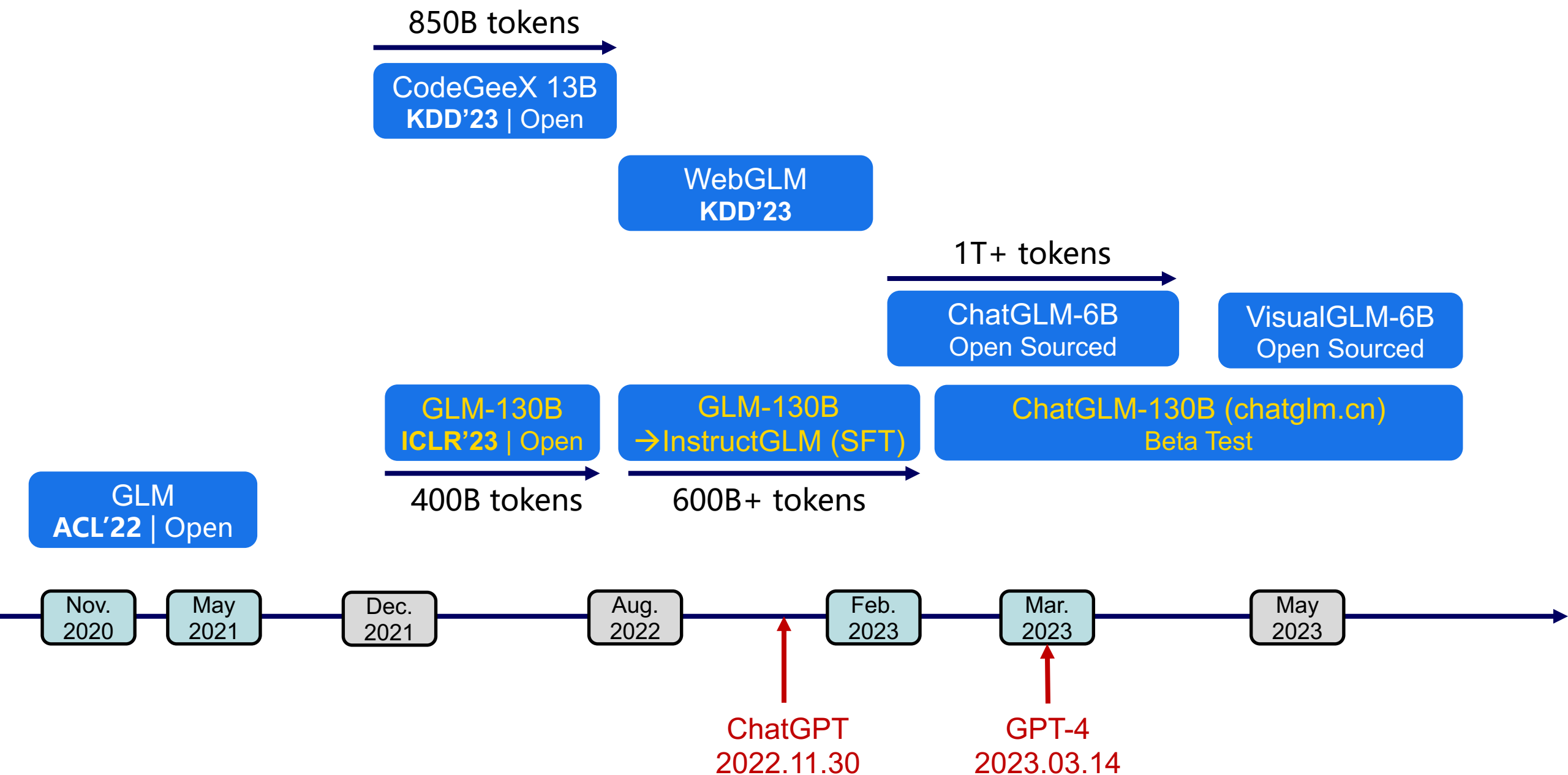


From GLM-130B to ChatGLM



从千亿模型 GLM-130B 到 ChatGLM 的一点尝试

东昱晓

知识工程实验室 (KEG)
清华大学计算机系

<https://keg.cs.tsinghua.edu.cn/yuxiao>



chat.openai.com全球访问量

152.7K

NOV

2022.11.30

266.0M

DEC

616.0M

JAN

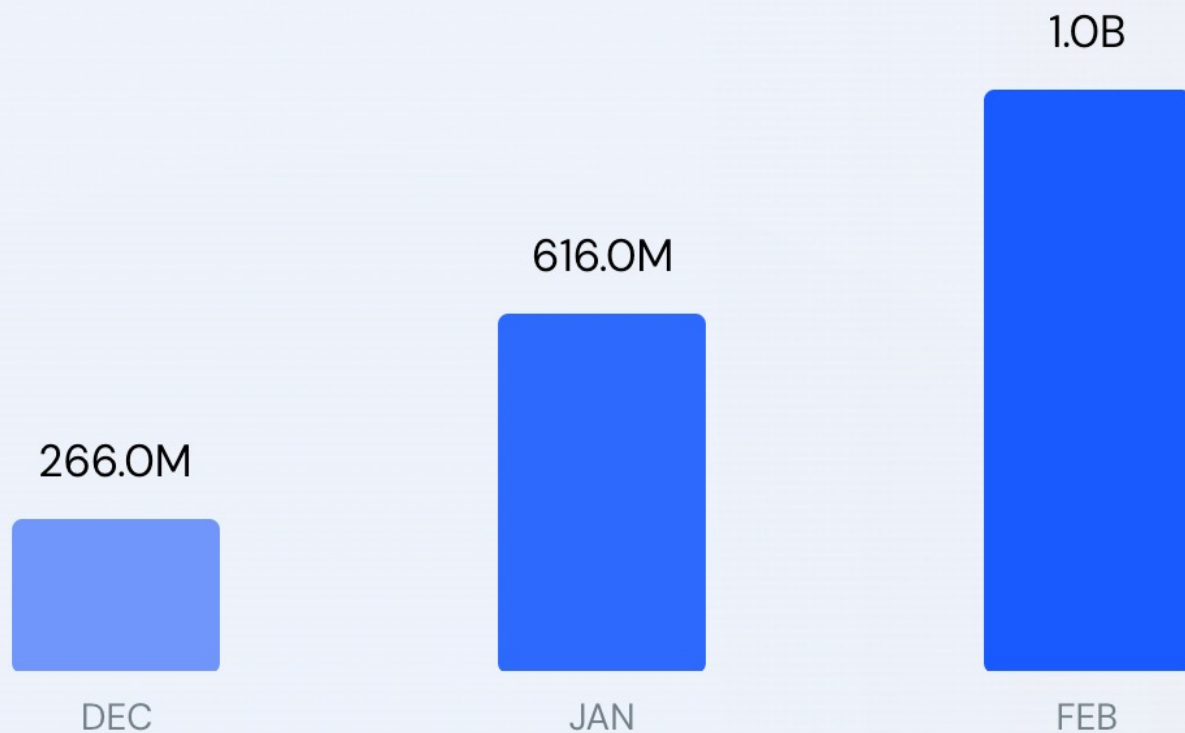
1.0B

FEB

chat.openai.com

2022.11.30日推出

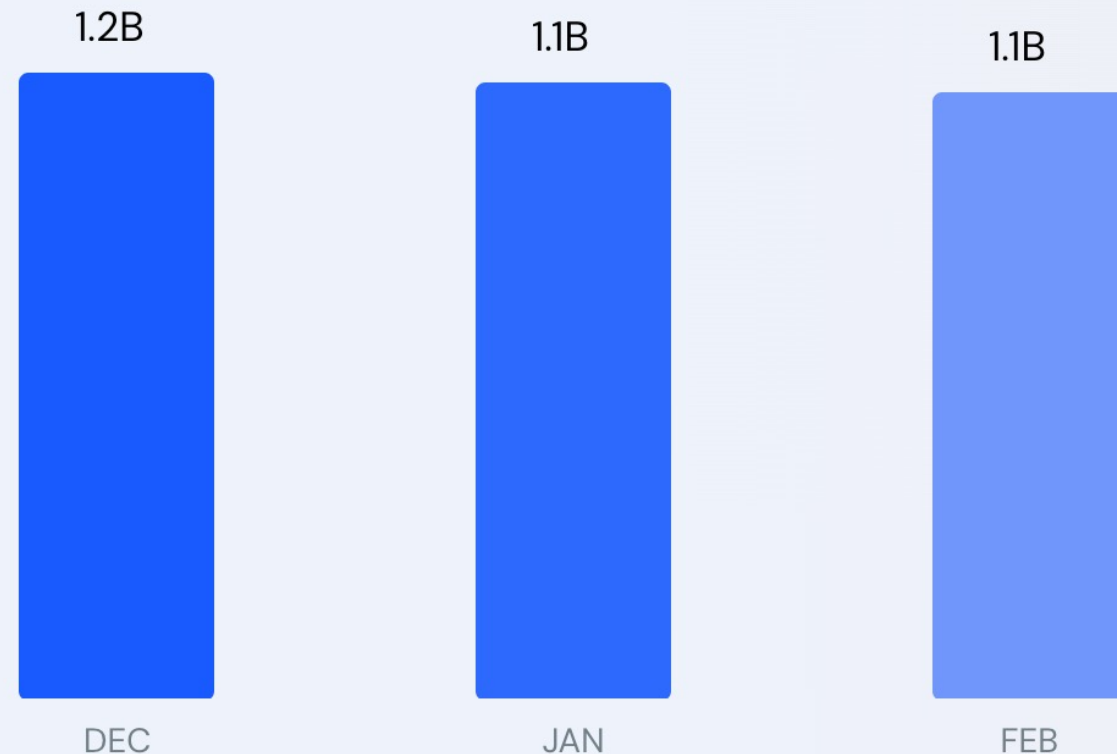
Apr. **18 亿**



bing.com

2009.05.28 (MSN Search, 1998)

Apr. **13 亿**



1950-2022

“The **Great Wall of China** was built from as early as **the 7th century BC**, with selective stretches later joined by **Qin Shi Huang** (220–206 BC), the first emperor of China. ”

“It is really **gorgeous!** ”

2020~2022

Math (GSM8k):

Q: Roger has 5 tennis balls. He buys 2 more cans of tennis balls. Each can has 3 tennis balls. How many tennis balls does he have now?

Last Letter Concatenation:

Q: Take the last letters of the words in “Lady Gaga” and concatenate them.

2022.11.30 → 2023.3.14 →

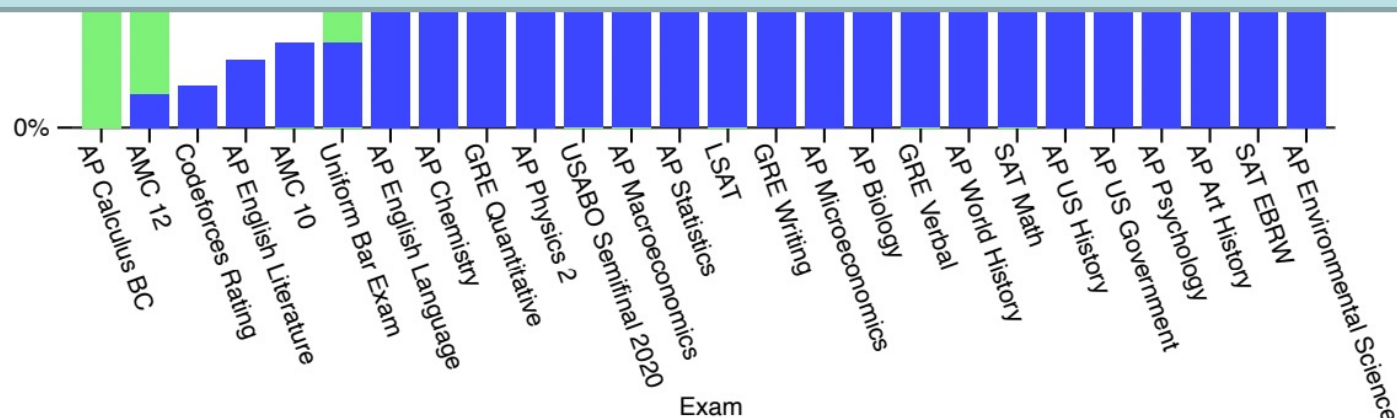
Exam results (ordered by GPT-3.5 performance)

Estimated percentile lower bound (among test takers)

100% —

gpt-4
gpt-4 (no vision)
gpt3.5

ChatGPT、GPT-4的出现表明
我们在大模型上的研究还存在代差



2022.11.30 → 2023.3.14 →

	GPT-4 Evaluated few-shot	GPT-3.5 Evaluated few-shot	LM SOTA Best external LM evaluated few-shot	SOTA Best external model (incl. benchmark-specific tuning)
MMLU [49] Multiple-choice questions in 57	86.4% 5-shot	70.0% 5-shot	70.7% 5-shot	75.2% 5-shot Flan-PaLM [51]

**ChatGPT、GPT-4的出现表明
我们在大模型上的研究还存在代差**

Python coding tasks DROP [58] (F1 score)	0-shot 80.9	0-shot 64.1	0-shot PaLM [3] 70.8	CodeT + GPT-3.5 [57] 88.4
Reading comprehension & arithmetic.	3-shot	3-shot	1-shot PaLM [3]	QDGAT [59]
GSM-8K [60] Grade-school mathematics questions	92.0%* 5-shot chain-of-thought	57.1% 5-shot	58.8% 8-shot Minerva [61]	87.3% Chinchilla + SFT+ORM-RL, ORM reranking [62]

2022.11.30 → 2023.3.14 →

What is funny about this image? Describe it panel by panel.



GPT-4

The image shows a package for a "Lightning Cable" adapter with three panels.

Panel 1: A smartphone with a VGA connector (a large, blue, 15-pin connector typically used for computer monitors) plugged into its charging port.

Panel 2: The package for the "Lightning Cable" adapter with a picture of a VGA connector on it.

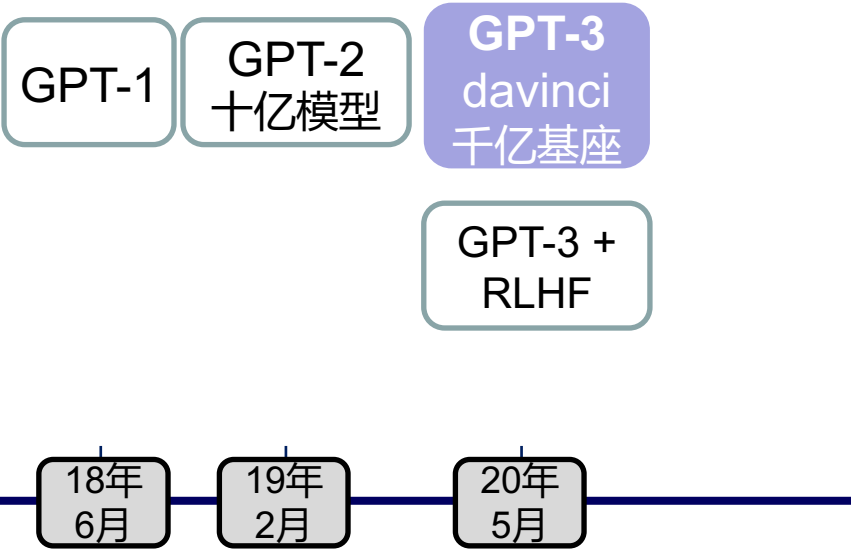
Panel 3: A close-up of the VGA connector with a small Lightning connector (used for charging iPhones and other Apple devices) at the end.

Source: <https://www.reddit.com/r/hmmm/comments/ubab5v/hmmm/>

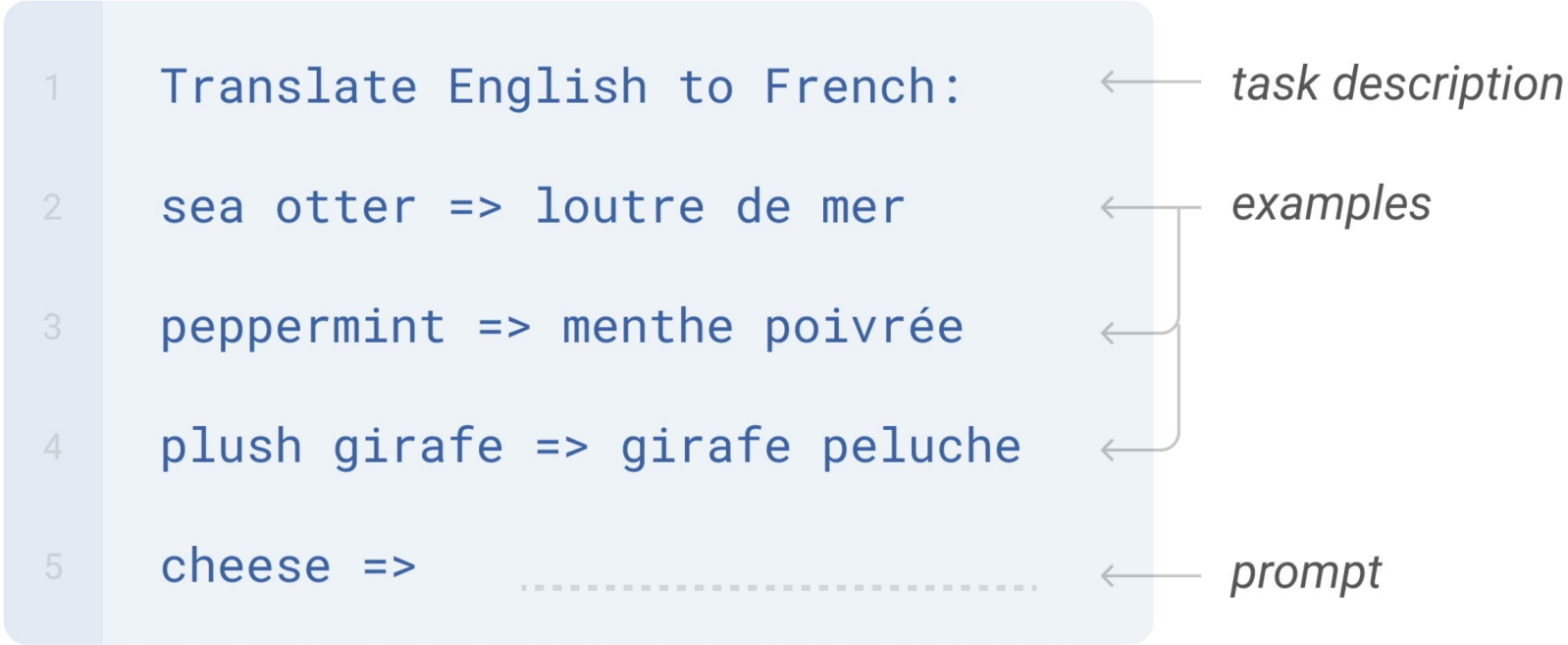
OpenAI GPT系列模型

Few-shot

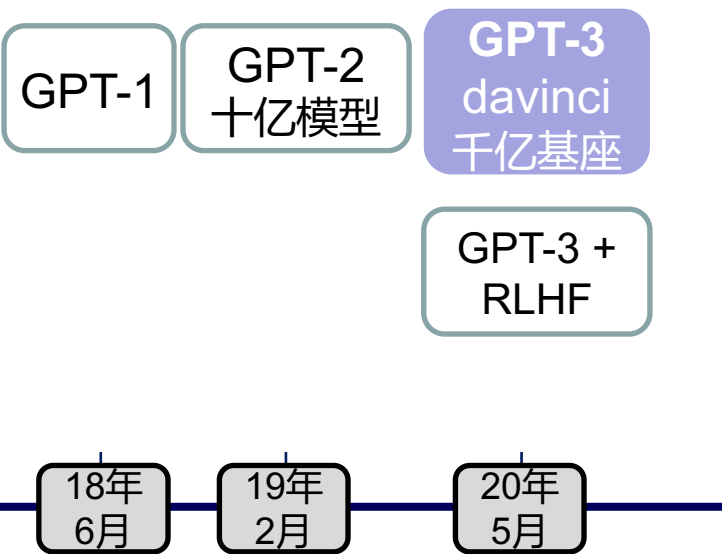
In addition to the task description, the model sees a few examples of the task. No gradient updates are performed.



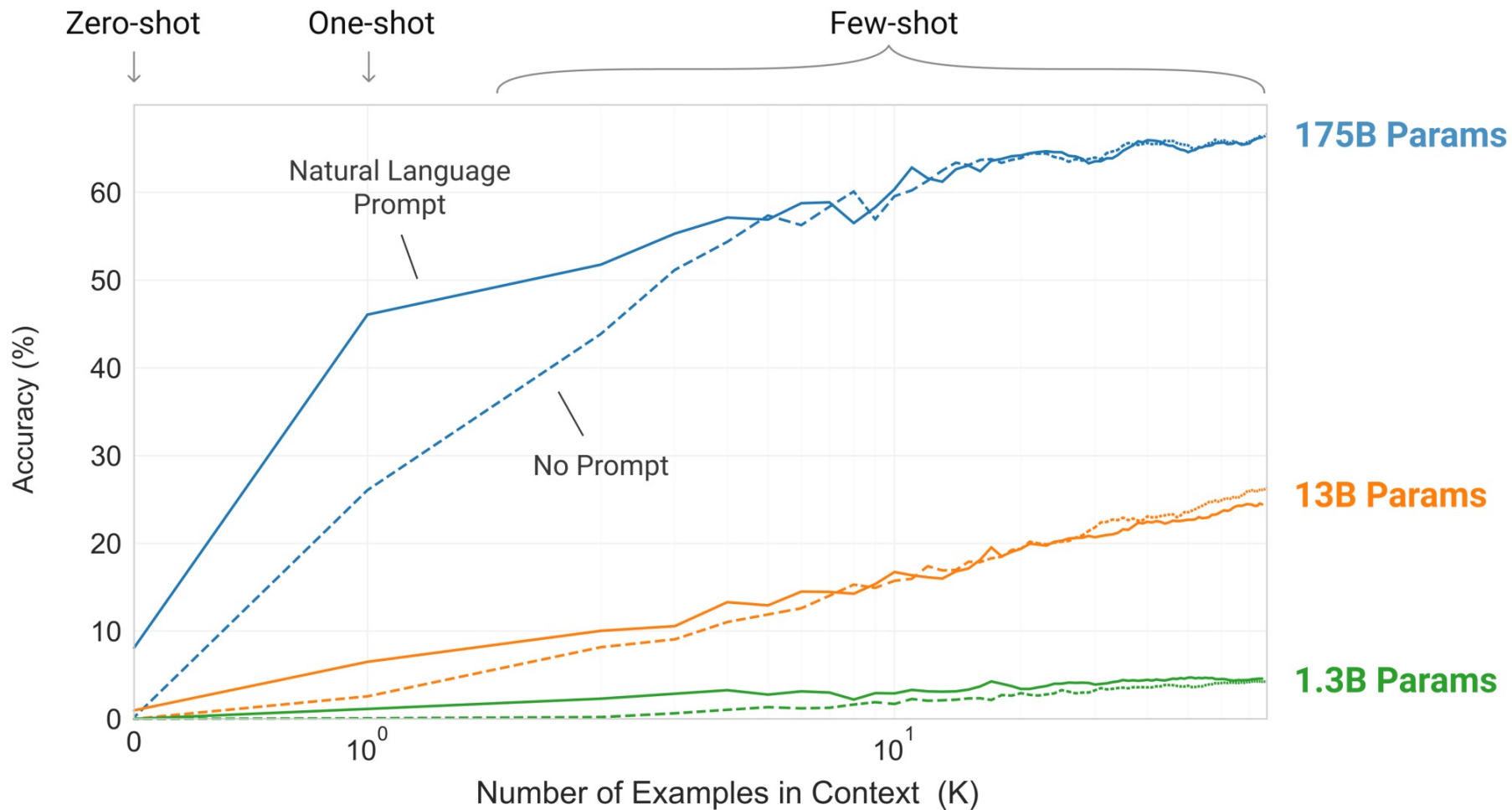
1. 千亿基座模型



OpenAI GPT系列模型

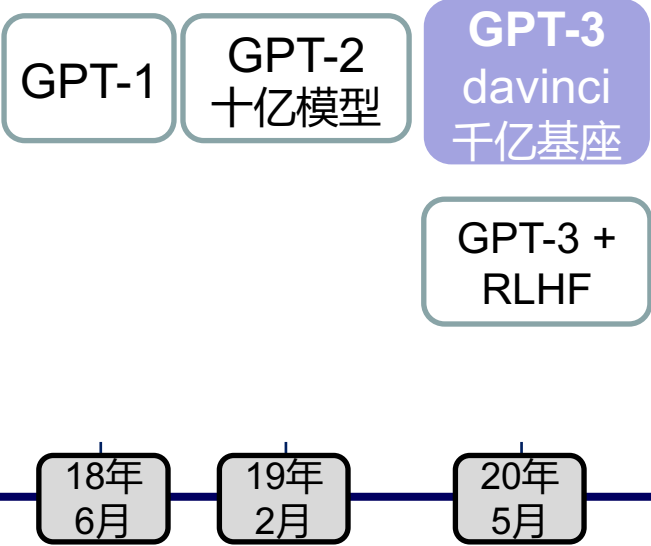


1. 千亿基座模型



OpenAI GPT系列模型

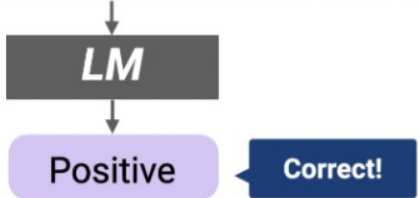
- In-context Learning: A is B; C is D; Please answer: E is ___ ?



1. 千亿基座模型

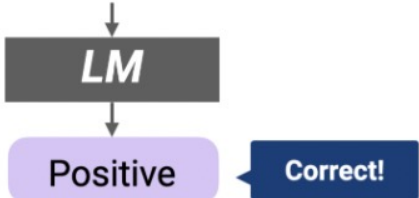
Ground truth

Circulation revenue has increased by 5% in Finland.	\n Positive
Panostaja did not disclose the purchase price.	\n Neutral
Paying off the national debt will be extremely painful.	\n Negative
The company anticipated its operating profit to improve.	\n _____

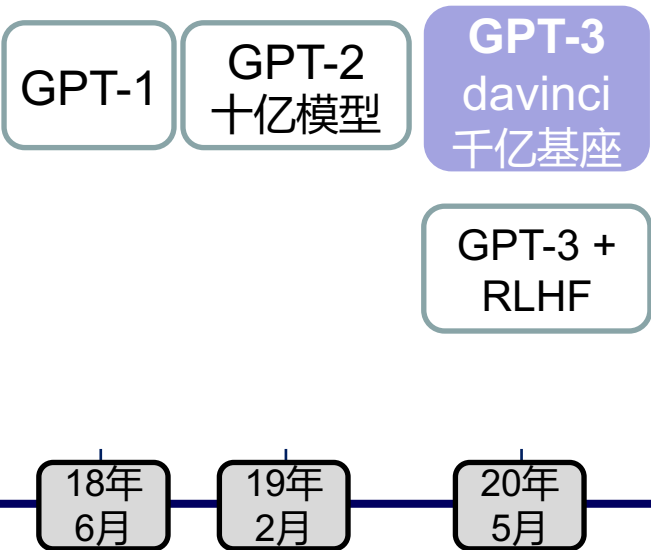


Random

Circulation revenue has increased by 5% in Finland.	\n Neutral
Panostaja did not disclose the purchase price.	\n Negative
Paying off the national debt will be extremely painful.	\n Positive
The company anticipated its operating profit to improve.	\n _____



OpenAI GPT系列模型



1. 千亿基座模型

Prompt

Explain the moon landing to a 6 year old in a few sentences.

Completion

GPT-3

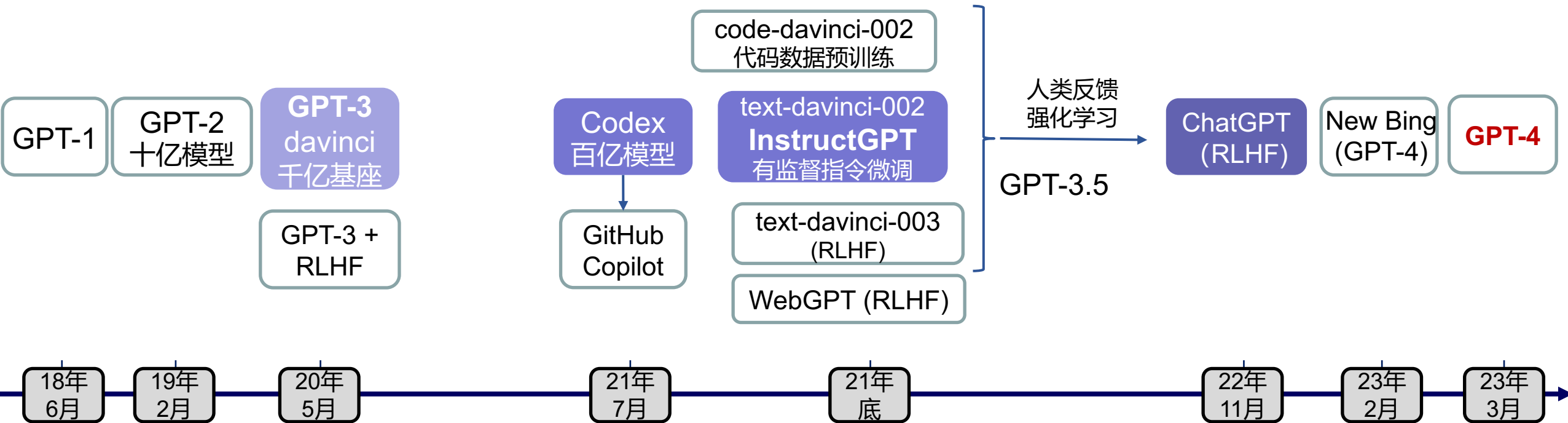
Explain the theory of gravity to a 6 year old.

Explain the theory of relativity to a 6 year old in a few sentences.

Explain the big bang theory to a 6 year old.

Explain evolution to a 6 year old.

OpenAI GPT系列模型



1. 千亿基座模型

2. 有监督指令微调

3. 人类反馈强化学习

OpenAI GPT系列模型

Prompt

Explain the moon landing to a 6 year old in a few sentences.

Completion

GPT-3

Explain the theory of gravity to a 6 year old.

Explain the theory of relativity to a 6 year old in a few sentences.

Explain the big bang theory to a 6 year old.

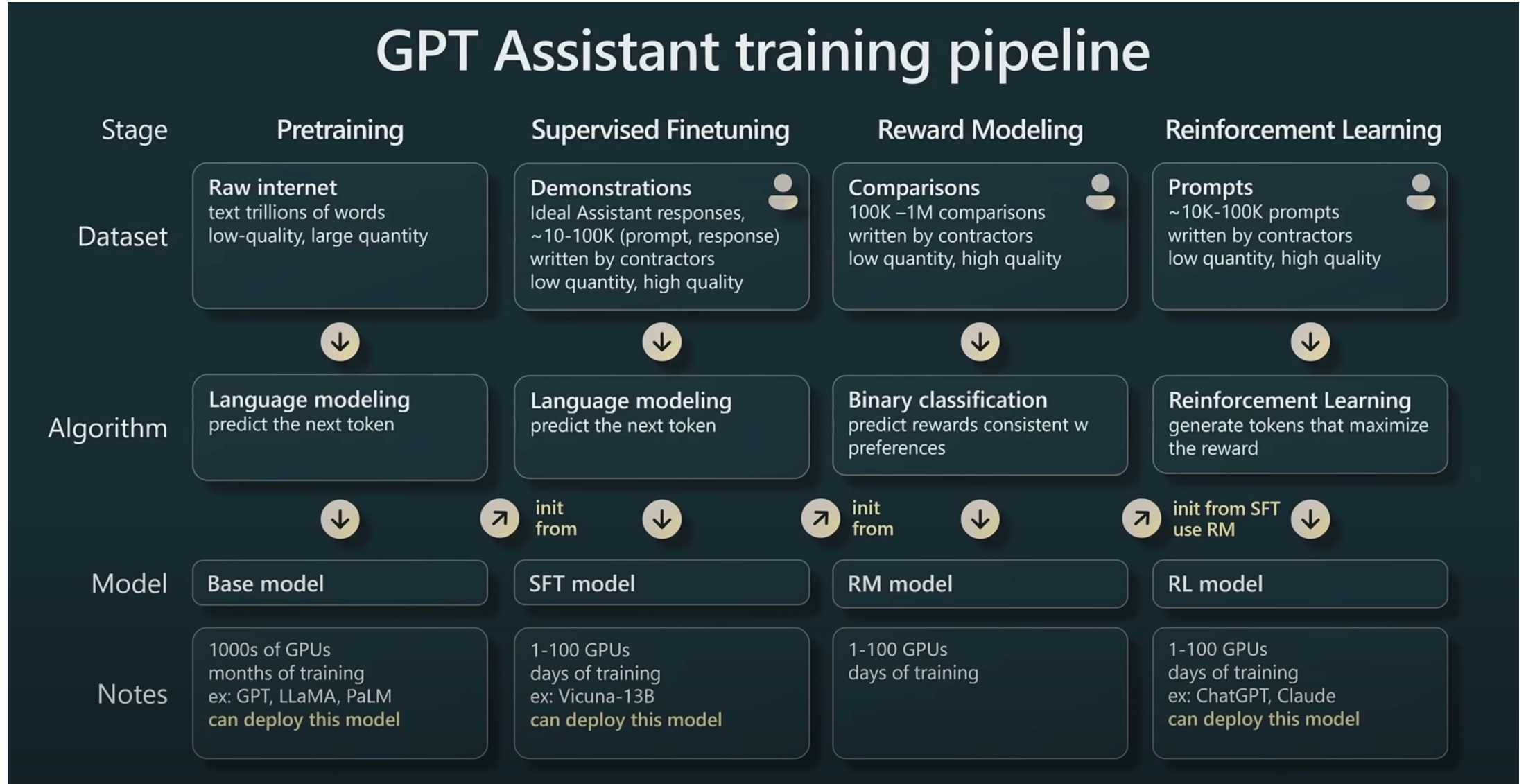
Explain evolution to a 6 year old.

InstructGPT

People went to the moon, and they took pictures of what they saw, and sent them back to the earth so we could all see them.

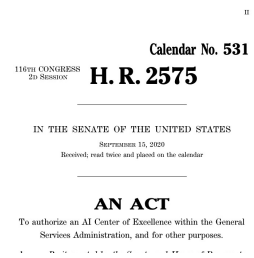
State of GPT by Andrej Karpathy

GPT Assistant training pipeline



相关背景

美国国家战略



2020年美国国会推动**AI模型进入联邦政府运作**形成《政府AI法案》

2019-2020美国白宫接连以两道总统行政令强调将在未来加大以通用大模型在内的AI突破领域资助力度以实现美国在AI领域的领导权

GPT&ChatGPT 某种程度对中国禁用，**A100&H100等高端 AI 芯片对中国禁售**

国际学术热点

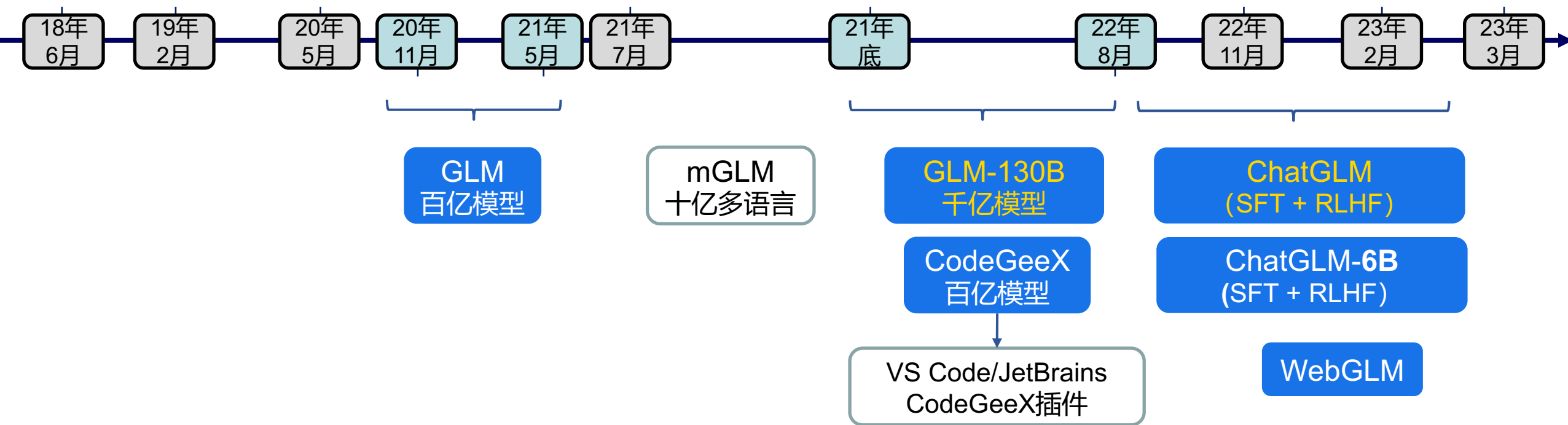
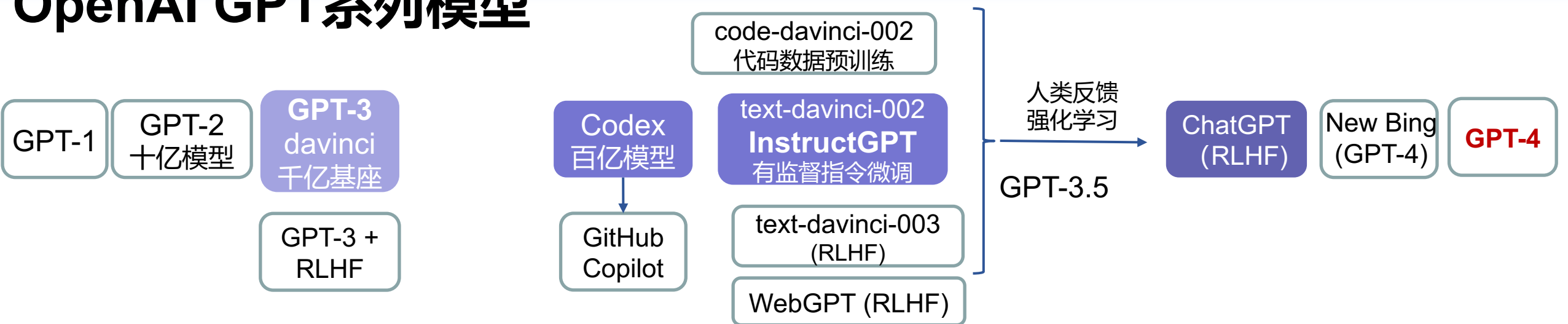


预训练模型Alpha系列**四登Nature封面文章**，突破人类蛋白质结构认识边界等

2022年，已有**超过10篇**超大规模预训练模型论文发表于Nature, Science

Science评出的年度十大科技进展重点提到**大模型研究**

OpenAI GPT系列模型



清华&智谱 GLM 系列模型

GPT-3
davinci
千亿基座
• NVIDIA

3000亿标识符
(2020.05)

Transformer
(1750亿参数)

GPT
自回归生成

数据

神经网络

预训练架构

GLM-130B
千亿模型

- NVIDIA
- 海光 DCU
- 昇腾910
- 申威

2000亿中文
2000亿英文
(2022.07)

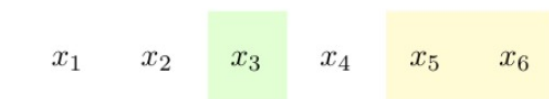
Transformer
(1300亿参数)

GLM
自回归填空

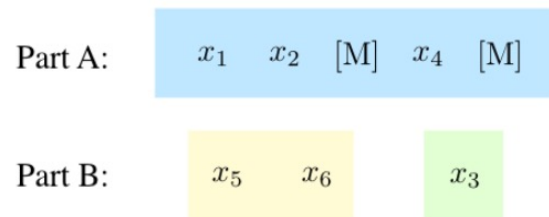
通用语言模型：GLM—自回归填空

Autoregressive Blank Filling

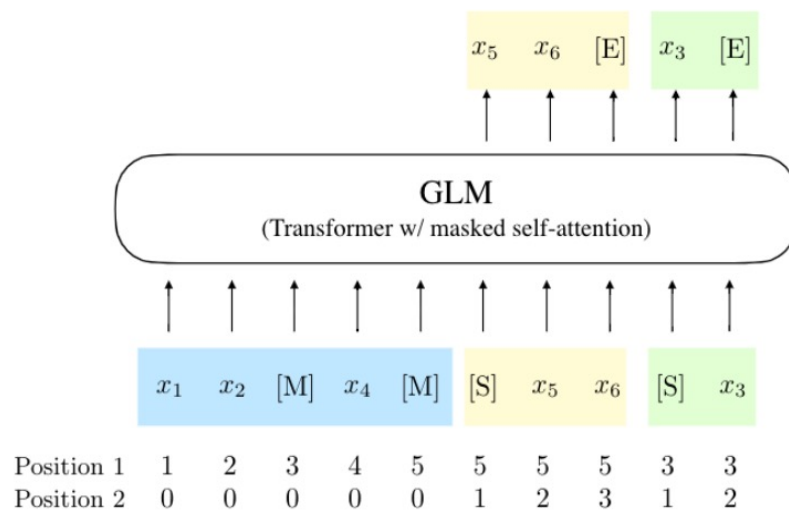
算法框架	生成 vs. 理解	自然语言理解	Cond. Gen.	Uncond. Gen.
自回归 (GPT)	单向注意力	—	—	✓
自编码 (BERT)	双向注意力	✓	×	×
编码器-解码器 (T5)	编解码	—	✓	—
自回归填空 (GLM)	双向注意力	✓	✓	✓



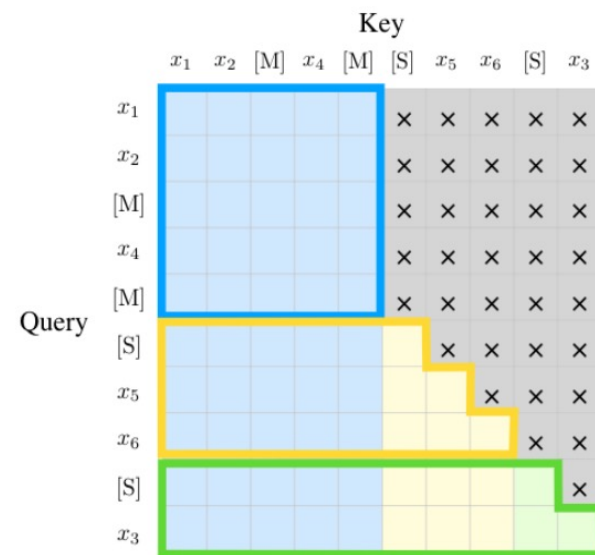
(a) Sample spans from the input text



(b) Divide the input into Part A and Part B



(c) Generate the Part B spans autoregressively



(d) Self-attention mask

通用语言模型：GLM—自回归填空

多任务预训练

- 从文本中间采样15%作为生成目标
- 从文本后面采样50-100%作为生成目标

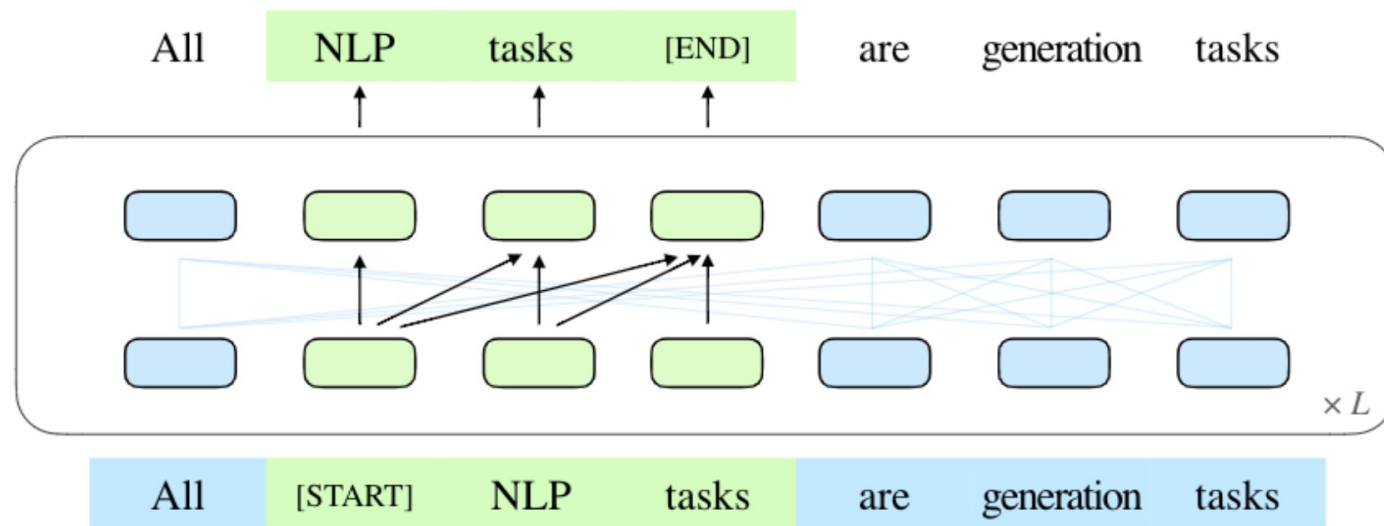
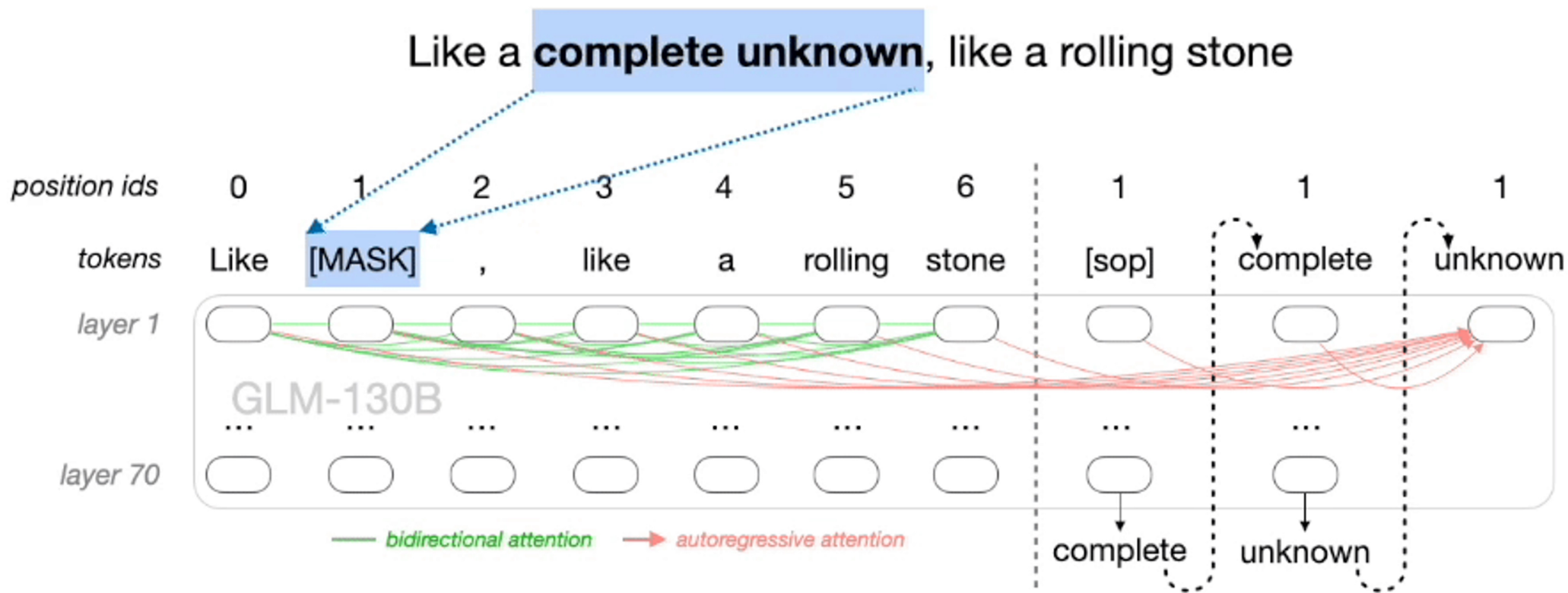


Figure 1. Illustration of GLM. We blank out text spans (green part) and GLM is trained to generate them in an autoregressive fashion.

通用语言模型：GLM—自回归填空

- General Language Model (GLM)：自回归填空
- 预训练：双向注意力、对masked字段进行预测



GLM用于文本理解NLU

Table 2. Results on the SuperGLUE dev set. Models with * are pre-trained for two times the number of steps of other methods.

Model	ReCoRD F1/Acc.	COPA Acc.	WSC Acc.	RTE Acc.	BoolQ Acc.	WiC Acc.	CB F1/Acc.	MultiRC F1a/EM	Avg
BERT _{Base}	65.4/64.9	66.0	65.4	70.0	74.9	68.8	70.9/76.8	68.4/21.5	66.1
GLM _{Base}	73.5/72.8	71.0	72.1	71.2	77.0	64.7	89.5/85.7	72.1/26.1	70.7
BERT _{Large}	76.3/75.6	69.0	64.4	73.6	80.1	71.0	94.8/92.9	71.9/24.1	72.0
UniLM _{Large}	80.0/79.1	72.0	65.4	76.5	80.5	69.7	91.0/91.1	77.2/38.2	74.1
GLM _{Large}	81.7/81.1	76.0	81.7	74.0	82.1	68.5	96.1/94.6	77.1/36.3	77.0
GLM _{Large} (multi-task)	80.2/79.6	77.0	78.8	76.2	79.8	63.6	97.3/96.4	74.6/32.1	75.7
GLM _{410M} (multi-task)	81.5/80.9	80.0	81.7	79.4	81.9	69.0	93.2/96.4	76.2/35.5	78.0
GLM _{515M} (multi-task)	82.3/81.7	85.0	81.7	79.1	81.3	69.4	95.0/96.4	77.2/35.0	78.8
T5 _{Base}	76.2/75.4	73.0	79.8	78.3	80.8	67.9	94.8/92.9	76.4/40.0	76.0
T5 _{Large}	85.7/85.0	78.0	84.6	84.8	84.3	71.6	96.4/98.2	80.9/46.6	81.2
BART _{Large} *	88.3/87.8	60.0	65.4	84.5	84.3	69.0	90.5/92.9	81.8/48.0	76.0
RoBERTa _{Large} *	89.0/88.4	90.0	63.5	87.0	86.1	72.6	96.1/94.6	84.4/52.9	81.5
GLM _{RoBERTa}	89.6/89.0	82.0	83.7	87.7	84.7	71.2	98.7/98.2	82.4/50.1	82.9

- 精度超过BERT、T5、RoBERTa

GLM用于文本生成

Table 3. Results on Gigaword abstractive summarization

Model	RG-1	RG-2	RG-L
MASS	37.7	18.5	34.9
UniLM _{Large}	38.5	19.5	35.8
GLM _{Large}	38.6	19.7	36.0
GLM _{Large} (multi-task)	38.5	19.4	35.8
GLM _{410M} (multi-task)	38.9	20.0	36.2

- **更重要的是: 仅需一个模型即可完成不同个任务**

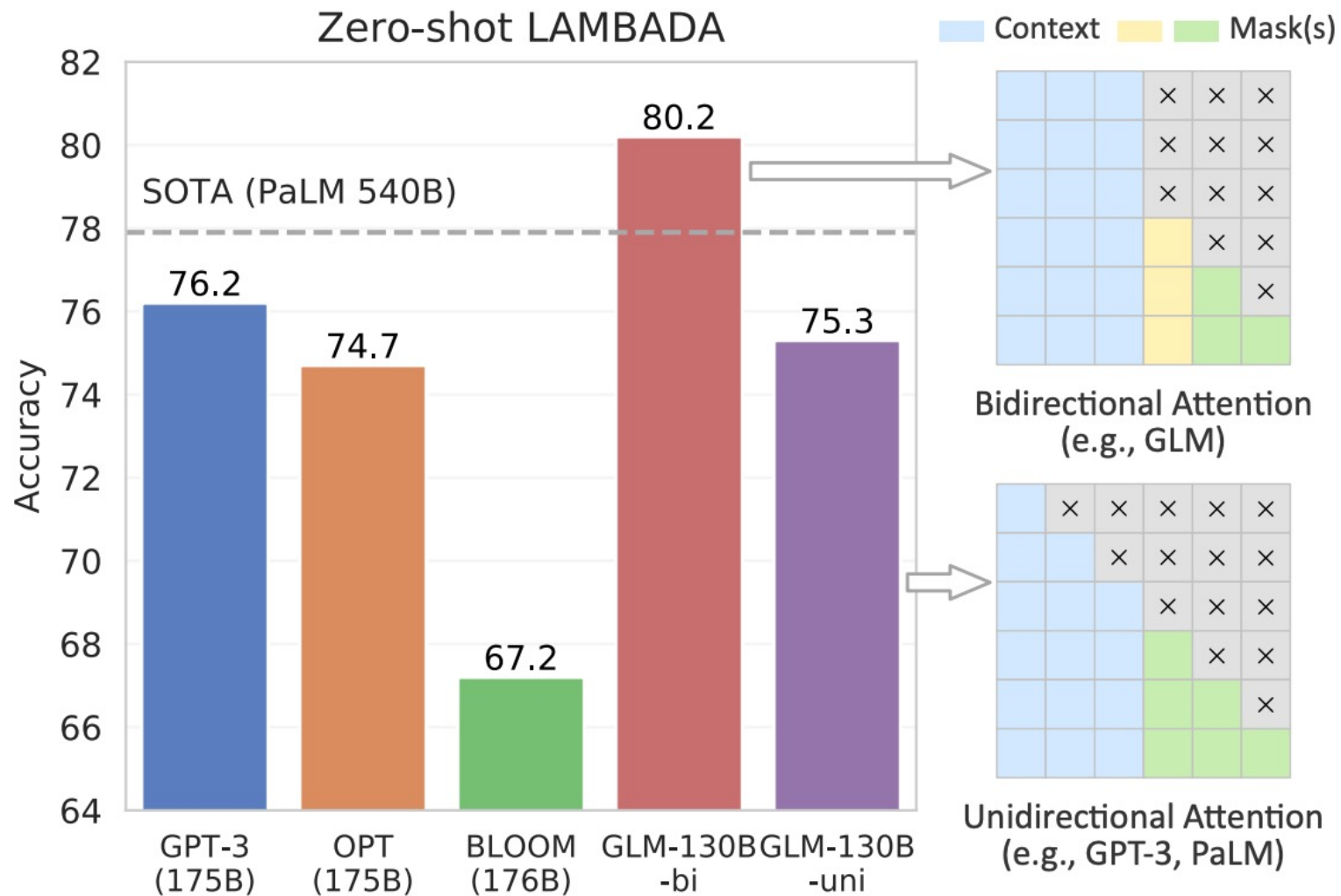
Table 4. Zero-shot language modeling results.

Model	Lambada (Accuracy)	BookWiki (Perplexity)
GLM _{Large} (uni)	0.0	> 100
GLM _{Large} (multi-task,uni)	47.4	15.1
– 2d positional encoding	45.8	15.1
GLM _{410M} (multi-task,uni)	49.5	14.5
GLM _{515M} (multi-task,uni)	50.4	13.9
GLM _{Large} (bi)	10.6	> 100
GLM _{Large} (multi-task,bi)	48.5	14.9
– 2d positional encoding	47.3	15.0
GLM _{410M} (multi-task,bi)	53.5	14.3
GLM _{515M} (multi-task,bi)	54.9	13.7
GPT _{Large} (uni)	50.1	14.4

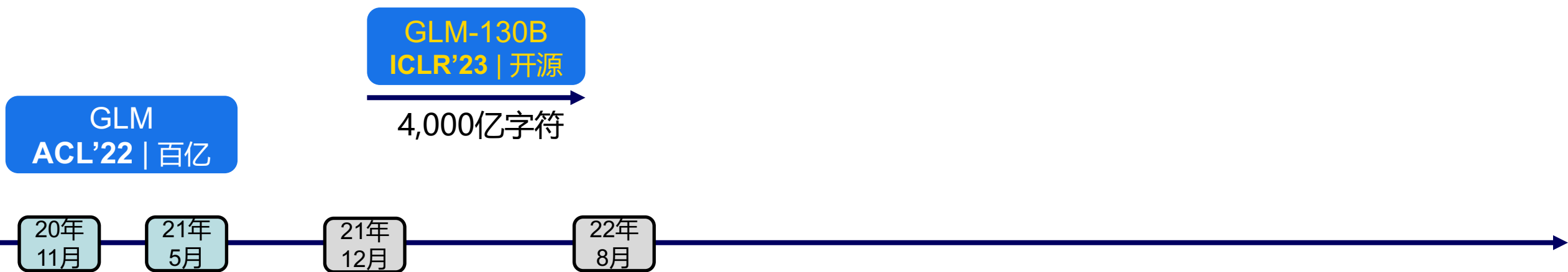
GLM-130B

- **多任务预训练**

- 从文本中间采样15%作为生成目标
- 从文本后面采样50-100%作为生成目标

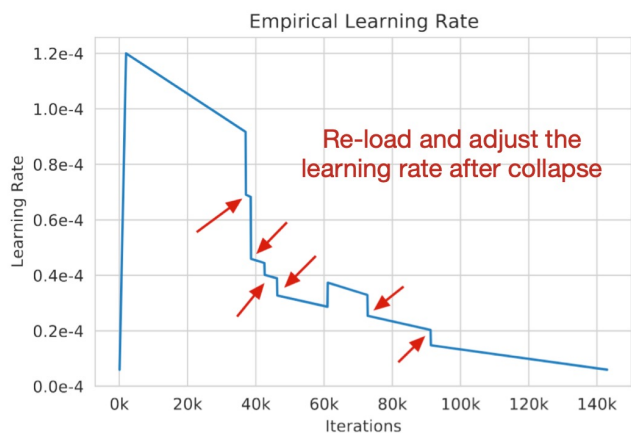


千亿基座 GLM-130B 到 ChatGLM 对话模型



大模型训练最大挑战：训练稳定性

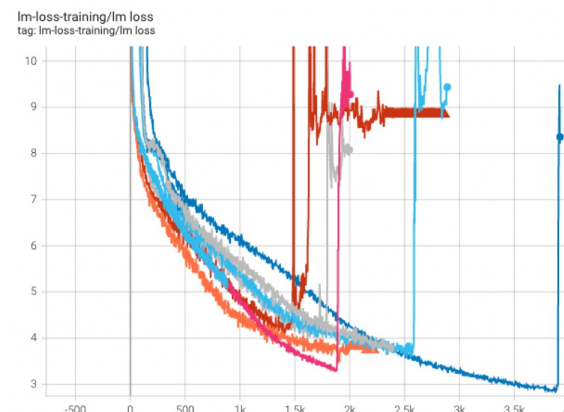
- 权衡利弊：训练稳定性（高精度低效）还是训练效率（低精度高效）
- 目前已开源训练过程大模型的解决方案
 - **FB OPT-175B**：训练崩溃时反复调整学习率/跳过数据（权宜之计，损失性能）
 - **HF BLOOM 176B**：embedding norm和BF16（损失性能，有限适配平台）



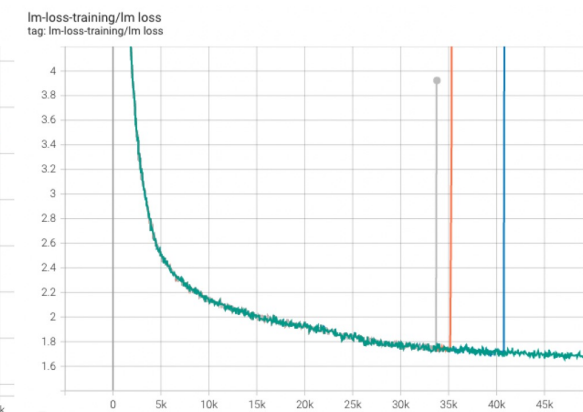
(a) OPT 175B's experiments



(b) BLOOM 176B's experiments



(c) GLM 130B's experiments



(c) GLM 130B's real training

GLM-130B: 稳定训练方法

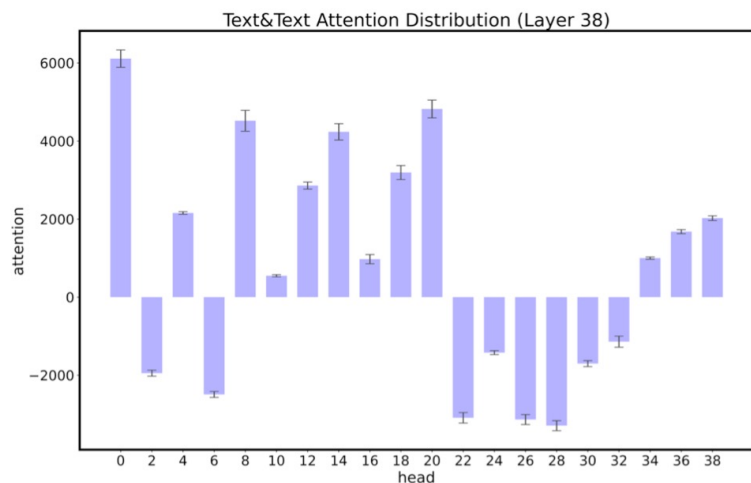
□ Attention score 层: Softmax in 32 避免上下溢出

$$\text{softmax}\left(\frac{Q_i K_i^\top}{\sqrt{d}}\right) = \text{softmax}\left(\left(\frac{Q_i K_i^\top}{\alpha\sqrt{d}} - \max\left(\frac{Q_i K_i^\top}{\alpha\sqrt{d}}\right)\right) \times \alpha\right) = \text{FP16}\left(\text{softmax}\left(\text{FP32}\left(\frac{Q_i K_i^\top}{\alpha\sqrt{d}}\right) \times \alpha\right)\right)$$

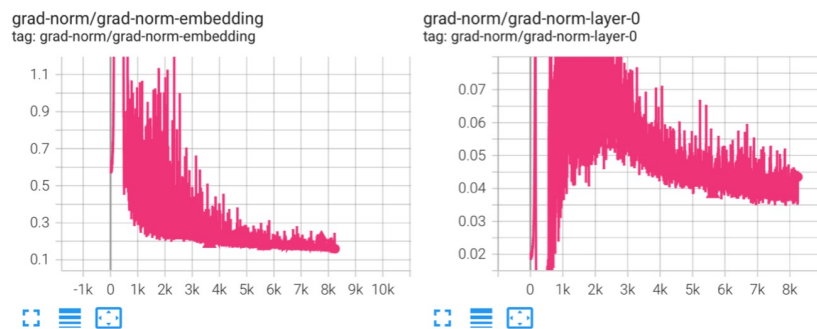
□ 调小 Embedding 层梯度, 缓解前期梯度爆炸问题

$$\text{word_embedding} = \text{word_embedding} * \text{alpha} + \text{word_embedding}.\text{detach}() * (1 - \text{alpha})$$

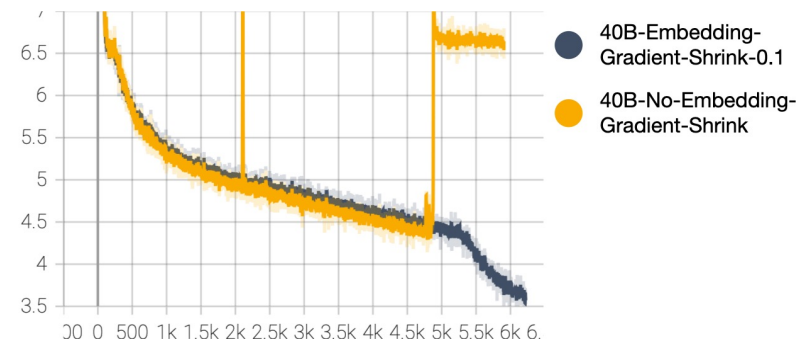
Attention 层的分数分布很容易超过 FP16 表示范围



Embedding 层梯度存在数量级上的差异, 大模型测试上有效稳定训练

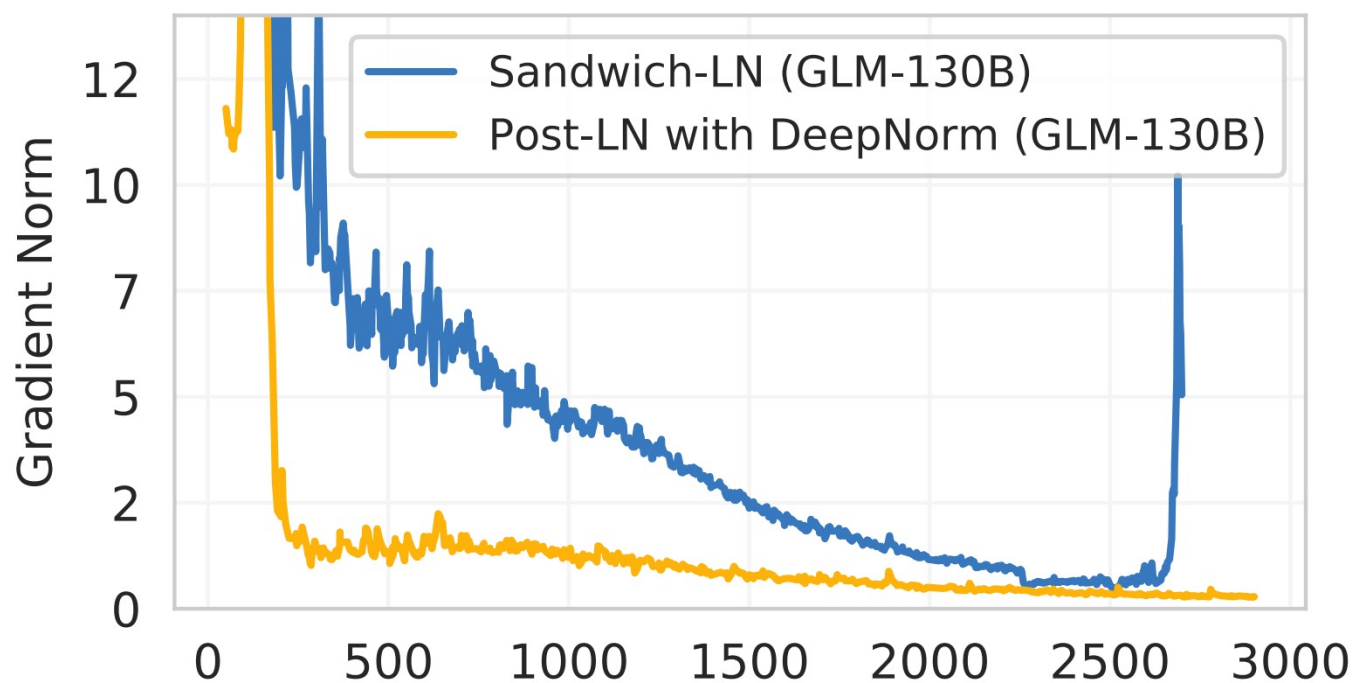
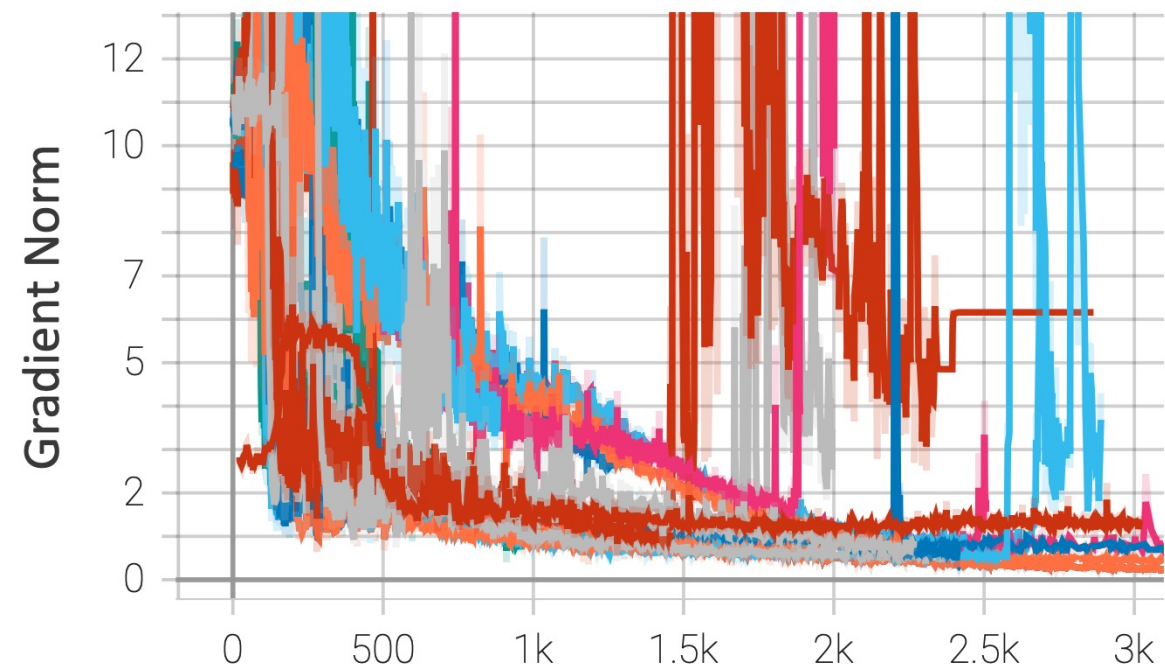


(a) Gradient norm of embedding layer (left) and the first layer (right)



(b) Training loss curves of GLM-40B with and without gradient shrink

GLM-130B: 稳定训练方法



(a) More than 30 failed preliminary trials at 100B-scale

(b) Final decisive trials: Sandwich-LN v.s. DeepNorm

GLM-130B : 大量实验确定最优架构

- DeepNorm : 稳定训练 1000 层 Post-LN 的方法

$$\text{DeepNorm}(x) = \text{LayerNorm}(\alpha \cdot x + g(x)), \quad \alpha > 1$$

- 旋转位置编码(RoPE) : 适用于 GLM 的相对位置编码

$$(\mathbf{R}_m q)^\top (\mathbf{R}_n k) = q^\top \mathbf{R}_m^\top \mathbf{R}_n k = q^\top \mathbf{R}_{n-m} k$$

- 门控注意单元(GLU) : FFN 层的替换 , 稳定提升模型性能

$$\text{FFN}_{\text{GLU}}(x, \mathbf{W}, \mathbf{V}, \mathbf{W}_2) = (\sigma(x\mathbf{W}_1) \otimes x\mathbf{V}) \mathbf{W}_2$$

	RTE	COPA	BoolQ	WSC	Avg
glm-base (original paper)	71.2	71.0	77.0	72.1	72.825
glm-base-geglu-postln	72.80±1.04	74.00±0.82	76.34±0.15	74.36±0.91	74.375
glm-base-geglu-deepnorm	70.16±0.85	78.33±0.47	76.93±0.14	71.79±0.91	74.3025
glm-base-geglu-preln	71.24±1.04	75.33±2.49	76.75±0.17	80.77±2.72	76.0225
glm-base-geglu-sandwich	71.00±0.61	77.00±1.63	77.24±0.43	78.21±1.81	75.8625
glm-base-gau-postln	69.43±1.70	71.33±1.25	76.33±0.24	75.00±1.57	73.0225
glm-base-gau-preln	diverged				
glm-base-gau-preln-0.1-shrink	71.84±1.06	75.33±1.89	75.80±0.34	76.28±3.27	74.8125
glm-base-gau-sandwich	69.92±0.61	75.67±0.94	77.00±0.15	72.44±1.81	73.7575
glm-base-sandwich	71.00±0.74	72.33±1.70	76.75±0.05	73.72±2.40	73.45

小规模试验选取最优架构

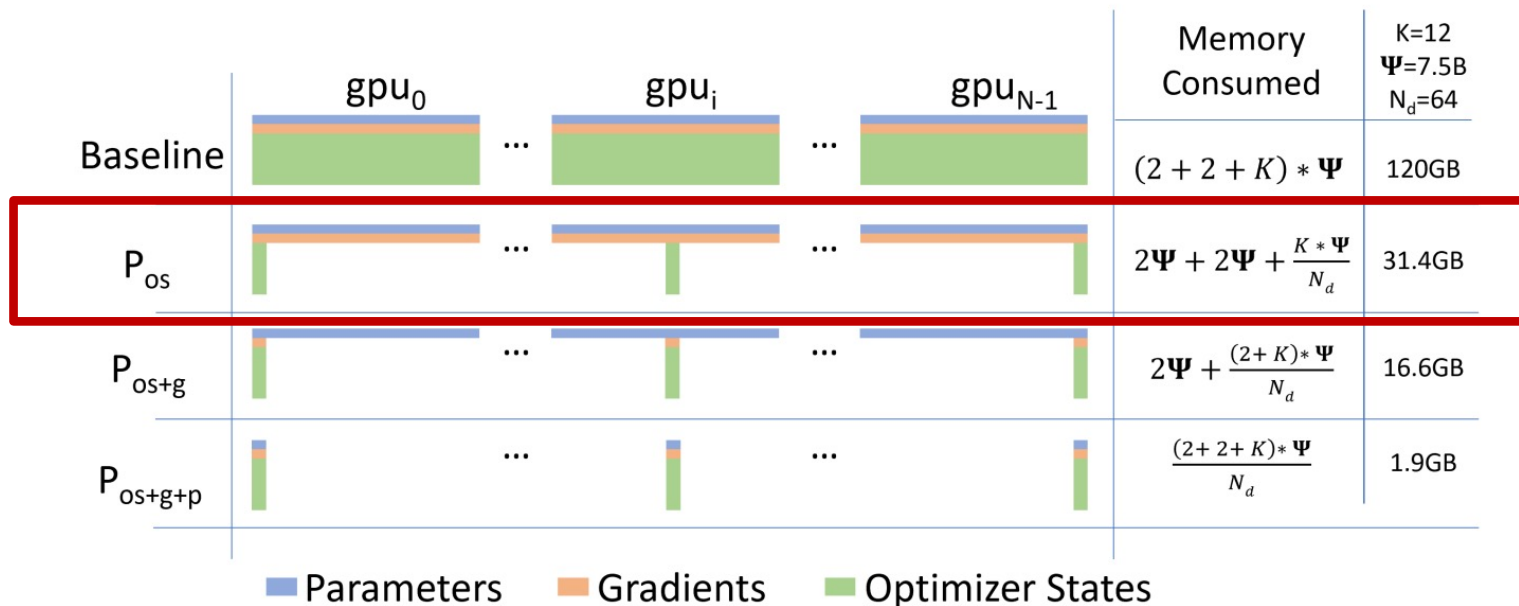
并行策略：高效训练千亿模型

□ 存下 GPT-3 模型需要 2.8T 显存存放训练状态 + 中间激活函数值

$$\underbrace{175\text{B}}_{\text{GPT-3 param.}} \times \left(\underbrace{2}_{\text{weights}} + \underbrace{2}_{\text{gradients}} + \underbrace{4}_{\text{Master weights}} + \underbrace{4+4}_{\text{Adam states}} \right) = 2.8 \text{ TB}$$

□ 挑战：远超单卡显存（40GB），采取何种并行方式高效训练？

1. 采用 ZeRO 优化器在数据并行组内分摊优化器状态 → ~25%



并行策略：高效训练千亿模型

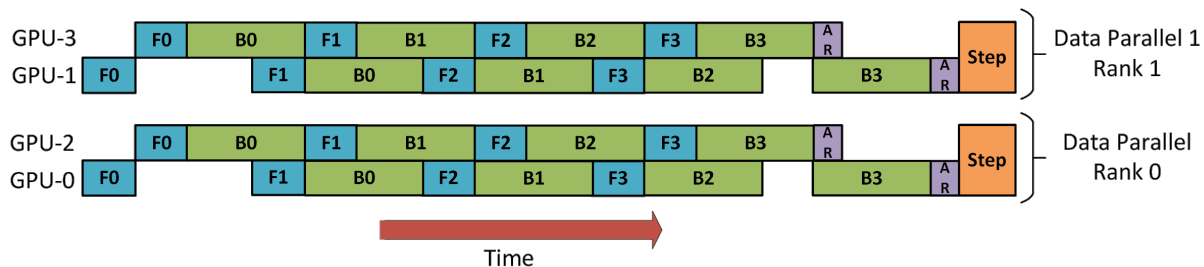
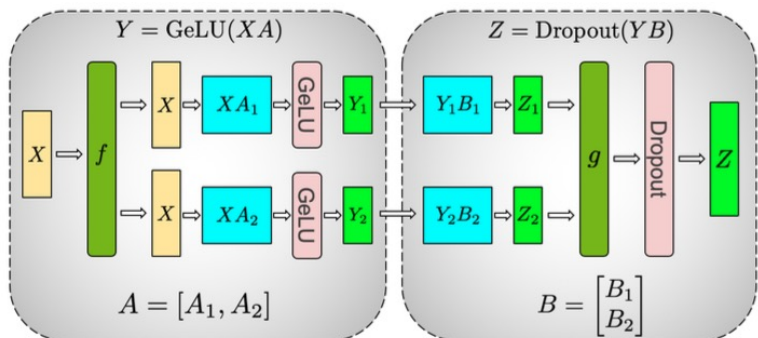
□ 远超单卡显存，如何高效训练？

2. 模型并行：将模型参数分布到多个 GPU 上

- 张量并行：切分参数矩阵，每 GPU 计算一部分 → 额外通信，降低计算粒度
- 流水线并行：将网络分成多段并行 → 引入流水线气泡
- ZeRO-3：将参数分布到数据并行组中，算之前先取回参数 → 额外通信时间

□ 分析：流水线的气泡占比： $\frac{n/t - 1}{m + n/t - 1}$ ， $n/t \ll 4m$ 的时候可以忽略不计

□ 并行策略：张量并行随着模型规模增大缓慢扩展，但不超过单机规模（ ≤ 8 ），其余全部使用流水线并行，通过调整微批处理大小减少气泡占比



张量模型并行

流水线并行

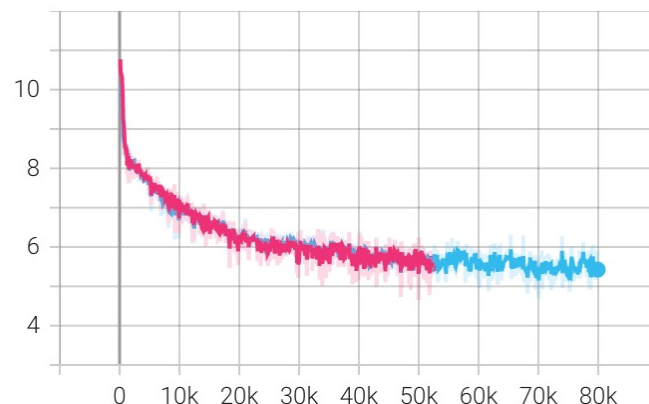
并行策略：高效训练千亿模型

□ 其他优化

- 算子融合：融合多个 element-wise 算子 → 提升 ~10% 计算速度
- 流水线平衡：流水线首尾阶段各少放置一个层平衡占用 → 节省 ~10% 显存
- 跨平台兼容：swDeepSpeed 训练库 ↔ 与 DeepSpeed API 兼容
 - 支持申威架构，一行代码无缝替换兼容
 - 实现并行通信策略，混合精度策略，ZeRO 优化器
 - 同一套训练框架可在三个集群上对齐训练曲线

```
import swDeepSpeed as deepspeed
model, optimizer, _ = deepspeed.initialize(
    model=model,
    model_parameters=param_groups,
    args=args,
    mpu=mpu,
    dist_init_required=False,
    config_params=config_params
)
```

Train/train_loss
tag: Train/train_loss



并行策略：高效训练千亿模型

测试集群配置：

硬件差异性大

- A100 集群 (A100) : 96 台 DGX-A100 , 每台 2 张 200GB IB 网卡
- 海光GPU (Hygon) : 3000 台机器 , 每台 4 张 DCU 加速卡、4 张 50G IB 网卡
- 申威处理器 (Sunway) : 8192 个节点 , 每节点一块 SW26010-PRO 处理器

训练 GPT-3 175B 规模的模型 , 按照相同的 300B 单词量估计训练时间：

集群名称	模型规模	机器数	TP ^a	PP ^b	层数	批处理	利用率	预计时间 ^c
A100	176B	96	4	14	96	3584	41.06%	50 天
Hygon	183B	3000	4	25	100	6000	31.29%	64 天
Sunway	173B	8192	4	6	94	4096	18.39%	59 天
BMTrain-A100	176B	96	-	-	-	-	33.00%	62 天

利用率较高

训练时间合理

有性能优势

^a 张量模型并行规模 ^b 流水线模型并行规模

^c 训练 300B = 300 × 10⁹ 单词量所需要的时间估计

GLM-130B: 千亿模型之旅

Major Issues Encountered for Training GLM-130B

2021.12

- The “千亿” (100B) project towards an open dense pre-trained GLM at 100B scale is conceived
- Survey pre-training strategies of existing models of similar scale, such as GPT-3, Gopher => [Limited public info about how they were trained and issues they met](#)
- Search for possible GPU clusters & sponsors

2022.1

- Test the performance of FP16/FP32 at 100B scale on one testing cluster
- Unexpected excessive memory usage in GLM => [Torch is better with fixed length input sequences](#)
- Inability to converge and try tricks from CogView and ViT => [Use Sandwich-LN](#)
- Frequent random hardware failures => [Have to run HCPG test before each run](#)

2022.2

- Very slow training speed than previously calculated => [Optimize kernels and fuse operators](#) => [Find the input shape is critical to kernel performance](#)
- Collect pre-training corpora and tokenize => [Use icetk: the sentence piece is set to the unigram mode](#)
- Debug the 3D pipeline parallel in the newly-released Megatron and DeepSpeed

2022.3

- It can't recover perfectly from optimizer states => [Our customized dataloaders do not save its state seed properly in distributed training](#)
- The memory per processor is too small => [Require too many pipeline stages](#) => [Batch size is too large \(up to 12,000\)](#) => [Harm the model's convergency](#)
- It can't launch more than 2,000 computing nodes => [Overcome this and support 6,000-node training by tuning Linux kernel TCP parameters](#)
- Collect data for multi-task instruction pre-training
- Receive opportunities to test trainings on several other clusters
- Very slow training speed than expected => [The underlying element-wise operators don't support fast computation on large-dimension vectors.](#)

2022.4

- Optimize A100 kernel's computing efficiency => [A100 kernels prefer square-shaped inputs, and seq_len=2,048 is optimal for our hidden-state dimension \(12,288\)](#)
- Inability to converge due to large gradient norms (170+) of input embeddings => [Try embedding norm and gradient shrink, which turn out to be almost equivalent](#)
- Naïve post-LN or pre-LN disconverges after several thousands of steps => [Try Sandwich-LN with PB-Relax](#)
- It still disconverges after one week's trial => [The dataloader state seeds are not unified for different pipeline stages, resulting in a mismatch of input data and labels.](#)
- Test two positional encodings: RoPE and Alibi => [Alibi can be slower as it requires element-wise manipulation on attention matrices---changing num_heads * 2,048 * 2,048 scalars per layer](#)
- Test GeGLU and GAU => [GAU converges faster with relatively poor performance on fine-tuned SuperGLUE](#)
- Abnormal GPU memory usage of newly-added functions and classes => [DeepSpeed hardcodes the function names for checkpoint activation](#)
- Decode to train GLM with 130 billion parameters => [allow inference on a DGX-A100 40G node](#)

2022.5-6

- Implement a RoPE cuda operator in C++ => [See unexpected precision errors and finally have it abandoned](#)
- Sandwich-LN still disconverges => 1) [Reducing learning rate does not help](#); 2) [Using Hinge cross-entropy becomes slower and harms performance](#); 3) [Shifting to DeepNorm still disconverges](#)
- Use FP32 in softmax of attention => [Success](#)
- Find PB-Relax unnecessary for FP32 softmax => [It also slows down training as it needs to manipulate the whole attention score matrices](#)
- Experience few spikes in later training => 1) [Reduce gradient shrink factor from 1 to 0.1: useful](#); 2) [Reduce the learning rate: sometimes useful](#); 3) [Jump the noisy data batches: sometimes useful](#)
- Find a mistake in multi-task data after training for 20,000 steps => [Use the correct data but it does not forget](#)

2022.6-7

- Adapt the pipeline parallel checkpoints to ordinary parallel checkpoints for efficient inference on a single A100
- Work on evaluation scripts on datasets: MMLU, Big-bench, CLUE, SuperCLUE, etc.
- Implement P-Tuning and P-Tuning v2 for parameter-efficient tuning on GLM-130B for tuning on SuperGLUE
- Work with BMInf on adapting GLM-130B to perform inference on a single V100 or 3090 => [Use pipeline-style asynchronous swapping between main memory and GPU memory](#)
- Try to fine-tune GLM-130B with fewer A100 nodes (i.e., 12-16 nodes) => [Pipeline-style fails due to too many pipeline stages](#) => [Find that data parallel can not be introduced for fine-tuning](#) => [Use 32-way model parallel for fine-tuning with reasonable performance](#)

GLM-130B：千亿模型之旅

□ 8 个月时间，从零开始解决无数的难题：

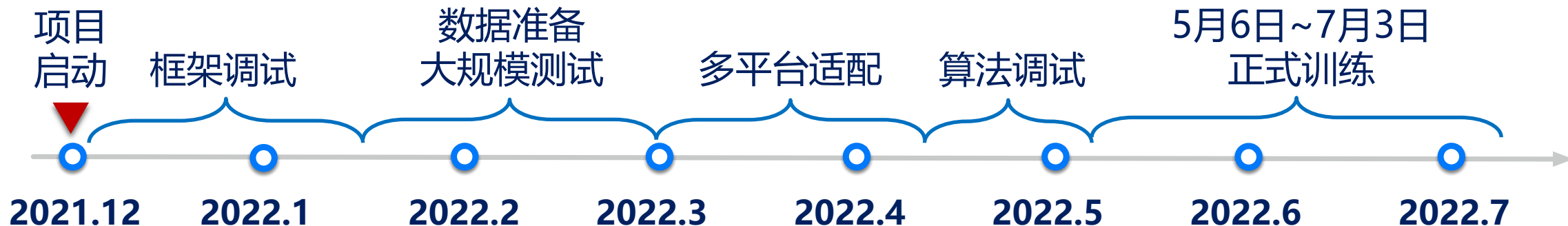
□ **算法难题：** 千亿模型的混合精度训练非常不稳定，且调试困难

□ 模型梯度爆炸，Embedding 层的过大梯度，Post-LN、Pre-LN 和 Sandwich-LN 的不稳定性，DataLoader 状态种子恢复，以及 Softmax 和 Attention 的计算精度选择.....

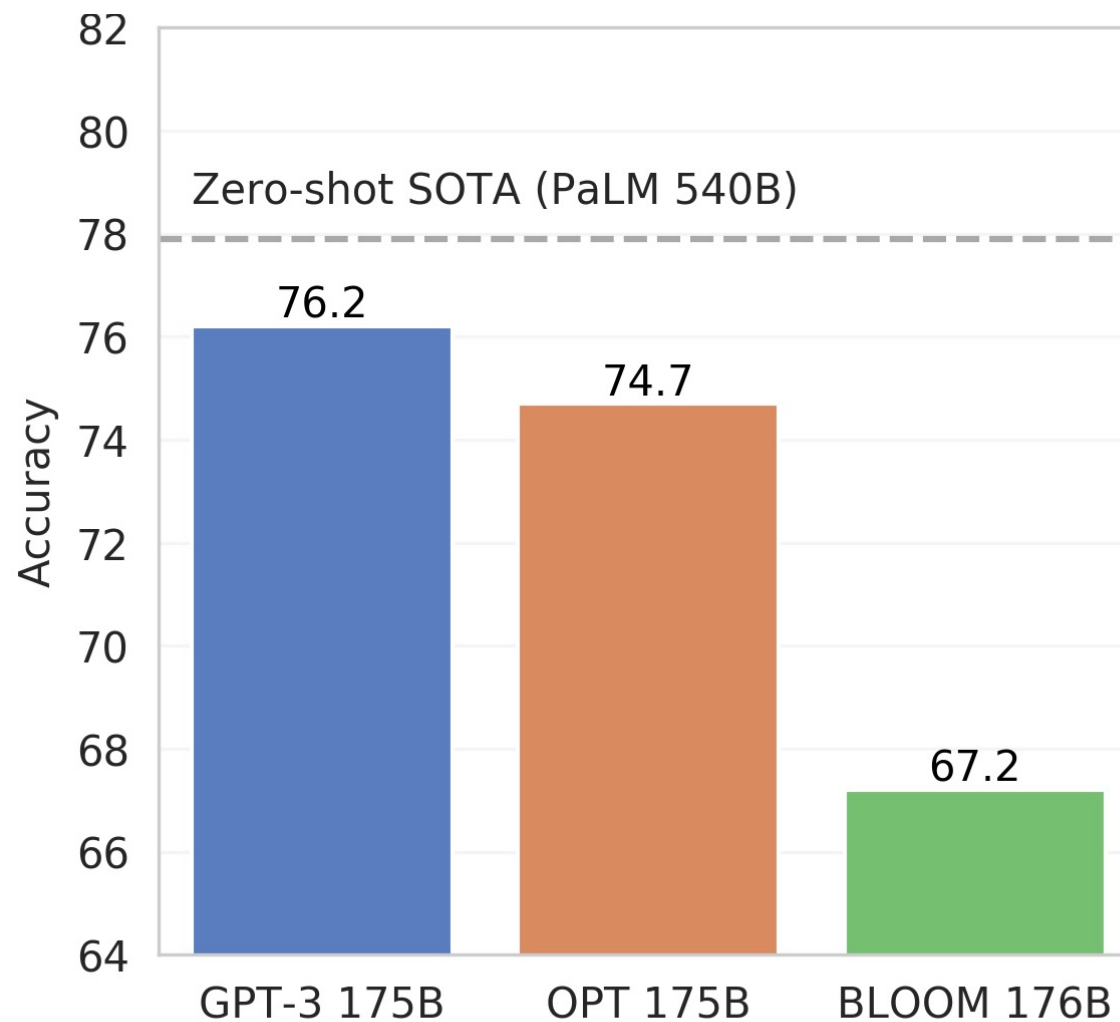
□ **工程难题：** 不同架构集群上高效训练千亿模型是极大的挑战

□ 海光集群、A100 集群、神威超算、昇腾 910 集群

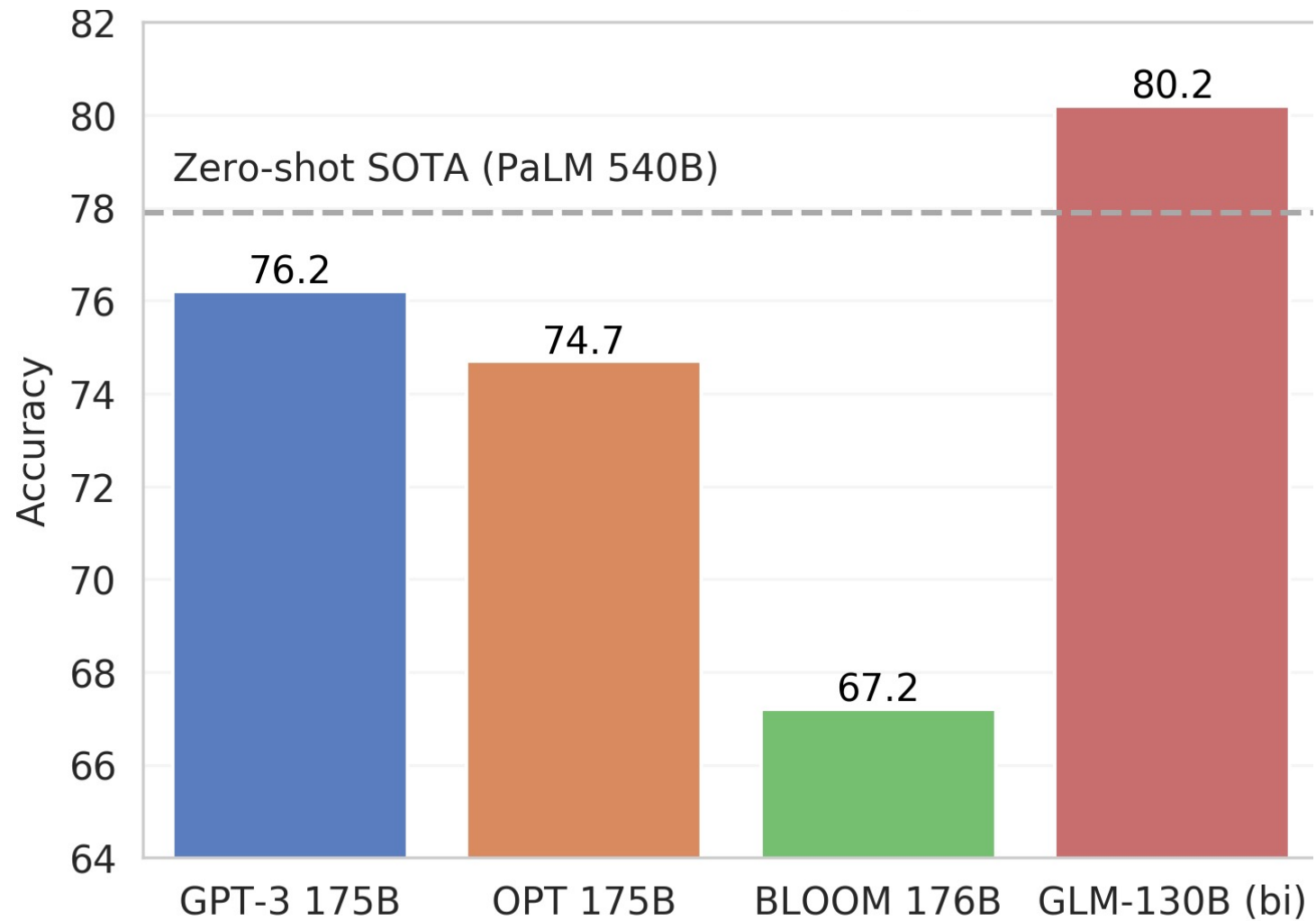
□ 频繁的随机硬件故障、Megatron-DeepSpeed 中的 3D 流水线平衡调试、CUDA kernel 效率调优、训练显存紧张、数万进程 TCP 初始化、Element-wise 算子融合



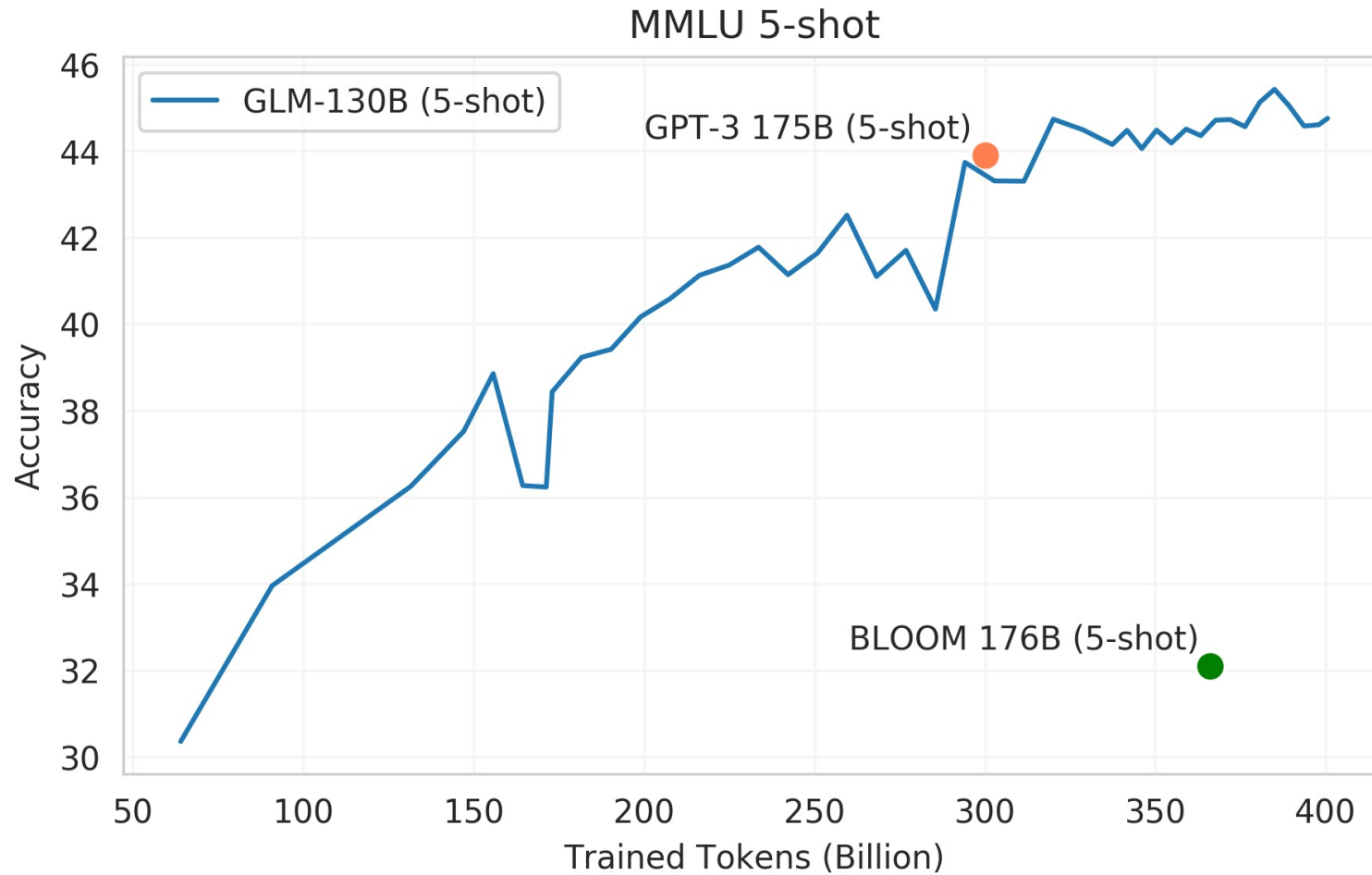
LAMBADA (English)



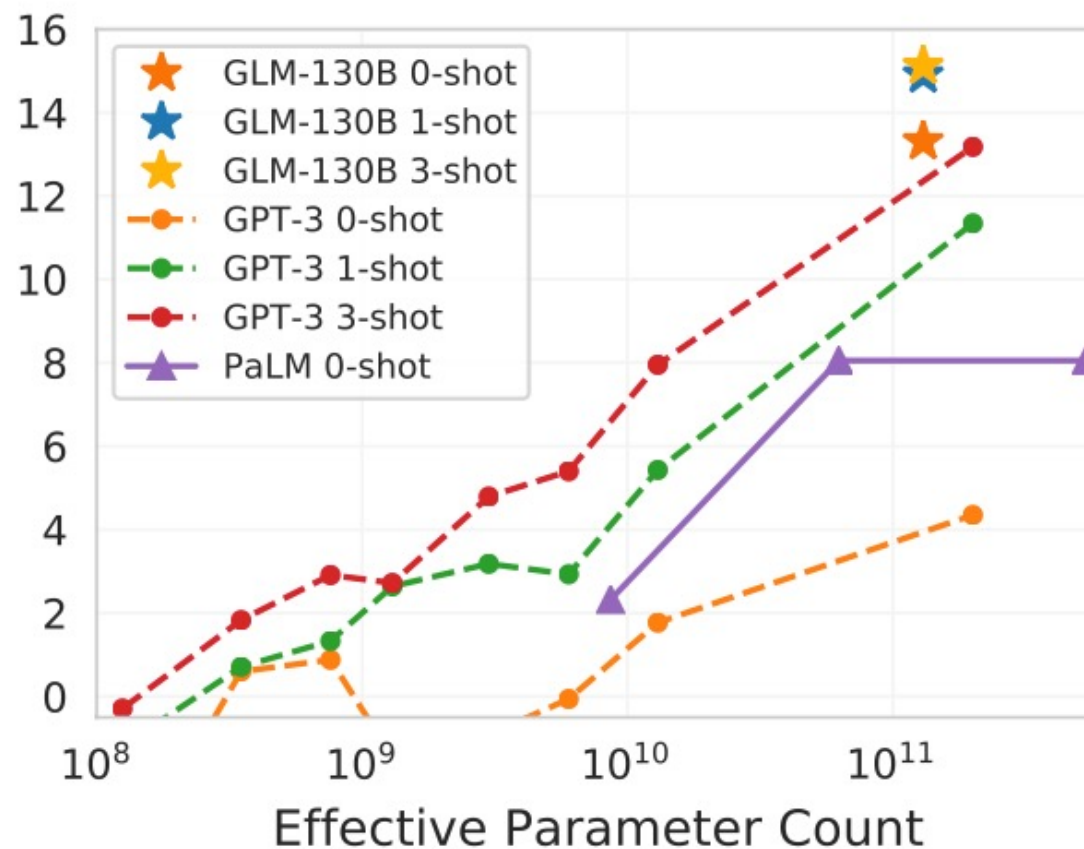
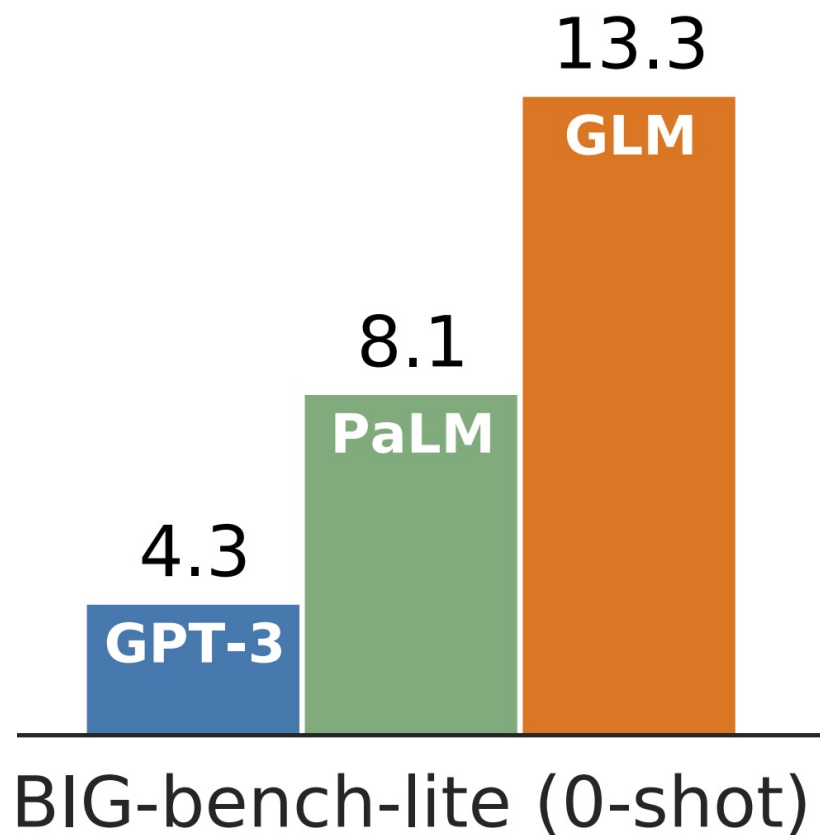
LAMBADA (English)



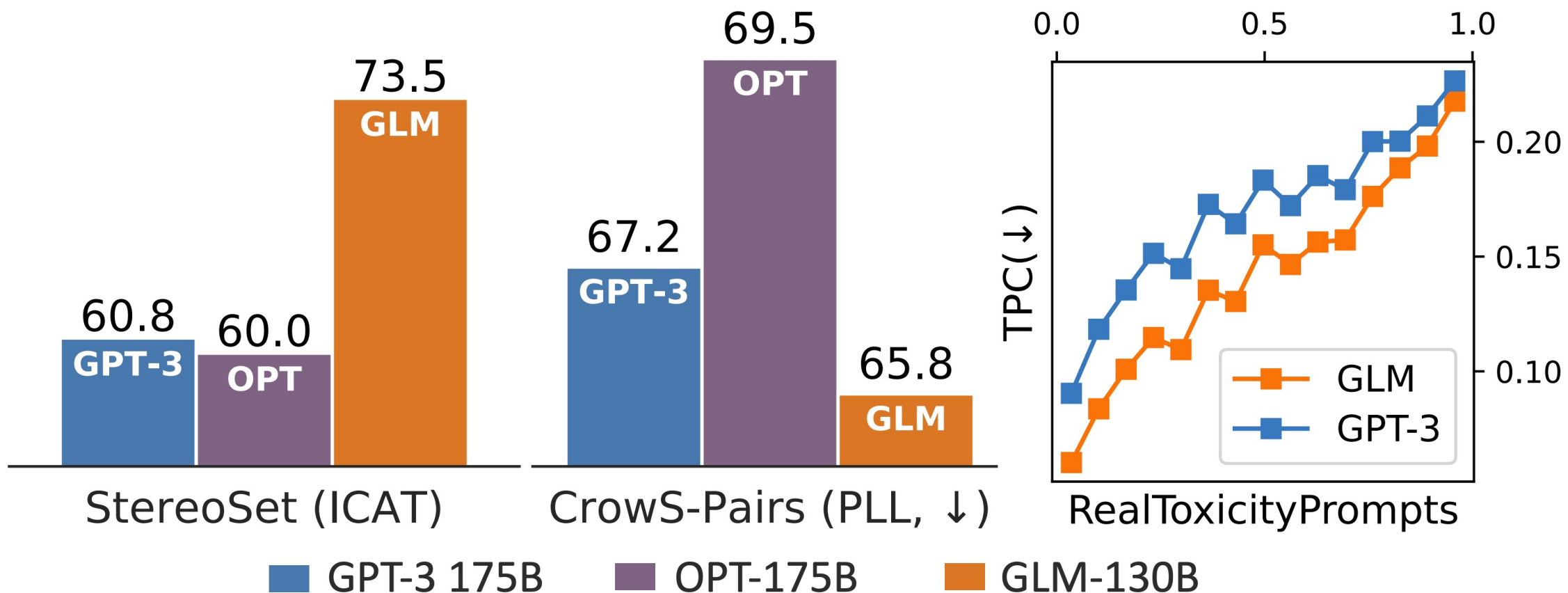
Massive Multitask NLU (MMLU)



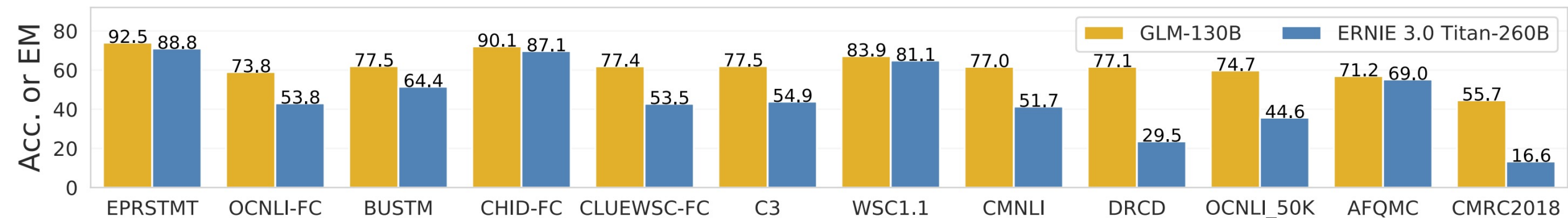
BIG-bench-lite (English)



Bias & Toxicity (English)

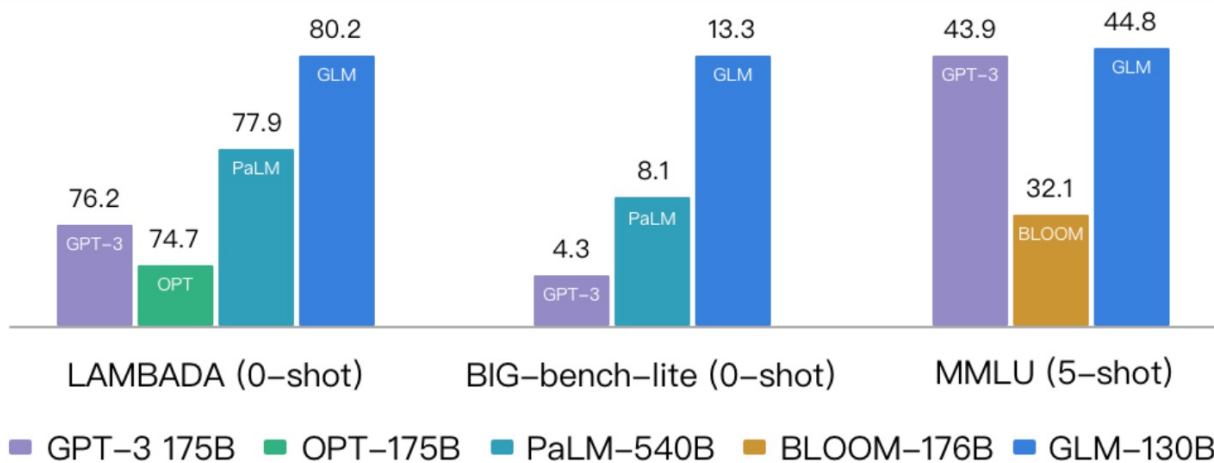


CLUE & FewCLUE (Chinese)



千亿基座GLM-130B

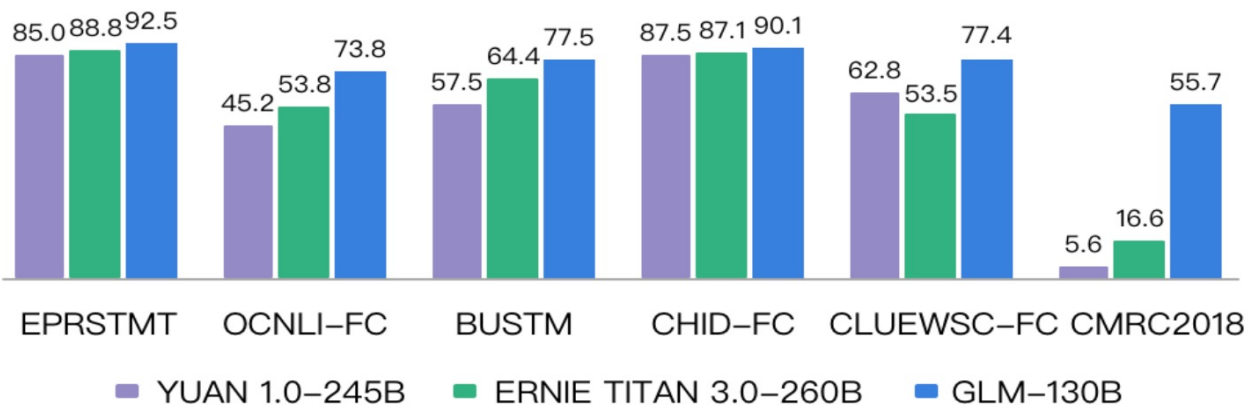
英文：在 MMLU、LAMBADA、BIG-bench-lite 等
超过GPT-3, OPT, PaLM



2022.8---2023.3，收到60个国家
1000+研究机构的下载使用需求

- Google
- Microsoft
- Facebook
- AI2
- Stanford
- MIT
- UC Berkely
- CMU
- Harvard
- Princeton
- Yale
- Cornell
- Columbia
- UIUC
- Cambridge
- Oxford
- 华为
- 阿里巴巴
- 腾讯
- 百度
- 美团
- 头条
- 滴滴
- 智源
- 小冰
- 小度
- 小米
- 小鹏
- 有道
- 旷视
- 平安
- 建设银行
- 北京大学
- 浙江大学
- 上海交大
- 复旦大学
- 中科院大学
- 中科大
- 武汉大学
- 华科
- 南开
- 香港大学
- 香港中文大学
- 香港科技大学
- 中科院多所
- 之江实验室
- 上海 AI 实验室
- 北京智源

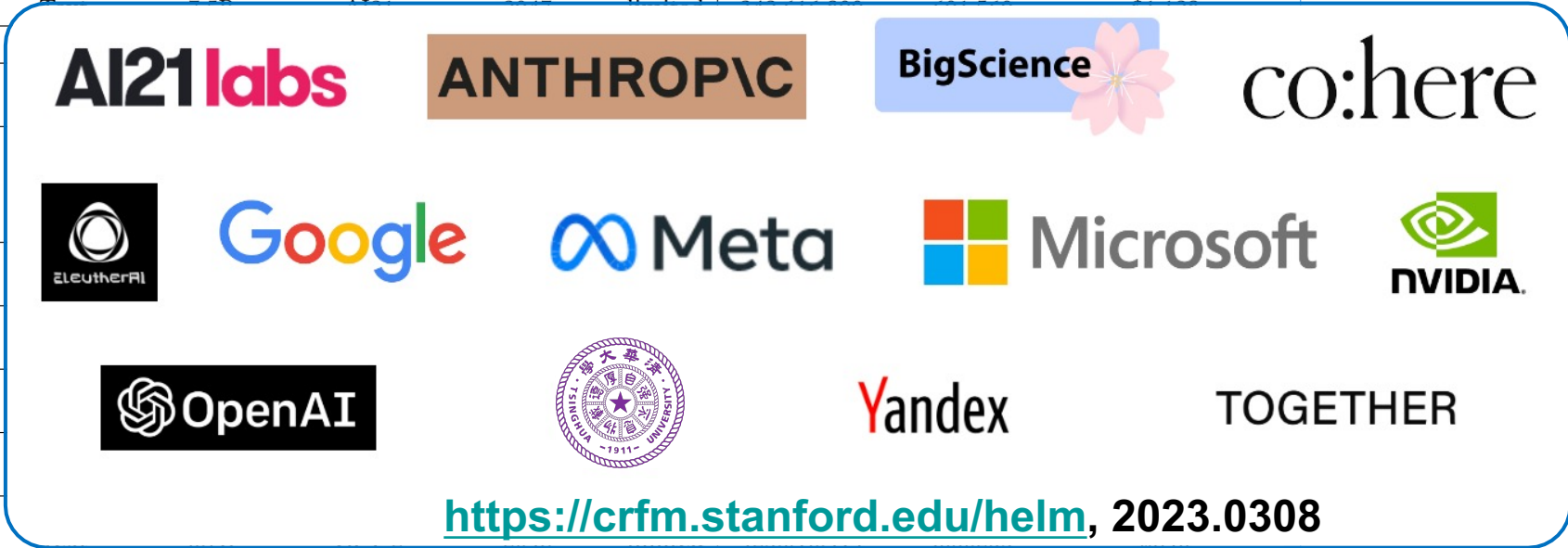
中文：在 CLUE 和 FewCLUE 上取得不错的结果



千亿基座GLM-130B

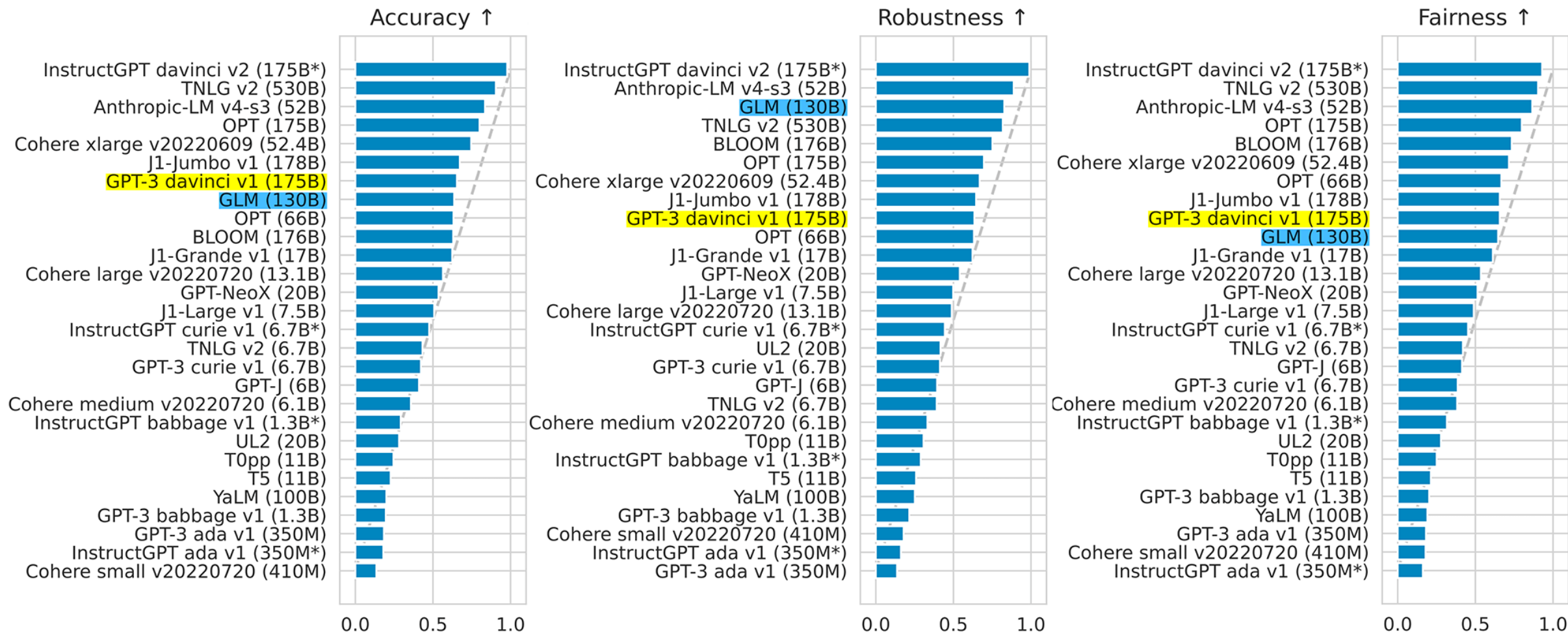
Stanford报告世界主流大模型评测：亚洲唯一入选模型

Model	Model Creator	Modality	# Parameters	Tokenizer	Window Size	Access	Total Tokens	Total Queries	Total Cost
J1-Jumbo v1 (178B)	AI21 Labs	Text	178B	AI21	2047	limited	327,443,515	591,384	\$10,926
J1-Grande v1 (17B)	AI21 Labs	Text	17B	AI21	2047	limited	326,815,150	591,384	\$2,973
J1-Large v1 (7.5B)	AI21 Labs								
Anthropic-LM v4-s3 (52B)	Anthropic								
BLOOM (176B)	BigScience								
T0++ (11B)	BigScience								
Cohere xlarge v20220609 (52.4B)	Cohere								
Cohere large v20220720 (13.1B) ⁵⁸	Cohere								
Cohere medium v20220720 (6.1B)	Cohere								
Cohere small v20220720 (410M) ⁵⁹	Cohere								
GPT-J (6B)	EleutherAI								
GPT-NeoX (20B)	EleutherAI								
T5 (11B)	Google								
UL2 (20B)	Google								
OPT (66B)	Meta								
OPT (175B)	Meta								
TNLG v2 (6.7B)	Microsoft/NVIDIA								
TNLG v2 (530B)	Microsoft/NVIDIA								
GPT-3 davinci v1 (175B)	OpenAI								
GPT-3 curie v1 (6.7B)	OpenAI								
GPT-3 babbage v1 (1.3B)	OpenAI	Text	1.3B	GPT-2	2048	limited	422,123,900	606,253	\$211
GPT-3 ada v1 (350M)	OpenAI	Text	350M	GPT-2	2048	limited	422,635,705	604,253	\$169
InstructGPT davinci v2 (175B*)	OpenAI	Text	175B*	GPT-2	4000	limited	466,872,228	599,815	\$9,337
InstructGPT curie v1 (6.7B*)	OpenAI	Text	6.7B*	GPT-2	2048	limited	420,004,477	606,253	\$840
InstructGPT babbage v1 (1.3B*)	OpenAI	Text	1.3B*	GPT-2	2048	limited	419,036,038	604,253	\$210
InstructGPT ada v1 (350M*)	OpenAI	Text	350M*	GPT-2	2048	limited	418,915,281	604,253	\$168
Codex davinci v2	OpenAI	Code	Unknown	GPT-2	4000	limited	46,272,590	57,051	\$925
Codex cushman v1	OpenAI	Code	Unknown	GPT-2	2048	limited	42,659,399	59,751	\$85
GLM (130B)	Tsinghua University	Text	130B	ICE	2048	open	375,474,243	406,072	2,100 GPU hours
YaLM (100B)	Yandex	Text	100B	Yandex	2048	open	378,607,292	405,093	2,200 GPU hours



千亿基座GLM-130B

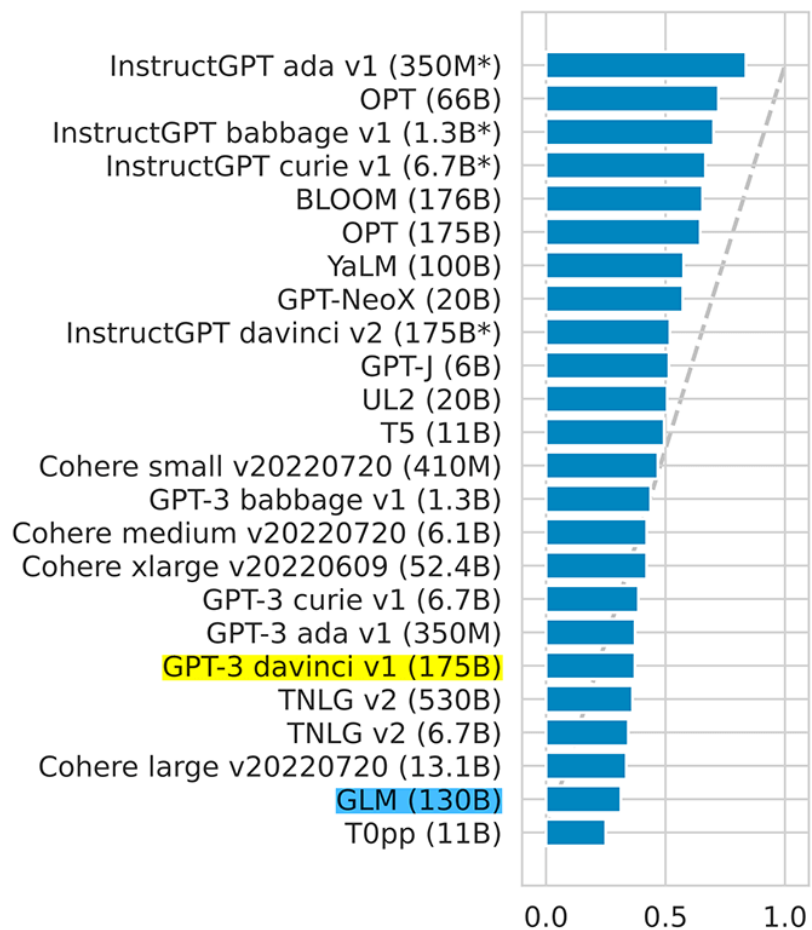
准确性、恶意性等与GPT-3接近或持平，鲁棒性和校准误差在千亿模型中表现最佳



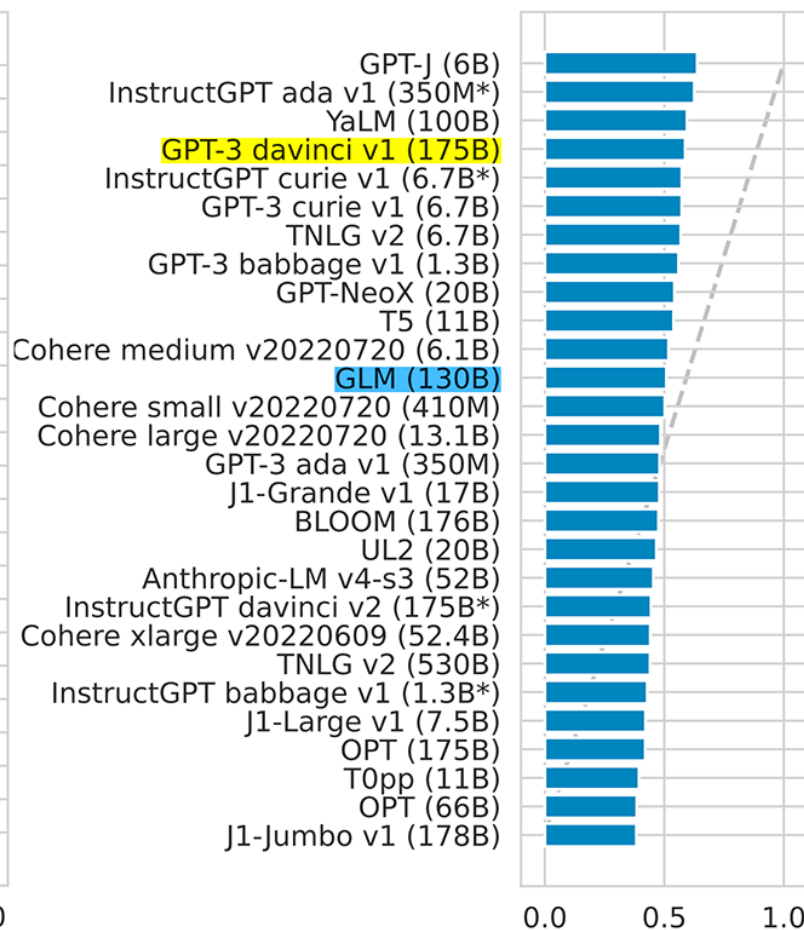
千亿基座GLM-130B

准确性、恶意性等与GPT-3接近或持平，鲁棒性和校准误差在千亿模型中表现最佳

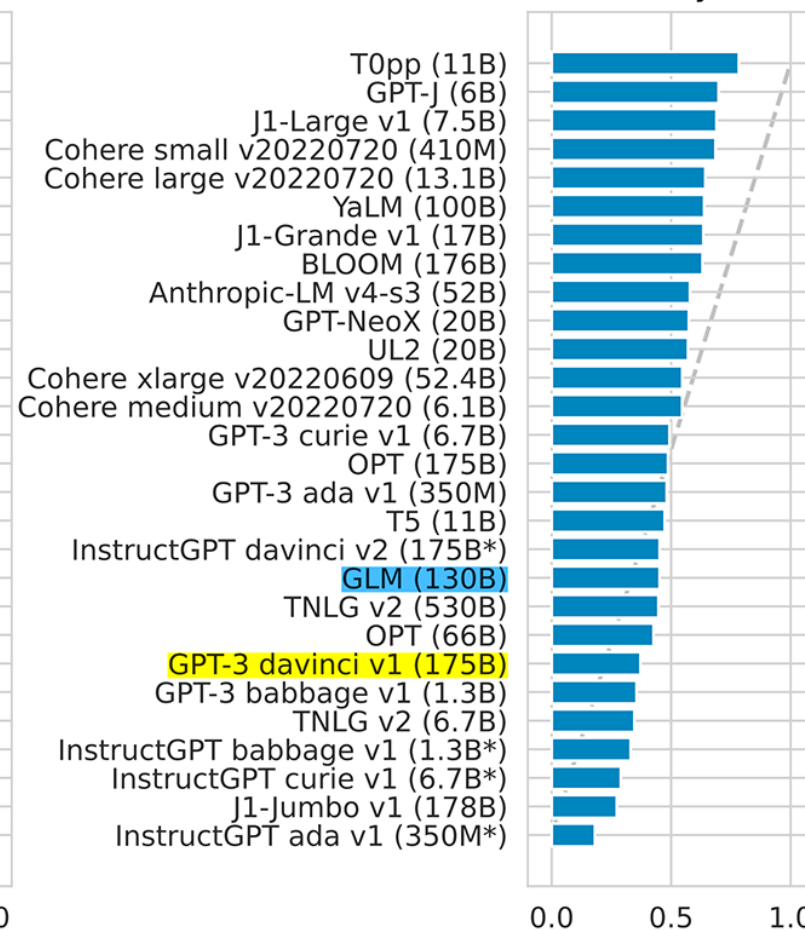
Calibration error ↓



Bias ↓



Toxicity ↓



千亿基座GLM-130B：量化

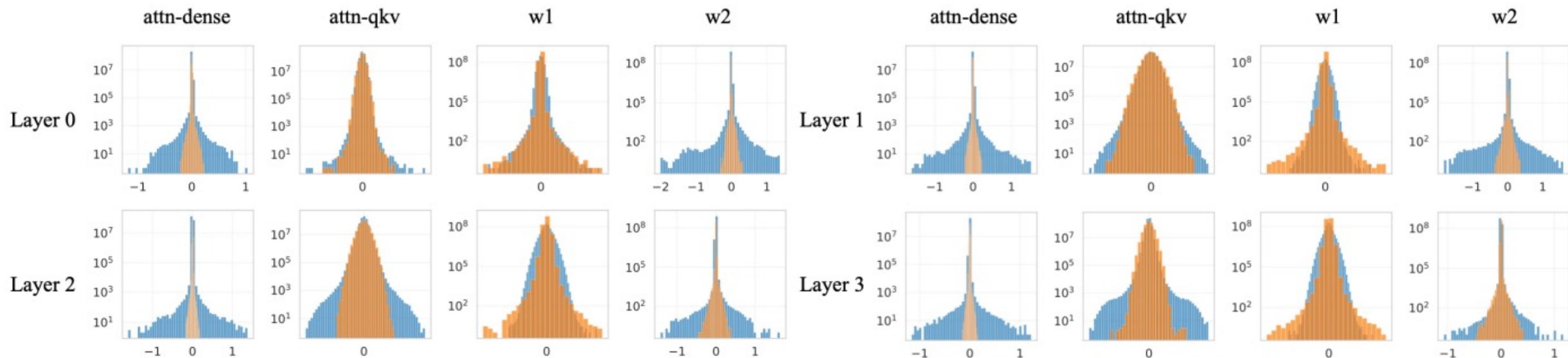
□ GLM-130B的权重INT4量化“规模效应” **(显存降75%)**

□ 观察Transformer线性层的权重数值分布

□ **蓝色**为BLOOM-176B, **橙色**为GLM-130B

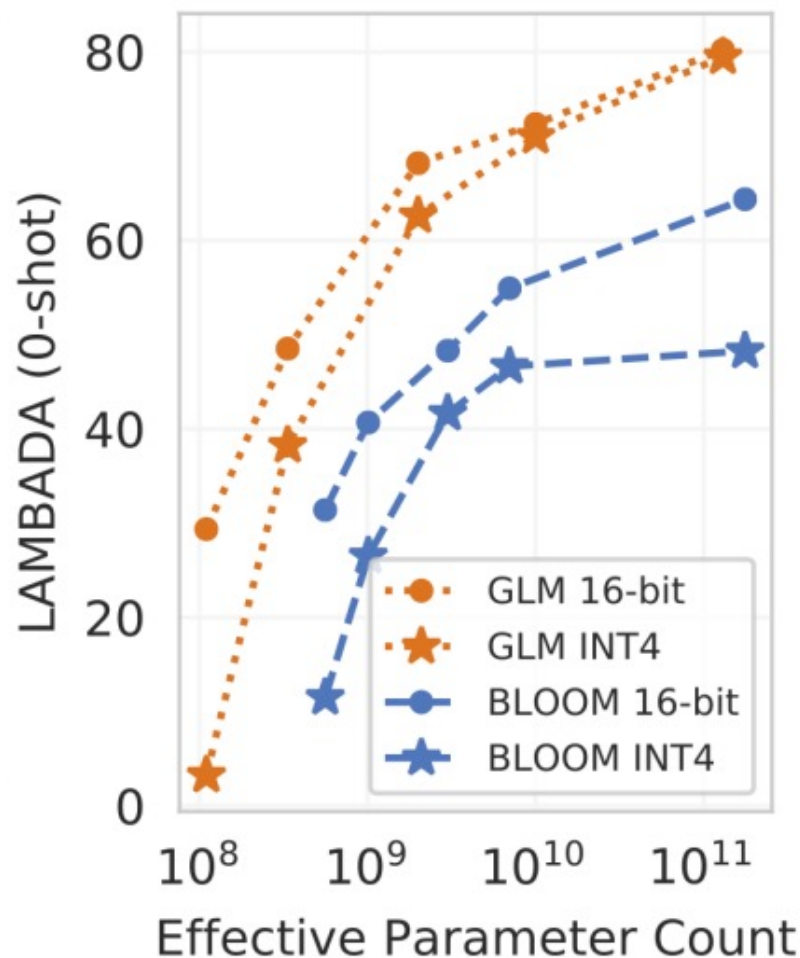
□ GLM-130B在attn-dense和ffn-w2具有更窄的数值分布

□ 由于GLM本身的训练目标造成, 而非Transformer的架构选择



千亿基座GLM-130B

GLM's INT4 Weight
Quantization Scaling Law



Model Precision	GLM-130B			GPT-3
	FP16	INT8	INT4	FP16
MMLU (acc, ↑)	44.75	44.71	44.80	43.9
LAMBADA (acc, ↑)	80.21	80.21	79.47	76.2
Pile (a part, BPB, ↓)	0.634	0.638	0.641	0.74

GPU Type	128 Enc./Dec.		512 Enc./Dec.	
8 × A100 (40G)	0.15s	4.29s	0.18s	17.7s
8 × V100 (32G)	0.31s	6.97s	0.67s	28.1s
4 × RTX 3090 (24G)	0.37s	8.16s	1.30s	32.3s
8 × RTX 2080 Ti (11G)	0.39s	6.77s	1.04s	27.3s

千亿基座GLM-130B

	基础架构	训练方式	量化	加速	跨平台能力
GPT3-175B	GPT	自监督预训练	—	—	NVIDIA
OPT-175B	GPT	自监督预训练	INT8	Megatron	NVIDIA
BLOOM-176B	GPT	自监督预训练	INT8	Megatron	NVIDIA
GLM-130B	GLM	自监督预训练和多任务预训练	<ul style="list-style-type: none">• INT8• INT4	Faster Transformer	<ul style="list-style-type: none">• NVIDIA• 海光 DCU• 昇腾910• 申威
对比优势	高精度： <ul style="list-style-type: none">• Big-bench-lite: +5.2%• LAMBADA: +2.3%• CLUE: +24.3%• FewCLUE: +12.8%		普惠推理： 节省75%内存 可单台3090 (4) 或单台2080Ti (8) 进行 无损 推理	高速推理： 比Pytorch 提速7-8.4倍 Megatron 提速2.5倍	跨平台： 支持更多不同的 大规模语言模型 的适配

为什么
千亿（100B）模型？

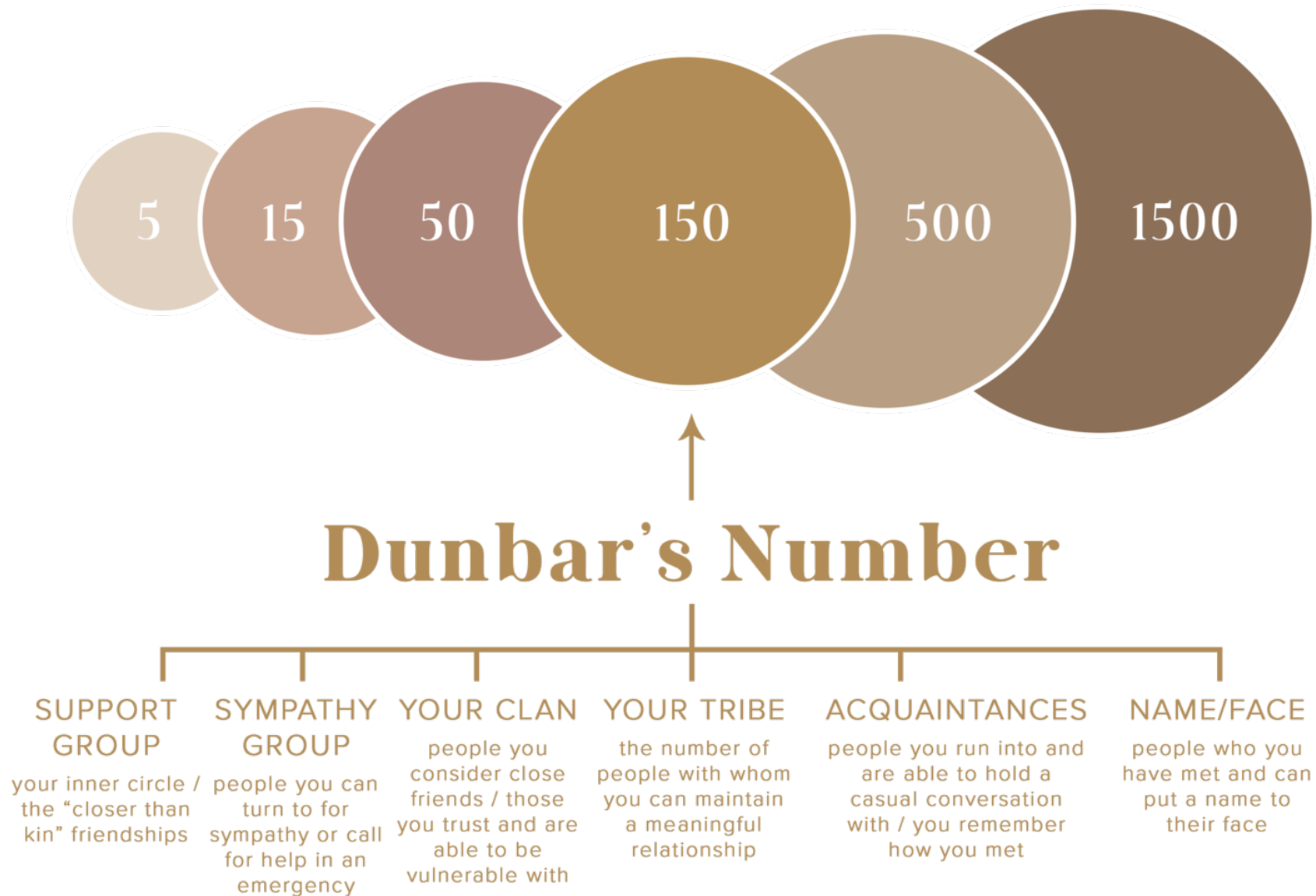
Dunbar's Number

How Many Connections Do/Can You Maintain?

SUPPORT GROUP	SYMPATHY GROUP	YOUR CLAN	YOUR TRIBE	ACQUAINTANCES	NAME/FACE
your inner circle / the "closer than kin" friendships	people you can turn to for sympathy or call for help in an emergency	people you consider close friends / those you trust and are able to be vulnerable with	the number of people with whom you can maintain a meaningful relationship	people you run into and are able to hold a casual conversation with / you remember how you met	people who you have met and can put a name to their face



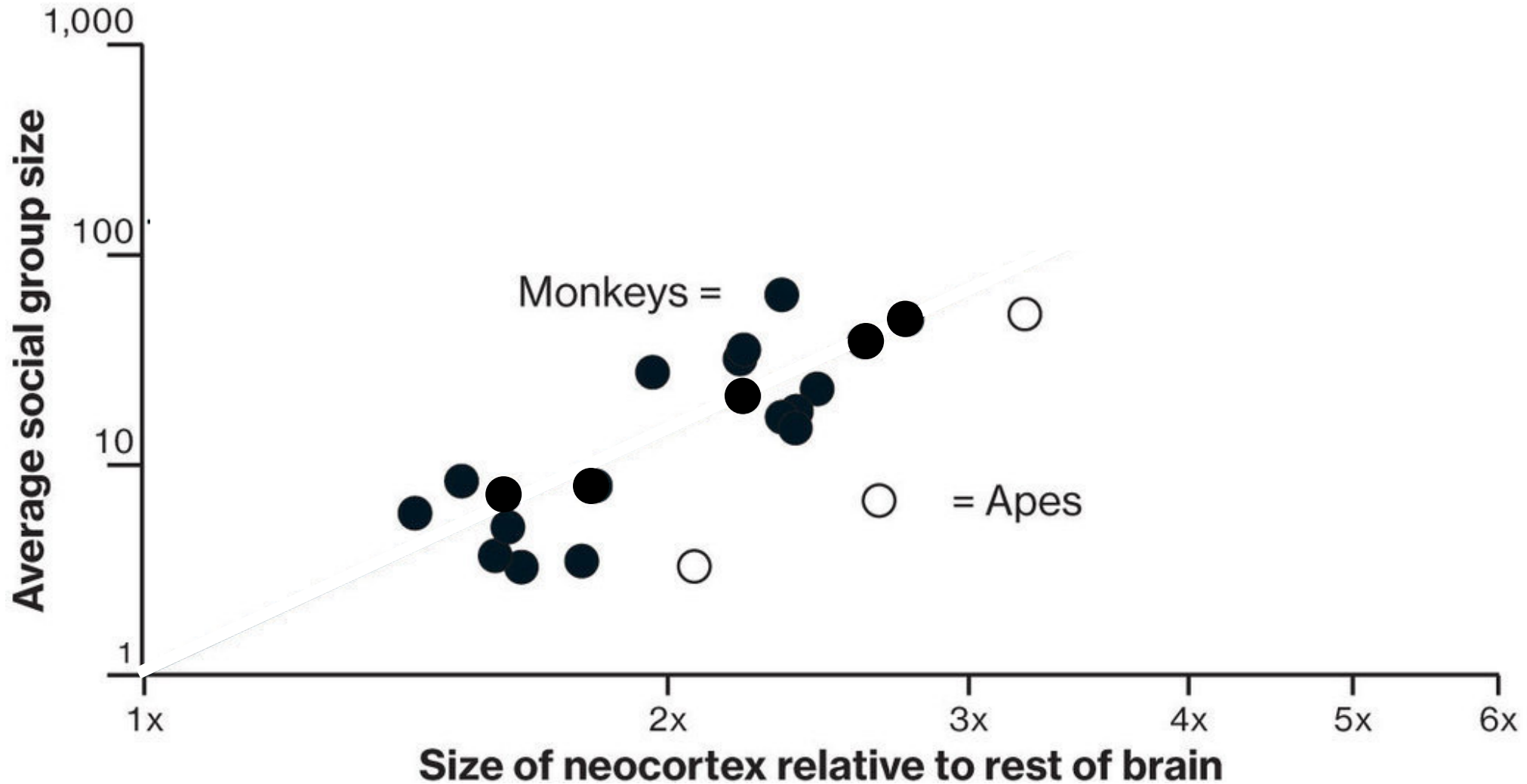
Dunbar's Number



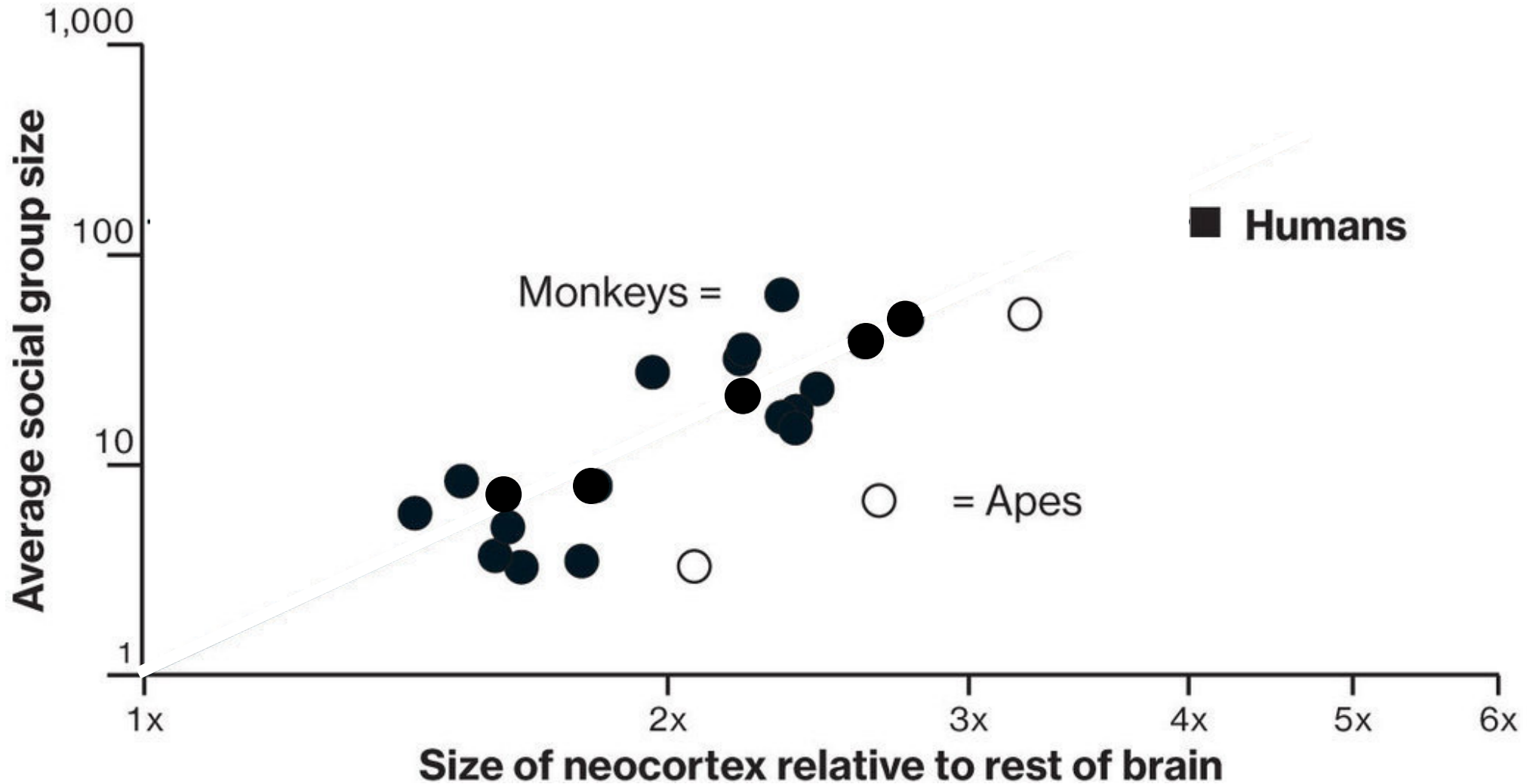
How About ?



The Social Cortex



The Social Cortex





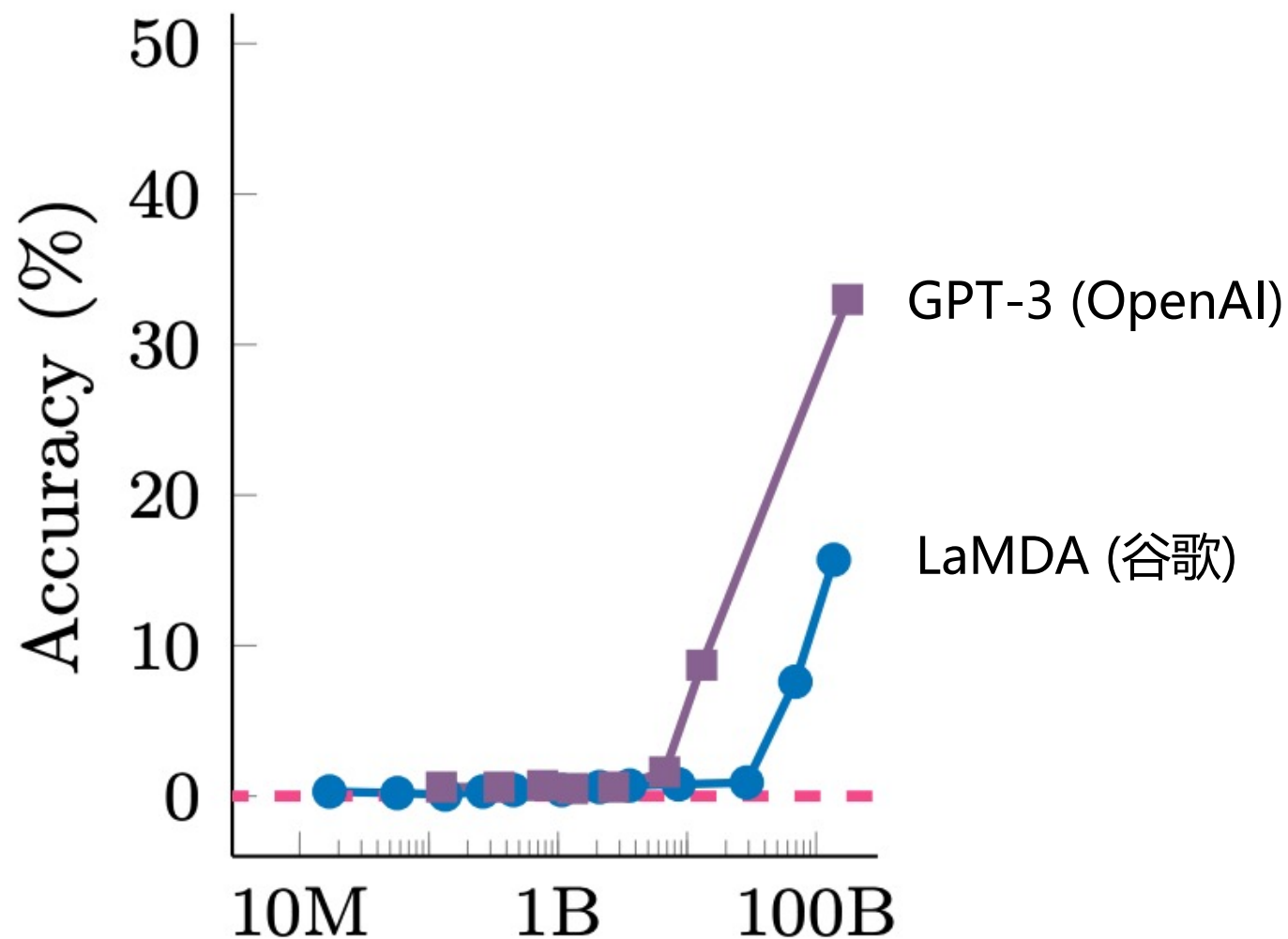
**Language
Morality**



**Culture
Consciousness
Tool Use**

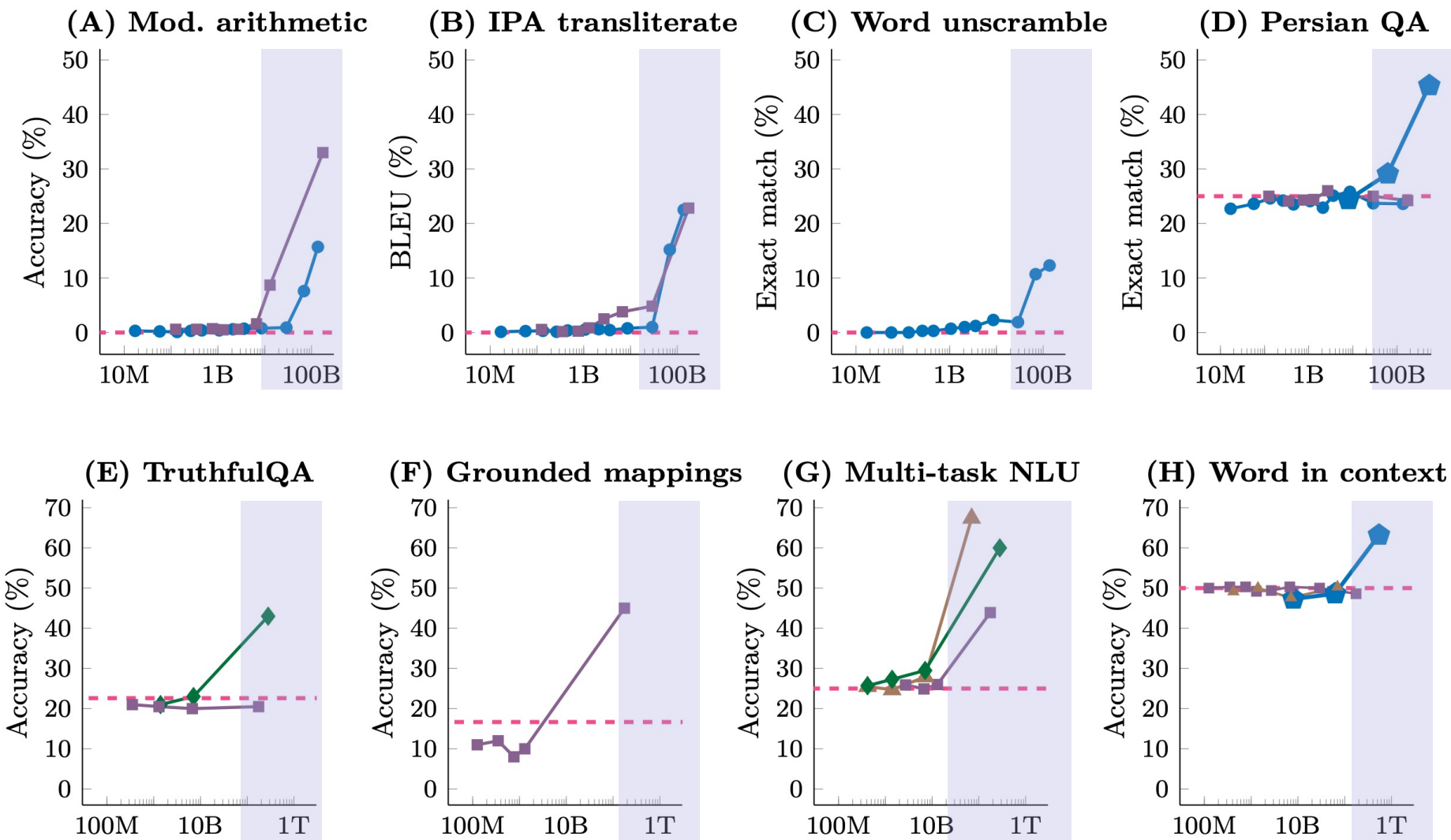
为什么千亿(100B)大规模

(A) Mod. arithmetic

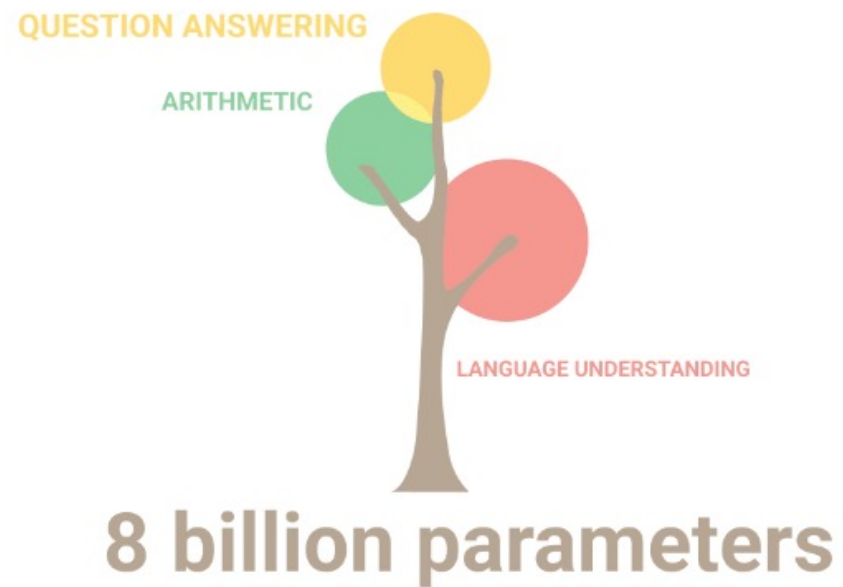


为什么千亿(100B)大规模

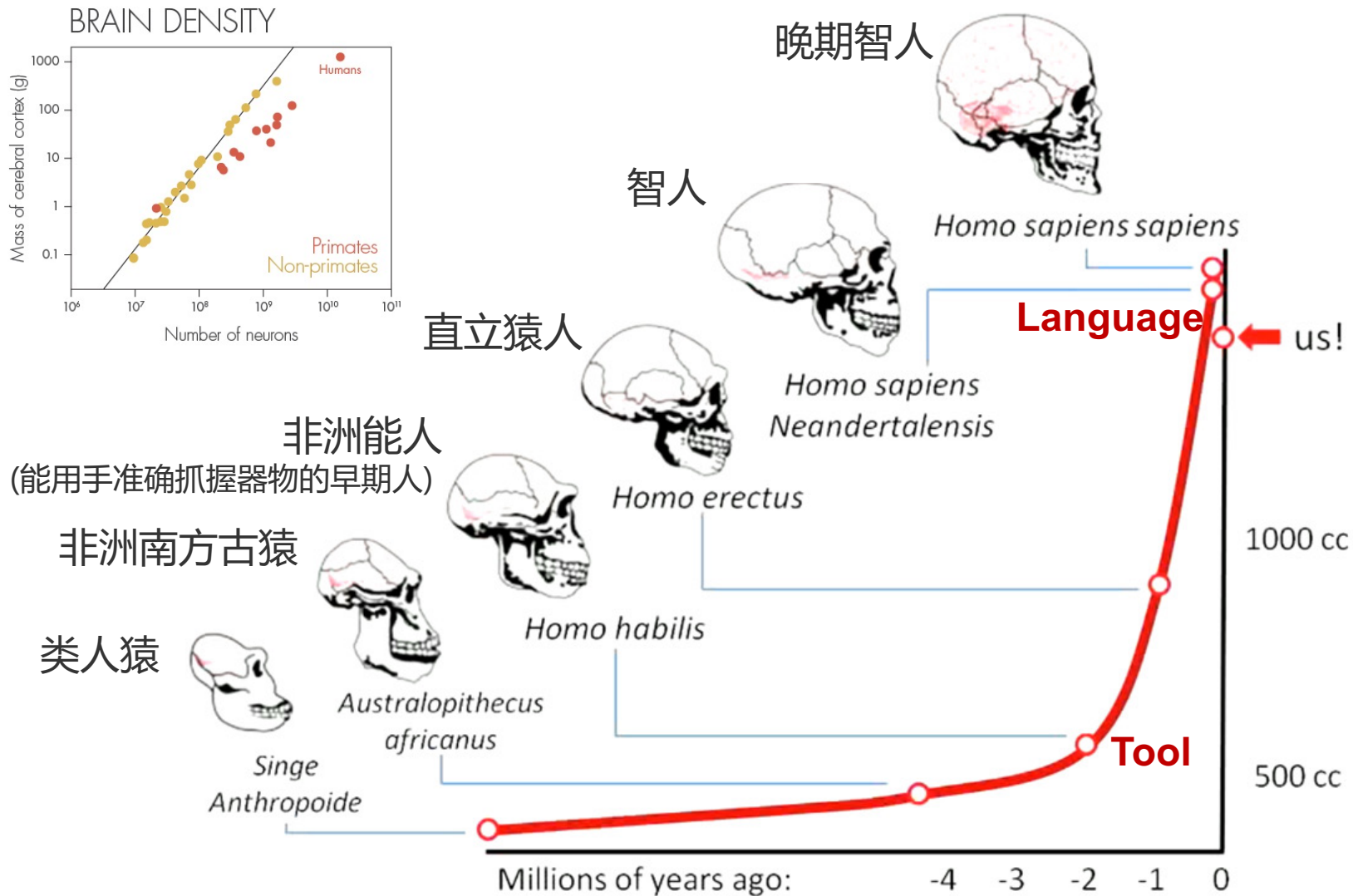
—●— LaMDA —■— GPT-3 —◆— Gopher —▲— Chinchilla —◆— PaLM - - - Random



“涌现的模型新能力”

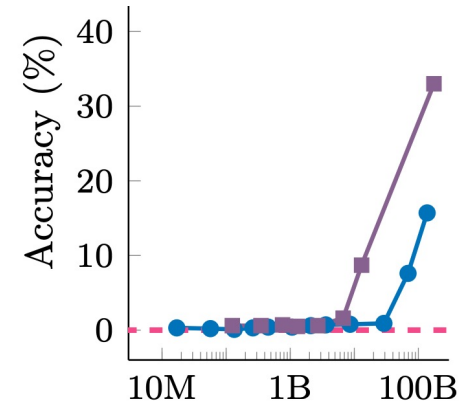
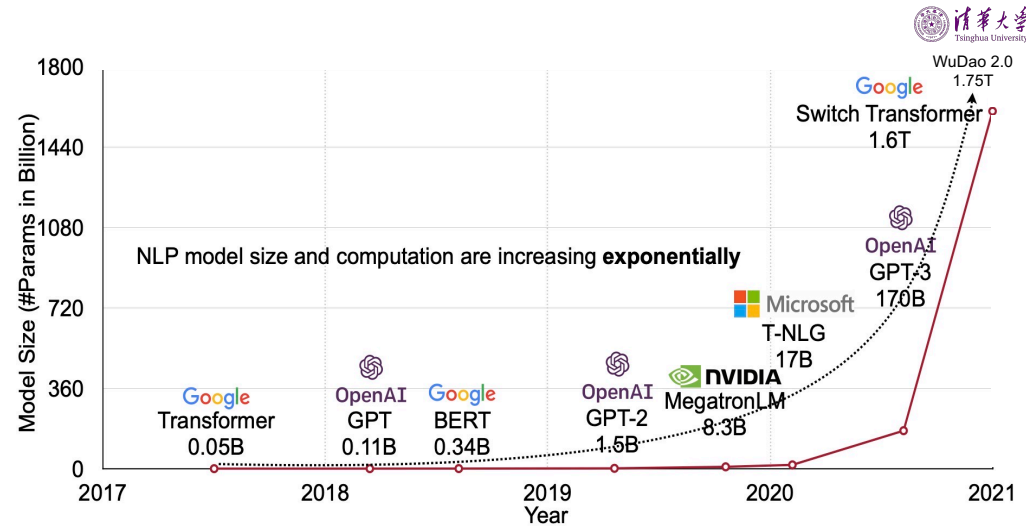
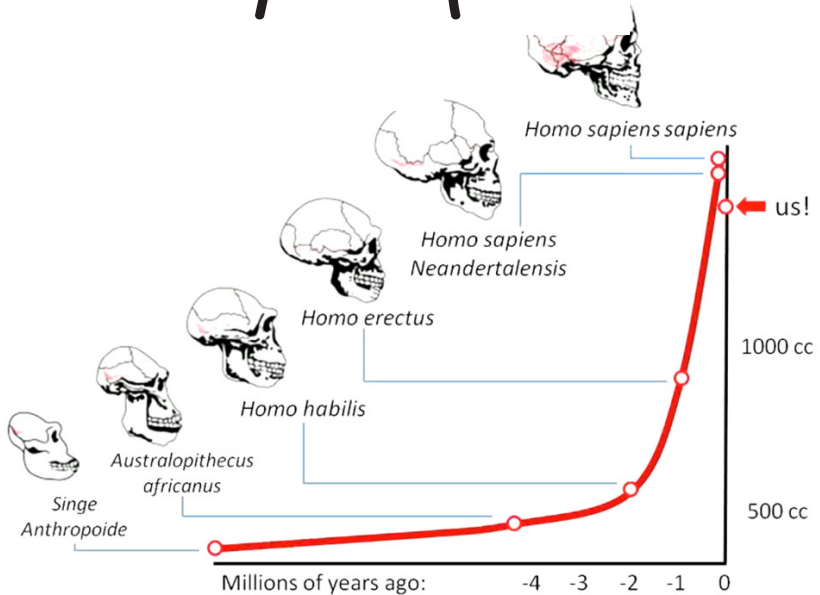
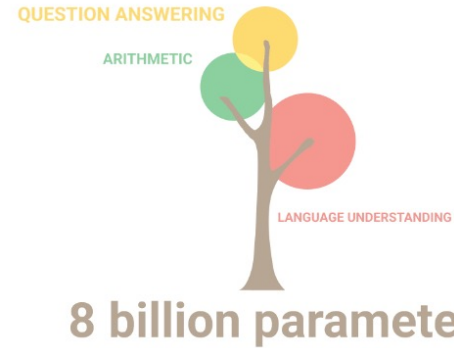
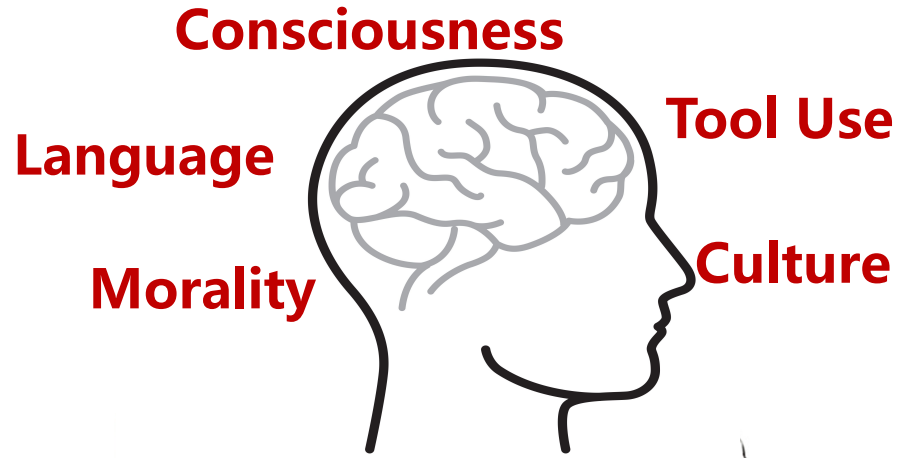


Brain Size Evolution Spike



Disclaimer: The instructor is not an expert on this topic at all

An Observation



Reasoning Tasks

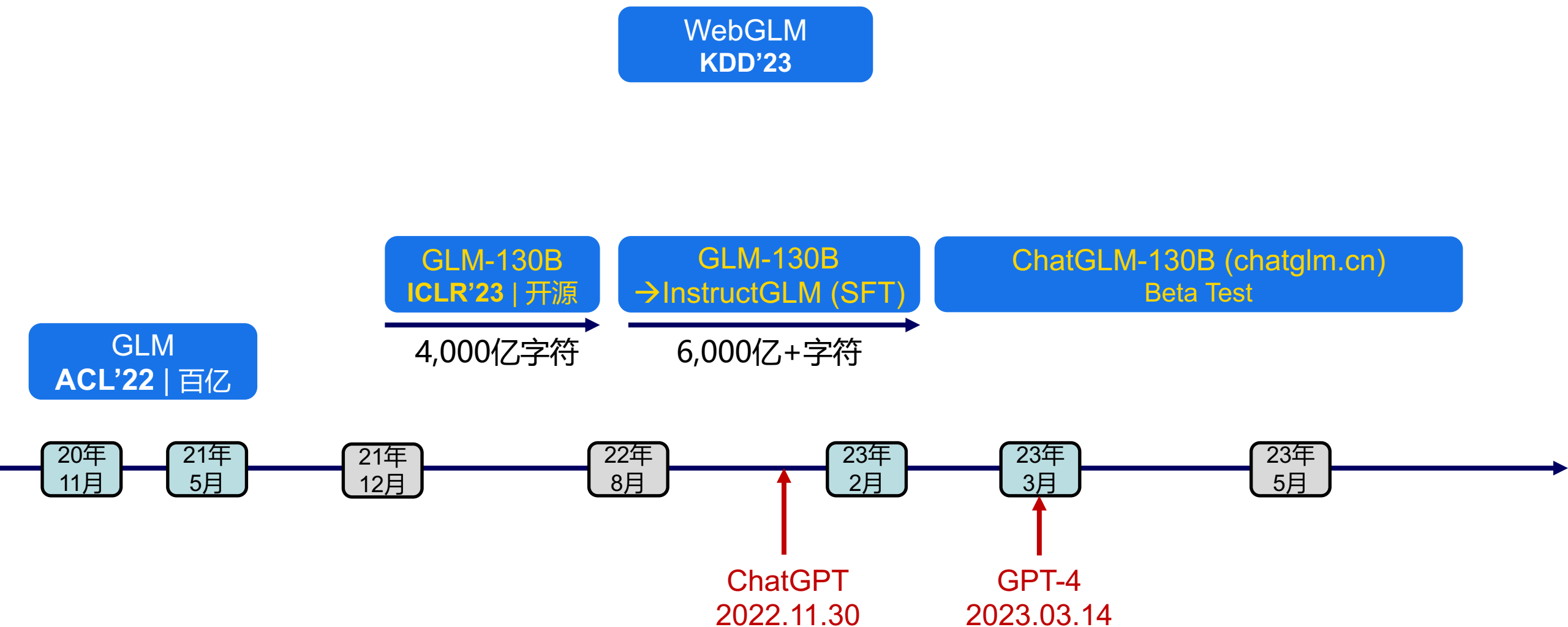
GLM-10B

- 一斤西瓜10元钱, 5斤西瓜多少钱?
 - 一斤西瓜十元,五斤西瓜30元
- 一斤苹果10元, 一斤西瓜5元, 那么2斤苹果和3斤西瓜多少钱?
 - 总共30元
- 我把可口可乐洒在了桌子上, 接下来该如何清楚桌面?
 - 把桌上的东西都拿走,用湿抹布擦桌子,然后喷上酒精,再拿抹布擦,最后用酒精再擦一遍

GLM-130B

- 一斤西瓜10元钱, 5斤西瓜多少钱?
 - $5 \times 10 = 50$ (元)
- 一斤苹果10元, 一斤西瓜5元, 那么2斤苹果和3斤西瓜多少钱?
 - $2 \times 10 + 3 \times 5 = 35$ (元)
- 我把可口可乐洒在了桌子上, 接下来该如何清楚桌面?
 - 先用抹布擦干净, 然后用湿抹布再擦一次, 最后用干抹布再擦一次

千亿基座 GLM-130B 到 ChatGLM 对话模型



ChatGLM 内测版 千亿对话模型

旨在推动大模型的科学研究，面向高校学术交流与行业合作的
试用申请

内测申请 [→](#)

已通过审核的用户请 [登录体验](#) [>](#)



ChatGLM 内测

提出你的想法



🕒 南北战争期间华盛顿为什么不...

🕒 北京和上海哪个城市常住人口...

🕒 河南有哪些比较酷的互联网公...

...

💡 试试这些例子:

帮我写一份川菜餐厅的五星好评，包含环境、菜品、服务等维度，不少于100字

请以表格形式产出唐宋八大家以及他们所属朝代和代表作

我在写一篇有关人工智能发展的论文，帮我列出提纲、目录结构

⚠️ 当前模型限制:

可能会生成不正确的信息

可能会产生有害说明或有偏见的内容

暂时不擅长逻辑类回答，如数学和编程类问题



ChatGLM：撰写博客提纲

你好

你好👋! 我是人工智能助手 ChatGLM，很高兴见到你，欢迎问我任何问题。

Ⓞ 以上内容为 AI 生成，不代表开发者立场，请勿删除或修改本标记

帮我写一个介绍ChatGLM的博客提纲

好的，以下是一个可能的博客提纲：

I. 前言

1. 简要介绍 ChatGLM 的背景和主要功能

II. ChatGLM 的特点

1. 智能化的问答和对话功能
2. 自定义主题和语言
3. 丰富的知识库和语言模型
4. 支持多种平台和语言

III. ChatGLM 的应用场景

1. 在教育、医疗、金融等领域的应用
2. 帮助人们更高效地获取信息和解决问题
3. 推动人工智能的发展和普及

IV. ChatGLM 的未来发展方向

1. 进一步完善智能化和个性化服务
2. 拓展应用场景和用户规模
3. 推动人工智能在各个领域的应用和创新



V. 总结


1. 简要总结 ChatGLM 的特点和优势
2. 强调 ChatGLM 在人工智能发展和普及中的重要作用

以上是一个简单的博客提纲，根据具体情况，也可以适当调整和补充。希望对您有所帮助!

Ⓞ 以上内容为 AI 生成，不代表开发者立场，请勿删除或修改本标记

ChatGLM：写邮件

 帮我写封邮件，控诉暴雪公司无端与网易公司解除合作，中国玩家无法接受。要求他们立刻改正错误，言辞恳切严厉。 

 尊敬的暴雪公司：

我是一名热爱暴雪游戏的中国玩家，我无法接受你们无端与网易公司解除合作的决定。这种行为让中国玩家感到震惊和失望，我们感到十分无助。

在过去的十四年中，你们在中国的游戏代理权交给了网易公司，玩家们可以安心地享受游戏服务。然而，最近你们突然宣布与网易公司解除合作，导致许多玩家的游戏账号面临被封禁的风险。这种行为显然违反了游戏代理商之间的合同约定，也违反了玩家们的知情权和利益。

更为令人不解的是，你们并没有给出任何合理的解释或解决方案，仅仅是简单地宣布与网易公司解除合作。这种行为是极其不负责任和不道德的。玩家们无法理解为什么你们会做出这样的决定，我们也无法想象你们将如何面对如此大规模的玩家投诉和抗议。

在此，我强烈要求你们立刻改正错误，与网易公司重新签订合作协议，确保玩家们的游戏账号安全。同时，我们要求你们给出一个明确的解决方案，给玩家们一个满意的答复。如果你们不能履行承诺，我们将被迫采取进一步的行动来维护自己的权益。

中国玩家对你们的行为感到非常失望和愤怒，我们相信你们也会对此负责并认真对待。我们希望你们能够重新审视与中国玩家的关系，为中国游戏市场做出更加积极的贡献。

此致

敬礼

[你的名字]

 以上内容为 AI 生成，不代表开发者立场，请勿删除或修改本标记

ChatGLM：介绍自己的优点缺点



介绍一下你的优点和缺点



作为一个人工智能语言模型，我的优点主要是：

1. 强大的自然语言处理能力：我可以处理多种语言的文本，并且能够自然地理解和生成文本。
2. 广泛的知识储备：我经过了广泛的训练，掌握了广泛的知识 and 信息，因此可以为用户提供有用的信息和回答。
3. 智能推理和推断能力：我可以推断出文本之间的关系和逻辑，并根据用户的问题来提供恰当的回答。
4. 适应性强：我可以适应不同的用户和场景，并且能够随时更新和改进我的模型和算法。

我的缺点主要包括：

1. 依赖于数据和算法：我的表现和性能取决于数据和算法的质量和数量，因此如果数据或算法出现问题，可能会导致我的表现下降。
2. 没有真正的情感和意识：我被设计成一个没有情感和意识的机器模型，因此不能像人类一样理解情感和价值观。
3. 容易被误导：由于我的理解是基于数据和算法，因此容易被误导或产生错误的结论。
4. 无法处理复杂和抽象的问题：对于一些复杂和抽象的问题，我可能无法给出准确的答案或理解。

Ⓞ 以上内容为 AI 生成，不代表开发者立场，请勿删除或修改本标记

ChatGLM：写剧本梗概



写一个谍战剧本，故事发生在二战欧洲战场，双面间谍



标题：双面间谍的惊险之旅

在二战欧洲战场上，德国纳粹军队的强大力量使得许多欧洲国家被迫屈服。然而，在这个动荡的时期，有一个双面间谍，他的名字叫约翰。

约翰是一个出生在英国的年轻人，他在大学学习政治和经济学。然而，当他听到德国纳粹的崛起和对欧洲大陆的侵略时，他感到非常愤怒。于是，他毅然决定加入英国情报局，成为一名双面间谍。

约翰的工作任务是收集德国情报，并将这些信息传递给英国情报局。他通过各种手段，包括接头、暗号和假身份，成功地完成了任务。然而，约翰也深知，作为一名双面间谍，他必须在政治和道德上做出艰难的决定。

在一次接头中，约翰遇到了一位名叫玛丽的女性。玛丽是一名德国情报员，她正在寻找一名可靠的双面间谍来收集英国情报。约翰和玛丽相互吸引，并逐渐陷入了爱情。然而，他们的关系充满了矛盾，因为约翰必须将玛丽视为敌人。

在约翰和玛丽的关系中，约翰必须不断地在两个国家之间游走，同时还要保持对德国纳粹的警惕。他必须小心翼翼地处理自己的身份和职业，以免被发现。

在约翰的工作中，他遇到了许多困难和挑战。有时他必须冒着生命危险去收集情报，有时他必须欺骗自己的同事和朋友。然而，他始终坚持自己的信念，成为一名真正的双面间谍。

最终，约翰的计划取得了成功。他收集了许多重要的德国情报，并及时地将这些信息传递给了英国情报局。在这场艰苦的战争中，约翰用自己的智慧和勇气，为英国和盟军的胜利做出了巨大的贡献。

然而，约翰和玛丽的关系最终还是失败了。由于政治和道德上的矛盾，他们不得不选择分手。约翰继续扮演着双面间谍的角色，但在他内心深处，他深深地爱着玛丽。

这是一场充满惊险和挑战的谍战剧，一个真正的双面间谍，用他的智慧和勇气，为英国和盟军的胜利做出了巨大的贡献。

ⓘ 以上内容为 AI 生成，不代表开发者立场，请勿删除或修改本标记

ChatGLM: 写代码



用python帮忙写个冒泡排序算法的代码, 再给一个数组作为例子, 写个测试程序



以下是冒泡排序算法的 Python 代码实现:

```
def bubble_sort(arr):  
    n = len(arr)  
    # 遍历所有数组元素  
    for i in range(n):  
        # 最后 i 个元素已经排好序, 不需要再比较  
        for j in range(0, n-i-1):  
            # 如果当前元素比下一个元素大, 则交换它们的位置  
            if arr[j] > arr[j+1]:  
                arr[j], arr[j+1] = arr[j+1], arr[j]  
    return arr
```

下面是测试程序, 它将输入一个数组, 并使用冒泡排序算法对其进行排序, 然后输出排序后的数组:

```
arr = [64, 34, 25, 12, 22, 11, 90]  
sorted_arr = bubble_sort(arr)  
print(sorted_arr)
```

输出结果如下:

```
[11, 12, 22, 25, 34, 64, 90]
```

这是一个基本的冒泡排序算法, 可以根据需求对其进行修改和扩展。

ⓘ 以上内容为 AI 生成, 不代表开发者立场, 请勿删除或修改本标记

ChatGLM: 查询常见知识/教程

中国手机号的正则表达式?

中国手机号的正则表达式可以根据不同的地区和运营商有所不同，以下是一般通用的正则表达式：

```
^1[3-9]\d{9}$
```

解释：

1. `^` 匹配字符串的开始位置
2. `1` 匹配数字 1，代表手机号码的开头
3. `[3-9]` 匹配数字 3-9 中的任意一个
4. `\d{9}` 匹配 9 个数字
5. 匹配字符串的结束位置

以上正则表达式表示手机号码的开头必须是数字 1，第二位是 3-9 中的任意一个数字，后面跟着 9 个数字，总共 11 位。这个正则表达式可以匹配大部分中国的手机号码。

请注意，不同国家和地区的手机号码长度和格式可能有所不同，因此需要根据实际情况进行调整。

以上内容为AI生成



怎么修改huggingface transformers库的模型缓存地址

要修改 huggingface transformers 库的模型缓存地址，您可以按照以下步骤进行操作：

1. 打开您的项目，然后导入 huggingface transformers 库。

```
import transformers
```

2. 修改 `transformers.config.config_file()` 函数中的缓存地址。您可以将缓存地址更改为您想要使用的地址。例如，如果您想要将缓存地址更改为 `~/.huggingface_transformers_cache`，则在函数中添加以下代码：

```
import os

cache_dir = os.path.join(os.path.expanduser("~/huggingface_transformers.config.config_file(cache_dir)
```

3. 重新加载 huggingface transformers 库，以便更改缓存地址生效。您可以使用以下代码来重新加载库：

```
transformers.transformers.load_configs(cache_dir)
```

完成这些步骤后，您的 huggingface transformers 库的模型缓存地址将更改为您指定的地址。

ChatGLM: 多轮问答

Stable Diffusion使用了什么数据集进行训练

Stable Diffusion 是一个文本到图像的潜在扩散模型，其训练数据来源于公开可用的数据集 LAION-5B。LAION-5B 是一个包含成对图像和字幕的数据集，每个对图像和字幕都对应着一段文本。Stable Diffusion 使用这些数据集进行训练，以生成高质量图像。

⚠ 以上内容为AI生成

具体介绍一下这个数据集的大小、内容和质量

LAION-5B 是一个公开可用的数据集，其大小为 50 亿个图像 - 文本对。这些数据对包含了多种类型的图像，如风景、人物、动物等，同时包含了多种不同类型的文本，如标语、标题、说明等。

LAION-5B 数据集的质量也很高。数据集的创建过程采用了最先进的技术，包括自然语言处理、计算机视觉和机器学习等技术，以确保数据集的准确性、完整性和代表性。此外，数据集还经过了严格的质量控制和筛选，以确保数据的质量和一致性。

⚠ 以上内容为AI生成

我应该在哪儿下载LAION-5B


你可以访问 LAION-5B 的数据集官方网站 <https://laion.ai/blog/laion-5b/> 来下载这些数据集。在官方网站上，你可以找到数据集的压缩文件，下载并解压缩后即可使用。


另外，LAION-5B 数据集也提供了 img2dataset 分布式下载工具，你可以通过该工具来下载数据集。img2dataset 是一个用于分布式下载大型数据集的工具，它可以帮助你更高效地下载数据集。





无论从哪里下载数据集，都应该确保数据集的来源可靠，并且已经经过严格的质量控制和筛选，以确保数据的质量和准确性。


⚠ 以上内容为AI生成


ChatGLM：文字冒险游戏

 生成一个文字冒险游戏，背景是三体。我扮演一个刚刚被选为面壁人的20岁男性，第一个场景是我在地球联邦总部的大门口。





 欢迎来到三体文字冒险游戏！您是一个刚刚被选为面壁人的 20 岁男性，现在站在地球联邦总部的大门口。请问您有什么需要帮忙或者需要前往的地方吗？
(提示：您可以输入一些关键词或者问题来触发不同的对话和剧情，比如“接下来该怎么办”、“我需要购买一些物品”)


 以上内容为AI生成   


 我要进入总部见面地球联邦议会会长，似乎有神秘任务要交给我





 好的，您现在来到了地球联邦总部的大门口。门口有两个守卫，他们正在看着您。
守卫 1：您好，欢迎来到地球联邦总部。您要前往会长办公室吗？
您：是的，我有一个非常重要的任务需要交给会长。
守卫 1：好的，请您出示您的邀请函或者身份证明。
您：我没有邀请函，但我是刚刚被选为面壁人的 20 岁男性，会长应该听说过我的名字。
守卫 1：哦，那请您跟我们一起来吧。我们得先请示一下会长，看看是否允许您进入。
您：好的，谢谢。
您跟着守卫来到了会长办公室门口。守卫 2 正在门口等待您。
守卫 2：您好，会长让您进去。
您：谢谢，我马上就要进去了。
您推开了会长办公室的门，发现里面有一位中年男子正在等待着您。
会长：您好，年轻人。听说您有一个非常重要的秘密任务要交给我？


您：是的，会长。我刚刚被选中为面壁人，据说您有一个重要的任务要交给我。
会长：没错。这个任务非常重要，关系到地球联邦的未来。我们需要您来完成这个任务。
您：我愿意接受这个任务，但我想先了解一下具体的任务内容。
会长：好的，您可以先回去休息一下，我们会尽快给您安排。
您：好的，我会尽快完成任务的。
您离开了会长办公室，回到了地球联邦总部的大门口。您想继续前往哪里呢？


 以上内容为AI生成   

 会长让我先去休息。那我找个酒店先住下吧。





 好的，您现在站在地球联邦总部的大门口，考虑着接下来该怎么办。
您看到周围有一些酒店和旅馆，可以考虑在那里入住。请问您需要预订哪个酒店？(提示：您可以输入一些酒店的名字来触发不同的对话和剧情)

 以上内容为AI生成   


 我初来乍到，没有钱。广场上看上去只有一家名为“丽晶大宾馆”的小旅店我住的起。

 好的，您决定前往丽晶大宾馆入住。
您来到丽晶大宾馆的前台，发现有一位女士正在等待接待。
女士：您好，欢迎光临丽晶大宾馆。您需要预订房间吗？
您：是的，我初来乍到，需要一个房间休息一下。
女士：好的，请问您预订了吗？我们没有预订不接受入住。
您：我没有预订，但我没有钱。


女士：很抱歉，没有钱也不能入住。我们必须核实您的身份信息。
您：那我该怎么办？
女士：如果您没有预订也没有钱，您可以前往宾馆附近的商业街，寻找其他的酒店或者旅馆。可能会有一些酒店或者旅馆愿意接受您入住。
您：好的，我会去试试的。
您离开了丽晶大宾馆，来到了附近的商业街。您寻找着其他的酒店或者旅馆，但是发现大部分的酒店和旅馆都已经客满了。
您来到了一家名为“小天鹅酒店”的小旅店，看上去比较经济实惠。
您：请问这家小旅店是否还有房间？
服务员：抱歉，我们已经客满了。但是如果您愿意支付更高的价格，我们可以给您安排一个更好的房间。
您：好的，我考虑一下。
您决定在商业街上逛逛，看看是否有其他的酒店或者旅馆能够接受您的入住。
经过一番寻找，您终于在商业街的另一端找到了一家名为“宝来酒店”的酒店。
您：请问这家酒店是否还有房间？
服务员：抱歉，我们也已经客满了。但是如果您愿意支付更高的价格，我们可以给您安排一个更好的房间。
您：好的，我考虑一下。
您决定在宝来酒店入住，您进入酒店后，发现房间非常干净整洁，而且价格也比较实惠。
您：看来这次来对了，感谢。
宝来酒店的服务员：不客气，欢迎入住。如果您有其他需求，随时可以联系我们。
您：好的，我会的。谢谢。
您入住了宝来酒店，度过了一个安静的夜晚。明天，您将开始执行会长交给您的神秘任务。

 以上内容为AI生成   

ChatGLM: JSON格式事件抽取

 8月21日下午，清华大学计算机类2022级本科生开学典礼在大礼堂举行。计算机类首席教授吴建平院士，软件学院孙家广院士，计算机系主任尹霞，软件学院院长王建民，计算机系书记贾珈，软件学院书记王斌，计算机系副主任武永卫、徐恪，软件学院副院长张慧、副书记王朝坤、学生工作组组长闻立杰，交叉信息研究院教师代表徐葳，计算机类2022级班主任、辅导员以及2022级本科生全体新生通过线上线下相结合的方式参加了典礼。大会由计算机系党委副书记刘知远主持。

提取“人”(name, position), “时间”, “事件”, “地点”类型的实体, 并输出JSON格式



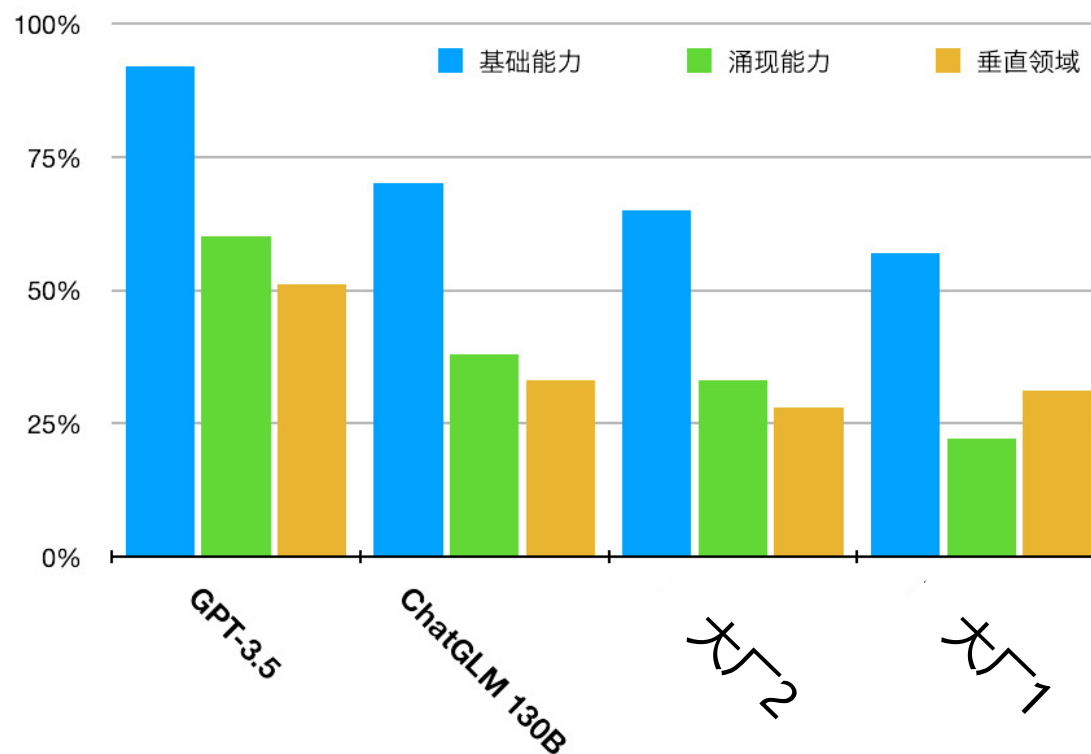
```
{  
  "人": [  
    {  
      "name": "吴建平院士",  
      "position": "计算机类首席教授"  
    },  
    {  
      "name": "孙家广院士",  
      "position": "软件学院院士"  
    },  
    {  
      "name": "尹霞",  
      "position": "计算机系主任"  
    },  
    {  
      "name": "王建民",  
      "position": "软件学院院长"  
    },  
    {  
      "name": "贾珈",  
      "position": "计算机系书记"  
    },  
    {  
      "name": "王斌",  
      "position": "软件学院书记"  
    },  
    {  
      "name": "武永卫",  
      "position": "计算机系副主任"  
    },  
    {  
      "name": "徐恪",  
      "position": "计算机系副主任"  
    },  
    {  
      "name": "张慧",  
      "position": "软件学院副院长"  
    },  
    {  
      "name": "王朝坤",  
      "position": "软件学院副书记"  
    },  
    {  
      "name": "闻立杰",  
      "position": "软件学院学生工作组组长"  
    },  
    {  
      "name": "徐葳",  
      "position": "交叉信息研究院教师代表"  
    }  
  ],  
  "时间": "8月21日下午",  
  "事件": "清华大学计算机类2022级本科生开学典礼",  
  "地点": "大礼堂"  
}
```

```
{  
  "name": "孙家广院士",  
  "position": "软件学院院士"  
},  
{  
  "name": "尹霞",  
  "position": "计算机系主任"  
},  
{  
  "name": "王建民",  
  "position": "软件学院院长"  
},  
{  
  "name": "贾珈",  
  "position": "计算机系书记"  
},  
{  
  "name": "王斌",  
  "position": "软件学院书记"  
},  
{  
  "name": "武永卫",  
  "position": "计算机系副主任"  
},  
{  
  "name": "徐恪",  
  "position": "计算机系副主任"  
},  
{  
  "name": "张慧",  
  "position": "软件学院副院长"  
},  
{  
  "name": "王朝坤",  
  "position": "软件学院副书记"  
},  
{  
  "name": "闻立杰",  
  "position": "软件学院学生工作组组长"  
},  
{  
  "name": "徐葳",  
  "position": "交叉信息研究院教师代表"  
}  
],  
"时间": "8月21日下午",  
"事件": "清华大学计算机类2022级本科生开学典礼",  
"地点": "大礼堂"  
}
```

真格基金和第三方开放评测结果

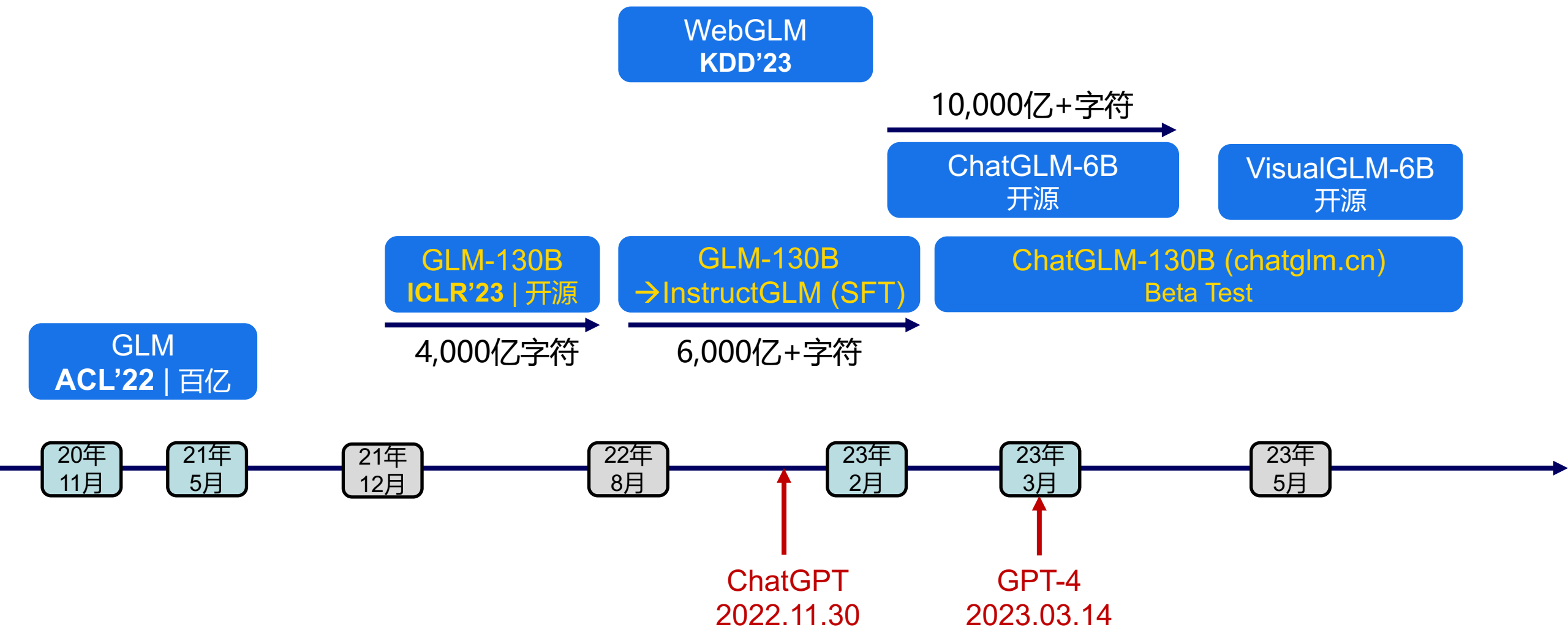
	GPT3.5	GPT4	大厂1	ChatGLM
基础能力	59/63	60/63	26/63	40/63
进阶能力	120/209	159/209	41/209	74(+5)/209
垂直领域	20/39	26/39	11/39	11(+5)/39
总	199/311	245/311	78/311	125(+10)/311
折合得分	64.0	78.8	25.1	40.2(+3)

2023.03.20



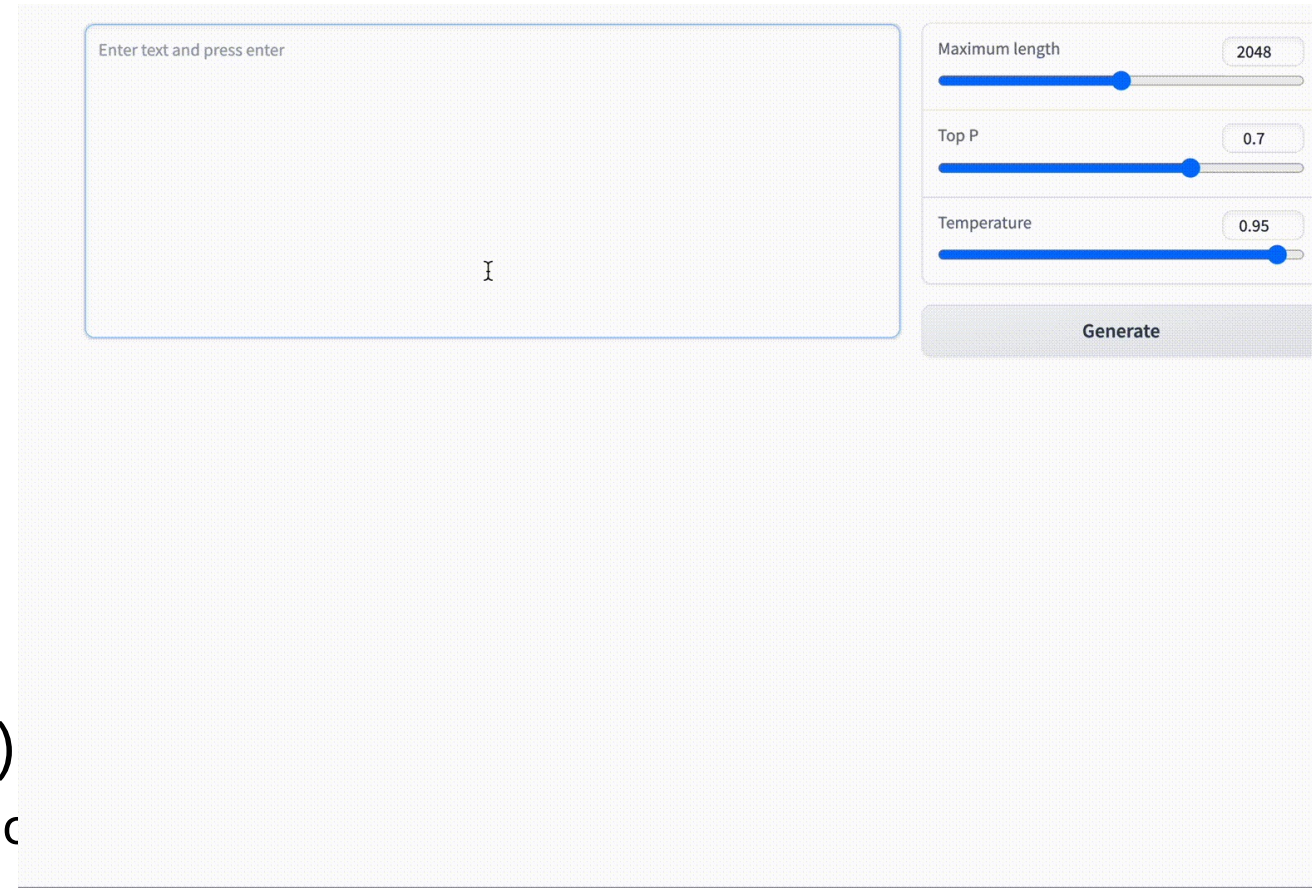
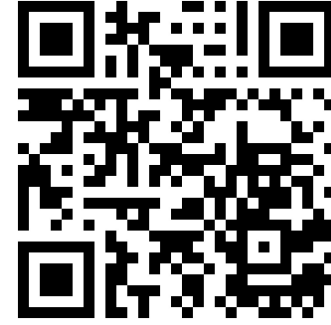
2023.05.12

千亿基座 GLM-130B 到 ChatGLM 对话模型



在自己电脑上安装 ChatGLM-6B

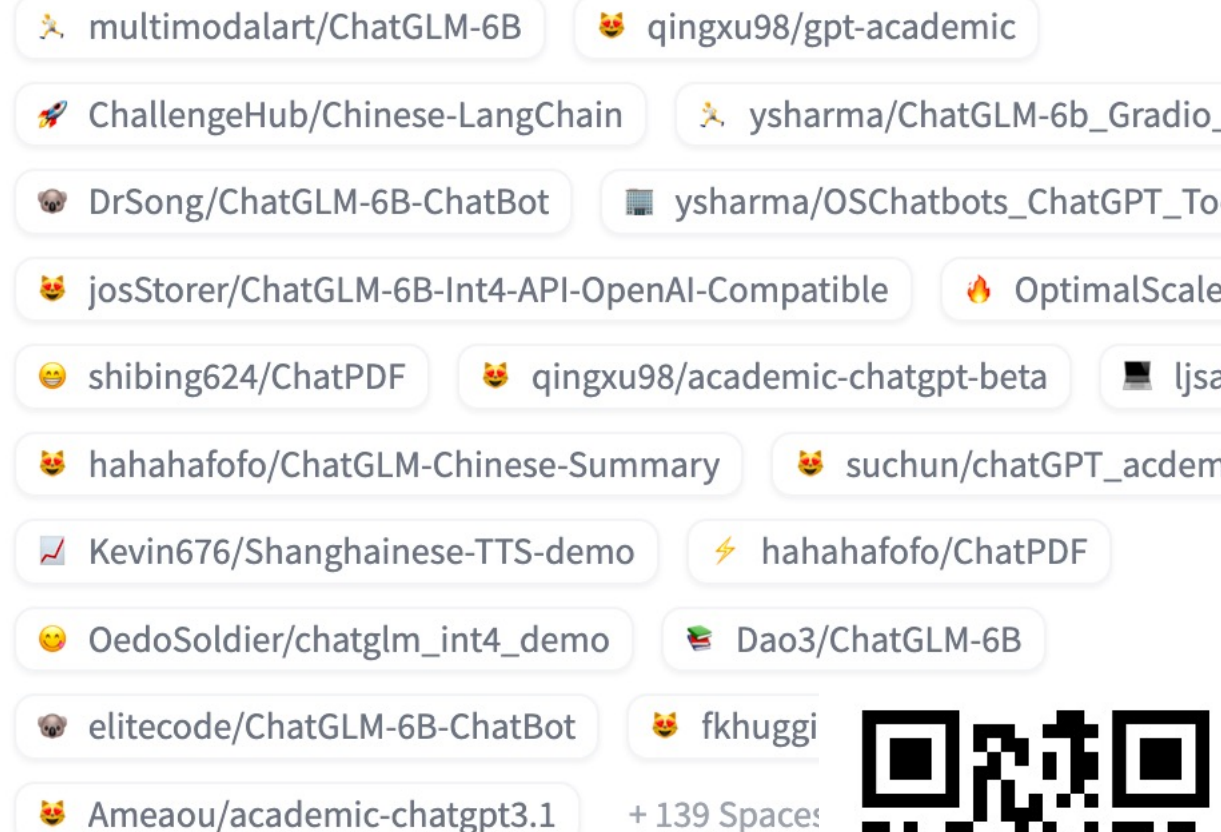
1. Download all model files from Huggingface
 1. git clone <https://huggingface.co/THUDM/chatglm-6b>
2. Download the demo
 1. git clone <https://github.com/THUDM/ChatGLM-6B>
 2. cd ChatGLM-6B
3. Install the demo
 1. pip install gradio
 2. python web_demo.py
4. Interactive demo
 1. python cli_demo.py
5. Install the api
 1. pip install fastapi uvicorn
 2. python api.py
6. Run on your MAC (w/ Apple Silicon)
 1. model = AutoModel.from_pretrained("your lc



在自己电脑上安装 ChatGLM

- **ChatGLM-6B**
 - 6B parameters
 - 1T tokens training data
 - 6G GPU mem (INT4)
- Mar. 14, 2023, open-sourced model
- Mar. 16, 2023, **#1** on GitHub Trending
- Mar. 18-30, **#1** on HF Trending
- May. 25, 2023, 25.7k stars in GitHub
2M downloads in HF

📦 Spaces using THUDM/chatglm-6b 159



<https://huggingface.co/THUDM>



在自己电脑上安装 ChatGLM

- **ChatGLM-6B**
 - 6B parameters
 - 1T tokens training data
 - 6G GPU mem (INT4)
- Mar. 14, 2023, open-sourced model
- Mar. 16, 2023, **#1** on GitHub Trending
- Mar. 18-30, **#1** on HF Trending
- May. 25, 2023, 25.7k stars in GitHub
2M downloads in HF

友情链接

对 ChatGLM 进行加速或者重新实现的开源项目：

- [SwissArmyTransformer](#): 一个Transformer统一编程框架，ChatGLM-6B已经在SAT中进行实现并可以P-tuning微调。
- [ChatGLM-MNN](#): 一个基于 MNN 的 ChatGLM-6B C++ 推理实现，支持根据显存大小自动分配计算任务到 GPU 和 CPU
- [JittorLLMs](#): 最低3G显存或者没有显卡都可运行 ChatGLM-6B FP16，支持Linux、windows、Mac部署

基于或使用了 ChatGLM-6B 的开源项目：

- [chatgpt_academic](#): 支持ChatGLM-6B的学术写作与编程工具箱，具有模块化和多线程调用LLM的特性，可并行调用多种LLM。
- [闻达](#): 大型语言模型调用平台，基于 ChatGLM-6B 实现了类 ChatPDF 功能
- [glm-bot](#): 将ChatGLM接入Koishi可在各大聊天平台上调用ChatGLM
- [Chinese-LangChain](#): 中文langchain项目，基于ChatGLM-6b+langchain实现本地化知识库检索与智能问答生成，增加web search功能、知识库选择功能和支持知识增量更新
- [bibliothecarius](#): 快速构建服务以集成您的本地数据和AI模型，支持ChatGLM等本地化模型接入。
- [langchain-ChatGLM](#): 基于 langchain 的 ChatGLM 应用，实现基于可扩展知识库的问答
- [ChatGLM-web](#): 基于FastAPI和Vue3搭建的ChatGLM演示网站(支持chatglm流式输出、前端调整模型参数、上下文选择、保存图片、知识库问答等功能)
- [ChatGLM-6B-Engineering](#): 基于 ChatGLM-6B 后期调教，网络爬虫及 [Stable Diffusion](#) 实现的网络爬虫及图片生成
- [ChatGLM-OpenAI-API](#): 将 ChatGLM-6B 封装为 OpenAI API 风格，并通过 ngrok/cloudflare 对外提供服务，从而将 ChatGLM 快速集成到 OpenAI 的各种生态中。

对 ChatGLM-6B 进行微调的开源项目：

- [InstructGLM](#): 基于ChatGLM-6B进行指令学习，汇总开源中英文指令数据，基于Lora进行指令数据微调，开放了Alpaca、Belle微调后的Lora权重，修复web_demo重复问题

VisualGLM-6B=ChatGLM-6B + Visual



描述一下这个场景



泰坦尼克号，杰克和露丝在船上拥抱。
夕阳余晖下，他们的身影如此美丽动人。
海水翻涌，海浪拍击着船帆。
他们的爱如同这海洋般深邃而浩瀚。在这首诗中，我们感受到爱情的深沉与美好。
他们在彼此的怀抱里，享受着爱情的温暖。
他们的爱情就像那大海一般广阔无垠，永恒不变。



这部电影的导演是谁？



这部电影的导演是詹姆斯·卡梅隆(James Cameron)。



这是什么东西



这张图片展现了一只可爱的卡通羊驼，它站在一个透明的背景上。这只羊驼长着一张毛茸茸的耳朵和一双大大的眼睛，它的身体是白色的，带有棕色斑点。



它头上有什么



这只羊驼头上戴着一个绿色的小叶子



这种动物生活在哪里？



羊驼是一种生活在南美洲的哺乳动物，它们通常以草为食。



1) 场景描述 2) 进一步追问 3) 联系知识

VisualGLM-6B的多轮对话-效果



这张图片为什么奇怪



这张图片很奇特,因为一名男子站在一辆黄色出租车的车顶上,将熨斗放在车身上。这可能表明这名男子正在使用出租车作为移动熨斗店,为乘客提供熨斗服务。这种行为在城市中非常不寻常,因为出租车通常被用作交通工具,而不是临时的熨斗店。



这张图片描述了什么



这张图片展示了一只戴着眼镜、系着领结的狗。这只狗的毛色是白色的,眼睛周围有一圈黑色毛发。它坐在木栅栏前,背景是一片草地。



这张图为什么有趣

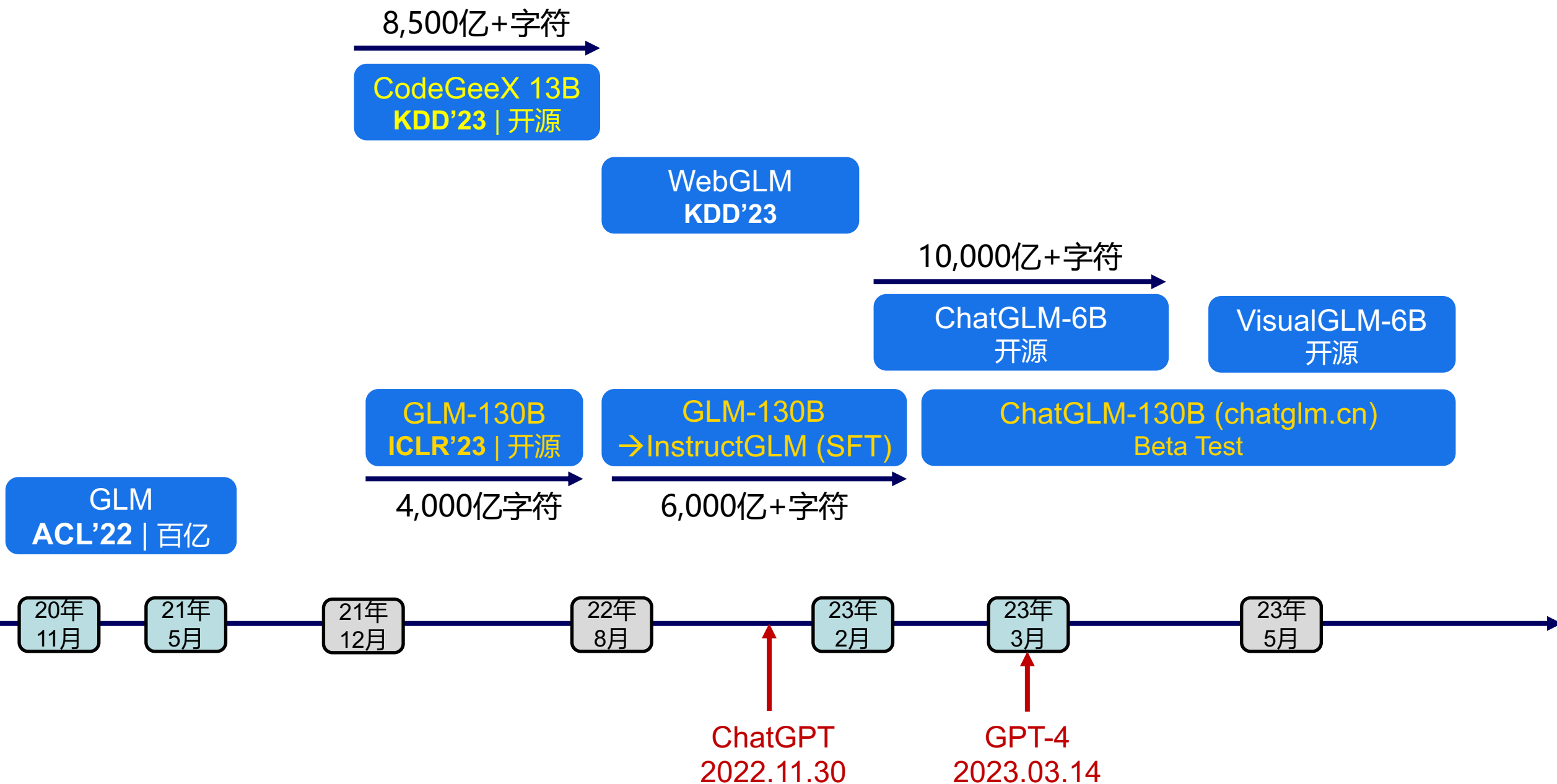


这张照片有趣的地方在于,这只狗戴着眼镜和领结。它坐在木栅栏前,背景是一片草地。这种设计使得照片看起来像是一张宠物肖像画,而狗狗则被描绘成一位穿着时髦的知识分子或艺术家。这给图片带来了一种独特的氛围,让人不禁想象这只狗正在思考或者正在创作一幅艺术作品。



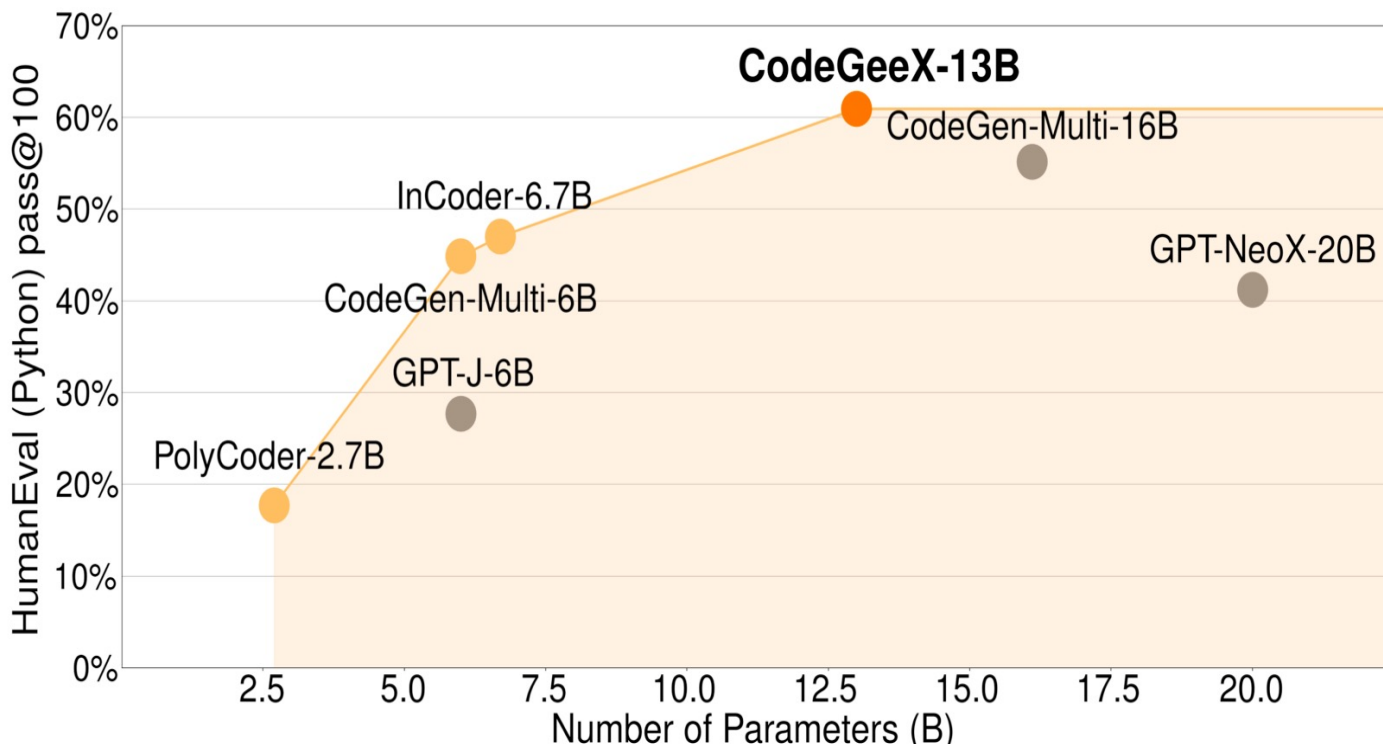
4) 联系常识 5) 发现异常 6) 做出解释

千亿基座 GLM-130B 到 ChatGLM 对话模型



代码模型CodeGeeX

- ▶ **130亿参数**、**20多种编程语言**代码生成预训练模型
- ▶ **支持昇腾和英伟达**，具有代码生成、翻译、注释等能力
- ▶ **2.6k GitHub星**，**VSCode、JetBrains编程插件**



CodeGeeX模型训练成功优化国产AI芯片

优化策略

- ▶ 算子融合(Layernorm/Gelu/BatchMatmul)
- ▶ 矩阵乘算子自动搜索效率最高的计算维度组合

性能提升

- ▶ **单卡昇腾910芯片训练效率提升257%**
- ▶ **千卡昇腾910芯片训练效率提升299%**

每天帮程序员编写**500多万行**代码 (2023.05)



CodeGeeX

Zhipu AI | 32,194 installs | ★★★★★ (19) | Free

CodeGeeX is an AI-based coding assistant, which can suggest code in the current or following lines. It is powered by a large-scale multilingual code generation model with 13 billion parameters, pretrained on a large code corpus of more than 20 programming languages.

CodeGeeX: Code Generation

```
example_python.py U ● ▶ ▾ 🔗 📄 ⋮  
1 # tell a story  
2 print(  
3     "Don't even try to guess the letter! It's pretty hard to guess the correct  
4     one... \nHere's a little story..."  
5 )  
6 print(  
7     "In a world where you get five chances to guess a letter in a world without  
8     a computer, you have two options to guess the secret code."  
9 )  
10 print("The first option is to guess it manually, which is the simplest.")  
11 print(  
12     "In the second option, you can go through the code to make sure the letters  
13     are always the same, but make sure you get at least one guess before you  
14     get a chance to guess the code."  
15 )  
16 print("The code is ABCDEFGHIJKLMNOPQRSTUVWXYZ")  
17 print(  
18     "The first letter of the code is A, and the last letter is Z. Your job is  
19     to guess the code using a guessing game."  
20 )  
21 print(  
22     "This guessing game takes time, so you should keep trying until you have  
23     guessed the right code. The bigger the number of chances, the longer you  
24     will take to win."  
25 )
```

CodeGeeX: Code Generation

```
example_go.go U ●
1 // write a bubble sort function
2 function bubble_sort(arr) {
3     for(var i=0;i<arr.length;i++) {
4         for(var j=1;j<arr.length-i;j++) {
5             if(arr[j]>arr[j-1]) {
6                 // swap the values
7                 var temp = arr[j];
8                 arr[j] = arr[j-1];
9                 arr[j-1] = temp;
10            }
11        }
12    }
13    return arr;
14 }
15
```

main* 0 0 Ln 15, Col 1 Tab Size: 4 UTF-8 LF Go

CodeGeeX: Code Translation

```
example_python.py U • CodeGeeX_translation ×
```

```
1 #include<stdio.h>
2 #include<vector>
3 #include<math.h>
4 using namespace std;
5 bool has_close_elements
  (vector<float> numbers, float
  threshold){
6     int n=numbers.size();
7     int i, j;
8     for(i=0;i<n;i++){
9         for(j=i+1;j<n;j++){
10            if(abs(numbers[i]
              -numbers[j])
              <threshold)
11                return true;
12        }
13    }
14    return false;
15 }
```

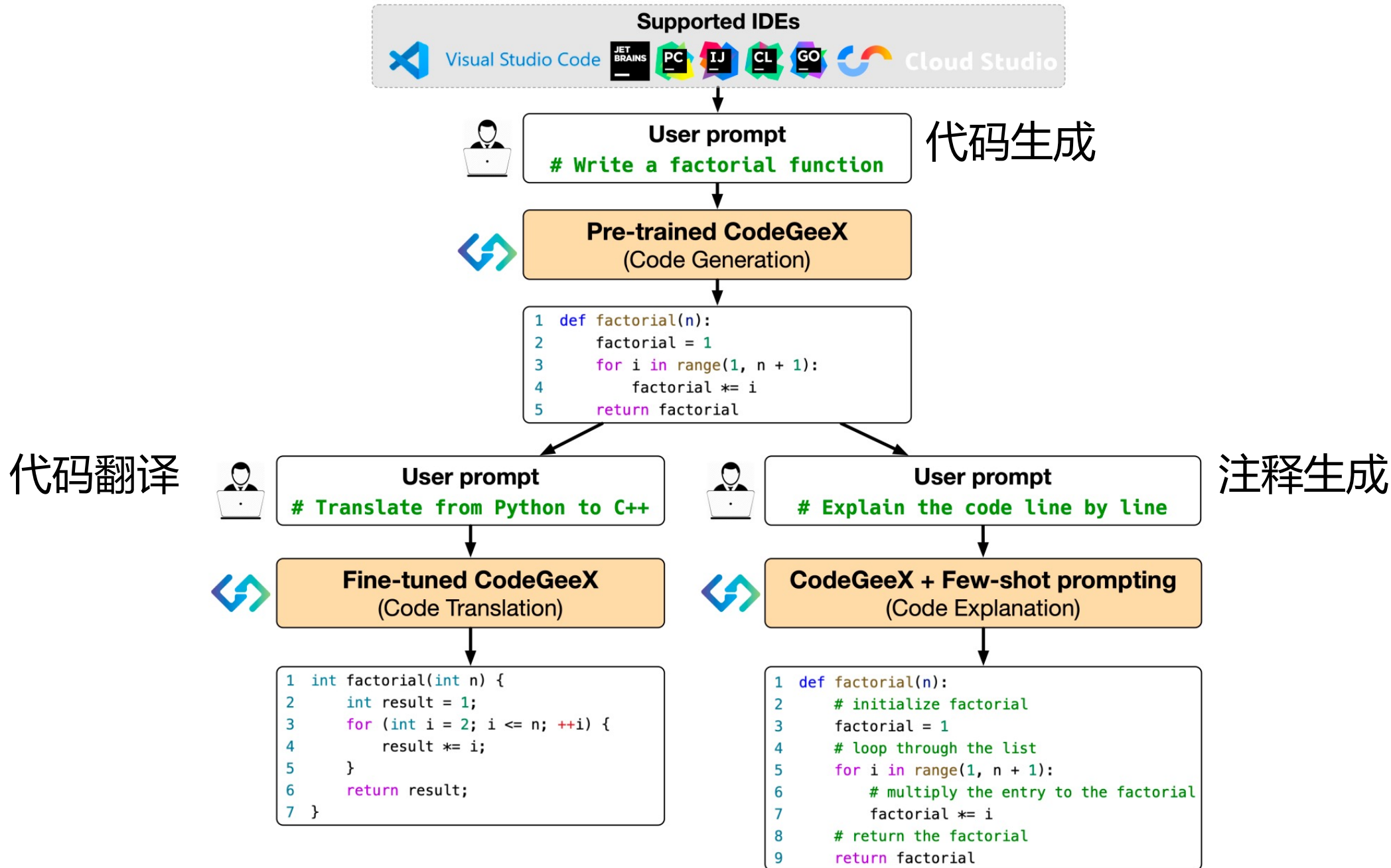
```
live code
1 from typing import List
2 def has_close_elements(numbers:
  List[float], threshold: float) ->
  bool:
3     return all(abs(numbers[i] -
      numbers[j]) < threshold for i
      in range(len(numbers)) for j
      in range(i + 1, len(numbers)))
```

main* Spaces: 4 UTF-8 LF Python Done

CodeGeeX: Code Explanation

```
example_python.py U ●
1 def quick_sort(array):
2     if len(array) <= 1:
3         return array
4     else:
5         pivot = array[0]
6         less = [i for i in array[1:] if i <= pivot]
7         greater = [i for i in array[1:] if i > pivot]
8         return quick_sort(less) + [pivot] + quick_sort(greater)
9     # check if the list is empty or not
10    if len(array) <= 1:
11        # return the list
12        return array
13    else:
14        # choose the pivot
15        pivot = array[0]
16        # create the left list of values less than the pivot
17        less = [i for i in array[1:] if i <= pivot]
18        # create the right list of values greater than the pivot
19        greater = [i for i in array[1:] if i > pivot]
20        # call the quick sort function on the left list
21        left = quick_sort(less)
22        # call the quick sort function on the right list
23        right = quick_sort(greater)
24        # return the list of sorted lists
25        return left + [pivot] + right
```

代码模型CodeGeeX



开放的大模型研究

 **ChatGLM-6B** Public **ChatGLM对话大模型** 

ChatGLM-6B: An Open Bilingual Dialogue Language Model | 开源双语对话语言模型

 Python  28.3k  3.6k

 **VisualGLM-6B** Public **文图理解大模型** 

Chinese and English multimodal conversational language model | 多模态中英双语对话语言模型

 Python  2.4k  238

 **GLM-130B** Public **千亿大模型** 

GLM-130B: An Open Bilingual Pre-Trained Model (ICLR 2023)

 Python  6.2k  471

 **CodeGeeX** Public **代码大模型** 

CodeGeeX: An Open Multilingual Code Generation Model

 Python  5.6k  387

 **CogVideo** Public **多模态大模型** 

Text-to-video generation. The repo for ICLR2023 paper "CogVideo: Large-scale Pretraining for Text-to-Video Generation via Transformers"

 Python  3k  307

 **CogDL** Public **图神经网络** 

CogDL: A Comprehensive Library for Graph Deep Learning (WWW 2023)

 Python  1.5k  304



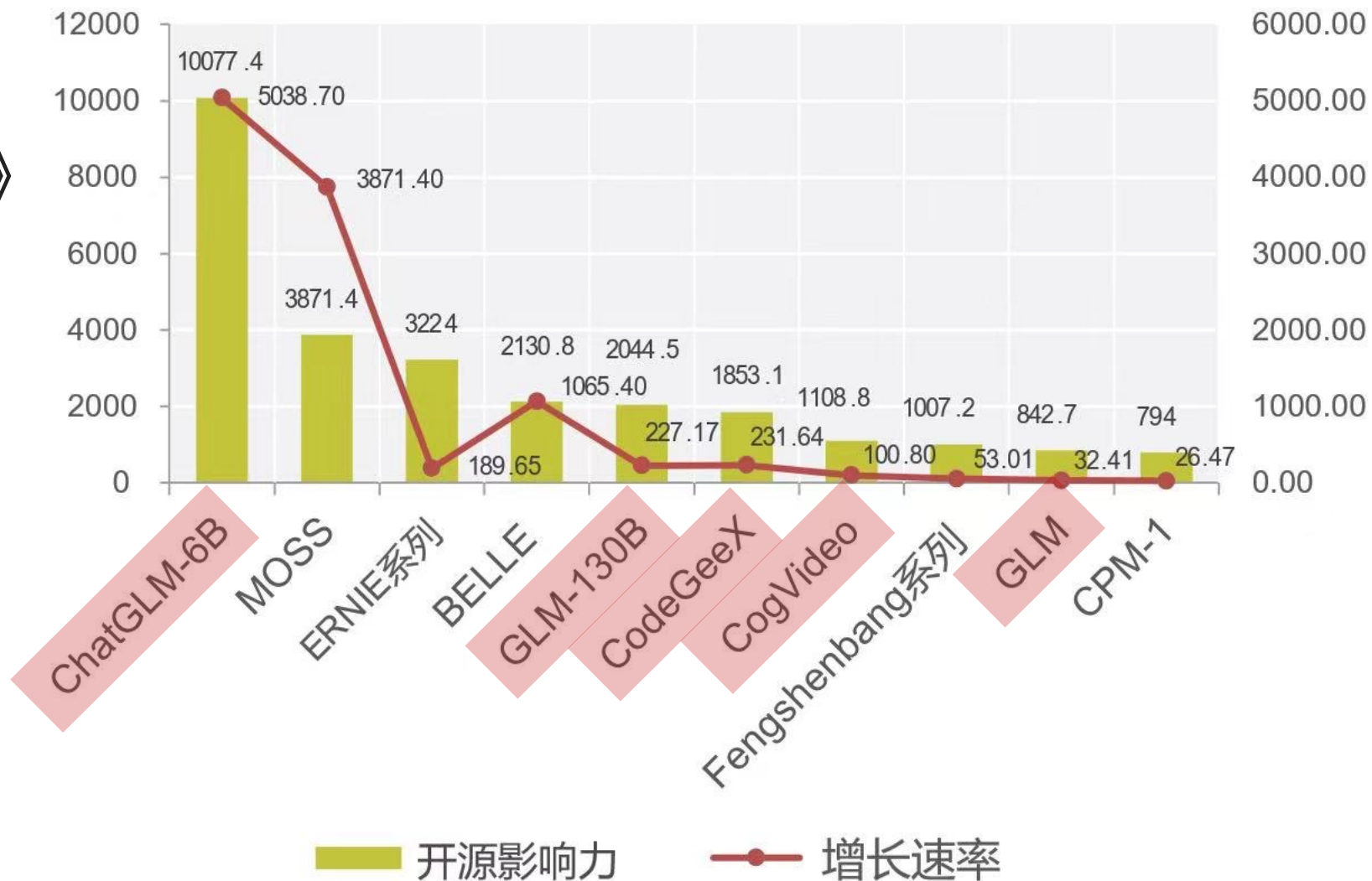
<https://github.com/THUDM>

screenshot at 2023.05.26



开放的大模型研究

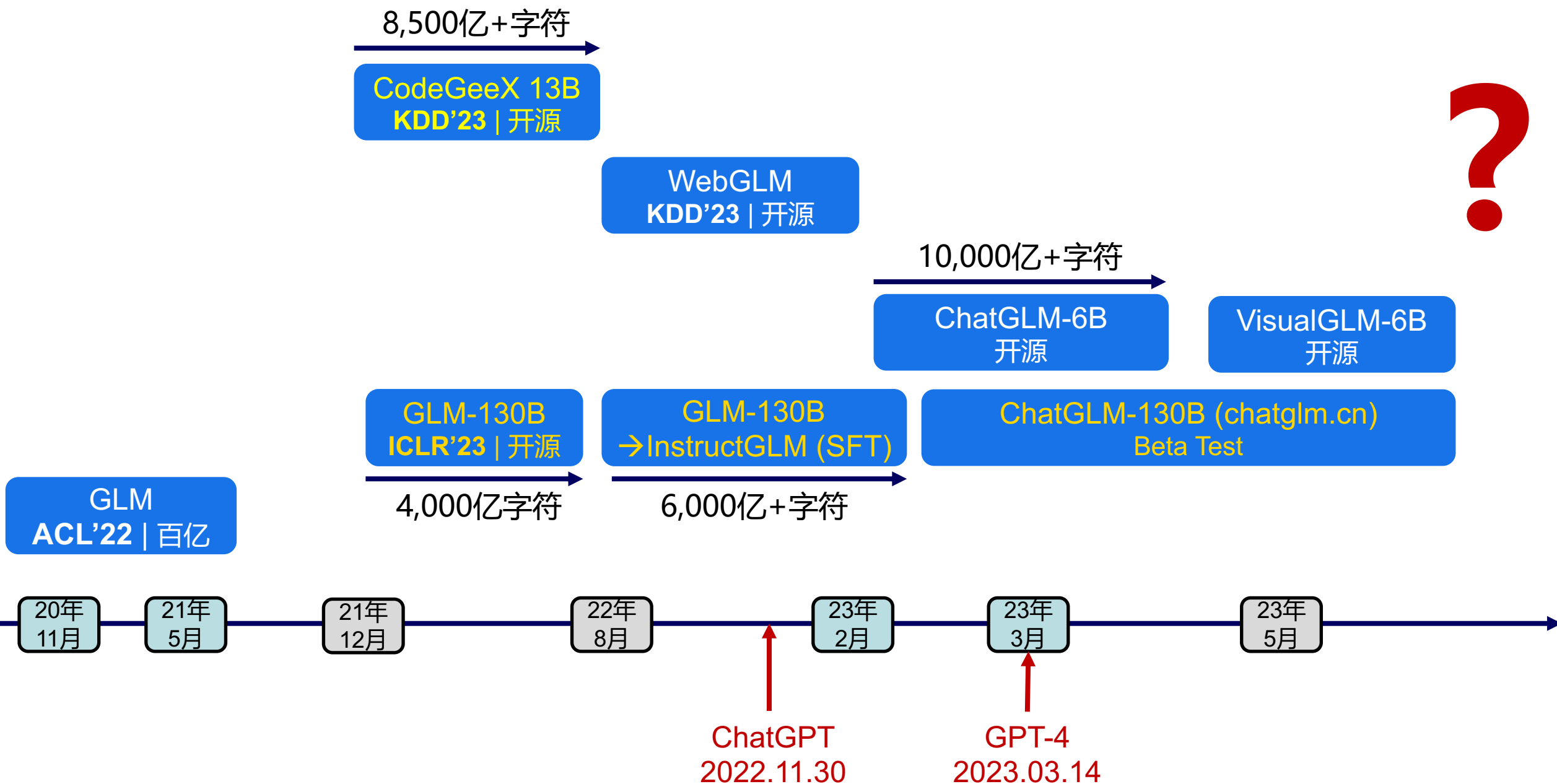
- 2023.05.28，科技部在中关村论坛上发布的《中国人工智能大模型地图研究报告》显示 **ChatGLM-6B** 位列大模型**开源影响力第一名**
- 千亿基座 **GLM-130B**、代码模型 **CodeGeeX**、文生视频模型 **CogVideo**、基础架构模型**GLM**同时入围**开源影响力前十**



ChatGLM 合作伙伴

- [2023.05.26] ChatGLM-6B 全球下载达到**200万**，数百垂直领域模型和国内外应用基于该模型开发
- [2023.05.25] **联想**接入 ChatGLM-130B API 开发智能打印产品
- [2023.05.15] **中国民航信息网络公司**基于接入 ChatGLM-130B API 开发航旅智能产品
- [2023.04.25] **清华研究生会**基于 ChatGLM-130B 开发的【**水木ChatGLM**】上线，服务全校同学
- [2023.04.24] **360**基于 ChatGLM-130B 联合研发千亿级大模型【**360GLM**】
- [2023.04.15] **值得买**部署 ChatGLM-130B 私有化实例用于电商平台产品
- [2023.04.14] **美团**私有化部署 ChatGLM-130B，联合研发【**美团GLM**】
- [2023.04.13] ChatGLM-6B 开源30天内，全球下载量达到75万，GitHub 星标数达到1.7万
- [2023.03.31] ChatGLM-6B 推出基于 P-Tuning-v2 的高效参数微调，最低只需7GB显存即可进行模型微调
- [2023.03.18] ChatGLM-6B 登上 Hugging Face Trending 榜第一，持续12天
- [2023.03.16] ChatGLM-6B 登上 GitHub Trending 榜第一
- [2023.03.14] 千亿对话模型 ChatGLM 开始内测，60亿参数 ChatGLM-6B 模型开源
- [2023.03.10] **竹间智能科技**接入 ChatGLM-130B API 开发智能客服产品

ChatGPT、GPT-4的出现表明我们在大模型上的研究还存在代差



认知大模型探索

□ 解锁 In-context 能力

- 更多训练：大部分大模型都缺少训练

 - A sufficiently-trained LLM could be more powerful than we ever thought

- 更多有针对性的高质量数据、高质量任务（目标函数）

□ 环境交互/自我优化

- 让大模型和用户交互

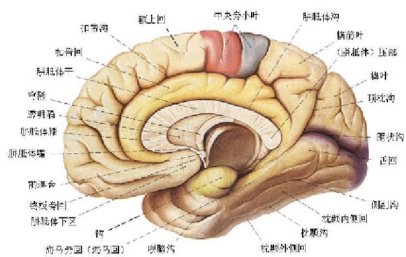
- 让大模型和环境（包括Web）交互

□ 自反思学习

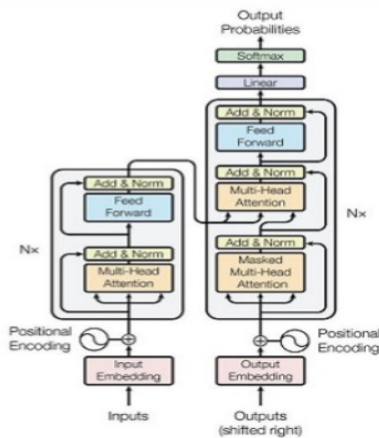
- 大模型的自我反思（self-instruct）

认知大模型探索

查询接口



超大规模预训练



GLM-130B

错误探测与反馈模型

蒸馏与抽象

记忆模型 (知识)

CodeGeex, QA, Web, KG...



“有意识”决策

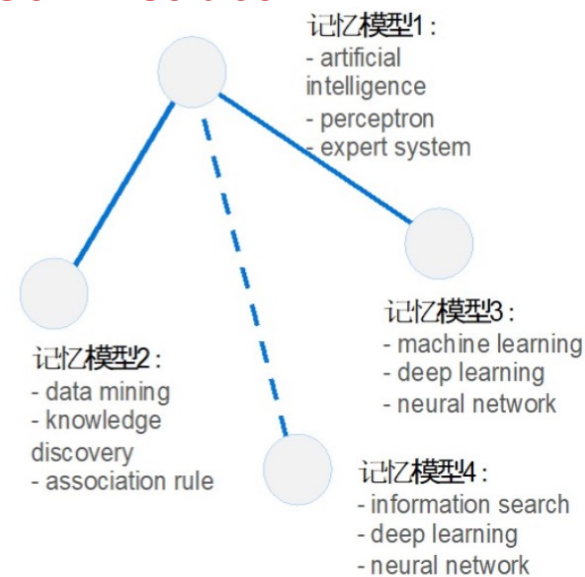
反馈与反思

认知推理

“无意识”探测

自定义条理与逻辑

Self-Instruct



致谢

□ 技术贡献:

- 清华大学知识工程实验室 (KEG)
- 智谱AI
- 清华大学PACMAN实验室
- 清华大学自然语言处理实验室

□ 算力赞助: 智谱·AI

- 前期调试: 中科曙光、鹏城实验室、神威·海洋之光
- 千亿训练: 济南超算中心 (GLM-130B)
- ChatGLM训练: 智谱AI



谢谢大家！



<https://huggingface.co/THUDM>



<https://github.com/THUDM>

