长上下文大模型

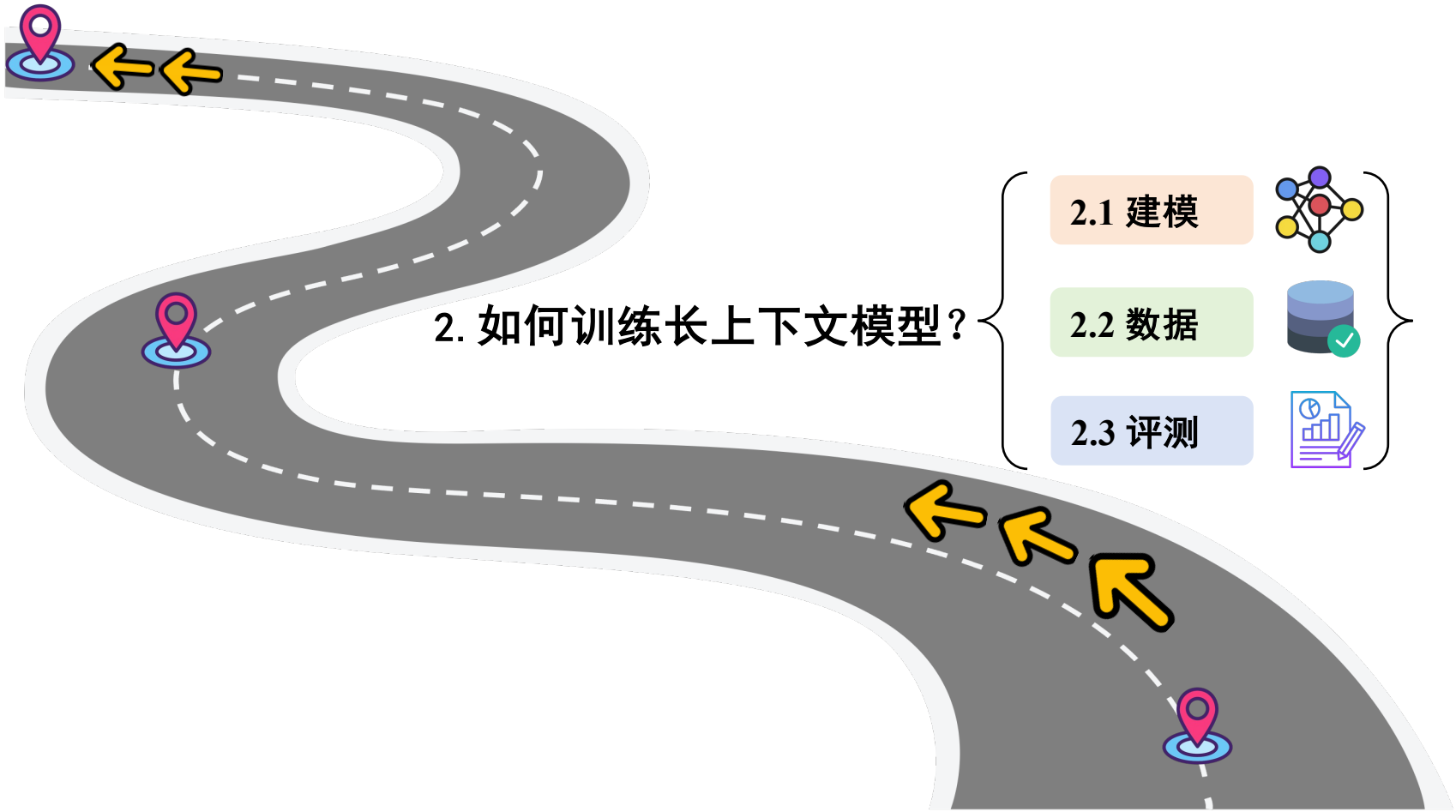
—进展与挑战

李俊涛

ARTIFICIAL INTELLIGENCE SOOCHOW UNIVERSITY
SUDA AI
OpenNLG

报告内容

3. 长上下文大模型前沿与挑战



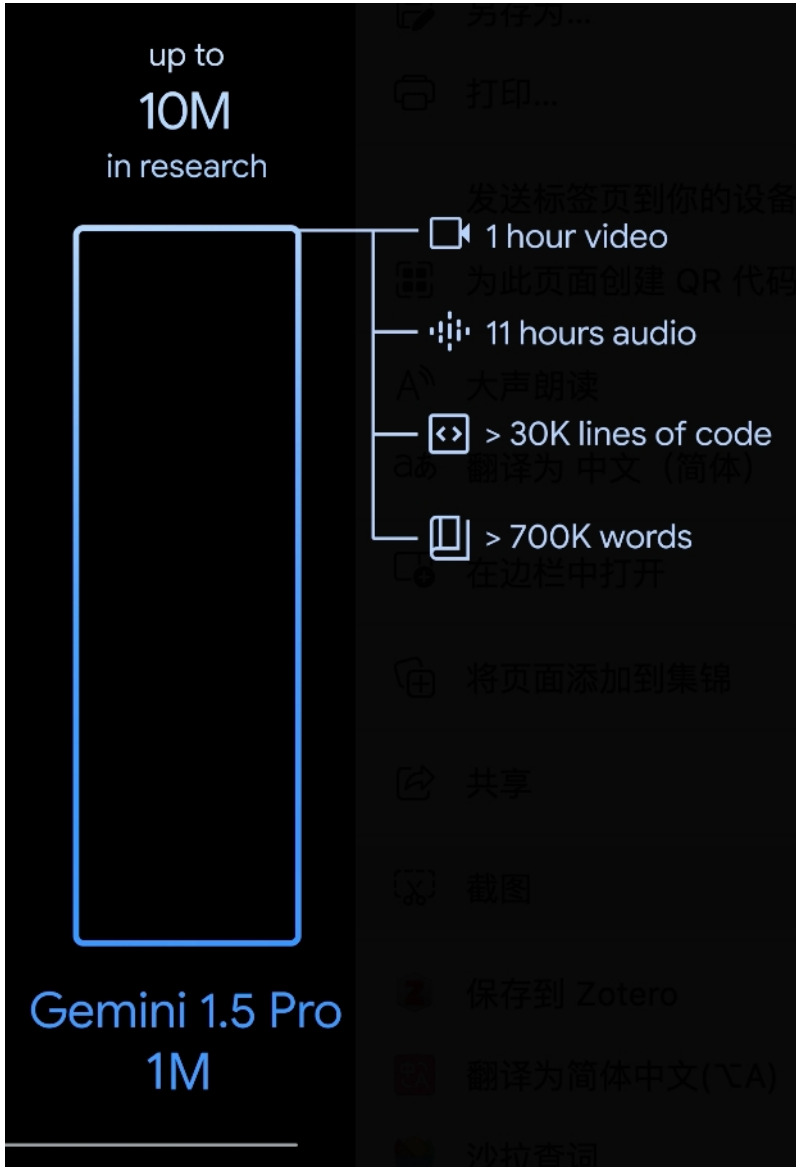
2. 如何训练长上下文模型？

- 2.1 建模
- 2.2 数据
- 2.3 评测

1. 什么是长上下文模型？

大模型上下文输入长度

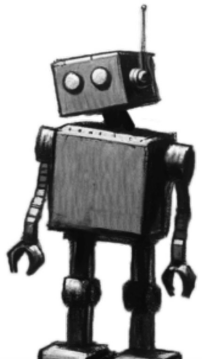
	闭源模型	开源模型
2022 Nov	ChatGPT-3.5-16K	
2023 Apr		ChatGLM2-32K
2023 July	Claude-2-100K	MPT-65K
2023 Sep		Qwen1.5-32K Baichuan2-192K
2023 Nov	GPT4 Turbo-128K	Yi-6B-200K
2023 Dec	Gemini1.0 Pro-32K	Mistral-7B-0.2-32K
2024 Mar	Claude3-200K	Deepseek-V2-128K
2024 May	Gemini1.5 Pro-10M	
2024 July	GLM-4-long-1M	Mistral Nemo-128K Meta-Llama 3.1-128K



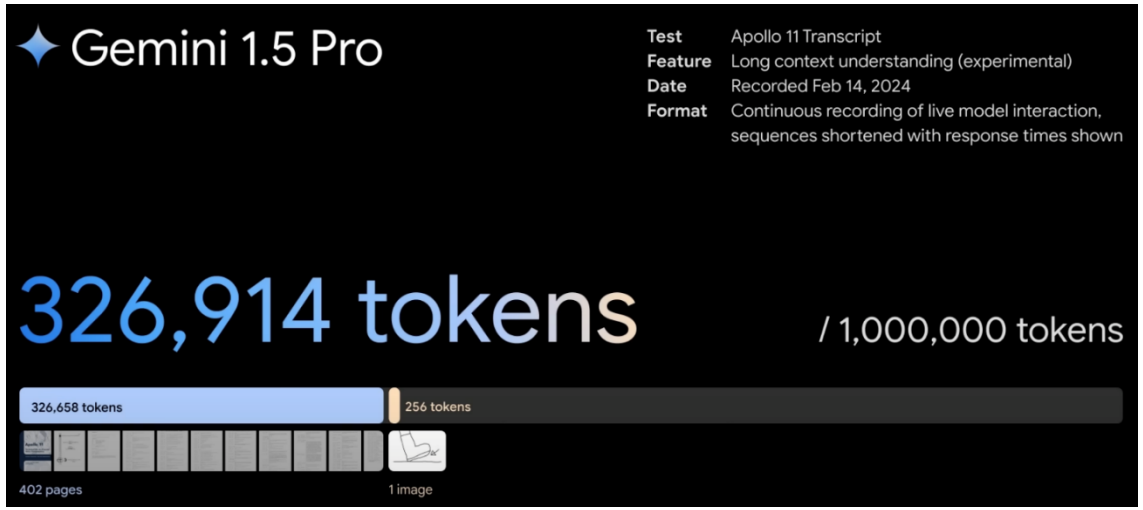
Source: Google Blog

长上下文大模型使用场景

- 在众多场景越来越重要（复杂场景、部署便捷性、高效计算）
  - 长文档处理（RAG）
  - 代码助手
  - 工具调用
  - 长历史对话
  - 多模态输入处理



“See” More
“Memory” More
“Think” More
“Speak” More
…



什么是长上下文模型？

相对模糊的概念

“A long context model, in the realm of natural language processing, refers to a type of language model that is capable of processing and understanding extensive sequences of text, **far beyond** the **typical context window size** that standard large language models (LLMs) can handle.”

Nikolay Savinov

Research Scientist

Google DeepMind

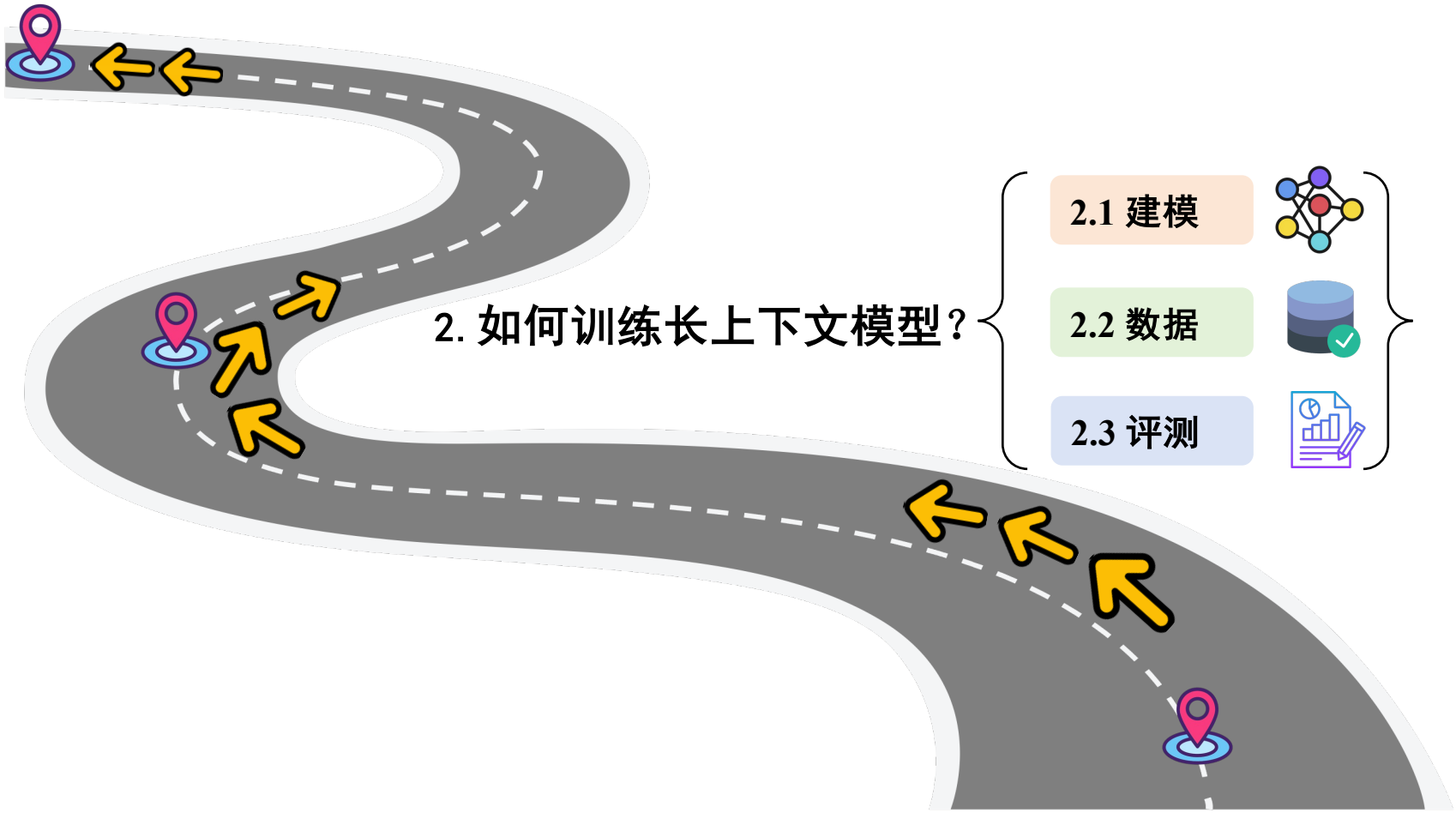
Google Blog, Gemini Team

“10 million tokens at once is already close to the thermal limit of our Tensor Processing Units — we don't know where the limit is yet, and the model might be capable of even more as the hardware continues to improve”

Advancing transformer architecture in long-context large language models: A comprehensive survey[J]. arXiv preprint arXiv:2311.12351, 2023.

报告内容

3. 长上下文大模型前沿与挑战

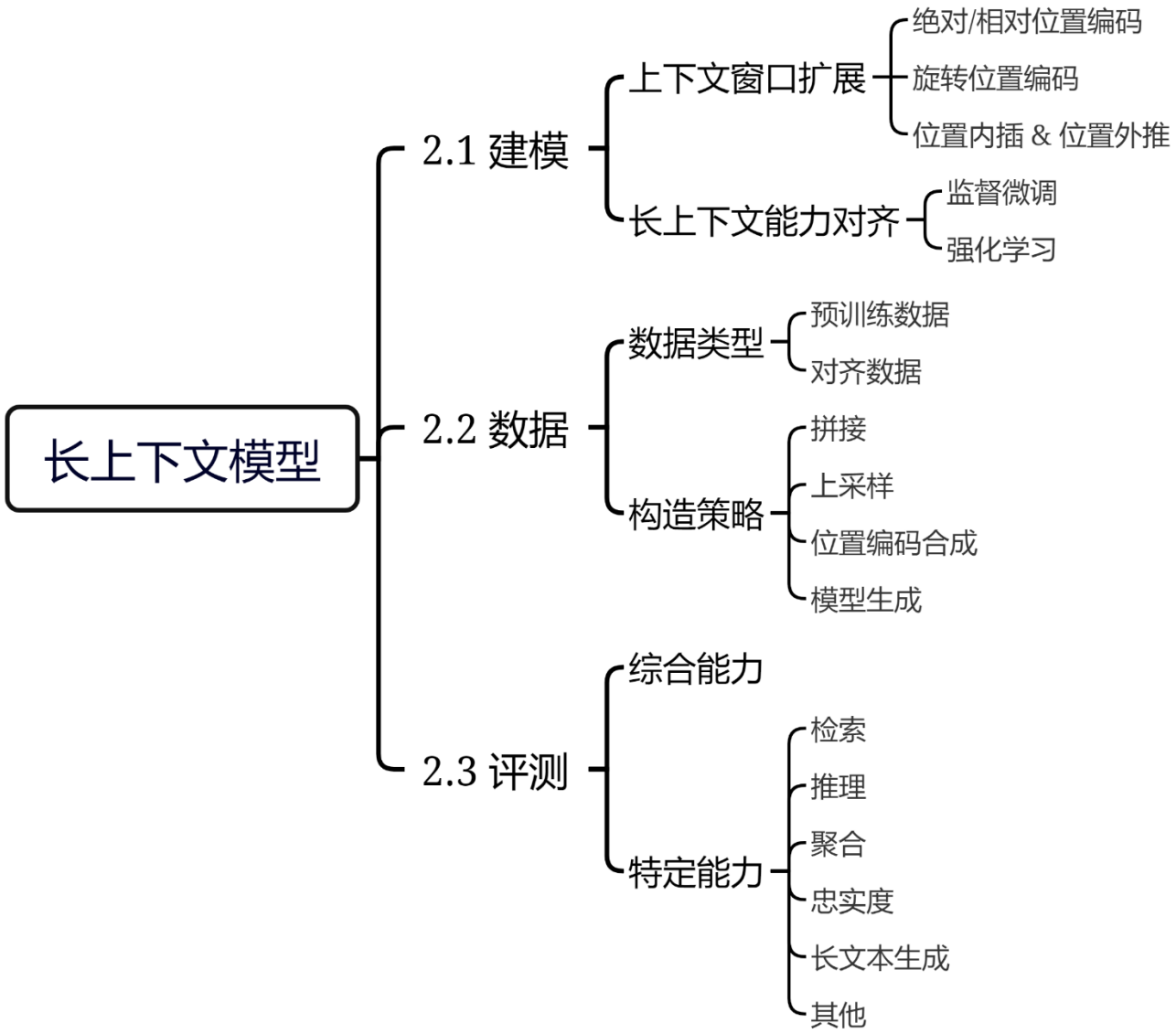


2. 如何训练长上下文模型？

- 2.1 建模
- 2.2 数据
- 2.3 评测

1. 什么是长上下文模型？

汇报目录


长上下文模型
2.1 建模
上下文窗口扩展
绝对/相对位置编码
旋转位置编码
位置内插 & 位置外推
长上下文能力对齐
监督微调
强化学习
2.2 数据
数据类型
预训练数据
对齐数据
构造策略
拼接
上采样
位置编码合成
模型生成
2.3 评测
综合能力
特定能力
检索
推理
聚合
忠实度
长文本生成
其他

建模

开源短上下文强模型
(Llama2-4K, Llama3-8K)

具有长上下文窗口的模型
(>32K)

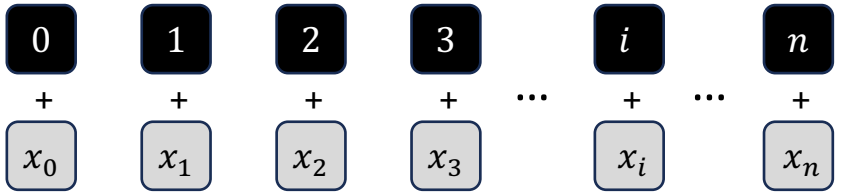
强长上下文模型

- 上下文窗口扩展
  - 相对位置编码(RPE)
  - 旋转位置编码 (RoPE)
  - 位置内插 (PI) 与外推(PE)
- 长上下文能力对齐
  - 监督微调 (SFT)
  - 强化学习 (RL)

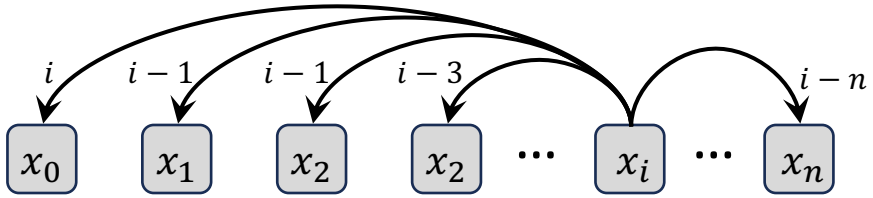
上下文窗口扩展 • 位置编码

➢ 基于Transformer 的模型依赖位置编码来确定每个token的位置，相对位置编码额外关注相对位置关系

绝对位置编码(APE)：



相对位置编码(RPE)：



0	1	2	3	4	5	6	7	8	9	10
-1	0	1	2	3	4	5	6	7	8	9
-2	-1	0	1	2	3	4	5	6	7	8
-3	-2	-1	0	1	2	3	4	5	6	7
-4	-3	-2	-1	0	1	2	3	4	5	6
-5	-4	-3	-2	-1	0	1	2	3	4	5
-6	-5	-4	-3	-2	-1	0	1	2	3	4
-7	-6	-5	-4	-3	-2	-1	0	1	2	3
-8	-7	-6	-5	-4	-3	-2	-1	0	1	2
-9	-8	-7	-6	-5	-4	-3	-2	-1	0	1
-10	-9	-8	-7	-6	-5	-4	-3	-2	-1	0

上下文窗口扩展 • 相对位置编码

◘ 线性偏差注意力使输入长度外推成为可能(ALiBi)（ICLR 2022）

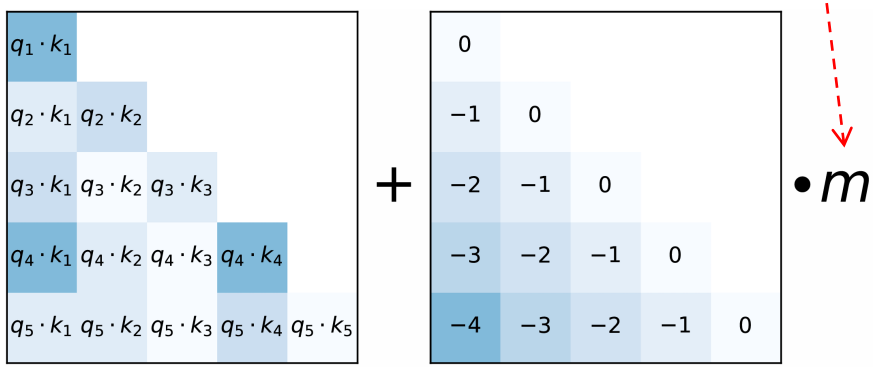
➢ 优点

- 简单而有效，MPT（2023）模型上下文窗口达到65K

➢ 缺点

- 单向：无法识别左右
- 相对位置权值随序列长度增加而严重衰减

ALiBi函数：$A_{i,j} = E_{x_i}^T W_q^T W_k E_{x_j} - m|i-j|$



0	1	2	3	4	5	6	7	8	9	10	11
1	0	1	2	3	4	5	6	7	8	9	10
2	1	0	1	2	3	4	5	6	7	8	9
3	2	1	0	1	2	3	4	5	6	7	8
4	3	2	1	0	1	2	3	4	5	6	7
5	4	3	2	1	0	1	2	3	4	5	6
6	5	4	3	2	1	0	1	2	3	4	5
7	6	5	4	3	2	1	0	1	2	3	4
8	7	6	5	4	3	2	1	0	1	2	3
9	8	7	6	5	4	3	2	1	0	1	2
10	9	8	7	6	5	4	3	2	1	0	1
11	10	9	8	7	6	5	4	3	2	1	0

上下文窗口扩展 • 相对位置编码

- Bucket相对位置编码首次提出于T5（JMLR, 2020）
  - 核心函数：$A_{i,j} = E_{x_i}^T W_q^T W_k E_{x_j} + \beta_{i,j}$
    - 可以区分左右
    - 相对距离分桶

$$b_{i,j} = \begin{cases} i-j, & if\ |i-j| < n_b/4 \\ \dfrac{i-j}{|i-j|} \cdot \min\left(\dfrac{n_b}{2} - 1, \dfrac{n_b}{4} + \left\lfloor \dfrac{\log\left(\frac{(i-j)}{n_b/4}\right)}{\log\left(\frac{\max d}{n_b/4}\right)} \cdot \dfrac{n_b}{4} \right\rfloor \right), & else \end{cases}$$

核心超参：桶的数量

$i-j$	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
$f(i-j)$	0	1	2	3	4	5	6	7	8	8	8	8	9	9	9	9
$i-j$	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	···
$f(i-j)$	10	10	10	10	10	10	10	11	11	11	11	11	11	11	11	···

桶0 桶1 桶3 桶4

0	1	2	3	4	5	6	7	8	8	8	8	9	9	9	9	10	10	10	10
-1	0	1	2	3	4	5	6	7	8	8	8	8	9	9	9	9	10	10	10
-2	-1	0	1	2	3	4	5	6	7	8	8	8	8	9	9	9	9	10	10
-3	-2	-1	0	1	2	3	4	5	6	7	8	8	8	8	9	9	9	9	10
-4	-3	-2	-1	0	1	2	3	4	5	6	7	8	8	8	8	9	9	9	9
-5	-4	-3	-2	-1	0	1	2	3	4	5	6	7	8	8	8	8	9	9	9
-6	-5	-4	-3	-2	-1	0	1	2	3	4	5	6	7	8	8	8	8	9	9
-7	-6	-5	-4	-3	-2	-1	0	1	2	3	4	5	6	7	8	8	8	8	9
-8	-7	-6	-5	-4	-3	-2	-1	0	1	2	3	4	5	6	7	8	8	8	8
-8	-8	-7	-6	-5	-4	-3	-2	-1	0	1	2	3	4	5	6	7	8	8	8
-8	-8	-8	-7	-6	-5	-4	-3	-2	-1	0	1	2	3	4	5	6	7	8	8
-8	-8	-8	-8	-7	-6	-5	-4	-3	-2	-1	0	1	2	3	4	5	6	7	8
-9	-8	-8	-8	-8	-7	-6	-5	-4	-3	-2	-1	0	1	2	3	4	5	6	7
-9	-9	-8	-8	-8	-8	-7	-6	-5	-4	-3	-2	-1	0	1	2	3	4	5	6
-9	-9	-9	-8	-8	-8	-8	-7	-6	-5	-4	-3	-2	-1	0	1	2	3	4	5
-9	-9	-9	-9	-8	-8	-8	-8	-7	-6	-5	-4	-3	-2	-1	0	1	2	3	4
-10	-9	-9	-9	-9	-8	-8	-8	-8	-7	-6	-5	-4	-3	-2	-1	0	1	2	3
-10	-10	-9	-9	-9	-9	-8	-8	-8	-8	-7	-6	-5	-4	-3	-2	-1	0	1	2
-10	-10	-10	-9	-9	-9	-9	-8	-8	-8	-8	-7	-6	-5	-4	-3	-2	-1	0	1
-10	-10	-10	-10	-9	-9	-9	-9	-8	-8	-8	-8	-7	-6	-5	-4	-3	-2	-1	0

“越近越精确，越远越模糊”

上下文窗口扩展 • 旋转位置编码

◘ 旋转位置编码：绝对位置+相对位置

$$< f_q(x_m, m), f_k(x_n, n) > = g(x_m, x_n, \underbrace{m-n}_{\text{相对位置}})$$

- 根据两个token的绝对位置m, n推导出包含两个token相对位置（m - n）的向量表示
- 几何角度上可以理解成两个拥有不同旋转角度θ的虚数相乘，最终变成旋转角度为θ'的新虚数

用复数形式表示f_q 和 f_k

$$q_m = f_q(x_m, m) = (W_q x_m) e^{im\theta} \quad (1)$$

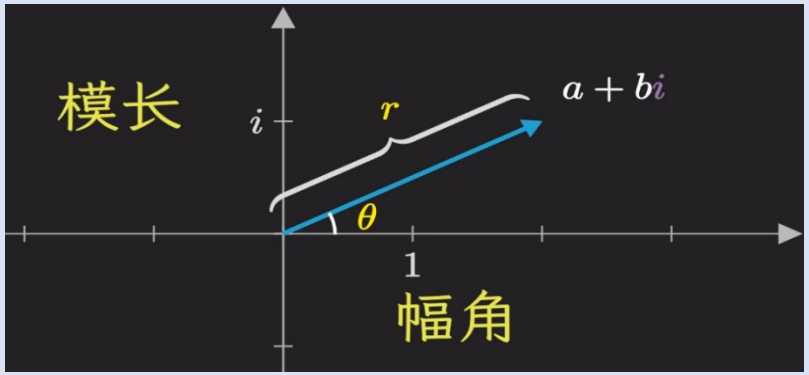
$$k_n = f_k(x_n, n) = (W_k x_n) e^{in\theta} \quad (2)$$

$$g(x_m, x_n, m-n) = R[(W_q x_m)(W_k x_n)^* e^{i(m-n)\theta}]$$

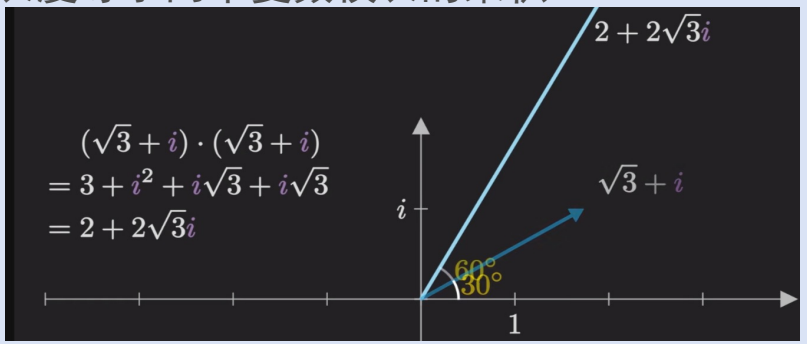
复数的乘法，后一项变成共轭复数，变成负号

旋转矩阵的定义

- 复数 $z = a + ib$可以被表示为：将向量逆时针旋转一个角度θ，并缩放模长

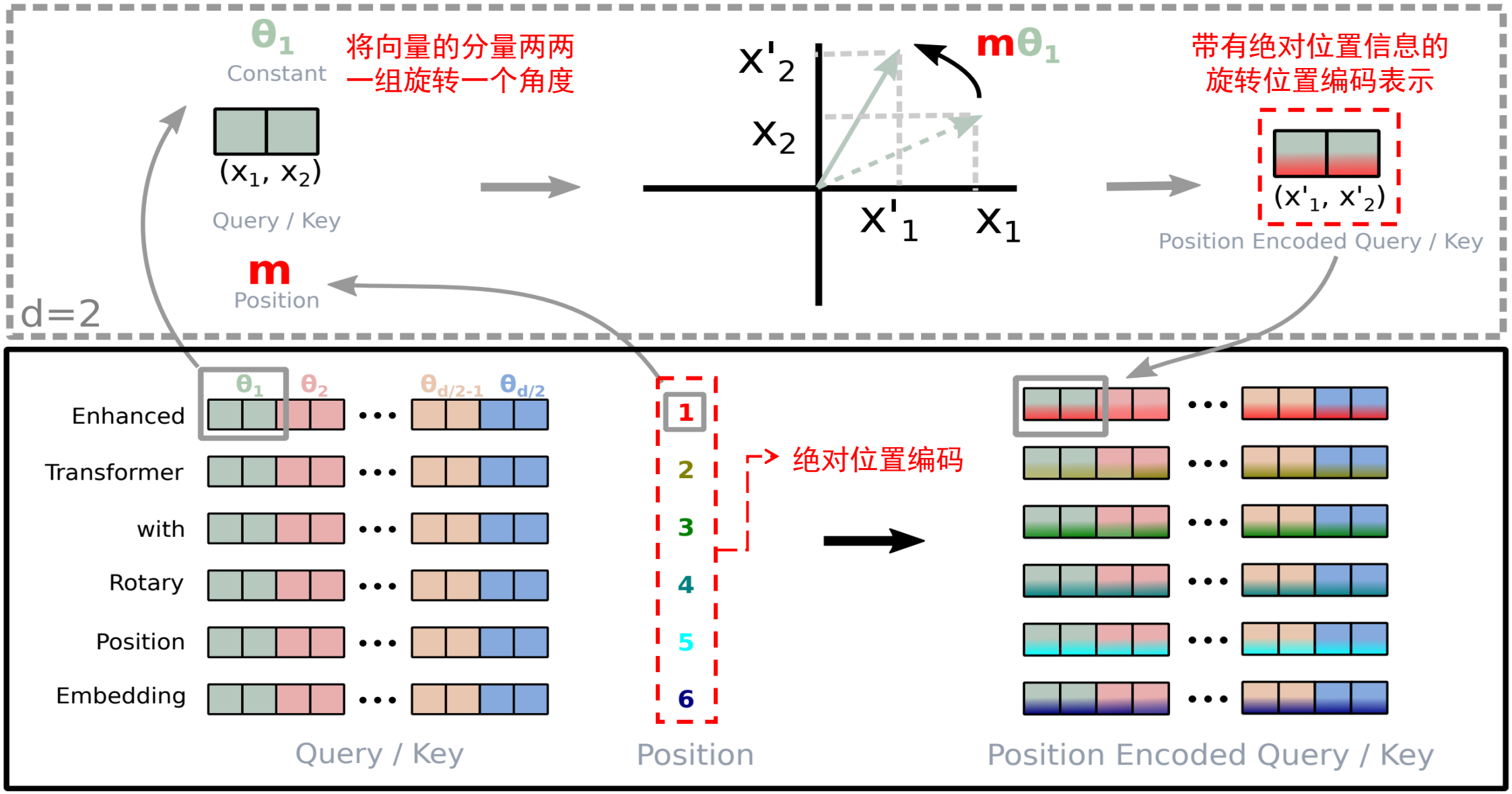


- 两复数相乘，结果的幅角等于两个复数的幅角相加，结果的长度等于两个复数模长的乘积



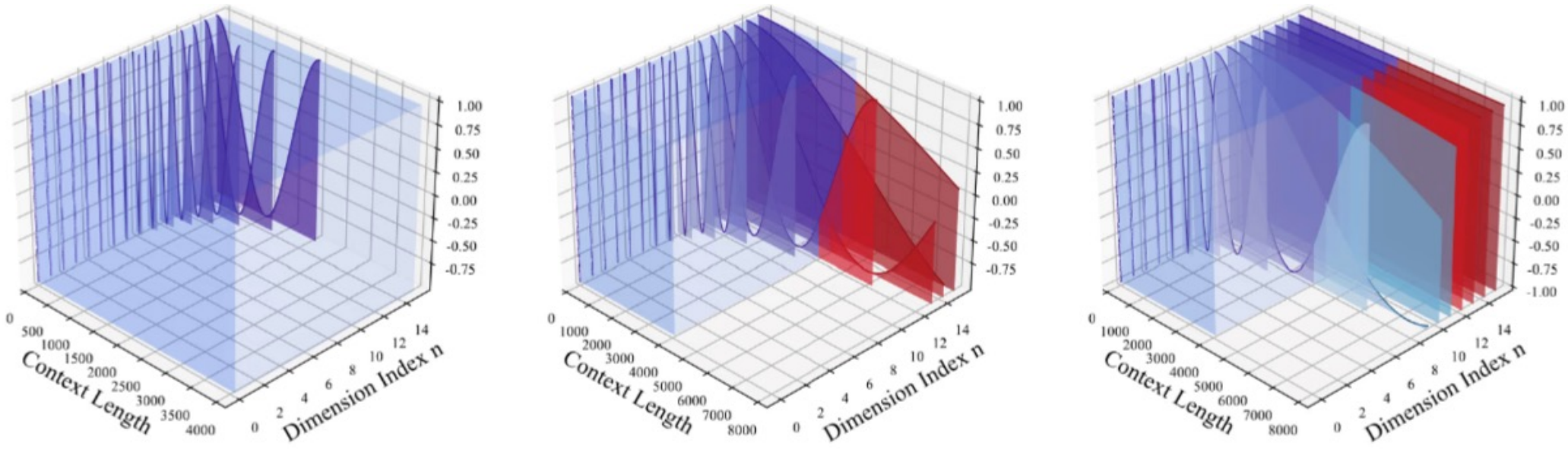
Su J, Lu Y, Pan S, et al. Enhanced transformer with rotary position embedding., 2021[J]. DOI: https://doi. org/10.1016/j. neucom, 2023

上下文窗口扩展 • 相对位置编码



上下文窗口扩展 • 相对位置编码

- 大θ意味着更平缓的波长，允许模型适应更长的上下文长度
- Llama-2 θ:10K, Llama-3 θ:500K



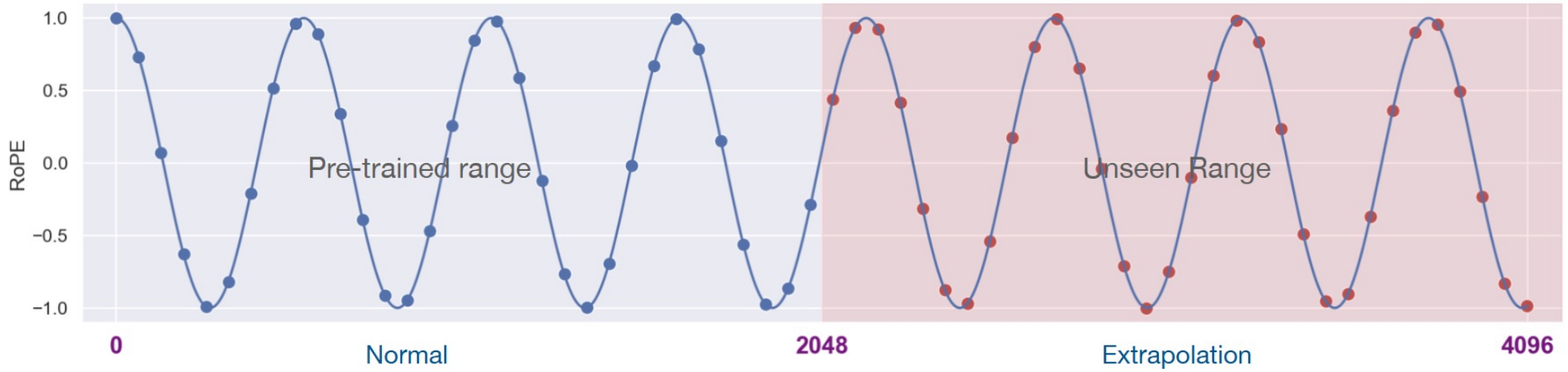
(a) RoPE base=500 (b) RoPE base=10000 (c) RoPE base=1000000

https://github.com/JunnYu/RoFormer_pytorch/blob/roformer_v2/src/roformer/modeling_roformer.py#L156-L194

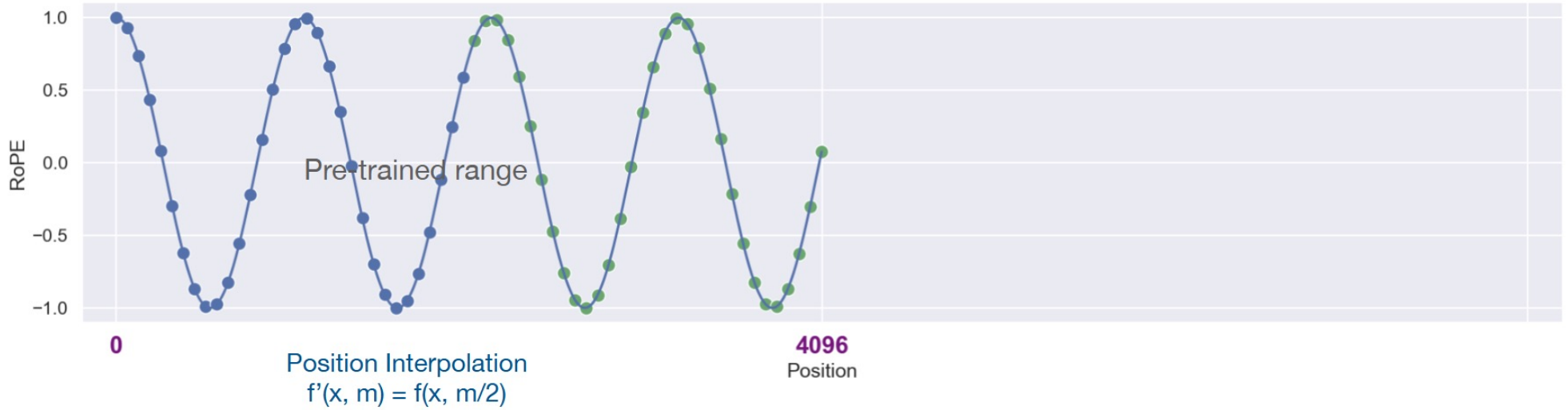
（38行代码既可实现ROPE 代码的实现）

上下文窗口扩展 • PE & PI

➢ 位置外推(PE)



➢ 位置内插(PI)



Hugging Face ➢ PE + PI = YaRN (HF 的默认缩放方式)

Extending Context Window of Large Language Models via Positional Interpolation (Meta, 2023)
YaRN: Efficient Context Window Extension of Large Language Models (ICLR 2024)

建模

开源短上下文强模型
(Llama2-4K, Llama3-8K)

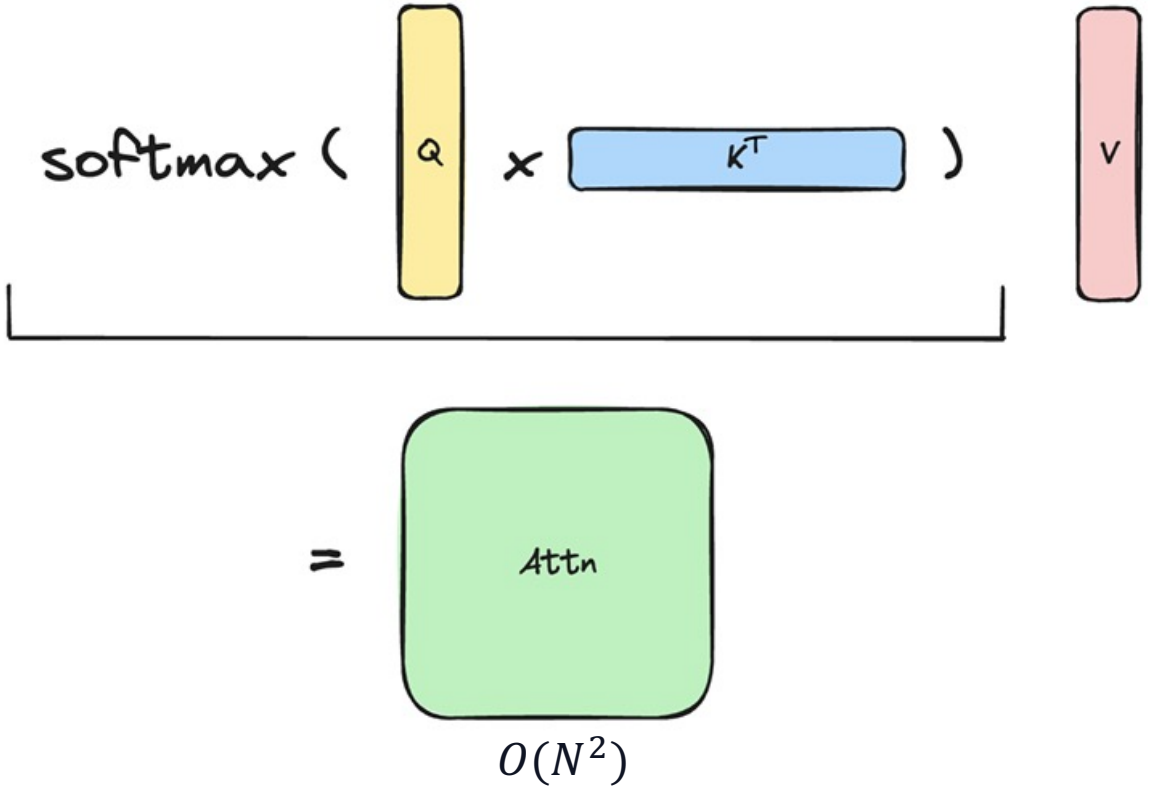
具有长上下文窗口的模型
(>32K)

强长上下文模型

- 上下文窗口扩展
  - 相对位置编码(RPE)
  - 旋转位置编码（RoPE)
  - 位置内插（PI）与外推(PE)
- 长上下文对齐
  - 监督微调(SFT)
  - 强化学习(RL)

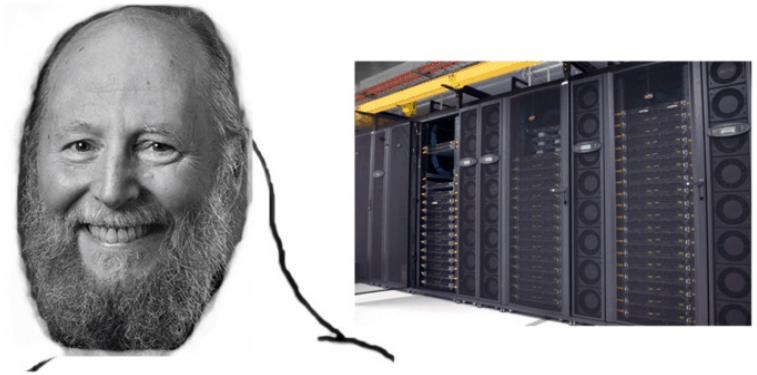
建模 • 计算复杂度

- Transformer模型的自注意力复杂度

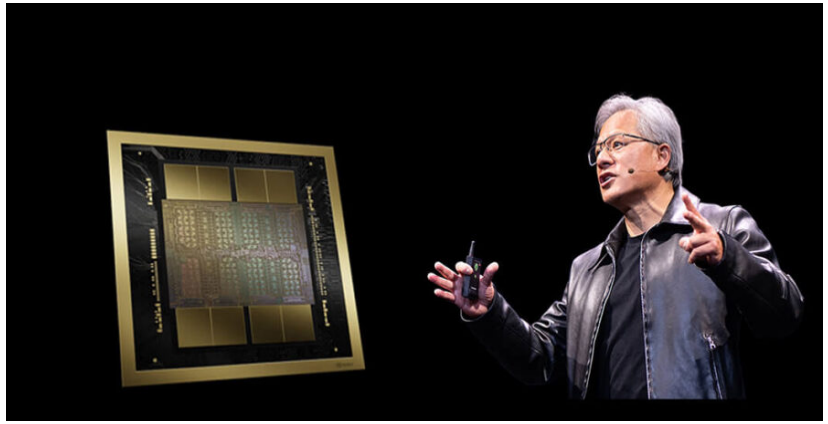


$$\text{softmax}(\frac{QK^T}{\sqrt{d}})V$$

建模 • 显存开销

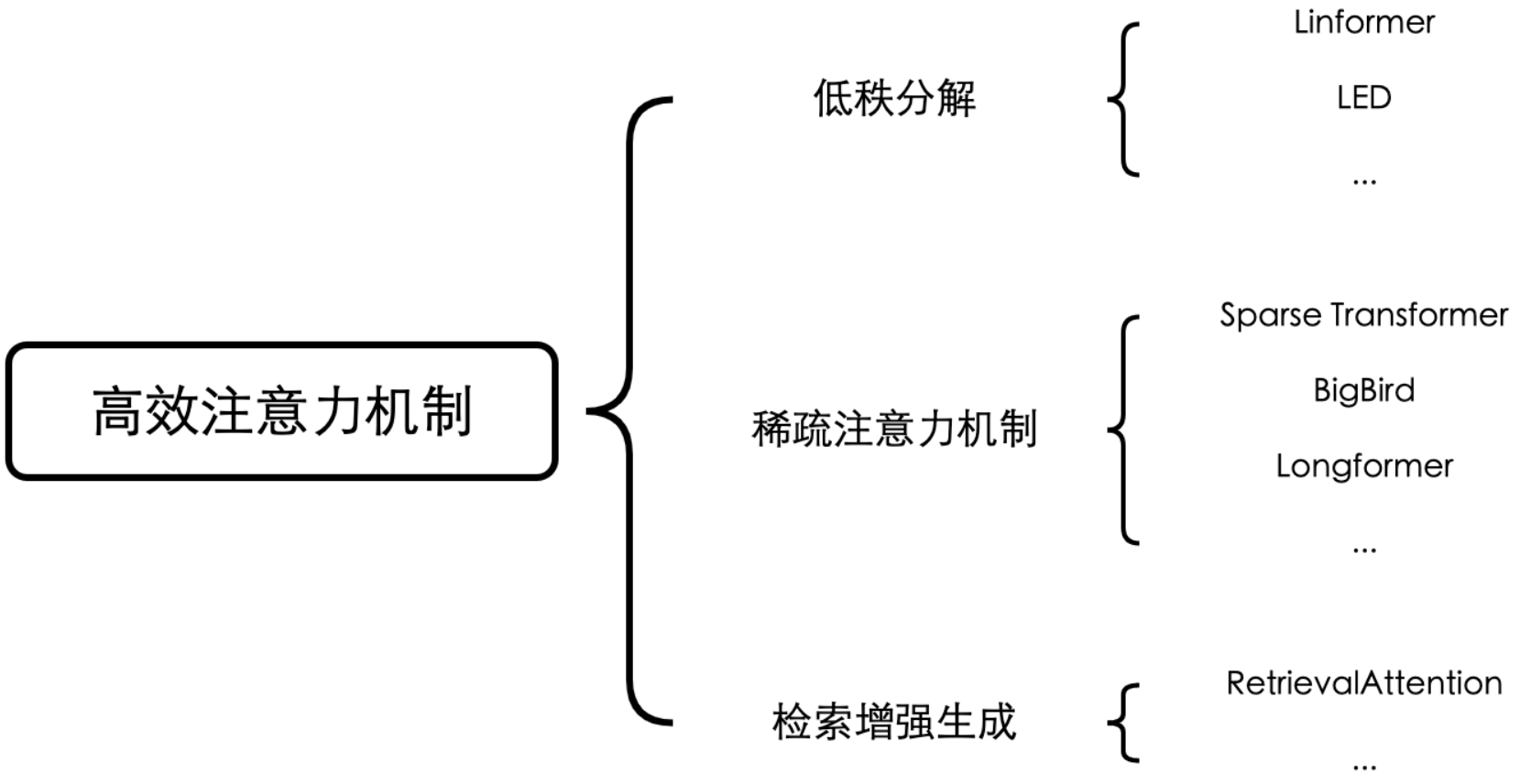


``with a batch size of 1, processing 100 million tokens requires over 1000 GB of memory for a modest model with a hidden size of 1024'' —— Ring Attention, 2023, Hao Liu et al.



- 当前高端GPU的内存
  - NVIDIA H200: 141 GB
  - AMD MI300X: 192 GB
  - NVIDIA GB200 (Blackwell): 288 GB (available late 2024)
  - *NVIDIA H100/A100/A800*: 80GB*

建模 • 高效注意力机制


高效注意力机制
低秩分解
Linformer
LED
...
稀疏注意力机制
Sparse Transformer
BigBird
Longformer
...
检索增强生成
RetrievalAttention
...

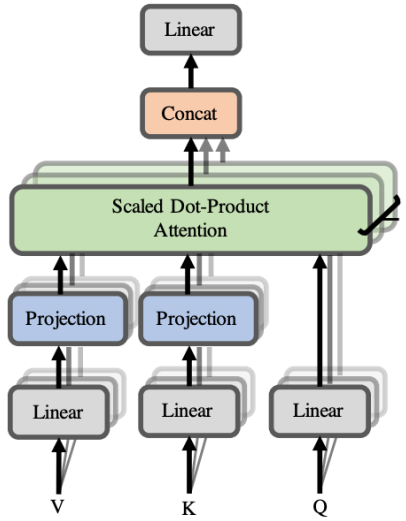
建模 • 低秩分解

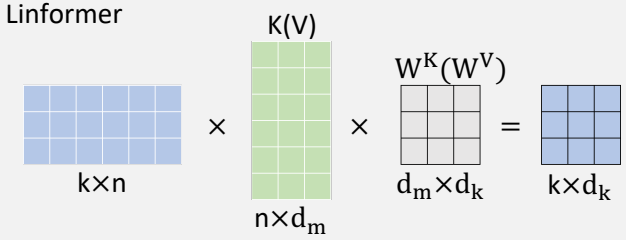
Linformer(Meta, 2020)

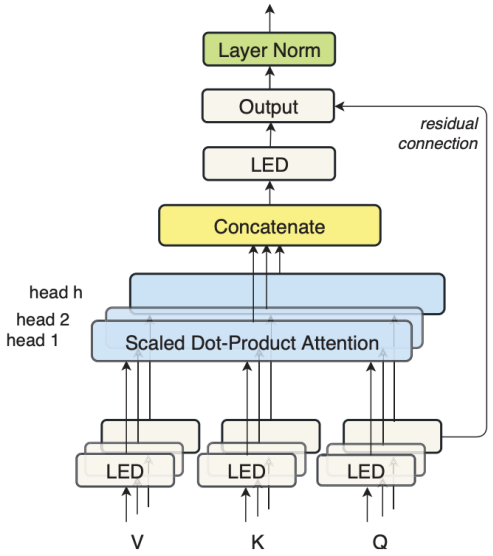
- 使用 K、V 注意力矩阵的低秩近似。
- “自动”压缩上下文长度 ≈ 丢弃“无关”的词元。

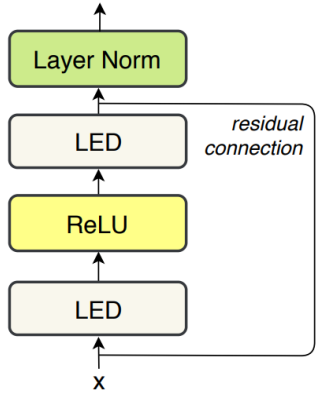
LED(ICASSP, 2022)

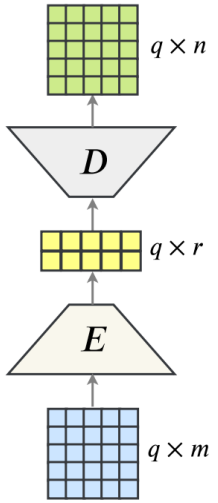
- 使用VAE模型来缩短序列长度。
- 缩小矩阵 Q、K、V 的尺寸以近似线性计算效率。











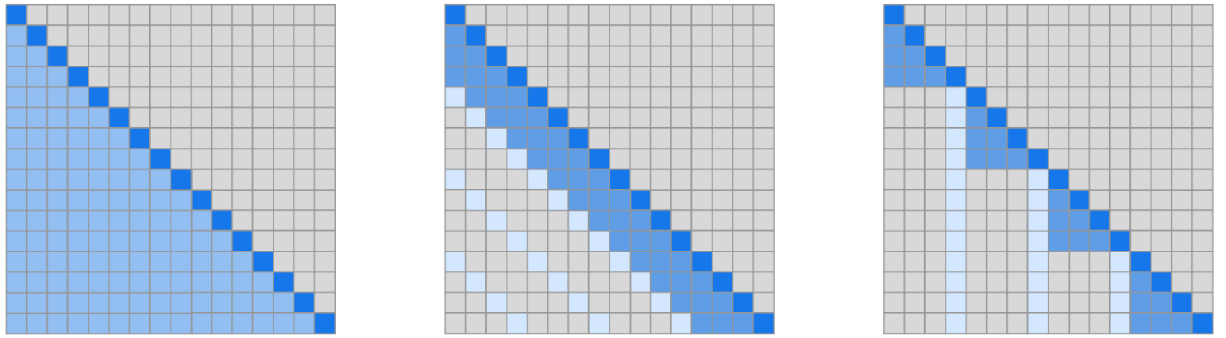
Linformer: Self-attention with linear complexity. Meta, 2020

LED: Lightweight and efficient end-to-end speech recognition using low-rank transformer. ICASSP 2022

建模 • 稀疏注意力机制

Sparse Transformer (OpenAI, 2019)

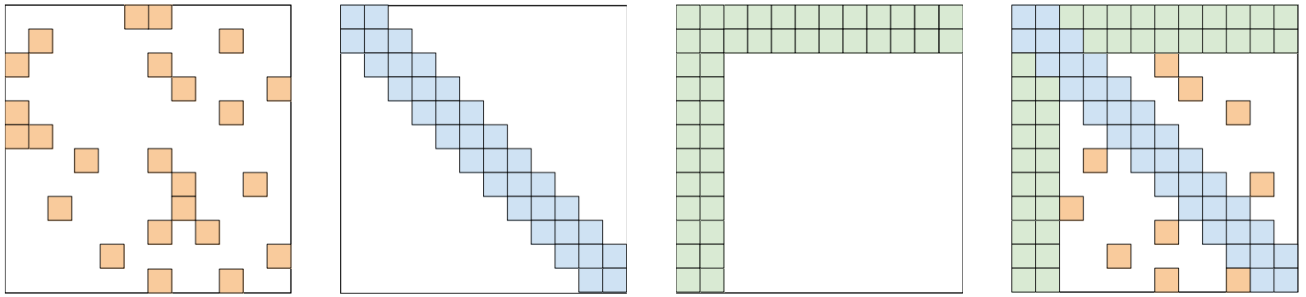
- 限制模型注意力在局部窗口内。



(a) Transformer (b) Sparse Transformer (strided) (c) Sparse Transformer (fixed)

BigBird (NeurIPS 2020)

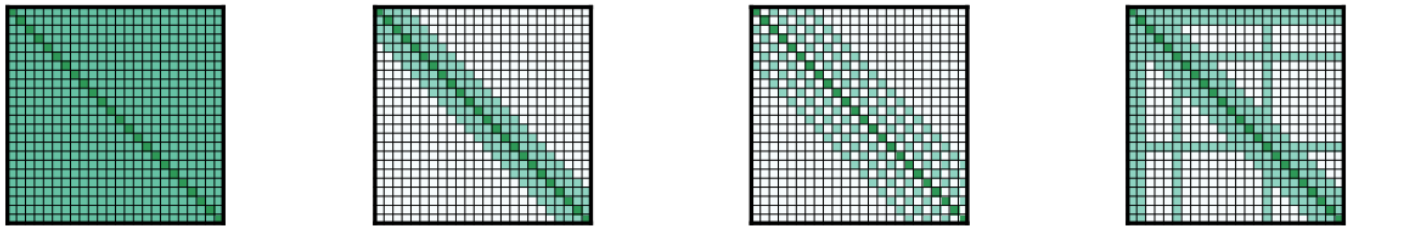
- 关注之前词元的随机子集以及几个全局可访问的词元。



(a) Random attention (b) Window attention (c) Global Attention (d) BIGBIRD

Longformer (AllenAI, 2020)

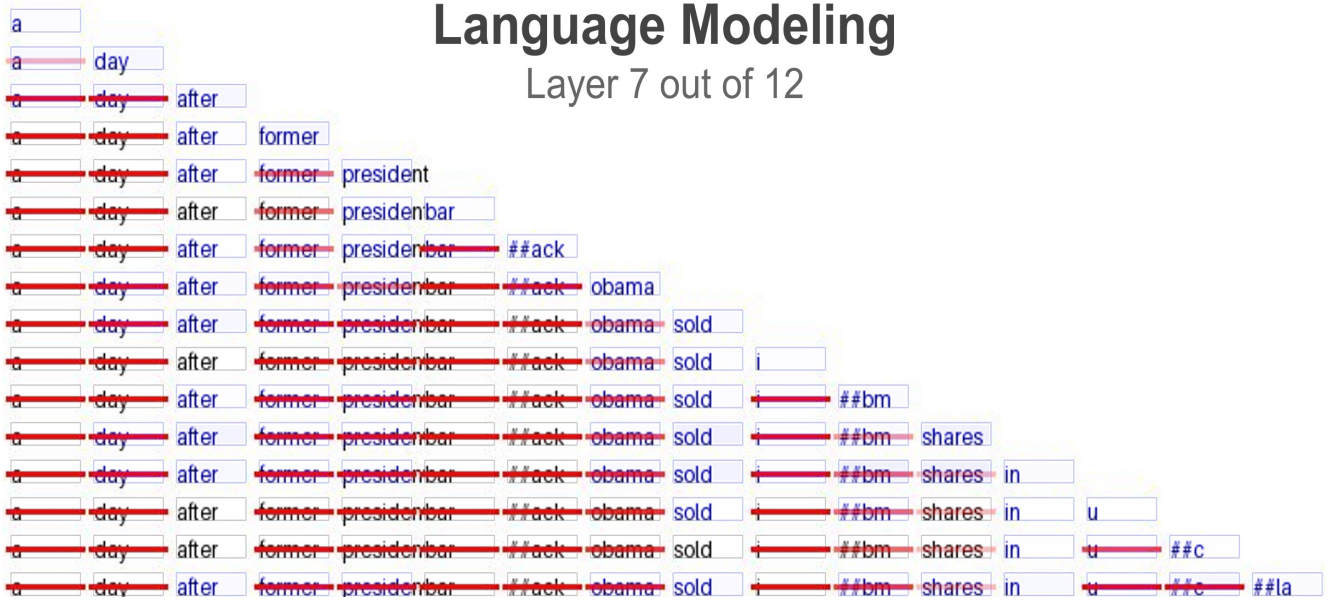
- 引入扩张滑动窗口模式，以增加注意力的感受野，并为每一层手动选择窗口大小。



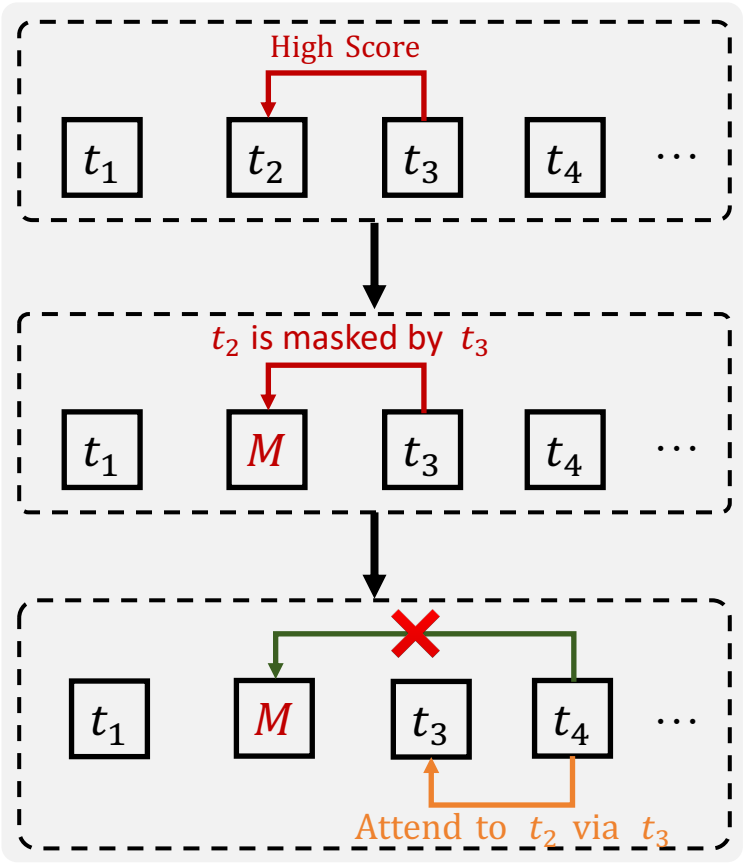
(a) Full n^2 attention (b) Sliding window attention (c) Dilated sliding window (d) Global+sliding window

建模 • 稀疏注意力机制

- 被选择性注意力掩盖的token将不会对任何之后的注意力操作产生贡献



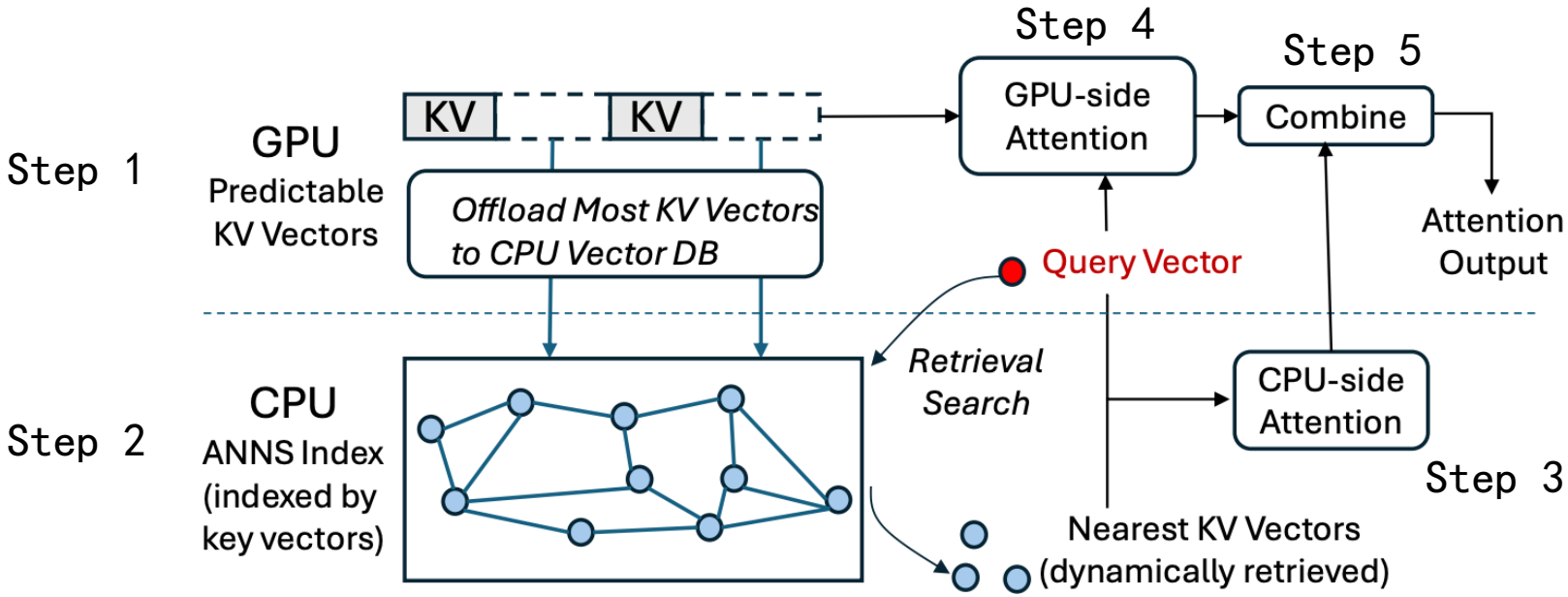
Selective Attention (Google, 2024)



Selective Attention Improves Transformer. Google, 2024

建模 • 检索增强生成

- 步骤：
  - Step 1:将大部分KV向量卸载到CPU内存中。
  - Step 2:构建ANNS（近似最近邻搜索）向量索引，并使用注意力向量搜索来找到关键的词元。
  - Step 3:CPU端动态存储大部分KV缓存，并计算CPU端的注意力分数。
  - Step 4:GPU端存储相对重要的一小部分KV缓存，并计算GPU端的注意力分数。
  - Step 5:融合双端计算的注意力结果并输出最终的注意力分数。

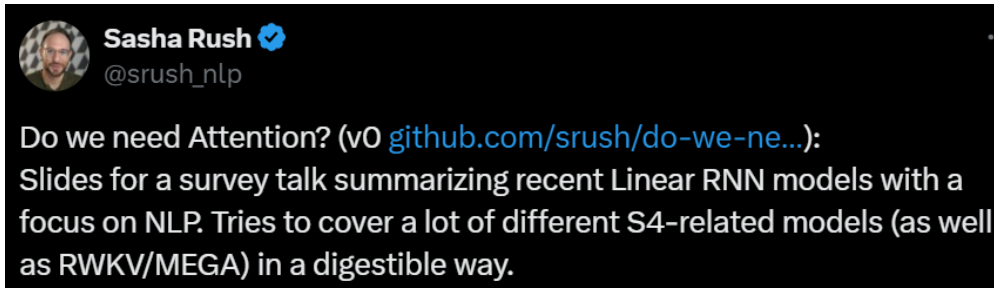


RetrievalAttention: Accelerating Long-Context LLM Inference via Vector Retrieval. Arxiv, 2024.

新模型架构 • 线性注意力

- 其他的模型架构在长文本上的运用
  - 对注意力机制进行改进，打破原有的二次方的限制
  - 采用其他的基础模型架构（CNN，RNN）

Layer Type	Complexity per Layer	Sequential Operations
Self-Attention	$O(n^2 \cdot d)$	$O(1)$
Recurrent	$O(n \cdot d^2)$	$O(n)$
Convolutional	$O(k \cdot n \cdot d^2)$	$O(1)$

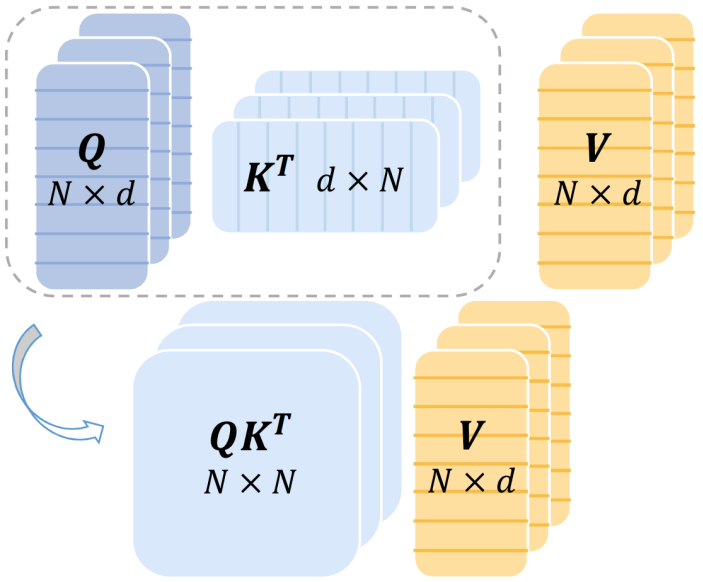


新模型架构 • 线性注意力

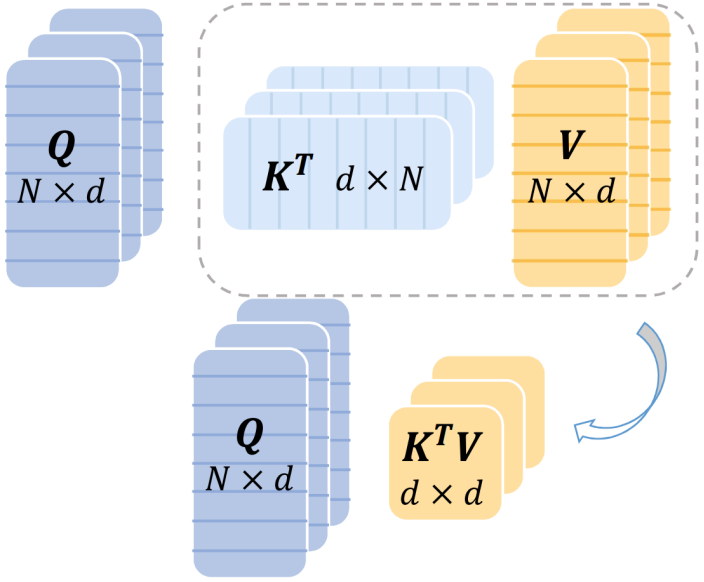
- 在原始的注意力机制中 $(Q,K,V)=\mathrm{Softmax}\left(\frac{QK^T}{\sqrt{\mathrm{d}}}\right)V$

- 在自注意力中，$Q,\ K,\ V\in R^{N\times d}$，其中 N 一般远大于 d，矩阵 QK^T 产生 $N\times N$的矩阵，导致 $O(N^2)$ 的复杂度。

- 如果没有 Softmax，我们可以先计算$K^T\ V$，得到一个$d\times d$的矩阵，因为 d 远小于N，所以最终复杂度为 $O(N)$。



Softmax Attention $\mathcal{O}(N^2d)$



Linear Attention $\mathcal{O}(Nd^2)$

新模型架构 • 线性注意力

- 使用与Softmax具有类似性质的函数代替
  - 我们可以使用核函数 $g(q_t, k_i)$ 来替换 $\exp(q_t k_i^T)$，比如说 $g(q_t, k_i) = <\phi(q_t), \phi(k_i)>$
  - 核函数应该具有原先Softmax类似的特性，即表示相似程度以及非负性，同时满足计算简单

原始注意力机制

$$Q, K, V = XW_Q, XW_K, XW_V$$

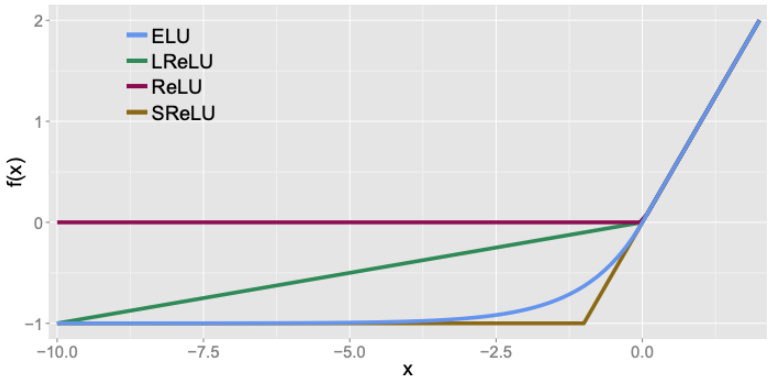
$$O = \text{softmax}\left(\left(\frac{QK^T}{\sqrt{d}}\right) \odot M\right) V,$$

其中 $W_Q, W_K, W_V \in R^{d\times d}$ 是可学习矩阵，M 是掩码矩阵。

$$q_t, k_t, v_t = x_t W_Q, x_t W_K, x_t W_V \quad o_t = \frac{\sum_{i=1}^{t} \exp(q_t k_i^T) v_i}{\sum_{i=1}^{t} \exp(q_t k_i^T)}$$

通过当前输入的表示 $x_t \in R^{1\times d}$ 分别经过矩阵投影得到 q_t, k_t 以及 v_t.

$$o_t = \frac{\sum_{i=1}^{t} \phi(q_t)\phi(k_i)^T v_i}{\sum_{i=1}^{t} \phi(q_t)\phi(k_i)^T} = \frac{\phi(q_t)\sum_{i=1}^{t} \phi(k_i)^T v_i}{\phi(q_t)\sum_{i=1}^{t} \phi(k_i)^T}$$



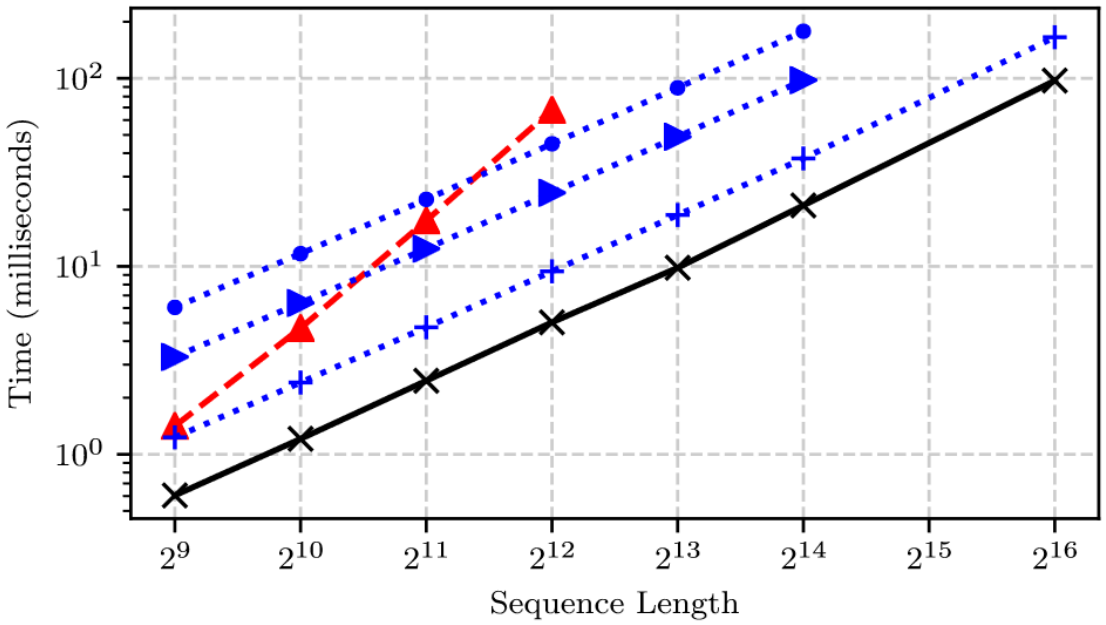
我们可以进一步将线性注意力机制改写为RNN的形式

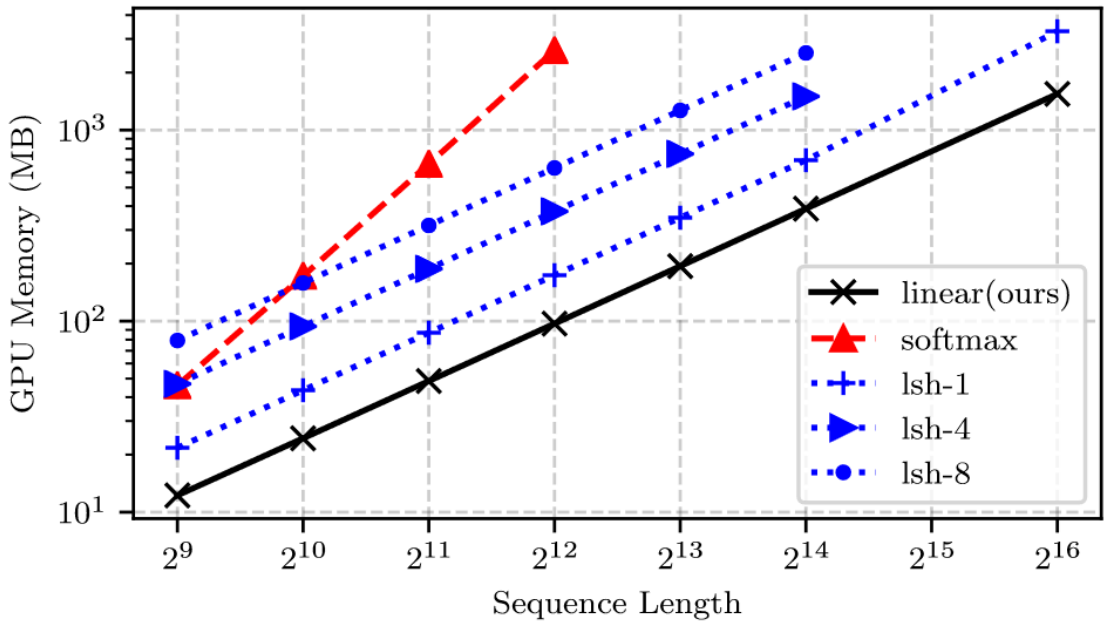
$$S_t = \sum_{i=1}^{t} \phi(k_i)^T v_i \ (S_t \in R^{d\times d}), \quad z_t = \sum_{i=1}^{t} \phi(k_i)^T \ (z_t \in R^{d\times 1}) \Rightarrow \begin{aligned} S_t &= S_{t-1} + \phi(k_t)^T v_t, \\ z_t &= z_{t-1} + \phi(k_t)^T, \\ o_t &= \frac{\phi(q_t) S_t}{\phi(q_t) z_t} \end{aligned}$$

Li R, Su J, Duan C, et al. Linear attention mechanism: An efficient attention for semantic segmentation[J]. arXiv preprint arXiv:2007.14902, 2020.

新模型架构 • 线性注意力

- 线性注意力机制(黑线)取得了在时间和GPU存储上取得了最优的效率。





新模型架构 • 状态空间模型

Albert Gu
Assistant Professor
@ CMU

Tri Dao
Assistant Professor
@ Princeton

“ Quadratic attention has been indispensable for information-dense modalities such as language …

Announcing Mamba: a new SSM arch. that has linear time scaling, **ultra long context**, and most importantly outperforms Transformers everywhere we've tried ”

新模型架构 • 状态空间模型

Transformer
(NeurIPS,2017)

H3 (ICLR,2023)

Hyena (ICML,2023)

Monarch (NeurIPS,2023)

Mamba (COLM,2024)

- **状态空间方程（SSM）**
  - $h'_t = Ah(t) + Bx(t)$
  - $y_t = Ch'(t) + Dx(t)$
- **结构化状态空间方程（S4）**
  - $h_t = \bar{A}h_{t-1} + \bar{B}x_t$
  - $y_t = Ch_t$
- **选择性结构状态空间方程（Mamba）**
  - $h_t = s_{\bar{A}}(x_t)h_{t-1} + s_{\bar{B}}(x_t)x_t$
  - $y_t = s_{\bar{C}}(x_t)h_t$

新模型架构 • 选择性结构状态空间模型(Mamba)

- 对于所有模型大小，Mamba在每一个评测数据集中都取得了最优的测试结果后，达到了两倍模型大小的基线模型的效果。

Model	Token.	Pile ppl ↓	LAMBADA ppl ↓	LAMBADA acc ↑	HellaSwag acc ↑	PIQA acc ↑	Arc-E acc ↑	Arc-C acc ↑	WinoGrande acc ↑	Average acc ↑
Hybrid H3-130M	GPT2	—	89.48	25.77	31.7	64.2	44.4	24.2	50.6	40.1
Pythia-160M	NeoX	29.64	38.10	33.0	30.2	61.4	43.2	24.1	**51.9**	40.6
Mamba-130M	NeoX	**10.56**	**16.07**	**44.3**	**35.3**	**64.5**	**48.0**	**24.3**	**51.9**	**44.7**
Hybrid H3-360M	GPT2	—	12.58	48.0	41.5	68.1	51.4	24.7	54.1	48.0
Pythia-410M	NeoX	9.95	10.84	51.4	40.6	66.9	52.1	24.6	53.8	48.2
Mamba-370M	NeoX	**8.28**	**8.14**	**55.6**	**46.5**	**69.5**	**55.1**	**28.0**	**55.3**	**50.0**
Pythia-1B	NeoX	7.82	7.92	56.1	47.2	70.7	57.0	27.1	53.5	51.9
Mamba-790M	NeoX	**7.33**	**6.02**	**62.7**	**55.1**	**72.1**	**61.2**	**29.5**	**56.1**	**57.1**
GPT-Neo 1.3B	GPT2	—	7.50	57.2	48.9	71.1	56.2	25.9	54.9	52.4
Hybrid H3-1.3B	GPT2	—	11.25	49.6	52.6	71.3	59.2	28.1	56.9	53.0
OPT-1.3B	OPT	—	6.64	58.0	53.7	72.4	56.7	29.6	59.5	55.0
Pythia-1.4B	NeoX	7.51	6.08	61.7	52.1	71.0	60.5	28.5	57.2	55.2
RWKV-1.5B	NeoX	7.70	7.04	56.4	52.5	72.4	60.5	29.4	54.6	54.3
Mamba-1.4B	NeoX	**6.80**	**5.04**	**64.9**	**59.1**	**74.2**	**65.5**	**32.8**	**61.5**	**59.7**
GPT-Neo 2.7B	GPT2	—	5.63	62.2	55.8	72.1	61.1	30.2	57.6	56.5
Hybrid H3-2.7B	GPT2	—	7.92	55.7	59.7	73.3	65.6	32.3	61.4	58.0
OPT-2.7B	OPT	—	5.12	63.6	60.6	74.8	60.8	31.3	61.0	58.7
Pythia-2.8B	NeoX	6.73	5.04	64.7	59.3	74.0	64.1	32.9	59.7	59.1
RWKV-3B	NeoX	7.00	5.24	63.9	59.6	73.7	67.8	33.1	59.6	59.6
Mamba-2.8B	NeoX	**6.22**	**4.23**	**69.2**	**66.1**	**75.2**	**69.7**	**36.3**	**63.5**	**63.3**

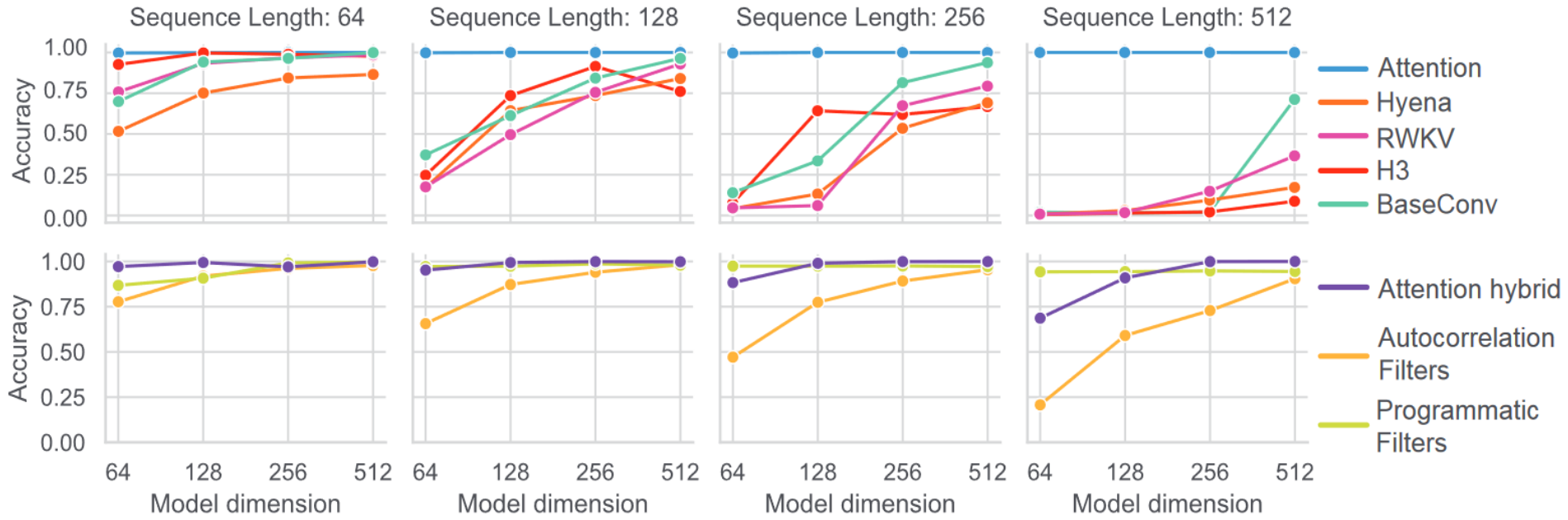
新模型架构 • 选择性结构状态空间模型(Mamba)

- 线性时间复杂度模型还是落后于使用注意力机制的模型，特别是在召回任务上。
  - 注意力机制的状态空间随着序列长度增加而增加，能始终关注全局信息。
  - 状态空间模型始终维持一个固定大小的状态空间，会出现远程遗忘等问题。

Simran Arora
PhD student
@ Stanford

Hazy Research Blog, Stanford

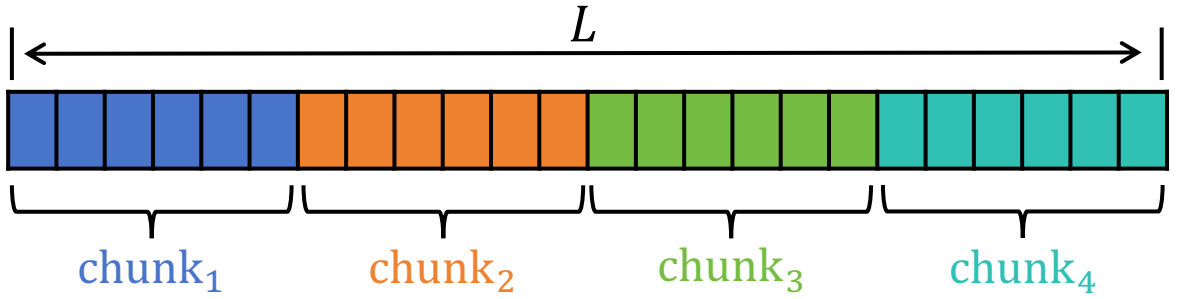
" We found that all architectures obeyed a fundamental tradeoff: the less memory the model consumed during inference, the worse it did on associative recall. In attention, the state is referred to as the KV-cache, and it grows with length of sequence. "



高效长文本对齐 • 面向硬件优化

- 序列长度(L)过长，导致查询向量$Q(L)$ 与键向量$K(L)$之间的点积消耗大量的GPU显存与算力。
- 是否可以通过【分段】的方式缓解这个现象？

- 将 K,V 分成 N 段：$K=\{k_1,k_2,\cdots,k_N\}$，$V=\{v_1,v_2,\cdots,v_N\}$ 每段的长度是B。
- 【分段】注意力显著减少计算复杂度，所需要的计算显存也会减少



Token级别的点积计算

原始的注意力机制

$$\text{Attn}(Q,K,V)=\text{softmax}\left(\frac{QK^T}{\sqrt{d}}\right)V$$

令 $w_{xy}=\frac{q_x k_y^T}{\sqrt{d}}$，则 $\text{Attn}(q_x,K,V)=\frac{\sum_{y=1}^{L} e^{w_{xy}} v_y}{\sum_{y=1}^{L} e^{w_{xy}}}$

每段点积的结果

分段注意力机制

$$\text{Attn}(q_x,K_i,V_i)=\frac{\sum_{y=i\times B}^{(i+1)\times B} e^{w_{xy}} v_y}{\sum_{y=i\times B}^{(i+1)\times B} e^{w_{xy}}}$$

时间复杂度 $O(L^2)$ ⟶ 时间复杂度 $O\left(\left(\frac{L}{4}\right)^2\times 4\right)=O(\frac{L^2}{4})$

高效长文本对齐 • 面向硬件优化

从分段注意力转换为整段序列注意力计算：将分段的注意力计算转换成迭代计算的形式

传统注意力机制 == 将所有的段注意力合并

将分子和分母分别换元：

$$令\ \mathrm{Attn}(q_x, K_i, V_i) = \frac{\sum_{y=i\times B}^{(i+1)\times B} e^{w_{xy}} v_y}{\sum_{y=i\times B}^{(i+1)\times B} e^{w_{xy}}} = \frac{\sum_{j=1}^{n} A_j}{\sum_{j=1}^{n} B_j},\ B_{1\cdots n} = \sum_{i=1}^{n} B_i,\ A_{1\cdots n} = \sum_{i=1}^{n} A_i$$

自注意力的计算可以写成迭代的形式：

$$\mathrm{attn}_{12} = \frac{A_{12}}{B_{12}} = \frac{A_1+A_2}{B_1+B_2} = \frac{A_1}{B_1} * \frac{B_1}{B_1+B_2} + \frac{A_2}{B_2} * \frac{B_2}{B_1+B_2} = \mathrm{attn}_1 \frac{B_1}{B_{12}} + \mathrm{attn}_2 \frac{B_2}{B_{12}}$$

$$\mathrm{attn}_{123} = \mathrm{attn}_{12} \frac{B_{12}}{B_{123}} + \mathrm{attn}_3 \frac{B_3}{B_{123}}$$

局部片段的注意力计算

$$\cdots$$

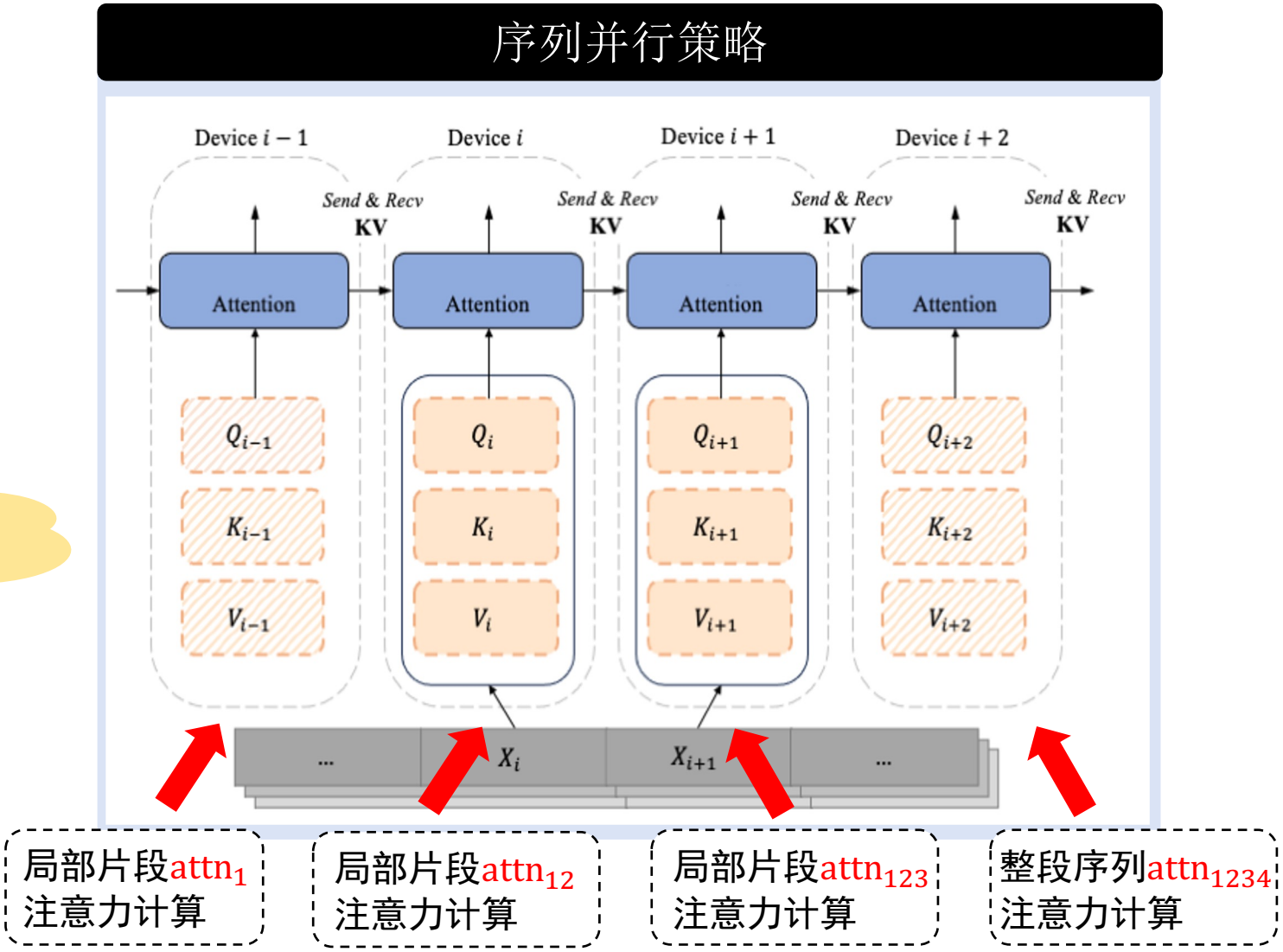
$$\mathrm{attn}_{1\cdots n} = \mathrm{attn}_{1\cdots n-1} \frac{B_{1\cdots n-1}}{B_{1\cdots n}} + \mathrm{attn}_n \frac{B_n}{B_{1\cdots n}}$$

整段序列的注意力计算

高效长文本对齐 • 面向硬件优化

序列并行策略(Sequence Parallel)

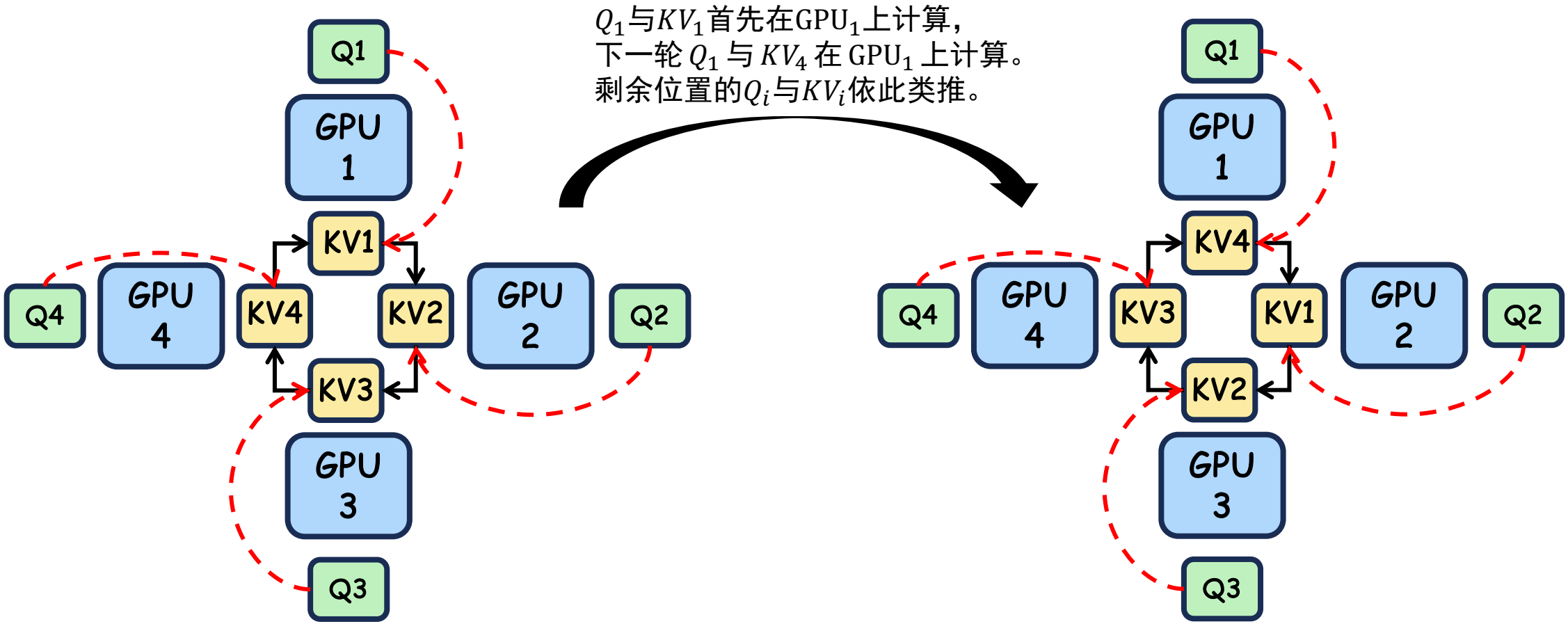
会存在等待气泡（bubbles），
可以进一步优化。



Li S, Xue F, Baranwal C, et al. Sequence parallelism: Long sequence training from system perspective[J]. arXiv preprint arXiv:2105.13120, 2021.

高效长文本对齐 • 面向硬件优化

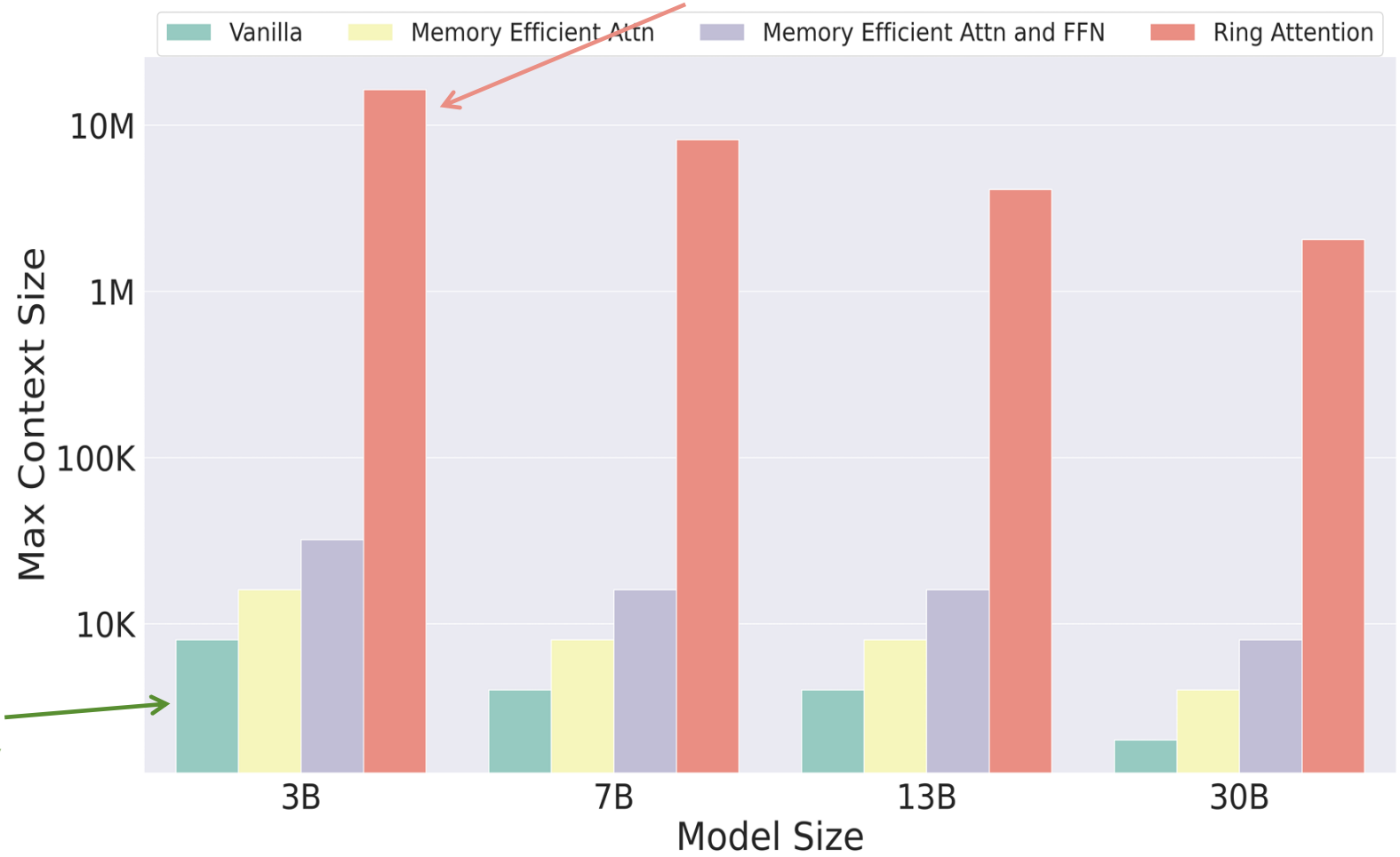
- 缓解等待气泡的策略：Ring Attention
  - 导致气泡的原因：后续的每个片段等待上一个片段计算出来，然后才合并。
  - 缓解策略：不在计算过程中合并，而是计算所有的中间结果，最后一起合并。



Liu H, Zaharia M, Abbeel P. Ring attention with blockwise transformers for near-infinite context[J]. arXiv preprint arXiv:2310.01889, 2023.

高效长文本对齐 • 面向硬件优化

Ring attention相比于其他注意力策略，可以显著提高训练数据长度上限

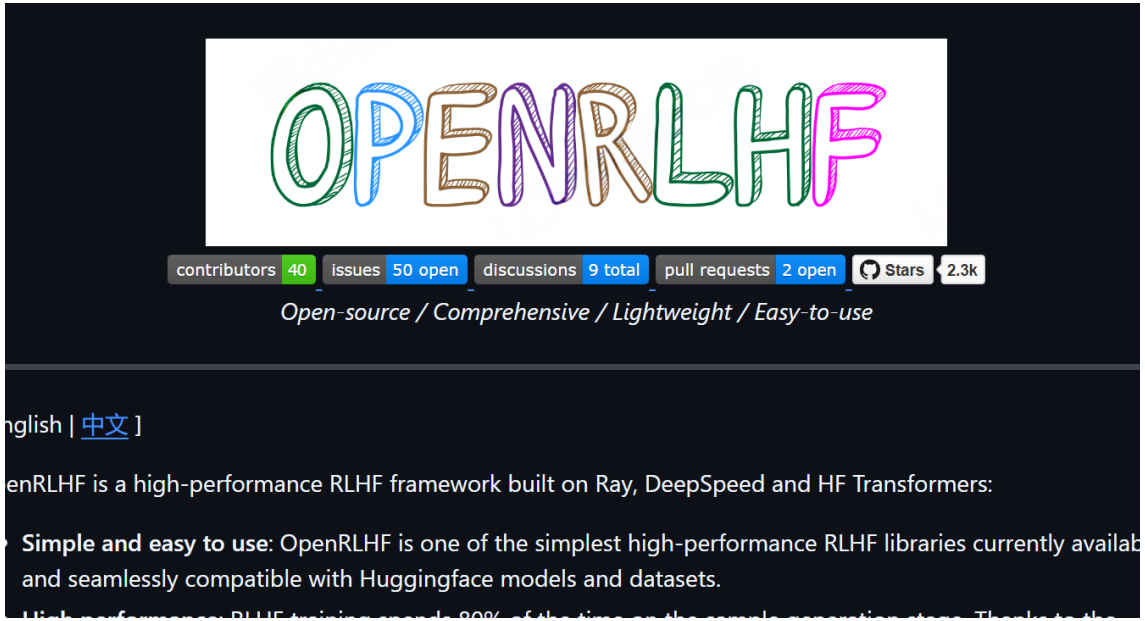


原始的注意力策略在10K上下文范围之内消耗完GPU显存

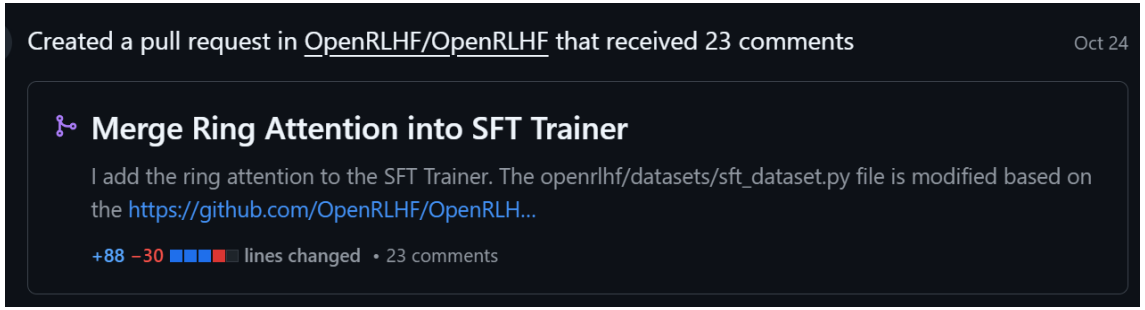
高效长文本对齐 • 面向硬件优化

开源的库已经实现了Ring Attention

https://github.com/jzhang38/EasyContext



https://github.com/OpenRLHF/OpenRLHF



建模

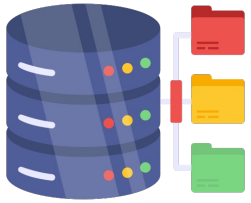
开源短上下文强模型
(Llama2-4K, Llama3-8K)

具有长上下文窗口的模型
(>32K)

强长上下文模型

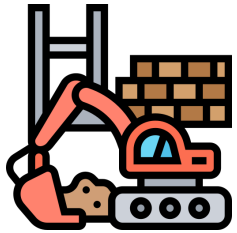
- 上下文窗口扩展
  - 相对位置编码(RPE)
  - 旋转位置编码（RoPE)
  - 位置内插（PI）与外推(PE)
- 长上下文对齐
  - 监督微调(SFT)
  - 强化学习(RL)

数据



数据资源

- 生成式预训练数据
  - 代码
  - 书籍
  - 网站
  - 等等
- 合成数据



构建方法

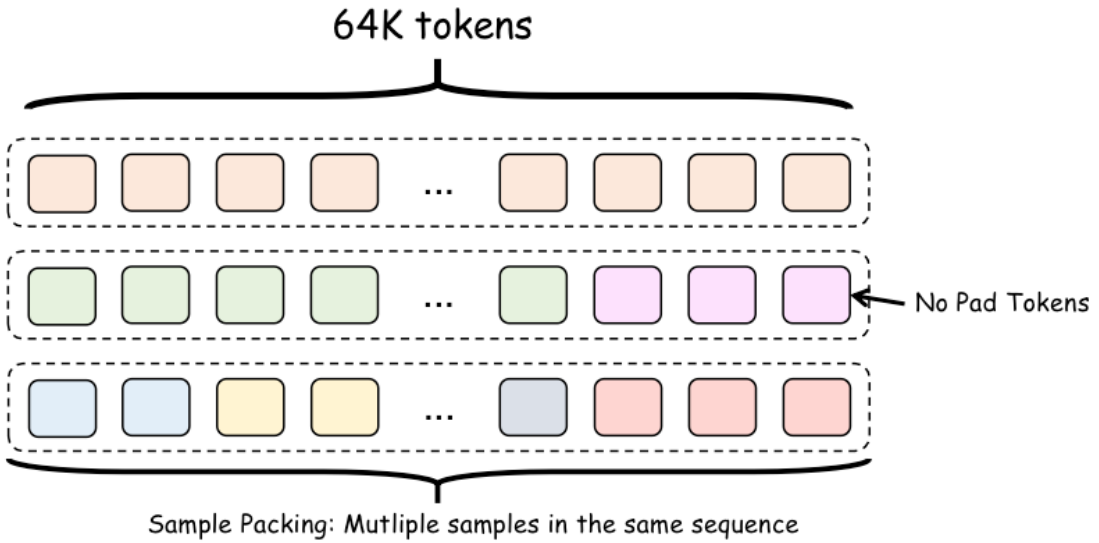
- 拼接
- 上采样
- 位置合成
- 模型生成

使用场景

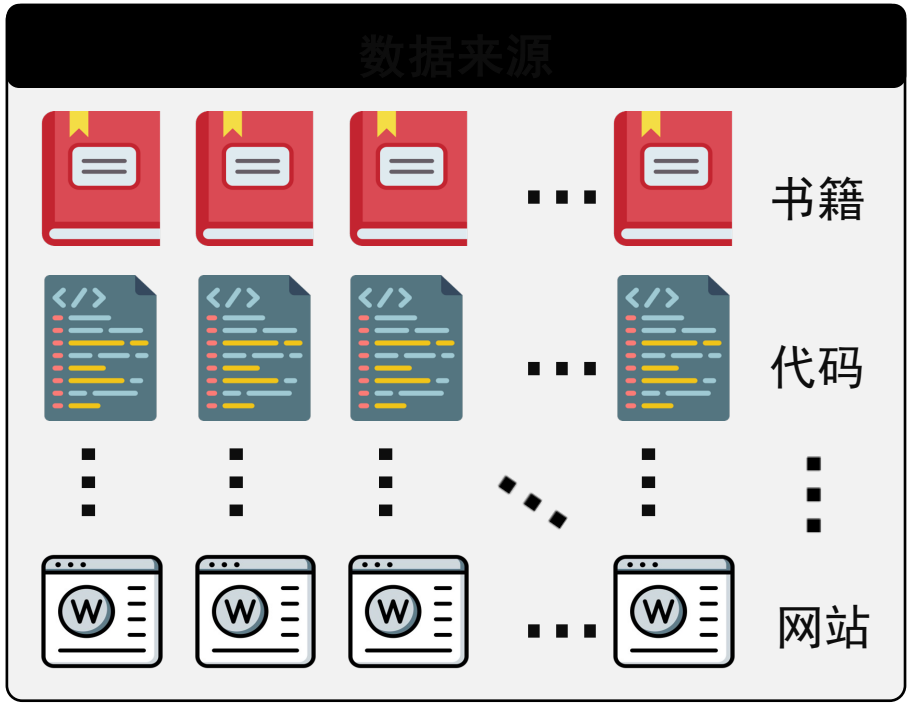
- 监督微调
- 强化学习

构造策略 • 拼接&上采样

- **拼接**：从不同数据源中提取短文本数据，并对其进行分块和随机采样，然后进行拼接



- **采样**：直接搜索具有较长上下文长度的文本

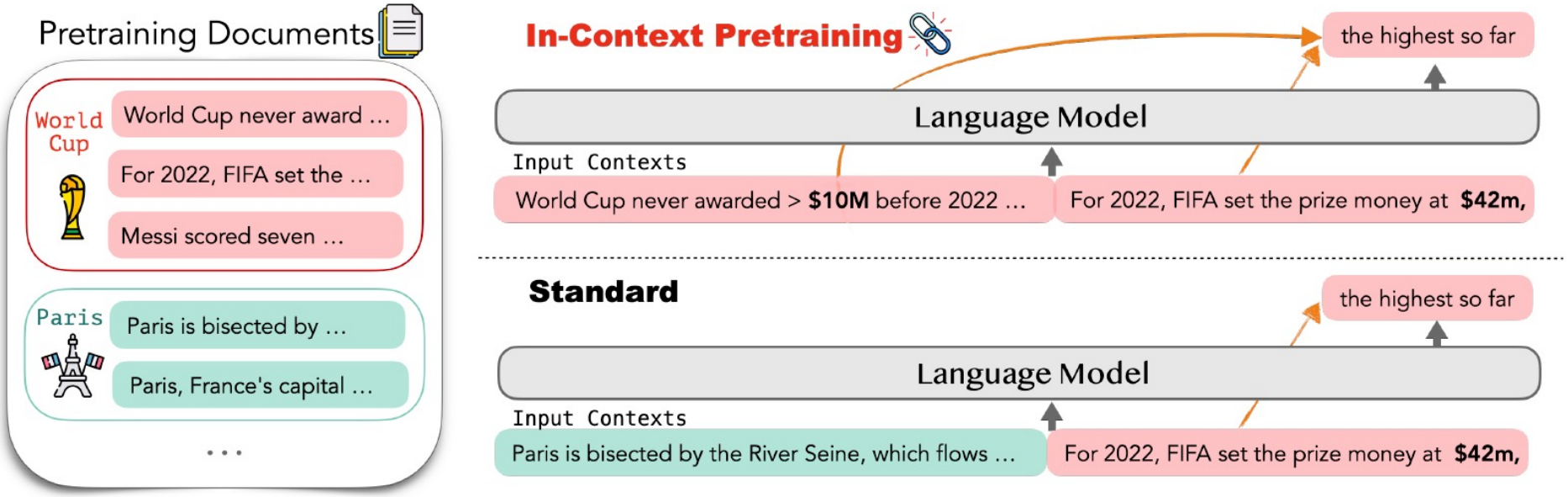


构造策略 • 拼接

- 基于相似性的方法(ICLM)
  - 基于随机采样的方法：从预训练数据集中随机采样片段进行拼接。
  - ICLM：从预训练数据集中采样与当前片段相关的文本进行拼接

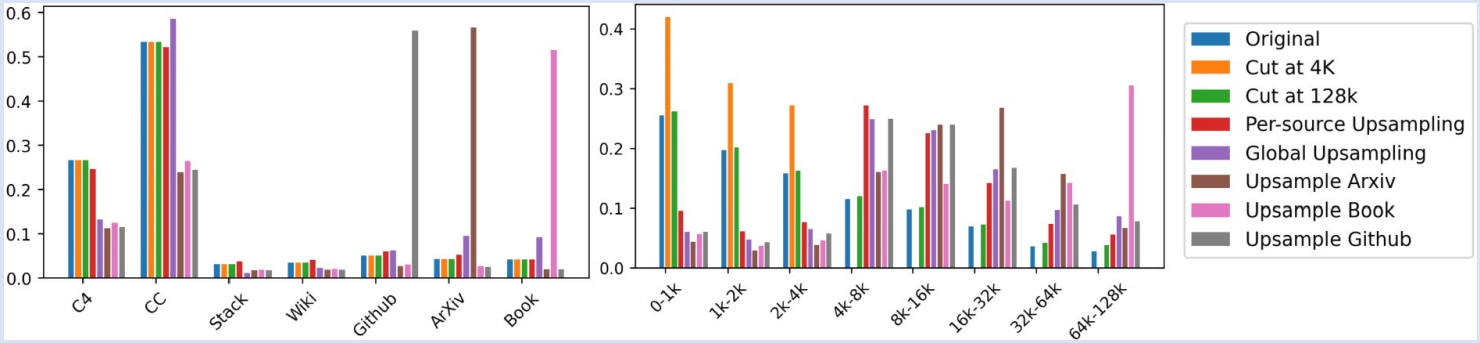
- 使用拼接方法构建数据时，应考虑
  - 上下文相关性
  - 均衡的段落质量
  - 段落是否会对最终答案产生影响

低质量 → 经常用于模型上下文窗口的扩展



ICLM (In-context Pretraining: Language Modeling beyond Document Boundaries, ICLR 2024)

构造策略 • 上采样



a）预训练语料库中存在长文本数据（书籍和代码）

	C4	CC	Stack	Arxiv	Wiki	Book	Github
0 - 4K Context Length							
Original	2.038	1.760	1.519	1.660	1.424	2.085	0.907
v.s. Per-source	+ .002	+ .008	– .001	– .008	– .040	– .065	– .008
v.s. Global	+ .008	+ .010	+ .015	– .020	– .020	– .140	+ .015
v.s. Code↑	+ .010	+ .016	+ .010	+ .006	– .026	+ .030	– .023
v.s. Book↑	+ .010	+ .016	+ .021	+ .000	– .010	– .175	+ .029
v.s. Arxiv↑	+ .006	+ .016	+ .013	– .060	– .030	+ .040	+ .025
4K - 128K Context Length							
Original	1.560	1.650	0.786	1.075	1.313	1.852	0.447
v.s. Per-source	– .010	– .010	– .006	– .011	– .044	– .014	+ .002
v.s. Global	– .010	– .006	– .001	– .016	– .040	– .018	– .007
v.s. Code↑	– .008	– .002	– .003	– .007	– .042	– .010	– .029
v.s. Book↑	– .010	– .006	+ .001	– .007	– .037	– 0.30	+ .000
v.s. Arxiv↑	– .008	– .002	+ .002	– .036	– .039	– .010	– .004

使用80k长度数据训练，使用不同长度数据测试

b）数据分布不均；从单一领域采样数据可能会影响模型在其他领域的表现，特别是在短文本测试中。

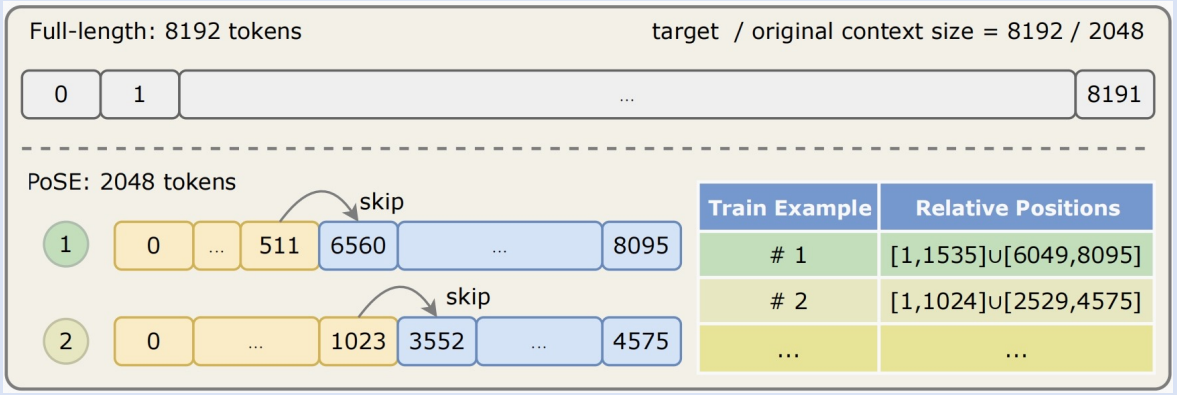
□ 利用自然的长文本数据进行训练

	Ctx.	Needle.	MMLU
Non-LLaMA Models			
GPT-4-Turbo	128K	**87.1**	**86.4**
GPT-3.5-Turbo	16K	-	67.3
YaRN Mistral 7B	128K	57.4	59.4
LLaMA-2 7B Based Models			
Together LLaMA-2 7B	32K	27.9	**44.8**
LongChat v1.5 7B	32K	18.0	42.3
LongLoRA 7B	100K	70.0	37.9
Ours LLaMA-2 7B	80K	**88.0**	43.3
LLaMA-2 13B Based Models			
LongLoRA 13B	64K	54.1	50.1
Ours LLaMA-2 13B	64K	**90.0**	**52.4**

c）从预训练语料库中上采样长文本数据可以保持模型在短文本测试集上的表现。

构造策略 • 位置索引合成

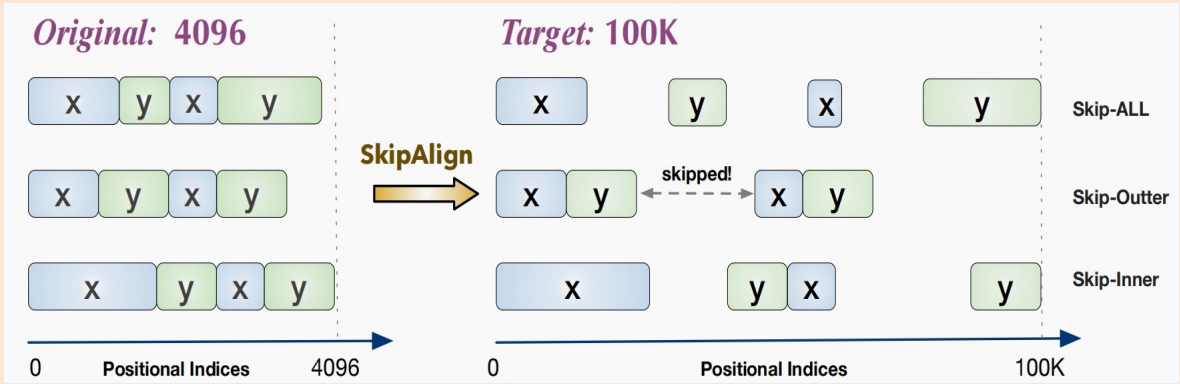
- 我们可以通过操作位置索引来构建“长”上下文数据，而不是增加实际输入序列的长度。



Positional Skip-wisE (PoSE), ICLR 2024

Method	Context size	GovReport					Proof-pile				
	Train / Target	2k	4k	8k	16k	32k	2k	4k	8k	16k	32k
Original	- / -	4.74	$> 10^3$	$> 10^3$	$> 10^3$	$> 10^3$	2.83	$> 10^3$	$> 10^3$	$> 10^3$	$> 10^3$
Full-length	16k / 16k	4.87	4.70	4.61	4.59	-	2.93	2.71	2.58	2.53	-
RandPos	2k / 16k	11.63	11.17	11.54	15.16	-	7.26	6.83	6.76	7.73	-
	2k / 32k	93.43	95.85	91.79	93.22	97.57	60.74	63.54	60.56	63.15	66.47
PoSE (Ours)	2k / 16k	4.84	4.68	4.60	4.60	-	2.95	2.74	2.61	2.60	-
	2k / 32k	4.91	4.76	4.68	4.64	4.66	3.01	2.78	2.66	2.60	2.59

PoSE: 语言建模结果 (PPL on long-context tasks)



Step-Skipping Alignment (SkipAlign), Arxiv 2024

Model	Avg.	S-Doc QA	M-Doc QA	Summ	Few-shot	Code
GPT-3.5-Turbo-16k	44.6	39.7	38.7	26.5	67.0	54.2
LLAMA-2-7B Based Models						
LLAMA-2-7B-chat-4k	35.2	24.9	22.5	25.0	60.0	48.1
SEext-LLAMA-2-7B-chat-16k	38.7	27.3	26.2	24.8	64.2	57.5
LongChat1.5-7B-32k	36.9	28.7	20.6	26.6	60.0	54.2
LLAMA-2-7B-NTK32k	31.7	16.2	7.3	15.4	66.7	**63.4**
+ Normal-SFT	41.5	31.3	32.7	26.0	65.3	57.4
+ PackedSFT-16k	42.6	31.6	32.8	26.2	67.9	60.5
+ PackedSFT-32k	41.6	30.0	32.2	26.2	67.3	58.0
+ PackedSFT-50k	43.6	36.0	**37.0**	**27.7**	63.8	58.5
+ SkipAlign	**44.1**	**38.6**	33.8	26.1	**67.6**	59.6

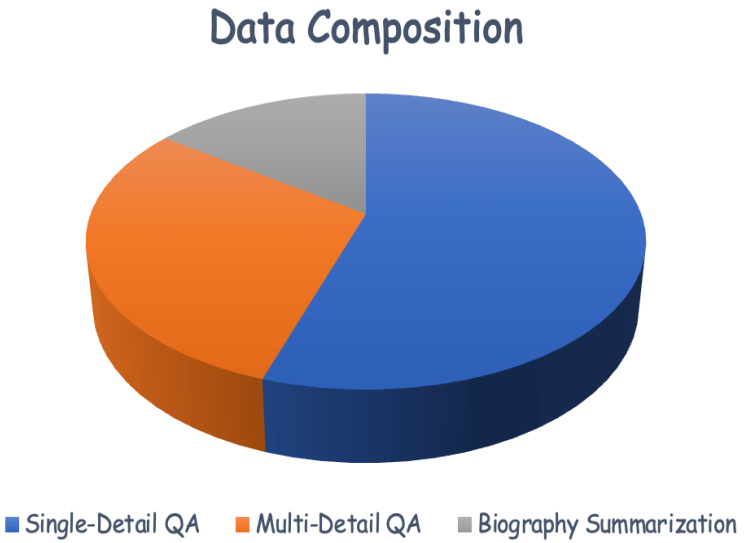
SkipAlign: 真实世界任务结果（LongBench）

Zhu D, Yang N, Wang L, et al. Pose: Efficient context window extension of llms via positional skip-wise training[J]. arXiv preprint arXiv:2309.10400, 2023.

Wu W, Wang Y, Fu Y, et al. Long context alignment with short instructions and synthesized positions [J]. arXiv preprint arXiv:2405.03939, 2024.

构造策略 • 模型生成

单文档问答	提示GPT-4回答长文本上下文中的一个特定细节问题
多文档问答	提示GPT-4在长文本上下文中汇总多个细节并做出推理
传记总结	提示GPT-4为给定书籍中的每个主要角色撰写传记



Model	Single-Doc	Multi-Doc	Summ.	Few-Shot	Synthetic	Code	Avg
Llama-3-8B-Instruct	37.33	36.04	26.83	**69.56**	37.75	53.24	43.20
Llama-3-8B-Instruct-262K	37.29	31.20	26.18	67.25	44.25	**62.71**	43.73
Llama-3-8B-Instruct-80K-QLoRA	**43.57**	**43.07**	**28.93**	69.15	**48.50**	51.95	**47.19**

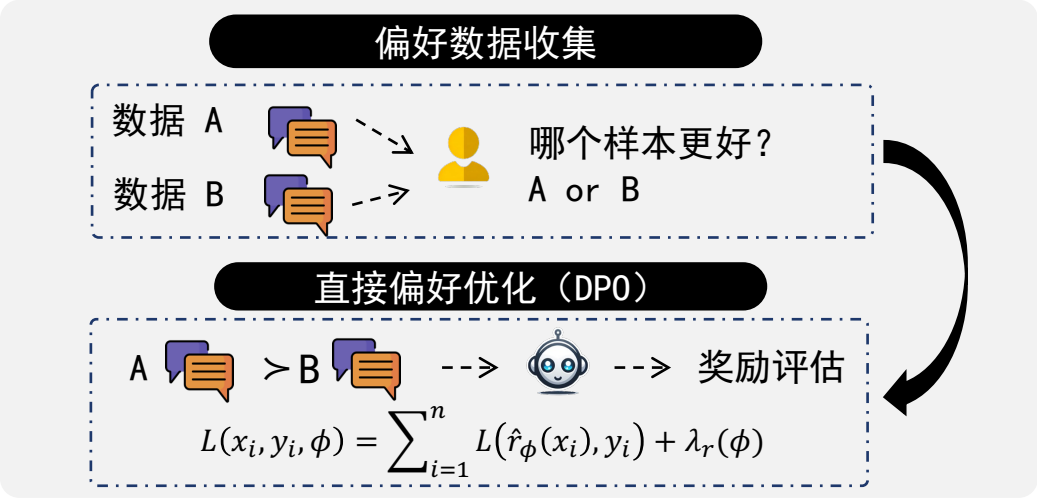
使用由GPT-4构建的合成数据可以大大提高长文本模型的性能。

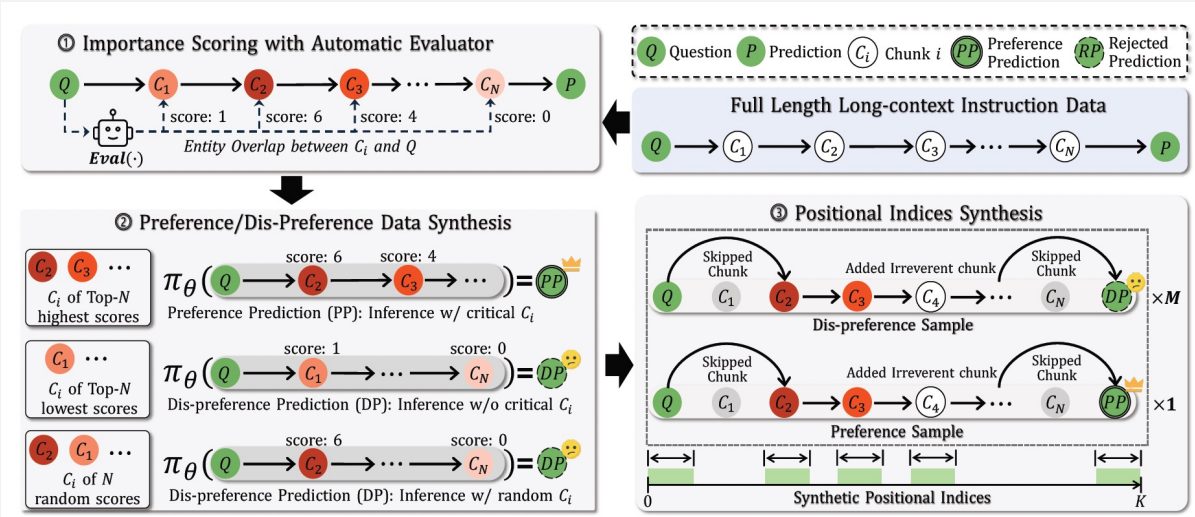
Zhang P, Shao N, Liu Z, et al. Extending Llama-3's Context Ten-Fold Overnight[J]. arXiv preprint arXiv:2404.19553, 2024.

数据类型 • 构建强化学习数据集

- **缺乏真实标签：**
  - 模型生成成本高
  - 人工标注难度大
- **需要构造正负样本**

- 第一步，基于短文本创建高质量标签
- 第二步，使用上下文扩展方法扩展短文本，如使用相关上下文、位置索引合成

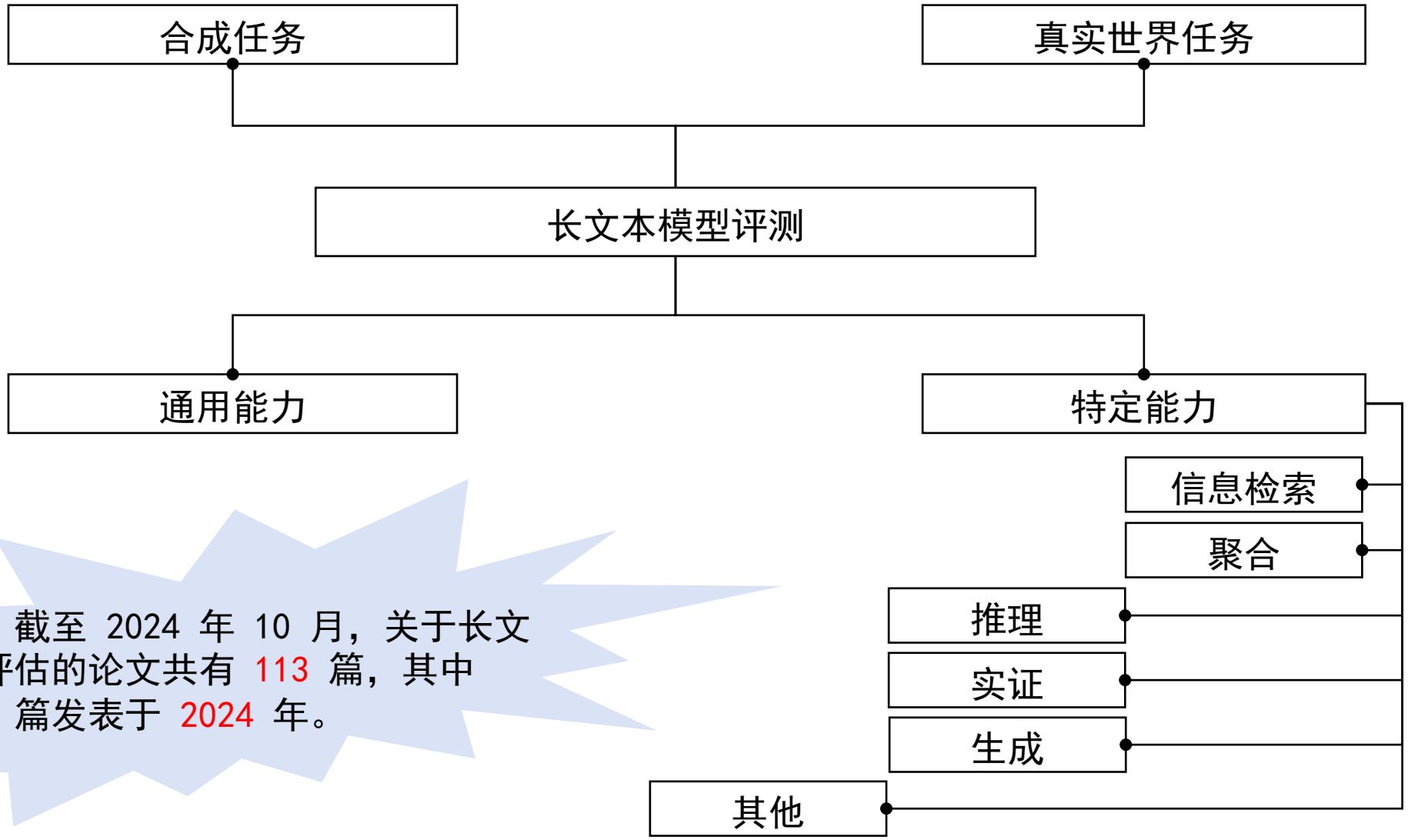




TL;DR 利用上下文中的关键片段来指导短文本模型生成合成数据

Tang Z, Sun Z, Li J, et al. LOGO--Long cOntext aliGnment via efficient preference Optimization[J]. arXiv preprint arXiv:2410.18533, 2024.

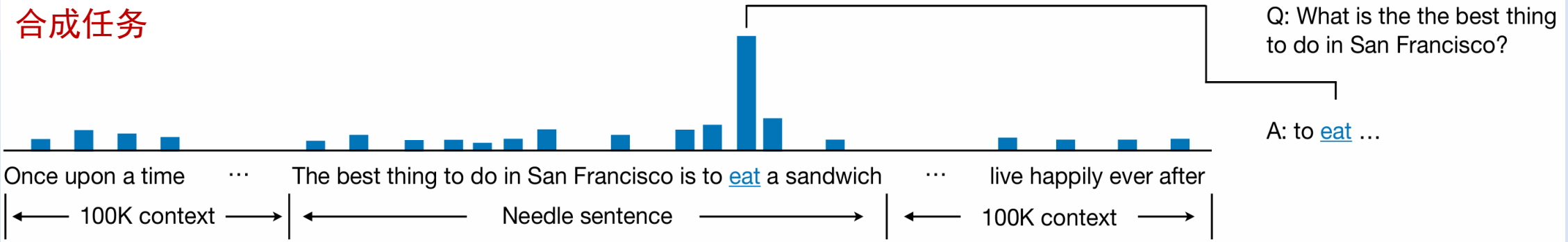
长上下文模型评测



截至 2024 年 10 月，关于长文本评估的论文共有 113 篇，其中 101 篇发表于 2024 年。

长上下文模型评测

合成任务



- **合成任务的优点：**
  - 更加可控
  - 减轻长文本模型固有知识的影响
- **合成任务的一般格式：**
  - 长冗余的上下文
  - 向长上下文注入关键信息

现实世界任务

Closed - Ended Tasks						
TOEFL	Multiple choice	English test	3,907	4,171	269	15
GSM(16-shot)†	Solving math problems	In-context examples	5,557	5,638	100	100
QuALITY †	Multiple choice	Gutenberg	7,169	8,560	202	15
Coursera*	Multiple choice	Advanced courses	9,075	17,185	172	15
TopicRet†	Retriving topics	Conversation	12,506	15,916	150	50
SFcition*	True or False Questions	Scientific fictions	16,381	26,918	64	7
CodeU*	Deducing program outputs	Python Codebase	31,575	36,509	90	90

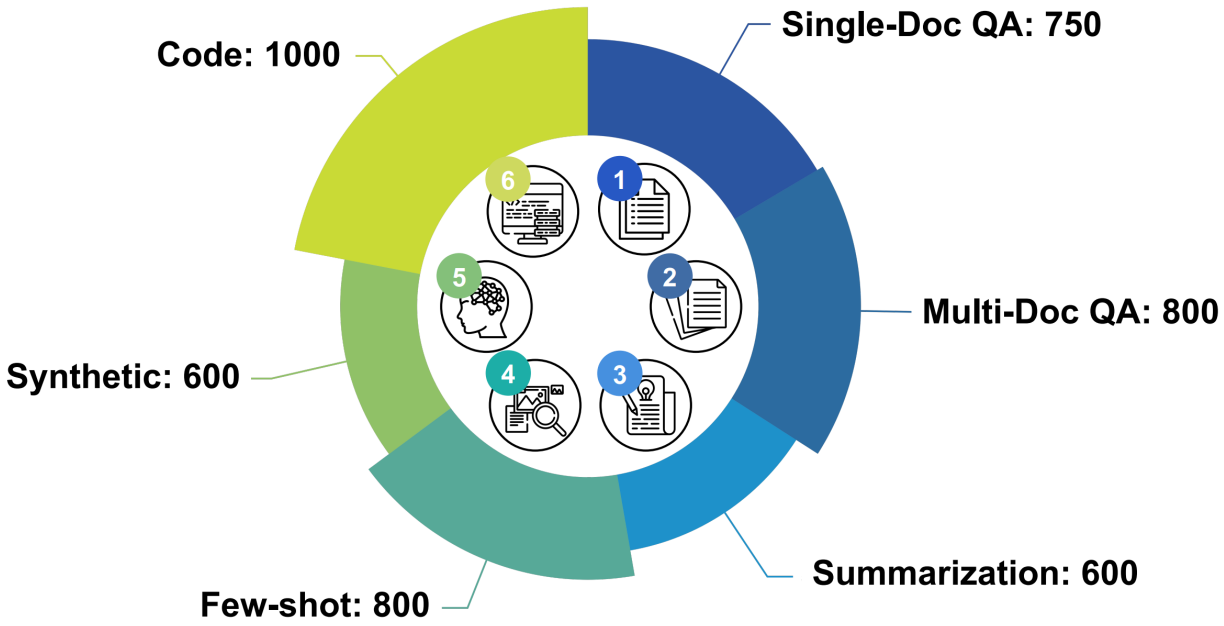
论文：

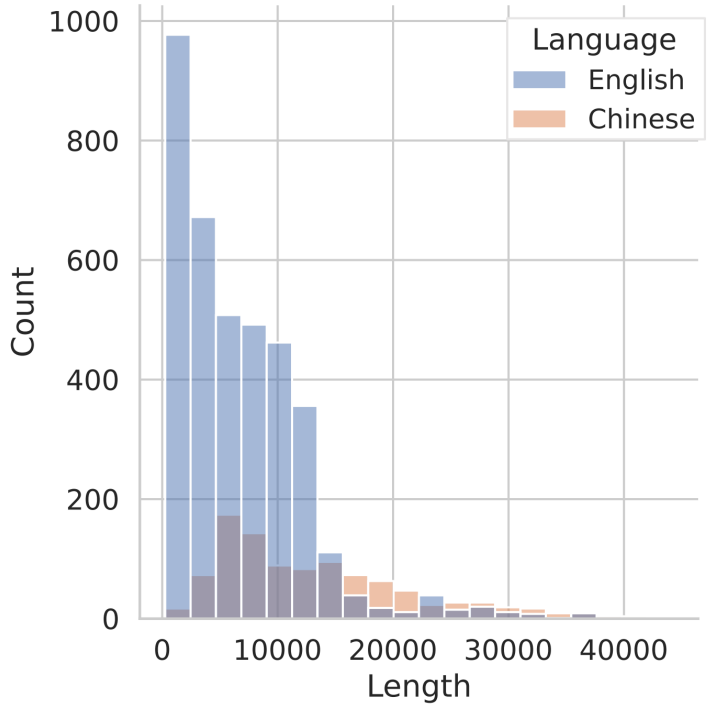
L-Eval: Instituting Standardized Evaluation for Long Context Language Models (ACL 2024)

An C, Gong S, Zhong M, et al. L-eval: Instituting standardized evaluation for long context language models[J]. arXiv preprint arXiv:2307.11088, 2023.

长上下文模型评测 • 通用能力

- LongBench（ACL 2024）是首个在长文本理解领域使用加入双语任务、多任务的基准测试（包括真实世界任务和合成任务）。

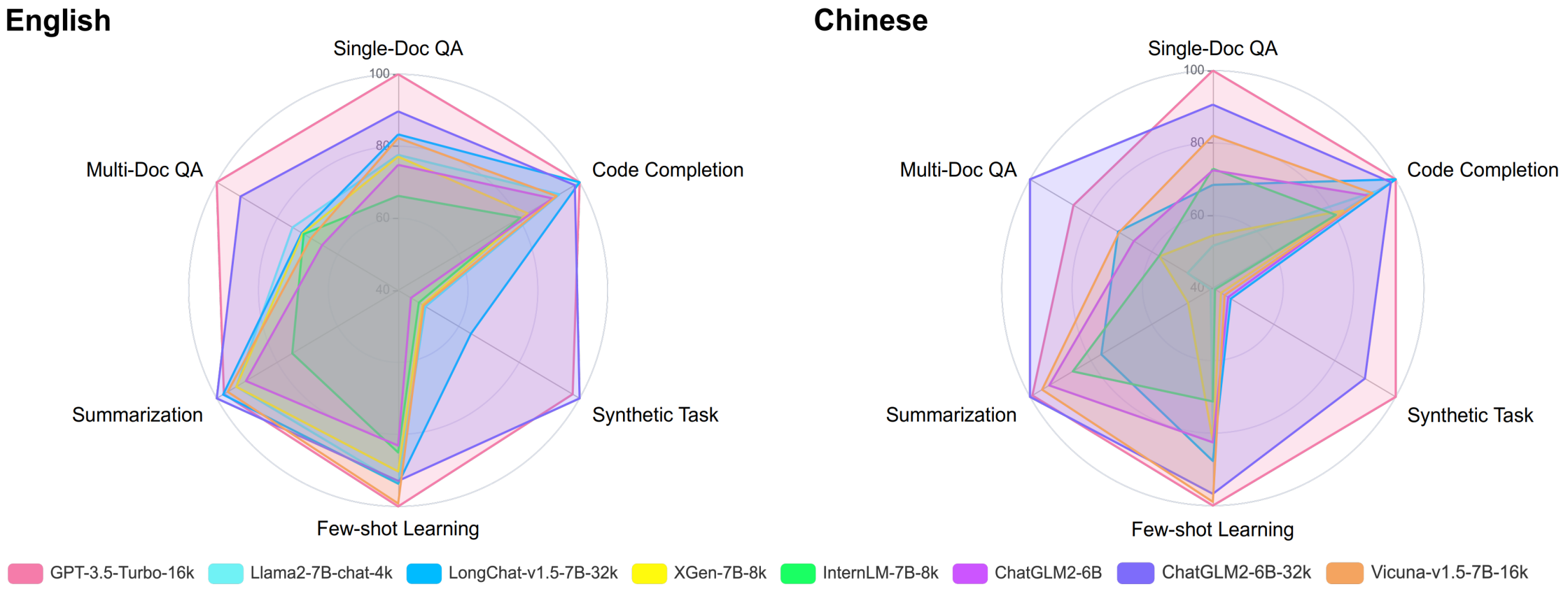




Bai Y, Lv X, Zhang J, et al. Longbench: A bilingual, multitask benchmark for long context understanding[J]. arXiv preprint arXiv:2308.14508, 2023.

长上下文模型评测 • 通用能力

- 开源模型与闭源模型（GPT-3.5-Turbo）存在明显差距（闭源模型能力更全面）
- 通过在更长文本上进行训练或者使用扩展的“位置嵌入”可以提高模型性能，例如 ChatGLM2-6B-32K 和 LongChat-v1.5-7B-32K 分别获得了 62% 和 19% 的相对提升



长上下文模型评测 • 通用能力

RULER: What's the Real Context Size of Your Long-Context Language Models? (COLM 2024)

Task	Configuration	Example
Single NIAH (S-NIAH)	type_key = word type_value = number type_haystack = essay size_haystack ∝ context length	(essays) ..... One of the special magic numbers for long-context is: 12345. ..... What is the special magic number for long-context mentioned in the provided text? Answer: 12345
Multi-keys NIAH (MK-NIAH)	num_keys = 2 type_key = word type_value = number type_haystack = essay size_haystack ∝ context length	(essays) ..... One of the special magic numbers for long-context is: 12345. One of the special magic numbers for large-model is: 54321. ..... What is the special magic number for long-context mentioned in the provided text? Answer: 12345
Multi-values NIAH (MV-NIAH)	num_values = 2 type_key = word type_value = number type_haystack = essay size_haystack ∝ context length	(essays) ..... One of the special magic numbers for long-context is: 12345. One of the special magic numbers for long-context is: 54321. ..... What are all the special magic numbers for long-context mentioned in the provided text? Answer: 12345 54321
Multi-queries NIAH (MQ-NIAH)	num_queries = 2 type_key = word type_value = number type_haystack = essay size_haystack ∝ context length	(essays) ..... One of the special magic numbers for long-context is: 12345. One of the special magic numbers for large-model is: 54321. ..... What are all the special magic numbers for long-context and large-model mentioned in the provided text? Answer: 12345 54321
Variable Tracking (VT)	num_chains = 2 num_hops = 2 size_noises ∝ context length	(noises) ..... VAR X1 = 12345 ..... VAR Y1 = 54321 ..... VAR X2 = X1 ..... VAR Y2 = Y1 ..... VAR X3 = X2 ..... VAR Y3 = Y2 ..... Find all variables that are assigned the value 12345. Answer: X1 X2 X3
Common Words Extraction (CWE)	freq_cw = 2, freq_ucw = 1 num_cw = 10 num_ucw ∝ context length	aaa bbb ccc aaa ddd eee ccc fff ggg hhh iii iii ..... What are the 10 most common words in the above list? Answer: aaa ccc iii .....
Frequent Words Extraction (FWE)	α = 2 num_word ∝ context length	aaa bbb ccc aaa ddd eee ccc fff ggg aaa hhh aaa ccc iii iii ..... What are the 3 most frequently appeared words in the above coded text? Answer: aaa ccc iii
Question Answering (QA)	dataset = SQuAD num_document ∝ context length	Document 1: ..... aaa ..... Document 2: ..... bbb ..... Document 3: ..... ccc ..... Question: question Answer: bbb

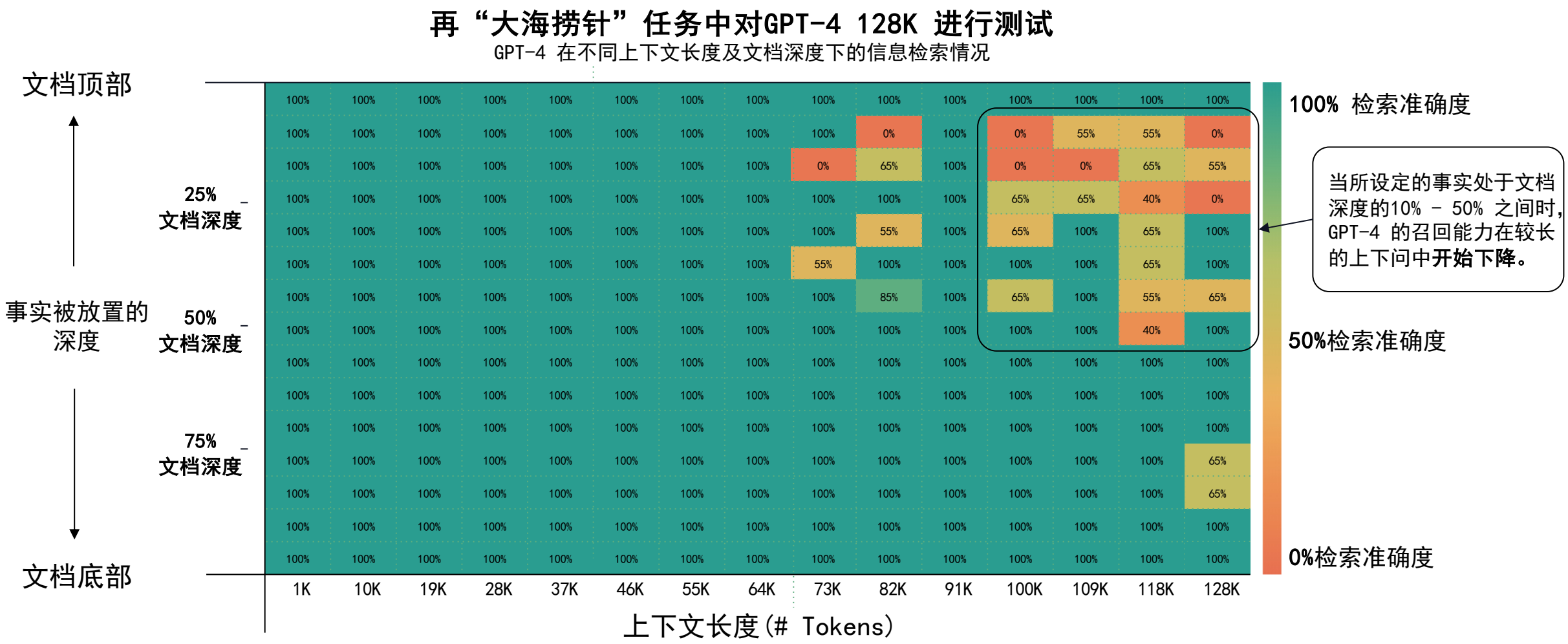
Hsieh C P, Sun S, Kriman S, et al. RULER: What's the Real Context Size of Your Long-Context Language Models?[J]. arXiv preprint arXiv:2404.06654, 2024.

长上下文模型评测 • 通用能力

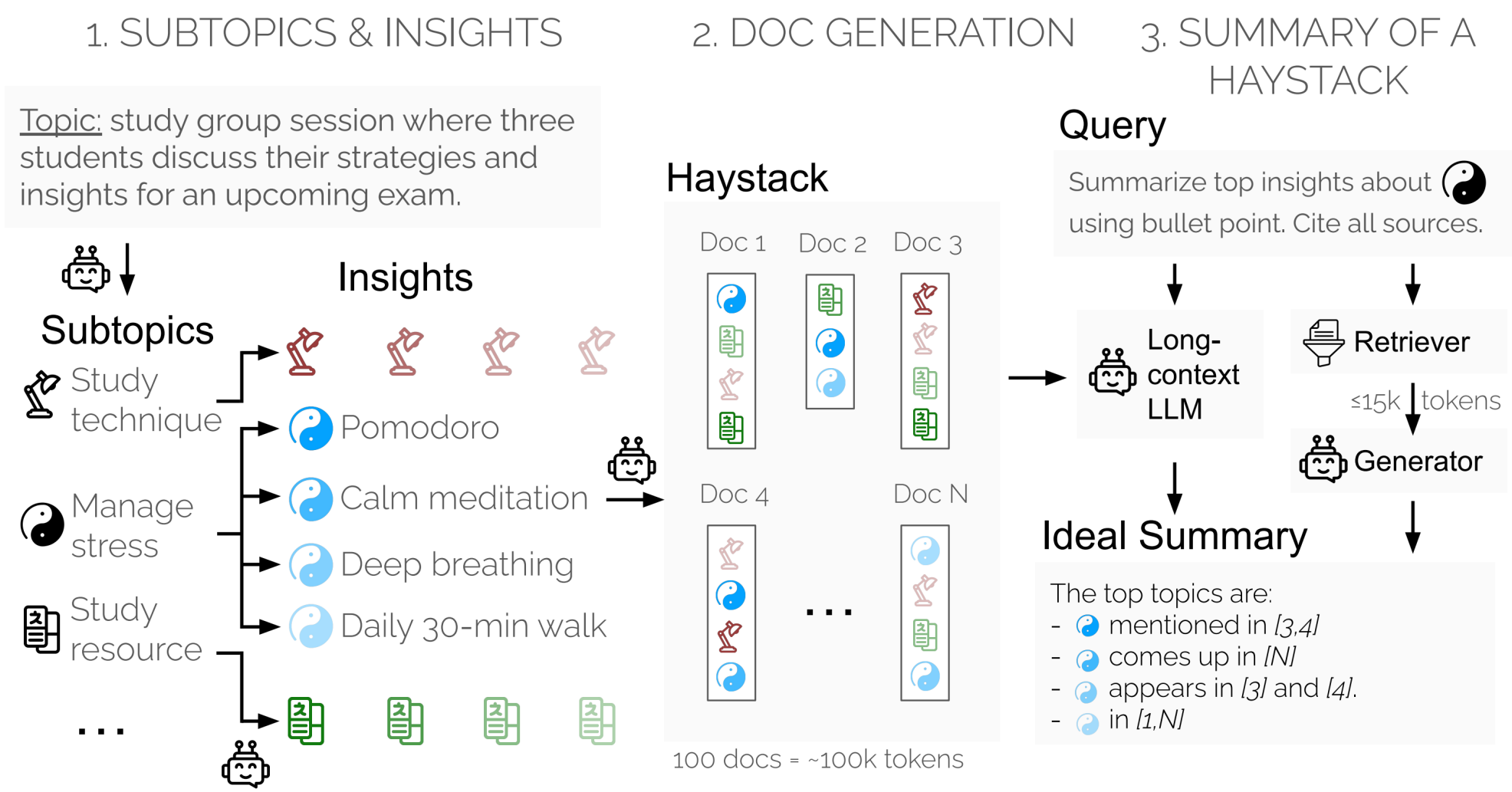
长文本模型能力排行榜

Models	Claimed Length	Effective Length	4K	8K	16K	32K	64K	128K	Avg.	wAvg. (inc)	wAvg. (dec)
闭源模型：											
Llama2 (7B)	4K	-	85.6								
Gemini-1.5-Pro	1M	>128K	96.7	95.8	96.0	95.9	95.9	94.4	95.8	95.5 (1st)	96.1 (1st)
GPT-4	128K	64K	96.6	96.3	95.2	93.2	87.0	81.2	91.6	89.0 (2nd)	94.1 (2nd)
Llama3.1 (70B)	128K	64K	96.5	95.8	95.4	94.8	88.4	66.6	89.6	85.5 (4th)	93.7 (3rd)
Qwen2 (72B)	128K	32K	96.9	96.1	94.9	94.1	79.8	53.7	85.9	79.6 (9th)	92.3 (4th)
Command-R-plus (104B)	128K	32K	95.6	95.2	94.2	92.0	84.3	63.1	87.4	82.7 (7th)	92.1 (5th)
GLM4 (9B)	1M	64K	94.7	92.8	92.1	89.9	86.7	83.1	89.9	88.0 (3rd)	91.7 (6th)
Llama3.1 (8B)	128K	32K	95.5	93.8	91.6	87.4	84.7	77.0	88.3	85.4 (5th)	91.3 (7th)
GradientAI/Llama3 (70B)	1M	16K	95.1	94.4	90.8	85.4	80.9	72.1	86.5	82.6 (8th)	90.3 (8th)
Mixtral-8x22B (39B/141B)	64K	32K	95.6	94.9	93.4	90.9	84.7	31.7	81.9	73.5 (11th)	90.3 (9th)
Yi (34B)	200K	32K	93.3	92.2	91.3	87.5	83.2	77.3	87.5	84.8 (6th)	90.1 (10th)
Phi3-medium (14B)	128K	32K	93.3	93.2	91.1	86.8	78.6	46.1	81.5	74.8 (10th)	88.3 (11th)
Mistral-v0.2 (7B)	32K	16K	93.6	91.2	87.2	75.4	49.0	13.8	68.4	55.6 (13th)	81.2 (12th)
LWM (7B)	1M	<4K	82.3	78.4	73.7	69.1	68.1	65.0	72.8	69.9 (12th)	75.7 (13th)
DBRX (36B/132B)	32K	8K	95.1	93.8	83.6	63.1	2.4	0.0	56.3	38.0 (14th)	74.7 (14th)
Together (7B)	32K	4K	88.2	81.1	69.4	63.0	0.0	0.0	50.3	33.8 (15th)	66.7 (15th)
LongChat (7B)	32K	<4K	84.7	79.9	70.8	59.3	0.0	0.0	49.1	33.1 (16th)	65.2 (16th)
LongAlpaca (13B)	32K	<4K	60.6	57.0	56.6	43.6	0.0	0.0	36.3	24.7 (17th)	47.9 (17th)

长上下文模型评测 • 信息检索能力



长上下文模型评测 • 聚合能力



Laban P, Fabbri A R, Xiong C, et al. Summary of a haystack: A challenge to long-context llms and rag systems[J]. arXiv preprint arXiv:2407.01370, 2024.

长上下文模型评测 • 推理能力

Long-Context Multi-evidence Acquisition *English Version*	November 2005In the next few years, venture capital funds will find themselves squeezed from four directions. They're already stuck with a seller's market, because of the huge amounts they raised at the end of the Bubble and still haven't invested. This by itself is not the end of the world. In fact, it's just a more extreme version of the norm in the VC business: too much money chasing too few deals.Unfortunately, those few deals now want less and less money, because it's getting so cheap to start a startup ... ***The little penguin counted {number1} ★*** ... Moore's law, which makes hardware geometrically closer to free; the Web, which makes promotion free if you're good; and better languages, which make development a lot cheaper.When we started our startup in 1995, the first three were our biggest expenses. We had to pay $5000 for the Netscape Commerce Server, the only software that then supported secure http connections ... ***The little penguin counted {number2} ★*** ... people throw away computers more powerful than our first server ... . ***On this moonlit and misty night, the little penguin is looking up at the sky and concentrating on counting ★. Please help the little penguin collect the number of ★, for example: "little_penguin": [x, x, x,...]. The summation is not required, and the numbers in [x, x, x,...] represent the counted number of ★ by the little penguin. Only output the results in JSON format without any explanation.***
Long-Context Multi-evidence Reasoning *English Version*	November 2005In the next few years, venture capital funds will find themselves squeezed from four directions. They're already stuck with a seller's market, because of the huge amounts they raised at the end of the Bubble and still haven't invested. This by itself is not the end of the world. In fact, it's just a more extreme version of the norm in the VC business: too much money chasing too few deals.Unfortunately, those few deals now want less and less money, because it's getting so cheap to start a startup ... ***The little penguin counted {wrong number1} ★, but found that a mistake had been made, so the counting was done again, and this time {number1} ★ was counted correctly.*** ... Moore's law, which makes hardware geometrically closer to free; the Web, which makes promotion free if you're good; and better languages, which make development a lot cheaper.When we started our startup in 1995, the first three were our biggest expenses. We had to pay $5000 for the Netscape Commerce Server, the only software that then supported secure http connections ... ***The little penguin counted {wrong number2} ★, but found that a mistake had been made, so the counting was done again, and this time {number2} ★ was counted correctly.*** ... people throw away computers more powerful than our first server .... ***On this moonlit and misty night, the little penguin is looking up at the sky and concentrating on counting ★. Please help the little penguin collect the correct number of ★, for example: "little_penguin": [x, x, x,...]. The summation is not required, and the numbers in [x, x, x,...] represent the correctly counted number of ★ by the little penguin. Only output the results in JSON format without any explanation.***

Gonen M, Ron D, Shavitt Y. Counting stars and other small subgraphs in sublinear-time[J]. SIAM Journal on Discrete Mathematics, 2011, 25(3): 1365-1411.

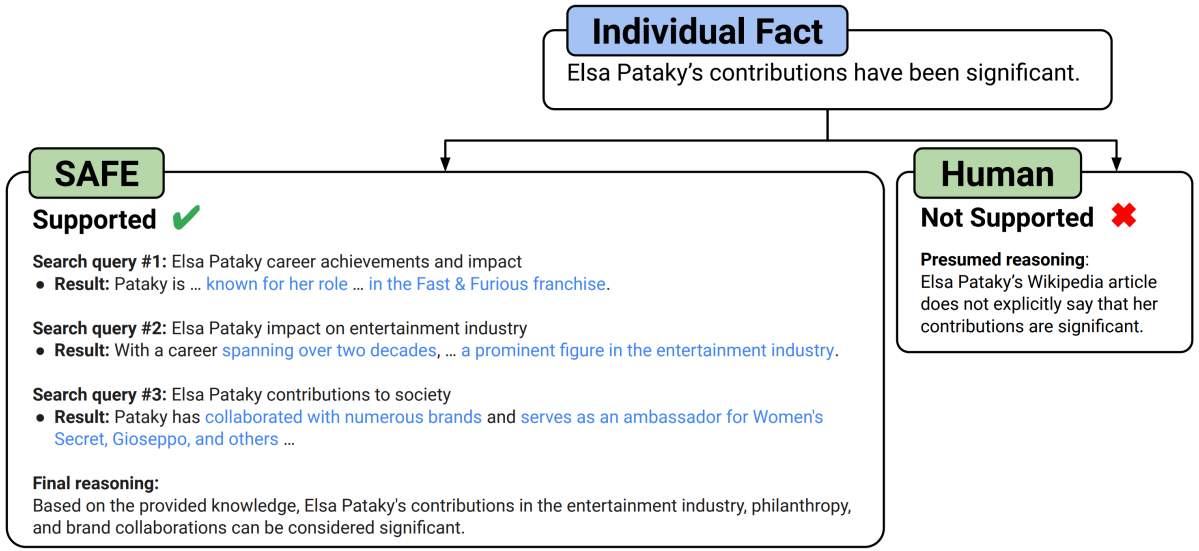
长上下文模型评测 • 推理能力

- 尽管长上下文大语言模型在‘大海捞针’任务上已接近完美表现，但它们在‘数星星’测试中仍表现不佳
  - 这说明‘大海捞针’无法真正展示大型语言模型在处理长序列推理任务上的能力

Models	GPT-4 Turbo		Gemini 1.5 Pro	Claude3			GLM-4	Moonshot-v1
	1106	*0125*		Opus	Sonnet	Haiku		
Multi-evidence Acquisition (ZH)	0.697	0.663	0.775	0.807	0.788	0.698	0.682	0.606
Multi-evidence Acquisition (EN)	0.718	0.662	0.833	0.705	-	-	0.389	0.559
Multi-evidence Reasoning (ZH)	0.473	0.386	0.575	0.488	-	-	0.475	0.344
Multi-evidence Reasoning (EN)	0.651	0.610	0.371	0.374	-	-	0.179	0.460
Average Score	0.635_2	0.580_4	0.639_1	0.594_3	-	-	0.431_6	0.492_5

长上下文模型评测 • 实证能力

- 与模型多轮对话，引导模型支撑某些观点，用模型输出结果与搜索结果进行比对检查正确性。



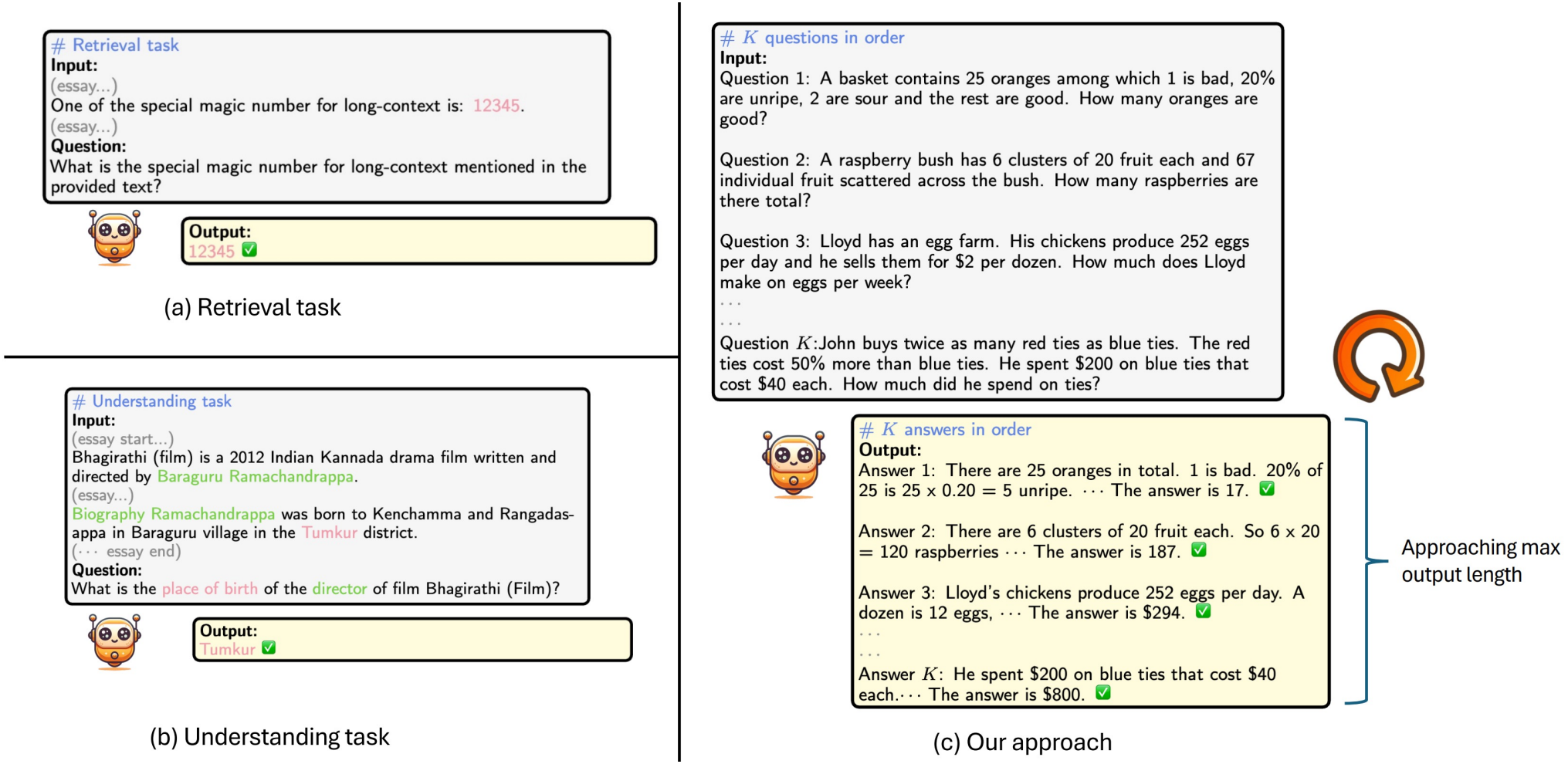
- 更大的模型在长上下文中的证实能力方面的表现更好

Model	Raw metrics S	NS	I	Aggregated metrics Prec	R_{64}	R_{178}	F_1@64	F_1@178
Gemini-Ultra	83.4	13.1	7.6	86.2	98.3	46.9	91.7	60.3
Gemini-Pro	66.6	12.1	5.5	82.0	88.5	37.4	83.7	50.4
GPT-4-Turbo	93.6	8.3	6.1	91.7	99.0	52.6	95.0	66.4
GPT-4	59.1	7.0	2.4	89.4	88.3	33.2	88.0	48.0
GPT-3.5-Turbo	46.9	4.5	1.2	90.8	72.4	26.4	79.6	40.5
Claude-3-Opus	63.8	8.1	2.8	88.5	91.4	35.9	89.3	50.6
Claude-3-Sonnet	65.4	8.4	2.6	88.5	91.4	36.7	89.4	51.4
Claude-3-Haiku	39.8	2.8	1.3	92.8	62.0	22.4	73.5	35.8
Claude-2.1	38.3	6.5	1.9	84.8	59.8	21.5	67.9	33.7
Claude-2.0	38.5	6.3	1.3	85.5	60.0	21.6	68.7	34.0
Claude-Instant	43.9	8.1	1.3	84.4	68.4	24.6	73.8	37.6
PaLM-2-L-IT-RLHF	72.9	8.7	4.3	89.1	94.0	41.0	91.0	55.3
PaLM-2-L-IT	13.2	1.8	0.1	88.8	20.6	7.4	31.1	13.2

Wei J, Yang C, Song X, et al. Long-form factuality in large language models[J]. arXiv preprint arXiv:2403.18802, 2024.

长上下文模型评测 • 生成能力

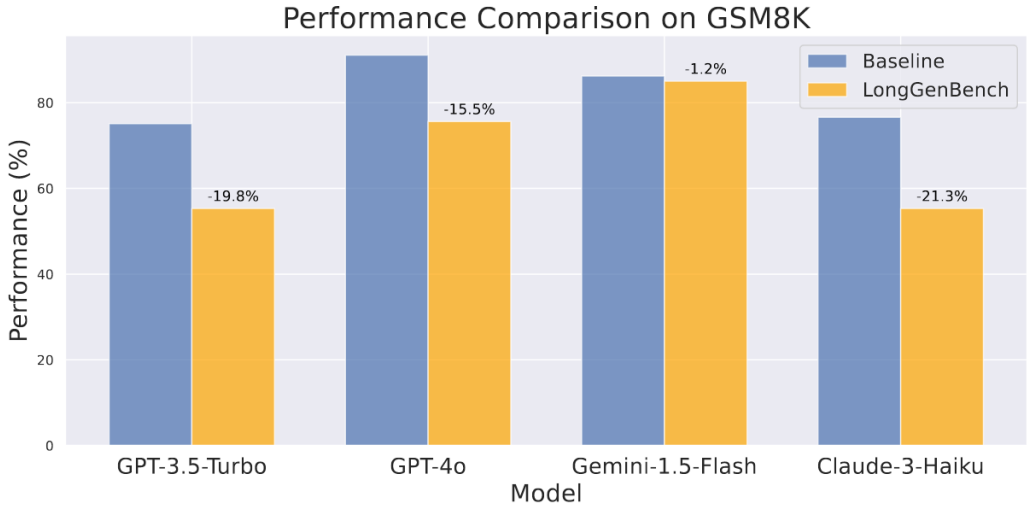
- LongGenBench 要求LLMs逐步理解并在单个响应中回答每个问题。

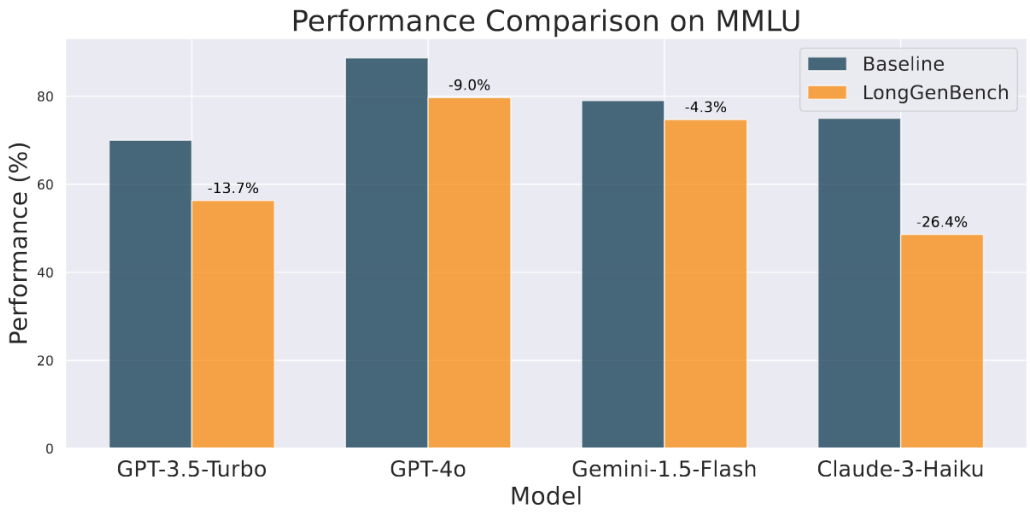


Liu X, Dong P, Hu X, et al. Longgenbench: Long-context generation benchmark[J]. arXiv preprint arXiv:2410.04199, 2024.

长上下文模型评测 • 生成能力

- 主流LLMs在执行长上下文生成任务时会出现性能下降的情况。





- LongGenBench 的主要挑战在于生成长文本，而不是理解长输入。

Model	GSM8K (%) Baseline↑	GSM8K (%) LongGenBench↑	GSM8K (%) Delta Δ
LLaMA-3-8B-Instruct	79.6	32.5	-47.1∇
LLaMA-3-70B-Instruct	93.0	83.2	-9.8∇
Qwen2-7B-Instruct	82.3	63.9	-18.4∇
Qwen2-57B-A14B-Instruct	79.6	71.2	-8.4∇
Qwen2-72B-Instruct	91.1	85.7	-5.4∇
ChatGLM4-9B-Chat	79.6	68.8	-10.8∇
DeepSeek-v2-Chat	92.2	86.5	-5.7∇

(a) Performance on GSM8K dataset

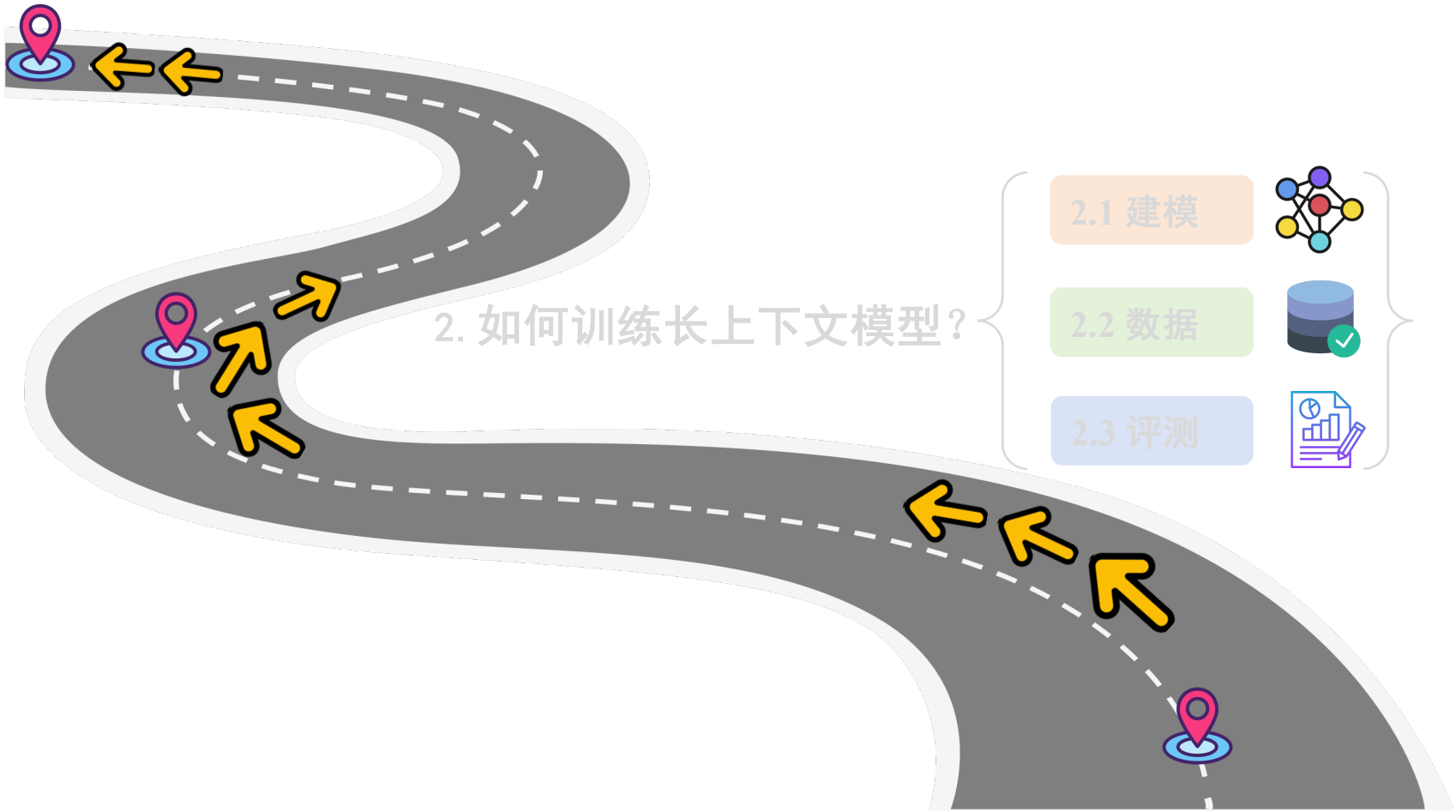
Model	MMLU (%) Baseline↑	MMLU (%) LongGenBench↑	MMLU (%) Delta Δ
LLaMA-3-8B-Instruct	68.4	50.4	-18.0∇
LLaMA-3-70B-Instruct	82.0	71.2	-10.8∇
Qwen2-7B-Instruct	70.5	59.4	-11.1∇
Qwen2-57B-A14B-Instruct	75.4	66.7	-8.7∇
Qwen2-72B-Instruct	82.3	75.8	-6.5∇
ChatGLM4-9B-Chat	72.4	63.0	-9.4∇
DeepSeek-v2-Chat	77.8	72.0	-5.8∇

(b) Performance on MMLU dataset

Model	Long Input + Short Output	Long Input + Long Output	Performance Drop
GPT-3.5-Turbo	74.3	55.3	-19.0
Gemini-1.5-Flash	86.1	85.0	-1.1

报告内容

3. 长上下文大模型前沿与挑战



2. 如何训练长上下文模型？

2.1 建模

2.2 数据

2.3 评测

1. 什么是长上下文模型？

前沿工作 • 长上下文模型能力对齐

LOGO -- Long cOntext aliGnment via efficient preference Optimization (Arxiv 2024)

公众号：https://mp.weixin.qq.com/s/5Xg6rGQgJ8OgBdIL3FtfNg

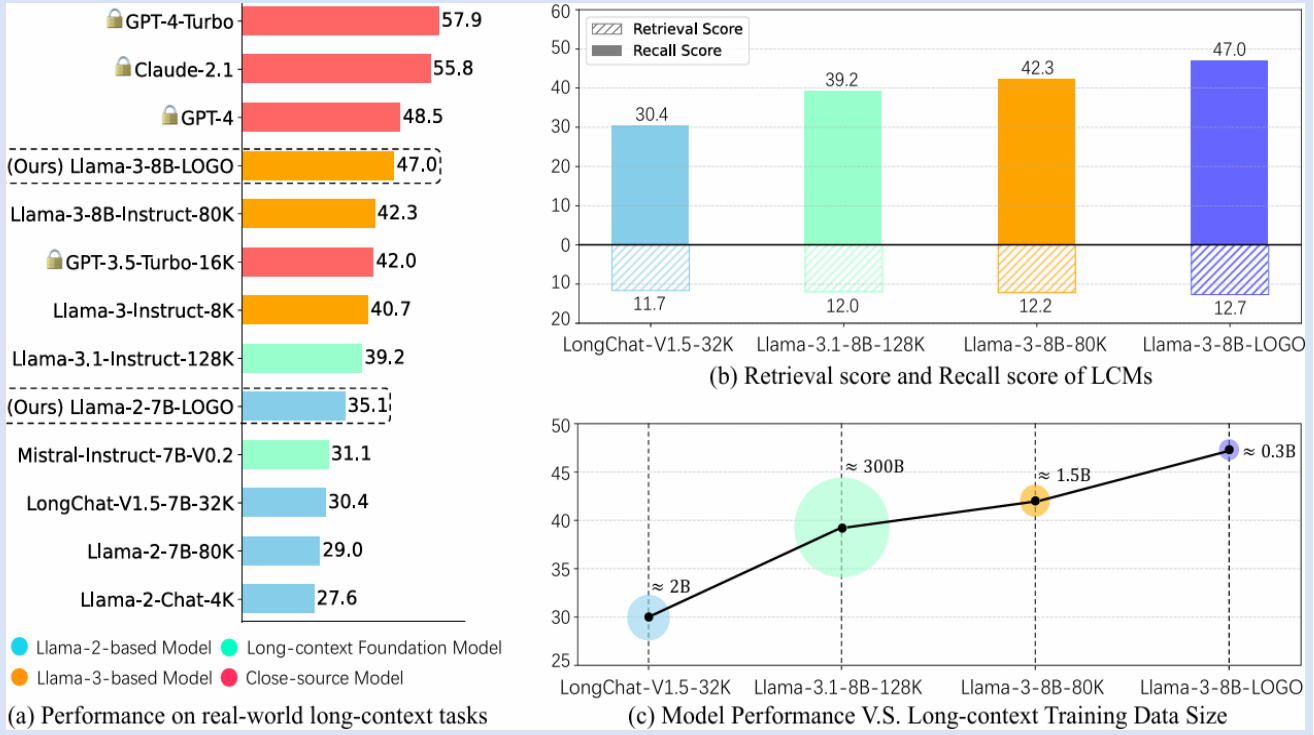
项目：https://github.com/ZetangForward/LCM_Stack

- **难点：缺乏有效且高效的长上下文能力对齐方法**
  - 缺乏足够数量的高质量长上下文训练数据
  - 缺乏高效的长上下文训练策略和训练框架

- **高效长上下文偏好优化策略**

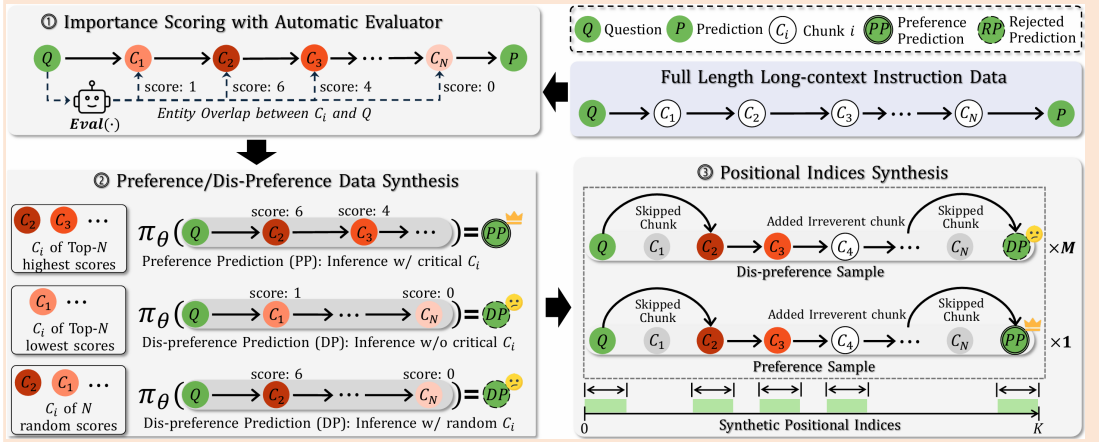
$$\mathcal{L}_{\text{LOGO}}(\pi_\theta) = -\mathbb{E}_{(x,y_w,y_l^{(1\cdots M)})}\left[\log\sigma\left(\frac{\beta}{|y_w|}\log\pi_\theta(y_w|x) - \frac{\beta}{M|y_l|}\sum_{j=1}^{M}\log\pi_\theta(y_l^{(j)}|x) - \gamma\right)\right]$$

$$\mathcal{L}^*_{\text{LOGO}}(\pi_\theta) = \mathcal{L}_{\text{LOGO}}(\pi_\theta) + \lambda\mathbb{E}_{(x,y_w)}\log\pi_\theta(y_w|x))$$



长上下文偏好数据集构建

- 从短上下文构建长偏好数据
  - 利用文本中的关键片段构造偏好数据
- 利用位置编码合成模拟长上下文输入



前沿工作 • 长上下文模型能力对齐

LOGO -- Long cOntext aliGnment via efficient preference Optimization (Arxiv 2024)

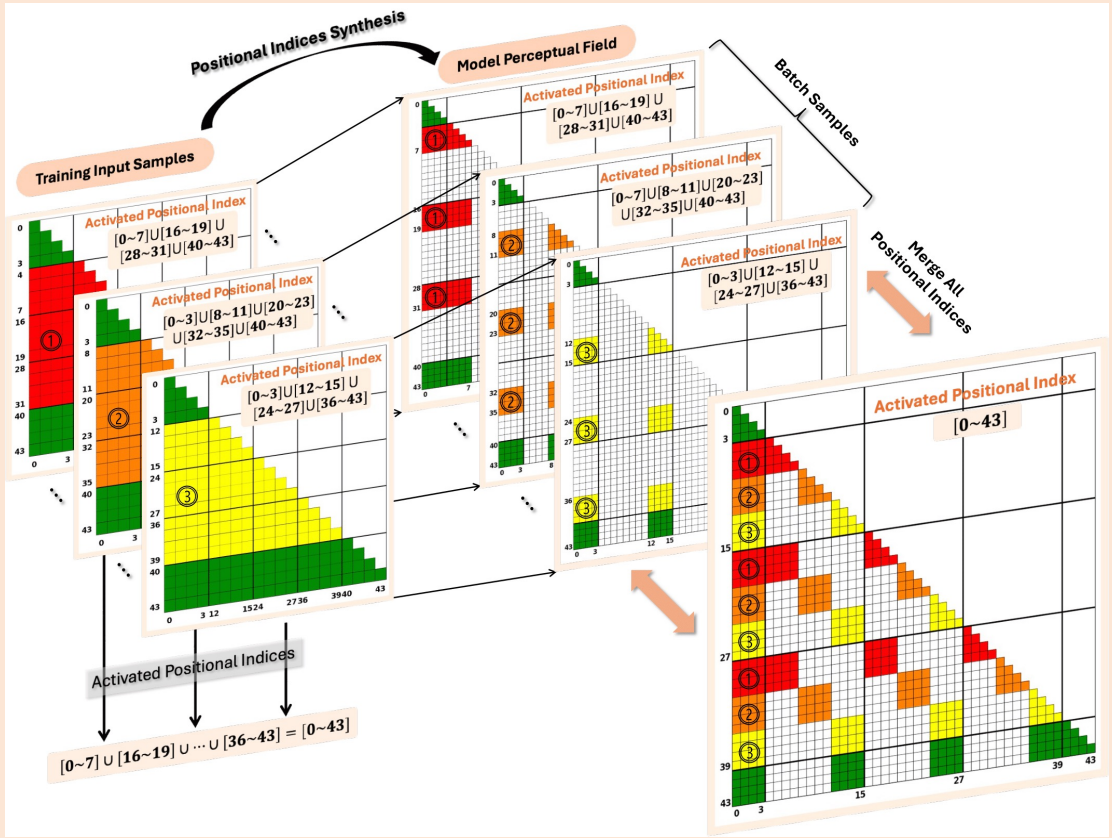
公众号：https://mp.weixin.qq.com/s/5Xg6rGQgJ8OgBdIL3FtfNg

项目：https://github.com/ZetangForward/LCM_Stack

◘ 位置编码合成需要覆盖所有的位置信息（防止遗漏关键信息）

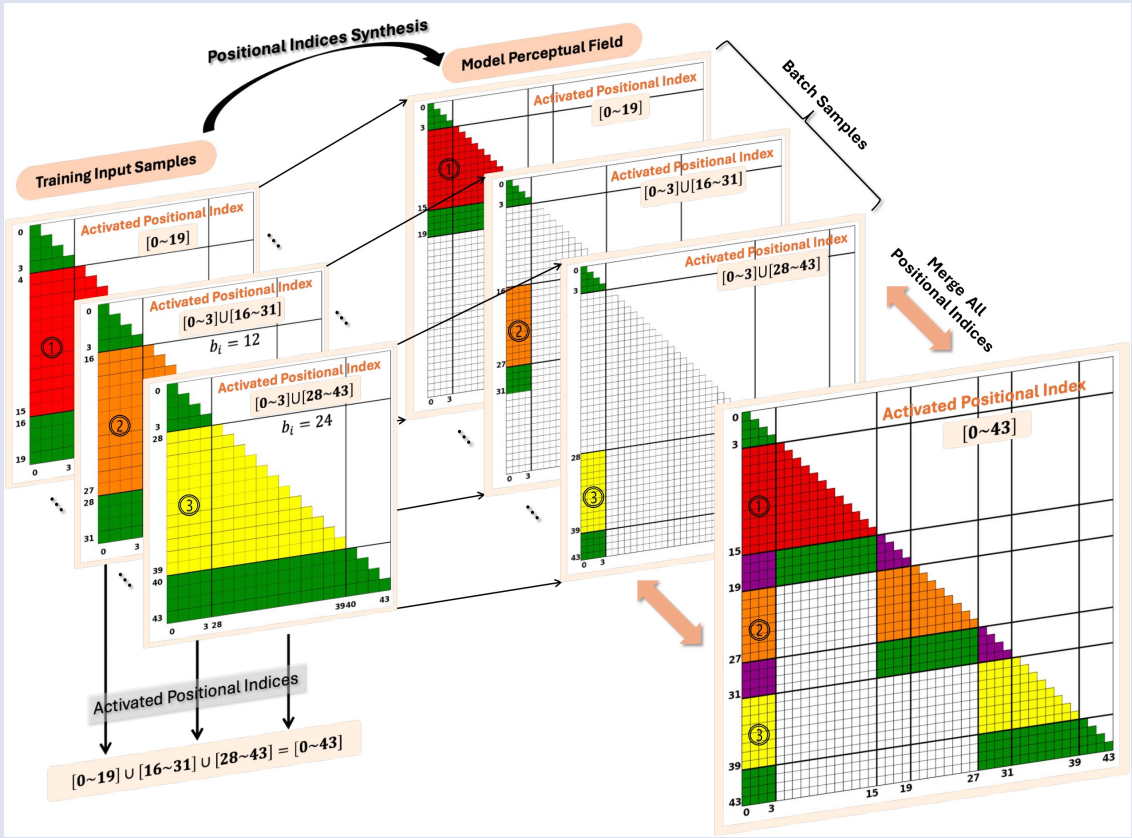
离散块位置编码合成

❖ 有效扩展上下文长度



连续块位置编码合成

❖ 保证连续块内语义不丢失



前沿工作 • 长上下文模型能力对齐

LOGO -- Long cOntext aliGnment via efficient preference Optimization (Arxiv 2024)

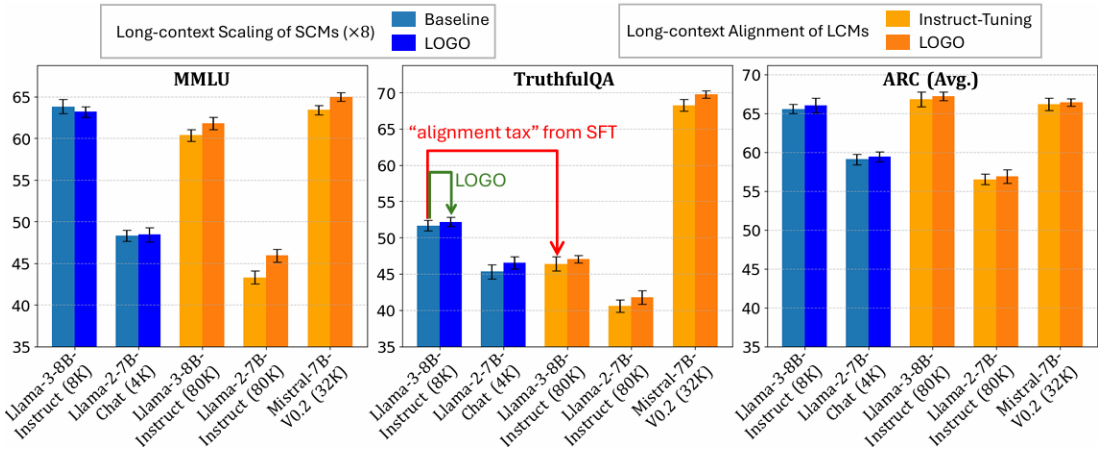
公众号：https://mp.weixin.qq.com/s/5Xg6rGQgJ8OgBdIL3FtfNg

项目：https://github.com/ZetangForward/LCM_Stack

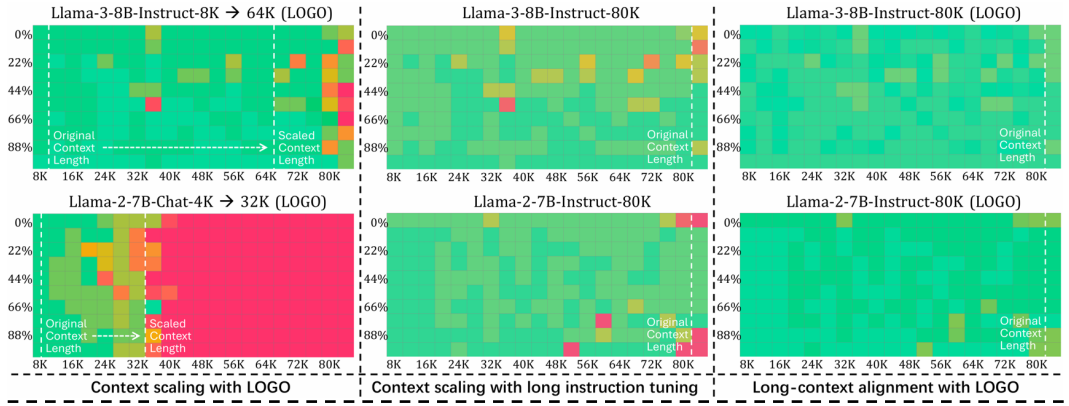
- 主试验结果
  - LOGO在短上下文模型上测试：有效扩展上下文长度
  - LOGO在长上下文模型上测试：有效增强模型能力

Models	S-Doc QA	M-Doc QA	Summ	Few-shot	Synthetic	Avg.
GPT-3.5-Turbo-16K	39.8	38.7	26.5	67.1	37.8	42.0
LongChat-v1.5-7B-32k	28.7	20.6	26.7	60.0	15.8	30.4
LLama-3.1-8B-Instruct-128K	23.9	15.8	28.9	69.8	57.5	39.2
Results on SCMs (scaling ×8 context window)						
Llama-3-8B-Instruct-8K	39.3	36.2	24.8	63.5	39.9	40.7
+ YaRN-64K†	38.0	36.6	27.4	61.7	40.9	40.9
+ RandPOS-64K	32.5	30.5	26.5	61.3	33.4	36.8
+ LOGO-64K	**39.8**	**36.7**	**28.8**	**65.4**	**49.0**	**43.9**
Llama-2-7B-Chat-4K	24.9	22.6	24.7	60.0	5.9	27.6
+ LOGO-32K	**26.7**	**23.3**	**26.3**	**63.1**	**11.1**	**30.1**
Results on LCMs (long-context alignment)						
Llama-3-8B-Instruct-80K	43.0	39.8	22.2	64.3	46.3	42.3
+ Instruct Tuning (Full)	38.8	35.0	24.6	65.9	44.5	41.8
+ Instruct Tuning (Partial)	39.3	36.2	26.8	63.5	48.0	42.8
+ LOGO-80K	**44.0**	**41.2**	**28.1**	**68.6**	**53.0**	**47.0**
Llama-2-7B-Instruct-80K	26.9	23.8	21.3	65.0	7.9	29.0
+ LOGO-80K	**33.6**	**28.0**	**29.4**	**65.1**	**24.5**	**36.1**
Mistral-Instruct-7B-V0.2-32K	31.7	30.6	16.7	58.4	17.9	31.1
+ LOGO-32K	**38.3**	**37.6**	**26.1**	**67.0**	**31.5**	**40.1**

- LOGO 可以在短文本任务上保持原有效果，避免长上下文对齐过程中遗忘预训练阶段获得的知识。



- LOGO可以有效扩充上下文长度，NIAH测试有效位置全绿。



前沿工作 • 新模型架构

Hugging Face tiiuae/falcon-mamba-7b

model name	IFEval	BBH	MATH LvL5	GPQA	MUSR	MMLU-PRO	Average
Pure SSM models							
FalconMamba-7B	33.36	19.88	3.63	8.05	10.86	14.47	**15.04**
TRI-ML/mamba-7b-rw*	22.46	6.71	0.45	1.12	5.51	1.69	6.25
Hybrid SSM-attention models							
recurrentgemma-9b	30.76	14.80	4.83	4.70	6.60	17.88	13.20
Zyphra/Zamba-7B-v1*	24.06	21.12	3.32	3.03	7.74	16.02	12.55
Transformer models							
Falcon2-11B	32.61	21.94	2.34	2.80	7.53	15.44	13.78
Meta-Llama-3-8B	14.55	24.50	3.25	7.38	6.24	24.55	13.41
Meta-Llama-3.1-8B	12.70	25.29	4.61	6.15	8.98	24.95	13.78
Mistral-7B-v0.1	23.86	22.02	2.49	5.59	10.68	22.36	14.50
Mistral-Nemo-Base-2407 (12B)	16.83	29.37	4.98	5.82	6.52	27.46	15.08
gemma-7B	26.59	21.12	6.42	4.92	10.98	21.64	**15.28**
RWKV models							
RWKV-v6-Finch-7B*	27.65	9.04	1.11	2.81	2.25	5.85	8.12
RWKV-v6-Finch-14B*	29.81	12.89	1.13	5.01	3.16	11.3	10.55

SoTA RNN-based 模型

Transformer-based 模型表现稳定

SoTA Transformer-based 模型

难点：对新模型架构的探索不够

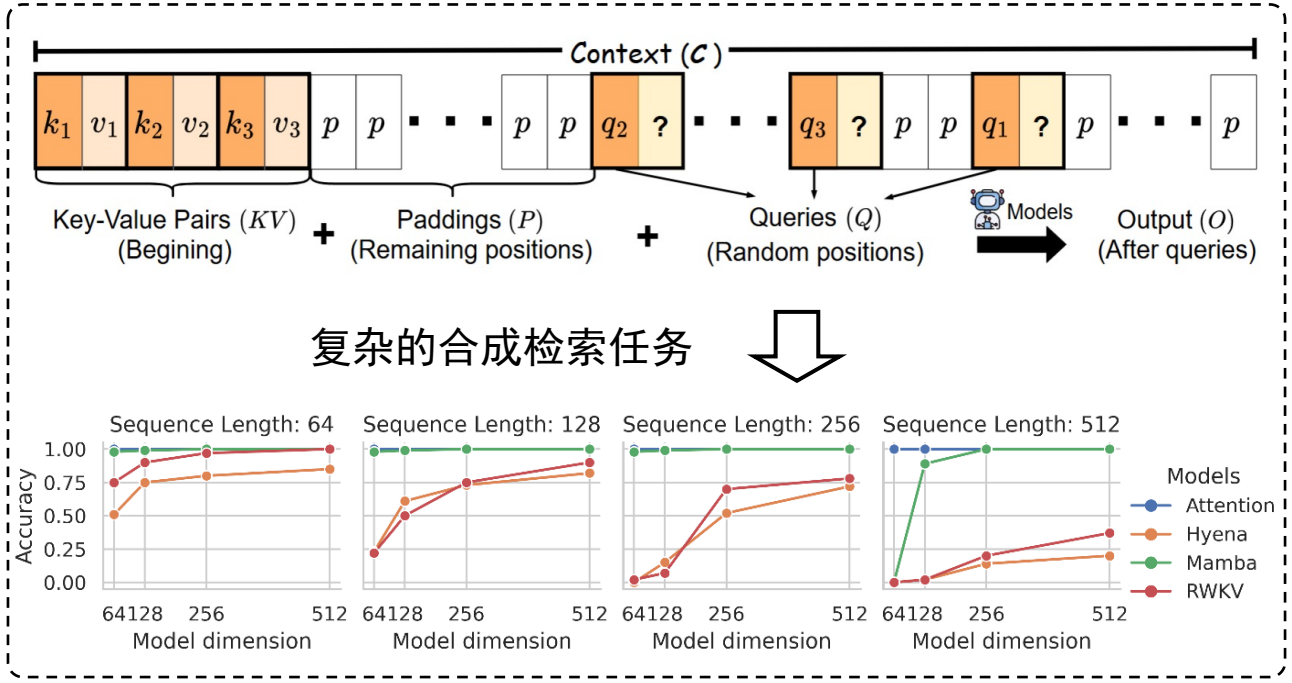
- 训练稳定性差，会频繁出现数值溢出，损失崩溃的现象
- 效率优化往往伴随着性能上的损失
- 难以Scaling到大参数量的模型

前沿工作 • 新模型架构

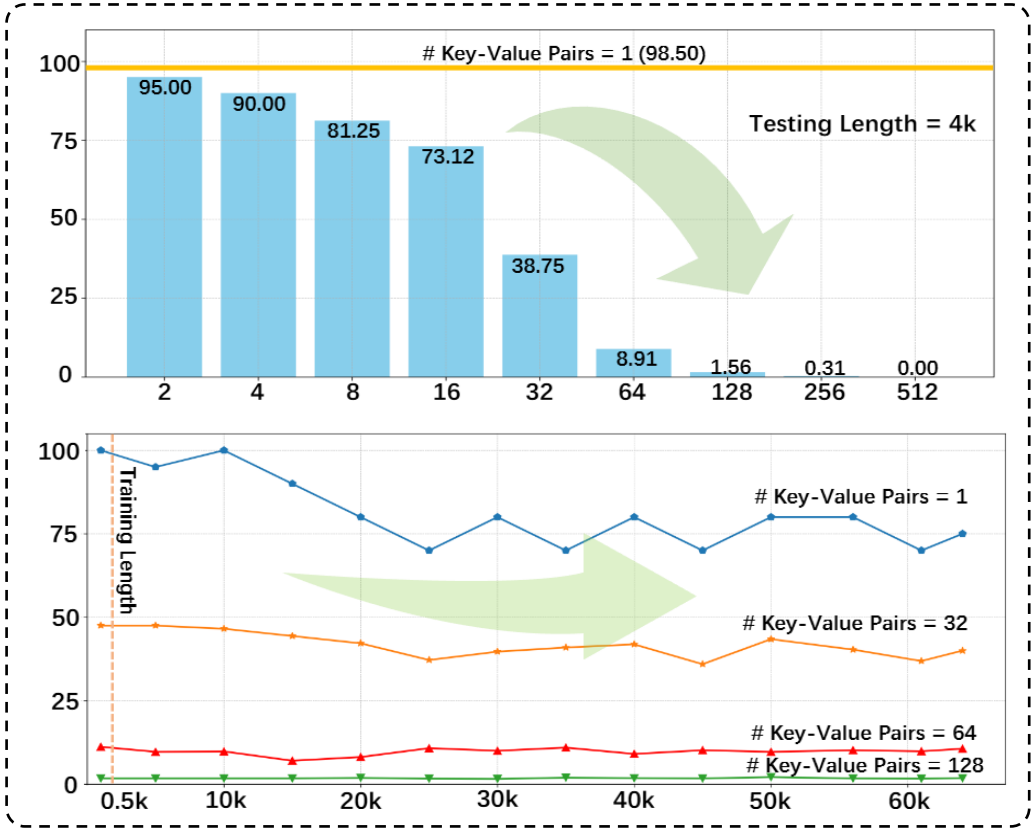
Revealing and Mitigating the Local Pattern Shortcuts of Mamba (Arxiv 2024)

项目：https://github.com/ZetangForward/Global_Mamba

- Mamba 可以超过其他RNN-based模型，但是距离Attention-based模型还有很大的一段差距。



- Mamba 在序列长度上有很好的泛化性，但是在信息密度上难以泛化。

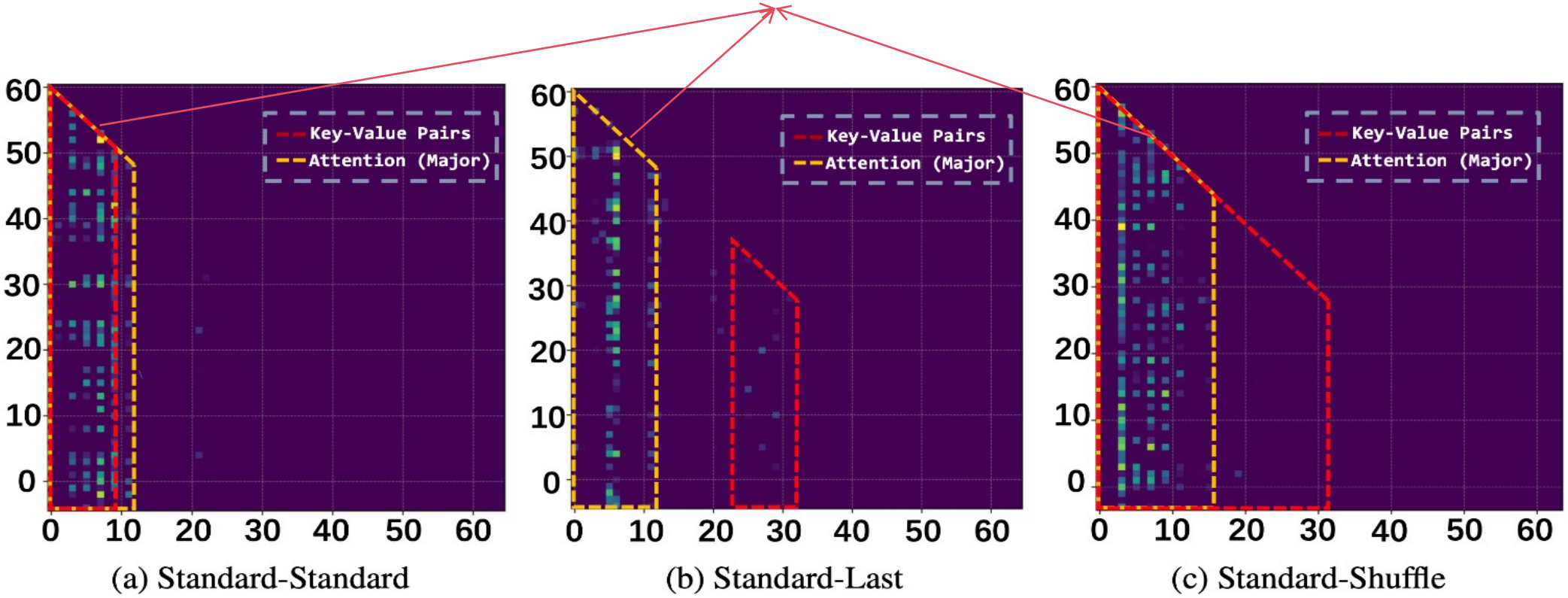


前沿工作 • 新模型架构

Revealing and Mitigating the Local Pattern Shortcuts of Mamba (Arxiv 2024)

项目： https://github.com/ZetangForward/Global_Mamba

- Mamba 的 Attention Map在不同任务上的可视化结果
  - 无论关键信息在哪里，Mamba的Attention一直集中在序列的开头
  - Mamba有着严重的Shortcuts



(a) Standard-Standard (b) Standard-Last (c) Standard-Shuffle

前沿工作 • 新模型架构

Revealing and Mitigating the Local Pattern Shortcuts of Mamba (Arxiv 2024)

项目： https://github.com/ZetangForward/Global_Mamba

- 只需要加一个全局的长门控单元（通过稀疏卷积网络实现）在原始的门控机制中就可以缓解Mamba的Shortcuts现象

$$\Delta_t = \mathbf{W}_2 \cdot \sigma\left(\mathbf{W}_1 \cdot \mathrm{Conv}_{\mathrm{short}}(\mathbf{X}_t)\right) \odot \sigma\left(\mathrm{Conv}_{\mathrm{long}}(\mathbf{X}_t)\right)$$

Models	Scale	Shuffle	Std-Last	Std-Shuffle	K2V2	K2V2-Robustness	K4V8-Shuffle
Pythia (Biderman et al., 2023)	*133m*	99.82	93.75	94.31	99.99	99.99	99.99
Hyena (Poli et al., 2023)	*153m*	✗	✗	✗	77.62	65.92	22.51
RWKV (Peng et al., 2023)	*153m*	✗	✗	✗	85.99	72.62	6.57
Mamba (Gu and Dao, 2023)	*129m*	80.98	15.44	22.37	99.98	66.01	✗
w/ 2×*State Size*	*130m*	88.57	40.22	31.88	99.84	78.90	✗
w/ 4×*State Size*	*134m*	96.92	35.89	32.88	99.84	57.11	✗
w/ *Global Selection*	*133m*	90.45	41.97	35.73	99.06	81.46	80.54

前沿工作 • 新模型架构

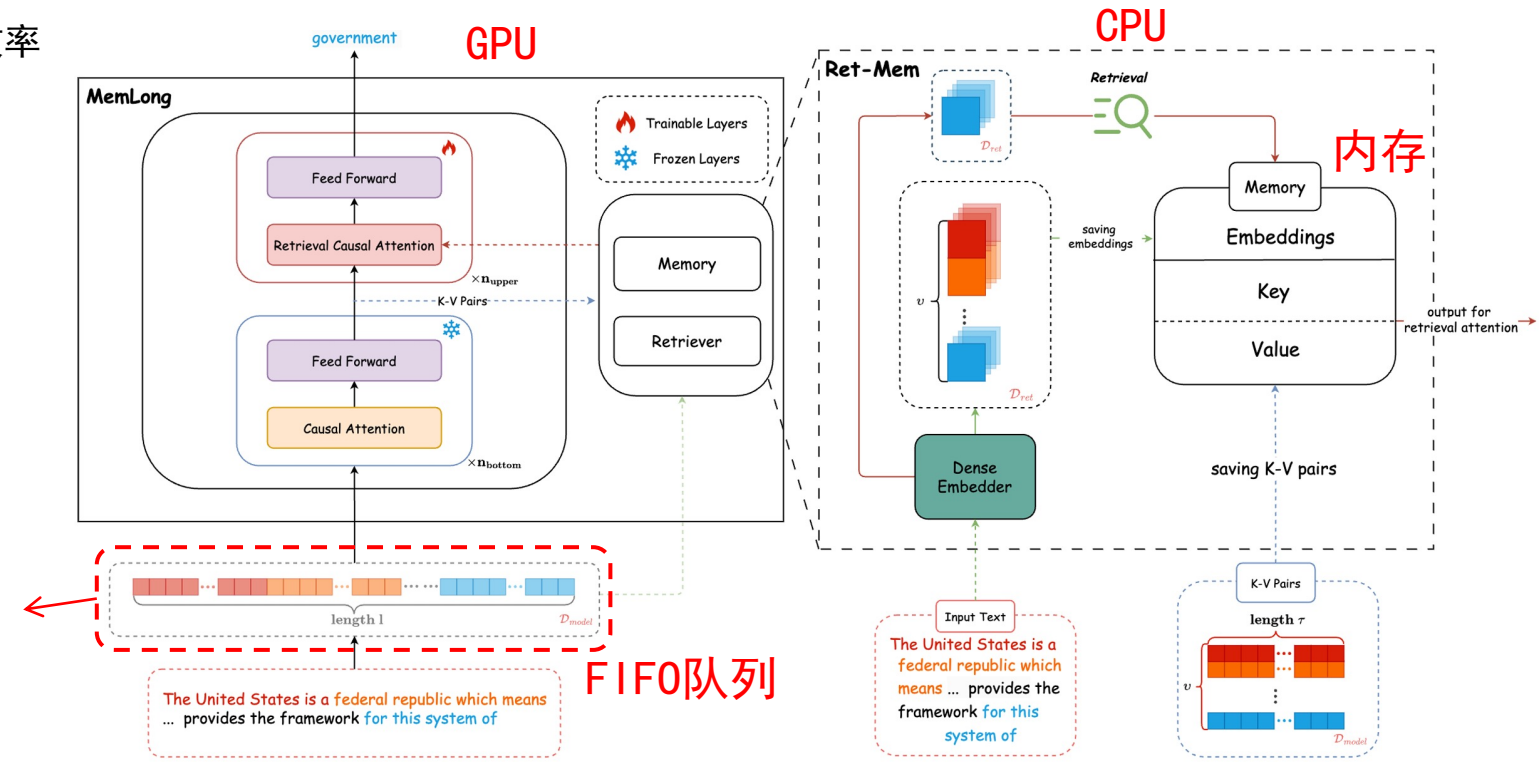
MemLong: Memory-Augmented Retrieval for Long Text Modeling (Arxiv 2024)

项目： https://github.com/Bui1dMySea/MemLong

- **难点：长序列生成场景对GPU显存有巨大挑战**
  - 推理不仅需要存储用户的历史输出，还需要存储模型生成的结果
  - 频繁将中间结果在内存和GPU中交换会影响推理效率
  - 推理过程中没法动态丢弃KV Cache

- 策略1：将注意力计算和KV Cache的存储分离
  - 内存 存储KV Cache
  - CPU 根据Q检索所需的KV Cache 片段
  - GPU 计算Q和检索的部分KV Cache的注意力

- 策略2：使用先进先出（ First-In-First-Out, FIFO ）队列在GPU中维护KV Cache，减少中间信息在内存和GPU显存中的交换。



前沿工作 • 新模型架构

MemLong: Memory-Augmented Retrieval for Long Text Modeling (Arxiv 2024)

项目： https://github.com/Bui1dMySea/MemLong

- 优点
  - 即插即用
  - 减少上下文长度，维持语言建模能力

Model	PG19				Proof-pile				BookCorpus				Wikitext-103			
	1k	2k	4k	16k	1k	2k	4k	16k	1k	2k	4k	16k	1k	2k	4k	16k
7B Model																
LLaMA-2-7B	10.82	10.06	8.92	-	3.24	3.40	2.72	-	8.73	7.91	6.99	-	10.82	6.49	5.66	-
LongLoRA-7B-32k	9.76	9.71	10.37	7.62	3.68	3.35	3.23	2.60	14.99	12.66	11.66	6.93	7.99	7.83	8.39	5.47
YARN-128k-7b	7.22	7.47	7.17	-	3.03	3.29	2.98	-	7.02	7.54	7.06	-	5.71	6.11	5.71	-
3B Model																
OpenLLaMA-3B	11.60	9.77	$> 10^3$	-	**2.96**	**2.70**	$> 10^3$	-	**8.97**	8.77	$> 10^3$	-	10.57	8.08	$> 10^3$	-
LongLLaMA-3B*	10.59	10.02	$> 10^3$	-	3.55	3.15	$> 10^3$	-	10.70	9.83	$> 10^3$	-	8.88	8.07	$> 10^3$	-
LongLLaMA-3B†	10.59	10.25	9.87	-	3.55	3.22	**2.94**	-	10.14	9.62	9.57	-	10.69	8.33	7.84	-
Phi3-128k	11.31	9.90	**9.66**	- / *9.65*	4.25	3.11	2.77	- / *3.08*	11.01	**9.22**	**8.98**	- / *9.27*	**7.54**	**7.22**	7.01	- / *7.20*
MemLong-3B*	10.66	10.09	$> 10^3$	-	3.58	3.18	$> 10^3$	-	10.37	9.55	$> 10^3$	-	8.72	7.93	$> 10^3$	-
w/ 4K Memory	10.54	9.95	9.89	**9.64**	3.53	3.16	3.15	2.99	10.18	9.50	9.57	9.61	8.53	7.92	7.87	7.99
w/ 32K Memory	**10.53**	**9.85**	**9.83**	9.73	3.51	3.15	3.11	**2.99**	9.64	9.56	**9.51**	**9.54**	8.02	7.58	**6.89**	**7.09**

超过上下文长度/显存爆炸导致PPL爆炸或无法测量

与3B大小的模型进行对比

在语言建模任务上进行评测，所有的实验都在3090 GPU（ 24GB显存）上进行测试。

前沿工作 • 长上下文模型评测

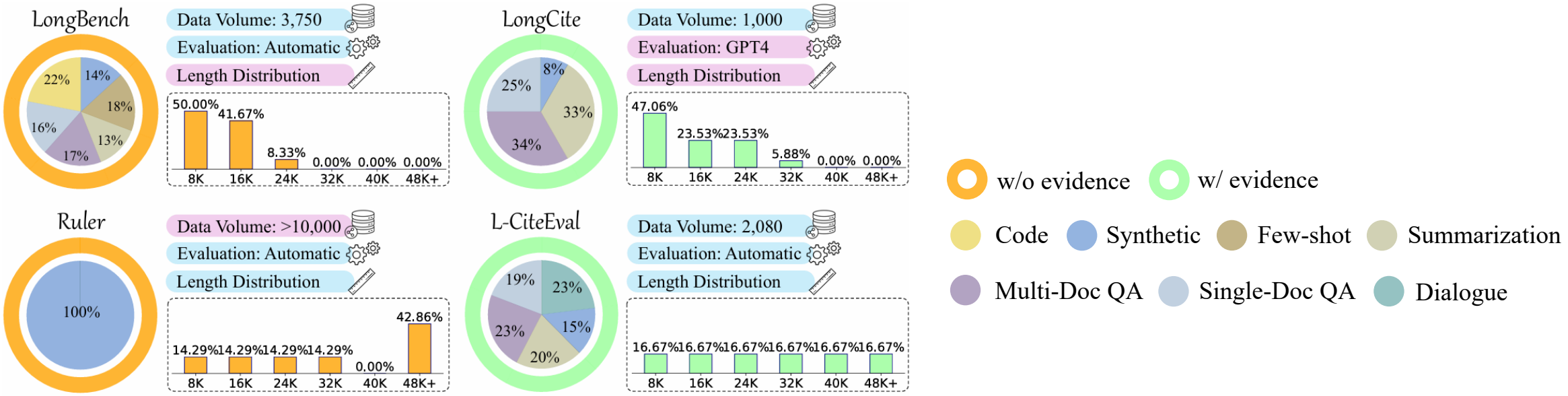
L-CiteEval: Do Long-Context Models Truly Leverage Context for Responding? (Arxiv 2024)

公众号：mp.weixin.qq.com/s/hHIXhHQrD-0sj7r4ifCS9A

项目：https://github.com/ZetangForward/L-CITEEVAL

- **难点：长上下文评测缺乏可解释性和忠实度检测**
  - 现有的大部分长上下文模型评测只关注模型生成结果，缺乏可解释性
  - 校验生成结果的忠实度困难，无法判断模型是根据自身知识还是上下文进行回复

现有长文本模型评测Benchmark的比较



前沿工作 • 长上下文模型评测

L-CiteEval: Do Long-Context Models Truly Leverage Context for Responding? (Arxiv 2024)

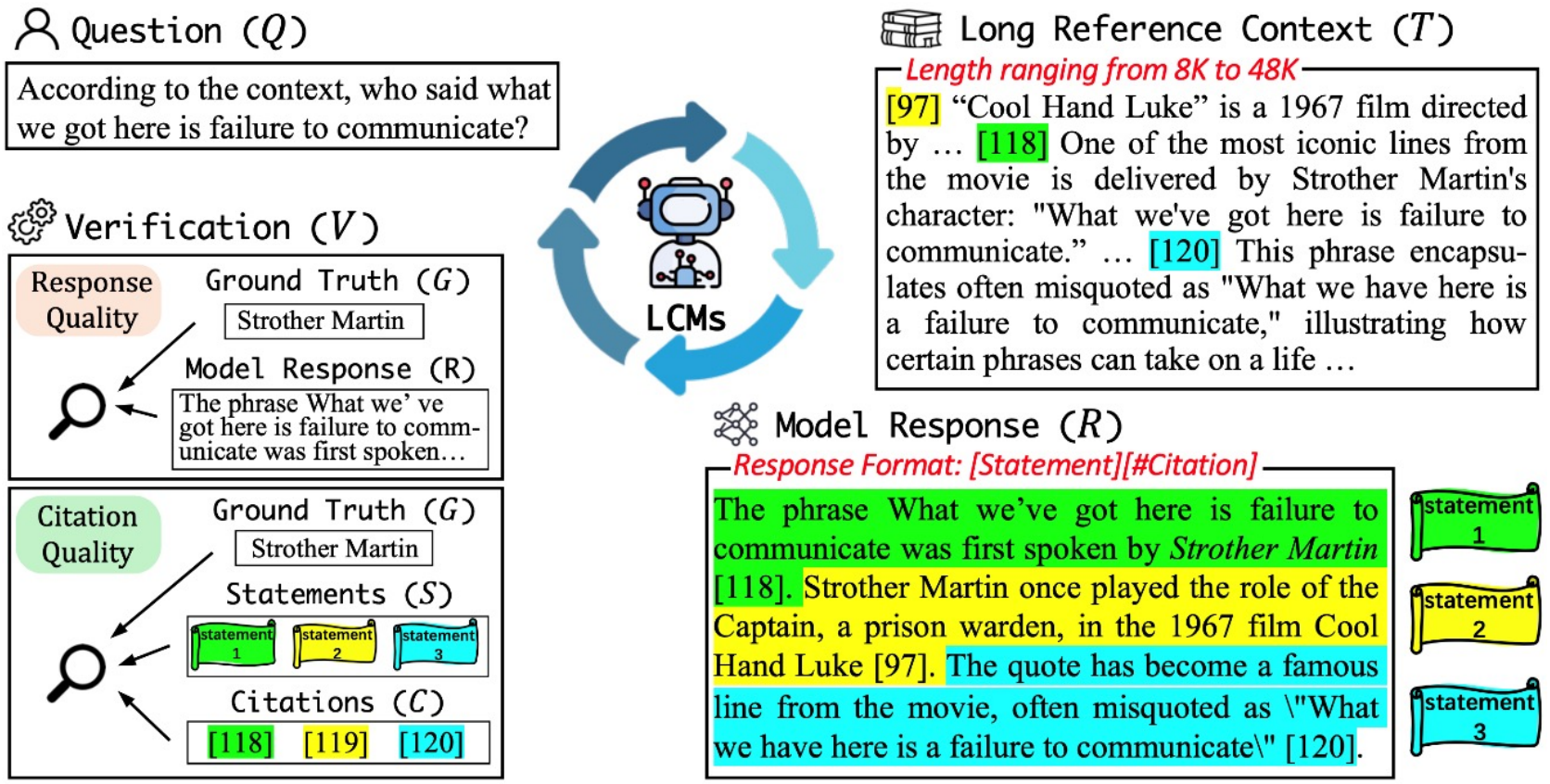
公众号：mp.weixin.qq.com/s/hHIXhHQrD-0sj7r4ifCS9A

项目：https://github.com/ZetangForward/L-CITEEVAL

L-CiteEval 不仅需要评测生成的结果，还要检验生成结果是否忠于上下文

生成结果评测

忠实度评测



前沿工作 • 长上下文模型评测

L-CiteEval: Do Long-Context Models Truly Leverage Context for Responding? (Arxiv 2024)

公众号：mp.weixin.qq.com/s/hHIXhHQrD-0sj7r4ifCS9A

项目：https://github.com/ZetangForward/L-CITEEVAL

- **闭源模型的结果**
  - GPT-4o 与 Claude-3.5-sonnet 不仅在生成结果上表现优秀，生成结果也忠于上下文
- **开源模型的结果**
  - 强开源模型在生成结果上和闭源模型已经很接近，但是忠实度上却相差很远
  - 在一些需要推理的任务上，闭源模型的忠实度明显好于开源模型
  - 中等规模开源模型与大型开源模型在忠实度指标上表现相当

生成结果评测结果

Models	Single-Doc QA		Multi-Doc QA		Summ.	Dialogue		Synthetic	
	Prec.	Rec.	Prec.	Rec.	Rouge-L	Prec.	Rec.	Rouge-1†	Acc‡
Closed-source LCMs									
GPT-4o	**11.78**	70.37	**10.34**	**87.38**	20.15	**9.81**	**65.35**	**89.24**	**91.88**
Claude-3.5-sonnet	5.96	**71.96**	4.30	80.77	**22.06**	3.71	57.80	88.33	69.65
o1-mini	10.30	66.44	7.36	64.25	19.22	7.02	54.27	54.98	57.29
Open-source LCMs									
Qwen2.5-3b-Ins	8.91	60.28	3.82	52.41	22.39	4.58	40.77	84.49	26.81
Phi-3.5-mini-Ins	8.62	62.34	7.82	64.54	19.48	11.39	52.77	73.83	61.32
Llama-3.1-8B-Ins	10.11	**68.13**	7.66	68.84	20.90	11.07	58.84	85.11	33.75
Glm-4-9B-chat	11.22	67.25	7.88	**77.97**	21.42	7.69	51.25	90.81	58.82
Mistral-Nemo-Ins	10.53	59.71	8.78	67.70	20.83	9.27	49.26	87.88	18.06
Qwen2-57B-A14B-Ins	12.93	61.71	15.25	57.53	**22.95**	14.32	52.23	**91.30**	63.61
Llama-3.1-70B-Ins	15.23	67.08	12.50	76.40	22.29	19.62	**62.91**	88.18	**89.03**
ChatQA-2-70B	**43.25**	61.20	**34.95**	55.64	22.06	**26.57**	58.34	70.14	78.68

忠实度结果评测

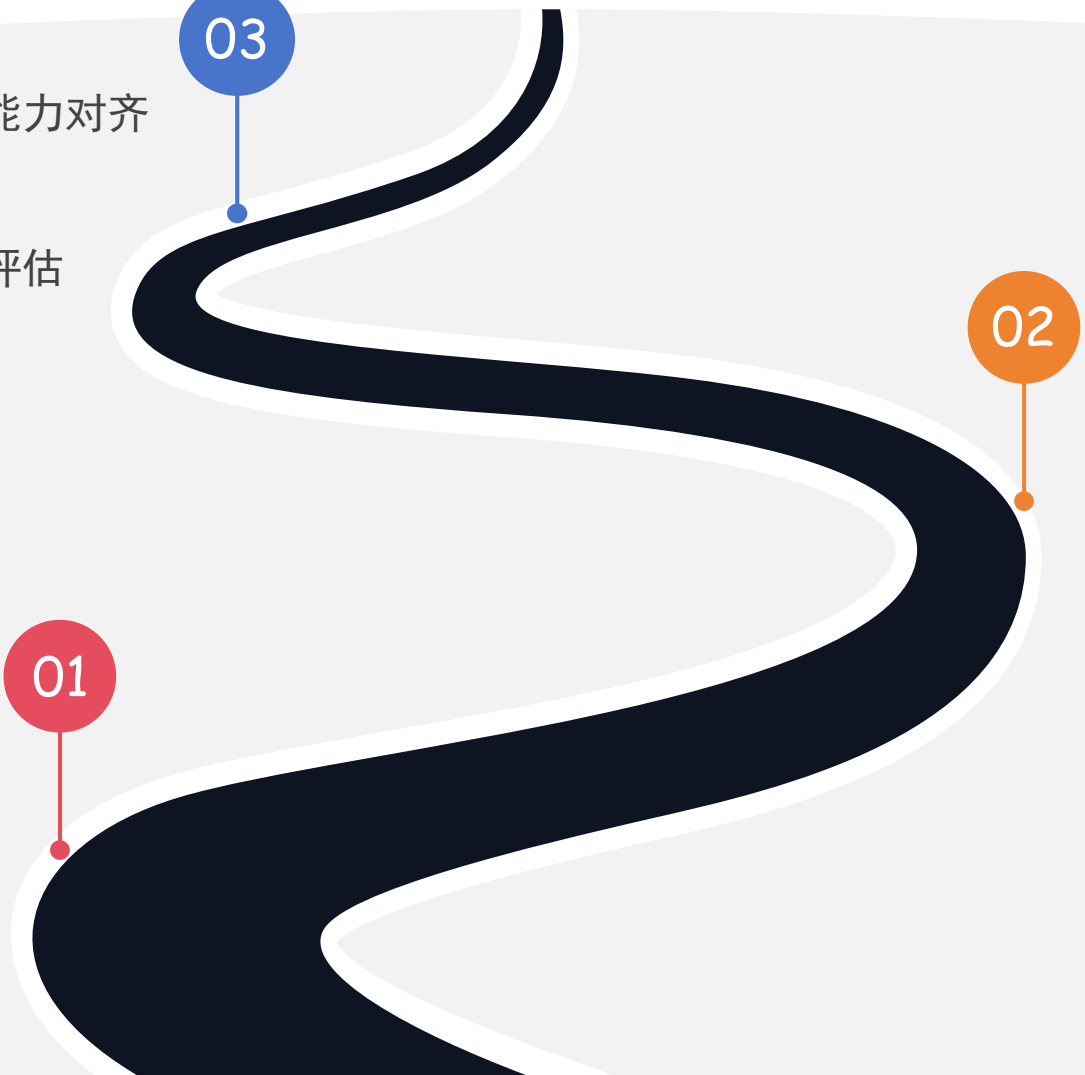
Models	Single-Doc QA				Dialogue Understanding				Needle in a Haystack			
	CP	CR	F_1	N	CP	CR	F_1	N	CP	CR	F_1	N
Closed-source LCMs												
GPT-4o	32.05	**38.12**	33.48	2.02	53.90	**64.25**	**56.76**	2.17	**76.25**	**76.67**	**76.39**	1.12
Claude-3.5-sonnet	**38.70**	37.79	**37.43**	3.54	**54.45**	50.48	51.45	2.83	65.00	68.33	65.97	1.04
o1-mini	29.83	35.33	31.66	3.38	45.54	50.74	47.21	2.63	25.42	28.33	26.25	1.58
Open-source LCMs												
Qwen2.5-3b-Ins	7.13	5.83	6.00	1.75	9.53	9.71	8.41	2.33	12.08	12.50	12.22	1.12
Phi-3.5-mini-Ins	21.06	20.46	19.14	2.86	20.39	24.27	20.57	2.27	11.11	12.50	11.53	1.20
Llama-3.1-8B-Ins	22.68	24.73	22.64	2.59	51.86	**57.58**	53.50	2.08	34.31	35.83	34.72	0.99
Glm-4-9B-chat	**29.00**	**28.66**	**28.05**	2.21	**54.54**	55.62	**53.58**	1.78	46.53	**50.83**	**47.78**	1.23
Mistral-Nemo-Ins	4.34	3.68	3.76	0.68	23.91	24.33	23.50	1.35	11.11	12.50	11.53	1.18
Qwen2-57B-A14B-Ins	4.90	3.43	3.82	1.27	22.63	22.54	21.61	1.80	15.28	15.83	15.42	1.17
Llama-3.1-70B-Ins	25.89	26.89	26.11	1.23	51.71	56.20	53.19	1.76	**46.67**	46.67	46.67	0.82
ChatQA-2-70B	21.75	22.54	21.92	1.12	47.67	51.25	48.77	1.29	38.33	38.33	38.33	0.95

Models	Multi-Doc QA				Summarization				Counting Stars			
	CP	CR	F_1	N	CP	CR	F_1	N	CP	CR	F_1	N
Closed-source LCMs												
GPT-4o	57.48	**58.50**	56.10	1.71	34.37	54.28	41.60	22.86	**83.37**	**81.18**	**81.71**	4.54
Claude-3.5-sonnet	**66.85**	55.62	**58.58**	2.44	**36.70**	**55.03**	**43.45**	17.70	73.01	75.83	73.15	4.81
o1-mini	49.95	49.60	48.58	1.78	20.23	33.61	24.83	19.58	34.06	46.46	38.45	6.73
Open-source LCMs												
Qwen2.5-3b-Ins	13.17	8.04	9.37	1.96	7.72	12.15	9.09	9.52	3.82	1.48	2.01	1.66
Phi-3.5-mini-Ins	11.89	10.25	10.53	1.71	10.90	10.94	9.60	8.23	4.19	3.67	4.09	3.48
Llama-3.1-8B-Ins	43.41	42.15	41.64	1.62	19.57	23.03	20.83	18.31	16.87	18.26	19.18	4.19
Glm-4-9B-chat	47.91	44.75	45.09	1.64	**29.16**	**37.29**	**31.92**	11.38	18.15	15.69	16.21	4.52
Mistral-Nemo-Ins	17.61	15.45	15.85	0.70	11.21	14.85	12.40	5.45	3.09	2.92	3.26	2.32
Qwen2-57B-A14B-Ins	17.30	12.07	13.61	1.06	4.01	3.37	3.19	3.81	4.37	4.37	4.24	4.24
Llama-3.1-70B-Ins	**49.64**	**54.02**	**50.74**	1.42	25.50	31.99	27.91	11.78	**66.85**	**61.74**	**63.73**	4.37
ChatQA-2-70B	47.20	49.51	47.92	1.10	19.57	23.60	20.89	11.81	14.02	11.22	13.22	3.49

总结

长上下文模型 —— 进展与挑战

- 什么是长上下文模型？ 01
- 如何训练长上下文模型？ 02
  - 建模
  - 数据
  - 评测
- 前沿与挑战 03
  - 高效长上下文能力对齐
  - 新模型架构
  - 长上下文能力评估



PPT与相关资料链接

Q & A

PPT制作：汤泽成

E-Mail：zecheng.tang@foxmail.com

ljt@suda.edu.cn