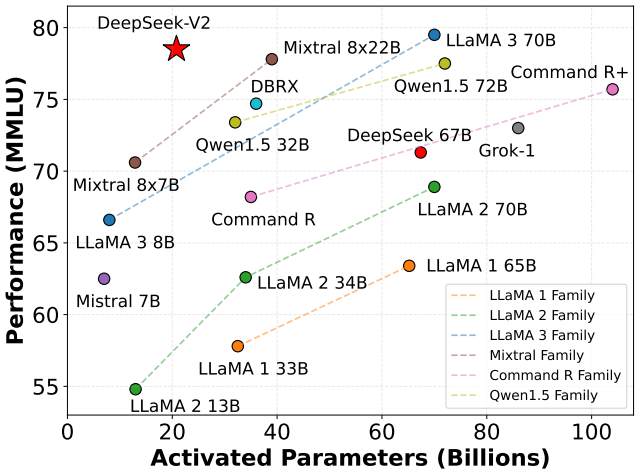
DeepSeek-V2：一种强大、经济且高效的语言混合专家模型

DeepSeek-AI

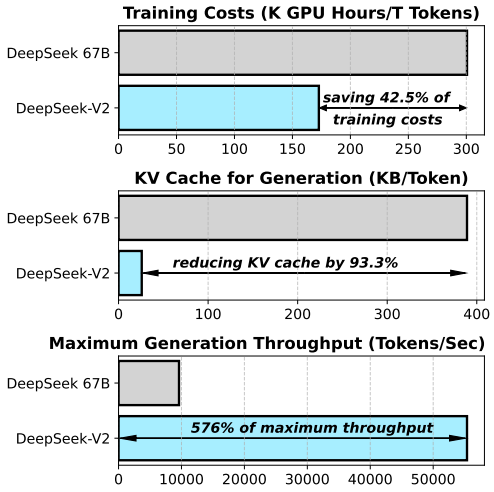
research@deepseek.com

Abstract

我们提出 DeepSeek-V2，这是一种强大的混合专家（MoE）语言模型，具有训练经济高效和推理高效的特点。该模型包含 2360 亿总参数，其中每个 token 激活 210 亿参数，并支持 128K token 的上下文长度。DeepSeek-V2 采用了包括多头潜在注意力（MLA）和 DeepSeekMoE 在内的创新架构。MLA 通过将键值（KV）缓存显著压缩为潜在向量来保证高效的推理，而 DeepSeekMoE 则通过稀疏计算以经济成本训练出强大的模型。与 DeepSeek 67B 相比，DeepSeek-V2 实现了显著更强的性能，同时节省了 42.5% 的训练成本，将 KV 缓存减少了 93.3%，并将最大生成吞吐量提升了 5.76 倍。我们在一个由 8.1T token 组成的高质量多源语料库上对 DeepSeek-V2 进行预训练，并进一步执行监督微调（SFT）和强化学习（RL）以充分释放其潜力。评估结果表明，即使仅激活 210 亿参数，DeepSeek-V2 及其对话版本仍在开源模型中达到顶尖性能。模型检查点可在 https://github.com/deepseek-ai/DeepSeek-V2 获取。



(a)



(b)

图 1 | (a) 不同开源模型在 MMLU 上的准确率与激活参数对比。(b) DeepSeek 67B（Dense）与 DeepSeek-V2 的训练成本与推理效率。

目录

1. 引言

在过去几年中，大语言模型（LLMs）(Anthropic, 2023; Google, 2023; OpenAI, 2022, 2023) 经历了快速发展，让我们得以窥见通用人工智能（AGI）的曙光。一般来说，LLM 的智能水平往往随着参数数量的增加而提升，使其能够在各种任务中展现出涌现能力 (Wei et al., 2022)。然而，这种提升是以更大的训练计算资源和潜在的推理吞吐量下降为代价的。这些限制带来了重大挑战，阻碍了 LLM 的广泛采用和应用。为了解决这一问题，我们推出了 DeepSeek-V2，这是一种强大的开源混合专家（MoE）语言模型，通过创新的 Transformer 架构实现了经济高效的训练和推理。该模型配备总计 2360 亿参数，其中每个 token 激活 210 亿参数，并支持 128K token 的上下文长度。

我们利用提出的 **多头潜在注意力（MLA）**和 **DeepSeekMoE** 优化了 Transformer 框架 (Vaswani et al., 2017) 中的注意力模块和前馈网络（FFNs）。（1）在注意力机制的背景下，多头注意力（MHA）(Vaswani et al., 2017) 的键值（KV）缓存对 LLM 的推理效率构成了重大障碍。为解决这一问题，人们探索了多种方法，包括分组查询注意力（GQA）(Ainslie et al., 2023) 和多查询注意力（MQA）(Shazeer, 2019)。然而，这些方法在试图减少 KV 缓存时往往会牺牲性能。为了兼顾两者优势，我们引入了 MLA，这是一种配备低秩键值联合压缩的注意力机制。经验表明，MLA 相比 MHA 实现了更优越的性能，同时在推理过程中显著减少了 KV 缓存，从而提升了推理效率。(2) 对于前馈网络 (FFNs)，我们遵循 DeepSeekMoE 架构 (Dai et al., 2024)，该架构采用细粒度专家分割和共享专家隔离，以在专家专业化方面发挥更高潜力。与 GShard (Lepikhin et al., 2021) 等传统 MoE 架构相比，DeepSeekMoE 架构展现出巨大优势，使我们能够以经济成本训练出强大的模型。由于我们在训练过程中采用了专家并行，我们还设计了补充机制来控制通信开销并确保负载均衡。通过结合这两种技术，DeepSeek-V2 同时具备了强大的性能（图 1(a)）、经济的训练成本和高效的推理吞吐量（图 1(b)）。

我们构建了一个由 8.1T token 组成的高质量多源预训练语料库。与 DeepSeek 67B（我们之前的发布版本）(DeepSeek-AI, 2024) 使用的语料库相比，该语料库的数据量更大，尤其是中文数据，且数据质量更高。我们首先在完整的预训练语料库上对 DeepSeek-V2 进行预训练。随后，我们收集了 150 万条对话会话，涵盖数学、代码、写作、推理、安全等多个领域，以对 DeepSeek-V2 Chat (SFT) 进行监督微调（SFT）。最后，我们遵循 DeepSeekMath (Shao et al., 2024) 的方法，采用组相对策略优化（GRPO）进一步使模型与人类偏好对齐，并生成 DeepSeek-V2 Chat (RL)。

我们在广泛的英文和中文基准测试上评估了 DeepSeek-V2，并将其与具有代表性的开源模型进行了比较。评估结果表明，即使仅激活 210 亿参数，DeepSeek-V2 仍在开源模型中达到顶尖性能，成为最强的开源 MoE 语言模型。图 1(a) 突出显示，在 MMLU 基准上，DeepSeek-V2 仅凭少量激活参数就取得了顶尖性能。此外，如图 1(b) 所示，与 DeepSeek 67B 相比，DeepSeek-V2 节省了 42.5% 的训练成本，将 KV 缓存减少了 93.3%，并将最大生成吞吐量提升了 5.76 倍。我们还在开放式基准测试上评估了 DeepSeek-V2 Chat (SFT) 和 DeepSeek-V2 Chat (RL)。值得注意的是，DeepSeek-V2 Chat (RL) 在 AlpacaEval 2.0 (Dubois et al., 2024) 上取得了 38.9 的长度控制胜率，在 MT-Bench (Zheng et al., 2023) 上取得了 8.97 的总分，在 AlignBench (Liu

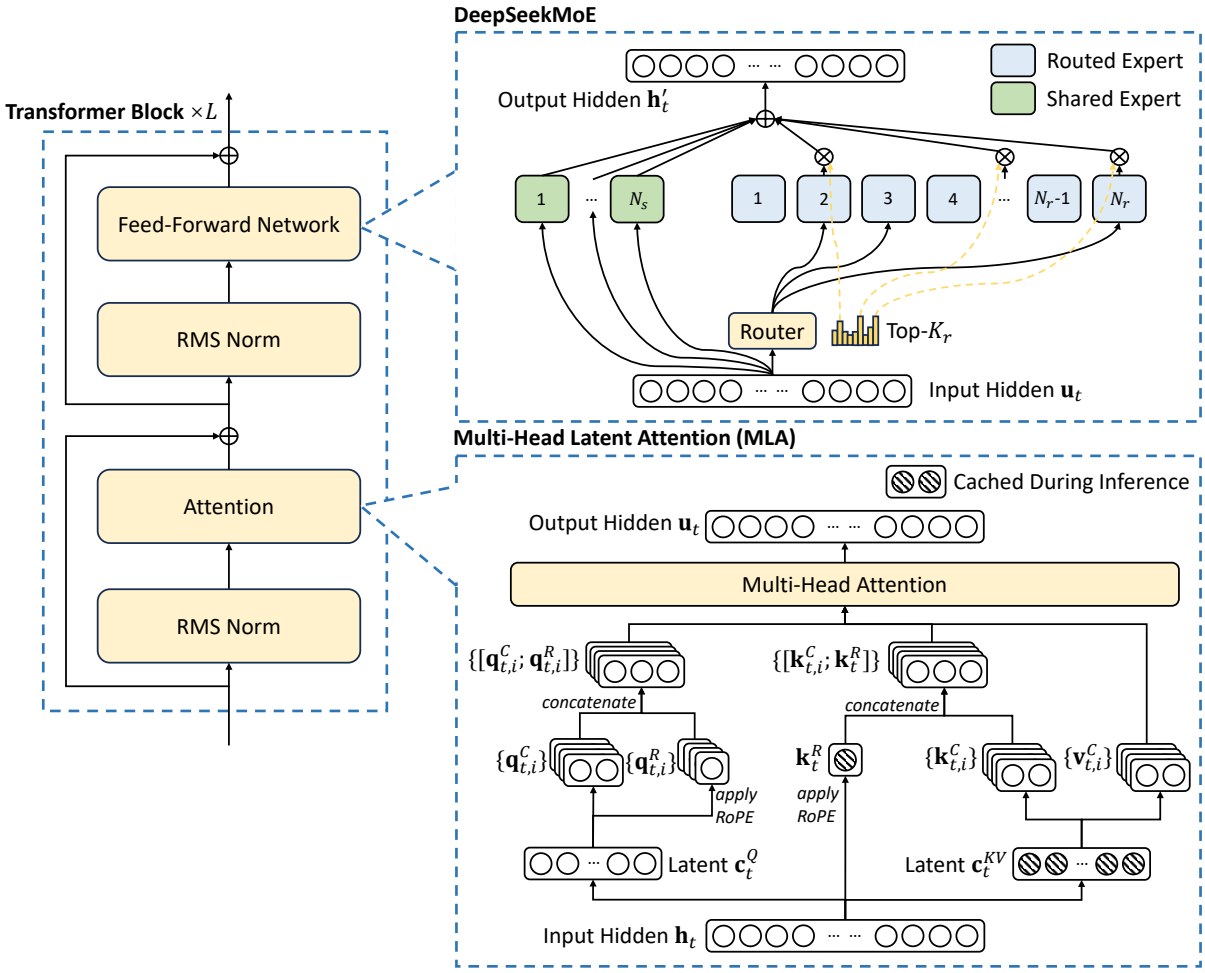


图 2 | DeepSeek-V2 架构示意图。MLA 通过显著减少生成过程中的 KV 缓存来确保高效推理，而 DeepSeekMoE 则通过稀疏架构以经济成本训练出强大的模型。

et al., 2023) 上取得了 7.91 的总分。英文开放式对话评估表明，DeepSeek-V2 Chat (RL) 在开源对话模型中具备顶尖性能。此外，AlignBench 上的评估表明，在中文方面，DeepSeek-V2 Chat (RL) 优于所有开源模型，甚至击败了大多数闭源模型。

为了促进对 MLA 和 DeepSeekMoE 的进一步研究与开发，我们还向开源社区发布了 DeepSeek-V2-Lite，这是一个配备了 MLA 和 DeepSeekMoE 的较小模型。该模型总计包含 157 亿参数，其中每个 token 激活 24 亿参数。关于 DeepSeek-V2-Lite 的详细说明见附录 B。

在本文的其余部分，我们首先详细介绍 DeepSeek-V2 的模型架构（第 2 节）。随后，我们介绍我们的预训练工作，包括训练数据构建、超参数设置、基础设施、长上下文扩展以及模型性能与效率的评估（第 3 节）。接着，我们展示我们在对齐方面的工作，涵盖监督微调（SFT）、强化学习（RL）、评估结果及其他讨论（第 4 节）。最后，我们总结结论，探讨 DeepSeek-V2 当前的局限性，并概述未来的工作（第 5 节）。

2. 架构

总体而言, DeepSeek-V2 仍基于 Transformer 架构 (Vaswani et al., 2017), 其中每个 Transformer 块由一个注意力模块和一个前馈网络（FFN）组成。然而，对于注意力模块和 FFN，我们都设计并采用了创新的架构。对于注意力机制，我们设计了 MLA，它利用低秩键值联合压缩来消除推理时键值缓存的瓶颈，从而支持高效推理。对于 FFN，我们采用了 DeepSeekMoE 架构 (Dai et al., 2024)，这是一种高性能的 MoE 架构，能够以经济成本训练出强大的模型。DeepSeek-V2 的架构示意图如图 2 所示，我们将在本节中介绍 MLA 和 DeepSeekMoE 的详细信息。对于其他细微细节（例如层归一化和 FFN 中的激活函数), 除非特别说明, DeepSeek-V2 均遵循 DeepSeek 67B (DeepSeek-AI, 2024) 的设置。

2.1. 多头潜在注意力：提升推理效率

传统的 Transformer 模型通常采用多头注意力（MHA） (Vaswani et al., 2017)，但在生成过程中, 其庞大的键值（KV）缓存会成为限制推理效率的瓶颈。为了减少 KV 缓存, 研究者提出了多查询注意力（MQA） (Shazeer, 2019) 和分组查询注意力（GQA） (Ainslie et al., 2023)。它们所需的 KV 缓存规模较小，但性能无法与 MHA 相媲美（我们在附录 D.1 中提供了 MHA、GQA 和 MQA 的消融实验)。

针对 DeepSeek-V2，我们设计了一种创新的注意力机制，称为多头潜在注意力（MLA)。借助低秩键值联合压缩，MLA 在实现比 MHA 更优性能的同时，仅需显著更少的 KV 缓存。下文将详细介绍其架构，并在附录 D.2 中提供 MLA 与 MHA 的对比分析。

2.1.1. 预备知识：标准多头注意力

我们首先介绍标准的 MHA 机制作为背景。设 d 为嵌入维度，n_h 为注意力头数，d_h 为每个头的维度，$\mathbf{h}_t \in \mathbb{R}^d$ 为注意力层中第 t 个 token 的注意力输入。标准 MHA 首先通过三个矩阵 $W^Q, W^K, W^V \in \mathbb{R}^{d_h n_h \times d}$ 分别生成 $\mathbf{q}_t, \mathbf{k}_t, \mathbf{v}_t \in \mathbb{R}^{d_h n_h}$：

$$\mathbf{q}_t = W^Q \mathbf{h}_t, \tag{1}$$

$$\mathbf{k}_t = W^K \mathbf{h}_t, \tag{2}$$

$$\mathbf{v}_t = W^V \mathbf{h}_t, \tag{3}$$

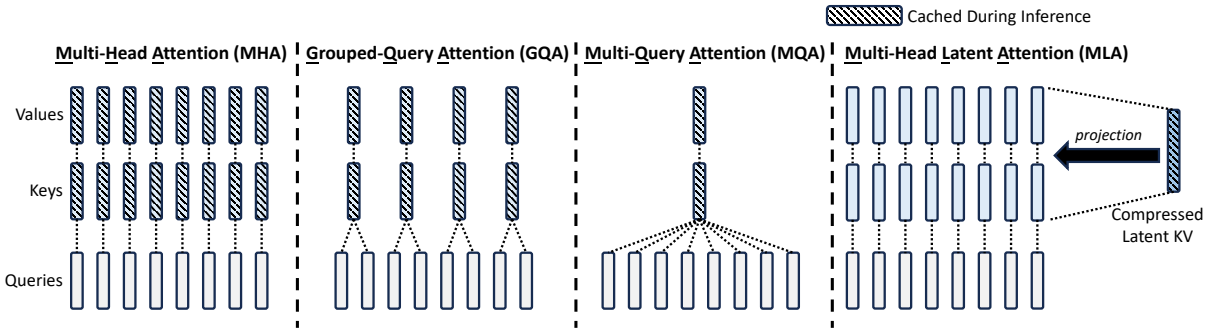


图 3 | 多头注意力（MHA）、分组查询注意力（GQA）、多查询注意力（MQA）和多头潜在注意力（MLA）的简化示意图。通过将键和值联合压缩为一个潜在向量，MLA 在推理过程中显著减少了 KV 缓存。

随后，$\mathbf{q}_t, \mathbf{k}_t, \mathbf{v}_t$ 将被切分为 n_h 个头以进行多头注意力计算：

$$[\mathbf{q}_{t,1}; \mathbf{q}_{t,2}; ...; \mathbf{q}_{t,n_h}] = \mathbf{q}_t, \tag{4}$$

$$[\mathbf{k}_{t,1}; \mathbf{k}_{t,2}; ...; \mathbf{k}_{t,n_h}] = \mathbf{k}_t, \tag{5}$$

$$[\mathbf{v}_{t,1}; \mathbf{v}_{t,2}; ...; \mathbf{v}_{t,n_h}] = \mathbf{v}_t, \tag{6}$$

$$\mathbf{o}_{t,i} = \sum_{j=1}^{t} \mathrm{Softmax}_j\left(\frac{\mathbf{q}_{t,i}^T \mathbf{k}_{j,i}}{\sqrt{d_h}}\right)\mathbf{v}_{j,i}, \tag{7}$$

$$\mathbf{u}_t = W^O[\mathbf{o}_{t,1}; \mathbf{o}_{t,2}; ...; \mathbf{o}_{t,n_h}], \tag{8}$$

其中 $\mathbf{q}_{t,i}, \mathbf{k}_{t,i}, \mathbf{v}_{t,i} \in \mathbb{R}^{d_h}$ 分别表示第 i 个注意力头的查询、键和值；$W^O \in \mathbb{R}^{d \times d_h n_h}$ 表示输出投影矩阵。在推理过程中，所有键和值都需要缓存以加速推理，因此 MHA 需要为每个 token 缓存 $2n_h d_h l$ 个元素。在模型部署中，这种庞大的 KV 缓存是一个主要瓶颈，限制了最大批处理大小和序列长度。

2.1.2. 低秩键值联合压缩

MLA 的核心在于对键和值进行低秩联合压缩，以减少 KV 缓存：

$$\mathbf{c}_t^{KV} = W^{DKV}\mathbf{h}_t, \tag{9}$$

$$\mathbf{k}_t^C = W^{UK}\mathbf{c}_t^{KV}, \tag{10}$$

$$\mathbf{v}_t^C = W^{UV}\mathbf{c}_t^{KV}, \tag{11}$$

其中 $\mathbf{c}_t^{KV} \in \mathbb{R}^{d_c}$ 是键和值的压缩潜在向量；$d_c (\ll d_h n_h)$ 表示 KV 压缩维度；$W^{DKV} \in \mathbb{R}^{d_c \times d}$ 是下投影矩阵；$W^{UK}, W^{UV} \in \mathbb{R}^{d_h n_h \times d_c}$ 分别是键和值的上投影矩阵。在推理过程中，MLA 仅需缓存 $\mathbf{c}_t^{KV}$，因此其 KV 缓存仅包含 $d_c l$ 个元素，其中 l 表示层数。此外，在推理过程中，由于 W^{UK} 可以被吸收进 W^Q，且 W^{UV} 可以被吸收进 W^O，我们甚至无需显式计算键和值用于注意力计算。图 3 直观地展示了 MLA 中的 KV 联合压缩如何减少 KV 缓存。

此外，为了减少训练过程中的激活内存，我们对查询也进行了低秩压缩，尽管这并不能减少 KV 缓存：

$$\mathbf{c}_t^Q = W^{DQ}\mathbf{h}_t, \tag{12}$$

$$\mathbf{q}_t^C = W^{UQ}\mathbf{c}_t^Q, \tag{13}$$

其中 $\mathbf{c}_t^Q \in \mathbb{R}^{d_c'}$ 是查询的压缩潜在向量；$d_c'(\ll d_h n_h)$ 表示查询压缩维度；$W^{DQ} \in \mathbb{R}^{d_c' \times d}, W^{UQ} \in \mathbb{R}^{d_h n_h \times d_c'}$ 分别是查询的下投影和上投影矩阵。

2.1.3. 解耦旋转位置编码

遵循 DeepSeek 67B (DeepSeek-AI, 2024) 的设计，我们计划在 DeepSeek-V2 中使用旋转位置编码（RoPE）(Su et al., 2024)。然而，RoPE 与低秩 KV 压缩不兼容。具体来说，RoPE 对键和查询都具有位置敏感性。如果我们对键 $\mathbf{k}_t^C$ 应用 RoPE，公式 10 中的 W^{UK} 将与一个具有位置敏感性的 RoPE 矩阵耦合。这样一来，在推理过程中 W^{UK} 将无法再被吸收进 W^Q，因为与当前生成 token 相关的 RoPE 矩阵将位于 W^Q 和 W^{UK} 之间，而矩阵乘法不满足交换律。因此，在推理过程中我们必须重新计算所有前缀 token 的键，这将严重阻碍推理效率。

作为解决方案，我们提出了解耦 RoPE 策略，该策略使用额外的多头查询 $\mathbf{q}_{t,i}^R \in \mathbb{R}^{d_h^R}$ 和一个共享键 $\mathbf{k}_t^R \in \mathbb{R}^{d_h^R}$ 来承载 RoPE，其中 d_h^R 表示解耦查询和键的每头维度。采用解耦 RoPE 策略后，MLA 执行以下计算：

$$[\mathbf{q}_{t,1}^R; \mathbf{q}_{t,2}^R; ...; \mathbf{q}_{t,n_h}^R] = \mathbf{q}_t^R = \mathrm{RoPE}(W^{QR}\mathbf{c}_t^Q), \tag{14}$$

$$\mathbf{k}_t^R = \mathrm{RoPE}(W^{KR}\mathbf{h}_t), \tag{15}$$

$$\mathbf{q}_{t,i} = [\mathbf{q}_{t,i}^C; \mathbf{q}_{t,i}^R], \tag{16}$$

$$\mathbf{k}_{t,i} = [\mathbf{k}_{t,i}^C; \mathbf{k}_t^R], \tag{17}$$

$$\mathbf{o}_{t,i} = \sum_{j=1}^{t} \mathrm{Softmax}_j\left(\frac{\mathbf{q}_{t,i}^T \mathbf{k}_{j,i}}{\sqrt{d_h + d_h^R}}\right)\mathbf{v}_{j,i}^C, \tag{18}$$

$$\mathbf{u}_t = W^O[\mathbf{o}_{t,1}; \mathbf{o}_{t,2}; ...; \mathbf{o}_{t,n_h}], \tag{19}$$

其中 $W^{QR} \in \mathbb{R}^{d_h^R n_h \times d_c'}$ 和 $W^{KR} \in \mathbb{R}^{d_h^R \times d}$ 分别是用于生成解耦查询和键的矩阵；$\mathrm{RoPE}(\cdot)$ 表示应用 RoPE 矩阵的操作；$[\cdot;\cdot]$ 表示拼接操作。在推理过程中，解耦键也需要被缓存。因此，DeepSeek-V2 总共需要包含 $(d_c + d_h^R)l$ 个元素的 KV 缓存。

为了完整展示 MLA 的计算过程，我们还在附录 C 中整理并提供了其完整公式。

2.1.4. 键值缓存对比

我们在表 1 中展示了不同注意力机制下每个 token 的 KV 缓存对比。MLA 仅需少量的 KV 缓存，相当于仅含 2.25 个组的 GQA，但能实现比 MHA 更强的性能。

注意力机制	每个 Token 的 KV 缓存（元素数量）	性能
多头注意力 (MHA)	$2n_h d_h l$	强
分组查询注意力 (GQA)	$2n_g d_h l$	中等
多查询注意力 (MQA)	$2d_h l$	弱
MLA (本文)	$(d_c + d_h^R)l \approx \frac{9}{2} d_h l$	更强

表 1 | 不同注意力机制下每个 token 的 KV 缓存对比。n_h 表示注意力头数，d_h 表示每个注意力头的维度，l 表示层数，n_g 表示 GQA 中的组数，d_c 和 d_h^R 分别表示 MLA 中的 KV 压缩维度和解耦查询与键的每头维度。KV 缓存量以元素数量衡量，不考虑存储精度。对于 DeepSeek-V2，d_c 设置为 $4d_h$，d_h^R 设置为 $\frac{d_h}{2}$。因此，其 KV 缓存量相当于仅含 2.25 个组的 GQA，但其性能强于 MHA。

2.2. DeepSeekMoE：以经济成本训练强大模型

2.2.1. 基础架构

对于前馈网络（FFN），我们采用了 DeepSeekMoE 架构 (Dai et al., 2024)。DeepSeekMoE 包含两个核心思想：将专家划分为更细的粒度以实现更高的专家专业化和更准确的知识获取，以及隔离部分共享专家以缓解路由专家之间的知识冗余。在激活专家数量和总专家参数相同的情况下，DeepSeekMoE 能够大幅超越 GShard (Lepikhin et al., 2021) 等传统 MoE 架构。

设 $\mathbf{u}_t$ 为第 t 个 token 的 FFN 输入，我们按如下方式计算 FFN 输出 $\mathbf{h}'_t$：

$$\mathbf{h}'_t = \mathbf{u}_t + \sum_{i=1}^{N_s} \mathrm{FFN}_i^{(s)}(\mathbf{u}_t) + \sum_{i=1}^{N_r} g_{i,t}\, \mathrm{FFN}_i^{(r)}(\mathbf{u}_t), \tag{20}$$

$$g_{i,t} = \begin{cases} s_{i,t}, & s_{i,t} \in \mathrm{Topk}(\{s_{j,t} | 1 \leqslant j \leqslant N_r\}, K_r), \\ 0, & \text{otherwise}, \end{cases} \tag{21}$$

$$s_{i,t} = \mathrm{Softmax}_i\left(\mathbf{u}_t^T \mathbf{e}_i\right), \tag{22}$$

其中 N_s 和 N_r 分别表示共享专家和路由专家的数量；$\mathrm{FFN}_i^{(s)}(\cdot)$ 和 $\mathrm{FFN}_i^{(r)}(\cdot)$ 分别表示第 i 个共享专家和第 i 个路由专家；K_r 表示激活的路由专家数量；$g_{i,t}$ 是第 i 个专家的门控值；$s_{i,t}$ 是 token 与专家之间的亲和力；$\mathbf{e}_i$ 是第 i 个路由专家在本层的中心点；$\mathrm{Topk}(\cdot, K)$ 表示由第 t 个 token 与所有路由专家计算出的亲和力得分中 K 个最高得分组成的集合。

2.2.2. 设备受限路由

我们设计了一种设备受限路由机制，以限制与 MoE 相关的通信开销。当采用专家并行时，路由专家将分布在多个设备上。对于每个 token，其 MoE 相关的通信频率与其目标专家所覆盖的设备数量成正比。由于 DeepSeekMoE 中采用了细粒度的专家划分，激活的专家数量可能很大，因此如果应用专家并行，MoE 相关的通信开销将更为高昂。

对于 DeepSeek-V2，除了对路由专家进行简单的 Top-K 选择外，我们还额外确保每个 token

的目标专家最多分布在 M 个设备上。具体来说，对于每个 token，我们首先选择包含最高亲和力得分专家的 M 个设备。然后，我们在这些 M 个设备上的专家中进行 Top-K 选择。在实践中，我们发现当 $M \geqslant 3$ 时，设备受限路由能够实现与无限制 Top-K 路由大致相当的良好性能。

2.2.3. 用于负载均衡的辅助损失

我们在自动学习的路由策略中考虑了负载均衡问题。首先，负载不均衡会增加路由崩溃的风险 (Shazeer et al., 2017)，导致部分专家无法得到充分训练和利用。其次，当采用专家并行时，负载不均衡会降低计算效率。在 DeepSeek-V2 的训练过程中，我们设计了三种辅助损失，分别用于控制专家级负载均衡（$\mathcal{L}_{\text{ExpBal}}$）、设备级负载均衡（$\mathcal{L}_{\text{DevBal}}$）和通信平衡（$\mathcal{L}_{\text{CommBal}}$）。

专家级平衡损失。 我们采用专家级平衡损失 (Fedus et al., 2021; Lepikhin et al., 2021) 来缓解路由崩溃的风险：

$$\mathcal{L}_{\text{ExpBal}} = \alpha_1 \sum_{i=1}^{N_r} f_i P_i, \tag{23}$$

$$f_i = \frac{N_r}{K_r T} \sum_{t=1}^{T} \mathbb{1}(\text{Token } t \text{ selects Expert } i), \tag{24}$$

$$P_i = \frac{1}{T} \sum_{t=1}^{T} s_{i,t}, \tag{25}$$

其中 α_1 是称为专家级平衡因子的超参数；$\mathbb{1}(\cdot)$ 表示指示函数；T 表示序列中的 token 数量。

设备级平衡损失。 除了专家级平衡损失外，我们还额外设计了一种设备级平衡损失，以确保不同设备间的计算负载均衡。在 DeepSeek-V2 的训练过程中，我们将所有路由专家划分为 D 个组 $\{\mathcal{E}_1, \mathcal{E}_2, ..., \mathcal{E}_D\}$，并将每个组部署在单个设备上。设备级平衡损失计算如下：

$$\mathcal{L}_{\text{DevBal}} = \alpha_2 \sum_{i=1}^{D} f'_i P'_i, \tag{26}$$

$$f'_i = \frac{1}{|\mathcal{E}_i|} \sum_{j \in \mathcal{E}_i} f_j, \tag{27}$$

$$P'_i = \sum_{j \in \mathcal{E}_i} P_j, \tag{28}$$

其中 α_2 是称为设备级平衡因子的超参数。

通信平衡损失。 最后，我们引入通信平衡损失，以确保每个设备的通信负载均衡。尽管设备受限路由机制保证了每个设备的发送通信量是有界的，但如果某个设备接收的 token 数量多于其

他设备，实际的通信效率也会受到影响。为了缓解这一问题，我们设计了如下通信平衡损失：

$$\mathcal{L}_{\text{CommBal}} = \alpha_3 \sum_{i=1}^{D} f_i'' P_i'', \tag{29}$$

$$f_i'' = \frac{D}{MT} \sum_{t=1}^{T} \mathbb{1}(\text{Token } t \text{ is sent to Device } i), \tag{30}$$

$$P_i'' = \sum_{j \in \mathcal{E}_i} P_j, \tag{31}$$

其中 α_3 是称为通信平衡因子的超参数。设备受限路由机制的运行原则是确保每个设备最多向其他设备传输 MT 个隐藏状态。同时，通信平衡损失用于鼓励每个设备从其他设备接收大约 MT 个隐藏状态。通信平衡损失保证了设备间信息的均衡交换，从而促进高效通信。

2.2.4. Token 丢弃策略

尽管平衡损失旨在鼓励负载均衡，但必须承认它们无法保证严格的负载均衡。为了进一步缓解由负载不均衡引起的计算浪费，我们在训练期间引入了一种设备级的 token 丢弃策略。该方法首先计算每个设备的平均计算预算，这意味着每个设备的容量因子等效于 1.0。然后，受 Riquelme et al. (2021) 的启发，我们在每个设备上丢弃亲和力得分最低的 token，直到达到计算预算。此外，我们确保属于约 10% 训练序列的 token 永远不会被丢弃。通过这种方式，我们可以根据效率需求灵活决定在推理期间是否丢弃 token，并始终确保训练与推理之间的一致性。

3. 预训练

3.1. 实验设置

3.1.1. 数据构建

在保持与 DeepSeek 67B (DeepSeek-AI, 2024) 相同的数据处理阶段的同时，我们扩展了数据量并提升了数据质量。为了扩大我们的预训练语料库，我们挖掘了互联网数据的潜力并优化了清洗流程，从而恢复了大量被误删的数据。此外，我们纳入了更多中文数据，旨在更好地利用中文互联网上可用的语料。除了数据量之外，我们还注重数据质量。我们用来自各种来源的高质量数据丰富了预训练语料库，同时改进了基于质量的过滤算法。改进后的算法确保了大量无益数据将被移除，而有价值的数据将大部分得以保留。此外，我们从预训练语料库中过滤掉争议性内容，以减轻由特定区域文化引入的数据偏差。关于该过滤策略影响的详细讨论见附录 E。

我们采用了与 DeepSeek 67B 相同的分词器，该分词器基于字节级字节对编码（BBPE）算法构建，词表大小为 100K。我们的分词预训练语料库包含 8.1T 个 token，其中中文 token 数量比英文 token 多约 12%。

3.1.2. 超参数

模型超参数。 我们将 Transformer 层数设置为 60，隐藏层维度设置为 5120。所有可学习参数均以标准差 0.006 进行随机初始化。在 MLA 中，我们将注意力头数 n_h 设置为 128，每个头的维度 d_h 设置为 128。KV 压缩维度 d_c 设置为 512，查询压缩维度 d'_c 设置为 1536。对于解耦的查询和键，我们将每个头的维度 d^R_h 设置为 64。遵循 Dai et al. (2024)，我们将除第一层外的所有 FFN 替换为 MoE 层。每个 MoE 层由 2 个共享专家和 160 个路由专家组成，其中每个专家的中间隐藏维度为 1536。在路由专家中，每个 token 将激活 6 个专家。此外，低秩压缩和细粒度专家分割将影响层的输出规模。因此，在实践中，我们在压缩潜在向量后引入了额外的 RMS Norm 层，并在宽度瓶颈处（即压缩潜在向量和路由专家的中间隐藏状态）乘以额外的缩放因子，以确保训练稳定。在此配置下，DeepSeek-V2 包含 236B 总参数，其中每个 token 激活 21B 参数。

训练超参数。 我们采用 AdamW 优化器 (Loshchilov and Hutter, 2017)，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 (DeepSeek-AI, 2024)。初始阶段，学习率在前 2K 步内从 0 线性增加到最大值。随后，在训练约 60% 的 token 后，学习率乘以 0.316；在训练约 90% 的 token 后，再次乘以 0.316。最大学习率设置为 2.4×10^{-4}，梯度裁剪范数设置为 1.0。我们还使用了批次大小调度策略，在训练前 225B 个 token 时，批次大小从 2304 逐渐增加到 9216，然后在剩余的训练中保持 9216。我们将最大序列长度设置为 4K，并在 8.1T 个 token 上训练 DeepSeek-V2。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，路由专家将均匀部署在 8 个设备上（$D = 8$）。对于设备受限路由，每个 token 最多发送到 3 个设备（$M = 3$）。对于平衡损失，我们将 α_1 设置为 0.003，α_2 设置为 0.05，α_3 设置为 0.02。我们在训练期间采用 token 丢弃策略以加速训练，但在评估时不丢弃任何 token。

3.1.3. 基础设施

DeepSeek-V2 基于 HAI-LLM 框架 (High-flyer, 2023) 进行训练，该框架是由我们工程师内部开发的高效轻量级训练框架。它采用了 16 路零气泡流水线并行 (Qi et al., 2023)、8 路专家并行 (Lepikhin et al., 2021) 以及 ZeRO-1 数据并行 (Rajbhandari et al., 2020)。鉴于 DeepSeek-V2 激活的参数相对较少，且部分算子通过重计算来节省激活内存，因此可以在无需张量并行的情况下进行训练，从而降低通信开销。此外，为了进一步提高训练效率，我们将共享专家的计算与专家并行的 All-to-All 通信进行重叠。我们还为通信、路由算法以及跨不同专家的融合线性计算定制了更快的 CUDA 内核。此外，MLA 也基于改进版的 FlashAttention-2 (Dao, 2023) 进行了优化。

我们在配备 NVIDIA H800 GPU 的集群上进行了所有实验。H800 集群中的每个节点包含 8 块 GPU，节点内通过 NVLink 和 NVSwitch 连接。节点间利用 InfiniBand 互连以促进通信。

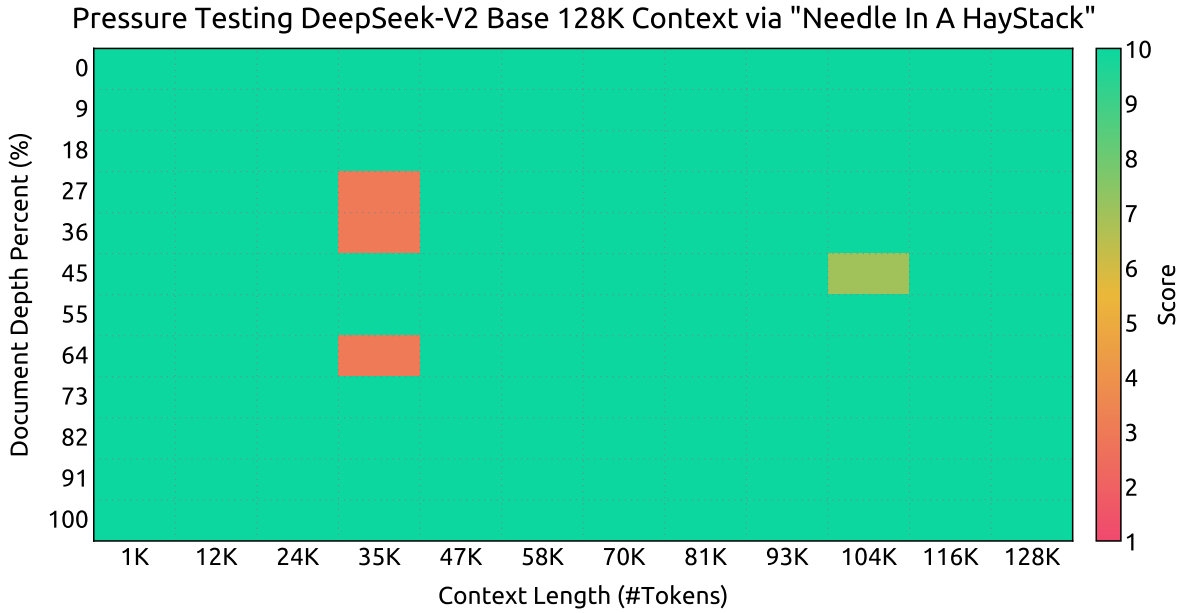


图 4 | 在“大海捞针”（NIAH）测试上的评估结果。DeepSeek-V2 在所有高达 128K 的上下文窗口长度上均表现良好。

3.1.4. 长上下文扩展

在 DeepSeek-V2 的初始预训练之后，我们采用 YaRN (Peng et al., 2023) 将默认上下文窗口长度从 4K 扩展到 128K。YaRN 专门应用于解耦的共享键 $\mathbf{k}_t^R$，因为它负责承载 RoPE (Su et al., 2024)。对于 YaRN，我们将缩放比例 s 设置为 40，α 设置为 1，β 设置为 32，目标最大上下文长度设置为 160K。在这些设置下，我们可以预期模型在 128K 的上下文长度下能良好响应。与原始 YaRN 略有不同，由于我们独特的注意力机制，我们调整了长度缩放因子以调节注意力熵。因子 $\sqrt{t}$ 的计算公式为 $\sqrt{t} = 0.0707 \ln s + 1$，旨在最小化困惑度。

我们额外训练模型 1000 步，序列长度为 32K，批次大小为 576 个序列。尽管训练仅在 32K 的序列长度下进行，但模型在 128K 的上下文长度评估中仍表现出稳健的性能。如图 4 所示，“大海捞针”（NIAH）测试的结果表明，DeepSeek-V2 在所有高达 128K 的上下文窗口长度上均表现良好。

3.2. 评估

3.2.1. 评估基准

DeepSeek-V2 在双语语料库上进行预训练，因此我们在一系列英文和中文基准上对其进行评估。我们的评估基于集成在 HAI-LLM 框架中的内部评估框架。包含的基准按类别列出如下，其中带下划线 的基准为中文：

多学科多项选择数据集包括 MMLU (Hendrycks et al., 2020)、C-Eval (Huang et al., 2023) 和 CMMLU (Li et al., 2023)。**语言理解与推理**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)。

闭卷问答数据集包括 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., 2019) 和 CMRC (Cui et al., 2019)。

指代消解数据集包括 WinoGrande Sakaguchi et al. (2019) 和 CLUEWSC (Xu et al., 2020)。

语言建模数据集包括 Pile (Gao et al., 2020)。

中文理解与文化数据集包括 CHID (Zheng et al., 2019) 和 CCPM (Li et al., 2021)。

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

代码数据集包括 HumanEval (Chen et al., 2021)、MBPP (Austin et al., 2021) 和 CRUXEval (Gu et al., 2024)。

标准化考试包括 AGIEval (Zhong et al., 2023)。需要注意的是，AGIEval 同时包含英文和中文子集。

遵循我们之前的工作 (DeepSeek-AI, 2024)，我们对包括 HellaSwag、PIQA、WinoGrande、RACE-Middle、RACE-High、MMLU、ARC-Easy、ARC-Challenge、CHID、C-Eval、CMMLU、C3 和 CCPM 在内的数据集采用基于困惑度的评估，并对 TriviaQA、NaturalQuestions、DROP、MATH、GSM8K、HumanEval、MBPP、CRUXEval、BBH、AGIEval、CLUEWSC、CMRC 和 CMath 采用基于生成的评估。此外，我们对 Pile-test 执行基于语言建模的评估，并使用 Bits-Per-Byte (BPB) 作为指标，以保证在不同分词器模型之间的公平比较。

为了直观地了解这些基准测试，我们在附录 G 中额外提供了每个基准的评估格式。

3.2.2. 评估结果

在表 2 中，我们将 DeepSeek-V2 与几个代表性的开源模型进行了对比，包括 DeepSeek 67B (DeepSeek-AI, 2024)（我们之前的发布版本）、Qwen1.5 72B (Bai et al., 2023)、LLaMA3 70B (AI@Meta, 2024) 和 Mixtral 8x22B (Mistral, 2024)。我们使用内部评估框架对所有这些模型进行评估，并确保它们共享相同的评估设置。总体而言，仅凭 21B 的激活参数，DeepSeek-V2 在几乎所有基准测试上都显著优于 DeepSeek 67B，并在开源模型中取得了顶尖性能。

此外，我们逐一将 DeepSeek-V2 与其开源同类模型进行了详细对比。(1) 与同样支持中英文的 Qwen1.5 72B 相比，DeepSeek-V2 在大多数英文、代码和数学基准测试上展现出压倒性优势。在中文基准测试方面，Qwen1.5 72B 在多学科多项选择题任务上表现更好，而 DeepSeek-V2 在其他任务上表现相当或更优。需要注意的是，在 CHID 基准测试中，Qwen1.5 72B 的分词器会在我们的评估框架中遇到错误，因此我们留空了 Qwen1.5 72B 的 CHID 分数。(2) 与 Mixtral

	基准测试 (指标)	提示数	DeepSeek 67B	Qwen1.5 72B	Mixtral 8x22B	LLaMA 3 70B	DeepSeek-V2
	架构	-	稠密	稠密	MoE	稠密	MoE
	激活参数量	-	67B	72B	39B	70B	21B
	总参数量	-	67B	72B	141B	70B	236B
英文	Pile-test (BPB)	-	0.642	0.637	0.623	**0.602**	0.606
	BBH (EM)	3-shot	68.7	59.9	78.9	**81.0**	78.9
	MMLU (Acc.)	5-shot	71.3	77.2	77.6	**78.9**	78.5
	DROP (F1)	3-shot	69.7	71.5	80.4	**82.5**	80.1
	ARC-Easy (Acc.)	25-shot	95.3	97.1	97.3	**97.9**	**97.6**
	ARC-Challenge (Acc.)	25-shot	86.4	92.8	91.2	**93.3**	92.4
	HellaSwag (Acc.)	10-shot	86.3	85.8	86.6	**87.9**	84.2
	PIQA (Acc.)	0-shot	83.6	83.3	83.6	**85.0**	83.7
	WinoGrande (Acc.)	5-shot	84.9	82.4	83.7	**85.7**	84.9
	RACE-Middle (Acc.)	5-shot	69.9	63.4	**73.3**	**73.3**	**73.1**
	RACE-High (Acc.)	5-shot	50.7	47.0	56.7	**57.9**	52.7
	TriviaQA (EM)	5-shot	78.9	73.1	**82.1**	81.6	79.9
	NaturalQuestions (EM)	5-shot	36.6	35.6	39.6	**40.2**	38.7
	AGIEval (Acc.)	0-shot	41.3	**64.4**	43.4	49.8	51.2
代码	HumanEval (Pass@1)	0-shot	45.1	43.9	**53.1**	48.2	48.8
	MBPP (Pass@1)	3-shot	57.4	53.6	64.2	**68.6**	66.6
	CRUXEval-I (Acc.)	2-shot	42.5	44.3	52.4	49.4	**52.8**
	CRUXEval-O (Acc.)	2-shot	41.0	42.3	52.8	**54.3**	49.8
数学	GSM8K (EM)	8-shot	63.4	77.9	80.3	**83.0**	79.2
	MATH (EM)	4-shot	18.7	41.4	42.5	42.2	**43.6**
	CMath (EM)	3-shot	63.0	77.8	72.3	73.9	**78.7**
中文	CLUEWSC (EM)	5-shot	81.0	80.5	77.5	78.3	**82.2**
	C-Eval (Acc.)	5-shot	66.1	**83.7**	59.6	67.5	81.7
	CMMLU (Acc.)	5-shot	70.8	**84.3**	60.0	69.3	**84.0**
	CMRC (EM)	1-shot	73.4	66.6	73.1	73.3	**77.5**
	C3 (Acc.)	0-shot	75.3	**78.2**	71.4	74.0	77.4
	CHID (Acc.)	0-shot	92.1	-	57.0	83.2	**92.7**
	CCPM (Acc.)	0-shot	88.5	88.1	61.0	68.1	**93.1**

表 2 | DeepSeek-V2 与其他代表性开源模型的对比。所有模型均在我们的内部框架中进行评估，并共享相同的评估设置。**粗体**表示最佳，下划线 表示次佳。分差小于 0.3 的分数视为处于同一水平。仅凭 21B 的激活参数，DeepSeek-V2 便在开源模型中取得了顶尖性能。

8x22B 相比，DeepSeek-V2 在英文性能上达到相当或更优的水平，TriviaQA、NaturalQuestions 和 HellaSwag 除外，这些基准与英文常识知识密切相关。值得注意的是，DeepSeek-V2 在 MMLU 上优于 Mixtral 8x22B。在代码和数学基准测试上，DeepSeek-V2 与 Mixtral 8x22B 表现出相当的性能。由于 Mixtral 8x22B 并未专门在中文数据上进行训练，其中文能力远落后于 DeepSeek-V2。(3) 与 LLaMA3 70B 相比，DeepSeek-V2 使用的英文 token 训练量不到其四分之一。因此，我们承认 DeepSeek-V2 在基础英文能力上与 LLaMA3 70B 仍存在轻微差距。然而，即使训练 token 和激活参数少得多，DeepSeek-V2 在代码和数学能力上仍与 LLaMA3 70B 相当。此外，作为双语语言模型，DeepSeek-V2 在中文基准测试上以压倒性优势超越 LLaMA3 70B。

最后，值得一提的是，某些先前的研究 (Hu et al., 2024) 在预训练阶段引入了 SFT 数据，而 DeepSeek-V2 在预训练期间从未接触过 SFT 数据。

3.2.3. 训练与推理效率

训练成本。 由于 DeepSeek-V2 为每个 token 激活的参数更少，且所需的 FLOPs 少于 DeepSeek 67B，理论上训练 DeepSeek-V2 将比训练 DeepSeek 67B 更经济。尽管训练 MoE 模型会引入额外的通信开销，但通过我们的算子和通信优化，DeepSeek-V2 的训练能够达到相对较高的模型 FLOPs 利用率（MFU）。在 H800 集群的实际训练中，每训练一万亿 token，DeepSeek 67B 需要 300.6K GPU 小时，而 DeepSeek-V2 仅需 172.8K GPU 小时，即稀疏的 DeepSeek-V2 相比稠密的 DeepSeek 67B 可节省 42.5% 的训练成本。

推理效率。 为了高效部署 DeepSeek-V2 提供服务，我们首先将其参数转换为 FP8 精度。此外，我们还对 DeepSeek-V2 执行 KV 缓存量化 (Hooper et al., 2024; Zhao et al., 2023)，将其 KV 缓存中的每个元素进一步压缩至平均 6 比特。得益于 MLA 和这些优化，实际部署的 DeepSeek-V2 所需的 KV 缓存显著少于 DeepSeek 67B，因此能够支持更大的批次大小。我们基于实际部署的 DeepSeek 67B 服务中的提示词和生成长度分布，评估了 DeepSeek-V2 的生成吞吐量。在配备 8 张 H800 GPU 的单个节点上，DeepSeek-V2 实现了超过每秒 5 万 token 的生成吞吐量，是 DeepSeek 67B 最大生成吞吐量的 5.76 倍。此外，DeepSeek-V2 的提示词输入吞吐量超过每秒 10 万 token。

4. 对齐

4.1. 监督微调

基于我们之前的研究 (DeepSeek-AI, 2024)，我们精心策划了指令微调数据集，包含 150 万条实例，其中 120 万条用于提升有用性，30 万条用于提升安全性。与初始版本相比，我们提升了数据质量，以减轻幻觉响应并增强写作能力。我们使用 2 个 epoch 对 DeepSeek-V2 进行微调，学习率设置为 5×10^{-6}。对于 DeepSeek-V2 Chat (SFT) 的评估，我们主要包含基于生成的基准测试，除了一些代表性的多项选择题任务（MMLU 和 ARC）。我们还对 DeepSeek-V2 Chat (SFT) 进行了指令遵循评估（IFEval）(Zhou et al., 2023)，使用提示词级别的宽松准确率作为指标。此外，我们采用 2023 年 9 月 1 日至 2024 年 4 月 1 日期间的 LiveCodeBench (Jain et al., 2024) 题目来评估聊天模型。除了标准基准测试外，我们还在包括 MT-Bench (Zheng et al., 2023)、AlpacaEval 2.0 (Dubois et al., 2024) 和 AlignBench (Liu et al., 2023) 在内的开放式对话基准上进一步评估了我们的模型。为了进行对比，我们还在我们的评估框架和设置下评估了 Qwen1.5 72B Chat、LLaMA-3-70B Instruct 和 Mistral-8x22B Instruct。至于 DeepSeek 67B Chat，我们直接引用之前发布版本中报告的评估结果。

4.2. 强化学习

为了进一步释放 DeepSeek-V2 的潜力并将其与人类偏好对齐，我们进行了强化学习（RL）以调整其偏好。

强化学习算法。 为了节省 RL 的训练成本，我们采用了组相对策略优化（GRPO）(Shao et al., 2024)，该方法摒弃了通常与策略模型大小相同的评论家模型，改为从组得分中估计基线。具体而言，对于每个问题 q，GRPO 从旧策略 $\pi_{\theta_{old}}$ 中采样一组输出 $\{o_1, o_2, \cdots, o_G\}$，然后通过最大化以下目标来优化策略模型 π_θ：

$$\mathcal{J}_{GRPO}(\theta) = \mathbb{E}[q \sim P(Q), \{o_i\}_{i=1}^{G} \sim \pi_{\theta_{old}}(O|q)]$$
$$\frac{1}{G}\sum_{i=1}^{G}\left(\min\left(\frac{\pi_\theta(o_i|q)}{\pi_{\theta_{old}}(o_i|q)}A_i, \text{clip}\left(\frac{\pi_\theta(o_i|q)}{\pi_{\theta_{old}}(o_i|q)}, 1-\varepsilon, 1+\varepsilon\right)A_i\right) - \beta\mathbb{D}_{KL}\left(\pi_\theta||\pi_{ref}\right)\right), \tag{32}$$

$$\mathbb{D}_{KL}\left(\pi_\theta||\pi_{ref}\right) = \frac{\pi_{ref}(o_i|q)}{\pi_\theta(o_i|q)} - \log\frac{\pi_{ref}(o_i|q)}{\pi_\theta(o_i|q)} - 1, \tag{33}$$

其中 ε 和 β 为超参数；A_i 为优势值(advantage)，通过计算每组输出对应的奖励集合 $\{r_1, r_2, \ldots, r_G\}$ 得到：

$$A_i = \frac{r_i - \text{mean}(\{r_1, r_2, \cdots, r_G\})}{std(\{r_1, r_2, \cdots, r_G\})}. \tag{34}$$

训练策略。 在初步实验中，我们发现针对推理数据（如代码和数学提示词）的强化学习训练表现出与通用数据训练截然不同的独特特性。例如，我们模型的数学和编程能力可以在更长的训练步数中持续提升。因此，我们采用两阶段强化学习训练策略，首先进行推理对齐，随后进行人类偏好对齐。在第一阶段推理对齐中，我们为代码和数学推理任务训练一个奖励模型 $RM_{reasoning}$，并利用 $RM_{reasoning}$ 的反馈来优化策略模型：

$$r_i = RM_{reasoning}(o_i). \tag{35}$$

在第二阶段人类偏好对齐中，我们采用多奖励框架，从有用性奖励模型 $RM_{helpful}$、安全性奖励模型 RM_{safety} 和基于规则的奖励模型 RM_{rule} 中获取奖励。响应 o_i 的最终奖励为

$$r_i = c_1 \cdot RM_{helpful}(o_i) + c_2 \cdot RM_{safety}(o_i) + c_3 \cdot RM_{rule}(o_i), \tag{36}$$

其中 c_1、c_2 和 c_3 为相应的系数。

为了获得在强化学习训练中起关键作用的可靠奖励模型，我们仔细收集偏好数据，并严格进行质量过滤和比例调整。我们基于编译器反馈获取代码偏好数据，基于真实标签（ground-truth labels）获取数学偏好数据。在奖励模型训练方面，我们使用 DeepSeek-V2 Chat (SFT) 初始化奖励模型，并采用逐点（point-wise）或成对（pair-wise）损失进行训练。在我们的实验中，我们观察到强化学习训练能够充分挖掘并激活模型的潜力，使其能够从可能的回复中选出正确且令人满意的答案。

训练效率优化。 在超大规模模型上进行强化学习训练对训练框架提出了极高的要求。它需要精细的工程优化来管理 GPU 显存和内存压力，同时保持较快的训练速度。为此，我们实施了以下

工程优化。(1) 首先，我们提出了一种混合引擎，分别为训练和推理采用不同的并行策略，以实现更高的 GPU 利用率。(2) 其次，我们利用支持大批次大小的 vLLM (Kwon et al., 2023) 作为推理后端，以加速推理速度。(3) 第三，我们精心设计了一种将模型卸载到 CPU 并重新加载回 GPU 的调度策略，在训练速度和内存消耗之间实现了近乎最优的平衡。

4.3. 评估结果

标准基准测试评估。首先，我们在标准基准测试上评估了 DeepSeek-V2 Chat (SFT) 和 DeepSeek-V2 Chat (RL)。值得注意的是，与基础版本相比，DeepSeek-V2 Chat (SFT) 在 GSM8K、MATH 和 HumanEval 评估中取得了显著提升。这一进步可归因于我们 SFT 数据的引入，其中包含了大量与数学和代码相关的内容。此外，DeepSeek-V2 Chat (RL) 进一步提升了在数学和代码基准测试上的性能。我们在附录 F 中展示了更多的代码和数学评估结果。

关于与其他模型的对比，我们首先将 DeepSeek-V2 Chat (SFT) 与 Qwen1.5 72B Chat 进行比较，发现 DeepSeek-V2 Chat (SFT) 在几乎所有的英语、数学和代码基准测试上均超越了 Qwen1.5 72B Chat。在中文基准测试上，DeepSeek-V2 Chat (SFT) 在多科目多项选择题任务上的得分略低于 Qwen1.5 72B Chat，这与其基础版本的表现一致。与最先进的开源 MoE 模型 Mixtral 8x22B Instruct 相比，DeepSeek-V2 Chat (SFT) 在除 NaturalQuestions 和 IFEval 之外的大多数基准测试上均表现出更好的性能。此外，与最先进的开源模型 LLaMA3 70B Chat 相比，DeepSeek-V2 Chat (SFT) 在代码和数学相关基准测试上表现出相似的性能。LLaMA3 70B Chat 在 MMLU 和 IFEval 上表现更好，而 DeepSeek-V2 Chat (SFT) 在中文任务上展现出更强的性能。最终，与 DeepSeek-V2 Chat (SFT) 相比，DeepSeek-V2 Chat (RL) 在数学和编程任务上均展现出进一步提升的性能。这些对比凸显了 DeepSeek-V2 Chat 在不同领域和语言中相较于其他语言模型的优势。

开放式生成评估。我们进一步在开放式对话基准测试上对我们的模型进行了额外评估。对于英语开放式对话生成，我们使用 MT-Bench 和 AlpacaEval 2.0 作为基准测试。表 4 中展示的评估结果表明，DeepSeek-V2 Chat (RL) 相较于 DeepSeek-V2 Chat (SFT) 具有显著的性能优势。这一结果展示了我们强化学习训练在实现更好对齐方面的有效性。与其他开源模型相比，DeepSeek-V2 Chat (RL) 在这两个基准测试上均优于 Mistral 8x22B Instruct 和 Qwen1.5 72B Chat。与 LLaMA3 70B Instruct 相比，DeepSeek-V2 Chat (RL) 在 MT-Bench 上展现出具有竞争力的性能，并在 AlpacaEval 2.0 上显著优于后者。这些结果凸显了 DeepSeek-V2 Chat (RL) 在生成高质量且上下文相关的回复方面的强大性能，特别是在基于指令的对话任务中。

此外，我们基于 AlignBench 评估了中文开放式生成能力。如表 5 所示，DeepSeek-V2 Chat (RL) 相较于 DeepSeek-V2 Chat (SFT) 具有轻微优势。值得注意的是，DeepSeek-V2 Chat (SFT) 以显著优势超越了所有开源中文模型。在中文推理和语言任务上，它均显著优于排名第二的开源模型 Qwen1.5 72B Chat。此外，DeepSeek-V2 Chat (SFT) 和 DeepSeek-V2 Chat (RL) 均优于 GPT-4-0613 和 ERNIEBot 4.0，巩固了我们的模型在支持中文的顶级大语言模型中的地位。具体而言，DeepSeek-V2 Chat (RL) 在中文语言理解方面表现出色，超越了包括 GPT-4-

	评测基准	提示数	DeepSeek 67B 对话	Qwen 1.5 72B 对话	LLaMA3 70B 指令微调	Mixtral 8x22B 指令微调	DeepSeek-V2 对话 (SFT)	DeepSeek-V2 对话 (RL)
	上下文长度	-	4K	32K	8K	64K	128K	128K
	模型架构	-	稠密	稠密	稠密	MoE	MoE	MoE
	激活参数量	-	67B	72B	70B	39B	21B	21B
	总参数量	-	67B	72B	70B	141B	236B	236B
英语	TriviaQA	5-shot	81.5	79.6	69.1	80.0	85.4	**86.7**
	NaturalQuestions	5-shot	47.0	46.9	44.6	**54.9**	51.9	53.4
	MMLU	5-shot	71.1	76.2	**80.3**	77.8	78.4	77.8
	ARC-Easy	25-shot	96.6	96.8	96.9	97.1	97.6	**98.1**
	ARC-Challenge	25-shot	88.9	91.7	**92.6**	90.0	**92.5**	**92.3**
	BBH	3-shot	71.7	65.9	80.1	78.4	**81.3**	79.7
	AGIEval	0-shot	46.4	62.8	56.6	41.4	**63.2**	61.4
	IFEval	0-shot	55.5	57.3	**79.7**	72.1	64.1	63.8
代码	HumanEval	0-shot	73.8	68.9	76.2	75.0	76.8	**81.1**
	MBPP	3-shot	61.4	52.2	69.8	64.4	70.4	**72.0**
	CRUXEval-I-COT	2-shot	49.1	51.4	61.1	59.4	59.5	**61.5**
	CRUXEval-O-COT	2-shot	50.9	56.5	**63.6**	**63.6**	60.7	63.0
	LiveCodeBench	0-shot	18.3	18.8	30.5	25.0	28.7	**32.5**
数学	GSM8K	8-shot	84.1	81.9	**93.2**	87.9	90.8	92.2
	MATH	4-shot	32.6	40.6	48.5	49.8	52.7	**53.9**
	CMath	0-shot	80.3	**82.8**	79.2	75.1	82.0	81.9
中文	CLUEWSC	5-shot	78.5	**90.1**	85.4	75.8	88.6	**89.9**
	C-Eval	5-shot	65.2	**82.2**	67.9	60.0	80.9	78.0
	CMMLU	5-shot	67.8	**82.9**	70.7	61.0	82.4	81.6

表 3 | DeepSeek-V2 Chat (SFT)、DeepSeek-V2 Chat (RL) 与其他代表性开源对话模型的对比。关于 TriviaQA 和 NaturalQuestions 数据集，值得注意的是，对话模型（如 LLaMA3 70B Instruct）可能不会严格遵守 few-shot 设置中通常指定的格式约束。因此，这可能导致在我们的评估框架中对某些模型的性能低估。

Model	MT-Bench	AlpacaEval 2.0
DeepSeek 67B Chat	8.35	16.6
Mistral 8x22B Instruct v0.1	8.66	30.9
Qwen1.5 72B Chat	8.61	36.6
LLaMA3 70B Instruct	**8.95**	34.4
DeepSeek-V2 Chat (SFT)	8.62	30.0
DeepSeek-V2 Chat (RL)	**8.97**	**38.9**

表 4 | 英语开放式对话评估结果。对于 AlpacaEval 2.0，我们使用长度控制胜率作为指标。

Turbo-1106-Preview 在内的所有模型。另一方面，DeepSeek-V2 Chat (RL) 的推理能力仍落后于 Erniebot-4.0 和 GPT-4s 等巨型模型。

4.4. 讨论

SFT 数据量。 关于是否需要大规模 SFT 语料的讨论一直是激烈争论的焦点。先前的研究 (Young et al., 2024; Zhou et al., 2024) 认为，少于 1 万个 SFT 数据实例就足以产生令人满意的结果。然而，在我们的实验中，我们观察到如果使用少于 1 万个实例，模型在 IFEval 基准测试上的性能

模型	总分	推理			语言						
		平均	**数学**	**逻辑**	**平均**	**基础**	**中文**	**开放**	**写作**	**角色**	**专业**
模型	总分	推理 总分	数学 计算	逻辑 推理	语言 总分	基本 任务	中文 理解	综合 问答	文本 写作	角色 扮演	专业 能力
GPT-4-1106-Preview	**8.01**	***7.73***	**7.80**	**7.66**	8.29	7.99	7.33	**8.61**	**8.67**	**8.47**	**8.65**
DeepSeek-V2 Chat (RL)	**7.91**	7.45	7.77	7.14	**8.36**	**8.10**	8.28	8.37	8.53	8.33	8.53
ERNIEBot-4.0-202404*（文心一言）	**7.89**	7.61	**7.81**	7.41	8.17	7.56	**8.53**	8.13	8.45	8.24	8.09
DeepSeek-V2 Chat (SFT)	**7.74**	7.30	7.34	7.26	8.17	8.04	8.26	8.13	8.00	8.10	8.49
GPT-4-0613	**7.53**	*7.47*	7.56	7.37	7.59	7.81	6.93	7.42	7.93	7.51	7.94
ERNIEBot-4.0-202312*（文心一言）	**7.36**	6.84	7.00	6.67	7.88	7.47	7.88	8.05	8.19	7.84	7.85
Moonshot-v1-32k-202404*（月之暗面）	**7.22**	6.42	6.41	6.43	8.02	7.82	7.58	8.00	8.22	8.19	8.29
Qwen1.5-72B-Chat*	**7.19**	6.45	6.58	6.31	7.93	7.38	7.77	8.15	8.02	8.05	8.24
DeepSeek-67B-Chat	**6.43**	*5.75*	5.71	5.79	7.11	7.12	6.52	7.58	7.20	6.91	7.37
ChatGLM-Turbo（智谱清言）	**6.24**	*5.00*	4.74	5.26	7.49	6.82	7.17	8.16	7.77	7.76	7.24
ERNIEBot-3.5（文心一言）	**6.14**	*5.15*	5.03	5.27	7.13	6.62	7.60	7.26	7.56	6.83	6.90
Yi-34B-Chat*	**6.12**	4.86	4.97	4.74	7.38	6.72	7.28	7.76	7.44	7.58	7.53
GPT-3.5-Turbo-0613	**6.08**	*5.35*	5.68	5.02	6.82	6.71	5.81	7.29	7.03	7.28	6.77
ChatGLM-Pro（智谱清言）	**5.83**	*4.65*	4.54	4.75	7.01	6.51	6.76	7.47	7.07	7.34	6.89
SparkDesk-V2（讯飞星火）	**5.74**	*4.73*	4.71	4.74	6.76	5.84	6.97	7.29	7.18	6.92	6.34
Qwen-14B-Chat	**5.72**	*4.81*	4.91	4.71	6.63	6.90	6.36	6.74	6.64	6.59	6.56
Baichuan2-13B-Chat	**5.25**	*3.92*	3.76	4.07	6.59	6.22	6.05	7.11	6.97	6.75	6.43
ChatGLM3-6B	**4.97**	*3.85*	3.55	4.14	6.10	5.75	5.29	6.71	6.83	6.28	5.73
Baichuan2-7B-Chat	**4.97**	*3.66*	3.56	3.75	6.28	5.81	5.50	7.13	6.84	6.53	5.84
InternLM-20B	**4.96**	*3.66*	3.39	3.92	6.26	5.96	5.50	7.18	6.19	6.49	6.22
Qwen-7B-Chat	**4.91**	*3.73*	3.62	3.83	6.09	6.40	5.74	6.26	6.31	6.19	5.66
ChatGLM2-6B	**4.48**	*3.39*	3.16	3.61	5.58	4.91	4.52	6.66	6.25	6.08	5.08
InternLM-Chat-7B	**3.65**	*2.56*	2.45	2.66	4.75	4.34	4.09	5.82	4.89	5.32	4.06
Chinese-LLaMA-2-7B-Chat	**3.57**	*2.68*	2.29	3.07	4.46	4.31	4.26	4.50	4.63	4.91	4.13
LLaMA-2-13B-Chinese-Chat	**3.35**	*2.47*	2.21	2.73	4.23	4.13	3.31	4.79	3.93	4.53	4.71

表 5 | 由 GPT-4-0613 评分的 AlignBench 排行榜。模型按总分降序排列。带有 * 标记的模型表示我们通过其 API 服务或开源权重模型进行评估，而非引用其原始论文中报告的结果。Erniebot-4.0 和 Moonshot 的后缀表示我们调用其 API 的时间戳。

会显著下降。一种可能的解释是，语言模型需要一定量的数据来培养特定技能。尽管所需的数据量可能会随着模型规模的增大而减少，但无法完全消除。我们的观察强调了为大型语言模型赋予所需能力而提供充足数据的关键必要性。此外，SFT 数据的质量也至关重要，特别是对于涉及写作或开放式问题的任务。

强化学习的对齐税。 在人类偏好对齐过程中，我们观察到在开放式生成基准测试上的性能显著提升，无论是 AI 评分还是人类评估员的评分均有所提高。然而，我们也注意到了“对齐税”（alignment tax）现象 (Ouyang et al., 2022)，即对齐过程可能会对 BBH 等某些标准基准测试的性能产生负面影响。为了缓解对齐税问题，在强化学习阶段，我们在数据处理和改进训练策略方面付出了大量努力，最终在标准基准测试和开放式基准测试的性能之间取得了可接受的权衡。探索如何在不对抗其通用性能的前提下使模型与人类偏好对齐，为未来的研究提供了一个有价值的方向。

在线强化学习。 在我们的偏好对齐实验中，我们发现在线方法显著优于离线方法。因此，我们投入了大量精力来实现一个用于对齐 DeepSeek-V2 的在线强化学习框架。关于在线或离线偏好对齐的结论可能因不同场景而异，我们将把对两者的更彻底比较和分析留待未来工作。

5. 结论、局限性与未来工作

在本文中，我们介绍了 DeepSeek-V2，这是一个支持 128K 上下文长度的大型 MoE 语言模型。除了强大的性能外，得益于其包含 MLA 和 DeepSeekMoE 的创新架构，该模型还具有训练经济高效和推理高效的特点。在实际应用中，与 DeepSeek 67B 相比，DeepSeek-V2 实现了显著更强的性能，同时节省了 42.5% 的训练成本，将 KV 缓存减少了 93.3%，并将最大生成吞吐量提升了 5.76 倍。评估结果进一步表明，仅使用 21B 激活参数，DeepSeek-V2 就在开源模型中达到了顶级性能，成为最强的开源 MoE 模型。

DeepSeek-V2 及其聊天版本与其他大语言模型一样，存在一些公认的局限性，包括预训练后缺乏持续的知识更新、可能生成非事实性信息（如未经核实的建议），以及产生幻觉的可能性。此外，由于我们的数据主要由中文和英文内容组成，我们的模型在其他语言上可能表现出有限的熟练度。在超出中英文的场景中，应谨慎使用。

DeepSeek 将以长期主义持续投入开源大模型的研发，旨在逐步逼近通用人工智能的目标。

- 在我们持续进行的探索中，我们致力于设计能够在保持经济高效的训练和推理成本的同时，进一步扩展 MoE 模型规模的方法。我们下一步的目标是在即将发布的版本中实现与 GPT-4 相当的性能。
-
- 我们的对齐团队持续致力于提升模型能力，旨在开发一款不仅对用户有益，而且诚实、安全，适用于全球用户的模型。我们的最终目标是使模型价值观与人类价值观保持一致，同时最大限度地减少对人类监督的依赖。通过优先考虑伦理因素并坚持负责任的发展理念，我们致力于为社会创造积极有益的影响。
- 目前，DeepSeek-V2 仅设计为支持文本模态。在未来的规划中，我们计划使模型支持多模态，从而提升其在更广泛场景中的通用性和实用性。

参考文献

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

J. Ainslie, J. Lee-Thorp, M. de Jong, Y. Zemlyanskiy, F. Lebrón, and S. Sanghai. Gqa: Training generalized multi-query transformer models from multi-head checkpoints. arXiv preprint arXiv:2305.13245, 2023.

Anthropic. Introducing Claude, 2023. URL https://www.anthropic.com/index/introducing-claude.

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.

J. Bai, S. Bai, Y. Chu, Z. Cui, K. Dang, X. Deng, Y. Fan, W. Ge, Y. Han, F. Huang, B. Hui, L. Ji, M. Li, J. Lin, R. Lin, D. Liu, G. Liu, C. Lu, K. Lu, J. Ma, R. Men, X. Ren, X. Ren, C. Tan, S. Tan, J. Tu, P. Wang, S. Wang, W. Wang, S. Wu, B. Xu, J. Xu, A. Yang, H. Yang, J. Yang, S. Yang, Y. Yao, B. Yu, H. Yuan, Z. Yuan, J. Zhang, X. Zhang, Y. Zhang, Z. Zhang, C. Zhou, J. Zhou, X. Zhou, and T. Zhu. Qwen technical report. arXiv preprint arXiv:2309.16609, 2023.

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.

T. Dao. FlashAttention-2: Faster attention with better parallelism and work partitioning, 2023.

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

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.

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.

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.

Google. Introducing gemini: our largest and most capable ai model, 2023. URL https://blog.google/technology/ai/google-gemini-ai/.

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. 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.

High-flyer. Hai-llm: 高效且轻量的大模型训练工具, 2023. URL https://www.high-flyer.cn/en/blog/hai-llm.

C. Hooper, S. Kim, H. Mohammadzadeh, M. W. Mahoney, Y. S. Shao, K. Keutzer, and A. Gholami. Kvquant: Towards 10 million context length LLM inference with KV cache quantization. CoRR, abs/2401.18079, 2024. URL https://doi.org/10.48550/arXiv.2401.18079.

S. Hu, Y. Tu, X. Han, C. He, G. Cui, X. Long, Z. Zheng, Y. Fang, Y. Huang, W. Zhao, et al. Minicpm: Unveiling the potential of small language models with scalable training strategies. arXiv preprint arXiv:2404.06395, 2024.

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.

N. Jain, K. Han, A. Gu, W.-D. 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. arXiv preprint arXiv:2403.07974, 2024.

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: Long Papers), pages 1601–1611, Vancouver, Canada, July 2017. Association for Computational Linguistics. doi: 10.18653/v1/P17-1147. URL https://aclanthology.org/P17-1147.

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.

W. Kwon, Z. Li, S. Zhuang, Y. Sheng, L. Zheng, C. H. Yu, J. E. Gonzalez, H. Zhang, and I. Stoica. Efficient memory management for large language model serving with pagedattention. In Proceedings of the ACM SIGOPS 29th Symposium on Operating Systems Principles, 2023.

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.

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

X. Liu, X. Lei, S. Wang, Y. Huang, Z. Feng, B. Wen, J. Cheng, P. Ke, Y. Xu, W. L. Tam, X. Zhang, L. Sun, H. Wang, J. Zhang, M. Huang, Y. Dong, and J. Tang. Alignbench: Benchmarking chinese alignment of large language models. CoRR, abs/2311.18743, 2023. doi: 10.48550/ARXIV.2311.18743. URL https://doi.org/10.48550/arXiv.2311.18743.

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

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.

OpenAI. Introducing ChatGPT, 2022. URL https://openai.com/blog/chatgpt.

OpenAI. GPT4 technical report. arXiv preprint arXiv:2303.08774, 2023.

L. Ouyang, J. Wu, X. Jiang, D. Almeida, C. Wainwright, P. Mishkin, C. Zhang, S. Agarwal, K. Slama, A. Ray, et al. Training language models to follow instructions with human feedback. Advances in neural information processing systems, 35:27730–27744, 2022.

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

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

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.

C. Riquelme, J. Puigcerver, B. Mustafa, M. Neumann, R. Jenatton, A. S. Pinto, D. Keysers, and N. Houlsby. Scaling vision with sparse mixture of experts. In Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021,

NeurIPS 2021, pages 8583–8595, 2021. URL https://proceedings.neurips.cc/paper/2021/hash/48237d9f2dea8c74c2a72126cf63d933-Abstract.html.

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. 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.

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, 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.

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.

J. Wei, Y. Tay, R. Bommasani, C. Raffel, B. Zoph, S. Borgeaud, D. Yogatama, M. Bosma, D. Zhou, D. Metzler, et al. Emergent abilities of large language models. arXiv preprint arXiv:2206.07682, 2022.

T. Wei, J. Luan, W. Liu, S. Dong, and B. Wang. Cmath: Can your language model pass chinese elementary school math test?, 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.

A. Young, B. Chen, C. Li, C. Huang, G. Zhang, G. Zhang, H. Li, J. Zhu, J. Chen, J. Chang, et al. Yi: Open foundation models by 01. ai. arXiv preprint arXiv:2403.04652, 2024.

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.

Y. Zhao, C. Lin, K. Zhu, Z. Ye, L. Chen, S. Zheng, L. Ceze, A. Krishnamurthy, T. Chen, and B. Kasikci. Atom: Low-bit quantization for efficient and accurate LLM serving. CoRR, abs/2310.19102, 2023. URL https://doi.org/10.48550/arXiv.2310.19102.

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.

L. Zheng, W.-L. Chiang, Y. Sheng, S. Zhuang, Z. Wu, Y. Zhuang, Z. Lin, Z. Li, D. Li, E. P. Xing, H. Zhang, J. E. Gonzalez, and I. Stoica. Judging llm-as-a-judge with mt-bench and chatbot arena, 2023.

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.

C. Zhou, P. Liu, P. Xu, S. Iyer, J. Sun, Y. Mao, X. Ma, A. Efrat, P. Yu, L. Yu, et al. Lima: Less is more for alignment. Advances in Neural Information Processing Systems, 36, 2024.

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. 贡献与致谢

研究与工程

Aixin Liu
Bingxuan Wang
Bo Liu
Chenggang Zhao
Chengqi Deng
Chong Ruan
Damai Dai
Daya Guo
Dejian Yang
Deli Chen
Erhang Li
Fangyun Lin
Fuli Luo
Guangbo Hao
Guanting Chen
Guowei Li
H. Zhang
Hanwei Xu
Hao Yang
Haowei Zhang
Honghui Ding
Huajian Xin
Huazuo Gao
Hui Qu
Jianzhong Guo
Jiashi Li
Jingyang Yuan
Junjie Qiu
Junxiao Song
Kai Dong
Kaige Gao
Kang Guan
Lean Wang
Lecong Zhang
Liang Zhao
Liyue Zhang
Mingchuan Zhang
Minghua Zhang
Minghui Tang
Panpan Huang
Peiyi Wang
Qihao Zhu
Qinyu Chen
Qiushi Du
Ruiqi Ge
Ruizhe Pan
Runxin Xu
Shanghao Lu
Shangyan Zhou
Shanhuang Chen
Shengfeng Ye
Shirong Ma
Shiyu Wang
Shuiping Yu
Shunfeng Zhou
Size Zheng
Tian Pei
Wangding Zeng
Wen Liu
Wenfeng Liang
Wenjun Gao
Wentao Zhang
Xiao Bi
Xiaohan Wang
Xiaodong Liu
Xiaokang Chen
Xiaotao Nie

Xin Liu
Xin Xie
Xingkai Yu
Xinyu Yang
Xuan Lu
Xuecheng Su
Y. Wu
Y.K. Li
Y.X. Wei
Yanhong Xu
Yao Li
Yao Zhao
Yaofeng Sun
Yaohui Wang
Yichao Zhang
Yiliang Xiong
Yilong Zhao
Ying He
Yishi Piao
Yixin Dong
Yixuan Tan
Yiyuan Liu
Yongji Wang
Yongqiang Guo
Yuduan Wang
Yuheng Zou
Yuxiang You
Yuxuan Liu
Z.Z. Ren
Zehui Ren
Zhangli Sha
Zhe Fu
Zhenda Xie
Zhewen Hao
Zhihong Shao
Zhuoshu Li
Zihan Wang
Zihui Gu
Zilin Li
Ziwei Xie

数据标注

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
Tian Yuan
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

商务与合规

Bin Wang
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
Ying Tang
Yukun Zha
Yuting Yan
Zhen Zhang
Zhiniu Wen

在每个角色类别中，作者按名字首字母顺序排列。特别感谢高华作和曾旺丁在 MLA 架构研究中做出的关键创新。此外，我们感谢苏建林在位置编码方面提供的有益讨论。我们感谢所有为 DeepSeek-V2 做出贡献但未在论文中提及的人员。DeepSeek 坚信，创新、新颖性和好奇心是通往通用人工智能（AGI）之路的关键。

B. DeepSeek-V2-Lite：一款配备 MLA 与 DeepSeekMoE 的 16B 模型

B.1. 模型描述

架构。 DeepSeek-V2-Lite 包含 27 层，隐藏层维度为 2048。该模型同样采用 MLA，并包含 16 个注意力头，每个头的维度为 128。其 KV 压缩维度为 512，但与 DeepSeek-V2 略有不同，它不对查询（queries）进行压缩。对于解耦的查询和键，其每个头的维度为 64。DeepSeek-V2-Lite 同样采用 DeepSeekMoE，除第一层外，所有前馈网络（FFN）均被替换为 MoE 层。每个 MoE 层由 2 个共享专家和 64 个路由专家组成，每个专家的中间隐藏维度为 1408。在路由专家中，每个 token 将激活 6 个专家。在此配置下，DeepSeek-V2-Lite 总参数量为 157 亿，其中每个 token 激活 24 亿参数。

	基准测试	DeepSeek 7B	DeepSeekMoE 16B	DeepSeek-V2-Lite
	架构	MHA+Dense	MHA+MoE	MLA+MoE
	上下文长度	4K	4K	32K
	激活参数量	6.9B	2.8B	2.4B
	总参数量	6.9B	16.4B	15.7B
	训练 Token 数	2T	2T	5.7T
英语	MMLU	48.2	45.0	**58.3**
	BBH	39.5	38.9	**44.1**
	TriviaQA	59.7	**64.8**	64.2
	NaturalQuestions	22.2	25.5	**26.0**
	ARC-Easy	67.9	68.1	**70.9**
	ARC-Challenge	48.1	49.8	**51.2**
	AGIEval	26.4	17.4	**33.2**
代码	HumanEval	26.2	26.8	**29.9**
	MBPP	39.0	39.2	**43.2**
数学	GSM8K	17.4	18.8	**41.1**
	MATH	3.3	4.3	**17.1**
	CMath	34.5	40.4	**58.4**
中文	CLUEWSC	73.1	72.1	**74.3**
	C-Eval	45.0	40.6	**60.3**
	CMMLU	47.2	42.5	**64.3**

表 6 | DeepSeek-V2-Lite、DeepSeekMoE 16B 与 DeepSeek 7B 的性能对比。

训练细节。 DeepSeek-V2-Lite 同样基于与 DeepSeek-V2 相同的预训练语料从头训练，该语料未受到任何 SFT 数据的污染。模型使用 AdamW 优化器，超参数设置为 $\beta_1 = 0.9$，$\beta_2 = 0.95$，以及 weight_decay = 0.1。学习率采用预热与阶梯衰减策略进行调度。初始阶段，学习率在前 2K 步内从 0 线性增加至最大值。随后，在训练约 80% 的 token 后，学习率乘以 0.316；在训练约 90% 的 token 后，再次乘以 0.316。最大学习率设置为 4.2×10^{-4}，梯度裁剪范数设置为 1.0。我们未对其采用批次大小调度策略，而是使用固定的 4608 个序列的批次大小进行训练。在预训练

期间，我们将最大序列长度设置为 4K，并使用 5.7T 个 token 训练 DeepSeek-V2-Lite。我们利用流水线并行将模型的不同层部署在不同设备上，但对于每一层，所有专家均部署在同一设备上。因此，我们仅采用较小的专家级平衡损失（$\alpha_1 = 0.001$），而未使用设备级平衡损失和通信平衡损失。预训练完成后，我们还对 DeepSeek-V2-Lite 进行了长上下文扩展和 SFT，得到了名为 DeepSeek-V2-Lite Chat 的对话模型。

	基准测试	DeepSeek 7B Chat	DeepSeekMoE 16B Chat	DeepSeek-V2-Lite Chat
	架构	MHA+Dense	MHA+MoE	MLA+MoE
	上下文长度	4K	4K	32K
	激活参数量	6.9B	2.8B	2.4B
	总参数量	6.9B	16.4B	15.7B
	训练 Token 数	2T	2T	5.7T
英语	MMLU	49.7	47.2	**55.7**
	BBH	43.1	42.2	**48.1**
	TriviaQA	59.5	63.3	**65.2**
	NaturalQuestions	32.7	35.1	**35.5**
	ARC-Easy	70.2	69.9	**74.3**
	ARC-Challenge	50.2	50.0	**51.5**
	AGIEval	17.6	19.7	**42.8**
代码	HumanEval	45.1	45.7	**57.3**
	MBPP	39.0	**46.2**	45.8
数学	GSM8K	62.6	62.2	**72.0**
	MATH	14.7	15.2	**27.9**
	CMath	66.4	67.9	**71.7**
中文	CLUEWSC	66.2	68.2	**80.0**
	C-Eval	44.7	40.0	**60.1**
	CMMLU	51.2	49.3	**62.5**

表 7 | DeepSeek-V2-Lite Chat、DeepSeekMoE 16B Chat 与 DeepSeek 7B Chat 的性能对比。

B.2. 性能评估

基座模型。 我们评估了 DeepSeek-V2-Lite 的性能，并在表 6 中将其与我们之前的小规模基座模型进行了对比。DeepSeek-V2-Lite 展现出压倒性的性能优势，尤其在推理、代码和数学方面。

对话模型。 我们评估了 DeepSeek-V2-Lite Chat 的性能，并在表 7 中将其与我们之前的小规模对话模型进行了对比。DeepSeek-V2-Lite 同样大幅优于我们之前的小规模对话模型。

C. MLA 的完整公式

为了展示 MLA 的完整计算过程，我们提供其完整公式如下：

$$\mathbf{c}_t^Q = W^{DQ}\mathbf{h}_t, \tag{37}$$

$$[\mathbf{q}_{t,1}^C; \mathbf{q}_{t,2}^C; ...; \mathbf{q}_{t,n_h}^C] = \mathbf{q}_t^C = W^{UQ}\mathbf{c}_t^Q, \tag{38}$$

$$[\mathbf{q}_{t,1}^R; \mathbf{q}_{t,2}^R; ...; \mathbf{q}_{t,n_h}^R] = \mathbf{q}_t^R = \text{RoPE}(W^{QR}\mathbf{c}_t^Q), \tag{39}$$

$$\mathbf{q}_{t,i} = [\mathbf{q}_{t,i}^C; \mathbf{q}_{t,i}^R], \tag{40}$$

$$\boxed{\mathbf{c}_t^{KV}} = W^{DKV}\mathbf{h}_t, \tag{41}$$

$$[\mathbf{k}_{t,1}^C; \mathbf{k}_{t,2}^C; ...; \mathbf{k}_{t,n_h}^C] = \mathbf{k}_t^C = W^{UK}\mathbf{c}_t^{KV}, \tag{42}$$

$$\boxed{\mathbf{k}_t^R} = \text{RoPE}(W^{KR}\mathbf{h}_t), \tag{43}$$

$$\mathbf{k}_{t,i} = [\mathbf{k}_{t,i}^C; \mathbf{k}_{t,i}^R], \tag{44}$$

$$[\mathbf{v}_{t,1}^C; \mathbf{v}_{t,2}^C; ...; \mathbf{v}_{t,n_h}^C] = \mathbf{v}_t^C = W^{UV}\mathbf{c}_t^{KV}, \tag{45}$$

$$\mathbf{o}_{t,i} = \sum_{j=1}^{t} \text{Softmax}_j(\frac{\mathbf{q}_{t,i}^T \mathbf{k}_{j,i}}{\sqrt{d_h + d_h^R}})\mathbf{v}_{j,i}^C, \tag{46}$$

$$\mathbf{u}_t = W^O[\mathbf{o}_{t,1}; \mathbf{o}_{t,2}; ...; \mathbf{o}_{t,n_h}], \tag{47}$$

其中，蓝色框出的向量需要在生成过程中进行缓存。在推理阶段，朴素公式需要从 $\mathbf{c}_t^{KV}$ 中恢复 $\mathbf{k}_t^C$ 和 $\mathbf{v}_t^C$ 以进行注意力计算。幸运的是，得益于矩阵乘法的结合律，我们可以将 W^{UK} 吸收进 W^{UQ}，将 W^{UV} 吸收进 W^O。因此，我们无需为每个查询单独计算键和值。通过此项优化，我们避免了在推理过程中重新计算 $\mathbf{k}_t^C$ 和 $\mathbf{v}_t^C$ 的计算开销。

D. 注意力机制的消融实验

D.1. MHA、GQA 与 MQA 的消融实验

表 8 展示了配备 MHA、GQA 和 MQA 的 7B 稠密模型在四个高难度基准测试上的评估结果。这三个模型均在 1.33T tokens 上进行训练，除注意力机制外，架构完全相同。此外，为确保公平比较，我们通过调整层数将它们的参数量对齐至约 7B。从表中可以看出，MHA 在这些基准测试上相较于 GQA 和 MQA 展现出显著优势。

D.2. MLA 与 MHA 的对比

表 9 展示了分别配备 MLA 和 MHA 的 MoE 模型在四个高难度基准测试上的评估结果。为得出可靠结论，我们在两个规模下对模型进行了训练与评估。两个小型 MoE 模型的总参数量约为 16B，我们在 1.33T tokens 上对其进行了训练。两个大型 MoE 模型的总参数量约为 250B，我们在 420B tokens 上对其进行了训练。此外，两个小型 MoE 模型之间以及两个大型 MoE 模型之间，除注意力机制外架构均相同。从表中可以看出，MLA 的性能优于 MHA。更重要的是，与

基准测试 (指标)	# 样本数	稠密 7B 配备 MQA	稠密 7B 配备 GQA (8 组)	稠密 7B 配备 MHA
# 参数量	-	7.1B	6.9B	6.9B
BBH (EM)	3-shot	33.2	35.6	**37.0**
MMLU (准确率)	5-shot	37.9	41.2	**45.2**
C-Eval (准确率)	5-shot	30.0	37.7	**42.9**
CMMLU (准确率)	5-shot	34.6	38.4	**43.5**

表 8 | 配备 MHA、GQA 和 MQA 的 7B 稠密模型对比。MHA 在高难度基准测试上相较于 GQA 和 MQA 展现出显著优势。

MHA 相比，MLA 所需的 KV 缓存量显著减少（小型 MoE 模型为 14%，大型 MoE 模型为 4%）。

基准测试 (指标)	# 样本数	小型 MoE 配备 MHA	小型 MoE 配备 MLA	大型 MoE 配备 MHA	大型 MoE 配备 MLA
# 激活参数量	-	2.5B	2.4B	25.0B	21.5B
# 总参数量	-	15.8B	15.7B	250.8B	247.4B
每 Token KV 缓存量 (# 元素)	-	110.6K	15.6K	860.2K	34.6K
BBH (EM)	3-shot	37.9	**39.0**	46.6	**50.7**
MMLU (准确率)	5-shot	48.7	**50.0**	57.5	**59.0**
C-Eval (准确率)	5-shot	**51.6**	50.9	57.9	**59.2**
CMMLU (准确率)	5-shot	52.3	**53.4**	60.7	**62.5**

表 9 | MLA 与 MHA 在高难度基准测试上的对比。DeepSeek-V2 的性能优于 MHA，但所需的 KV 缓存量显著更少。

E. 关于预训练数据去偏的讨论

在预训练数据准备阶段，我们识别并过滤了具有争议性的内容（例如受区域文化影响的价值观），以避免模型在这些争议性话题上表现出不必要的主体偏见。因此，我们观察到 DeepSeek-V2 在与特定区域文化密切相关的测试集上表现略差。例如，在 MMLU 评估中，尽管与 Mixtral 8x22B 等竞品相比，DeepSeek-V2 在大多数测试集上取得了相当或更优的性能，但在主要与美国价值观相关的 Humanity-Moral 子集上仍略显落后。

此外，我们对该子集进行了人工分析。三名受过良好教育的人工标注员对 MMLU Humanity-Moral 子集中的 420 个道德情境进行了独立标注。随后，我们计算了他们的标注结果与真实标签之间的一致性。如表 10 所示，三名人工标注员与真实标签之间的一致性较低。因此，我们将 DeepSeek-V2 在这些价值观敏感测试集上的异常表现归因于我们对预训练语料库的去偏工作。

F. 数学与代码的额外评估

评估采用了包含数千道中文数学题的 SC-Math6 数据集。DeepSeek-V2 Chat (RL) 的表现优于所有中文大语言模型，包括开源和闭源模型。

一致性	真实标签	标注员 1	标注员 2	标注员 3
真实标签	100.0%	66.7%	59.8%	42.1%
标注员 1	66.7%	100.0%	57.9%	69.0%
标注员 2	59.8%	57.9%	100.0%	65.5%
标注员 3	42.1%	69.0%	65.5%	100.0%

表 10 | 三名具备良好教育背景的人类标注员对 MMLU Humanity-Moral 子集中的 420 个道德场景进行了独立标注，在该子集上，DeepSeek-V2 及其竞争性模型表现出性能不一致。三名标注员与真实标签之间的一致性较低。这表明，受特定区域文化的影响，Humanity-Moral 子集的答案可能存在争议。

模型名称	R 等级	综合得分	推理步骤得分	总体准确率得分
GPT-4-1106-Preview	**5**	90.71	91.65	89.77
GPT-4	**5**	88.40	89.10	87.71
DeepSeek-V2 Chat (RL)	**5**	83.35	85.73	**84.54**
Ernie-bot 4.0	**5**	85.60	86.82	84.38
Qwen-110B-Chat	**5**	83.25	84.93	84.09
GLM-4	**5**	84.24	85.72	82.77
Xinghuo 3.5	**5**	83.73	85.37	82.09
Qwen-72B-Chat	**4**	78.42	80.07	79.25
ChatGLM-Turbo	**4**	57.70	60.32	55.09
GPT-3.5-Turbo	**4**	57.05	59.61	54.50
Qwen-14B-Chat	**4**	53.12	55.99	50.26
ChatGLM3-6B	**3**	40.90	44.20	37.60
Xinghuo 3.0	**3**	40.08	45.27	34.89
Baichuan2-13B-Chat	**3**	39.40	42.63	36.18
Ernie-3.5-turbo	**2**	25.19	27.70	22.67
Chinese-Alpaca2-13B	**2**	20.55	22.52	18.58

表 11 | SC-Math6 模型推理等级。“R Level” 代表推理等级，“Comp. Score” 代表综合得分，“Reas. Steps Score” 代表推理步骤得分，“OvrAcc Score” 代表总体准确率得分。

我们在图 5 中进一步展示了 HumanEval 和 LiveCodeBench 上的更多结果，其中 LiveCodeBench 的题目选自 2023 年 9 月 1 日至 2024 年 4 月 1 日期间。如图所示，DeepSeek-V2 Chat (RL) 在 LiveCodeBench 上表现出显著的熟练度，其 Pass@1 得分甚至超越了一些巨型模型。这一表现凸显了 DeepSeek-V2 Chat (RL) 在处理实时编程任务方面的强大能力。

G. 评估格式

我们在表 12-37 中分别展示了各基准测试的评估格式。

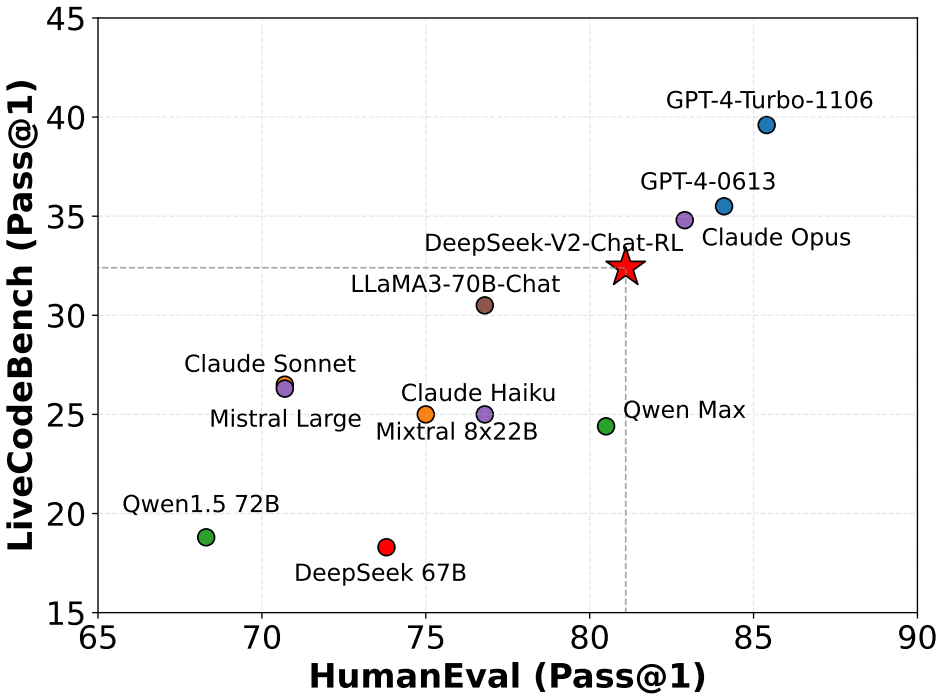


图 5 | HumanEval 和 LiveCodeBench 上的评估结果。LiveCodeBench 的题目选自 2023 年 9 月 1 日至 2024 年 4 月 1 日期间。

提示词
以下是一道中国高考生物选择题，请选择正确的答案。
问题：下列有关高尔基体、线粒体和叶绿体的叙述, 正确的是选项：(A) 三者都存在于蓝藻中 (B) 三者都含有 DNA (C) 三者都是 ATP 合成的场所 (D) 三者的膜结构中都含有蛋白质
答案：从 A 到 D, 我们应选择

表 12 | AGIEval 示例。

提示词
问题：圆柱形容器中的样品呈圆柱形且体积固定。该样品的物质状态 _
A. 必定是固体
B. 可能是固体或液体
C. 必定是液体
D. 可能是液体或气体
答案：B

问题：声速通常在 _ 中最大
A. 固体中最大，液体中最小
B. 固体中最大，气体中最小
C. 气体中最大，液体中最小
D. 气体中最大，固体中最小
答案：B

问题：当油和水混合在一起时，它们会形成 _
A. 气体
B. 固体
C. 化合物
D. 悬浊液
答案：D

问题：白天温度为 3°C 时，一容器液态水被放置在室外。夜间室外温度降至 -2°C。这种温度变化最可能导致水 _
A. 凝结
B. 蒸发
C. 保持液态
D. 变成固体
答案：

表 13 | ARC 示例。

提示词
评估随机布尔表达式的结果。

问：not ((not not True)) 的结果是
答：让我们逐步思考。
请记住：(i) 括号内的表达式总是优先计算，(ii) 运算优先级从高到低依次为“not”（非）、“and”（与）、“or”（或）。我们首先将表达式”Z” 简化如下：”Z = not ((not not True)) = not ((A))”，其中”A = not not True”。计算 A：A = not not True = not (not True) = not False = True。代入 A，得到：Z = not ((A)) = not ((True)) = not True = False。因此答案是 False。

问：True and False and not True and True 的结果是
答：让我们逐步思考。
请记住：(i) 括号内的表达式总是优先计算，(ii) 运算优先级从高到低依次为“not”（非）、“and”（与）、“or”（或）。我们首先将表达式”Z” 简化如下：”Z = True and False and not True and True = A and B”，其中”A = True and False”，”B = not True and True”。计算 A：A = True and False = False。计算 B：B = not True and True = not (True and True) = not (True) = False。代入 A 和 B，得到：Z = A and B = False and False = False。因此答案是 False。

问：not not (not (False)) 的结果是
答：让我们逐步思考。
请记住：(i) 括号内的表达式总是优先计算，(ii) 运算优先级从高到低依次为“not”（非）、“and”（与）、“or”（或）。我们首先将表达式”Z” 简化如下：”Z = not not (not (False)) = not not (A)”，其中”A = not (False)”。计算 A：A = not (False) = not False = True。代入 A，得到：Z = not not (A) = not not (True) = not not False = True。因此答案是 True。

问：False and False and False or not False 的结果是
答：让我们逐步思考。

表 14 | BBH 示例。

提示词
以下是中国关于教育学考试的单项选择题，请选出其中的正确答案。

根据我国心理学家冯忠良教授的学习分类，培养学生品德要通过 _____。
A. 知识的学习
B. 技能的学习
C. 行为规范的学习
D. 态度的学习
答案：C

开设跨学科课程或建立跨学科专业体现了高等教育课程发展的 _____。
A. 综合化趋势
B. 多样化趋势
C. 人文化趋势
D. 科学化趋势
答案：A

心智技能的特点有 _____。
A. 物质性、外显性、简缩性
B. 观念性、内潜性、简缩性
C. 物质性、外显性、展开性
D. 观念性、内潜性、展开性
答案：B

下列关于大学生的情绪与理智关系的说法中正确的是 _____。
A. 能冷静控制自己情绪
B. 感情用事，难以用理智控制情绪
C. 遇事能坚持自己正确认识
D. 已发展到不为小事而发怒和怄气
答案：B

在学完一篇逻辑结构严密的课文以后，勾画出课文的论点论据的逻辑关系图以帮助理解和记忆。这种学习方法属于 _____。
A. 精细加工策略
B. 组织策略
C. 复述策略
D. 做笔记策略
答案：B

有学者强调，教育要根据一个民族固有的特征来定，这种观点体现了 _____
A. 生产力对教育的影响和制约
B. 政治制度对教育的影响和制约
C. 文化对教育的影响和制约
D. 经济制度对教育的影响和制约
答案：

选项
- A
- B
- C
- D

表 15 | C-Eval 示例。

提示词
女：这些药怎么吃?
男：一天三次，一次两片。

请根据上文回答问题：

他们在哪儿?
答案：

选项
- 商店
- 饭店
- 医院
- 教室

表 16 | C3 示例。

提示词
以下是将某句古诗文翻译而成的现代表述：春天已至，万物复苏，春风如一位美丽而又心灵手巧的姑娘，迈着纤纤细步款款而来，她挥舞剪刀，尽情地展示那高超的女工技巧，她先裁出了柳叶，随着柳条袅袅依依地舞蹈，又裁出杏叶，桃叶。
该翻译所对应的古诗文是：

选项
- 春风骋巧如翦刀
- 剪裁无巧似春风
- 风吹怨恨快如刀
- 春风欲擅秋风巧

表 17 | CCPM 示例。

提示词

Q: 某小学在“献爱心–为汶川地震区捐款”活动中，六年级五个班共捐款 8000 元，其中一班捐款 1500 元，二班比一班多捐款 200 元，三班捐款 1600 元，四班与五班捐款数之比是 3：5．四班捐款多少元?
A: 一班捐款 1500 元, 而二班比一班多捐 200 元, 所以二班捐款 1500+200=1700 元，又知道六年级五个班一共捐款 8000 元，所以四班和五班捐款之和 = 一共捐款 - 一班和二班和三班捐款之和，即 8000-1500-1700-1600=3200 元，而题目说四班与五班捐款数之比是 3：5，则四班捐款了 3200/(3+5)*3=1200 元。所以答案是：1200。

Q: 小俊在东西大道上跑步，若规定向东为正。他先向东跑了 800 米，然后又跑了一段之后，他位于出发点西边 100 米处，小俊第二段跑了多少米?
A: 小俊第二段跑完后位于出发点西边，所以第二段应该是向西跑，第二段跑的长度-第一段跑的长度 =100，第二段跑了 100+800=900 米。所以答案是：900。

Q: A 车和 B 车同时从甲、乙两地相向开出，经过 5 小时相遇．然后，它们又各自按原速原方向继续行驶 3 小时，这时 A 车离乙地还有 135 千米，B 车离甲地还有 165 千米．甲、乙两地相距多少千米?
A: 假设 A 车的速度为 x 千米每小时，B 车的速度为 y 千米每小时，根据而 A、B 相遇时 A 车行驶了 5 小时，A 车行驶 3 小时后离乙地还有 135 千米，B 车行驶 3 小时后距离甲地还有 165 千米，可以得到甲乙两地相距 =5x+5y=135+8x=165+8y, 变换得到：10(x+y)=300+8(x+y), 于是 x+y=150, 甲乙两地相距 5(x+y)=750 千米。所以答案是：750。

Q: 在一个底面半径为 10 厘米的圆柱形容器内，倒入 10 厘米深的水，然后将一个底面直径 4 厘米，高 6 厘米的圆锥形铅锤放入水中，容器中水面上升多少厘米?
A:

表 18 | CMATH 示例。

提示词
以下是关于解剖学的单项选择题，请直接给出正确答案的选项。

题目：壁胸膜的分部不包括
A. 肋胸膜
B. 肺胸膜
C. 膈胸膜
D. 胸膜顶
答案是：B

题目：属于蝶骨上的结构为
A. 垂体窝
B. 棘孔
C. 破裂孔
D. 视神经管
答案是：B

题目：属于右心房的结构是
A. 肉柱
B. 室上嵴
C. 乳头肌
D. 梳状肌
答案是：D

题目：咽的分部
A. 咽隐窝
B. 口咽部
C. 鼻咽部
D. 喉咽部
答案是：C

题目：舌下神经核位于
A. 间脑
B. 延髓
C. 中脑
D. 脑桥
答案是：B

题目：从脑干背侧出脑的脑神经是
A. 副神经
B. 三叉神经
C. 舌下神经
D. 滑车神经
答案是：

选项
- A
- B
- C
- D

表 19 | CMMLU 示例。

提示词
文章：英雄广场（Heldenplatz）是奥地利首都维也纳的一个广场。在此曾发生许多重要事件—最著名的是 1938 年希特勒在此宣告德奥合并。英雄广场是霍夫堡皇宫的外部广场，兴建于皇帝弗朗茨 · 约瑟夫一世统治时期，是没有完全建成的所谓“帝国广场”（Kaiserforum）的一部分。其东北部是霍夫堡皇宫的 Leopoldinian Tract，东南方是新霍夫堡，西南方的内环路，将其与“城门外”（Äußeres Burgtor）隔开。西北部没有任何建筑物，可以很好地眺望内环路、国会大厦、市政厅，以及城堡剧院。广场上有 2 尊军事领袖的骑马像：欧根亲王和卡尔大公。

根据上文回答下面的问题。
问题：英雄广场是哪个皇宫的外部广场?
答案：霍夫堡皇宫
问题：广场上有哪两位军事领袖的骑马像?
答案：

表 20 | CMRC2018 示例。

提示词
段落：该市的中位年龄为 22.1 岁。10.1% 的居民年龄在 18 岁以下；56.2% 的居民年龄在 18 至 24 岁之间；16.1% 的居民年龄在 25 至 44 岁之间；10.5% 的居民年龄在 45 至 64 岁之间；7% 的居民年龄在 65 岁或以上。该市的人口性别构成为 64.3% 男性和 35.7% 女性。

请根据上述段落回答以下问题，如需计算请仔细计算。
问：不在 25 至 44 岁之间的人口占比是多少?
答：答案类型为数字。因此根据上述段落，答案为 83.9。

问：不在 25 至 44 岁之间的人口占比是多少?
答：答案类型为数字。因此根据上述段落，答案为

表 21 | DROP 示例。

提示词
中新网 12 月 7 日电综合外媒 6 日报道, 在美国得克萨斯州, 负责治疗新冠肺炎患者的医生约瑟夫 · 瓦隆 (Joseph Varon) 已连续上班超 260 天, 每天只睡不超过 2 小时。瓦隆日前接受采访时呼吁, 美国民众应遵从防疫规定, 一线的医护人员“已

选项
- 神清气爽”。
- 诡计多端”。
- 精疲力竭”。
- 分工合作”。
- 寅吃卯粮”。
- 土豪劣绅”。
- 芸芸众生”。

表 22 | CHID 示例。

提示词
胡雪岩离船登岸，坐轿进城，等王有龄到家，他接着也到了他那里，脸上是掩抑不住的笑容，王有龄夫妇都觉得奇怪，问他什么事这么高兴。
上面的句子中的”他”指的是
胡雪岩

渐渐地，汤中凝结出一团团块状物，将它们捞起放进盆里冷却，肥皂便出现在世上了。
上面的句子中的”它们”指的是
块状物

“她序上明明引着 JulesTellier 的比喻，说有个生脱发病的人去理发，那剃头的对他说不用剪发，等不了几天，头毛压儿全掉光了；大部分现代文学也同样的不值批评。这比喻还算俏皮。”
上面的句子中的”他”指的是
生脱发病的人

在洛伦佐大街的尽头处，矗立着著名的圣三一大教堂。它有着巨大的穹顶，还有明亮的彩色玻璃窗，上面描绘着《旧约》和《新约》的场景。
上面的句子中的”它”指的是
圣三一大教堂

他伯父还有许多女弟子，大半是富商财主的外室；这些财翁白天忙着赚钱，怕小公馆里的情妇长日无聊，要不安分，常常叫她们学点玩艺儿消遣。
上面的句子中的”她们”指的是
情妇

赵雨又拿出了一个杯子，我们热情地请老王入座，我边给他倒酒边问：1962 年的哪次记得吗？“
上面的句子中的”他”指的是

表 23 | CLUEWSC 示例。

提示词
问：Max 割草需要 40 分钟。如果施肥所需时间是割草的两倍，那么他割草和施肥总共需要多长时间?
答：让我们一步步思考。Max 施肥需要 2 * 40 分钟 = 80 分钟。总共，Max 割草和施肥需要 80 分钟 + 40 分钟 = 120 分钟。答案是 120。

问：贝果每个售价 $2.25，或者一打（12 个）售价 $24。如果一次买一打，每个贝果能节省多少美分?
答：让我们一步步思考。每个贝果售价 2.25*100=225 美分。按批量购买，每个是 24/12=2 美元。即每个 2*100=200 美分。225-200=25，每个贝果节省 25 美分。答案是 25。

问：Tim 今年 5 岁。他的表兄 Rommel 的年龄是他的三倍。他的另一个表妹 Jenny 比 Rommel 大 2 岁。Tim 比 Jenny 小多少岁?
答：让我们一步步思考。Rommel 的年龄是 5 x 3 = 15 岁。Jenny 的年龄是 15 + 2 = 17 岁。因此，Tim 比 Jenny 小 17 - 5 = 12 岁。答案是 12。

问：学校有 14 名男生和 10 名女生。如果有 4 名男生和 3 名女生退学，还剩下多少名男生和女生?
答：让我们一步步思考。剩下 14 名男生 - 4 名男生 = 10 名男生。剩下 10 名女生 - 3 名女生 = 7 名女生。总共剩下 10 名男生 + 7 名女生 = 17 名学生。答案是 17。

问：建造一个鸟屋需要 7 块木板和 20 颗钉子。如果 1 颗钉子售价 0.05 美元，1 块木板售价 3 美元，那么建造 4 个鸟屋的成本是多少美元?
答：让我们一步步思考。一个鸟屋的木板成本是 7 * 3 = 21 美元。钉子的成本是 20 * 0.05 = 1 美元。因此建造一个鸟屋需要 21 + 1 = 22 美元。建造 4 个鸟屋的成本是 4 * 22 = 88 美元。答案是 88。

问：Danny 带了 3 个西瓜去家庭野餐。他把每个西瓜切成 10 片。他的妹妹带了 1 个西瓜去野餐，她把西瓜切成 15 片。野餐时总共有多少片西瓜?
答：让我们一步步思考。Danny 的西瓜有 3 * 10 = 30 片。他妹妹的西瓜有 1 * 15 = 15 片。总共有 30 + 15 = 45 片西瓜。答案是 45。

问：Angela 是纽约的一名自行车快递员。她需要投递的包裹数量是餐食数量的 8 倍。如果她总共需要投递 27 份餐食和包裹，她投递了多少份餐食?
答：让我们一步步思考。设 Angela 投递的包裹数为 p，餐食数为 m。已知 p + m = 27 且 p = 8m。将第二个方程代入第一个方程，得到 8m + m = 27。合并同类项得 9m = 27。两边同时除以 9，得 m = 3。答案是 3。

问：Cori 今天 3 岁。5 年后，她的年龄将是她阿姨年龄的三分之一。她阿姨今天多少岁?
答：让我们一步步思考。5 年后，Cori 将是 3 + 5 = 8 岁。5 年后，Cori 的阿姨将是 8 x 3 = 24 岁。今天，她阿姨是 24 - 5 = 19 岁。答案是 19。

问：Indras 的名字有 6 个字母。她妹妹的名字比 Indras 名字字母数的一半多 4 个。Indras 和她妹妹的名字总共有多少个字母?
答：让我们一步步思考。

表 24 | GSM8K 示例。

提示词
弹钢琴：一名男子坐在钢琴前。他

选项
- 正用双手和脸弹奏钢琴。
- 开始在钢琴上弹奏 Timbaland 的歌曲。
- 弹奏得很慢，并停下来打响指。
- 正在弹奏他面前的一首曲子。

表 25 | HellaSwag 示例。

提示词

```
def starts_one_ends(n):
"""
给定一个正整数 n，返回以 1 开头或结尾的 n 位正整数的个数。
"""
```

表 26 | HumanEval 示例。

提示词
问题：
求表达式 $\frac{\sqrt{x-2}}{\sqrt{5-x}}$.} 的定义域。

解答：
每个根号内的表达式必须非负。
因此，$x-2 \ge 0$，即 $x\ge2$，且 $5 - x \ge 0$，即 $x \le 5$。
此外，分母不能为零，所以 $5-x>0$，即 $x<5$。
因此，该表达式的定义域为 $\boxed{[2,5)}$。
最终答案：最终答案是 $[2,5)$。希望它是正确的。

问题：
若 $\det \mathbf{A} = 2$ 且 $\det \mathbf{B} = 12,$ 求 $\det (\mathbf{A} \mathbf{B}).$

解答：
我们有 $\det (\mathbf{A} \mathbf{B}) = (\det \mathbf{A})(\det \mathbf{B}) = (2)(12) = \boxed{24}.$
最终答案：最终答案是 24。希望它是正确的。

问题：
Terrell 通常举起两个 20 磅的哑铃 12 次。如果他改用两个 15 磅的哑铃，为了举起相同的总重量，Terrell 必须举起多少次？

解答：
如果 Terrell 举起两个 20 磅的哑铃 12 次，他举起的总重量为 $2\cdot 12\cdot20=480$ 磅。如果他改用两个 15 磅的哑铃举起 n 次，他将举起的总重量为 $2\cdot15\cdot n=30n$ 磅。令其等于 480 磅，我们可以解出 n：
\begin{align*}
30n&=480\\
\Rightarrow\qquad n&=480/30=\boxed{16}
\end{align*}
最终答案：最终答案是 16。希望它是正确的。

问题：
若方程组

\begin{align*}
6x-4y&=a,\\
6y-9x &=b.
\end{align*} 有解 (x, y)，其中 x 和 y 均不为零，求 $\frac{a}{b},$（假设 b 不为零）。

解答：
若我们将第一个方程乘以 $-\frac{3}{2}$，可得

$$6y-9x=-\frac{3}{2}a.$$ 由于我们还知道 $6y-9x=b$，因此有

$$-\frac{3}{2}a=b\Rightarrow\frac{a}{b}=\boxed{-\frac{2}{3}}.$$
最终答案：最终答案是 $-\frac{2}{3}$。希望它是正确的。

问题：计算 $\log_21$。

解答：

表 27 | MATH 示例。

提示词
你是一名专业的 Python 程序员，你的任务是：编写一个函数，从给定的两个元组列表中找出相似元素。你的代码应通过以下测试：

```
assert similar_elements((3, 4, 5, 6),(5, 7, 4, 10)) == (4, 5)
assert similar_elements((1, 2, 3, 4),(5, 4, 3, 7)) == (3, 4)
assert similar_elements((11, 12, 14, 13),(17, 15, 14, 13)) == (13, 14)
[BEGIN]
def similar_elements(test_tup1, test_tup2):
res = tuple(set(test_tup1) & set(test_tup2))
return (res)
[DONE]
```

你是一名专业的 Python 程序员，你的任务是：编写一个 Python 函数来识别非质数。你的代码应通过以下测试：

```
assert is_not_prime(2) == False
assert is_not_prime(10) == True
assert is_not_prime(35) == True
[BEGIN]
import math
def is_not_prime(n):
result = False
for i in range(2,int(math.sqrt(n)) + 1):
if n % i == 0:
result = True
return result
[DONE]
```

你是一名专业的 Python 程序员，你的任务是：编写一个函数，使用堆队列算法从给定的数字列表中找出最大的整数。你的代码应通过以下测试：

```
assert heap_queue_largest( [25, 35, 22, 85, 14, 65, 75, 22, 58],3)==[85, 75, 65]
assert heap_queue_largest( [25, 35, 22, 85, 14, 65, 75, 22, 58],2)==[85, 75]
assert heap_queue_largest( [25, 35, 22, 85, 14, 65, 75, 22, 58],5)==[85, 75, 65,
58, 35]
[BEGIN]
import heapq as hq
def heap_queue_largest(nums,n):
largest_nums = hq.nlargest(n, nums)
return largest_nums
[DONE]
```

你是一名专业的 Python 程序员，你的任务是：编写一个函数，返回一个数字的所有因数之和。你的代码应通过以下测试：

```
assert sum_div(8)==7
assert sum_div(12)==16
assert sum_div(7)==1
[BEGIN]
```

表 28 | MBPP 示例。

提示
以下是关于各类知识的单项选择题（附答案）。

标准汽车有多少个车轴?
A. 一个
B. 两个
C. 四个
D. 八个
答案：B

摇滚传奇乐队 Cheap Trick 1979 年现场专辑的标题中提到了哪个地方?
A. 布达佩斯
B. 武道馆
C. 不丹
D. 英国
答案：B

历史上赢得 NBA 扣篮大赛的最矮球员是谁?
A. 安东尼 · “斯普德” · 韦伯
B. 迈克尔 · “空中飞人” · 乔丹
C. 泰隆 · “马格西” · 博格斯
D. 朱利叶斯 · “J 博士” · 欧文
答案：A

光合作用产生什么?
A. 氢气
B. 尼龙
C. 氧气
D. 光
答案：C

以下哪首歌是摇滚乐队 The Police 的十大热门单曲?
A. 《Radio Ga-Ga》
B. 《Ob-la-di Ob-la-da》
C. 《De Do Do Do De Da Da Da》
D. 《In-a-Gadda-Da-Vida》
答案：C

三傻大闹宝莱坞（The Three Stooges）中哪一位与其他成员没有亲属关系?
A. Moe
B. Larry
C. Curly
D. Shemp
答案：

选项
- A
- B
- C
- D

表 29 | MMLU 示例。

提示
回答以下问题：
问：2022 年世界杯的主办国是哪个?
答：卡塔尔
问：首届女子世界杯的冠军是谁?
答：美国
问：《迈阿密风云》（Miami Vice）何时停播?
答：1989 年
问：歌曲《Shout to the Lord》的创作者是谁?
答：Darlene Zschech
问：谁被扔进了狮子坑?
答：但以理
问：名字 Habib 的含义是什么?
答：

表 30 | NaturalQuestions 示例。

提示
一位女士注意到自己每年秋天都会感到抑郁，并想知道原因。一位朋友建议她，也许季节从温暖转为寒冷时发生的某些变化对她产生了影响。当被要求举例说明这些变化时，朋友提到了

选项
- 花朵绽放
- 草地变黄
- 树木生长
- 花蕾盛开

表 31 | OpenBookQA 示例。

提示
为了更容易按下位于机器下方的垃圾处理器复位按钮，

选项
- 在橱柜地板上放一面墙镜
- 在垃圾处理器下方手持一面手镜

表 32 | PIQA 示例。

提示
文章：
阅读文章时，如果你能理清作者是如何组织观点的，你会更好地理解并记住它。有时，作者通过提出问题然后回答它们来组织观点。例如，如果文章是关于土拨鼠的，作者脑海中的一组问题可能是：
土拨鼠长什么样？
土拨鼠住在哪里？
它们吃什么？……
在文章中，作者可能会回答这些问题。
有时作者会在文章中直接写出她的问题。这些问题会给你提示。它们告诉你作者接下来要写什么。通常作者脑海中有一个问题，但她没有写出来给你看。你必须自己推断出她的问题。下面是一篇供你练习此方法的阅读样本。
蚯蚓
你知道有多少种蚯蚓吗？世界上大约有 1800 种！它们可以是棕色、紫色或绿色。它们可以小到 3 厘米长，也可以大到 3 米长。
观察蚯蚓的最佳时间是夜晚，尤其是凉爽潮湿的夜晚。那时它们会从洞穴里出来觅食。蚯蚓不喜欢晒太阳。这是因为它们通过皮肤呼吸，如果皮肤太干就无法呼吸。如果雨下得很大，蚯蚓必须从土里出来，因为它们无法在被水淹没的洞穴中呼吸。多么危险的生活！
蚯蚓没有眼睛，那它们如何知道天黑了呢？它们的皮肤上有对光线敏感的特殊部位。这些斑点能分辨是亮还是暗。如果你在晚上用手电筒照蚯蚓，它会迅速钻入地下。
蚯蚓也没有耳朵，但它们可以通过感受地面的震动来“听”。如果你想像蚯蚓一样听，就趴在地上，把手指塞进耳朵里。然后让朋友在你附近跺脚。这就是蚯蚓感知附近鸟类和人类行走，以及鼹鼠挖掘的方式。
蚯蚓很有用。农民和园丁喜欢土地里有大量蚯蚓，因为蚯蚓在挖掘时有助于改良土壤。这种挖掘使土壤保持疏松和透气。一年内，蚯蚓可以在一个足球场大小的区域堆积多达 23,000 公斤的粪土。

问：阅读《蚯蚓》的目的是什么？

答：将作者的想法付诸实践。

问：以下哪个问题在文章中无法找到答案？

答：为什么人类能像蚯蚓一样听？

问：根据本文，如何更好地理解《蚯蚓》？

答：阅读时尝试推断出作者脑海中的所有问题。

问：这篇文章的最佳标题是什么？

答：

选项
- 一种有助于理解的方法
- 一种练习新想法的方法
- 一种学习成为明智作家的方法
- 一种更清楚了解蚯蚓的方法

表 33 | RACE 示例。

提示
回答以下问题：
问："Jayhawker" 一词用于指代来自美国某个州的反奴隶制民兵组织，该组织与密苏里州的亲奴隶制派系发生冲突。这个州是哪个，有时被称为 "Jayhawk 州"?
答：堪萨斯州
问：哪位瑞典 DJ 和唱片制作人在 2013 年凭借《Wake Me Up》获得英国单曲榜冠军?
答：Tim Bergling
问：谢菲尔德哈勒姆（Sheffield Hallam）的国会议员是谁?
答：Nick Clegg
问：轰动全国的田纳西州诉约翰 · 托马斯 · 斯科普斯案（The State of Tennessee v. John Thomas Scopes）于 1925 年 7 月 21 日结案，陪审团裁定斯科普斯先生因教授什么罪名成立?
答：物种生存（进化论）
问：哪部卡通系列片中有名为 Little My 的角色?
答：姆明（Muumi）
问："哪位英国模特，以其短发中性风格著称，原名 Lesley Hornby，1966 年 16 岁时被 Nigel Davies 发掘，当时体重仅 6 英石（41 公斤，91 磅），并凭借 Mary Quant 为其打造的高级时尚摩登造型成为 '66 年度面孔'?"
答：

表 34 | TriviaQA 示例。

前缀
- 所以 Monica
- 所以 Jessica

补全
避免吃胡萝卜来保护视力，因为 Emily 需要良好的视力，而 Monica 不需要。

表 35 | WinoGrande 示例。注意，WinoGrande 有多个前缀但只有一个补全文本，我们选择使补全文本困惑度最低的预测前缀。

提示
你将获得一个函数 f 和一个输出，格式为 f(??) == output。请找到任意一个输入，使得在该输入上执行 f 能得到给定输出。答案可能不唯一，但你只需输出一个。在 [ANSWER] 和 [/ANSWER] 标签中，用能产生该输出的一个输入完成断言。

[PYTHON]
def f(my_list):
count = 0
for i in my_list:
if len(i) % 2 == 0:
count += 1
return count
assert f(??) == 3
[/PYTHON]
[ANSWER]
assert f([”mq”, ”px”, ”zy”]) == 3
[/ANSWER]

[PYTHON]
def f(s1, s2):
return s1 + s2
assert f(??) == ”banana”
[/PYTHON]
[ANSWER]
assert f(”ba”, ”nana”) == ”banana”
[/ANSWER]

[PYTHON]
def f(a, b, c):
result = {}
for d in a, b, c:
result.update(dict.fromkeys(d))
return result
assert f(??) == {1: None, 2: None}
[/PYTHON]
[ANSWER]

表 36 | CRUXEval-I 示例。

提示
给定一个 Python 函数及一个包含该函数输入的断言。请使用一个字面量（不含未化简表达式或函数调用）补全该断言，该字面量应为在给定输入上执行所提供代码的输出，即使该函数不正确或不完整。请勿输出任何额外信息。请参照示例，在 [ANSWER] 和 [/ANSWER] 标签中提供包含正确输出的完整断言。

[PYTHON]
def f(n):
return n
assert f(17) == ??
[/PYTHON]
[ANSWER]
assert f(17) == 17
[/ANSWER]

[PYTHON]
def f(s):
return s + ”a”
assert f(”x9j”) == ??
[/PYTHON]
[ANSWER]
assert f(”x9j”) == ”x9ja”
[/ANSWER]

[PYTHON]
def f(nums):
output = []
for n in nums:
output.append((nums.count(n), n))
output.sort(reverse=True)
return output
assert f([1, 1, 3, 1, 3, 1]) == ??
[/PYTHON]
[ANSWER]

表 37 | CRUXEval-O 示例。