

## DeepSeek-V2: 强大、经济、高效的专家混合语言模型

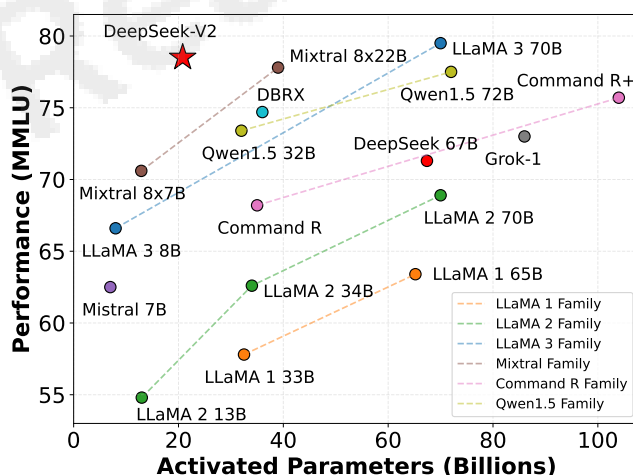
DeepSeek-AI

research@deepseek.com

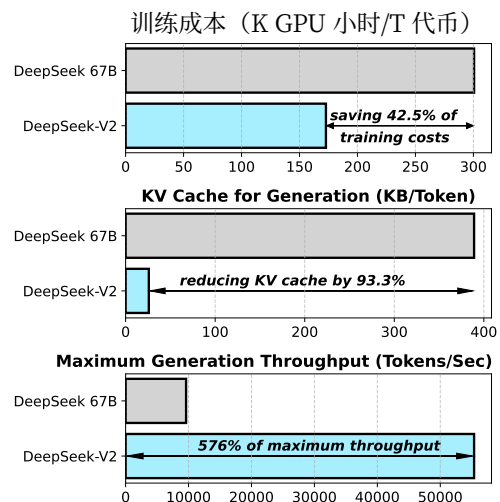
### Abstract

我们提出了 DeepSeek-V2，一种强大的专家混合 (MoE) 语言模型，其特点是经济的训练和高效的推理。它总共包括236B个参数，其中每个令牌激活21B个参数，并支持128K令牌的上下文长度。DeepSeek-V2采用多头潜在注意力 (MLA) 和DeepSeekMoE等创新架构。MLA通过将Key-Value (KV) 缓存显著压缩为潜在向量来保证高效推理，而DeepSeekMoE则通过稀疏计算以经济成本训练强大的模型。与 DeepSeek 67B 相比，DeepSeek-V2 性能显著增强，同时节省了 42.5% 的训练成本，减少了 93.3% 的 KV 缓存，最大生成吞吐量提升至 5.76 倍。我们在由 8.1T 代币组成的高质量多源语料库上对 DeepSeek-V2 进行预训练，并进一步进行监督微调 (SFT) 和强化学习 (RL) 以充分释放其潜力。评估结果表明，即使只有21B个激活参数，DeepSeek-V2及其聊天版本仍然达到了开源模型中顶级的性能。该模型

检查点位于 <https://github.com/deepseek-ai/DeepSeek-V2>。



(a)



(b)

图1| (a) 不同开源模型中的 MMLU 精度与激活参数。(b) DeepSeek 67B (密集) 和 DeepSeek-V2 的训练成本和推理效率。

# Contents

1 简介	4
2 架构	6
2.1 多头潜在注意力：提高推理效率。 . . . . .	6
2.1.1 Preliminaries: Standard Multi-Head Attention . . . . .	6
2.1.2 低秩键值联合压缩。 . . . . .	7
2.1.3 解耦旋转位置嵌入。 . . . . .	8
2.1.4 Comparison of Key-Value Cache . . . . .	8
2.2 DeepSeekMoE：以经济的成本训练强大的模型。 . . . . .	9
2.2.1 Basic Architecture . . . . .	9
2.2.2 Device-Limited Routing . . . . .	9
2.2.3 Auxiliary Loss for Load Balance . . . . .	10
2.2.4 Token-Dropping Strategy . . . . .	11
3 Pre-Training	11
3.1 Experimental Setups . . . . .	11
3.1.1 Data Construction . . . . .	11
3.1.2 Hyper-Parameters . . . . .	12
3.1.3 Infrastructures . . . . .	12
3.1.4 Long Context Extension . . . . .	13
3.2 Evaluations . . . . .	13
3.2.1 Evaluation Benchmarks . . . . .	13
3.2.2 Evaluation Results . . . . .	14
3.2.3 训练和推理效率。 . . . . .	
4 对齐	16
4.1 Supervised Fine-Tuning . . . . .	16
4.2 Reinforcement Learning . . . . .	17
4.3 Evaluation Results . . . . .	18
4.4 Discussion . . . . .	20
5 结论、局限性和未来的工作	21
A 贡献和致谢	27
B DeepSeek-V2-Lite：配备 MLA 和 DeepSeekMoE 的 16B 型号	29

B.1 Model Description . . . . .	29
B.2 Performance Evaluation . . . . .	30
C MLA 完整公式	31
D 注意力机制的消融	31
D.1 Ablation of MHA, GQA, and MQA . . . . .	31
D.2 Comparison Between MLA and MHA . . . . .	31
E 关于预训练数据去偏的讨论	32
F 对数学和代码的附加评估	32
G 评估格式	33

## 一、简介

在过去的几年里，大型语言模型（LLM）（Anthropic, 2023; Google, 2023; OpenAI, 2022、2023）经历了快速发展，让人们一睹通用人工智能（AGI）的曙光。一般来说，法学硕士的智力往往会随着参数数量的增加而提高，从而使其能够在各种任务中展现出新兴的能力（Wei et al., 2022）。然而，这种改进是以更大的训练计算资源和推理吞吐量的潜在下降为代价的。这些限制提出了重大挑战，阻碍了法学硕士的广泛采用和利用。为了解决这个问题，我们引入了 DeepSeek-V2，这是一种强大的开源专家混合 (MoE) 语言模型，其特点是通过创新的 Transformer 架构进行经济的训练和高效的推理。它总共配备了 236B 个参数，其中每个令牌激活 21B 个参数，并支持 128K 令牌的上下文长度。

我们使用我们提出的多头潜在注意力（MLA）和 DeepSeekMoE 优化 Transformer 框架（Vaswani 等人，2017）内的注意力模块和前馈网络（FFN）。

(1) 在注意力机制的背景下，多头注意力（MHA）的键值（KV）缓存（Vaswani et al., 2017）对 LLM 的推理效率造成了重大障碍。人们已经探索了各种方法来解决这个问题，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，这些方法在尝试减少 KV 缓存时常常会损害性能。为了实现两全其美，我们引入了 MLA，一种配备低秩键值联合压缩的注意力机制。从经验来看，MLA 比 MHA 取得了更优越的性能，同时大幅减少了推理过程中的 KV 缓存，从而提高了推理效率。(2) 对于前馈网络（FFN），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，以提高专家专业化的潜力。与 GShard（Lepikhin 等人，2021）等传统 MoE 架构相比，DeepSeekMoE 架构展示了巨大的优势，使我们能够以经济的成本训练强大的模型。当我们在训练过程中采用专家并行性时，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 具有强大的性能（图 1(a)）、经济的训练成本和高效的推理吞吐量（图 1(b)），同时。

我们构建了一个由 8.1T 个 token 组成的高质量、多源预训练语料库。与 DeepSeek 67B（我们之前的版本）（DeepSeek-AI, 2024）中使用的语料库相比，该语料库的数据量更大，尤其是中国数据，数据质量更高。我们首先在完整的预训练语料库上预训练 DeepSeek-V2。然后，我们收集 150 万个会话，其中涵盖数学、代码、写作、推理、安全等各个领域，以便为 DeepSeek-V2 聊天（SFT）执行监督微调（SFT）。最后，我们遵循 DeepSeekMath（Shao et al., 2024）采用组相对策略优化（GRPO）进一步使模型与人类偏好保持一致，并生成 DeepSeek-V2 Chat（RL）。

我们在多种英文和中文基准上对 DeepSeek-V2 进行了评估，并将其与代表性的开源模型进行了比较。评估结果表明，即使只有 21B 个激活参数，DeepSeek-V2 仍然达到了开源模型中顶级的性能，成为最强的开源 MoE 语言模型。图 1(a) 突出表明，在 MMLU 上，DeepSeek-V2 仅用少量的激活参数就实现了顶级性能。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，减少了 93.3% 的 KV 缓存，并将最大生成吞吐量提升至 5.76 倍。我们还评估了 DeepSeek-V2 Chat（SFT）和

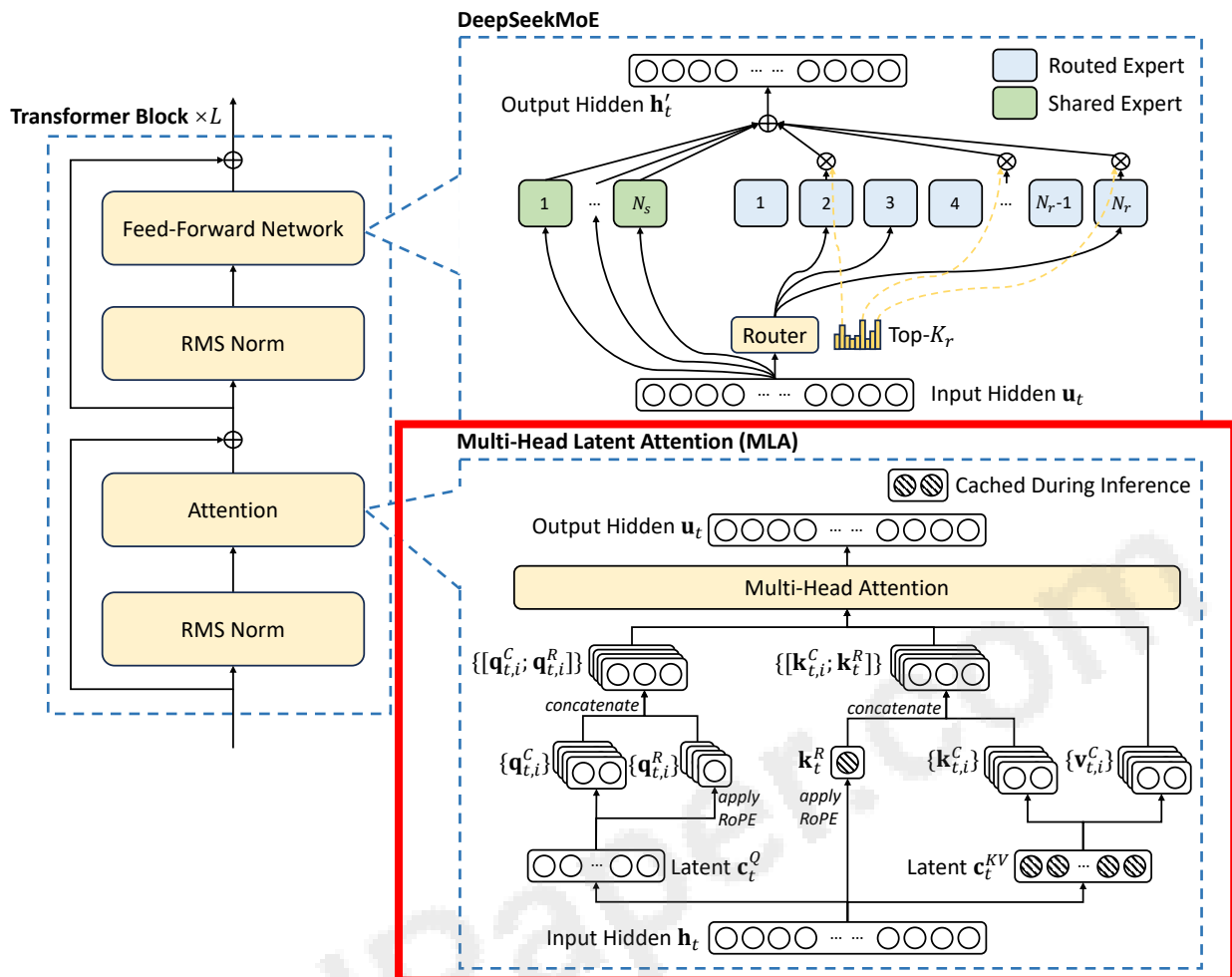


图2| DeepSeek-V2 的架构图。MLA 通过显著减少生成的 KV 缓存来确保高效推理，而 DeepSeekMoE 则可以通过稀疏架构以经济的成本训练强模型。

基于开放式基准的 DeepSeek-V2 Chat (RL)。值得注意的是，DeepSeek-V2 Chat (RL) 在 AlpacaEval 2.0 上获得了 38.9 的长度控制胜率 (Dubois 等人, 2024)，在 MT-Bench 上获得了 8.97 的总分 (Zheng 等人, 2023)，在 AlignBench 上获得了 7.91 的总分 (刘等人, 2023)。英语开放式对话评估表明，DeepSeek-V2 Chat (RL) 在开源聊天模型中具有顶级性能。此外，AlignBench 上的评估表明，在中文中，DeepSeek-V2 Chat (RL) 的表现优于所有开源模型，甚至击败了大多数闭源模型。

为了方便 MLA 和 DeepSeekMoE 的进一步研究和开发，我们还为开源社区发布了配备 MLA 和 DeepSeekMoE 的较小模型 DeepSeek-V2-Lite。它总共有 15.7B 个参数，其中每个令牌激活 2.4B 个参数。关于 DeepSeek-V2-Lite 的详细描述可以在附录 B 中找到。

在本文的其余部分，我们首先详细描述 DeepSeek-V2 的模型架构（第 2 节）。随后，我们介绍了我们的预训练工作，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和效率的评估（第 3 节）。在此之后，我们展示了我们的一致努力，包括监督微调 (SFT)、强化

学习 (RL)、评估结果和其他讨论 (第 4 节)。最后, 我们总结结论, 思考 DeepSeek-V2 当前的局限性, 并概述我们未来的工作 (第 5 节)。

## 2. 架构

总的来说, DeepSeek-V2 仍然采用 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, 我们设计并采用创新的架构。为了引起注意, 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, 从而支持高效的推理。对于 FFN, 我们采用 DeepSeekMoE 架构 (Dai et al., 2024), 一种高性能的 MoE architecture, 能够以经济的成本训练强大的模型。图 2 展示了 DeepSeek-V2 的架构图, 本节我们将介绍 MLA 和 DeepSeekMoE 的详细信息。对于其他微小细节 (例如, FFN 中的层归一化和激活函数), 除非特别说明, 否则 DeepSeek-V2 遵循 DeepSeek 67B (DeepSeek-AI, 2024) 的设置。

### 2.1. 多头潜在注意力: 提高推理效率

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, 其大量的键值 (KV) 缓存将成为限制推理效率的瓶颈。为了减少 KV 缓存, 提出了多查询注意 (MQA) (Shazeer, 2019) 和分组查询注意 (GQA) (Ainslie 等人, 2023)。它们需要较小量级的 KV 缓存, 但它们的性能不匹配 MHA (我们在附录 D.1 中提供了 MHA、GQA 和 MQA 的消融)。

对于 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)。MLA 配备低秩键值联合压缩, 比 MHA 具有更好的性能, 但需要的 KV 缓存量要少得多。我们在下面介绍它的架构, 并在附录 D.2 中提供 MLA 和 MHA 之间的比较。

#### 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。令  $d$  为嵌入维度,  $n_h$  为注意力头的数量,  $d_h$  为每个头的维度,  $\mathbf{h}_t \in \mathbb{R}^d$  为注意力层第  $t$  标记的注意力输入。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  $\mathbf{q}_t, \mathbf{k}_t, \mathbf{v}_t \in \mathbb{R}^{d_h n_h}$ :

$$\mathbf{q}_t = W^Q \mathbf{h}_t, \quad (1)$$

$$\mathbf{k}_t = W^K \mathbf{h}_t, \quad (2)$$

$$\mathbf{v}_t = W^V \mathbf{h}_t, \quad (3)$$

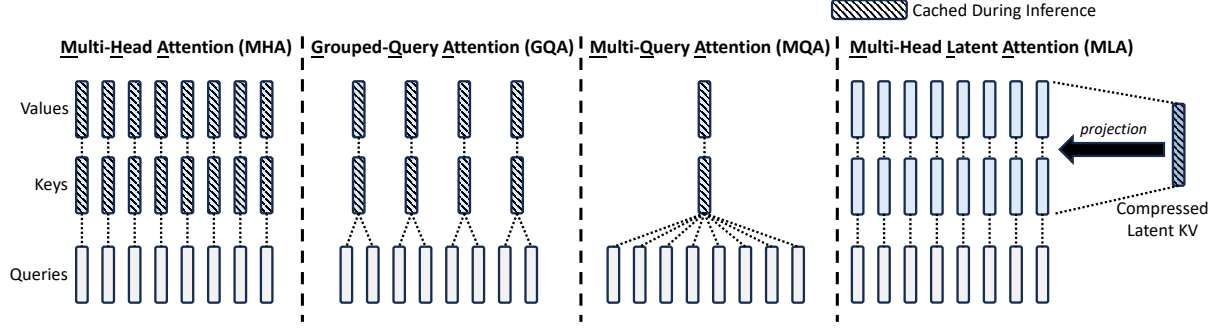


图3 |多头注意力（MHA）、分组查询注意力（GQA）、多查询注意力（MQA）和多头潜在注意力（MLA）的简化说明。通过将键和值联合压缩为潜在向量，MLA 显著减少了推理过程中的 KVcache。

然后， $q_t$ ， $k_t$ ， $v_t$  将被切片为  $n h$  头进行多头注意力计算：

$$[q_{t,1}; q_{t,2}; \dots; q_{t,n_h}] = q_t, \quad (4)$$

$$[k_{t,1}; k_{t,2}; \dots; k_{t,n_h}] = k_t, \quad (5)$$

$$[v_{t,1}; v_{t,2}; \dots; v_{t,n_h}] = v_t, \quad (6)$$

$$o_{t,i} = \sum_{j=1}^t \text{Softmax}_j \left( \frac{q_{t,i}^T k_{j,i}}{\sqrt{d_h}} \right) v_{j,i}, \quad (7)$$

$$u_t = W^O [o_{t,1}; o_{t,2}; \dots; o_{t,n_h}], \quad (8)$$

其中  $q_{t,i}$ ， $k_{t,i}$ ， $v_{t,i} \in \mathbb{R}^{d_h}$  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，需要缓存所有键和值以加速推理，因此 MHA 需要为每个 token 缓存  $2n h d h l$  元素。在模型部署中，这种沉重的 KV 缓存是一个很大的瓶颈，限制了最大批量大小和序列长度。

### 2.1.2. 低秩键值联合压缩

MLA 的核心是对 key 和 value 进行低秩联合压缩，以减少 KV 缓存：

$$c_t^{KV} = W^{DKV} h_t, \quad (9)$$

$$k_t^C = W^{UK} c_t^{KV}, \quad (10)$$

$$v_t^C = W^{UV} c_t^{KV}, \quad (11)$$

$c_t^{KV}$  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；和  $W^{UK}, W^{UV} \in \mathbb{R}^{d_h n h \times d_c}$  分别是键和值的上投影矩阵。在推断期间，仅 MLA 对缓存  $c_t^{KV}$ ，因此其 KV 缓存仅具有  $d d$  元素，其中  $l$  表示层的数量。加上推断，因为  $W$  可以吸收到  $W$  中，并且可以吸收。被  $W^O$  吸收，我们甚至不需要计算键和值来引起注意。图3直观地说明了 MLA 中的 KV 联合压缩如何减少 KV 缓存。

此外，为了减少训练期间的激活记忆，我们还执行

对查询进行低等级压缩，即使它不能减少 KV 缓存：

$$\mathbf{c}_t^Q = W^{DQ} \mathbf{h}_t, \quad (12)$$

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

其中  $\mathbf{c}_t^Q \in \mathbb{R}^{d'c}$  是查询的压缩潜在向量； $d'c (\ll d h \text{ for queries; } d'_c (\ll d_h n_h))$  denotes the query compression dimension; 和  $W^{DQ} \in \mathbb{R}^{d'c \times d}$ 、 $W^{UQ} \in \mathbb{R}^{d_h n_h \times d'_c}$  are the down-projection and up-projection matrices.  $\varepsilon \in \mathbb{R}^{d_h n_h \times d'c}$  分别是查询的下投影和上投影矩阵。

### 2.1.3. 解耦旋转位置嵌入

继 DeepSeek 67B (DeepSeek-AI, 2024) 之后，我们打算将旋转位置嵌入 (RoPE) (Su 等人, 2024) 用于 DeepSeek-V2。然而，RoPE 与低rankKV 压缩不兼容。具体来说，RoPE 对于键和查询都是位置敏感的。如果我们对键  $\mathbf{k}_t$  应用 RoPE，等式 10 中的  $W^{UK}$  将与位置敏感的 RoPE 矩阵相耦合。这样，在推理过程中， $W^{UK}$  就不能再被吸收到  $W^Q$  中，因为 RoPE 矩阵与当前生成的 token 将位于  $W^Q$  和  $W^{UK}$  之间，并且矩阵乘法不遵守交换律。因此，我们在推理过程中必须重新计算所有前缀令牌的键，这将极大地影响推理效率。

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  $\mathbf{q}_{t,i}^R \in \mathbb{R}^{d Rh}$  和共享密钥  $\mathbf{k}_{t,i}^R \in \mathbb{R}^{d Rh}$  来承载 RoPE，其中  $d Rh$  表示解耦查询的每个头维度和密钥。配备解耦的 RoPE 策略，MLA 执行以下计算：

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

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

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

$$\mathbf{k}_{t,i} = [\mathbf{k}_{t,i}^C; \mathbf{k}_{t,i}^R], \quad (17)$$

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

$$\mathbf{u}_t = W^O [\mathbf{o}_{t,1}; \mathbf{o}_{t,2}; \dots; \mathbf{o}_{t,n_h}], \quad (19)$$

$c$  和  $W^{KR} \in \mathbb{R}^{d Rh \times d}$  分别是生成解耦查询和密钥的矩阵； $\text{RoPE}(\cdot)$  表示应用 RoPE 矩阵的运算；和  $[\cdot; \cdot]$  表示串联操作。在推理过程中，解耦的密钥也应该被缓存。因此，DeepSeek-V2 需要一个包含  $(d c + d Rh)l$  元素的总 KV 缓存。为了演示 MLA 的完整计算过程，我们还在附录 C 中整理并提供了其完整公式。

### 2.1.4. Key-Value 缓存对比

我们在表1中展示了不同注意机制中每个令牌的KV缓存的比较。MLA只需要少量的KV缓存，相当于只有2.25组的GQA，但可以实现比MHA更强的性能。



Attention Mechanism	KV Cache per Token (# Element)	Capability
Multi-Head Attention (MHA)	$2n_h d_h l$	Strong
Grouped-Query Attention (GQA)	$2n_g d_h l$	Moderate
Multi-Query Attention (MQA)	$2d_h l$	Weak
MLA (Ours)	$(d_c + d_h^R)l \approx \frac{9}{2}d_h l$	Stronger

表 1 |不同注意力机制中每个令牌的 KV 缓存比较。  $n_h$ 表示注意力头的数量，  $d_h$ 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 $d_h^R$ 表示KV压缩维度和每头维度分别是 MLA 中解耦的查询和密钥。 KV缓存的数量以元素数量来衡量，与存储精度无关。对于 DeepSeek-V2，  $d_c$  设置为  $4d_h$ ，  $d_h^R$  设置为  $d_h/2$ 。所以，它的KV缓存相当于GQA，只有2.25组，但性能却比MHA强。

## 2.2. DeepSeekMoE: 以经济的成本训练强大的模型

### 2.2.1. 基本架构

对于 FFN，我们采用 DeepSeekMoE 架构 (Dai 等人, 2024)。DeepSeekMoE 有两个关键思想：将专家细分为更细的粒度，以实现更高的专家专业化 and 更准确的知识获取；以及隔离一些共享专家，以减少路由专家之间的知识冗余。在激活和总专家参数数量相同的情况下，DeepSeekMoE 可以大幅优于 GShard (Lepikhin 等人, 2021) 等传统 MoE 架构。

令  $\mathbf{u}_t$  为第  $t$  个 token 的 FFN 输入，我们计算 FFN 输出  $\mathbf{h}'_t$  如下：

as follows:

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

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

$$s_{i,t} = \text{Softmax}_i \mathbf{u}_t^T \mathbf{e}_i, \quad (22)$$

其中  $N_s$  和  $N_r$  分别表示共享专家和路由专家的数量；  $\text{FFN}^{(s)}_i(\cdot)$  和  $\text{FFN}^{(r)}_i(\cdot)$  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  $i$  个专家的阈值；  $s_{i,t}$  是代币与专家的亲和力；  $\mathbf{e}_i$  是该层中第  $i$  个路由专家的质心；  $\text{Topk}(\cdot, K)$  表示由为第  $t$  个令牌和所有路由专家计算的亲和力分数中的  $K$  最高分数组成的集合。

### 2.2.2. Device-Limited Routing

我们设计了一种设备限制的路由机制来限制 MoE 相关的通信成本。当采用专家并行性时，路由的专家将分布在多个设备上。对于每个代币，其与 MoE 相关的通信频率与其目标专家覆盖的设备数量成正比。由于 DeepSeekMoE 中的细粒度专家分割，激活的专家数量可能很大，因此如果我们应用专家并行性，与 MoE 相关的通信成本将会更高。

对于 DeepSeek-V2，除了简单地选择路由专家的 top-K 之外，我们还确保每个代币的目标专家将分布在最多  $M$  设备上。具体来说，对于每个令牌，我们首先选择其中具有亲和力得分最高的专家的  $M$  设备。然后，我们在这些  $M$  设备上的专家中进行 top-K 选择。在实践中，我们发现当  $M \geq 3$  时，设备限制路由可以实现与不受限制的 top-K 路由大致一致的良好性能。

### 2.2.3. 负载均衡的辅助损耗

我们在自动学习路由策略时考虑了负载均衡。首先，负载不平衡会增加路由崩溃的风险 (Shazeer et al., 2017)，阻碍一些专家得到充分的培训和利用。其次，当采用专家并行时，不平衡的负载会降低计算效率。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 (L ExpBal)、设备级负载均衡 (L DevBal) 和通信平衡 (L CommBal)。

专家级平衡损失。我们使用专家级余额损失 (Fedus et al., 2021; Lepikhinet al., 2021) 来减轻路由崩溃的风险：

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

$$f_i = \frac{N_r}{K_r T} \sum_{t=1}^T \mathbf{1}(\text{代币 } t \text{ 选择专家 } i), \quad (24)$$

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

其中  $\alpha_1$  是一个超参数，称为专家级平衡因子； $\mathbf{1}(\cdot)$  表示指示函数； $T$  表示序列中标记的数量。

设备级平衡损失。除了专家级的平衡损失之外，我们还设计了设备级的平衡损失，以确保不同设备之间的平衡计算。在 DeepSeek-V2 的训练过程中，我们将所有路由专家分为  $D$  组  $\{E_1, E_2, \dots, E_D\}$ ，并将每个组部署在单个设备上。设备级平衡损失计算如下：

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

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

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

其中  $\alpha_2$  是一个称为设备级平衡因子的超参数。

沟通平衡丧失。最后，我们引入一个通信平衡损耗，以保证各个设备的通信是平衡的。虽然设备限制路由机制保证了各个设备的发送通信是有界的，但是如果某个设备

比其他设备接收更多的令牌，实际通信效率也会受到影响。为了缓解这个问题，我们设计了如下的通信平衡损失：

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

$$f_i'' = \frac{D}{MT} \sum_{t=1}^T 1(\text{令牌}_t \text{发送到设备 } i), \quad (30)$$

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

其中 $\alpha_3$ 是一个超参数，称为通信平衡因子。设备限制路由机制的运行原理是确保每个设备最多将 $MT$ 隐藏状态传输到其他设备。同时，利用通信平衡损失来鼓励每个设备从其他设备接收大约 $MT$ 隐藏状态。通信平衡损失保证了设备之间信息的平衡交换，促进高效通信。

#### 2.2.4. Token-Dropping Strategy

虽然平衡损失旨在鼓励平衡负载，但重要的是要承认它们不能保证严格的负载平衡。为了进一步减轻负载不平衡造成的计算浪费，我们在训练期间引入了设备级令牌丢弃策略。这种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相当于1.0。然后，受到Riquelme et al. (2021)等人的启发，我们在每台设备上丢弃亲和力分数最低的令牌，直到达到计算预算。此外，我们确保属于大约10%训练序列的token永远不会被丢弃。这样，我们就可以根据效率要求灵活决定是否在推理过程中丢弃token，始终保证训练和推理的一致性。

### 3. Pre-Training

#### 3.1. 实验设置

##### 3.1.1. 数据建设

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高了数据质量。为了扩大我们的预训练语料库，我们探索互联网数据的潜力并优化我们的清理流程，从而恢复大量误删除的数据。此外，我们纳入了更多的中文数据，旨在更好地利用中文互联网上可用的语料库。除了数据量，我们还关注数据质量。我们利用各种来源的高质量数据丰富了我们的预训练语料库，同时改进了基于质量的过滤算法。改进后的算法保证了大量无益数据被剔除，而有价值的数据大部分被保留。此外，我们从预训练语料库中过滤掉有争议的内容，以减轻特定区域文化引入的数据偏差。附录 E 详细讨论了这种过滤策略的影响。

我们采用与 DeepSeek 67B 中使用的相同的分词器，它是基于字节级字节对编码（BBPE）算法构建的，词汇量为 100K。我们的标记化预训练语料库包含 8.1T 个标记，其中中文标记比英文标记多约 12%。

### 3.1.2. Hyper-Parameters

模型超参数。我们将 Transformer 层数设置为 60，隐藏维度设置为 5120。所有可学习参数均以 0.006 的标准差随机初始化。在 MLA 中，我们将注意力头数量  $n_h$  设置为 128，每头维度  $d_h$  设置为 128。KV 压缩维度  $d_c$  设置为 512，查询压缩维度

$c$  设置为 1536。对于解耦的查询和键，我们将每头维度  $d_{Rh}$  设置为 64。

（2024），我们用 MoE 层替换除第一层之外的所有 FFN。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，将激活 6 个专家每个令牌。此外，低秩压缩和细粒度专家分割会影响层的输出规模。因此，在实践中，我们在压缩潜在向量之后采用额外的 RMS Norm layer，并在宽度瓶颈（即压缩的潜在向量和 Route Experts 的中间隐藏状态）处乘以额外的缩放因子，以确保稳定的训练。在此配置下，DeepSeek-V2 总共包含 236B 个参数，其中每个令牌激活 21B 个参数。

训练超参数。我们采用 AdamW 优化器（Loshchilov 和 Hutter, 2017），超参数设置为  $\beta_1 = 0.9$ ， $\beta_2 = 0.95$ ， $\text{weight\_decay} = 0.1$ 。学习率是使用预热和逐步衰减策略来安排的（DeepSeek-AI, 2024）。最初，学习率在前 2K 步中从 0 线性增加到最大值。随后，在训练大约 60% 的 token 后，将学习率乘以 0.316，在训练大约 90% 的 token 后，再次乘以 0.316。最大学习率设置为  $2.4 \times 10^{-4}$ ，梯度裁剪范数设置为 1.0。我们还使用批量大小调度策略，在前 225B 个令牌的训练中，批量大小逐渐从 2304 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 令牌上训练 DeepSeek-V2。我们利用管道并行性将模型的不同层部署在不同的设备上，对于每一层，路由专家将统一部署在 8 个设备上（ $D = 8$ ）。对于设备限制路由，每个令牌将被发送至最多 3 个设备（ $M = 3$ ）。至于余额损失，我们将  $\alpha_1$  设置为 0.003， $\alpha_2$  设置为 0.05， $\alpha_3$  设置为 0.02。我们在训练期间采用令牌丢弃策略来加速，但在评估时不丢弃任何令牌。

### 3.1.3. 基础设施

DeepSeek-V2 基于 HAI-LLM 框架（High-flyer, 2023）进行训练，这是我们工程师内部开发的高效、轻量级的训练框架。它采用 16 路零气泡管道并行性（Qi 等人, 2023）、8 路专家并行性（Lepikhin 等人, 2021）和 ZeRO-1 数据并行性（Rajbhandari 等人, 2020）。鉴于 DeepSeek-V2 的激活参数相对较少，并且重新计算了部分算子以节省激活内存，因此无需张量并行即可进行训练，从而减少了通信开销。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全对全通信重叠。我们还为通信、路由算法和融合定制了更快的 CUDA 内核。

通过“大海捞针”对 DeepSeek-V2 Base 128K 上下文进行压力测试

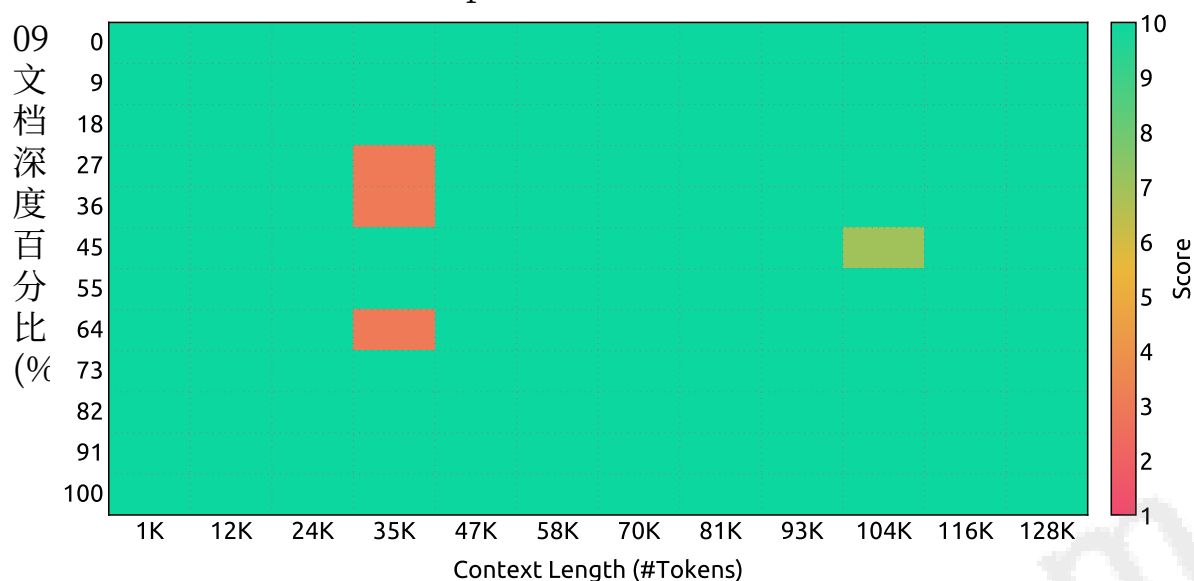


图4 | “大海捞针”（NIAH）测试的评估结果。DeepSeek-V2 在高达 128K 的所有上下文窗口长度上都表现良好。

不同专家之间的线性计算。此外，MLA还基于FlashAttention-2 (Dao, 2023) 的改进版本进行了优化。

我们在配备 NVIDIA H800 GPU 的集群上进行所有实验。H800 集群中的每个节点包含 8 个 GPU，通过节点内的 NVLink 和 NVSwitch 连接。跨节点，InfiniBand 互连用于促进通信。

#### 3.1.4.长上下文扩展

在 DeepSeek-V2 的初始预训练之后，我们使用 YaRN (Peng et al., 2023) 将默认上下文窗口长度从 4K 扩展到 128K。YaRN 专门应用于解耦共享密钥  $k R_t$ ，因为它负责承载 RoPE (Su et al., 2024)。对于 YaRN，我们将  $scales$  设置为 40， $\alpha$  设置为 1， $\beta$  设置为 32，目标最大上下文长度设置为 160K。在这些设置下，我们可以预期模型对于 128K 的上下文长度有良好的响应。与原始 YaRN 略有不同，由于我们独特的注意力机制，我们调整长度缩放因子来调节注意力熵。因子  $\sqrt{t}$  计算如下旨在最小化困惑度。

我们还对模型进行了 1000 个步骤的训练，序列长度为 32K，批量大小为 576 个序列。尽管训练仅在 32K 的序列长度下进行，但该模型在以 128K 的上下文长度进行评估时仍然表现出稳健的性能。如图 4 所示，“大海捞针” (NIAH) 测试的结果表明 DeepSeek-V2 在高达 128K 的所有上下文窗口长度上都表现良好。

### 3.2.评价

#### 3.2.1.评估基准

DeepSeek-V2 是在双语语料库上进行预训练的，因此我们在一系列英语和中文基准上对其进行评估。我们的评估基于我们综合的内部评估框架

在我们的 HAI-LLM 框架中。包含的基准分类和列出如下，其中下划线基准为中文：

多主题多项选择数据集包括 MMLU (Hendrycks et al., 2020)、C-Eval(Huang et al., 2023) 和 val 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)。

闭卷问答数据集包括 TriviaQA (Joshi et al., 2017) 和 Natu-ralQuestions (Kwiatkowski et al., 2019)。

阅读理解数据集包括 RACE Lai 等人。 (2017)、DROP (Dua et al., 2019)、C3 (Sun et al., 2019) 和 19), CMRC (Cui et al., 2019)。

参考消歧数据集包括 WinoGrande Sakaguchi 等人。 (2019) 和 CLUEWSC (Xu 等人, 2020)。

语言建模数据集包括 Pile (Gao et al., 2020)。

中国理解和文化数据集包括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 等人, 2023)。请注意，AGIEval 包括英语和中文子集。

继我们之前的工作 (DeepSeek-AI, 2024) 之后，我们对数据集采用基于困惑的评估，包括 HellaSwag、PIQA、WinoGrande、RACE-Middle、RACE-High、MMLU、ARC-Easy、ARC-Challenge、CHID、C-Eval、CMMLU、C3 和 CCPM，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 进行基于语言建模的评估，并使用每字节位数 (BPB) 作为衡量标准，以保证具有不同分词器的模型之间的公平比较。

为了直观地概述这些基准，我们还在附录 G 中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比较，包括DeepSeek 67B (DeepSeek-AI, 2024) (我们之前的版本)、Qwen1.5 72B (Bai et al., 2023)、LLaMA3 70B (AI@Meta, 2024) 和 Mixtral 8x22B (Mistral, 2024)。我们使用我们的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，仅用 21B 个激活参数，DeepSeek-V2 在几乎所有基准测试上都显著优于 DeepSeek67B，在开源模型中实现了顶级性能。

此外，我们还详细地将 DeepSeek-V2 与其开源版本进行了详细比较。(1) 与另一款支持中英文的型号Qwen1.5 72B相比，DeepSeek-V2在大多数英文、代码、数学基准测试中展现出压倒性的优势。至于中国基准测试，Qwen1.5 72B 在以下方面表现出更好的性能

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

表 2 | DeepSeek-V2与其他代表性开源模型的比较。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。粗体表示最好，下划线表示次佳。差距小于0.3的分数视为同一水平。仅通过 21B 个激活参数，DeepSeek-V2 就实现了开源模型中顶级的性能。

多主题多项选择任务，而 DeepSeek-V2 与其他任务相当或更好。请注意，对于 CHID 基准测试，Qwen1.5 72B 的分词器将在我们的评估框架中遇到错误，因此我们将 Qwen1.5 72B 的 CHID 分数留空。（2）与 Mixtral 8x22B 相比，除了与英语常识知识密切相关的 TriviaQA、NaturalQuestions 和 HellaSwag 之外，DeepSeek-V2 实现了相当或更好的英语性能。值得注意的是，DeepSeek-V2 在 MMLU 上的性能优于 Mixtral 8x22B。在代码和数学基准测试中，DeepSeek-V2 表现出与 Mixtral 8x22B 相当的性能。由于 Mixtral 8x22B 没有专门针对中文数据进行训练，其中文能力远远落后于 DeepSeek-V2。（3）与 LLaMA3 70B 相比，DeepSeek-V2 的训练英文 token 不到四分之一。因此，我们承认 DeepSeek-V2 在基础英语能力上与 LLaMA3 70B 仍有细微差距。然而，即使训练令牌和激活参数少得多，DeepSeek-V2 仍然表现出与 LLaMA3 70B 相当的代码和数学能力。此外，作为双语语言模型，DeepSeek-V2 优于 LLaMA3

在中国基准测试中，70B 占压倒性优势。

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

### 3.2.3. 训练和推理效率

培训费用。由于 DeepSeek-V2 为每个令牌激活的参数更少，并且比 DeepSeek 67B 需要更少的 FLOP，因此理论上训练 DeepSeek-V2 将比训练 DeepSeek 67B 更经济。虽然训练 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 的训练可以获得相对较高的模型浮点运算利用率（MFU）。我们在 H800 集群上的实际训练中，每万亿个 token 的训练，DeepSeek 67B 需要 300.6K GPU 小时，而 DeepSeek-V2 只需要 172.8K GPU 小时，即稀疏的 DeepSeek-V2 与密集的 DeepSeek 67B 相比可以节省 42.5% 的训练成本。

推理效率。为了高效部署 DeepSeek-V2 进行服务，我们首先将其参数转换为 FP8 的精度。此外，我们还对 DeepSeek-V2 进行 KV 缓存量化（Hooper et al., 2024; Zhao et al., 2023），以进一步将其 KV 缓存中的每个元素平均压缩为 6 位。受益于 MLA 和这些优化，实际部署的 DeepSeek-V2 所需的 KV 缓存比 DeepSeek 67B 少得多，因此可以提供更大的批量大小。我们根据实际部署的 DeepSeek 67B 服务的提示和生成长度分布来评估 DeepSeek-V2 的生成吞吐量。在具有 8 个 H800 GPU 的单节点上，DeepSeek-V2 实现了每秒超过 50K 令牌的生成吞吐量，是 DeepSeek 67B 最大生成吞吐量的 5.76 倍。此外，DeepSeek-V2 的即时输入吞吐量超过每秒 100K 令牌。

## 4. 对齐

### 4.1. Supervised Fine-Tuning

基于我们之前的研究（DeepSeek-AI, 2024），我们整理了指令调整数据集以包含 150 万个实例，其中 120 万个用于帮助的实例和 30 万个用于安全的实例。与初始版本相比，我们提高了数据质量，以减轻幻觉反应并提高写作水平。我们对 DeepSeek-V2 进行了 2 个 epoch 的微调，学习率设置为  $5 \times 10^{-6}$ 。对于 DeepSeek-V2 Chat (SFT) 的评估，除了几个代表性的多项选择任务（MMLU 和 ARC）外，我们主要包括基于生成的基准。我们还使用提示级松散准确性作为指标，对 DeepSeek-V2 Chat (SFT) 进行指令跟踪评估 (IFEval) (Zhou et al., 2023)。此外，我们使用 2023 年 9 月 1 日至 2024 年 4 月 1 日的 LiveCodeBench (Jain et al., 2024) 问题来评估聊天模型。除了标准基准之外，我们还进一步在开放式对话基准上评估我们的模型，包括 MT-Bench (Zheng et al., 2023)、AlpacaEval 2.0 (Dubois et al., 2024) 和 AlignBench (Liu et al., 2023)。为了进行比较，我们还在我们的评估框架和设置中评估了 Qwen1.5 72B Chat、LLaMA-3-70B Instruct 和 Mistral-8x22B Instruct。对于 DeepSeek 67B Chat，我们直接参考之前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 的潜力并使其与人类偏好保持一致，我们进行强化学习（RL）来调整其偏好。

强化学习算法。为了节省强化学习的训练成本，我们采用群组相对策略优化（GRPO）（Shao et al., 2024），它放弃了通常与策略模型大小相同的批评家模型，而是根据群组分数估计基线。具体来说，对于每个问题  $q$ ，GRPO 从旧策略  $\pi_{\theta_{old}}$  中采样一组输出  $\{o_1, o_2, \dots, o_G\}$ ，然后通过最大化以下目标来优化策略模型  $\pi_{\theta}$ ：

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

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

其中  $\epsilon$  和  $\beta$  是超参数； $A_i$  是优势，使用一组奖励  $\{r_1, r_2, \dots, r_G\}$  计算。。。。， $r_G\}$  对应于每组内的输出：

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

培训策略。在我们的初步实验中，我们发现对推理数据（例如代码和数学提示）的强化学习训练表现出与一般数据训练不同的独特特征。例如，我们模型的数学和编码能力可以在较长时期的训练步骤中不断提高。因此，我们采用两阶段强化学习训练策略，首先执行推理对齐，然后执行人类偏好对齐。在第一个推理对齐阶段，我们训练用于代码和数学推理任务的奖励模型  $RM_{reasoning}$ ，并根据  $RM_{reasoning}$  的反馈优化策略模型：

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

在第二个人类偏好调整阶段，我们采用多奖励框架，该框架从有用奖励模型  $RM_{helpful}$ 、安全奖励模型  $RM_{safety}$  和基于规则的奖励模型中获取奖励  $RM_{rule}$ 。响应  $o_i$  的最终奖励是

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

其中  $c_1$ 、 $c_2$  和  $c_3$  是相应的系数。

为了获得在强化学习训练中发挥关键作用的可靠奖励模型，我们仔细收集偏好数据，并精心进行质量过滤和比例调整。我们根据编译器反馈获得代码偏好数据，并根据真实标签获得数学偏好数据。对于奖励模型训练，我们使用 DeepSeek-V2 Chat (SFT) 初始化奖励模型，并使用逐点或成对损失来训练它们。在我们的实验中，我们观察到强化学习训练可以充分挖掘和激活我们模型的潜力，使其能够从可能的响应中选择正确且令人满意的答案。

优化培训效率。在超大型模型上进行强化学习训练对训练框架提出了很高的要求。它需要仔细的工程优化来管理GPU内存和RAM压力，同时保持较快的训练速度。为此，我们实现了以下工程优化。(1)首先，我们提出了一种混合引擎，分别采用不同的并行策略进行训练和推理，以实现更高的GPU利用率。(2)其次，我们利用具有大批量大小的 vLLM (Kwon et al., 2023) 作为推理后端来加快推理速度。(3)第三，我们精心设计了将模型卸载到CPU并将模型加载回GPU的调度策略，从而在训练速度和内存消耗之间实现了近乎最佳的平衡。

#### 4.3. 评估结果

标准基准评估。最初，我们在标准基准上评估 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 评估方面有了显著改进。这一进展可归因于我们的 SFT 数据的纳入，其中包含大量数学和代码相关内容。此外，DeepSeek-V2 Chat (RL) 进一步提升了数学和代码基准测试的性能。我们在附录 F 中展示了更多代码和数学评估。

至于与其他模型的比较，我们首先将 DeepSeek-V2 Chat (SFT) 与 Qwen1.5 72B Chat 进行比较，发现 DeepSeek-V2 Chat (SFT) 几乎所有的英语、数学、代码都超越了 Qwen1.5 72B Chat 基准。在中国基准测试中，DeepSeek-V2 Chat (SFT) 在多主题多项选择任务上的得分略低于 Qwen1.5 72B Chat，这与从其基础版本观察到的性能一致。与最先进的开源 MoE 模型 Mixtral 8x22B Instruct 相比，DeepSeek-V2 Chat (SFT) 在大多数基准测试中表现出更好的性能，除了 NaturalQuestions 和 IFEval 之外。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中表现出相似的性能。LLaMA3 70B Chat 在 MMLU 和 IFEval 上表现出更好的性能，而 DeepSeek-V2 Chat (SFT) 在中文任务上表现出更强的性能。最终，与 DeepSeek-V2 Chat (SFT) 相比，DeepSeek-V2 Chat (RL) 在数学和编码任务中展示了进一步增强的性能。这些比较突出了 DeepSeek-V2 Chat 相对于各个领域和语言中的其他语言模型的优势。

对开放式一代的评估。我们继续在开放式对话基准上对我们的模型进行额外评估。对于英语开放式对话的生成，我们使用 MT-Bench 和 AlpacaEval 2.0 作为基准。表 4 中给出的评估结果表明 DeepSeek-V2 Chat (RL) 相对于 DeepSeek-V2 Chat (SFT) 具有显著的性能优势。这一结果展示了我们的强化学习训练在改善一致性方面的有效性。与其他开源模型相比，DeepSeek-V2 Chat (RL) 在两个基准测试中均表现出优于 Mistral 8x22B Instruct 和 Qwen1.5 72B Chat 的性能。与 LLaMA3 70B Instruct 相比，DeepSeek-V2 Chat (RL) 在 MT-Bench 上展示了有竞争力的性能，并且在 AlpacaEval 2.0 上的表现明显优于它。这些结果凸显了 DeepSeek-V2 Chat (RL) 在生成高质量且上下文相关的响应方面的强大性能，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) 比 DeepSeek-V2 Chat (SFT) 表现出轻微优势。值得注意的是，DeepSeek-V2 Chat (SFT) 显著超越了所有开源中国模型。它显著优于第二好的开源模型 Qwen1.5

	Benchmark	# Shots	DeepSeek 67B Chat	Qwen 1.5 72B Chat	LLaMA3 70B Inst.	Mixtral 8x22B Inst.	DeepSeek-V2 Chat (SFT)	DeepSeek-V2 Chat (RL)
	Context Length	-	4K	32K	8K	64K	128K	128K
	Architecture	-	Dense	Dense	Dense	MoE	MoE	MoE
	# Activated Params	-	67B	72B	70B	39B	21B	21B
	# Total Params	-	67B	72B	70B	141B	236B	236B
English	TriviaQA	5-shot	81.5	79.6	69.1	80.0	85.4	<b>86.7</b>
	NaturalQuestions	5-shot	47.0	46.9	44.6	<b>54.9</b>	51.9	<u>53.4</u>
	MMLU	5-shot	71.1	76.2	<b>80.3</b>	77.8	<u>78.4</u>	77.8
	ARC-Easy	25-shot	96.6	96.8	96.9	97.1	<u>97.6</u>	<b>98.1</b>
	ARC-Challenge	25-shot	88.9	<u>91.7</u>	<b>92.6</b>	90.0	<b>92.5</b>	<b>92.3</b>
	BBH	3-shot	71.7	<u>65.9</u>	<u>80.1</u>	78.4	<b>81.3</b>	79.7
	AGIEval	0-shot	46.4	<u>62.8</u>	<u>56.6</u>	41.4	<b>63.2</b>	61.4
	IFEval	0-shot	55.5	57.3	<b>79.7</b>	<u>72.1</u>	64.1	63.8
Code	HumanEval	0-shot	73.8	68.9	76.2	75.0	<u>76.8</u>	<b>81.1</b>
	MBPP	3-shot	61.4	52.2	69.8	64.4	<u>70.4</u>	<b>72.0</b>
	CRUXEval-I-COT	2-shot	49.1	51.4	<u>61.1</u>	59.4	59.5	<b>61.5</b>
	CRUXEval-O-COT	2-shot	50.9	56.5	<b>63.6</b>	<b>63.6</b>	60.7	<u>63.0</u>
	LiveCodeBench	0-shot	18.3	18.8	<u>30.5</u>	25.0	28.7	<b>32.5</b>
Math	GSM8K	8-shot	84.1	81.9	<b>93.2</b>	87.9	90.8	<u>92.2</u>
	MATH	4-shot	32.6	40.6	48.5	49.8	<u>52.7</u>	<b>53.9</b>
	CMATH	0-shot	80.3	<b>82.8</b>	79.2	75.1	<u>82.0</u>	<u>81.9</u>
Chinese	CLUEWSC	5-shot	78.5	<b>90.1</b>	85.4	75.8	<u>88.6</u>	<b>89.9</b>
	C-Eval	5-shot	65.2	<b>82.2</b>	67.9	60.0	<u>80.9</u>	78.0
	CMMLU	5-shot	67.8	<b>82.9</b>	70.7	61.0	<u>82.4</u>	81.6

表 3 | DeepSeek-V2 Chat (SFT)、DeepSeek-V2 Chat (RL) 和其他代表性开源聊天模型之间的比较。关于 TriviaQA 和 NaturalQuestions，值得注意的是，聊天模型（例如 LLaMA3 70B Instruct）可能不严格遵守通常在少样本设置中指定的格式约束。因此，这可能导致我们的评估框架中的某些模型被低估。

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	<b>8.95</b>	34.4
DeepSeek-V2 Chat (SFT)	8.62	30.0
DeepSeek-V2 Chat (RL)	<b>8.97</b>	<b>38.9</b>

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

72B 聊中国推理和语言。此外，DeepSeek-V2 Chat (SFT) 和 DeepSeek-V2 Chat (RL) 的性能均优于 GPT-4-0613 和 ERNIEBot 4.0，巩固了我们的模型在支持中文的顶级 LLM 中的地位。具体来说，DeepSeek-V2 Chat (RL) 在中文理解方面表现出色，优于包括 GPT-4-Turbo-1106-Preview 在内的所有模型。另一方面，DeepSeek-V2 Chat (RL) 的推理能力仍然落后于 Erniebot-4.0 和 GPT-4s 等巨型模型。

Model 模型	Overall 总分	Reasoning 中文推理			Language 中文语言						
		Avg. 推理 总分	Math. 数学 计算	Logi. 逻辑 推理	Avg. 语言 总分	Fund. 基本 任务	Chi. 中文 理解	Open. 综合 问答	Writ. 文本 写作	Role. 角色 扮演	Pro. 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 | AlignBench 排行榜由 GPT-4-0613 评级。模型根据总分按降序排列。标有 \* 的模型代表我们通过他们的 API 服务或开放加权模型对其进行评估，而不是参考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他们的 API 时的时间戳。

#### 4.4. 讨论

SFT 数据量。围绕大型 SFT 语料库必要性的讨论一直是激烈争论的话题。之前的工作 (Young et al., 2024; Zhou et al., 2024) 认为少于 10K 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，如果我们使用少于 10K 的实例，我们会观察到 IFEval 基准的性能显著下降。一种可能的解释是，语言模型需要一定量的数据来开发特定技能。尽管所需的数据量可能会随着模型大小的增加而减少，但不能完全消除。我们的观察强调了对足够数据的迫切需求，以使法学硕士具备所需的能力。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。在人类偏好调整过程中，我们观察到开放式生成基准的性能显著提高，无论是人工智能还是人类评估者的评分。然而，我们也注意到一种“对齐税”现象 (Ouyang et al., 2022)，即对齐过程会对某些标准基准（例如 BBH）的性能产生负面影响。为了减轻对齐税，在强化学习阶段，我们在数据处理和改进训练策略方面做出了巨大的努力，最终在标准基准和开放基准的性能之间实现了可容忍的权衡。探索如何使模型与人类偏好保持一致，而无需

损害其总体性能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，我们发现在线方法明显优于离线方法。因此，我们投入了巨大的努力来实现在线 RL 框架来对齐 DeepSeek-V2。关于线上或线下偏好一致的结论在不同的背景下可能会有所不同，我们为未来的工作保留对它们之间更全面的比较和分析。

## 5. 结论、局限性和未来的工作

在本文中，我们介绍了 DeepSeek-V2，一个支持 128K 上下文长度的大型 MoE 语言模型。除了强大的性能之外，它还具有经济训练和高效推理的特点，这得益于其包括 MLA 和 DeepSeekMoE 在内的创新架构。实际应用中，与 DeepSeek 67B 相比，DeepSeek-V2 的性能明显增强，同时节省了 42.5% 的训练成本，减少了 93.3% 的 KV 缓存，最大生成吞吐量提升至 5.76 倍。评估结果进一步表明，仅用 21B 个激活参数，DeepSeek-V2 就达到了开源模型中顶级的性能，成为最强的开源 MoE 模型。

DeepSeek-V2 及其聊天版本具有其他 LLM 中常见的公认局限性，包括预训练后缺乏持续的知识更新、可能生成非事实信息（例如未经验证的建议）以及产生幻觉的机会。此外，由于我们的数据主要由中文和英文内容组成，因此我们的模型对其他语言的熟练程度可能有限。中英文以外的场景请谨慎使用。

DeepSeek 将以长远的眼光持续投资开源大型模型，逐步向通用人工智能的目标迈进。ing

· 在我们持续的探索中，我们致力于设计能够进一步扩展 MoE 模型的方法，同时保持经济的训练和推理成本。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。· 我们的对齐团队不断努力增强我们的模型，旨在开发一个对全球用户不仅有帮助而且诚实且安全的模型。我们的最终目标是使我们的模型的价值观与人类价值观保持一致，同时最大限度地减少人类监督的需要。通过优先考虑道德考虑和负责任的发展，我们致力于为社会创造积极和有益的影响。· 目前，DeepSeek-V2 旨在专门支持文本模式。在我们的前瞻性议程中，我们打算使我们的模型能够支持多种模式，增强其在更广泛场景中的多功能性和实用性。

## References

人工智能@元。Llama 3 模型卡，2024 年。URL [https://github.com/meta-llama/llama3/blob/main/MODEL\\_CARD.md](https://github.com/meta-llama/llama3/blob/main/MODEL_CARD.md)。

J. Ainslie, J. Lee-Thorp, M. de Jong, Y. Zemlyanskiy, F. Lebrón 和 S. Sanghai.  
Gqa: 从多头检查点训练通用多查询变压器模型。arXiv 预印本 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. 勒, 等人。使用大型语言模型进行程序综合。arXiv 预印本 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. 李, 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. 涂, 王鹏, 王S.王, W.王, S.吴, 徐B., J.徐, A.杨, H. 杨, J.杨, S.杨, Y.姚, B.于, 袁宏, 袁志, 张建, 张X, 张Y, 张Z, 周C, 周J, 周X, 朱T。Qwen技术报告。arXiv 预印本 arXiv:2309.16609, 2023。

Y.Bisk、R.Zellers、R.L.Bras、J.Gao 和 Y.Choi。PIQA: 用自然语言推理物理常识。第三十四届 AAAI 人工智能大会, AAAI2020, 第三十二届人工智能创新应用大会, IAAI2020, 第十届 AAAI 人工智能教育进展研讨会, EAAI2020, 美国—纽约, 2 月 7-12 日, 2020 年, 第 7432–7439 页。AAAI 出版社, 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. 约瑟夫、G. 布罗克曼、A. 雷、R. 普里、G. 克鲁格、M. 彼得罗夫、H. 克拉夫、G. 萨斯特里、P. 米什金、B. Chan、S. Gray、N. Ryder、M. Pavlov、A. Power、L. Kaiser、M. Bavarian、C. Winter、P. Tillet、F. P. Such、D. Cummings、M. Plapert、F. Chantzis、E. Barnes、A. Herbert-Voss、W. H. Guss、A. 尼科尔、A. Paino、N. Tezak、J. Tang、I. Babuschkin、S. Balaji、S. Jain、W. Saunders、C. Hesse、A. N. 卡尔、J. 雷克、J. 阿希姆、V. 米斯拉、E. 森川、A. 雷德福德、M. 奈特、M. 布伦戴奇、M. Murati、K. Mayer、P. Welinder、B. McGrew、D. Amodei、S. McCandlish、I. Sutskever 和 W. 扎伦巴。评估在代码上训练的大型语言模型。CoRR, abs/2107.03374, 2021。

URL <https://arxiv.org/abs/2107.03374>.

P. Clark、I. Cowhey、O. Etzioni、T. Khot、A. Sabharwal、C. Schoenick 和 O. Tafjord。您认为您已经解决了问题回答问题吗? 尝试 arc, AI2 推理挑战赛。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. Plapert、J. Tworek、J. 希尔顿、R. 中野, 等人。培训验证员解决数学应用题。arXiv 预印本 arXiv:2110.14168, 2021。

Y. Cui、T. Liu、W. Che、L. Shaw、Z. Chen、W. Ma、S. Wang 和 G. Hu。用于中文机器阅读理解跨度提取数据集。K. Inui、J. Jiang、V. Ng 和 X. Wan, 编辑, 2019 年自然语言处理经验方法会议和第九届自然语言处理国际联合会议 (EMNLP—IJCNLP) 会议记录, 第 5883 页 - 5889, 中国香港, 2019 年 11 月。计算协会——  
tional Linguistics. doi: 10.18653/v1/D19-1600. URL  
600.

戴大、邓长、赵长、徐瑞新、高浩、陈大、李建、曾伟、余小、吴勇、谢子、李勇、黄平、F. 罗、C. 阮、Z. 隋, 和 W. 梁。Deepseekmoe: 迈向混合专家语言模型的终极专家专业化。CoRR, abs/2401.06066, 2024。网址  
<https://doi.org/10.48550/arXiv.2401.06066>.

T. 道。FlashAttention-2: 更快的注意力, 更好的并行性和工作分区, 2023。

DeepSeek-AI. Deepseek LLM: 以长期主义扩展开源语言模型。钴RR, abs/2401.02954, 2024. URL

D. Dua、Y. Wang、P. Dasigi、G. Stanovsky、S. Singh 和 M. Gardner. DROP: 阅读理解基准, 需要对段落进行离散推理。J. Burstein、C. Doran 和 T. Solorio, 编辑, 计算语言学协会北美分会 2019 年会议记录: 人类语言技术, NAACL-HLT2019, 美国明尼苏达州明尼阿波利斯, 2019 年 6 月 2-7 日, 第 1 卷 (长论文和短论文), 第 2368-2378 页。计算语言学协会, 2019。doi: 10.18653/V1/N19-1246。网址 <https://doi.org/10.18653/v1/n19-1246>。

Y. Dubois、B. Galambosi、P. Liang 和 T. B. Hashimoto. 长度控制的 alpacaEval: 一种消除自动评估器偏差的简单方法。arXiv 预印本 arXiv:2404.04475, 2024。

W. Fedus、B. Zoph 和 N. Shazeer. 开关变压器: 扩展到万亿参数模型具有简单高效的稀疏性。CoRR, abs/2101.03961, 2021。URL <https://arxiv.org/abs/2101.03961>。

L. 高、S. Biderman、S. Black、L. Golding、T. Hoppe、C. Foster、J. Phang、H. He、A. Thite、N. 锅岛等人。The Pile: 用于语言建模的 800GB 不同文本数据集。arXiv 预印本 arXiv:2101.00027, 2020。

谷歌。介绍 Gemini: 我们最大、最有能力的人工智能模型, 2023 年。URL <https://blog.google/technology/ai/google-gemini-ai/>。

A. Gu、B. Rozière、H. Leather、A. Solar-Lezama、G. Synnaeve 和 S. I. Wang. CruxEval: 代码推理、理解和执行的基准, 2024 年。

D. Hendrycks、C. Burns、S. Basart、A. Zou、M. Mazeika、D. Song 和 J. Steinhardt. 测量大规模多任务语言理解。arXiv 预印本 arXiv:2009.03300, 2020。

D. Hendrycks、C. Burns、S. Kadavath、A. Arora、S. Basart、E. Tang、D. Song 和 J. Steinhardt. 使用数学数据集衡量数学问题的解决。arXiv 预印本 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 和 A. Gholami. Kvquant: 通过 KV 缓存量化实现 1000 万上下文长度的 LLM 推理。钴RR, 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: 揭示小型的潜力具有可扩展训练策略的语言模型。arXiv 预印本 arXiv:2404.06395, 2024。

黄Y.黄, Y.白, Z.朱, J.张, J.张, T.苏, J.刘, C.Lv, Y.张, J.雷, 等。C-Eval: 针对基础模型的多层次多学科中文评估套件。arXiv 预印本 arXiv: 2305.08322, 2023。

N. Jain、K. Han、A. Gu、W.-D. Li、F. Yan、T. Zhang、S. Wang、A. Solar-Lezama、K. Sen 和 I. Stoica. Livecodebench: 对 code.arXiv 大型语言模型进行全面且无污染的评估。arXiv 预印本 arXiv:2403.07974, 2024。

M. Joshi、E. Choi、D. Weld 和 L. Zettlemoyer。TriviaQA: 用于阅读理解的大规模远程监督挑战数据集。在 R. Barzilay 和 M.-Y. Kan, 编辑, 计算语言学协会第 55 届年会论文集 (第一卷: 长论文), 第 1601–1611 页, 加拿大温哥华, 2017 年 7 月。计算语言学协会

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 和 S. Petrov。自然问题: 问答研究的基准。跨。副教授。计算。语言学, 7: 452–466, 2019. doi: 10.1162/tacl\\_a\\_00276.

URL [https://doi.org/10.1162/tacl\\\_a\\\_00276](https://doi.org/10.1162/tacl\_a\_00276).

W. Kwon、Z. Li、S. Zhuang、Y. Shen、L. Cheng、C. H. Yu、J. E. Gonzalez、H. Zhang 和 I. Stoica。用于页面注意力机制的大型语言模型的高效内存管理。2023 年 ACM SIGOPS 第 29 届操作系统原理研讨会论文集。

G. Lai、Q. Xie、H. Liu、Y. Yang 和 E. H. Hovy。RACE: 来自考试的大规模阅读理解数据集。M. Palmer、R. Hwa 和 S. Riedel, 编辑, 2017 年自然语言处理经验方法会议论文集, EMNLP 2017, 丹麦哥本哈根, 2017 年 9 月 9–11 日, 第 785–794 页。计算协会

Linguistics, 2017. doi: 10.18653/V1/D17-1082. URL <https://doi.org/10.18653/v1/d17-1082>.

D. Lepikhin、H. Lee、Y. Xu、D. Chen、O. Firat、Y. Huang、M. Krikun、N. Shazeer 和 Z. Chen。Gshard: 通过条件计算和自动分片扩展巨型模型。第九届国际学习表征会议, 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 和 T. Baldwin。CMMLU: 测量中文的大规模多任务语言理解。arXiv 预印本 arXiv: 2306.09212, 2023。

W. Li、F. Qi、M. Sun、X. Yi 和 J. 张。CCPM: 中国古典诗歌匹配数据集, 2021。

刘X., 雷X., 王S.王, Y.黄, Z.冯, B.文, J.程, P.柯, Y.徐, W.L.谭, X.张, L.孙, H. Wang, J. 张, M. Huang, Y. Dong, J. Tang。Alignbench: 大型语言模型的中文对齐基准测试。CoRR, abs/2311.18743, 2023. doi: 10.48550/A

RXIV.2311.18743. URL <https://doi.org/10.48550/arXiv.2311.18743>.

I. 洛希洛夫和 F. 哈特。解耦权重衰减正则化。arXiv 预印本 arXiv: 1711.05101, 2017。

米斯特拉尔。更便宜、更好、更快、更强: 继续推动人工智能前沿并实现它

2024 年, 所有人均可访问。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. 欧阳, J. 吴, X. 江, D. 阿尔梅达, C. 温赖特, P. 米什金, C. 张, S. 阿加瓦尔, K. 斯拉马, A. 雷, 等人。训练语言模型以遵循人类反馈的指令。神经信息处理系统的进展, 35: 27730–27744, 2022。



B. Peng, J. Quesnelle, H. Fan 和 E. Shippole. Yarn: 大型语言模型的高效上下文窗口扩展。arXiv 预印本 arXiv:2309.00071, 2023。

P. Qi, X. Wan, G. Huang 和 M. Lin. 零气泡管道并行性。arXiv 预印本 arXiv: 2401.10241, 2023。

S. Rajbhandari, J. Rasley, O. Ruwase 和 Y. He. 零: 针对训练万亿参数模型的内存优化。在 SC20: 高性能计算、网络、存储和分析国际会议, 第 1-16 页。IEEE, 2020。

C. Riquelme, J. Puigcerver, B. Mustafa, M. Neumann, R. Jenatton, A. S. Pinto, D. Keysers 和 N. 霍尔斯比。通过稀疏的专家组合来扩展愿景。在《神经信息处理系统进展 34: 2021 年神经信息处理系统年会》中, NeurIPS 2021, pages 8583–8595, 2021. URL <https://proceedings.neurips.cc/paper/2021/hash/48237d9f2dea8c74c2a72126cf63d933-Abstract.html>.

K. Sakaguchi, R. L. Bras, C. Bhagavatula 和 Y. Choi. Winogrande: 大规模对抗性 winogradschema 挑战, 2019 年。

Z. Shao, P. Wang, Q. Zhu, R. Xu, J. Song, M. 张, Y. Li, Y. Wu 和 D. Guo. Deepseekmath: 在开放语言模型中突破数学推理的极限。arXiv 预印本 arXiv: 2402.03300, 2024。

N. 沙泽尔。快速变压器解码: 您只需要一个写入头。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 和 J. Dean. 极其庞大的神经网络: 稀疏门控的专家混合层。第五届国际学习表征会议, ICLR 2017. OpenReview.net, 2017. URL <https://openreview.net/forum?id=B1ckMDqlg>.

J. Su, M. Ahmed, Y. Lu, S. Pan, W. Bo 和 Y. Liu. Roformer: 具有旋转位置嵌入的增强型变压器。神经计算, 568: 127063, 2024。

K. Sun, D. Yu, D. Yu 和 C. Cardie. 研究挑战中文机器阅读理解的先验知识, 2019。

M. Suzgun, N. Scales, N. Schärli, S. Gehrmann, Y. Tay, H. W. Chung, A. Chowdhery, Q. V. Le, E. H. Chi, D. Zhou, 等人。具有挑战性的大工作任务以及思维链能否解决它们。arXiv 预印本 arXiv:2210.09261, 2022。

A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. 凯撒和 I. 波罗-苏欣。您所需要的就是关注。神经信息处理系统的进展, 30, 2017。

J. Wei, Y. Tay, R. Bommasani, C. Raffel, B. Zoph, S. Borgeaud, D. Yogatama, M. Bosma, D. Zhou, D. 梅茨勒等人。大型语言模型的新兴能力。arXiv 预印本 arXiv: 2206.07682, 2022。

T. Wei, J. Luan, W. Liu, S. Dong 和 B. Wang. Cmath: 你的语言模型能通过中国小学数学考试吗?, 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.刘, Z.赵, Q.赵, C.Yue, X.Zhang, Z.Yang, K.Richardson, 和Z.Lan。线索: 汉语理解评估基准。D. Scott、N. Bel 和 C. Zong, 编辑, 第 28 届国际计算语言学会会议论文集, COLING2020, 西班牙巴塞罗那(在线), 2020 年 12 月 8-13 日, 第 4762-4772 页。国际计算语言学委员会, 2020。doi: 10.18653/V1/2020.COLING-MAIN.419。网址 <https://doi.org/10.18653/v1/2020.coling-main.419>。
- A. Young、B. Chen、C. Li、C. Huang、G. Zhang、G. Zhu、H. Li、J. Zhu、J. Chen、J. Chang 等人。Yi: 01 的开放基础模型。人工智能。arXiv 预印本 arXiv:2403.04652, 2024。
- R. Zellers、A. Holtzman、Y. Bisk、A. Farhadi 和 Y. Choi。HellaSwag: 机器真的能完成你的句子吗? A. Korhonen、D. R. Traum 和 L. Màrquez, 编辑, 第 57 届计算语言学协会会议记录, ACL 2019, 意大利佛罗伦萨, 2019 年 7 月 28 日至 8 月 2 日, 第 1 卷: 长论文, 第 4791-4800 页。计算协会 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 和 B.卡西克奇。Atom: 低位量化, 实现高效、准确的 LLM 服务。钴RR, abs/2310.19102, 2023。URL <https://doi.org/10.48550/arXiv.2310.19102>。
- C. 郑、M. 黄和 A. 孙。Chid: 用于完形填空测试的大规模汉语成语数据集。在一个。Korhonen、D. R. Traum 和 L. Màrquez, 编辑, 第 57 届计算语言学协会会议记录, ACL 2019, 意大利佛罗伦萨, 2019 年 7 月 28 日至 8 月 2 日, 第 1 卷: 长论文, 第 778-787 页。计算语言学协会, 2019。doi: 10.18653/V1/P19-1075。URL <https://doi.org/10.18653/v1/p19-1075>。
- L.郑, W.-L. 蒋Y.盛, 庄S.庄, Z.吴, Y.庄, Z.林, Z.李, D.Li, E.P.Xing, H.张, J.E.冈萨雷斯, I.斯托伊卡。使用 mt-bench 和 chatbotarena 来评判 llm-as-a-judge, 2023 年。
- W.Zhong、R.Cui、Y.Guo、Y.Liang、S.Lu、Y.Wang、A.Saied、W.Chen 和 N.Duan。AGIEval: 以人类为中心的评估基础模型的基准。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, 等。Lima: 对于一致性来说, 少即是多。神经信息处理系统的进展, 36, 2024。
- J. Zhou、T. Lu、S. Mishra、S. Brahma、S. Basu、Y. Luan、D. Zhou 和 L. Hou。大型语言模型的指令跟踪评估。arXiv 预印本 arXiv:2311.07911, 2023。

## Appendix

### A. 贡献和致谢

#### Research & Engineering

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  
Zhenwen Hao  
Zhihong Shao  
Zhuoshu Li  
Zihan Wang  
Zihui Gu  
Zilin Li  
Ziwei Xie

#### **Data Annotation**

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

#### **Business & Compliance**

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

在每个角色中，作者按名字字母顺序列出。特别是 Huazuo Gau 和 Wangding Zeng 在 MLA 架构的研究中做出了关键的创新。此外，我们还要感谢 Jialin Su 对位置嵌入的有益讨论。我们感谢所有为 DeepSeek-V2 做出贡献但没有做出贡献的人。DeepSeek认为，创新、新颖和好奇心是通向AGI之路的关键。

## B. DeepSeek-V2-Lite: 配备 MLA 和 DeepSeekMoE 的 16B 型号

### B.1. 型号说明

架构。DeepSeek-V2-Lite有27层，隐藏维度为2048。它也采用MLA，有16个注意力头，每个头的维度为128。它的KV压缩维度为512，但与DeepSeek-V2略有不同，它没有压缩查询。对于解耦的查询和密钥，它的每头维度为 64。

DeepSeek-V2-Lite 还使用 DeepSeekMoE，除了第一层之外的所有 FFN 都替换为 MoE 层。每个 MoE 层由 2 个共享层组成专家和64个路由专家，其中每个专家的中间隐藏维度为1408。在路由专家中，每个令牌将激活6个专家。在此配置下，DeepSeek-V2-Lite总共包含15.7B个参数，其中每个令牌激活2.4B个参数。

	Benchmark	DeepSeek 7B	DeepSeekMoE 16B	DeepSeek-V2-Lite
	Architecture	MHA+Dense	MHA+MoE	MLA+MoE
	Context Length	4K	4K	32K
	# Activated Params	6.9B	2.8B	2.4B
	# Total Params	6.9B	16.4B	15.7B
	# Training Tokens	2T	2T	5.7T
English	MMLU	48.2	45.0	<b>58.3</b>
	BBH	39.5	38.9	<b>44.1</b>
	TriviaQA	59.7	<b>64.8</b>	64.2
	NaturalQuestions	22.2	25.5	<b>26.0</b>
	ARC-Easy	67.9	68.1	<b>70.9</b>
	ARC-Challenge	48.1	49.8	<b>51.2</b>
	AGIEval	26.4	17.4	<b>33.2</b>
Code	HumanEval	26.2	26.8	<b>29.9</b>
	MBPP	39.0	39.2	<b>43.2</b>
Math	GSM8K	17.4	18.8	<b>41.1</b>
	MATH	3.3	4.3	<b>17.1</b>
	CMATH	34.5	40.4	<b>58.4</b>
Chinese	CLUEWSC	73.1	72.1	<b>74.3</b>
	C-Eval	45.0	40.6	<b>60.3</b>
	CMMLU	47.2	42.5	<b>64.3</b>

表 6 | DeepSeek-V2-Lite、DeepSeekMoE 16B 和 DeepSeek 7B 的性能。

培训细节。DeepSeek-V2-Lite也在DeepSeek-V2相同的预训练语料库上从头开始训练，没有受到任何SFT数据的污染。它使用 AdamW 优化器，超参数设置为  $\beta_1 = 0.9$ 、 $\beta_2 = 0.95$  和  $\text{weight\_decay} = 0.1$ 。学习率是使用预热和逐步衰减策略来安排的。最初，学习率在前 2K 步中从 0 线性增加到最大值。随后，在训练大约 80% 的标记后，学习率乘以 0.316，在训练大约 90% 的标记后，再次乘以 0.316。最大学习率设置为  $4.2 \times 10^{-4}$ ，梯度裁剪范数设置为 1.0。我们没有对其采用批量大小调度策略，而是使用 4608 个序列的恒定批量大小进行训练。在预训练期间，我们设置最大序列

长度为 4K，并在 5.7T 令牌上训练 DeepSeek-V2-Lite。我们利用管道并行性将其不同层部署在不同的设备上，但对于每一层，所有专家都将部署在同一设备上。因此，我们只采用了小的专家级平衡损失，即  $\alpha_1 = 0.001$ ，而不采用设备级平衡损失和通信平衡损失。预训练后，我们还对 DeepSeek-V2-Lite 进行长上下文扩展和 SFT，得到名为 DeepSeek-V2-Lite Chat 的聊天模型。

Benchmark		DeepSeek 7B Chat	DeepSeekMoE 16B Chat	DeepSeek-V2-Lite Chat
Architecture		MHA+Dense	MHA+MoE	MLA+MoE
Context Length		4K	4K	32K
# Activated Params		6.9B	2.8B	2.4B
# Total Params		6.9B	16.4B	15.7B
# Training Tokens		2T	2T	5.7T
English	MMLU	49.7	47.2	<b>55.7</b>
	BBH	43.1	42.2	<b>48.1</b>
	TriviaQA	59.5	63.3	<b>65.2</b>
	NaturalQuestions	32.7	35.1	<b>35.5</b>
	ARC-Easy	70.2	69.9	<b>74.3</b>
	ARC-Challenge	50.2	50.0	<b>51.5</b>
	AGIEval	17.6	19.7	<b>42.8</b>
Code	HumanEval	45.1	45.7	<b>57.3</b>
	MBPP	39.0	<b>46.2</b>	45.8
Math	GSM8K	62.6	62.2	<b>72.0</b>
	MATH	14.7	15.2	<b>27.9</b>
	CMATH	66.4	67.9	<b>71.7</b>
Chinese	CLUEWSC	66.2	68.2	<b>80.0</b>
	C-Eval	44.7	40.0	<b>60.1</b>
	CMMLU	51.2	49.3	<b>62.5</b>

表 7 | DeepSeek-V2-Lite Chat、DeepSeekMoE 16B Chat 和 DeepSeek 7BChat 的性能。

## B.2. 绩效评估

基础模型。我们评估了 DeepSeek-V2-Lite 的性能，并将其与表 6 中我们之前的小尺寸基础模型进行了比较。DeepSeek-V2-Lite 展现了压倒性的性能优势，特别是在推理、编码和数学方面。

聊天模型。我们评估了 DeepSeek-V2-Lite Chat 的性能，并将其与我们之前的小规模聊天模型进行了比较，如表 7 所示。DeepSeek-V2-Lite 的性能也大幅优于我们之前的小规模聊天模型。

### C. MLA 的完整公式

为了演示MLA的完整计算过程，我们提供了其完整的公式如下：

$$\mathbf{c}_t^Q = W^{DQ} \mathbf{h}_t, \quad (37)$$

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

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

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

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

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

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

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

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

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

$$\mathbf{u}_t = W^O [\mathbf{o}_{t,1}; \mathbf{o}_{t,2}; \dots; \mathbf{o}_{t,n_h}], \quad (47)$$

其中蓝色的盒装向量需要缓存以供生成。在推理过程中，朴素公式需要从  $\mathbf{c}_t^{KV}$  中恢复  $\mathbf{k}_t^C$  和  $\mathbf{v}_t^C$  以引起注意。幸运的是，由于矩阵乘法的结合律，我们可以将  $W^{UK}$  吸收为  $W^{UQ}$ ，并将  $W^{UV}$  吸收为  $W^O$ 。因此，我们不需要为每个查询计算键和值。通过这种优化，我们避免了推理期间重新计算  $\mathbf{k}_t^C$  和  $\mathbf{v}_t^C$  的计算开销。

### D. 消除注意力机制

#### D.1. MHA、GQA 和 MQA 消融

我们在表 8 中显示了具有 MHA、GQA 和 MQA 的 7B 密集模型在四个硬基准上的评估结果。所有这三个模型都在 1.33T 令牌上进行训练，并且除了注意力机制之外共享相同的架构。另外，为了公平比较，我们通过调整层数将它们的参数数量调整到 7B 左右。从表中我们可以发现，MHA 在这些基准测试上比 GQA 和 MQA 表现出显著的优势。

#### D.2. MLA 和 MHA 的比较

在表 9 中，我们显示了分别配备 MLA 和 MHA 的 MoE 模型在四个硬基准上的评估结果。为了得到可靠的结论，我们在两个尺度上训练和评估模型。两个小型 MoE 模型总共包含大约 16B 个参数，我们在 1.33T 代币上训练它们。两个大型 MoE 模型总共包含约 250B 个参数，并在 420B 个令牌上训练它们。此外，除了注意力机制之外，两个小型 MoE 模型和两个大型 MoE 模型分别共享相同的架构。从表中我们可以观察到 MLA 表现出比 MHA 更好的性能。更重要的是，MLA 需要的 KV 缓存量比 MHA 少得多（小型 MoE 模型为 14%，大型 MoE 模型为 4%）。

Benchmark (Metric)	# Shots	Dense 7B w/ MQA	Dense 7B w/ GQA (8 Groups)	Dense 7B w/ MHA
# Params	-	7.1B	6.9B	6.9B
BBH (EM)	3-shot	33.2	35.6	<b>37.0</b>
MMLU (Acc.)	5-shot	37.9	41.2	<b>45.2</b>
C-Eval (Acc.)	5-shot	30.0	37.7	<b>42.9</b>
CMMLU (Acc.)	5-shot	34.6	38.4	<b>43.5</b>

表 8 | 7B 密集模型分别与 MHA、GQA 和 MQA 的比较。MHA 在硬基准测试中表现出优于 GQA 和 MQA 的显著优势。

Benchmark (Metric)	# Shots	Small MoE w/ MHA	Small MoE w/ MLA	Large MoE w/ MHA	Large MoE w/ MLA
# Activated Params	-	2.5B	2.4B	25.0B	21.5B
# Total Params	-	15.8B	15.7B	250.8B	247.4B
KV Cache per Token (# Element)	-	110.6K	15.6K	860.2K	34.6K
BBH (EM)	3-shot	37.9	<b>39.0</b>	46.6	<b>50.7</b>
MMLU (Acc.)	5-shot	48.7	<b>50.0</b>	57.5	<b>59.0</b>
C-Eval (Acc.)	5-shot	<b>51.6</b>	50.9	57.9	<b>59.2</b>
CMMLU (Acc.)	5-shot	52.3	<b>53.4</b>	60.7	<b>62.5</b>

表 9 | MLA 和 MHA 在硬基准上的比较。DeepSeek-V2 显示出比 MHA 更好的性能，但需要的 KV 缓存量要少得多。

#### E. 关于预训练数据去偏的讨论

在训练前数据准备过程中，我们识别并过滤掉有争议的内容，例如受地区文化影响的价值观，以避免我们的模型对这些有争议的话题表现出不必要的主观偏见。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稍差。例如，在 MMLU 上进行评估时，尽管 DeepSeek-V2 与 Mixtral 8x22B 等竞争对手相比，在大多数测试集上都取得了相当或更好的性能，但它在人性道德子集上仍然落后，这主要与美国价值观相关。

此外，我们对此子集进行手动分析。三位受过良好教育的人类注释者对 MMLU 人类道德子集中的 420 个道德场景进行独立注释。然后，我们计算它们的注释和真实标签之间的一致性。如表 10 所示，三个人工注释者和真实标签之间的一致性较低。因此，我们将 DeepSeek-V2 在这些值敏感测试集上的异常表现归因于我们对预训练语料库去偏的努力。

#### F. 对数学和代码的附加评估

评估采用的是 SC-Math6 语料库，该语料库包含数千道中国数学问题。DeepSeek-V2 Chat (RL) 的表现优于所有中国 LLM，包括开源和闭源模型。

我们在图 5 中进一步分享了 HumanEval 和 LiveCodeBench 上的更多结果，其中



Agreement	Ground-Truth Label	Annotator 1	Annotator 2	Annotator 3
Ground-Truth Label	100.0%	66.7%	59.8%	42.1%
Annotator 1	66.7%	100.0%	57.9%	69.0%
Annotator 2	59.8%	57.9%	100.0%	65.5%
Annotator 3	42.1%	69.0%	65.5%	100.0%

表 10 | 三名受过良好教育的人类注释者对 MMLU 人类道德子集的 420 个道德场景进行独立注释，DeepSeek-V2 及其竞争模型在这些场景上表现出性能不一致。三个注释者和真实标签之间的一致性较低。这表明，根据特定的区域文化，人性道德子集的答案可能会存在争议。

Model Name	R Level	Comp. Score	Reas. Steps Score	OvrAcc Score
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	<b>84.54</b>
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 模型推理水平。“R Level”代表推理水平，“Comp.Score”代表综合分数，“Reas.Score”代表综合分数。“步骤分数”代表推理步骤分数，“OvrAcc 分数”代表总体准确性分数。

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中表现出了相当的熟练程度，取得了Pass@1的分数，甚至超越了一些巨型模型。这一性能凸显了DeepSeek-V2 Chat (RL) 在处理实时编码任务方面的强大能力。

## G. 评估形式

我们分别在表 12-37 中列出了每个基准的评估格式。

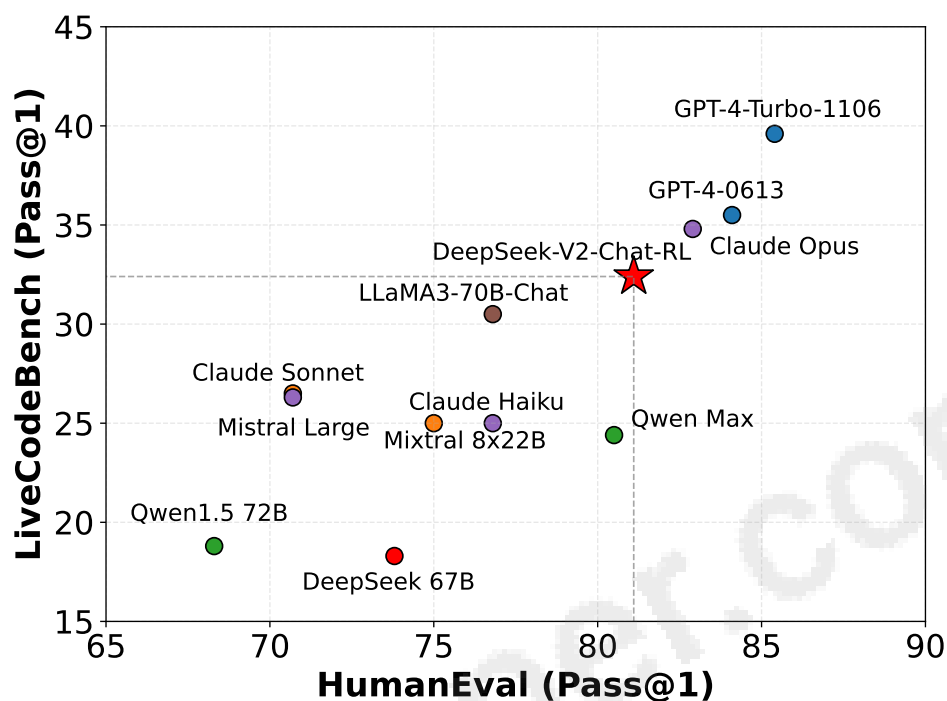


图5| HumanEval 和 LiveCodeBench 上的评估结果。Live-CodeBench的问题选自2023年9月1日至2024年4月1日期间。

#### PROMPT

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 的示例。

---

提示问题：圆柱形容器中的样品具有圆柱形形状和固定体积。样品\_A的物质状态。必须是固体 B. 可以是固体也可以是液体C.必须是液体D.可以是液体也可以是气体答案：B

问题：\_A 处的声速通常最大。固体含量最低，液体含量最低 B.固体含量最低，气体含量最低 C.气体中含量最低，液体中含量最低 D.气体和固体含量最低答案：B

问题：当油和水混合在一起时，它们形成\_A。气体 B.固体C.化合物 D.悬吊答案：D

问题：白天温度为  $3^{\circ}\text{C}$  时，将一个装有液态水的容器放在室外。晚上室外温度降至  $-2^{\circ}\text{C}$ 。这种温度变化很可能导致水变成\_A。凝结 B.蒸发 C.保持液态 D.成为坚实的答案：

---

表 13 | ARC 的一个例子。

---

PROMPTE评估随机布尔表达式的结果。

Q:  $\text{not}((\text{not not True}))$  isA: 让我们一步一步思考。记住 (i) 括号内的表达式总是首先被求值, 并且 (ii) 从最高优先级到最低优先级的运算顺序是“not”, “和”, “或”, 分别。我们首先将表达式“Z”简化如下: “ $Z = \text{not}((\text{not not True})) = \text{not}((A))$ ”, 其中“ $A = \text{not not True}$ ”。让我们评估 A:  $A = \text{not not True} = \text{not}(\text{not True}) = \text{not False} = \text{True}$ 。代入 A, 我们得到:  $Z = \text{not}((A)) = \text{not}((\text{True})) = \text{not True} = \text{False}$ 。所以答案是错误的。

问: True 和 False 以及 not True 和 True isA: 让我们一步一步思考。请记住, (i) 括号内的表达式始终首先计算, 并且 (ii) 从最高优先级到最低优先级的运算顺序是“not”, “和”, “或”, 分别。我们首先将表达式“Z”简化如下: “ $Z = \text{True 和 False 并且 not True 和 True} = A \text{ 和 B}$ ”, 其中“ $A = \text{True 和 False}$ ”并且“ $B = \text{not True 和 True}$ ”。让我们评估 A:  $A = \text{True, False} = \text{False}$ 。让我们评估 B:  $B = \text{not True and True} = \text{not}(\text{True and True}) = \text{not}(\text{True}) = \text{False}$ 。代入 A 和 B, 我们得到:  $Z = A \text{ and B} = \text{False 并且 False} = \text{False}$ 。所以答案是错误的。

Q:  $\text{not not}(\text{not}(\text{False}))$  isA: 让我们一步一步思考。记住 (i) 括号内的表达式总是首先被求值, 并且 (ii) 从最高优先级到最低优先级的运算顺序是“not”, “和”, “或”, 分别。我们首先将这个表达式“Z”简化如下: “ $Z = \text{not not}(\text{not}(\text{False})) = \text{not not}(A)$ ”, 其中“ $A = \text{not}(\text{False})$ ”。让我们评估 A:  $A = \text{not}(\text{False}) = \text{not False} = \text{True}$ 。代入 A, 我们得到:  $Z = \text{not not}(A) = \text{not not}(\text{True}) = \text{not not False} = \text{True}$ 。所以答案是正确的。

问: 错与错以及错与否是错是A: 让我们一步步思考。

---

表 14 | BBH 的一个例子。

---

**PROMPT**

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呕气

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：

---

**OPTIONS**

- A
  - B
  - C
  - D
- 

表 15 | C-Eval 的示例。

---

**PROMPT**

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：

---

**OPTIONS**

- 商店
  - 饭店
  - 医院
  - 教室
- 

表 16 | C3 的一个例子。

---

**PROMPT**

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：

---

**OPTIONS**

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

表 17 | CCPM 的一个例子。

---

**PROMPT**

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。

---

表 18 | CMATH 的一个例子。

---

**PROMPT**

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：

---

**OPTIONS**

- A
  - B
  - C
  - D
- 

表 19 | CMMLU 的一个示例。



---

**PROMPT**

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 的一个例子。

---

PROMPTPassage：该城市的平均年龄为 22.1 岁。10.1%的居民年龄在18岁以下； 56.2%年龄在18岁至24岁之间； 16.1%为25岁至44岁； 10.5%为45岁至64岁； 65岁以上占7%。全市性别构成成为男性64.3%，女性35.7%。

根据以上段落回答下列问题，如需计算请仔细计算。问：不是在 25 到 44 之间的百分比是多少？答：答案类型为数字。所以根据上面的段落，答案是83.9。

问：不是 25 到 44 的百分比是多少？答：答案类型是数字。所以根据上面的段落，答案是

---

表 21 | DROP 的一个例子。

---

**PROMPT**

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

---

**OPTIONS**

- 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 的一个例子。

<b>PROMPT</b>
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 的一个示例。

---

提示Q: Max 可以在 40 分钟内修剪草坪。如果他给草坪施肥需要两倍的时间, 那么他割草和给草坪施肥需要多长时间? 答: 让我们一步步思考。给草坪施肥最多需要  $2 * 40$  分钟 = 80 分钟。总共, Max 需要  $80$  分钟 +  $40$  分钟 =  $120$  分钟来修剪草坪和施肥。答案是120。

问: 百吉饼每个售价 2.25 美元, 一打百吉饼售价 24 美元。一次买一打百吉饼可以节省多少钱? 答: 让我们一步步思考。每张售价  $2.25 * 100 = 225$  美分。按批量价格计算, 每件为  $24 / 12 = 2$  美元。每张售价  $2 * 100 = 200$  美分。  $225 - 200 =$  每个百吉饼节省 25 美分。答案是25。

问: 蒂姆今年 5 岁。他的表弟隆美尔的年龄是他的三倍。他的另一个表弟珍妮比隆美尔大两岁。Timthan Jenny 年轻了多少岁? A: 我们一步步想。隆美尔的年龄是  $5 * 3 = 15$  岁。珍妮  $15 + 2 = 17$  岁。所以, 蒂姆比珍妮小  $17 - 5 = 12$  岁。答案是12。

问: 学校有 14 名男生和 10 名女生。如果有 4 个男孩和 3 个女孩辍学, 还剩下多少个男孩和女孩? A: 我们一步步想。剩下 14 个男孩 - 4 个男孩 = 10 个男孩。剩下 10 个女孩 - 3 个女孩 = 7 个女孩。总共还剩下 10 个男孩 + 7 个女孩 = 17 个男孩和女孩。答案是17。

问: 建造一座鸟舍需要 7 块木板和 20 颗钉子。如果 1 个钉子的成本为 0.05, 一块木板的成本为 3, 那么建造 4 个鸟舍的成本是多少美元? 答: 让我们一步步思考。一间鸟舍的木板成本为  $7 * 3 = 21$ 。每个鸟舍的钉子成本为  $20 * 0.05 = 1$ 。因此, 建造一个鸟舍需要  $21 + 1 = 22$ 。因此建造 4 个鸟舍的成本为  $4 * 22 = 88$ 。答案是 88。

问: 丹尼带了 3 个西瓜去参加他的家庭野餐。他把每个西瓜切成 10 片。他的姐姐带了 1 个西瓜去参加家庭野餐, 她把西瓜切成了 15 片。野餐时总共有多少片西瓜? A: 我们一步步想。丹尼有  $3 * 10 = 30$  片西瓜片。他妹妹有  $1 * 15 = 15$  片西瓜片。总共有  $30 + 15 = 45$  片西瓜片。答案是45。

问: 安吉拉是纽约的一名自行车送信员。她需要运送的包裹数量是餐食的 8 倍。如果她需要送 27 份餐食和包裹, 她总共送多少餐? 答: 我们一步步想。令  $p$  为安吉拉递送的包裹数量,  $m$  为餐食数量。我们知道  $p + m = 27$  且  $p = 8m$ 。将第二个方程代入第一个方程, 得到  $8m + m = 27$ 。组合类似项, 得到  $9m = 27$ 。两边除以 9, 得到  $m = 3$ 。答案是 3。

问: Cori 今天 3 岁了。5年后, 她的年龄将是她姨妈的三分之一。她姨妈今天多大了? A: 我们一步步想。5年后, Cori就  $3 + 5 = 8$  岁。5年后, Cori 的阿姨将是  $8 * 3 = 24$  岁。今天, 她的阿姨  $24 - 5 = 19$  岁。答案是19。

问: 因陀罗的名字中有 6 个字母。她姐姐的名字比因陀罗名字的一半多了 4 个字母。因陀罗和她姐姐的名字中有多少个字母? 答: 让我们一步步想。

---

提示弹钢琴：一个男人坐在钢琴前。他

---

选项- 用手和脸弹钢琴。- 开始在钢琴上  
弹奏 timbaland 的歌曲。- 缓慢弹奏，然  
后停下来打响指。- 在他面前弹奏一首歌  
曲。

---

表 25 | HellaSwag 的一个示例。

---

**PROMPT**

defstarts\_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 的一个示例。

---

PROMPT问题：求表达式 $\frac{\sqrt{x-2}}{\sqrt{5-x}}$ 的定义域

解：每个平方根内的表达式必须是非负的。因此， $x-2 \geq 0$ ，所以 $x \geq 2$ ， $5-x \geq 0$ ，所以 $x \leq 5$ 。另外，分母不能等于 0，因此  $5-x > 0$ ，即  $x < 5$ 。因此，表达式的定义域为  $\boxed{[2,5)}$ 。最终答案：答案是 $[2,5)$ 。我希望这是正确的。

Problem:

If  $\det \mathbf{A} = 2$  and  $\det \mathbf{B} = 12$ , then find  $\det (\mathbf{A} \mathbf{B})$ .

解： $\det (\mathbf{A} \mathbf{B}) = (\det \mathbf{A})(\det \mathbf{B}) = (2)(12) = \boxed{24}$ 。最终答案：最终答案是 $24$ 。我希望这是正确的。

问题：Terrell 通常会举起两个 20 磅的重物 12 次。如果他使用两个 15 磅的重物，特雷尔必须举起它们多少次才能举起相同的总重量？

解决方案：如果特雷尔举起两个 20 磅的重物 12 次，他总共举起  $2 \cdot 12 \cdot 20 = 480$  磅的重量。如果他举起两个 15 磅的重物  $n$  次，他将总共举起  $2 \cdot 15 \cdot n = 30n$  磅的重量。将其等同于 480 磅，我们可以求解  $n$ ：  
$$30n = 480 \implies n = 480/30 = \boxed{16}$$
最终答案：最终答案是  $16$ 。我希望这是正确的。

问题：如果方程组

$$\begin{cases} 6x-4y=a, \\ 6y-9x=b. \end{cases}$$
有一个解 $(x, y)$ ，其中 $x$ 和 $y$ 都是非零，找到 $\frac{a}{b}$ ，假设  $b \neq 0$ 。

解：如果我们将第一个方程乘以 $-\frac{3}{2}$ ，我们得到

$6y-9x=-\frac{3}{2}a$ 。因为我们也知道 $6y-9x=b$ ，所以我们有

$-\frac{3}{2}a=b \implies \frac{a}{b}=-\frac{2}{3}$ 。最终答案：最终答案是 $-\frac{2}{3}$ 。我希望这是正确的。

问题：评估 $\log_{21}$ 。

Solution:

---

---

提示 您是一名专业的 Python 程序员，您的任务是：编写一个函数以从给定的两个元组列表中查找相似的元素。您的代码应该通过这些测试：

断言相似元素((3, 4, 5, 6),(5, 7, 4, 10)) == (4, 5)断言相似元素((1, 2, 3, 4),(5, 4, 3, 7)) == (3, 4)断言相似元素((11, 12, 14, 13),(17, 15, 14, 13)) == (13, 14)[开始]def 相似元素(test\_tup1, test\_tup2):res = tuple(set(test\_tup1) & set(test\_tup2))return (res)[DONE]你是一名专家 Python 程序员，你的任务是：编写一个python函数来识别非素数。您的代码应该通过这些测试：

断言 is\_not\_prime(2) == False 断言 is\_not\_prime(10) == True 断言 is\_not\_prime(35) == True[开始]import mathdef is\_not\_prime(n):result = Falsefor i in range(2,int(math.sqrt(n)) + 1):if n % i == 0:result = Truereturn result[DONE]您是一名专业的 Python 程序员，您的任务是：编写一个函数，使用堆队列算法从给定的数字列表中查找最大整数。您的代码应该通过这些测试：

断言 heap\_queue\_largest( [25, 35, 22, 85, 14, 65, 75, 22, 58],3)==[85, 75, 65]断言 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 hqdef heap\_queue\_largest(nums,n):largest\_nums = hq.nlargest(n, nums)returnlargest\_nums[DONE]您是一名专业的 Python 程序员，这是您的任务：编写一个函数来返回一个数的所有约数的总和。您的代码应该通过这些测试：

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

---

表 28 | MBPP 的一个例子。

---

提示以下是有关杂项的多项选择题（附答案）。

一辆标准汽车有多少个车轴？

A.一。二。四。八、答案： B

摇滚传奇人物 CheapTrick 1979 年现场专辑的标题中提到了什么地方？布达佩斯 B.武道馆 C.不丹D.英国答案： B

谁是赢得 NBA 扣篮大赛的最矮的人？安东尼·“马铃薯”·韦伯。迈克尔·乔丹 (Michael 'Air' Jordan)蒂龙·“马格西”·博格斯。朱利叶斯·“J”·欧文博士 答案： A

光合作用过程中产生什么？ A.氢。尼龙。氧。光 答案： C

以下哪一首歌曲是摇滚乐队 The Police 的十大热门歌曲？ '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

三个臭皮匠中哪一个与其他人没有血缘关系？ A.莫伊B。拉里·C.卷毛 D.谢普回答：

---

**OPTIONS**

- A
  - B
  - C
  - D
- 

表 29 | MMLU 的一个示例。

---

**PROMPT**

Answer these questions:

Q: Who is hosting the fifa world cup in 2022?

A: Qatar

Q: Who won the first women 's fifa world cup?

A: United States

Q: When did miami vice go off the air?

A: 1989

Q: Who wrote the song shout to the lord?

A: Darlene Zschech

Q: Who was thrown in the lion 's den?

A: Daniel

Q: What is the meaning of the name habib?

A:

---

表 30 | 自然问题的一个例子。

---

PROMPTA 女士注意到每年秋天她都会感到沮丧，并想知道为什么。一位朋友向她建议，也许随着季节从温暖到寒冷而发生的某些变化可能会对她产生影响。当被问及这些变化的例子时，这位朋友引用了

---

选项-鲜花盛开-  
草变成棕色-树木  
生长-鲜花盛开

---

表 31 | OpenBookQA 的示例。

---

提示 为了更容易按下位于机器下方的垃圾处理机的重置按钮，

---

选项 - 在柜子地板上放置一面壁镜 - 在垃圾处理机下方放置一面手持镜

---

表 32 | PIQA 的一个例子。



---

提示文章：当您阅读一篇文章时，如果您能弄清楚作者如何将想法组合在一起，您就会更好地理解并记住它。有时，作者通过提出问题然后回答问题来将想法组合在一起。例如，如果这篇文章是关于土拨鼠的，那么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，作者会在文章中写出她的问题。这些问题给你信号。它们告诉你作者接下来要写什么。通常作者脑子里有一个问题但她不会给你写出来。你必须自己解决她的问题。这里有一个示例阅读供你练习这个方法。蚯蚓你知道蚯蚓有多少种吗？世界上大约有1800种！它们的颜色可以是棕色的、紫色的、绿色的。它们的长度可以小至3厘米，大至3米。观赏蚯蚓的最佳时间是晚上，尤其是凉爽、潮湿的夜晚。那时它们从洞里爬出来。挖洞寻找食物。蚯蚓不喜欢在阳光下。那是因为它们通过皮肤呼吸，如果皮肤太干燥，它们就无法呼吸。如果下很多雨，蚯蚓必须从土里出来，因为它们蚯蚓没有眼睛，怎么知道天黑了？它们的皮肤上有对光敏感的特殊地方。这些斑点可以分辨出是亮还是暗。如果你在晚上用手电筒照射蚯蚓，它很快就会消失在地下。蚯蚓也没有耳朵，但它们能听到声音通过感受地球上的运动。如果你想像蚯蚓一样听到声音，那就躺在地上，用手指捂住耳朵。然后让一个朋友在你附近跺脚。这就是蚯蚓感受鸟儿、人行走、鼯鼠挖掘的方式，在他们附近。蚯蚓很有用。农民和园丁喜欢在他们的土地上有许多蚯蚓，因为蚯蚓在挖掘时有助于改善土壤。挖掘可以保持土壤疏松和通风。一年内，蚯蚓可以堆积多达 23,000 公斤的铸件在大约一个足球场大小的区域。

问：读《蚯蚓》的目的是什么？

A：把作者的想法付诸实践。

问：文中哪个问题不能回答？

A：人为什么能像蚯蚓一样听声音？

问：根据这段话，你如何更好地了解蚯蚓？

答：阅读时要解决作者脑子里的所有问题。

问：这篇文章的最佳标题是什么？

A:

---

选项 - 一种帮助理解的方法 - 一种  
实践新想法的方法 - 一种学习成为  
明智作家的方法 - 一种更清楚地了  
解蠕虫的方法 9

---

请回答以下问题：问：Jayhawker 一词适用于美国某个州的反奴隶制激进组织，这些组织与密苏里州的支持奴隶制派系发生冲突。这是哪个州，有时被称为松鸦鹰州？答：堪斯州。问：哪位瑞典 DJ 和唱片制作人在 2013 年凭借《Wake Me Up》成为英国排名第一的单曲？答：蒂姆·伯格林问：谁是谢菲尔德哈勒姆的议员答：尼克·克莱格问：田纳西州诉约翰·托马斯·斯科普斯案是一个举国瞩目的案件，于 1925 年 7 月 21 日结束，陪审团认定斯科普斯先生犯有教导什么罪？答：《物种生存》问：什么动画片系列中有一个名叫 Little My 的角色？A: MuumiQ: “哪位英国模特，长着短发，雌雄同体，原名 LesleyHornby, 1966 年被 Nigel Davies 发现，当时她 16 岁，体重 6 英石（41 公斤，91 磅），并凭借 Mary Quant 打造的高级时尚造型成为“66 年面孔”？” A:

---

表 34 | TriviaQA 的一个示例。

---

<b>PREFIXES</b>
- So Monica
- So Jessica

---

<b>COMPLETION</b>
avoids eating carrots for their eye health because Emily needs good eyesight while Monica doesn't.

---

表 35 | 威诺格兰德的一个例子。请注意，WinoGrande 有多个前缀，只有一个补全，我们选择补全复杂度最低的预测前缀。

---

提示您将获得一个函数  $f$  和一个形式为  $f(??) == \text{output}$  的输出。查找任何输入，以便在输入上执行  $f$  会产生给定的输出。可能有多个答案，但你应该只输出一个。在 `[ANSWER]` 和 `[/ANSWER]` 标记中，使用一个这样的输入来完成断言，该输入将在执行函数时产生输出。

```
[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.from
resultassert f(??) == {1:
None , 2: 无}[/PYTHON]
[答案]
```

---

表 36 | CRUXEval-I 的一个示例。

---

**Prompt**

You are given a Python function and an assertion containing an input to the function. Complete the assertion with a literal (no unsimplified expressions, no function calls) containing the output when executing the provided code on the given input, even if the function is incorrect or incomplete. Do NOT output any extra information. Provide the full assertion with the correct output in [ANSWER] and [/ANSWER] tags, following the examples.

```
[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 的一个示例。