C-Eval: Amulti-level multi-discipline chinese evaluation suite for foundation models. arXiv preprintarXiv:2305.08322, 2023.
N. Jain, K. Han, A. Gu, W. Li, F. Yan, T. Zhang, S. Wang, A. Solar-Lezama, K. Sen, and I. Stoica.Livecodebench: Holistic and contamination free evaluation of large language models for code.CoRR, abs/2403.07974, 2024. URL https://doi.org/10.48550/arXiv.2403.07974.
A. Q. Jiang, A. Sablayrolles, A. Mensch, C. Bamford, D. S. Chaplot, D. d. l. Casas, F. Bressand,G. Lengyel, G. Lample, L. Saulnier, et al. Mistral 7b. arXiv preprint arXiv:2310.06825, 2023.
M. Joshi, E. Choi, D. Weld, and L. Zettlemoyer. TriviaQA: A large scale distantly supervised challenge dataset for reading comprehension. In R. Barzilay and M.-Y. Kan, editors, Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: LongPapers), pages 1601–1611, Vancouver, Canada, July 2017. Association for Computational Linguistics. doi: 10.18653/v1/P17-1147. URL https://aclanthology.org/P17-1147.
D. Kalamkar, D. Mudigere, N. Mellempudi, D. Das, K. Banerjee, S. Avancha, D. T. Vooturi, N. Jammalamadaka, J. Huang, H. Yuen, et al. A study of bfloat16 for deep learning training.arXiv preprint arXiv:1905.12322, 2019.
S. Krishna, K. Krishna, A. Mohananey, S. Schwarcz, A. Stambler, S. Upadhyay, and M. Faruqui. Fact, fetch, and reason: A unified evaluation of retrieval-augmented generation. CoRR, abs/2409.12941, 2024. doi: 10.48550/ARXIV.2409.12941. URL https://doi.org/10.48550/arXiv.2409.12941.
T. Kwiatkowski, J. Palomaki, O. Redfield, M. Collins, A. P. Parikh, C. Alberti, D. Epstein,I. Polosukhin, J. Devlin, K. Lee, K. Toutanova, L. Jones, M. Kelcey, M. Chang, A. M. Dai, J. Uszkoreit, Q. Le, and S. Petrov. Natural questions: a benchmark for question answering research. Trans. Assoc. Comput. Linguistics, 7:452–466, 2019. doi: 10.1162/tacl_a_00276. URL https://doi.org/10.1162/tacl_a_00276.
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.

N. Lambert, V. Pyatkin, J. Morrison, L. Miranda, B. Y. Lin, K. Chandu, N. Dziri, S. Kumar, T. Zick, Y. Choi, et al. Rewardbench: Evaluating reward models for language modeling. arXiv preprint arXiv:2403.13787, 2024.

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.

Y. Leviathan, M. Kalman, and Y. Matias. Fast inference from transformers via speculative decoding. In International Conference on Machine Learning, ICML 2023, 23-29 July 2023, Honolulu, Hawaii, USA, volume 202 of Proceedings of Machine Learning Research, pages 19274–19286. PMLR, 2023. URL https://proceedings.mlr.press/v202/leviathan23a.html.

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.

S. Li and T. Hoefler. Chimera: efficiently training large-scale neural networks with bidirectional pipelines. In Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis, SC '21, page 1–14. ACM, Nov. 2021. doi: 10.1145/3458817.3476145. URL http://dx.doi.org/10.1145/3458817.3476145.

T. Li, W.-L. Chiang, E. Frick, L. Dunlap, T. Wu, B. Zhu, J. E. Gonzalez, and I. Stoica. From crowdsourced data to high-quality benchmarks: Arena-hard and benchbuilder pipeline. arXiv preprint arXiv:2406.11939, 2024a.

W. Li, F. Qi, M. Sun, X. Yi, and J. Zhang. Ccpm: A chinese classical poetry matching dataset,2021.

Y. Li, F. Wei, C. Zhang, and H. Zhang. EAGLE: speculative sampling requires rethinking feature uncertainty. In Forty-first International Conference on Machine Learning, ICML 2024,Vienna, Austria, July 21-27, 2024. OpenReview.net, 2024b. URL https://openreview.net/forum?id=1NdN7eXyb4.

B. Y. Lin. ZeroEval: A Unified Framework for Evaluating Language Models, July 2024. URL https://github.com/WildEval/ZeroEval.

I. Loshchilov and F. Hutter. Decoupled weight decay regularization. arXiv preprintarXiv:1711.05101, 2017.

S. Lundberg. The art of prompt design: Prompt boundaries and token healing, 2023. URL https://towardsdatascience.com/the-art-of-prompt-design-prompt-boundaries-and-token-healing-3b2448b0be38.

Y. Luo, Z. Zhang, R. Wu, H. Liu, Y. Jin, K. Zheng, M. Wang, Z. He, G. Hu, L. Chen, et al. Ascend HiFloat8 format for deep learning. arXiv preprint arXiv:2409.16626, 2024.

MAA. American invitational mathematics examination - aime. In American Invitational Mathematics Examination - AIME 2024, February 2024. URL https://maa.org/math-competitions/american-invitational-mathematics-examination-aime.

P. Micikevicius, D. Stosic, N. Burgess, M. Cornea, P. Dubey, R. Grisenthwaite, S. Ha, A. Heinecke, P. Judd, J. Kamalu, et al. FP8 formats for deep learning. arXiv preprint arXiv:2209.05433, 2022. Mistral. Cheaper, better, faster, stronger: Continuing to push the frontier of ai and making it accessible to all, 2024. URL https://mistral.ai/news/mixtral-8x22b.

S. Narang, G. Diamos, E. Elsen, P. Micikevicius, J. Alben, D. Garcia, B. Ginsburg, M. Houston, O. Kuchaiev, G. Venkatesh, et al. Mixed precision training. In Int. Conf. on Learning Representation, 2017.

B. Noune, P. Jones, D. Justus, D. Masters, and C. Luschi. 8-bit numerical formats for deep neural networks. arXiv preprint arXiv:2206.02915, 2022. NVIDIA. Improving network performance of HPC systems using NVIDIA Magnum IO NVSMEM and GPUDirect Async. https://developer.nvidia.com/blog/improving-network-performance-of-hpc-systems-using-nvidia-magnum-io-nvshmem-and-GPUdirect-async, 2022.NVIDIA. Blackwell architecture. https://www.nvidia.com/en-us/data-center/technologies/blackwell-architecture/, 2024a.NVIDIA. TransformerEngine, 2024b. URL https://github.com/NVIDIA/TransformerEngine. Accessed: 2024-11-19.

OpenAI. Hello GPT-4o, 2024a. URL https://openai.com/index/hello-gpt-4o/.

OpenAI. Multilingual massive multitask language understanding (mmmlu), 2024b. URL https://huggingface.co/datasets/openai/MMMLU.

OpenAI. Introducing SimpleQA, 2024c. URL https://openai.com/index/introducing-simpleqa/.

OpenAI. Introducing SWE-bench verified we're releasing a human-validated subset of swebench that more, 2024d. URL https://openai.com/index/introducing-swe-bench-verified/.

B. Peng, J. Quesnelle, H. Fan, and E. Shippole. Yarn: Efficient context window extension of large language models. arXiv preprint arXiv:2309.00071, 2023a.

H. Peng, K. Wu, Y. Wei, G. Zhao, Y. Yang, Z. Liu, Y. Xiong, Z. Yang, B. Ni, J. Hu, et al. FP8-LM:Training FP8 large language models. arXiv preprint arXiv:2310.18313, 2023b.

P. Qi, X. Wan, G. Huang, and M. Lin. Zero bubble pipeline parallelism. arXiv preprint arXiv:2401.10241, 2023a.

P. Qi, X. Wan, G. Huang, and M. Lin. Zero bubble pipeline parallelism, 2023b. URL https://arxiv.org/abs/2401.10241.

Qwen. Qwen technical report. arXiv preprint arXiv:2309.16609, 2023.

Qwen. Introducing Qwen1.5, 2024a. URL https://qwenlm.github.io/blog/qwen1.5.

Qwen. Qwen2.5: A party of foundation models, 2024b. URL https://qwenlm.github.io/blog/qwen2.5.

S. Rajbhandari, J. Rasley, O. Ruwase, and Y. He. Zero: Memory optimizations toward training trillion parameter models. In SC20: International Conference for High Performance Computing,Networking, Storage and Analysis, pages 1–16. IEEE, 2020.

D. Rein, B. L. Hou, A. C. Stickland, J. Petty, R. Y. Pang, J. Dirani, J. Michael, and S. R. Bowman.GPQA: A graduate-level google-proof q&a benchmark. arXiv preprint arXiv:2311.12022, 2023.

B. D. Rouhani, R. Zhao, A. More, M. Hall, A. Khodamoradi, S. Deng, D. Choudhary, M. Cornea,E. Dellinger, K. Denolf, et al. Microscaling data formats for deep learning.

arXiv preprintarXiv:2310.10537, 2023a.
B. D. Rouhani, R. Zhao, A. More, M. Hall, A. Khodamoradi, S. Deng, D. Choudhary, M. Cornea,E. Dellinger, K. Denolf, et al. Microscaling data formats for deep learning. arXiv preprintarXiv:2310.10537, 2023b.
K. Sakaguchi, R. L. Bras, C. Bhagavatula, and Y. Choi. Winogrande: An adversarial winograd schema challenge at scale, 2019.
Z. Shao, P. Wang, Q. Zhu, R. Xu, J. Song, M. Zhang, Y. Li, Y. Wu, and D. Guo. Deepseekmath:Pushing the limits of mathematical reasoning in open language models. arXiv preprint arXiv:2402.03300, 2024.
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.
F. Shi, M. Suzgun, M. Freitag, X. Wang, S. Srivats, S. Vosoughi, H. W. Chung, Y. Tay, S. Ruder,D. Zhou, D. Das, and J. Wei. Language models are multilingual chain-of-thought reasoners. In The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali,
Rwanda, May 1-5, 2023. OpenReview.net, 2023. URL https://openreview.net/forum?id=fR3wGCk-IXp.
Y. Shibata, T. Kida, S. Fukamachi, M. Takeda, A. Shinohara, T. Shinohara, and S. Arikawa. Byte pair encoding: A text compression scheme that accelerates pattern matching. 1999.
J. Su, M. Ahmed, Y. Lu, S. Pan, W. Bo, and Y. Liu. Roformer: Enhanced transformer with rotary position embedding. Neurocomputing, 568:127063, 2024.
K. Sun, D. Yu, D. Yu, and C. Cardie. Investigating prior knowledge for challenging chinese machine reading comprehension, 2019a.
M. Sun, X. Chen, J. Z. Kolter, and Z. Liu. Massive activations in large language models. arXivpreprint arXiv:2402.17762, 2024.
X. Sun, J. Choi, C.-Y. Chen, N. Wang, S. Venkataramani, V. V. Srinivasan, X. Cui, W. Zhang, and K. Gopalakrishnan. Hybrid 8-bit floating point (HFP8) training and inference for deep neural networks. Advances in neural information processing systems, 32, 2019b.
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.
V. Thakkar, P. Ramani, C. Cecka, A. Shivam, H. Lu, E. Yan, J. Kosaian, M. Hoemmen, H. Wu, A. Kerr, M. Nicely, D. Merrill, D. Blasig, F. Qiao, P. Majcher, P. Springer, M. Hohnerbach, J. Wang, and M. Gupta. CUTLASS, Jan. 2023. URL https://github.com/NVIDIA/cutlass.
H. Touvron, T. Lavril, G. Izacard, X. Martinet, M.-A. Lachaux, T. Lacroix, B. Rozière, N. Goyal, E. Hambro, F. Azhar, et al. LLaMA: Open and efficient foundation language models. arXiv preprint arXiv:2302.13971, 2023a.
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, Ł. Kaiser, and I. Polosukhin. Attention is all you need. Advances in neural information processing systems, 30,2017.
L. Wang, H. Gao, C. Zhao, X. Sun, and D. Dai. Auxiliary-loss-free load balancing strategy for mixture-of-experts. CoRR, abs/2408.15664, 2024a. URL https://doi.org/10.48550/arXiv.2408.15664.
Y. Wang, X. Ma, G. Zhang, Y. Ni, A. Chandra, S. Guo, W. Ren, A. Arulraj, X. He, Z. Jiang, T. Li, M. Ku, K. Wang, A. Zhuang, R. Fan, X. Yue, and W. Chen. Mmlu-pro: A more robust and challenging multi-task language understanding benchmark. CoRR, abs/2406.01574, 2024b. URL https://doi.org/10.48550/arXiv.2406.01574.
T. Wei, J. Luan, W. Liu, S. Dong, and B. Wang. Cmath: Can your language model pass chinese elementary school math test?, 2023.
M. Wortsman, T. Dettmers, L. Zettlemoyer, A. Morcos, A. Farhadi, and L. Schmidt. Stable and low-precision training for large-scale vision-language models. Advances in Neural Information Processing Systems, 36:10271–10298, 2023.
H. Xi, C. Li, J. Chen, and J. Zhu. Training transformers with 4-bit integers. Advances in Neural Information Processing Systems, 36:49146–49168, 2023.
C. S. Xia, Y. Deng, S. Dunn, and L. Zhang. Agentless: Demystifying llm-based software engineering agents. arXiv preprint, 2024.
H. Xia, T. Ge, P. Wang, S. Chen, F. Wei, and Z. Sui. Speculative decoding: Exploiting speculative execution for accelerating seq2seq generation. In Findings of the Association for Computational Linguistics: EMNLP 2023, Singapore, December 6-10, 2023, pages 3909–3925. Association for Computational Linguistics, 2023. URL https://doi.org/10.18653/v1/2023.findings-emnlp.257.
G. Xiao, J. Lin, M. Seznec, H. Wu, J. Demouth, and S. Han. Smoothquant: Accurate and efficient post-training quantization for large language models. In International Conference on Machine Learning, pages 38087–38099. PMLR, 2023.
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.
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.
W. Zhong, R. Cui, Y. Guo, Y. Liang, S. Lu, Y. Wang, A. Saied, W. Chen, and N. Duan. AGIEval: A human-centric benchmark for evaluating foundation models. CoRR, abs/2304.06364, 2023. doi: 10.48550/arXiv.2304.06364. URL https://doi.org/10.48550/arXiv.2304.06364.
J. Zhou, T. Lu, S. Mishra, S. Brahma, S. Basu, Y. Luan, D. Zhou, and L. Hou. Instruction-following evaluation for large language models. arXiv preprint arXiv:2311.07911, 2023.

附录

A.贡献和致谢

研究与工程

Aixin Liu
Bing Xue
Bingxuan Wang
Bochao Wu
Chengda Lu
Chenggang Zhao
Chengqi Deng
Chenyu Zhang*
Chong Ruan
Damai Dai
Daya Guo
Dejian Yang
Deli Chen
Erhang Li
Fangyun Lin
Fucong Dai
Fuli Luo*
Guangbo Hao
Guanting Chen
Guowei Li
H. Zhang
Han Bao*
Hanwei Xu
Haocheng Wang*
Haowei Zhang
Honghui Ding
Huajian Xin*

Huazuo Gao
Hui Qu
Jianzhong Guo
Jiashi Li
Jiawei Wang*
Jingchang Chen
Jingyang Yuan
Junjie Qiu
Junlong Li
Junxiao Song
Kai Dong
Kai Hu*
Kaige Gao
Kang Guan
Kexin Huang
Kuai Yu
Lean Wang
Lecong Zhang
Liang Zhao
Litong Wang
Liyue Zhang
Mingchuan Zhang
Minghua Zhang
Minghui Tang
Panpan Huang
Peiyi Wang
Qiancheng Wang
Qihao Zhu
Qinyu Chen
Qiushi Du
Ruiqi Ge
Ruisong Zhang
Ruizhe Pan
Runji Wang
Runxin Xu
Ruoyu Zhang
Shanghao Lu
Shangyan Zhou
Shanhuang Chen
Shengfeng Ye
Shirong Ma
Shiyu Wang
Shuiping Yu
Shunfeng Zhou

Shuting Pan
Tao Yun
Tian Pei
Wangding Zeng
Wanjia Zhao*
Wen Liu
Wenfeng Liang
Wenjun Gao
Wenqin Yu
Wentao Zhang
Xiao Bi
Xiaodong Liu
Xiaohan Wang
Xiaokang Chen
Xiaokang Zhang
Xiaotao Nie
Xin Cheng
Xin Liu
Xin Xie
Xingchao Liu
Xingkai Yu
Xinyu Yang
Xinyuan Li
Xuecheng Su
Xuheng Lin
Y.K. Li
Y.Q. Wang
Y.X. Wei
Yang Zhang
Yanhong Xu
Yao Li
Yao Zhao
Yaofeng Sun
Yaohui Wang
Yi Yu
Yichao Zhang
Yifan Shi
Yiliang Xiong
Ying He
Yishi Piao
Yisong Wang
Yixuan Tan
Yiyang Ma*
Yiyuan Liu

Yongqiang Guo
Yu Wu
Yuan Ou
Yuduan Wang
Yue Gong
Yuheng Zou
Yujia He
Yunfan Xiong
Yuxiang Luo
Yuxiang You
Yuxuan Liu
Yuyang Zhou
Z.F. Wu
Z.Z. Ren
Zehui Ren
Zhangli Sha
Zhe Fu
Zhean Xu
Zhenda Xie
Zhengyan Zhang
Zhewen Hao
Zhibin Gou
Zhicheng Ma
Zhigang Yan
Zhihong Shao
Zhiyu Wu
Zhuoshu Li
Zihui Gu
Zijia Zhu
Zijun Liu*
Zilin Li
Ziwei Xie
Ziyang Song
Ziyi Gao
Zizheng Pan
数据注释
Bei Feng
Hui Li
J.L. Cai
Jiaqi Ni
Lei Xu
Meng Li
Ning Tian
R.J. Chen

R.L. Jin
Ruyi Chen
S.S. Li
Shuang Zhou
Tianyu Sun
X.Q. Li
Xiangyue Jin
Xiaojin Shen
Xiaosha Chen
Xiaowen Sun
Xiaoxiang Wang
Xinnan Song
Xinyi Zhou
Y.X. Zhu
Yanhong Xu
Yanping Huang
Yaohui Li
Yi Zheng
Yuchen Zhu
Yunxian Ma
Zhen Huang
Zhipeng Xu
Zhongyu Zhang
业务与合规
Dongjie Ji
Jian Liang
Jin Chen
Leyi Xia
Miaojun Wang
Mingming Li
Peng Zhang
Shaoqing Wu
Shengfeng Ye
T. Wang
W.L. Xiao
Wei An
Xianzu Wang
Xinxia Shan
Ying Tang
Yukun Zha
Yuting Yan
Zhen Zhang
在每个角色中，作者都按名字的字母顺序列出。标有*的名字表示已经离开我们团队的人。
B. 低精度训练中的消融研究

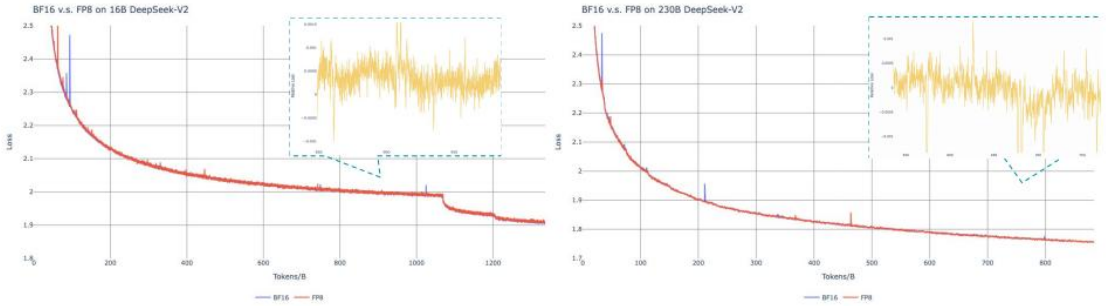


图 10 BF16 和 FP8 训练之间的|损失曲线比较。结果通过系数为 0.9 的指数移动平均线（EMA）进行平滑处理。

B.1.FP8 v.s.BF16 的训练

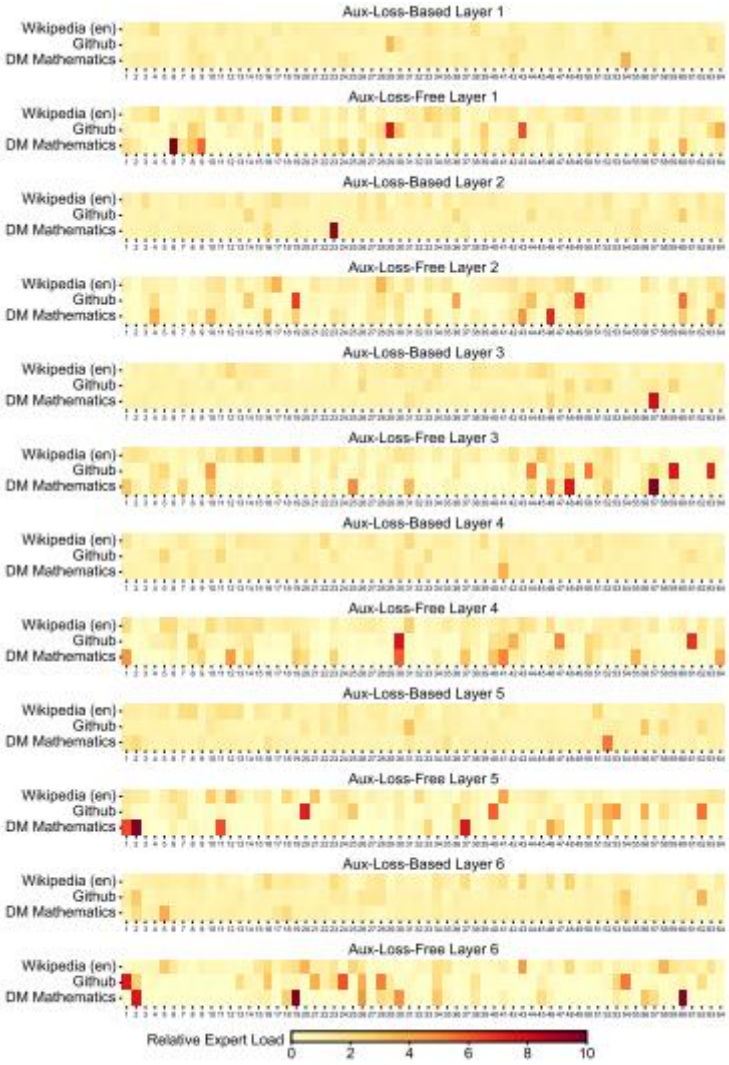
我们验证了我们的 FP8 混合精度框架，并与 BF16 训练进行了两个不同尺度的比较。在小尺度下，我们训练了一个包含大约 16B 个总参数的基线 MoE 模型。在大规模上，我们训练了一个包含约 0.9T 个总参数的基线 MoE 模型。我们展示了图 10 中的训练曲线，并证明了我们的高精度积累和细粒度量化策略的相对误差保持在 0.25%以下。

B.2.块量化探讨

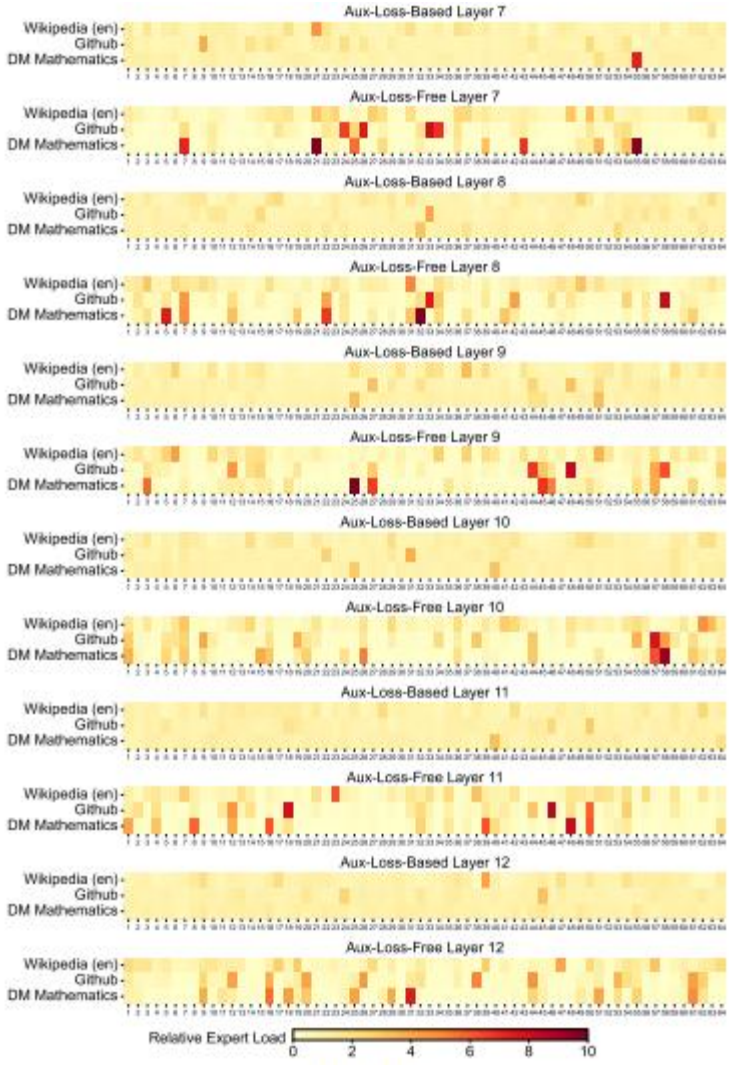
虽然我们的平铺细粒度量化有效地减轻了特征异常值引入的误差，但它需要不同的分组来进行激活量化，即前通 1x128，后通 128x1。活化梯度也需要类似的过程。一个简单的策略是对每 128x128 个元素应用块级量化，就像我们量化模型权重的方式一样。这样一来，就只需要反向换位了。因此，我们进行了一个实验，其中所有与 Dgrad 相关的张量在块的基础上被量化。结果表明，计算激活梯度和以链状方式反向传播到浅层的 Dglad 操作对精度高度敏感。具体来说，激活梯度的块级量化会导致在一个包含大约 16B 总参数的 MoE 模型上的模型发散，训练约 300B 令牌。我们假设，这种敏感性的产生是因为激活梯度在标记之间高度不平衡，导致标记相关的异常值（Xi et al.，2023）。这些异常值不能通过块级量化方法来有效地管理。

C. 16B 基于辅助损失和辅助无损失模型的专家专业化模式

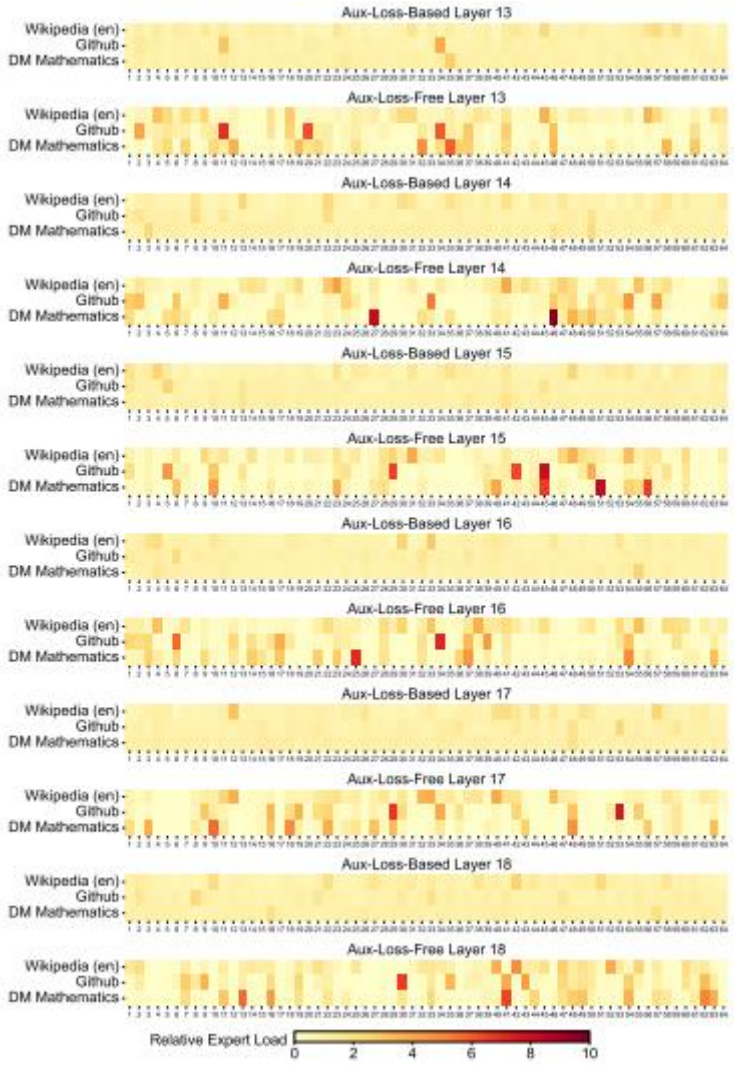
我们在桩测试集上记录了基于 16B 的辅助损失基线和无辅助损失模型的专家负载。无辅助损失模型往往在所有层中具有更大的专家专门化，如图 10 所示。


Aux-Loss-Based Layer 1
Aux-Loss-Free Layer 1
Aux-Loss-Based Layer 2
Aux-Loss-Free Layer 2
Aux-Loss-Based Layer 3
Aux-Loss-Free Layer 3
Aux-Loss-Based Layer 4
Aux-Loss-Free Layer 4
Aux-Loss-Based Layer 5
Aux-Loss-Free Layer 5
Aux-Loss-Based Layer 6
Aux-Loss-Free Layer 6
Wikipedia (en)
Github
DM Mathematics
Relative Expert Load
0
2
4
6
8
10

(a) Layers 1-7


Aux-Loss-Based Layer 7
Aux-Loss-Free Layer 7
Aux-Loss-Based Layer 8
Aux-Loss-Free Layer 8
Aux-Loss-Based Layer 9
Aux-Loss-Free Layer 9
Aux-Loss-Based Layer 10
Aux-Loss-Free Layer 10
Aux-Loss-Based Layer 11
Aux-Loss-Free Layer 11
Aux-Loss-Based Layer 12
Aux-Loss-Free Layer 12
Wikipedia (en)
Github
DM Mathematics
Relative Expert Load
0
2
4
6
8
10

(b) Layers 7-13



(c) Layers 13-19

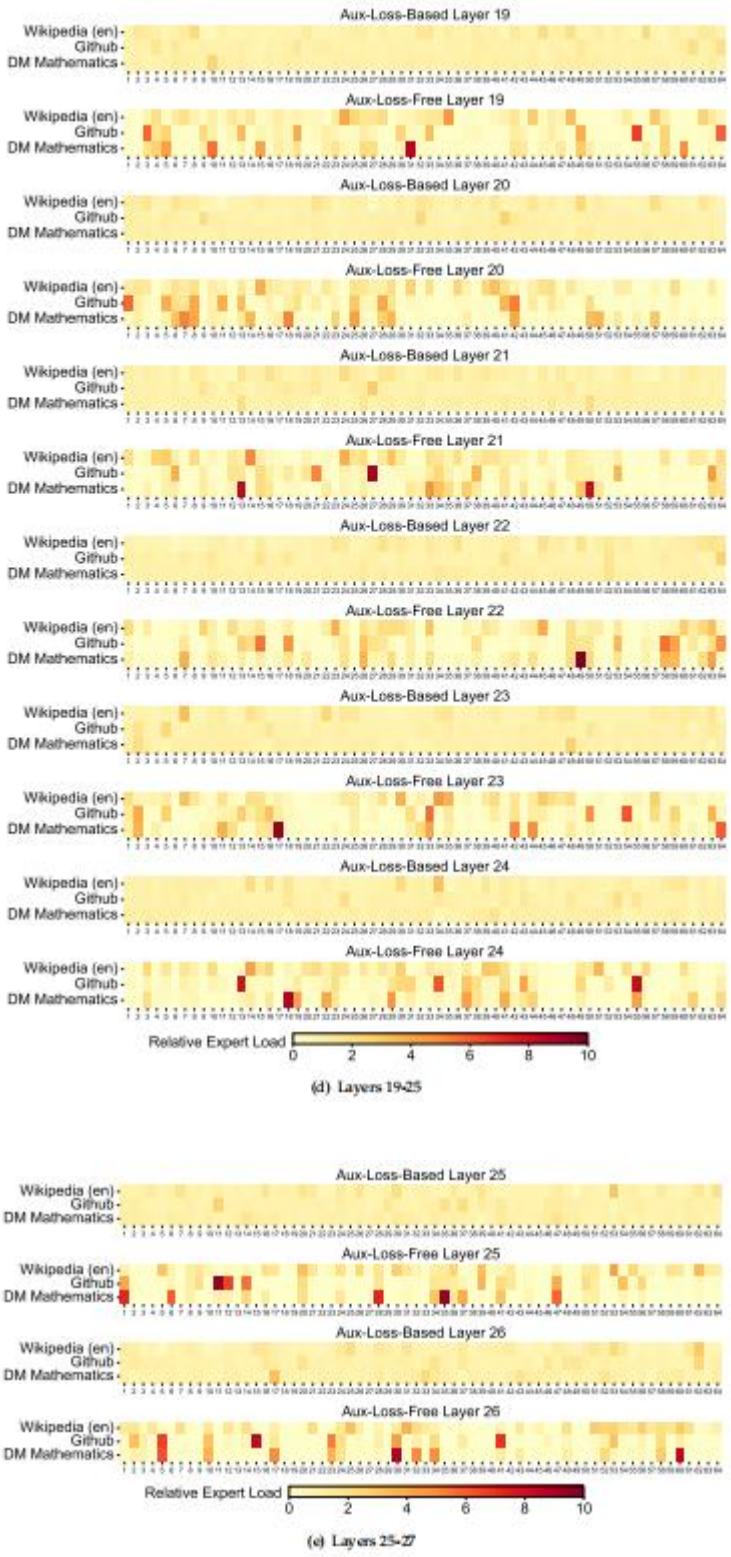


图 10 在桩测试集中的三个领域上的无辅助损耗和基于辅助损耗的模型的专家负载。无辅助损失模型比基于辅助损失的模型显示出更大的专家专业化模式。相对专家负荷表示实际专家负荷与理论上平衡的专家负荷的比值。