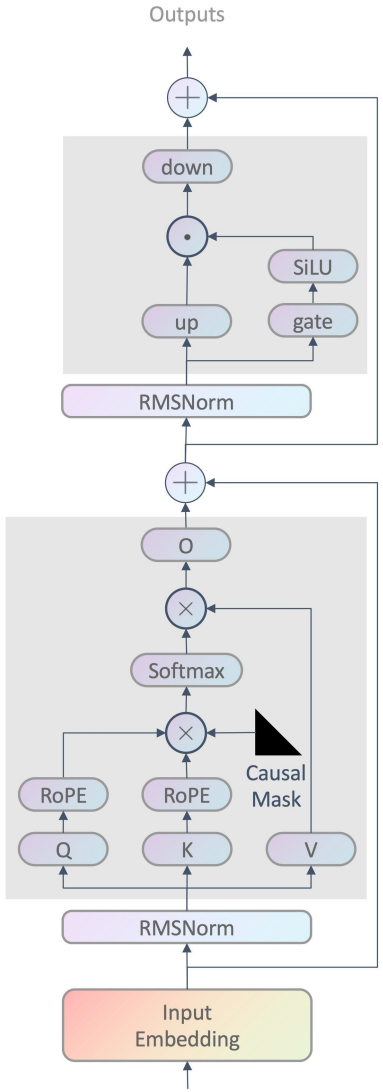
DeepSeek LLM: Scaling Open-Source Language Models with Longtermism

Scaling laws for LLM

Architecture



延用了LLaMA的结构，在一个标准的Transfromer Block上做了以下的修改:

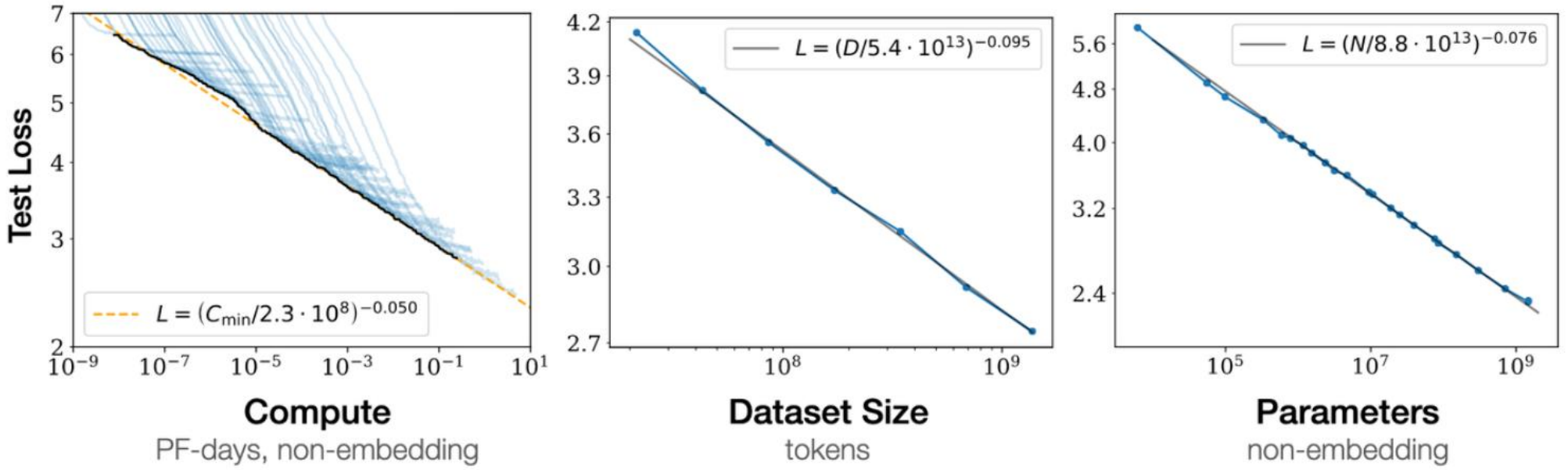
- RoPE
- RMSNorm
- SwiGLU
- GQA

Scaling Law

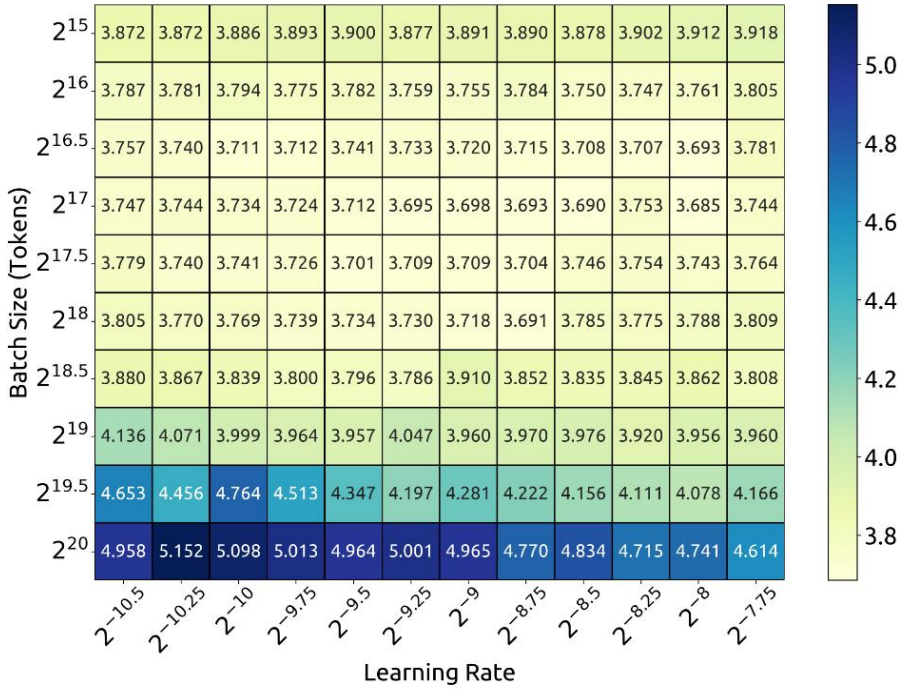
$$C \approx 6ND$$

最优效率

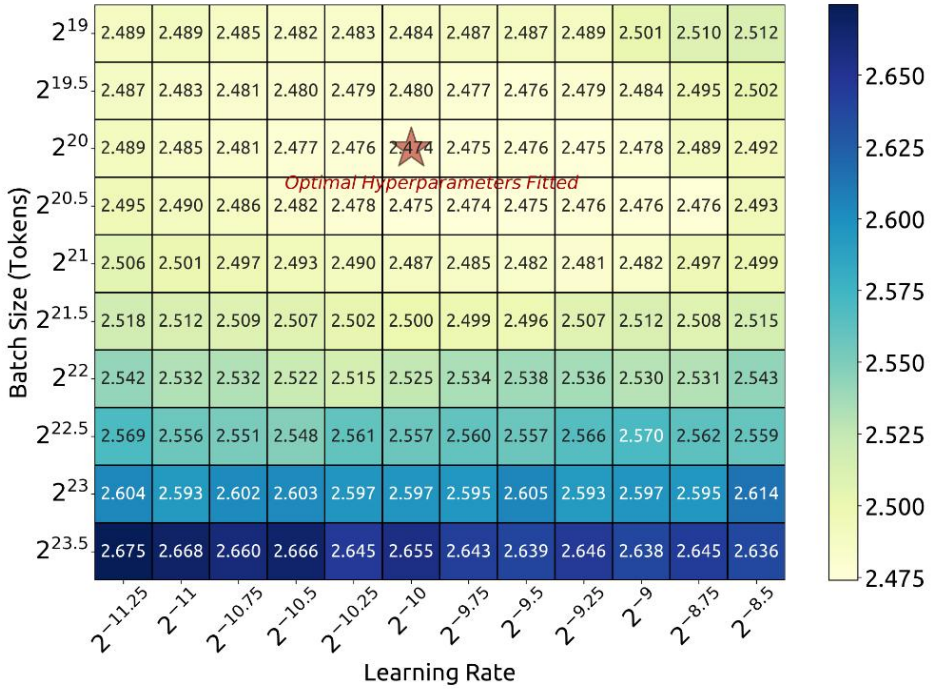
对于Decoder-only的模型，计算量C (FLOPs), 模型参数量N, 数据大小D (token数)，三者满足如上的定量关系。
模型的最终性能主要与计算量，模型参数量和数据大小三者相关，而与模型的具体结构(深度/宽度)基本无关。



Scaling Laws for Hyperparameters



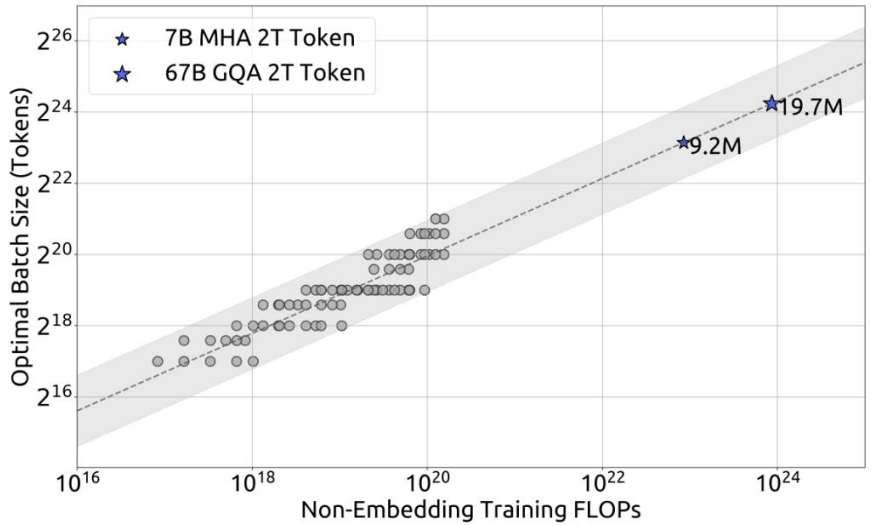
(a) 1e17 FLOPs (177M FLOPs/token)



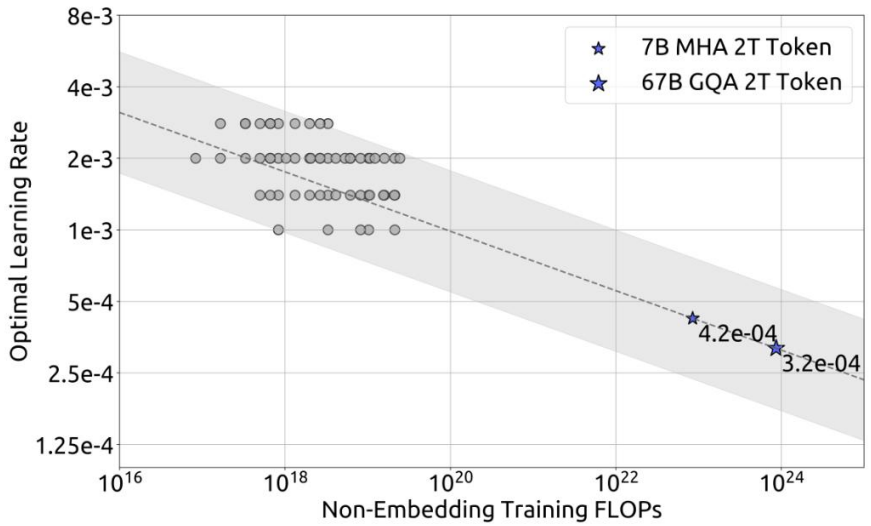
(b) 1e20 FLOPs (2.94B FLOPs/token)

超参数敏感度较低，精细化选择是非必需的。

Scaling Laws for Hyperparameters



(a) Batch size scaling curve



(b) Learning rate scaling curve

$$\eta_{\text{opt}} = 0.3118 \cdot C^{-0.1250}$$

$$B_{\text{opt}} = 0.2920 \cdot C^{0.3271}$$

模型的最优性能（计算效率最佳）和batch size，learning rate之间存在幂律分布。当计算量增加时，batch size越大越好，而learning rate越小越好。

Scaling Laws for Model and Data

$$C \approx 6ND$$

$$6N_1 = 72\, n_{\text{layer}}\, d^2_{\text{model}}$$

$$6N_2 = 72\, n_{\text{layer}}\, d^2_{\text{model}} + 6\, n_{\text{vocab}}\, d_{\text{model}}$$

$$C = MD$$

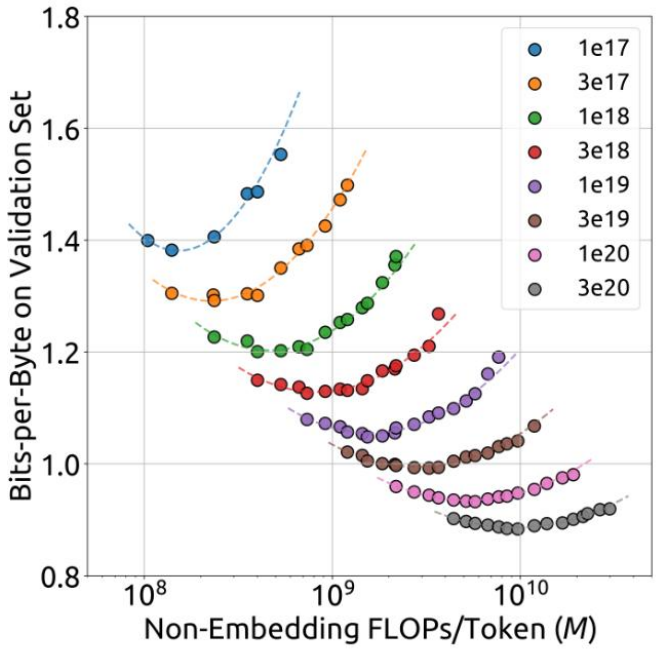
M is the value of non-embedding FLOPs/token

$$M = 72\, n_{\text{layer}}\, d^2_{\text{model}} + 12\, n_{\text{layer}}\, d_{\text{model}}\, l_{\text{seq}}$$

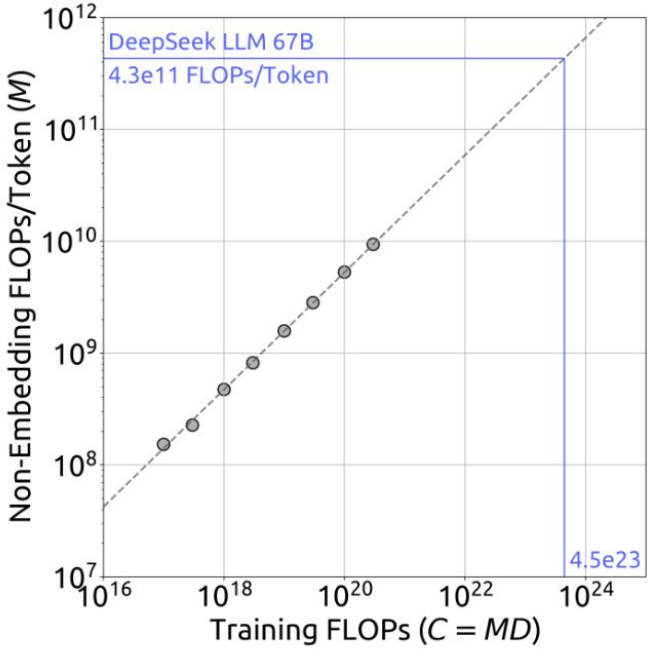
n_{layers}	d_{model}	n_{vocab}	l_{seq}	N_1	N_2	M	$\frac{6N_1}{M}$	$\frac{6N_2}{M}$
8	512			25.2M	77.6M	352M	0.43	1.32
12	768			84.9M	164M	963M	0.53	1.02
24	1024			302M	407M	3.02B	0.60	0.81
24	2048	102400	4096	1.21B	1.42B	9.66B	0.75	0.88
32	4096			6.44B	6.86B	45.1B	0.85	0.91
40	5120			12.6B	13.1B	85.6B	0.88	0.92
80	8192			64.4B	65.3B	419B	0.92	0.94

已有的方式估计不准确。

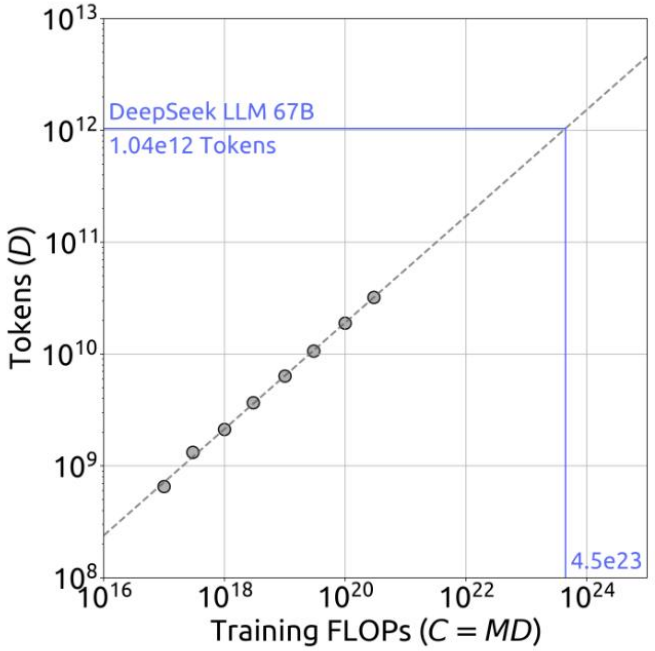
Scaling Laws for Model and Data



(a) IsoFLOP curve



(b) Optimal model scaling



(c) Optimal data scaling

$$M_{\text{opt}} = M_{\text{base}} \cdot C^{a}, \quad M_{\text{base}} = 0.1715, \quad a = 0.5243$$

$$D_{\text{opt}} = D_{\text{base}} \cdot C^{b}, \quad D_{\text{base}} = 5.8316, \quad b = 0.4757$$

Scaling Laws for Different Data

Approach	Coeff. a where $N_{\text{opt}}(M_{\text{opt}}) \propto C^a$	Coeff. b where $D_{\text{opt}} \propto C^b$
OpenAI (OpenWebText2)	0.73	0.27
Chinchilla (MassiveText)	0.49	0.51
Ours (Early Data)	0.450	0.550
Ours (Current Data)	0.524	0.476
Ours (OpenWebText2)	0.578	0.422

数据质量越高，所需要的数据规模越小，在同样的计算代价下，可以训练一个更大尺寸的模型。

DeepSeek-V2: A Strong, Economical, and Efficient Mixture-of-Experts Language Model

Architecture

- Multi-head Latent Attention (MLA)
- DeepSeekMoE

Architecture

Attention

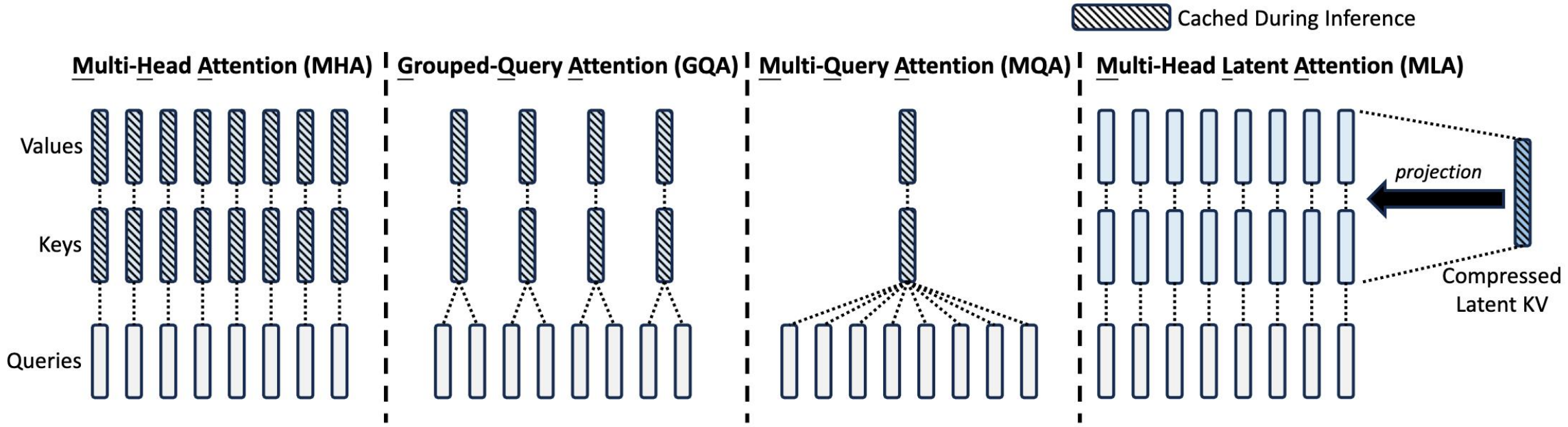
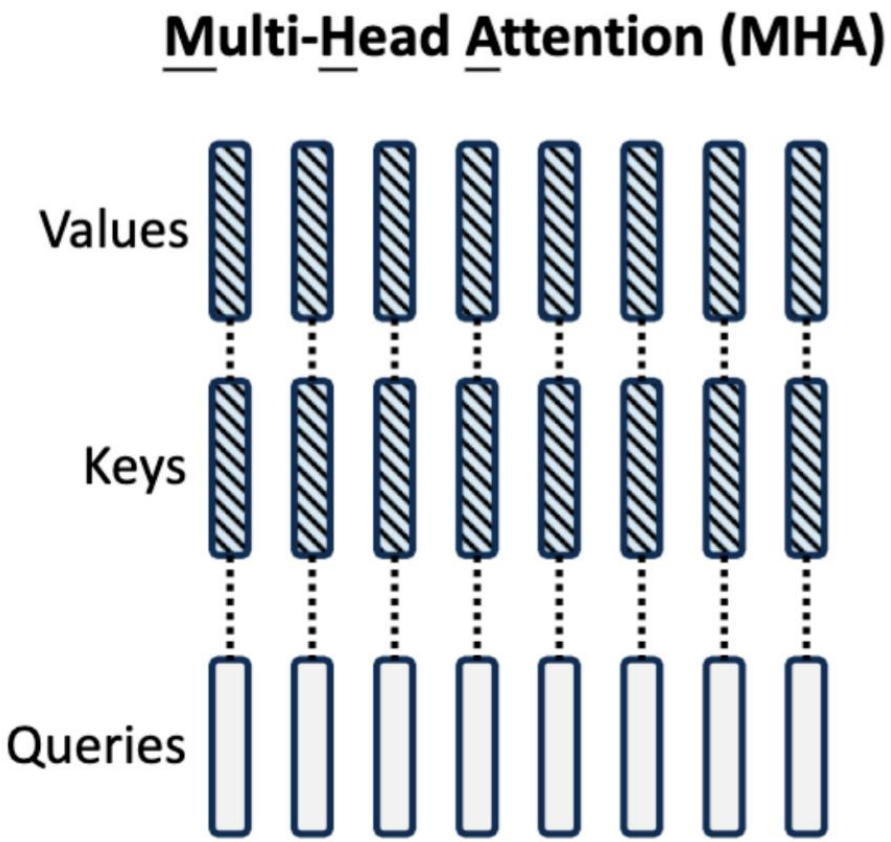


Figure 3 | Simplified illustration of Multi-Head Attention (MHA), Grouped-Query Attention (GQA), Multi-Query Attention (MQA), and Multi-head Latent Attention (MLA). Through jointly compressing the keys and values into a latent vector, MLA significantly reduces the KV cache during inference.

Architecture

Multi-Head Attention



$$\boldsymbol{o}_t = \left[\boldsymbol{o}_t^{(1)}, \boldsymbol{o}_t^{(2)}, \cdots, \boldsymbol{o}_t^{(h)}\right]$$

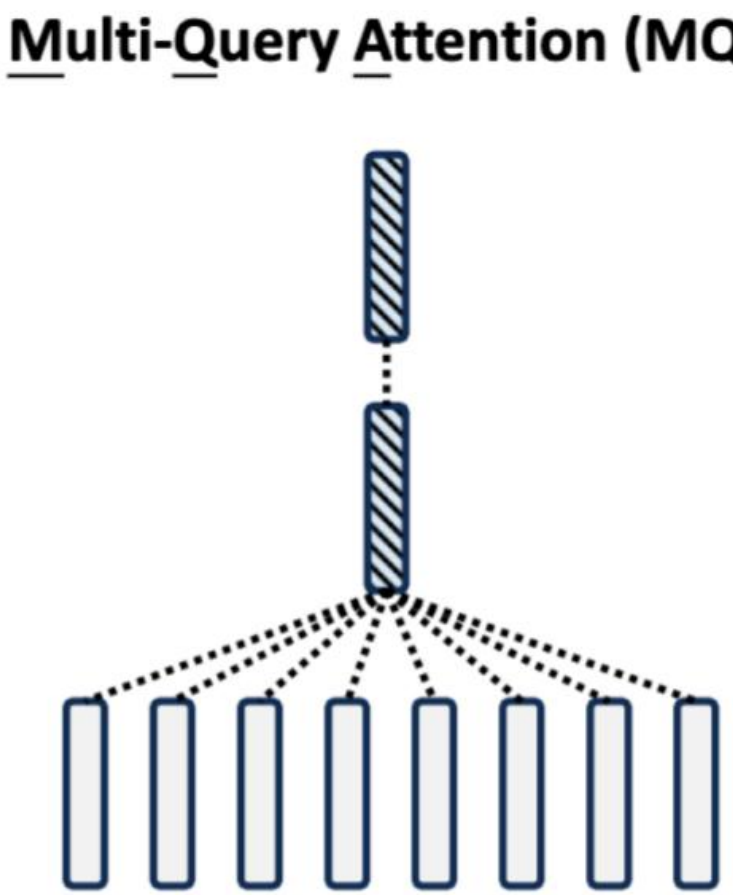
$$\boldsymbol{o}_t^{(s)} = Attention\left(\boldsymbol{q}_t^{(s)}, \boldsymbol{k}_{\leq t}^{(s)}, \boldsymbol{v}_{\leq t}^{(s)}\right) \triangleq \frac{\sum_{i\leq t} \exp\left(\boldsymbol{q}_t^{(s)} \boldsymbol{k}_i^{(s)\top}\right) \boldsymbol{v}_i^{(s)}}{\sum_{i\leq t} \exp\left(\boldsymbol{q}_t^{(s)} \boldsymbol{k}_i^{(s)\top}\right)}$$

$$\begin{aligned}
\boldsymbol{q}_i^{(s)} &= \boldsymbol{x}_i \boldsymbol{W}_q^{(s)} \in \mathbb{R}^{d_k}, \quad \boldsymbol{W}_q^{(s)} \in \mathbb{R}^{d\times d_k} \\
\boldsymbol{k}_i^{(s)} &= \boldsymbol{x}_i \boldsymbol{W}_k^{(s)} \in \mathbb{R}^{d_k}, \quad \boldsymbol{W}_k^{(s)} \in \mathbb{R}^{d\times d_k} \\
\boldsymbol{v}_i^{(s)} &= \boldsymbol{x}_i \boldsymbol{W}_v^{(s)} \in \mathbb{R}^{d_v}, \quad \boldsymbol{W}_v^{(s)} \in \mathbb{R}^{d\times d_v}
\end{aligned}$$

缓存与效果的极限拉扯：从MHA、MQA、GQA到MLA, 苏剑林 https://kexue.fm/archives/10091

Architecture

Multi-Query Attention



$$\boldsymbol{o}_t = \left[\boldsymbol{o}_t^{(1)}, \boldsymbol{o}_t^{(2)}, \cdots, \boldsymbol{o}_t^{(h)}\right]$$

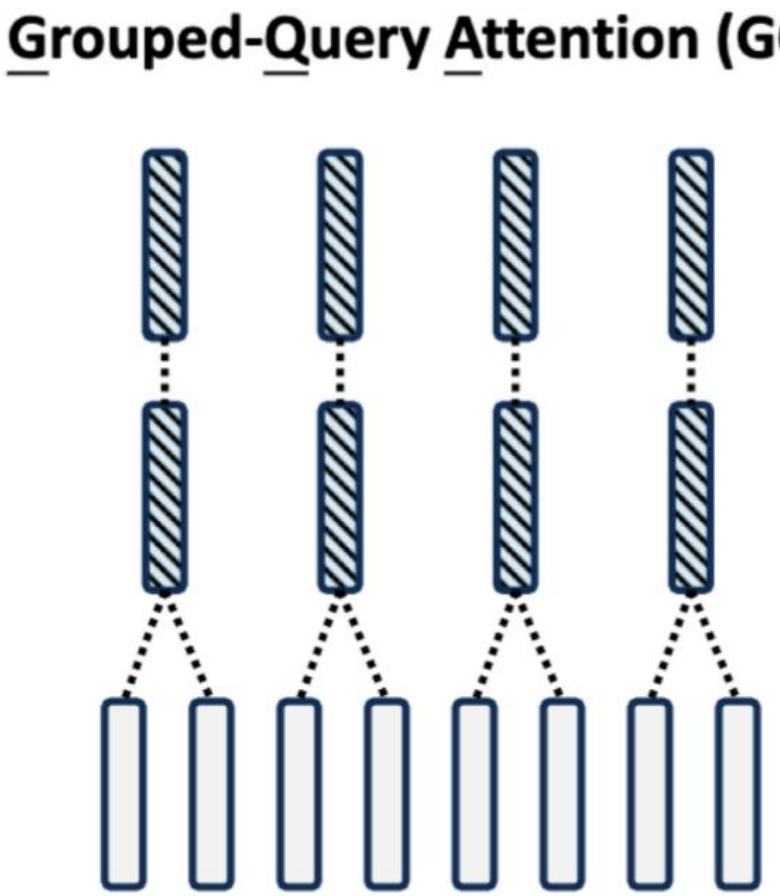
$$\boldsymbol{o}_t^{(s)} = Attention\left(\boldsymbol{q}_t^{(s)}, \boldsymbol{k}_{\leq t}^{(s)}, \boldsymbol{v}_{\leq t}^{(s)}\right) \triangleq \frac{\sum_{i\leq t} \exp\left(\boldsymbol{q}_t^{(s)} \boldsymbol{k}_i^{(s)\top}\right) \boldsymbol{v}_i^{(s)}}{\sum_{i\leq t} \exp\left(\boldsymbol{q}_t^{(s)} \boldsymbol{k}_i^{(s)\top}\right)}$$

$$\begin{aligned}
\boldsymbol{q}_i^{(s)} &= \boldsymbol{x}_i \boldsymbol{W}_q^{(s)} \in \mathbb{R}^{d_k}, \quad \boldsymbol{W}_q^{(s)} \in \mathbb{R}^{d\times d_k} \\
\boldsymbol{k}_i^{(s)} &= \boldsymbol{x}_i \boldsymbol{W}_k^{(s)} \in \mathbb{R}^{d_k}, \quad \boldsymbol{W}_k^{(s)} \in \mathbb{R}^{d\times d_k} \\
\boldsymbol{v}_i^{(s)} &= \boldsymbol{x}_i \boldsymbol{W}_v^{(s)} \in \mathbb{R}^{d_v}, \quad \boldsymbol{W}_v^{(s)} \in \mathbb{R}^{d\times d_v}
\end{aligned}$$

缓存与效果的极限拉扯：从MHA、MQA、GQA到MLA, 苏剑林 https://kexue.fm/archives/10091

Architecture

Grouped-Query Attention



$$\boldsymbol{o}_t = \left[\boldsymbol{o}_t^{(1)}, \boldsymbol{o}_t^{(2)}, \cdots, \boldsymbol{o}_t^{(h)}\right]$$

$$\boldsymbol{o}_t^{(s)} = Attention\left(\boldsymbol{q}_t^{(s)}, \boldsymbol{k}_{\leq t}^{(\lceil sg/h \rceil)}, \boldsymbol{v}_{\leq t}^{(\lceil sg/h \rceil)}\right) \triangleq \frac{\sum_{i\leq t} \exp\left(\boldsymbol{q}_t^{(s)} \boldsymbol{k}_i^{(\lceil sg/h \rceil)\top}\right) \boldsymbol{v}_i^{(\lceil sg/h \rceil)}}{\sum_{i\leq t} \exp\left(\boldsymbol{q}_t^{(s)} \boldsymbol{k}_i^{(\lceil sg/h \rceil)\top}\right)}$$

$$\begin{aligned}
\boldsymbol{q}_i^{(s)} &= \boldsymbol{x}_i \boldsymbol{W}_q^{(s)} \in \mathbb{R}^{d_k}, \quad \boldsymbol{W}_q^{(s)} \in \mathbb{R}^{d\times d_k} \\
\boldsymbol{k}_i^{(\lceil sg/h \rceil)} &= \boldsymbol{x}_i \boldsymbol{W}_k^{(\lceil sg/h \rceil)} \in \mathbb{R}^{d_k}, \quad \boldsymbol{W}_k^{(\lceil sg/h \rceil)} \in \mathbb{R}^{d\times d_k} \\
\boldsymbol{v}_i^{(\lceil sg/h \rceil)} &= \boldsymbol{x}_i \boldsymbol{W}_v^{(\lceil sg/h \rceil)} \in \mathbb{R}^{d_v}, \quad \boldsymbol{W}_v^{(\lceil sg/h \rceil)} \in \mathbb{R}^{d\times d_v}
\end{aligned}$$

缓存与效果的极限拉扯：从MHA、MQA、GQA到MLA, 苏剑林 https://kexue.fm/archives/10091

Architecture

减少KV Cache同时尽可能地保证效果

为什么降低KV Cache的大小如此重要？

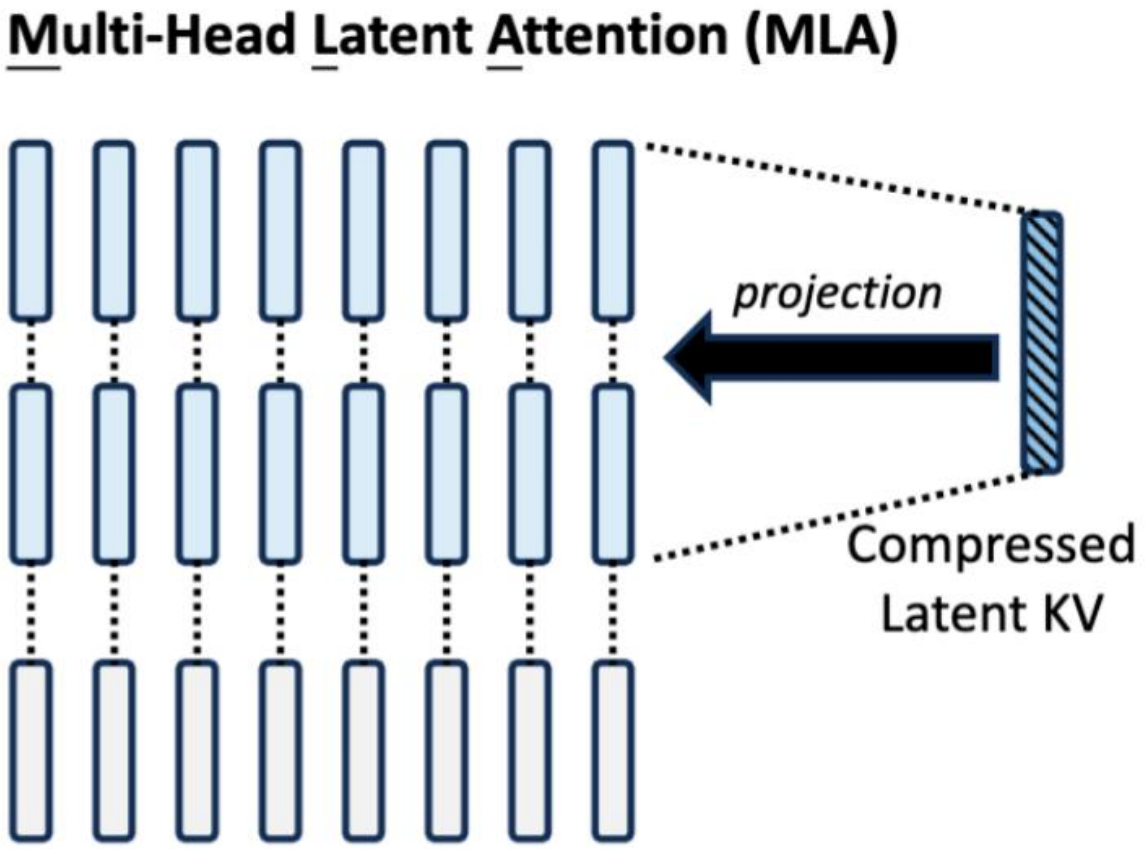
一般情况下LLM的推理都是在GPU上进行，GPU的显存是有限的，一部分要用来存放模型的参数和前向计算的激活值，这部分取决于模型的体量，选定模型后它就是个常数。

另外一部分我们要用来存放模型的KV Cache，这部分不仅依赖于模型的体量，还依赖于模型的输入长度，也就是在推理过程中是动态增长的，当Context长度足够长时，它的大小就会占主导地位，可能超出一张卡甚至一台机器（8卡）的总显存量。

缓存与效果的极限拉扯：从MHA、MQA、GQA到MLA, 苏剑林 https://kexue.fm/archives/10091

Architecture

Multi-Head Latent Attention

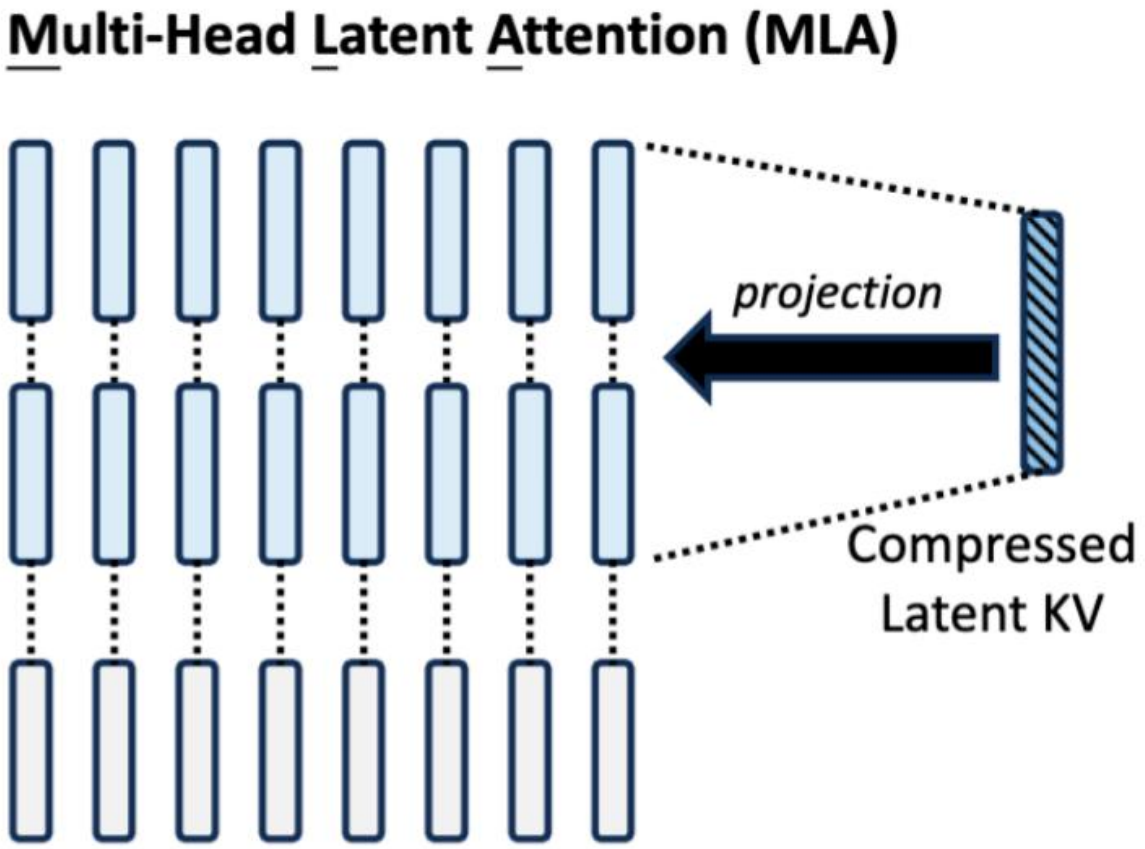


$$
\begin{aligned}
\mathbf{c}_t^Q &= W^{DQ}\mathbf{h}_t, \\
[\mathbf{q}_{t,1}^C; \mathbf{q}_{t,2}^C; \ldots; \mathbf{q}_{t,n_h}^C] = \mathbf{q}_t^C &= W^{UQ}\mathbf{c}_t^Q, \\
[\mathbf{q}_{t,1}^R; \mathbf{q}_{t,2}^R; \ldots; \mathbf{q}_{t,n_h}^R] = \mathbf{q}_t^R &= \mathrm{RoPE}(W^{QR}\mathbf{c}_t^Q), \\
\mathbf{q}_{t,i} &= [\mathbf{q}_{t,i}^C; \mathbf{q}_{t,i}^R], \\
\boxed{\mathbf{c}_t^{KV}} &= W^{DKV}\mathbf{h}_t, \\
[\mathbf{k}_{t,1}^C; \mathbf{k}_{t,2}^C; \ldots; \mathbf{k}_{t,n_h}^C] = \mathbf{k}_t^C &= W^{UK}\mathbf{c}_t^{KV}, \\
\boxed{\mathbf{k}_t^R} &= \mathrm{RoPE}(W^{KR}\mathbf{h}_t), \\
\mathbf{k}_{t,i} &= [\mathbf{k}_{t,i}^C; \mathbf{k}_t^R], \\
[\mathbf{v}_{t,1}^C; \mathbf{v}_{t,2}^C; \ldots; \mathbf{v}_{t,n_h}^C] = \mathbf{v}_t^C &= W^{UV}\mathbf{c}_t^{KV}, \\
\mathbf{o}_{t,i} &= \sum_{j=1}^{t} \mathrm{Softmax}_j\left(\frac{\mathbf{q}_{t,i}^T \mathbf{k}_{j,i}}{\sqrt{d_h + d_h^R}}\right)\mathbf{v}_{j,i}^C, \\
\mathbf{u}_t &= W^O[\mathbf{o}_{t,1}; \mathbf{o}_{t,2}; \ldots; \mathbf{o}_{t,n_h}],
\end{aligned}
$$

缓存与效果的极限拉扯：从MHA、MQA、GQA到MLA, 苏剑林 https://kexue.fm/archives/10091

Architecture

Multi-Head Latent Attention



$$\mathbf{c}_t^{KV} = W^{DKV}\mathbf{h}_t,$$
$$\mathbf{k}_t^C = W^{UK}\mathbf{c}_t^{KV},$$
$$\mathbf{v}_t^C = W^{UV}\mathbf{c}_t^{KV},$$
$$\mathbf{c}_t^Q = W^{DQ}\mathbf{h}_t,$$
$$\mathbf{q}_t^C = W^{UQ}\mathbf{c}_t^Q,$$

缓存与效果的极限拉扯：从MHA、MQA、GQA到MLA, 苏剑林 https://kexue.fm/archives/10091

Architecture

Multi-Head Latent Attention

$$
\begin{aligned}
q_i k_j^\top &= \left(h_i W^Q R_i\right)\left(h_j W^K R_j\right)^\top \\
&= h_i W^Q R_i R_j^\top W^{K\top} h_j^\top \\
&= h_i W^Q R_{j-i} W^{K\top} h_j^\top
\end{aligned}
$$

$$
\begin{aligned}
q_i k_j^\top &= \left(c_i^Q W^{UQ} R_i\right)\left(c_j^{KV} W^{UK} R_j\right)^\top \\
&= c_i^Q W^{UQ} R_i R_j^\top W^{UK\top} c_j^{KV\top} \\
&= c_i^Q W^{UQ} R_{j-i} W^{UK\top} c_j^{KV\top}
\end{aligned}
$$

与RoPE不兼容，从而导致二者相加并不能够节省KV Cache

缓存与效果的极限拉扯：从MHA、MQA、GQA到MLA, 苏剑林 https://kexue.fm/archives/10091

Architecture

Multi-Head Latent Attention

换位置编码

用别的位置编码替换已有的RoPE编码，效果会有损失。

拆位置编码

将表示拆分成两部分，一部分包含位置编码信息，一部分不包含。不包含的部分可以大幅度减少KV Cache。

缓存与效果的极限拉扯：从MHA、MQA、GQA到MLA, 苏剑林 https://kexue.fm/archives/10091

Architecture

Multi-Head Latent Attention

不包含位置编码

$$
\begin{aligned}
q_i k_j^\top &= \left(c_i^Q W^{UQ} R_i\right)\left(c_j^{KV} W^{UK} R_j\right)^\top \\
&= c_i^Q W^{UQ} R_i R_j^\top W^{UK\top} c_j^{KV\top} \\
&= c_i^Q W^{UQ} R_{j-i} W^{UK\top} c_j^{KV\top}
\end{aligned}
\qquad
\begin{aligned}
q_i k_j^\top &= \left(c_i^Q W^{UQ}\right)\left(c_j^{KV} W^{UK}\right)^\top \\
&= c_i^Q W^{UQ} W^{UK\top} c_j^{KV\top} \\
&= c_i^Q W^{UQ} W^{UK\top} c_j^{KV\top}
\end{aligned}
$$

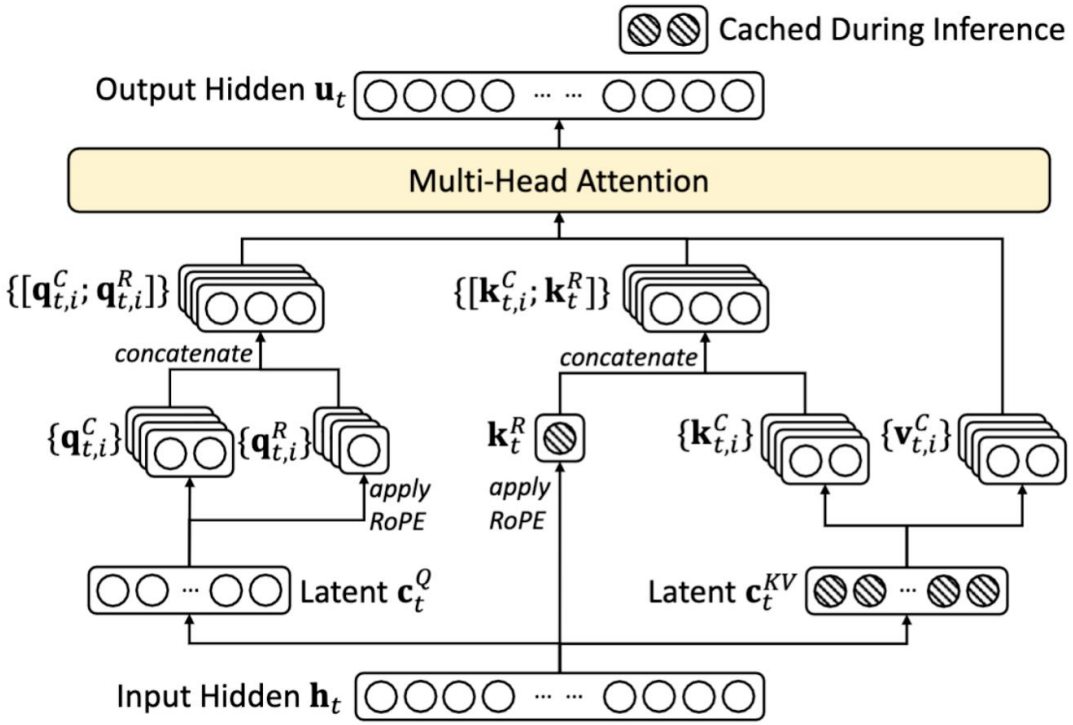
$$
\begin{aligned}
O &= (Matrix_{att})\left(c^{KV} W^{UV}\right) W^O \\
&= (Matrix_{att})\, c^{KV}\left(W^{UV} W^O\right)
\end{aligned}
$$

缓存与效果的极限拉扯：从MHA、MQA、GQA到MLA, 苏剑林 https://kexue.fm/archives/10091

Architecture

Multi-Head Latent Attention

包含位置编码



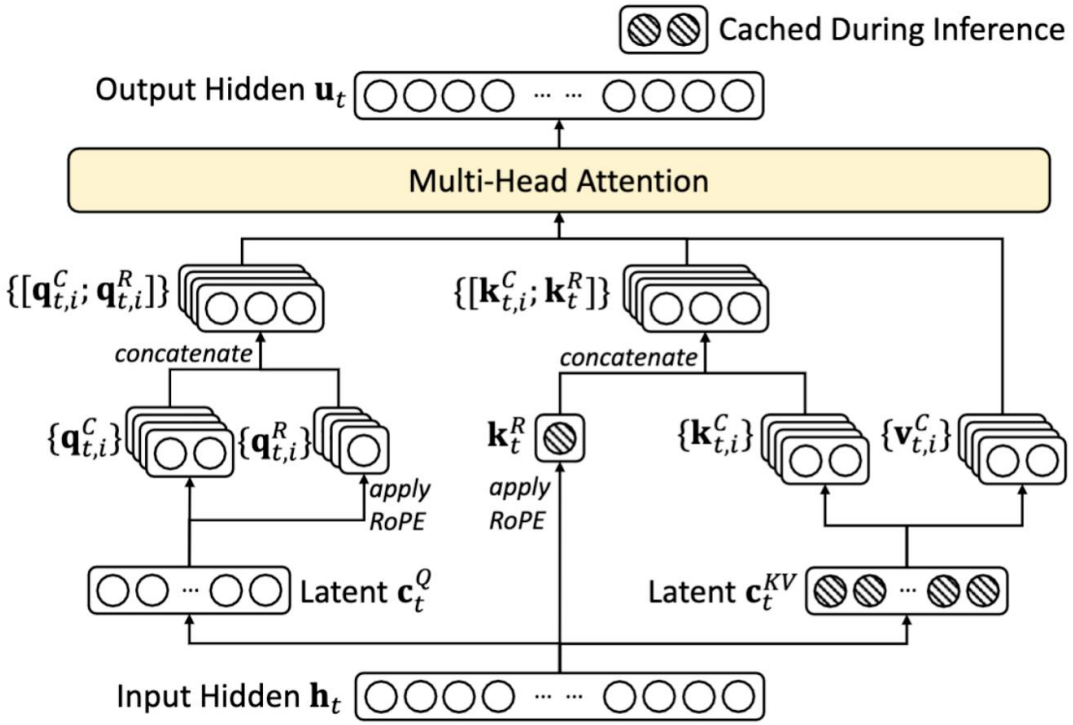
$$
\begin{aligned}
[\mathbf{q}^R_{t,1}; \mathbf{q}^R_{t,2}; \ldots; \mathbf{q}^R_{t,n_h}] = \mathbf{q}^R_t &= \mathrm{RoPE}(W^{QR}\mathbf{c}^Q_t), \\
\mathbf{k}^R_t &= \mathrm{RoPE}(W^{KR}\mathbf{h}_t), \\
\mathbf{q}_{t,i} &= [\mathbf{q}^C_{t,i}; \mathbf{q}^R_{t,i}], \\
\mathbf{k}_{t,i} &= [\mathbf{k}^C_{t,i}; \mathbf{k}^R_t], \\
\mathbf{o}_{t,i} &= \sum_{j=1}^{t} \mathrm{Softmax}_j\left(\frac{\mathbf{q}^T_{t,i}\mathbf{k}_{j,i}}{\sqrt{d_h + d^R_h}}\right)\mathbf{v}^C_{j,i}, \\
\mathbf{u}_t &= W^O[\mathbf{o}_{t,1}; \mathbf{o}_{t,2}; \ldots; \mathbf{o}_{t,n_h}],
\end{aligned}
$$

缓存与效果的极限拉扯：从MHA、MQA、GQA到MLA, 苏剑林 https://kexue.fm/archives/10091

Architecture

Multi-Head Latent Attention

包含位置编码



$$
\begin{aligned}
[\mathbf{q}^R_{t,1}; \mathbf{q}^R_{t,2}; \ldots; \mathbf{q}^R_{t,n_h}] = \mathbf{q}^R_t &= \mathrm{RoPE}(W^{QR}\mathbf{c}^Q_t), \\
\mathbf{k}^R_t &= \mathrm{RoPE}(W^{KR}\mathbf{h}_t), \\
\mathbf{q}_{t,i} &= [\mathbf{q}^C_{t,i}; \mathbf{q}^R_{t,i}], \\
\mathbf{k}_{t,i} &= [\mathbf{k}^C_{t,i}; \mathbf{k}^R_t], \\
\mathbf{o}_{t,i} &= \sum_{j=1}^{t} \mathrm{Softmax}_j\left(\frac{\mathbf{q}^T_{t,i}\mathbf{k}_{j,i}}{\sqrt{d_h + d^R_h}}\right)\mathbf{v}^C_{j,i}, \\
\mathbf{u}_t &= W^O[\mathbf{o}_{t,1}; \mathbf{o}_{t,2}; \ldots; \mathbf{o}_{t,n_h}],
\end{aligned}
$$

缓存与效果的极限拉扯：从MHA、MQA、GQA到MLA, 苏剑林 https://kexue.fm/archives/10091

Architecture

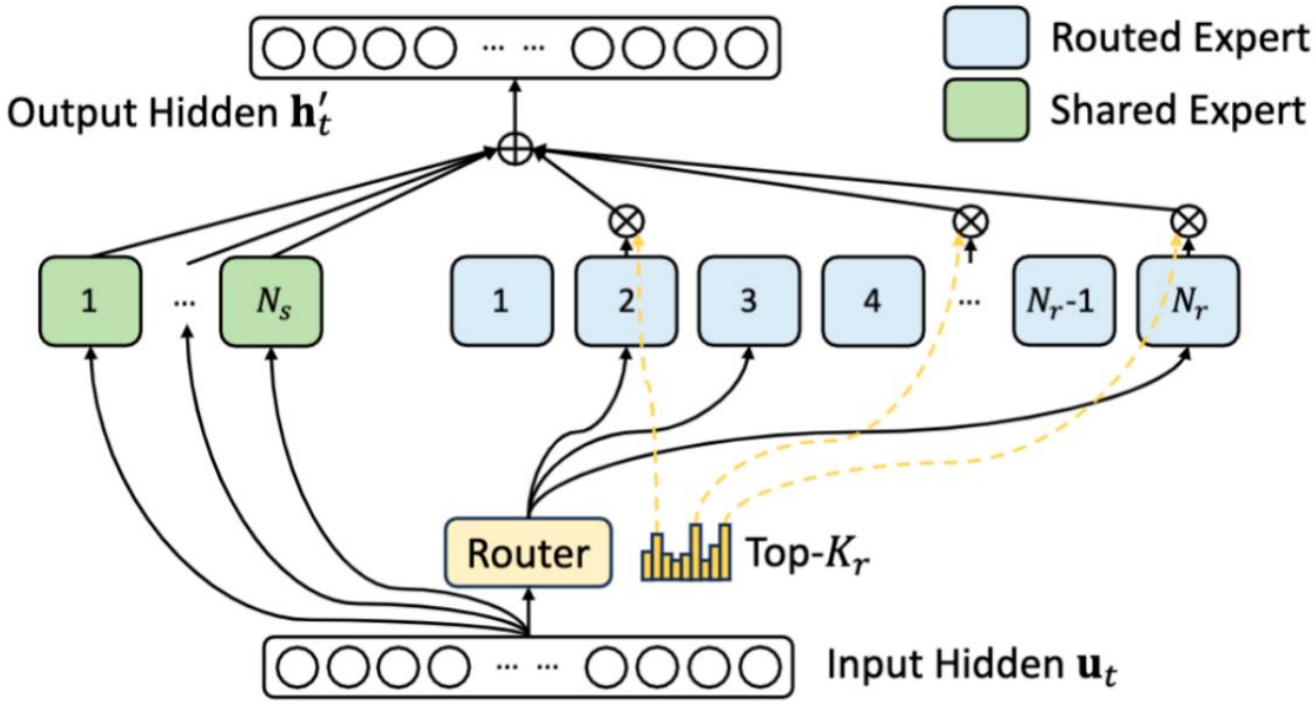
Multi-Head Latent Attention

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

Table 1 | Comparison of the KV cache per token among different attention mechanisms. n_h denotes the number of attention heads, d_h denotes the dimension per attention head, l denotes the number of layers, n_g denotes the number of groups in GQA, and d_c and d_h^R denote the KV compression dimension and the per-head dimension of the decoupled queries and key in MLA, respectively. The amount of KV cache is measured by the number of elements, regardless of the storage precision. For DeepSeek-V2, d_c is set to $4d_h$ and d_h^R is set to $\frac{d_h}{2}$. So, its KV cache is equal to GQA with only 2.25 groups, but its performance is stronger than MHA.

Architecture

DeepSeekMoE



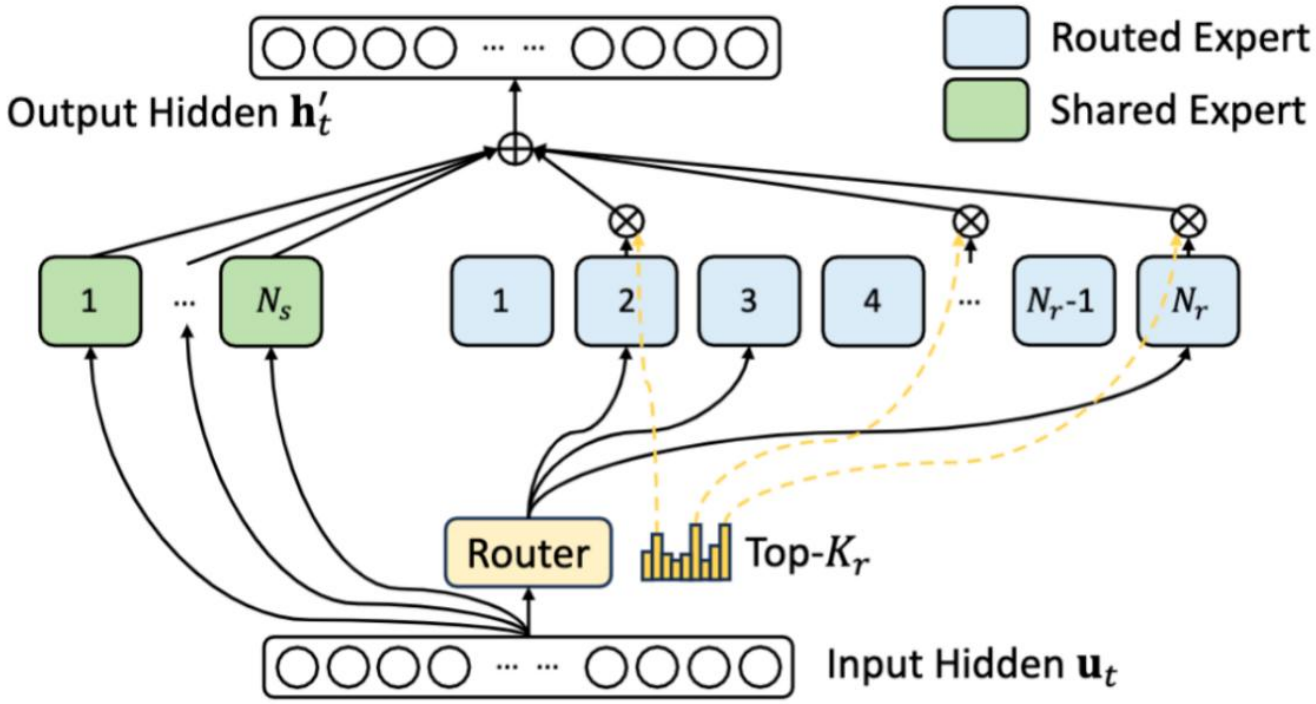
$$\mathbf{h}'_t = \mathbf{u}_t + \sum_{i=1}^{N_s} \mathrm{FFN}_i^{(s)}(\mathbf{u}_t) + \sum_{i=1}^{N_r} g_{i,t}\,\mathrm{FFN}_i^{(r)}(\mathbf{u}_t),$$

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

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

Architecture

DeepSeekMoE



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

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

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

Architecture

Auxiliary Loss for Load Balance

Expert-Level

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

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

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

Device-Level

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

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

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

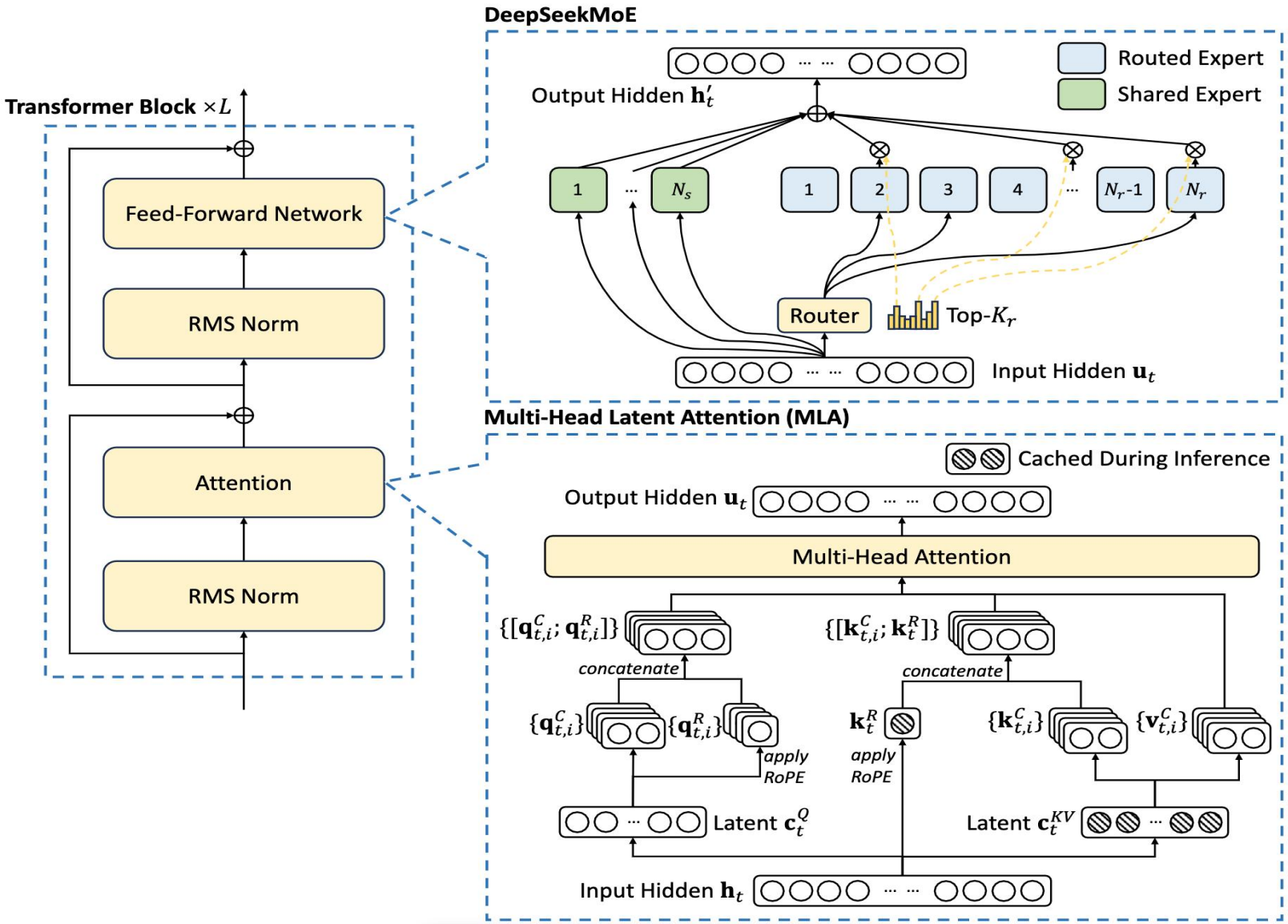
Communication

$$\mathcal{L}_{\text{CommBal}} = \alpha_3 \sum_{i=1}^{D} f''_i P''_i,$$

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

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

Architecture



Experiments

Attention Mechanisms

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	**37.0**
MMLU (Acc.)	5-shot	37.9	41.2	**45.2**
C-Eval (Acc.)	5-shot	30.0	37.7	**42.9**
CMMLU (Acc.)	5-shot	34.6	38.4	**43.5**

Experiments

Attention Mechanisms

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	**39.0**	46.6	**50.7**
MMLU (Acc.)	5-shot	48.7	**50.0**	57.5	**59.0**
C-Eval (Acc.)	5-shot	**51.6**	50.9	57.9	**59.2**
CMMLU (Acc.)	5-shot	52.3	**53.4**	60.7	**62.5**