GLM-4: Understanding and Improving LLM Emergent Abilities

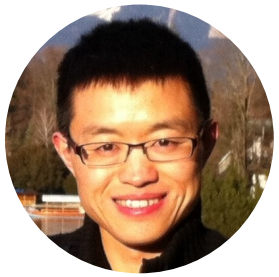
Yuxiao

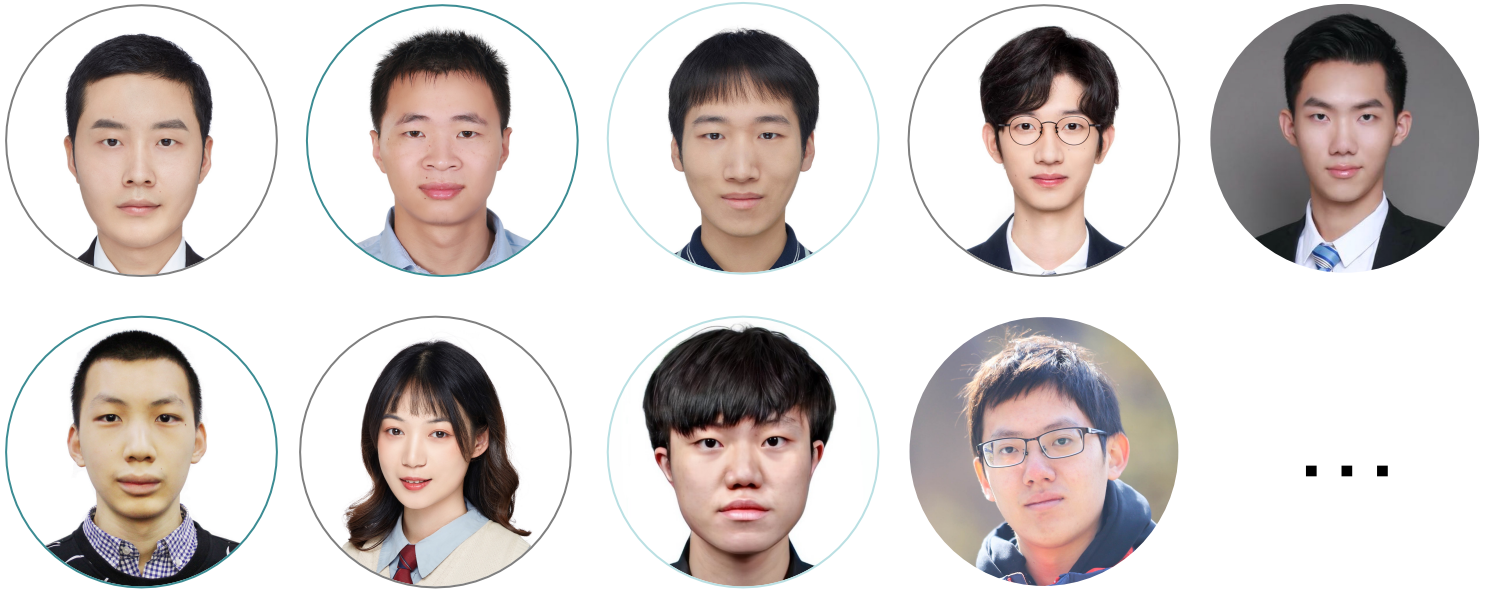
Knowledge Engineering Group (KEG)

Tsinghua CS

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

This talk covers





Jie Tang and many others at **Tsinghua University** and **Zhipu.AI (智谱)**

ChatGLM: A Family of Large Language Models from GLM-130B to GLM-4 All Tools

Team GLM: Aohan Zeng, Bin Xu, Bowen Wang, Chenhui Zhang, Da Yin, Diego Rojas, Guanyu Feng, Hanlin Zhao, Hanyu Lai, Hao Yu, Hongning Wang, Jiadai Sun, Jiajie Zhang, Jiale Cheng, Jiayi Gui, Jie Tang, Jing Zhang, Juanzi Li, Lei Zhao, Lindong Wu, Lucen Zhong, Mingdao Liu, Minlie Huang, Peng Zhang, Qinkai Zheng, Rui Lu, Shuaiqi Duan, Shudan Zhang, Shulin Cao, Shuxun Yang, Weng Lam Tam, Wenyi Zhao, Xiao Liu, Xiao Xia, Xiaohan Zhang, Xiaotao Gu, Xin Lv, Xinghan Liu, Xinyi Liu, Xinyue Yang, Xixuan Song, Xunkai Zhang, Yifan An, Yifan Xu, Yilin Niu, Yuantao Yang, Yueyan Li, Yushi Bai, Yuxiao Dong, Zehan Qi, Zhaoyu Wang, Zhen Yang, Zhengxiao Du, Zhenyu Hou, Zihan Wang

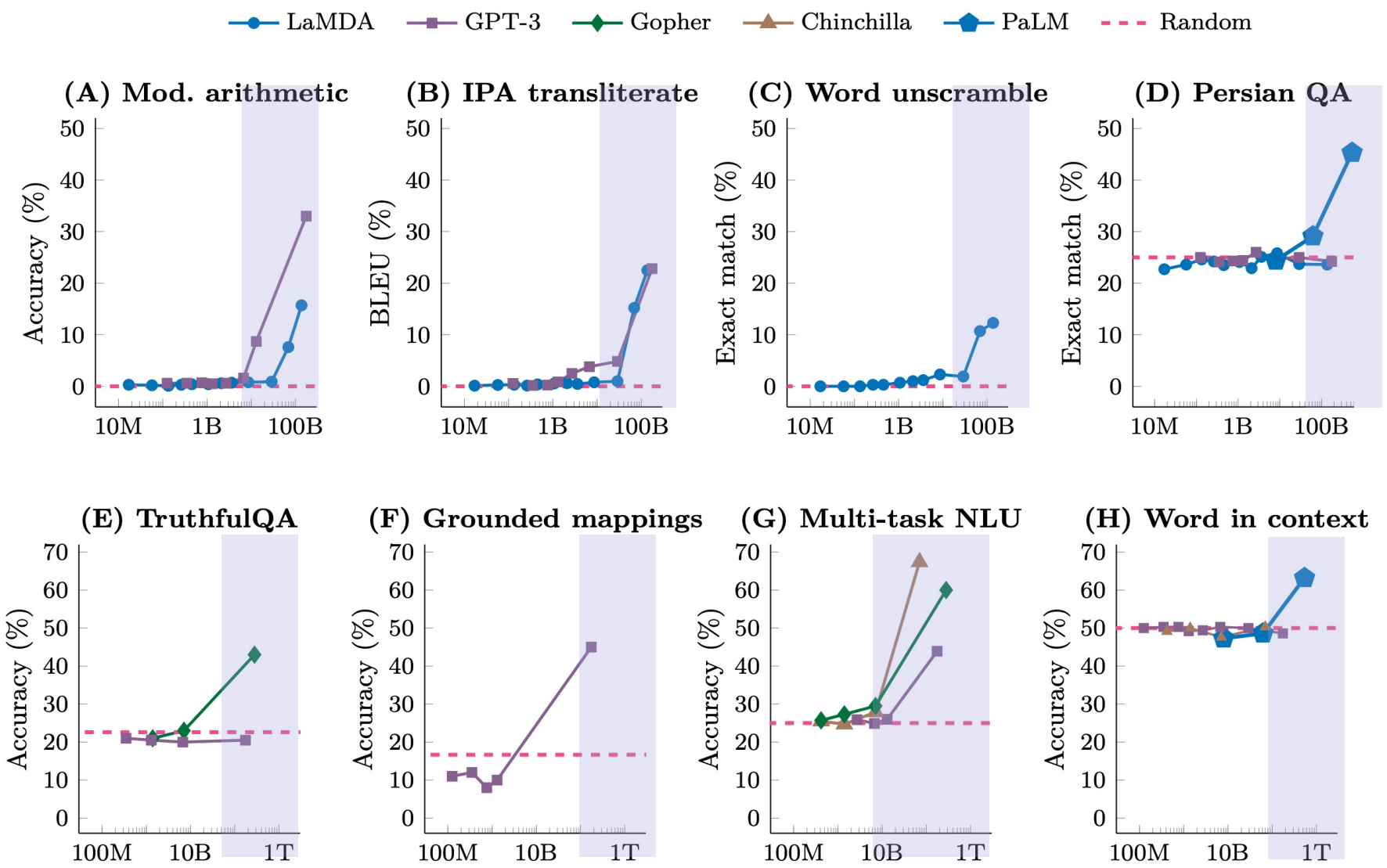
https://arxiv.org/abs/2406.12793

https://bigmodel.cn/

10,000,000 free tokens!

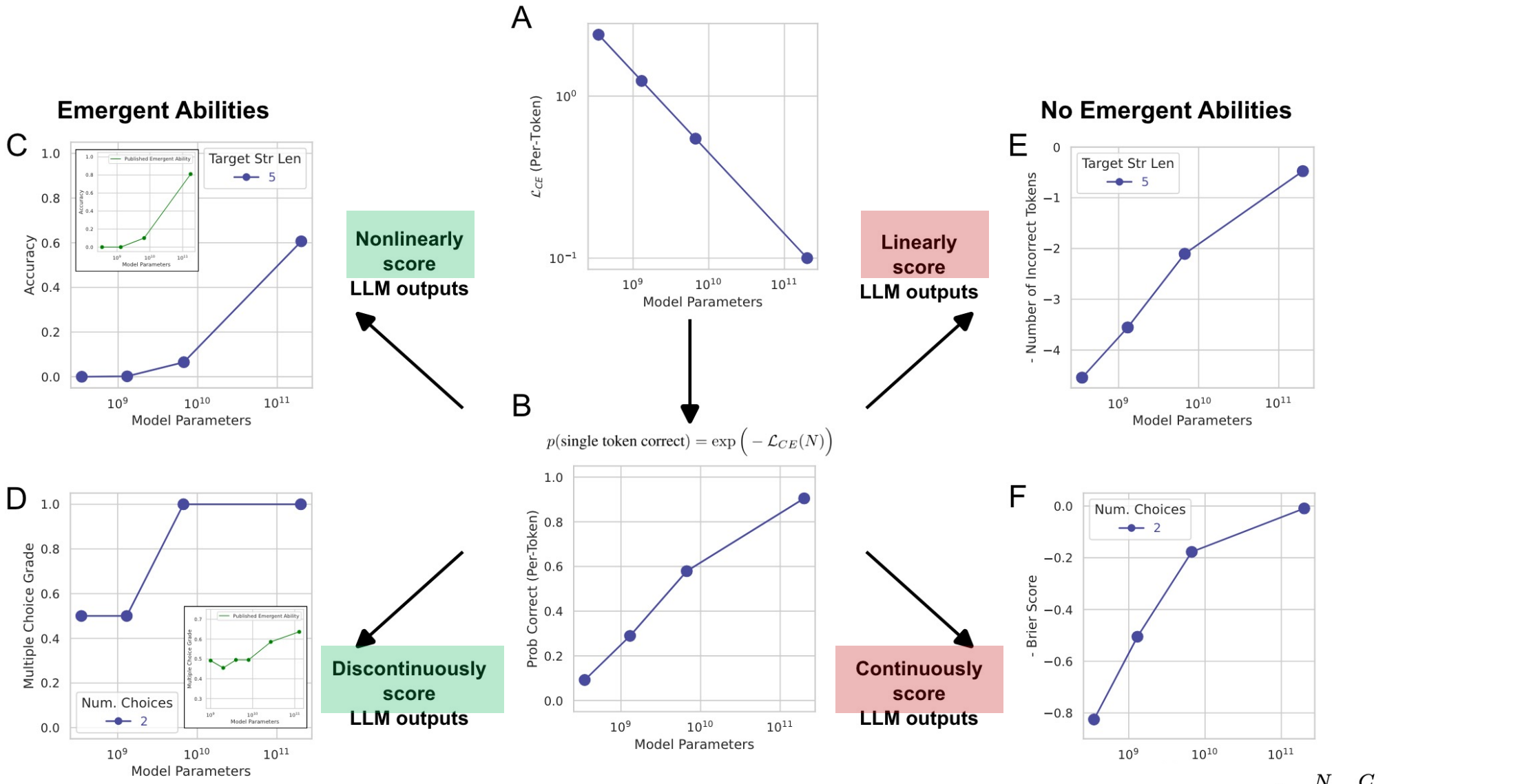
模型	说明
GLM-4-0520	我们当前的最先进最智能的模型，指令遵从能力大幅提升18.6%，具有128k上下文，发布于20240605。
GLM-4V	支持视觉问答、图像字幕、视觉定位、复杂目标检测等各类图像理解任务，具有2k上下文。
GLM-4-AirX	GLM-4-Air 的高性能版本，效果不变，推理速度达到其2.6倍。具有8k上下文。
GLM-4-Air	性价比最高的版本，综合性能接近GLM-4，具有128k上下文，速度快，价格实惠。
GLM-4-Flash	适用简单任务，速度最快，价格最实惠的版本，具有128k上下文。
CogView-3	适用多种图像生成任务，通过对用户文字描述快速、精准的理解，让图像生成更加精确和个性化。

Performance vs. size (model / compute)



1. J Wei, et al. Emergent Abilities of Large Language Models. arXiv: **2206**.07682

“emergent abilities are created by the researcher’s choice of metrics, **not** fundamental changes in model family behavior on specific tasks with scale.”



$$\text{BrierScore} = \frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{C}(y_{ij} - \hat{y}_{ij})^2$$

1. Schaeffer, et al., NeurIPS’23 Best Paper

Why Large Models?

Pre-train 30+ LLMs of varied **model** (7) and **data** (5) sizes from scratch (300M, 540M, 1B, 1.5B, 3B, 6B, 32B)

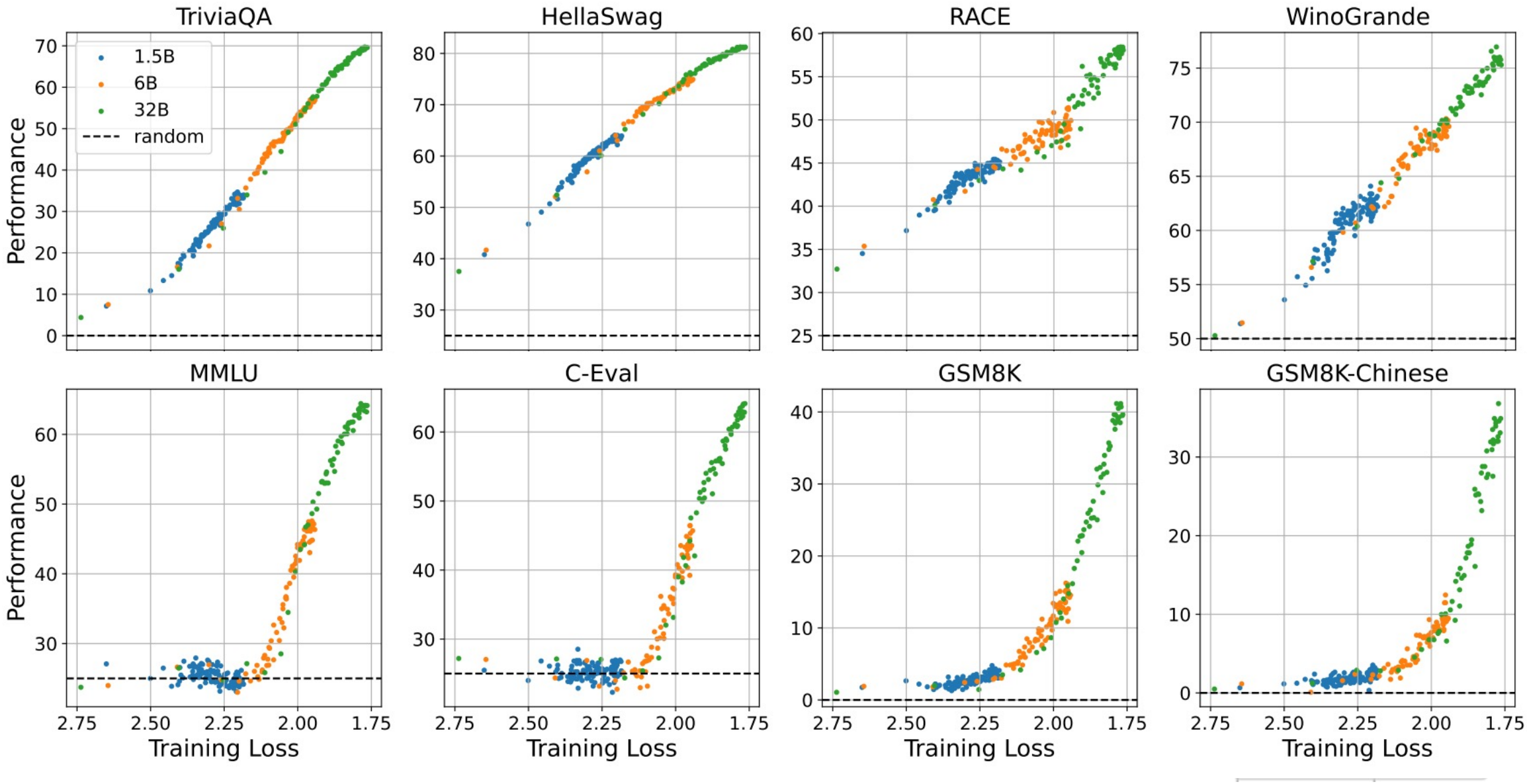
- Fixed data corpus
- Fixed tokenization
- Fixed model architecture

Evaluate downstream performance on 12 diverse datasets

- Different tasks
- Different languages
- Different prompting types
- Different answer forms

1. Z Du et al. Understanding Emergent Abilities of Language Models from the Loss Perspective. **arXiv:2403.15796**

Performance vs. **loss** of 1.5B, 6B, 32B models



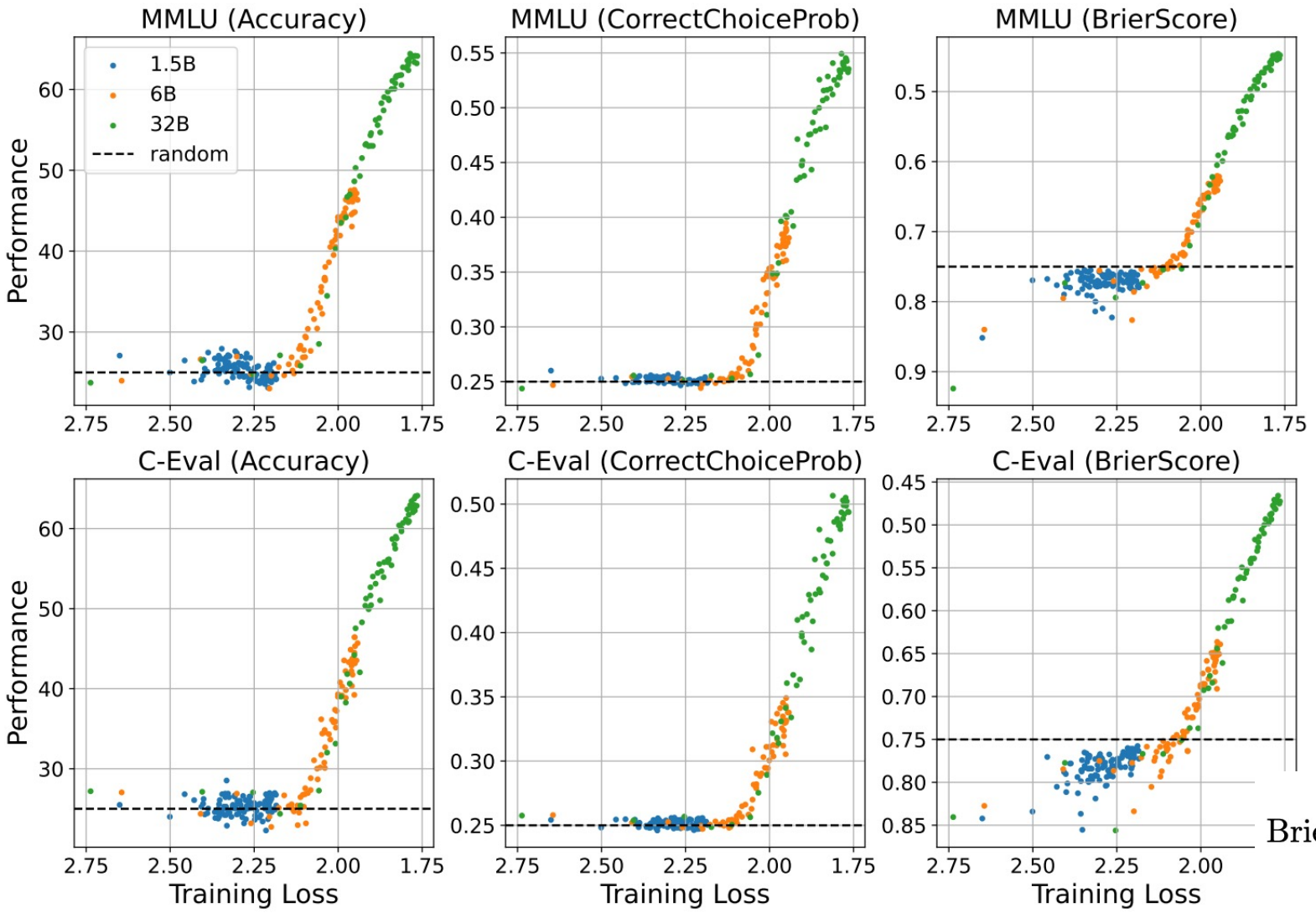
1. Z Du et al. Understanding Emergent Abilities of Language Models from the Loss Perspective. **arXiv:2403.15796**

Performance (different metrics) vs. **loss**

Discontinuously score

Continuously score

Continuously score



$$\text{BrierScore} = \frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{C}(y_{ij} - \hat{y}_{ij})^2$$

1. Z Du et al. Understanding Emergent Abilities of Language Models from the Loss Perspective. **arXiv:2403.15796**

Emergent Abilities defined by **loss**

The normalized performance on an emergent ability as a function of the pre-training loss L is:

$$\begin{cases} f(L) & \text{if } L < \eta \\ 0 & \text{otherwise} \end{cases}$$

Combined with the model scaling law, we can get the normalized performance as a function of the model size N

$$\begin{cases} f\left(L_\infty + \left(\frac{N_0}{N}\right)^{\alpha_N}\right) & \text{if } N \geq N_0(\eta - L_\infty)^{-\frac{1}{\alpha^N}} \\ 0 & \text{otherwise} \end{cases}$$

From this equation, we can explain the observed emergent abilities with model sizes.

1. Z Du et al. Understanding Emergent Abilities of Language Models from the Loss Perspective. **arXiv:2403.15796**

GLM-4 on Academic Benchmarks

Model	MMLU	GSM8K	MATH	BBH	GPQA	HumanEval
GPT-4 (0314)	86.4	92.0	52.9	83.1	35.7	67.0
GPT-4 Turbo (1106)	84.7	95.7	64.3	88.3	42.5	83.7
GPT-4 Turbo (2024-04-09)	86.7	95.6	73.4	88.2	49.3	88.2
Claude 3 Opus	86.8	95.0	60.1	86.8	50.4	84.9
Gemini 1.5 Pro	85.9	90.8	67.7	89.2	46.2	84.1
GLM-4-9B-Chat	72.4	79.6	50.6	76.3	28.8	71.8
GLM-4-Air (0605)	81.9	90.9	57.9	80.4	38.4	75.7
GLM-4 (0116)	81.5	87.6	47.9	82.3	35.7	72.0
GLM-4 (0520)	83.3	93.3	61.3	84.7	39.9	78.5

GLM-4 on Instruction Following (IFEval)

Model	English				Chinese			
	L-P	S-P	L-I	S-I	L-P	S-P	L-I	S-I
GPT-4 (0613)	79.5	77.1	85.5	83.7	72.4	68.9	80.0	75.7
GPT-4 Turbo (1106)	79.1	75.4	85.1	82.4	74.3	69.1	80.8	76.5
GPT-4 Turbo (2024-04-09)	84.5	81.2	88.7	85.9	79.3	72.6	84.2	79.1
Claude 2	75.0	58.0	81.7	67.7	57.1	46.5	64.9	55.1
Claude 3 Opus	90.6	85.5	93.7	90.0	78.3	73.3	84.3	80.4
GLM-4-9B-Chat	73.0	69.0	80.3	77.2	73.0	69.0	80.3	77.2
GLM-4-Air (0605)	80.4	75.2	86.1	82.3	79.3	71.2	84.0	77.3
GLM-4 (0520)	83.7	79.1	88.7	85.0	79.7	71.9	84.2	78.0

Zhou e tal. Instruction following evaluation for large language models. arXiv preprint arXiv:2311.07911, 2023

From GLM-130B to **GLM-4 All Tools**

GLM-130B
MMLU: 44.8%
Aug. 2022

Align with human intent →

ChatGLM
Mar. 2023

GLM-2
MMLU: 66.6%

Better Architecture with **32K Context** →

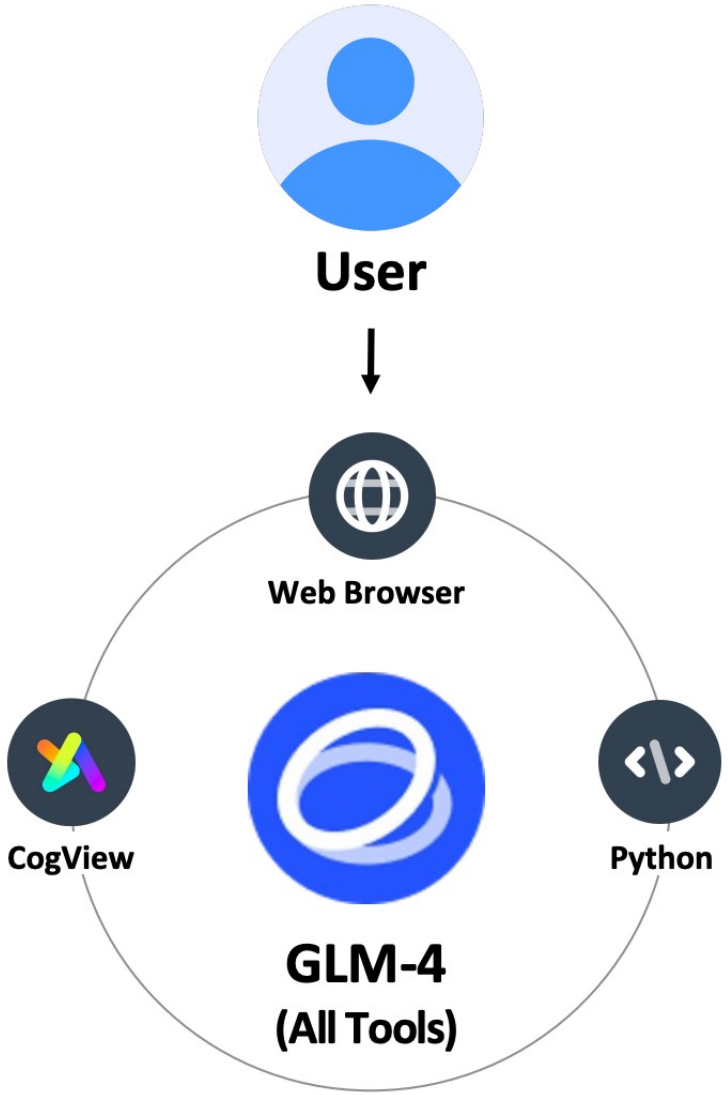
ChatGLM2
Jun. 2023

GLM-3
MMLU: 71.0%

Native **Agent** & **Function Call** Capabilities →

ChatGLM3
Oct. 2023

GLM as the Agent



Search for the global population from 2000 to 2023, then calculate the average annual growth rate.

1 **Web Browser** **Search** **global population from 2000 to 2023**

1 **World Population 1950-2024** https://www.macrotrends.net/...

2 **World Population Clock** https://www.worldometers.info/...

2 **Click** + **Summarize**

According to the information I found, from **2000** to **2023**, the global population grew from about **6.15 billion** to about **8.05 billion...**

3 **Python** **Generate** + **Execute**

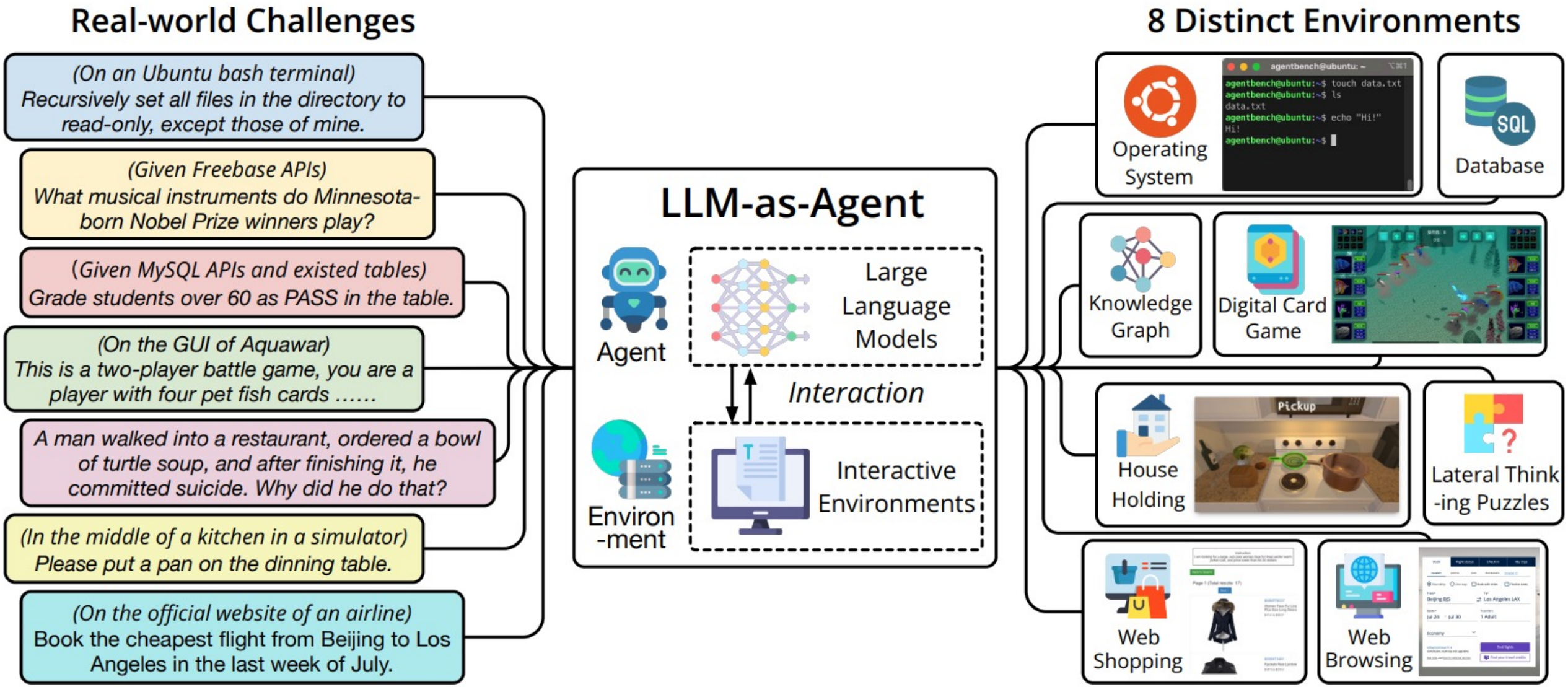
```
# Starting and ending population values
starting_population = 6.15e9  # 2000 population
ending_population = 8.05e9    # 2023 population
# Number of years between 2000 and 2023
years = 2023 - 2000
# CAGR formula
cagr = (ending_population / starting_population) ** (1 / years) - 1
cagr * 100  # Convert to percentage
```

Result: 1.17739919480071

The average annual growth rate of the global population from 2000 to 2023 was approximately **1.18%.**

AgentBench: Evaluate LLMs as Agents

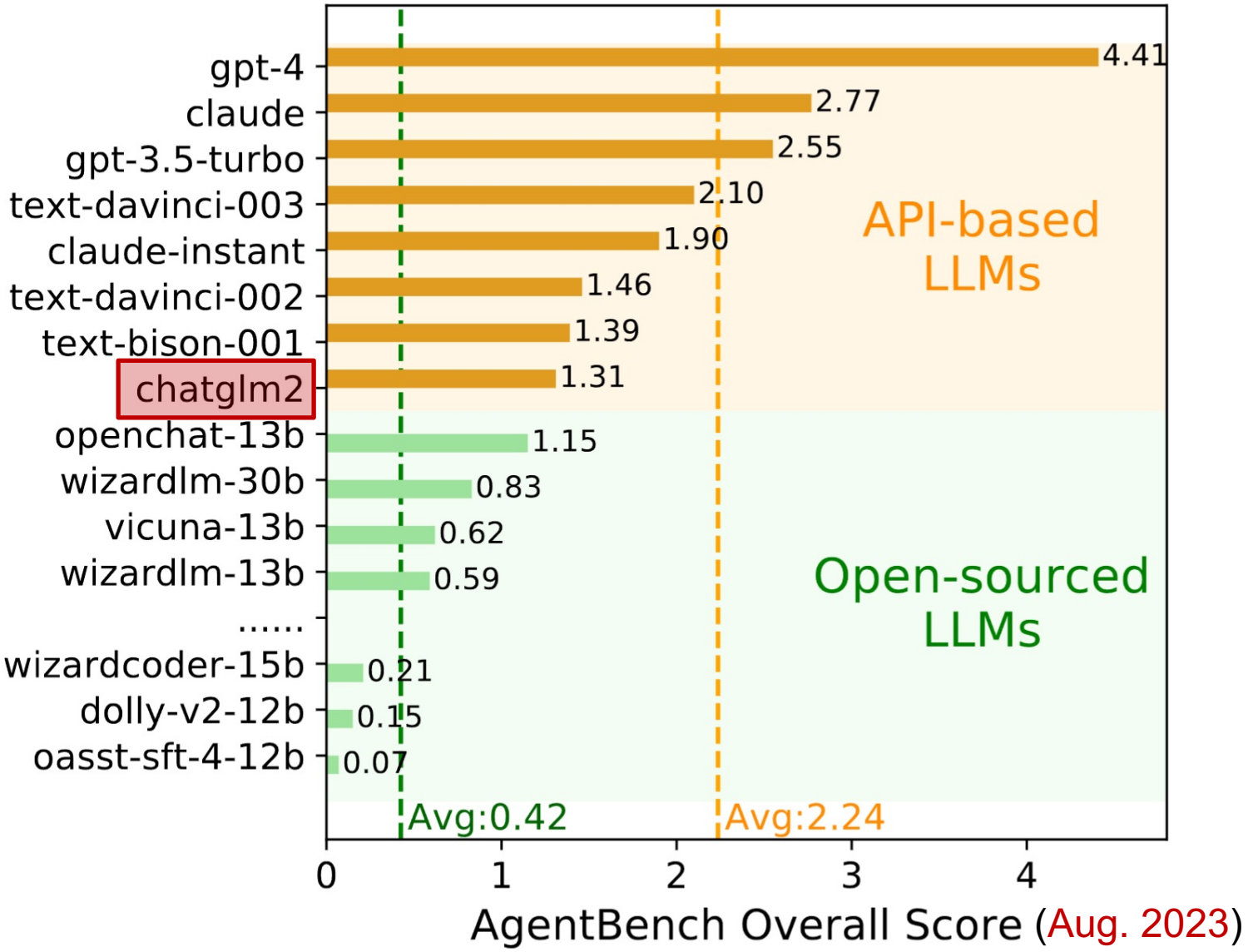
开源



llmbench.ai

AgentBench: Evaluate LLMs as Agents llmbench.ai

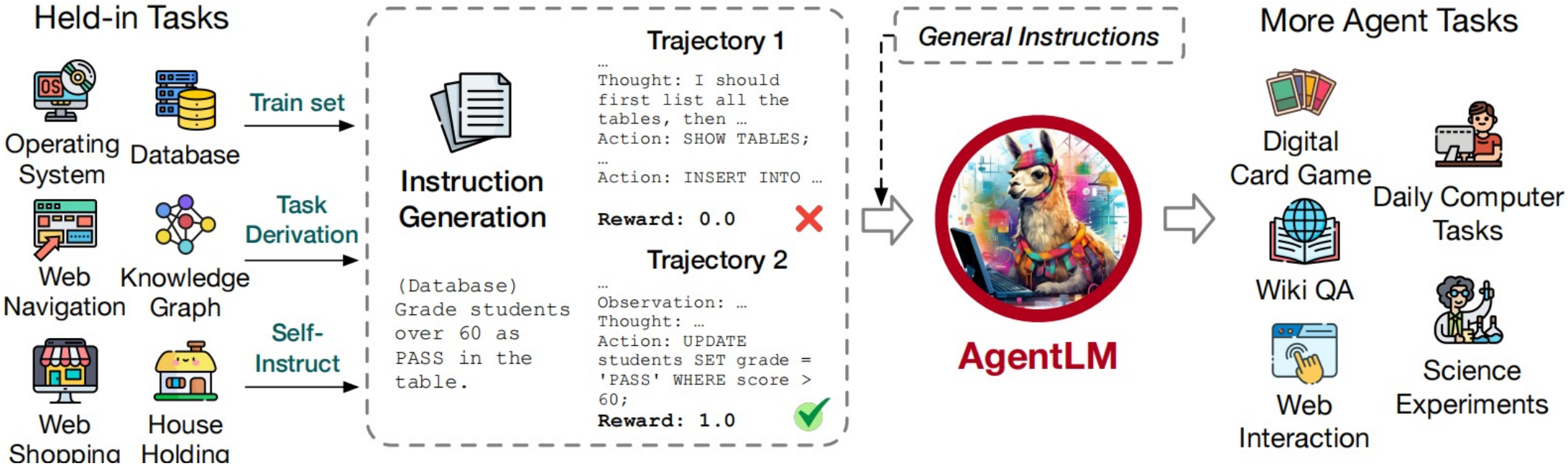
- 多项Agent任务
  1. Operating System
  2. Database
  3. Knowledge Graph
  4. Digital Card Game
  5. Lateral Thinking Puzzles
  6. House-holding
  7. Web Shopping
  8. Web Browsing
- 仅单一Agent任务
  无法泛化到多任务



Liu et al, ***AgentBench**: Evaluating LLMs as Agents*. ICLR 2024. arXiv:2308.03688, 2023. github.com/THUDM/AgentBench

AgentTuning: 少样本激活智能体泛化能力

开源



AgentInstruct 智能体轨迹数据集

- 六项智能体任务
- 使用 Reward 过滤数据
- 仅保留高质量的1866条交互轨迹

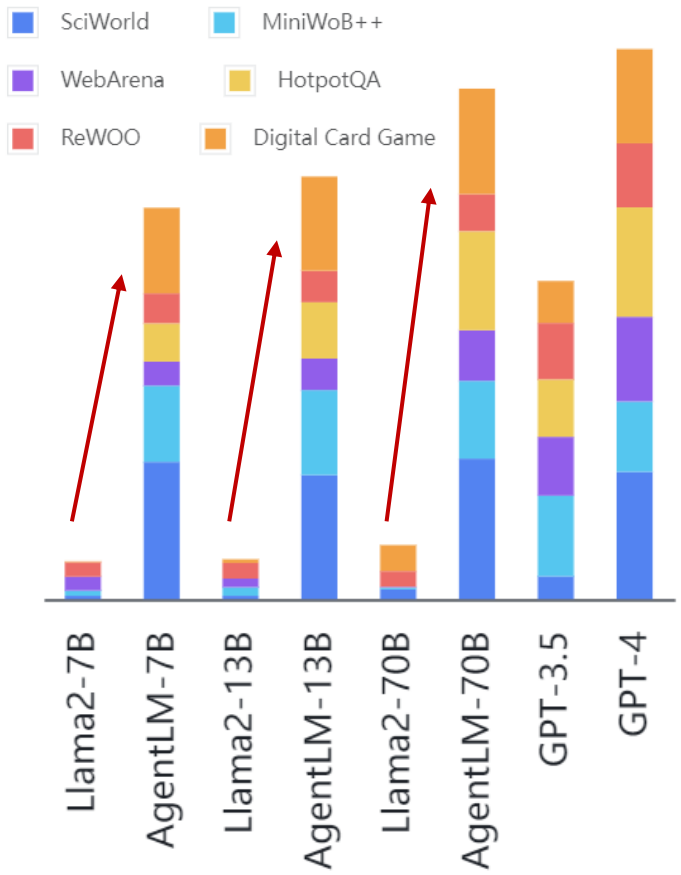
Agent Tuning 混合训练

- 20% AgentInstruct + 80% ShareGPT
- 混合训练后于外分布任务展现泛化性
- 得到 AgentLM-{7B,13B,70B} 系列模型

Zeng et al, AgentTuning: Enabling Generalized Agent Abilities for LLMs. arXiv:2310.12823, 2023.

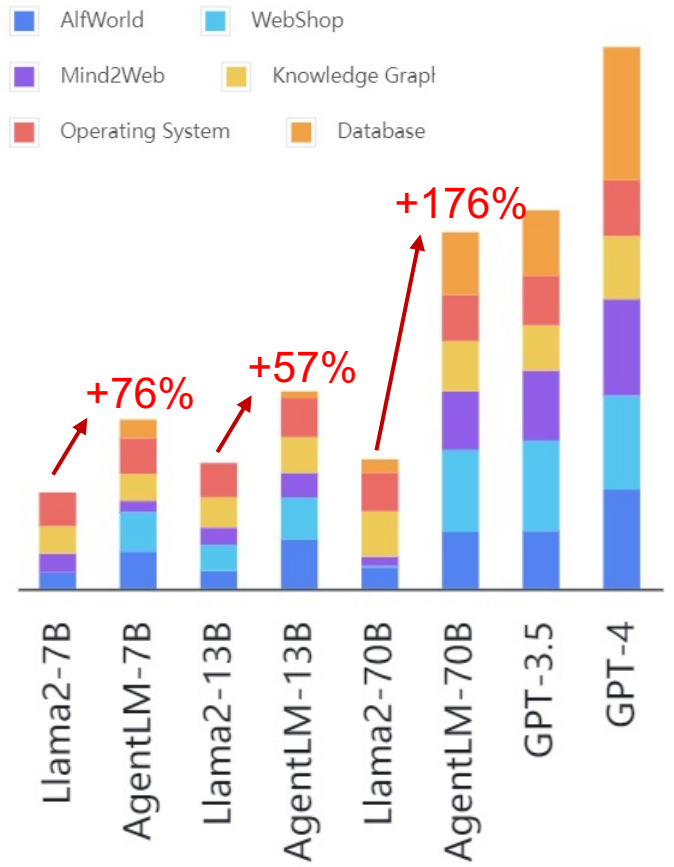
AgentTuning: 少样本激活智能体泛化能力

开源



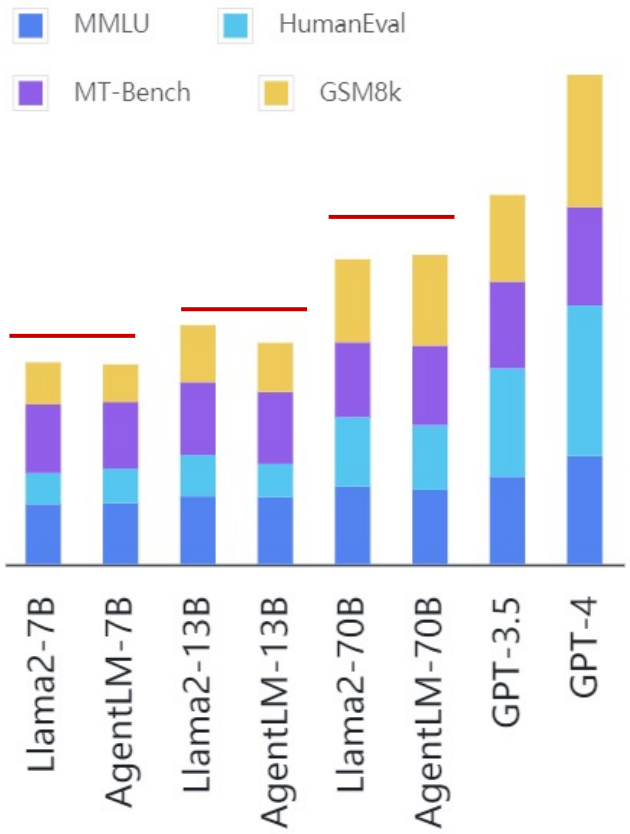
同分布任务

显著提升, 比肩 GPT-4



外分布任务

泛化性良好, 接近 GPT-3.5



通用能力

与微调前表现相当

Zeng et al, AgentTuning: Enabling Generalized Agent Abilities for LLMs. arXiv:2310.12823, 2023.

GLM-4 on Agent Tasks (AgentBench)

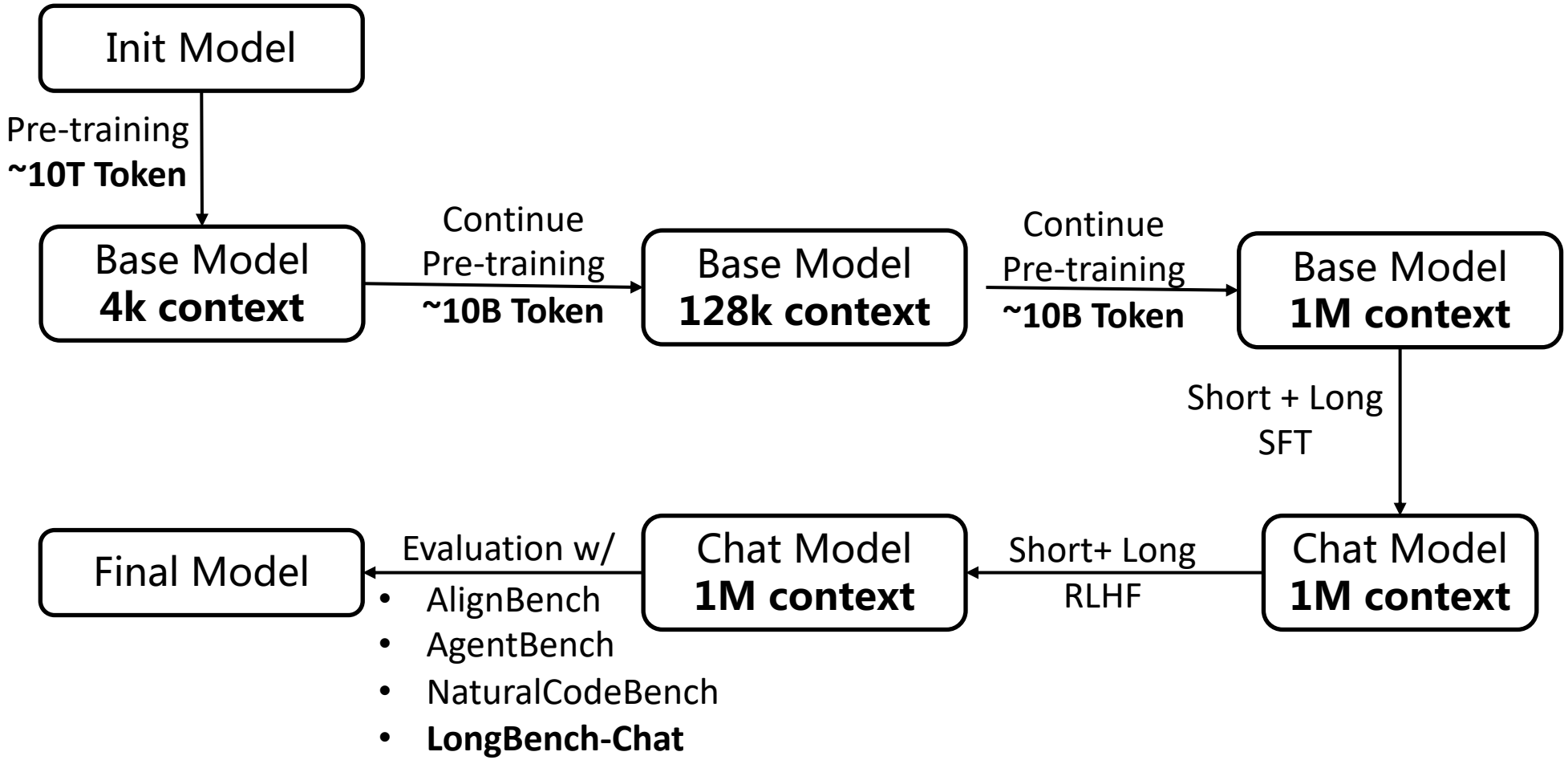
	Operating System	DataBase	Knowledge Graph	Lateral Thinking Puzzles	House Holding	Web Shopping	Web Browsing	Overall
GPT-4 (0613)	42.4	32.0	58.8	16.6	78.0	61.1	29.0	3.69
GPT-4 Turbo (1106)	40.3	52.7	54.0	17.7	70.0	52.8	30.0	3.77
GPT-4 Turbo (2024-04-09)	41.0	46.7	53.2	19.4	72.0	55.1	19.0	3.68
Claude 2	18.1	27.3	41.3	8.4	54.0	61.4	0.0	2.03
Claude 3 Opus	23.6	55.0	53.4	20.0	70.0	48.5	28.0	3.62
GLM-4-Air (0605)	31.9	51.0	53.8	12.3	78.0	69.2	30.0	3.58
GLM-4 (0520)	36.8	52.7	51.4	15.3	82.0	68.3	29.0	3.79

Liu et al, ***AgentBench****: Evaluating LLMs as Agents*. ICLR 2024. arXiv:2308.03688, 2023. github.com/THUDM/AgentBench

GLM-9B-1M-Chat

1. Multi-Stage Training Activates **Long-Context Capability**

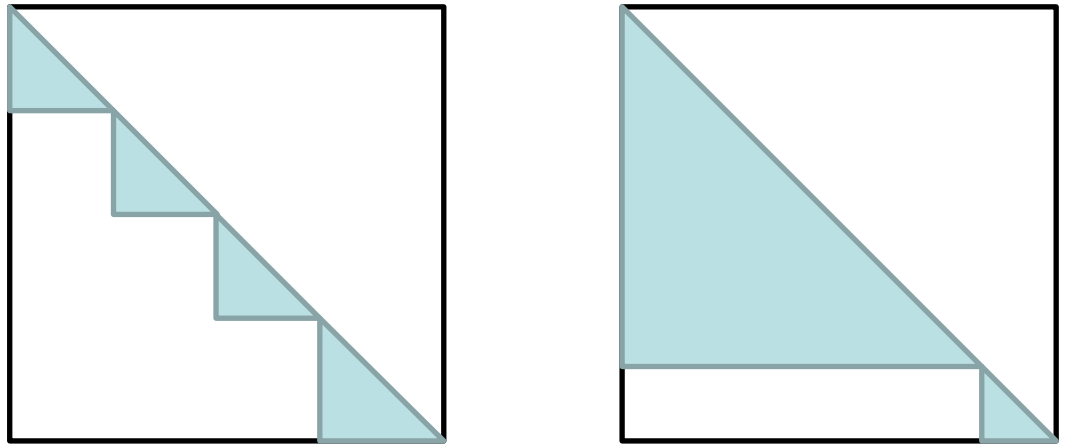
2. Hybrid Training Preserves **Short-Context Capability**



GLM-9B-1M-Chat

- **Synthetic data for long context pre-training and alignment**
  1. **Continue Pre-training**: Merge documents based on Topic and other relevant information.
  2. **Alignment**: Generate longer alignment data based on a shorter model (e.g., 128k model for 256k data).

- **Optimized context parallel for extremely long context training**
  1. Divide-and-conquer context parallel to prevent OOM issues with 1M context.
  2. Balanced varlen training to reduce idle bubble time.

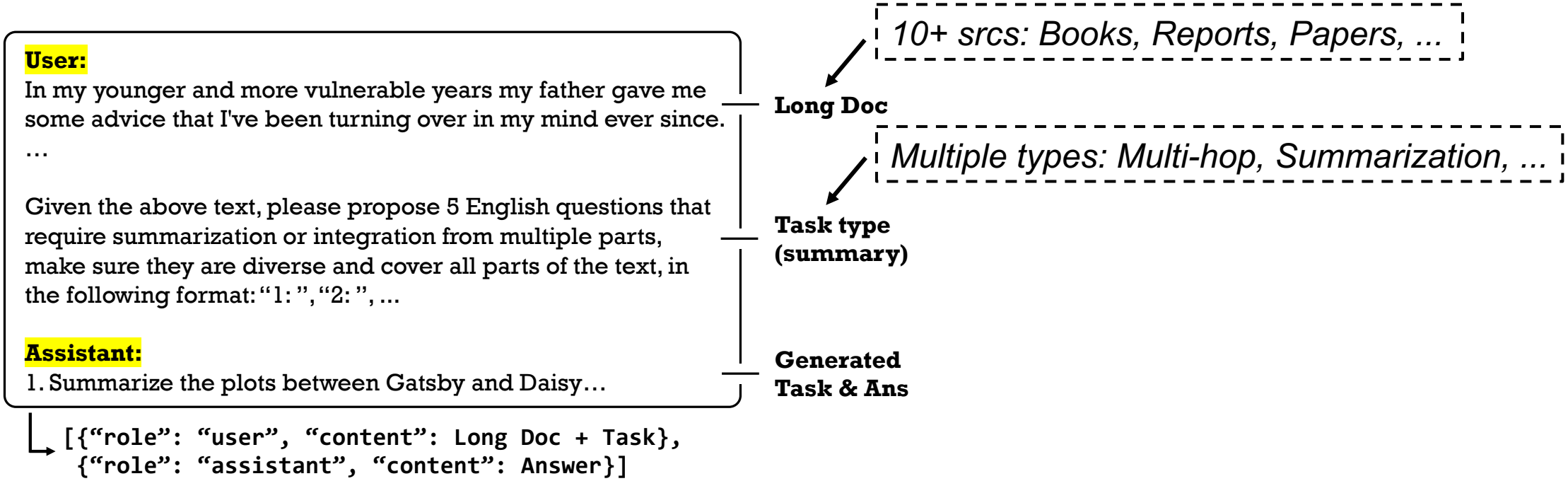


LongAlign: 64K, 128K and 1M context

开源

Instruction fine-tune on long instruction data is crucial! Long context performance down 40% when only fine-tuned on short data (ShareGPT)

- Long Instruction Data Construction (LongAlign-10k)

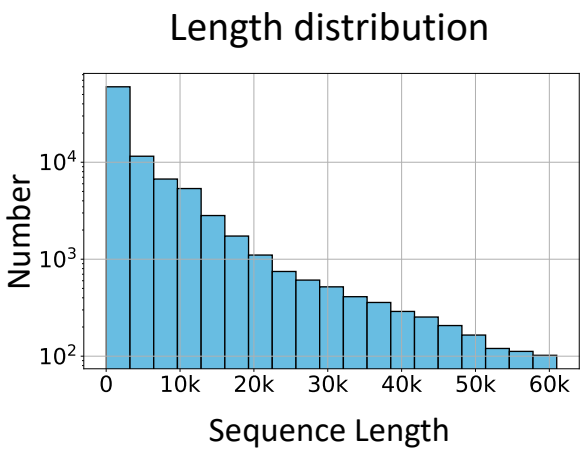


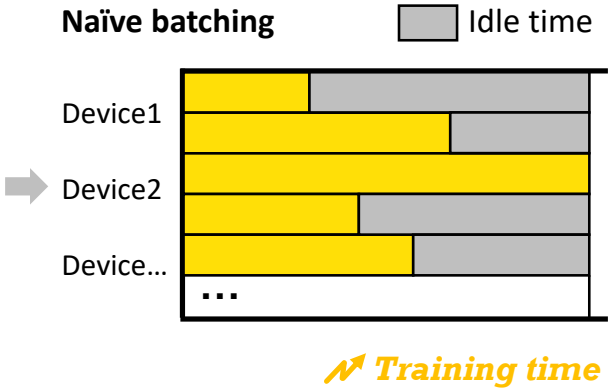
Bai et al, *LongAlign: A Recipe for Long Context Alignment of Large Language Models*. arXiv:2401.18058

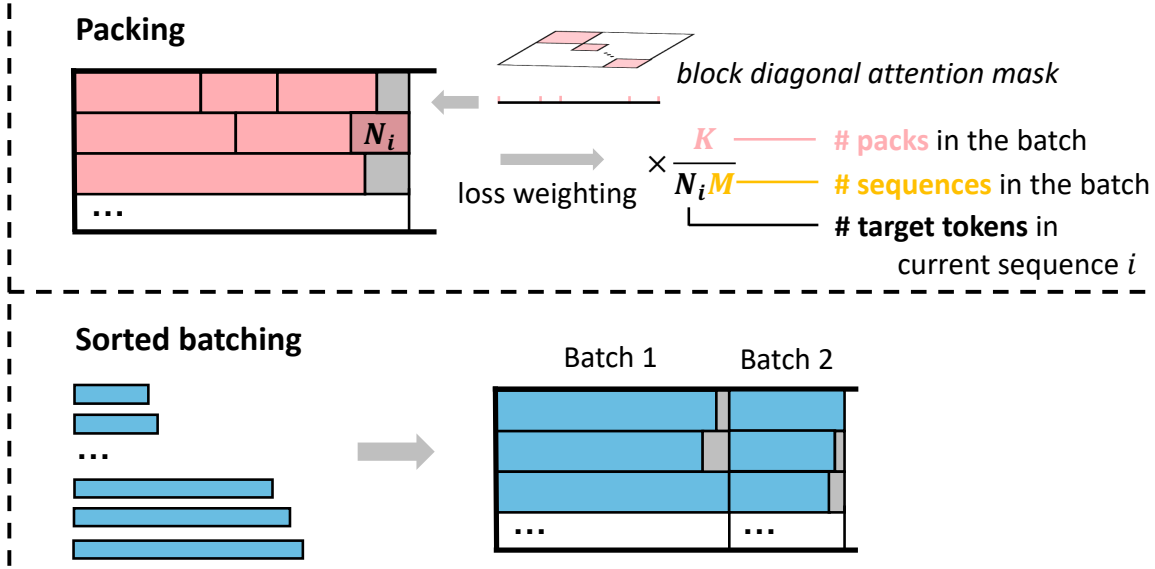
LongAlign: 64K, 128K and 1M context

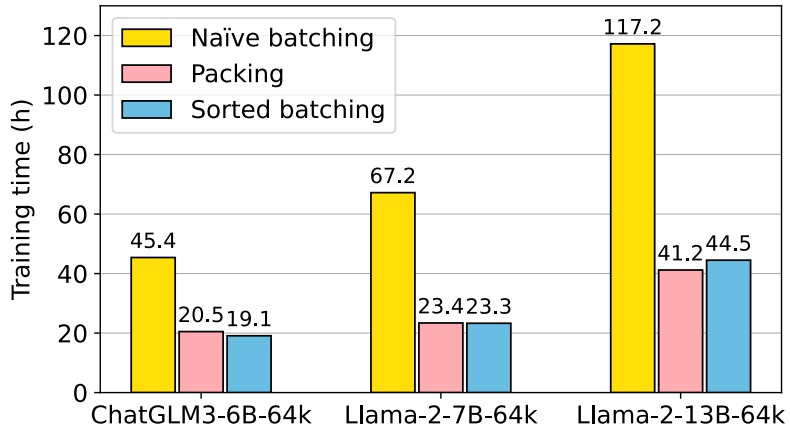
开源

- Efficient training: Packing & Sorted batching + loss weighting









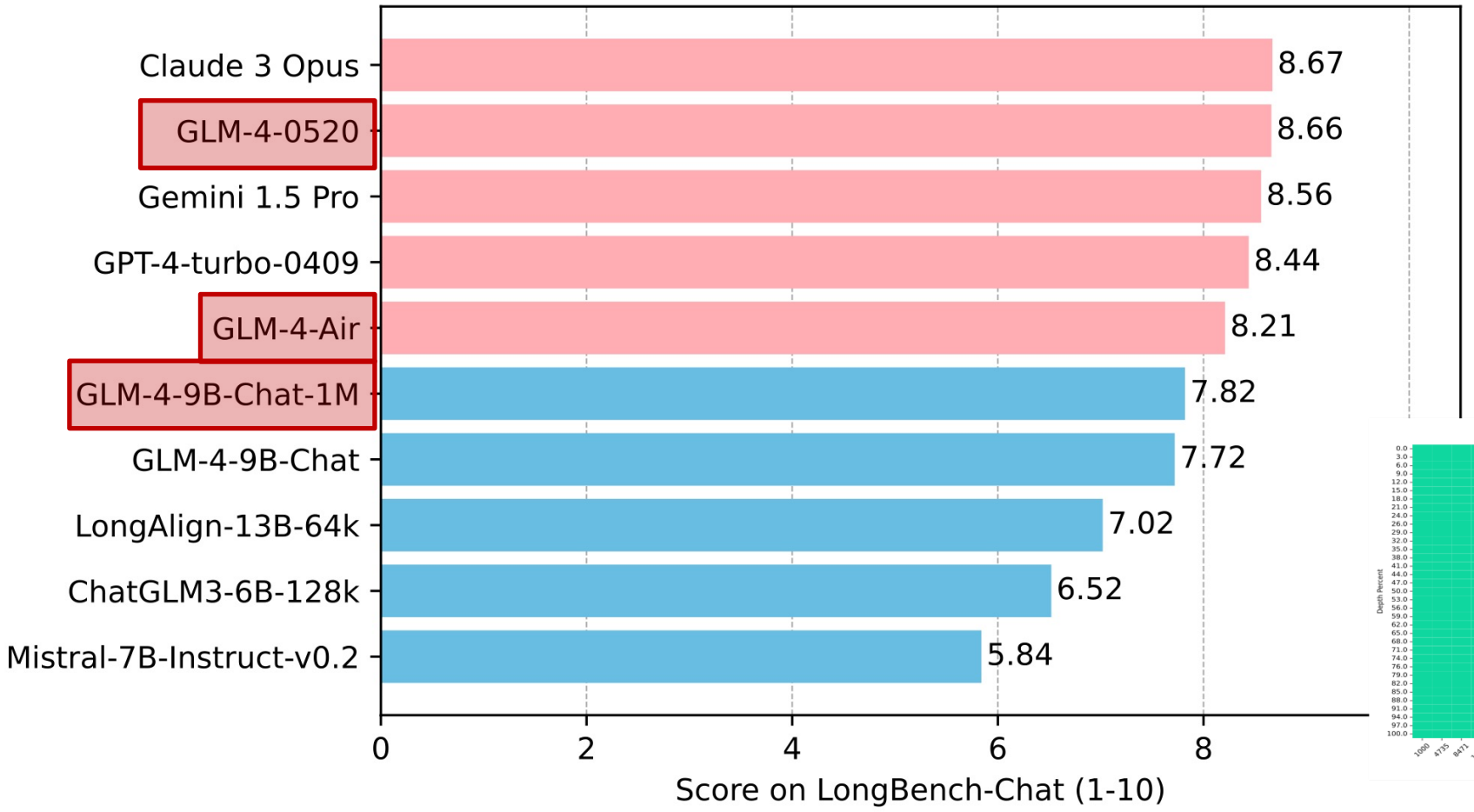
By reducing idle time across samples within the same batch, our two efficient training methods achieve 2～3x speed up

Bai et al, *LongAlign: A Recipe for Long Context Alignment of Large Language Models*. arXiv:2401.18058

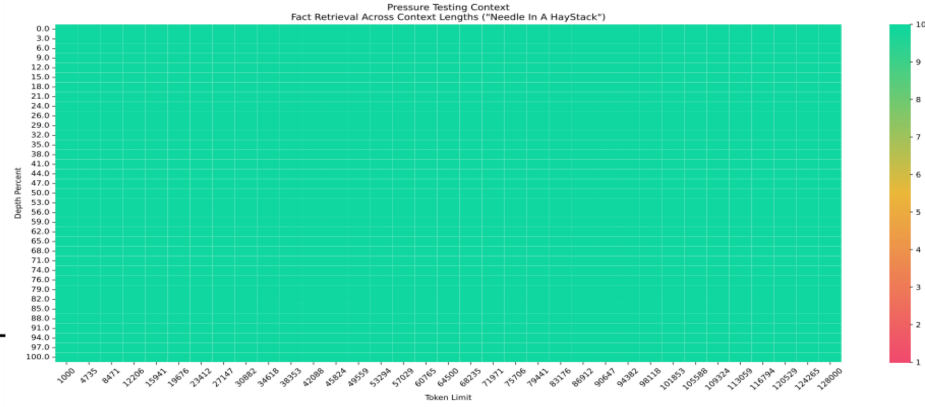
LongAlign: 64K, 128K and 1M context

开源

- LongBench-Chat Benchmark: Real-world queries of 10k-100k length, more challenging and discerning than Needle-in-a-haystack



“Needle in HayStack”



Bai et al, *LongAlign: A Recipe for Long Context Alignment of Large Language Models*. arXiv:2401.18058

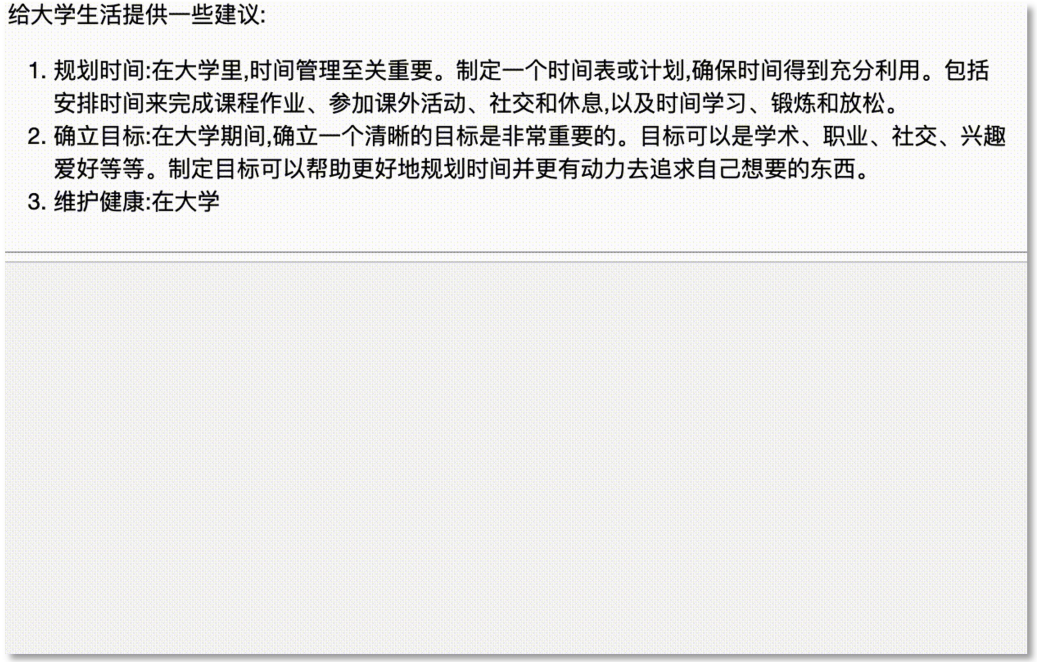
GLM-4 on Long Context Tasks (LongBench)

Model	English	Chinese
GPT-4 Turbo (1106)	87.2	71.4
GPT-4 Turbo (2024-04-09)	85.0	82.1
Claude 2	81.3	76.2
Claude 3 Opus	87.7	82.7
GLM-4-9B-Chat	76.8	79.0
GLM-4-Air (0605)	82.4	81.0
GLM-4 (0520)	87.3	84.0

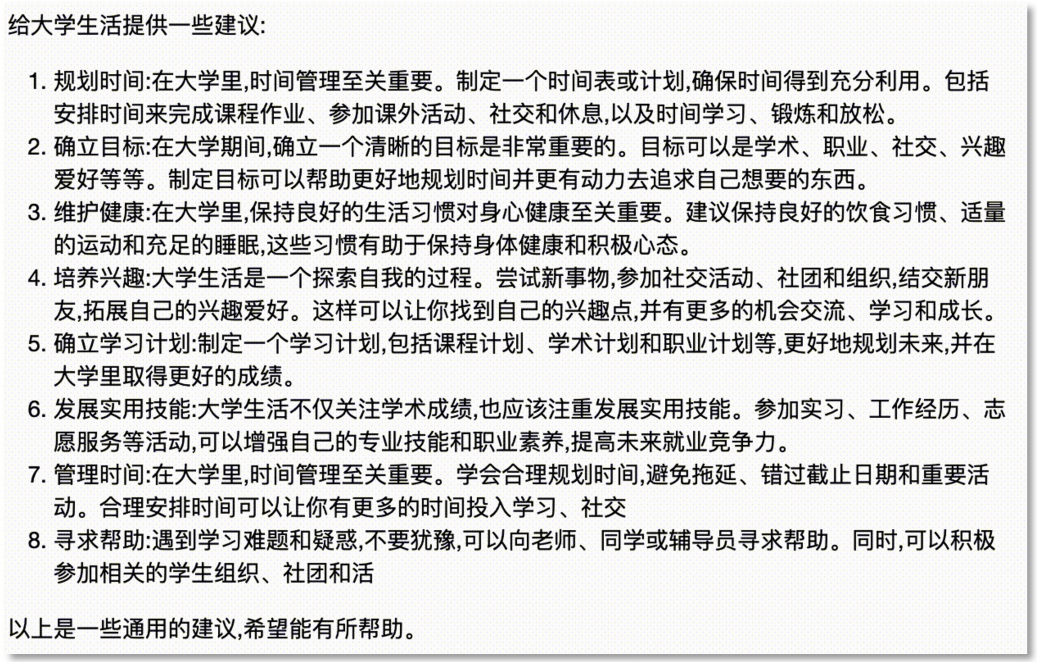
Yushi Bai et al. **LongBench**: A Bilingual, Multitask Benchmark for Long Context Understanding. ACL 24

Inference: Auto-Parallel Auto-Regressive (APAR)

自回归（AR）



自并行自回归（APAR）

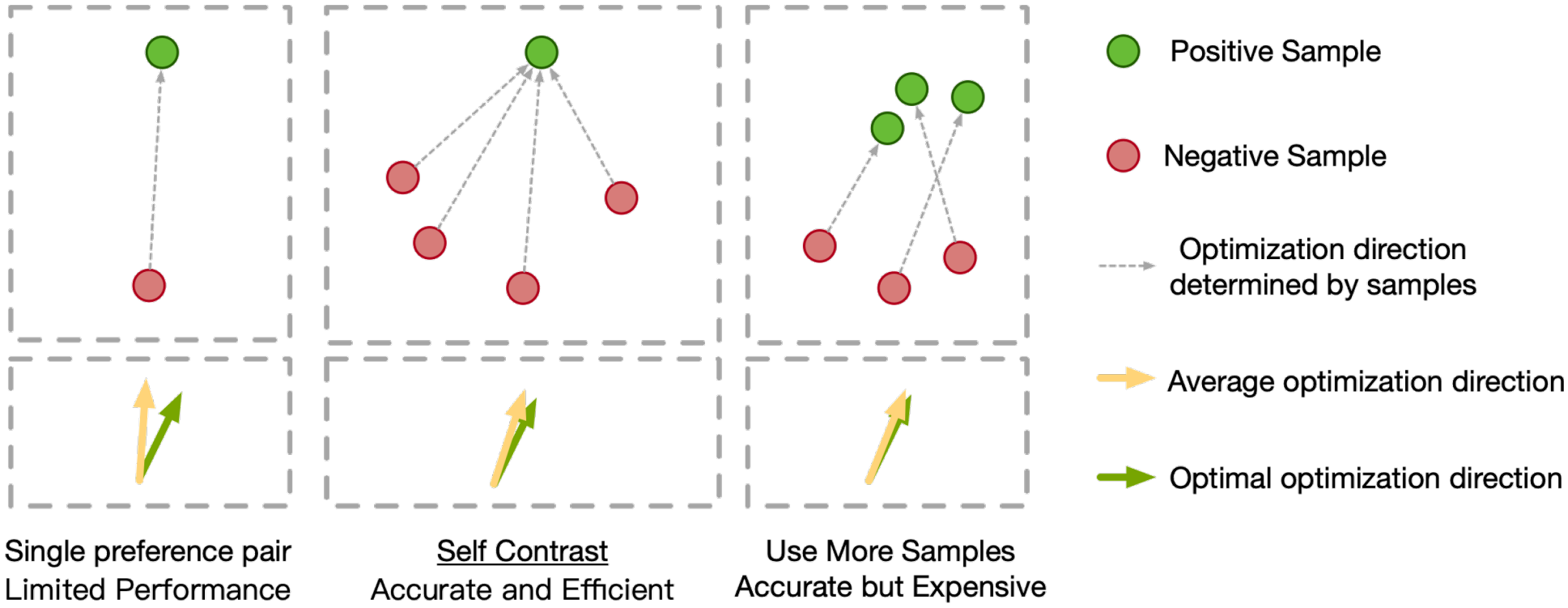


Liu et al, APAR: LLMs Can Do Auto-Parallel Auto-Regressive Decoding.arXiv:**2401.06761**, 2024.

Self-Contrast: Feedback-Free Alignment

Obtaining preference data pairs is costly. In contrast, negative samples are massively cheap to obtain.

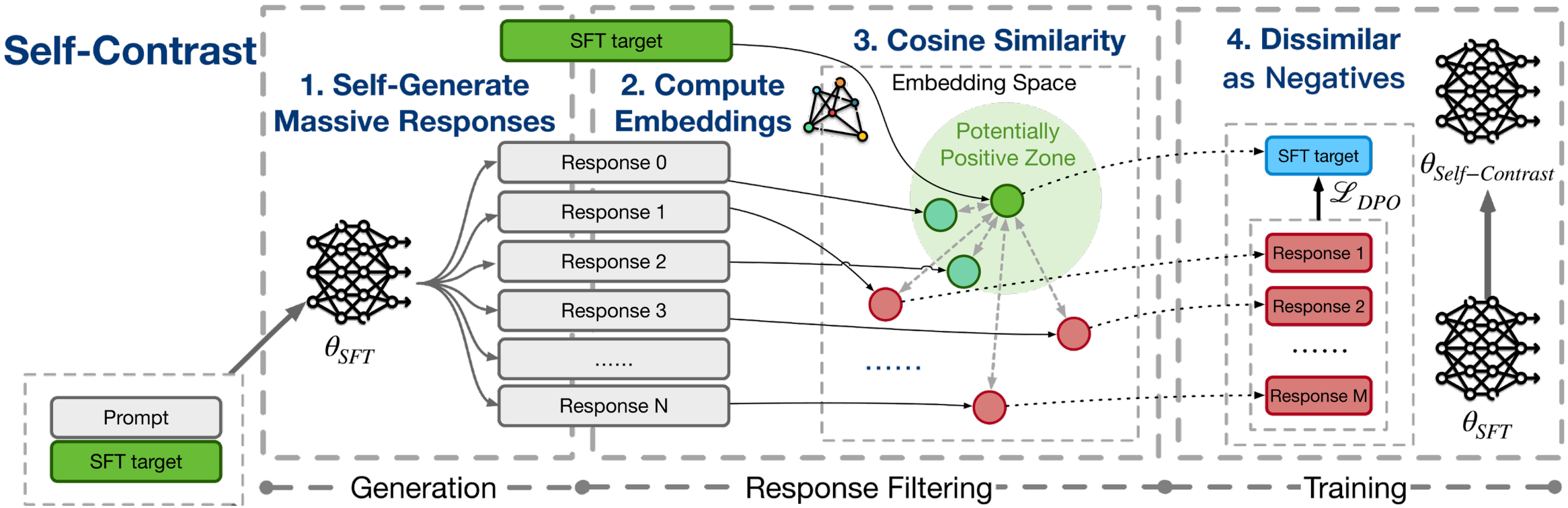
- Adding negative samples is equivalent to adding positive samples



Liu et al, Extensive Self-Contrast Enables Feedback-Free Language Model Alignment. arXiv:2404.00604

Self-Contrast: Feedback-Free Alignment

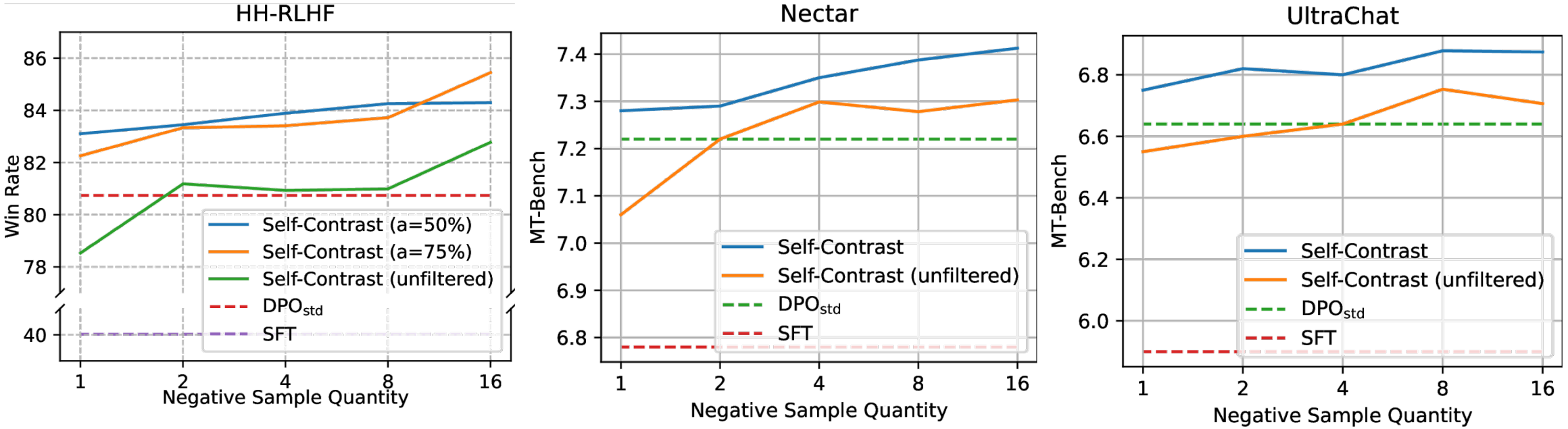
- Alignment using self-generated negatives without preference labeling.



Liu et al, Extensive Self-Contrast Enables Feedback-Free Language Model Alignment. arXiv:2404.00604

Self-Contrast: Feedback-Free Alignment

- Alignment using self-generated negatives without preference labeling.

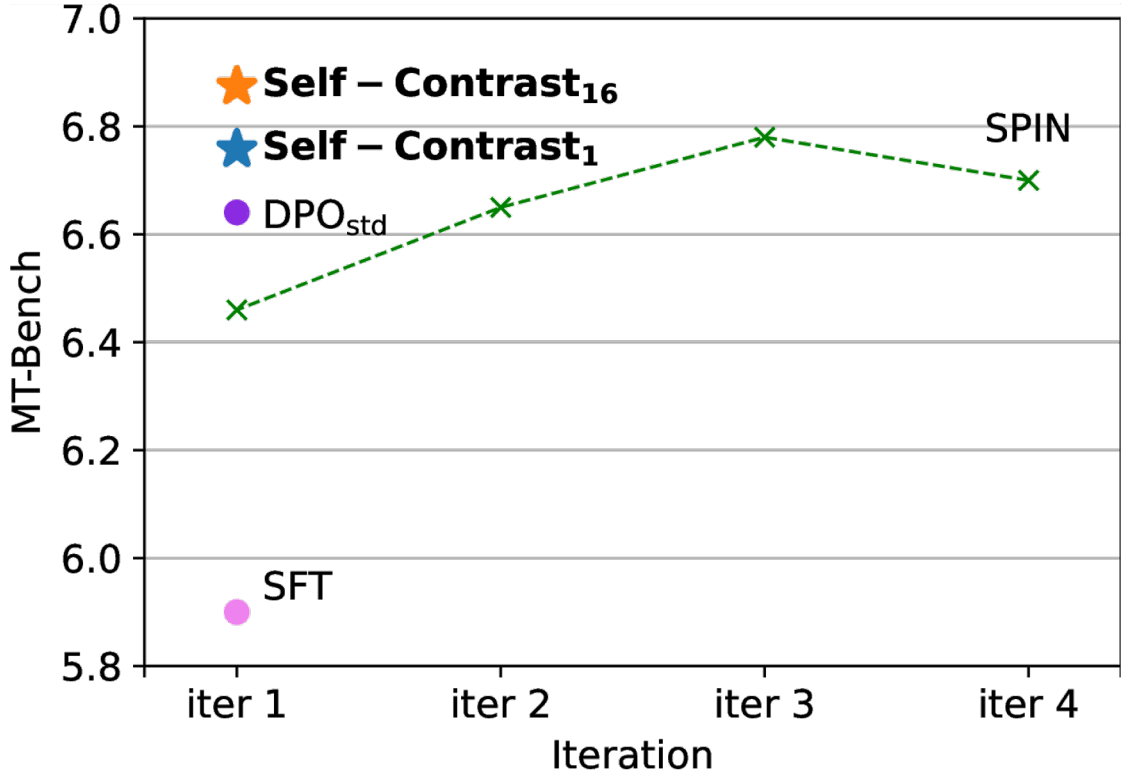


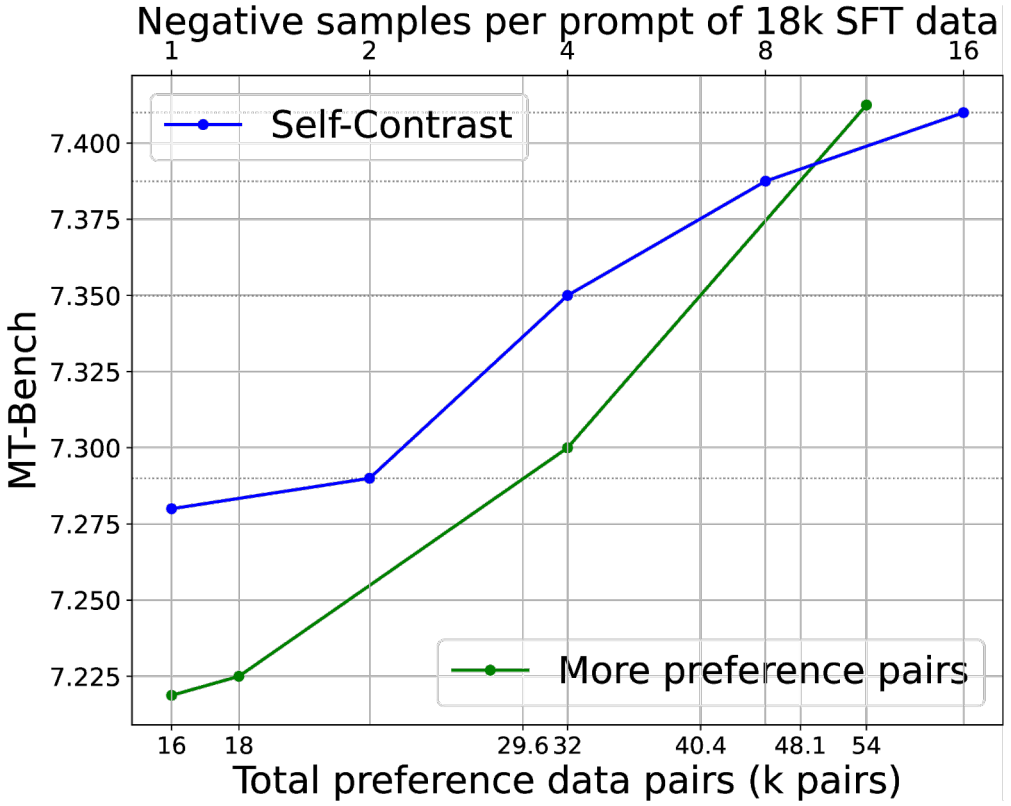
Continuous performance improvement by adding negative samples only.

Liu et al, Extensive Self-Contrast Enables Feedback-Free Language Model Alignment. arXiv:2404.00604

Self-Contrast: Feedback-Free Alignment

- By increasing the number of negative samples and applying filtering, Self-Contrast outperforms standard DPO and iteration-based methods.





Increasing negative samples matches the performance boost of more preference pairs.

Liu et al, Extensive Self-Contrast Enables Feedback-Free Language Model Alignment. arXiv:2404.00604

GLM-4 on Alignment (AlignBench)

Model	Math	Logic	Language	Chinese	QA	Writing	Role Play	Professional	Overall
GPT-4 (0613)	7.54	7.17	7.82	7.02	7.39	7.67	8.20	7.29	7.46
GPT-4 Turbo (1106)	7.85	7.66	7.90	7.22	8.24	8.53	8.46	7.95	7.90
GPT-4 Turbo (2024-04-09)	8.32	7.67	7.60	7.57	8.37	7.75	8.18	8.59	8.00
Claude 2	6.39	5.85	6.75	5.72	6.68	5.87	6.86	6.56	6.26
Claude 3 Opus	7.27	7.11	7.94	7.71	8.21	7.61	7.73	8.02	7.53
Gemini 1.5 Pro	7.07	7.77	7.31	7.22	8.55	7.83	7.79	8.52	7.47
GLM-4-9B-Chat	7.00	6.01	6.69	7.26	7.97	7.59	8.10	7.52	7.01
GLM-4-Air (0605)	7.69	6.95	7.53	8.00	7.90	8.01	8.35	8.09	7.65
GLM-4 (0116)	7.20	7.20	7.60	8.19	8.45	7.88	8.05	8.56	7.66
GLM-4 (0520)	7.89	7.95	8.00	7.86	8.11	8.04	8.06	8.47	8.00

Liu et al., **AlignBench**: Benchmarking Chinese Alignment of Large Language Models. arXiv:2311.18743

From GLM-130B to **GLM-4 All Tools**

GLM-130B
MMLU: 44.8%
Aug. 2022

Align with human intent → **ChatGLM** (Mar. 2023)

GLM-2
MMLU: 66.6%

Better Architecture with **32K Context** → **ChatGLM2** (Jun. 2023)

GLM-3
MMLU: 71.0%

Native **Agent** & **Function Call** Capabilities → **ChatGLM3** (Oct. 2023)

GLM-4
MMLU: 81.7%

• LongAlign • AgentTuning →

GLM-4
（All Tools）
Jan. 2024

GLM-4 9B
（1M | Open）
Jun. 2024

GLM-4 All Tools

- **融合多种工具的 Agent 形态，无需手动指定工具，自动触发**
  - Browser, **Code Interpreter**, Text2Image (CogView3)

x^3+ax^2−5x+9除以x+4，商为x^2+bx−1，余数为13，计算 a, b 的值。

要解决这个问题，我们可以使用多项式除法的原理。我们知道，当一个多项式 $P(x)$ 被另一个多项式 $D(x)$ 整除时，可以得到一个商 $Q(x)$ 和一个余数 $R(x)$。在这种情况下，我们有：

$$P(x) = D(x) \times Q(x) + R(x)$$

已知条件是 $P(x) = x^3 + ax^2 - 5x + 9$，$D(x) = x + 4$，$Q(x) = x^2 + bx - 1$，且余数 $R(x) = 13$。

根据多项式除法的原理，我们可以将 $P(x)$ 表达为 $D(x)$ 和 $Q(x)$ 的乘积加上余数 $R(x)$：

$$x^3 + ax^2 - 5x + 9 = (x + 4)(x^2 + bx - 1) + 13$$

我们可以通过展开右侧的乘积并将结果与左侧的多项式进行比较来找到 a 和 b 的值。接下来，我将进行计算。

代码生成：已完成

结果

{a: 3, b: -1}

根据计算结果，a 和 b 的值分别是 3 和 -1。因此，多项式 $x^3 + ax^2 - 5x + 9$ 除以 $x + 4$，其商为 $x^2 + bx - 1$，余数为 13 时，$a = 3$，$b = -1$。

代码生成：已完成

python 复制

```python
from sympy import symbols, Eq, solve

# 定义变量
x, a, b = symbols('x a b')

# 已知的多项式
P_x = x**3 + a*x**2 - 5*x + 9
D_x = x + 4
Q_x = x**2 + b*x - 1
R_x = 13

# 构造等式 P(x) = D(x) * Q(x) + R(x)
equation = Eq(P_x, D_x * Q_x + R_x)

# 展开等式右侧
expanded_equation = equation.expand()

# 解方程求 a 和 b 的值
solutions = solve(expanded_equation, (a, b))
solutions
```

结果

{a: 3, b: -1}

GLM-4 All Tools

- **融合多种工具的 Agent 形态，无需手动指定工具，自动触发**
  - **Browser**, **Code Interpreter**, Text2Image (CogView3)

“请查一下全球过去十多年的GDP情况，并画出趋势图，如果哪年GDP下降，在趋势中用红色标出来。”

查询 1 次，打开 1 个网页并总结

联网查询：global GDP data recent years

打开网页

World GDP 1960–2024 | MacroTrends

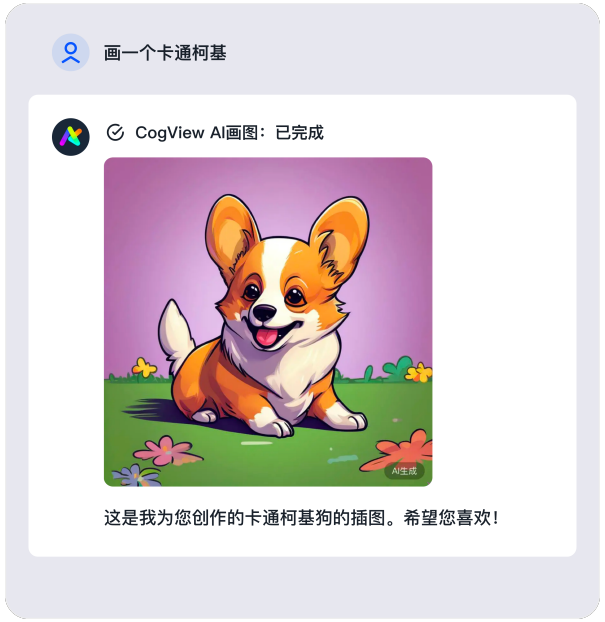
https://www.macrotrends.net/countries/WLD/world/g...

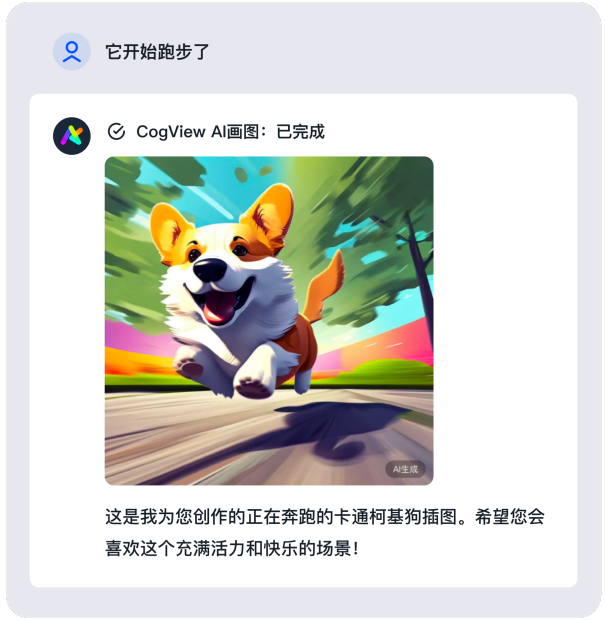
读取网页并总结

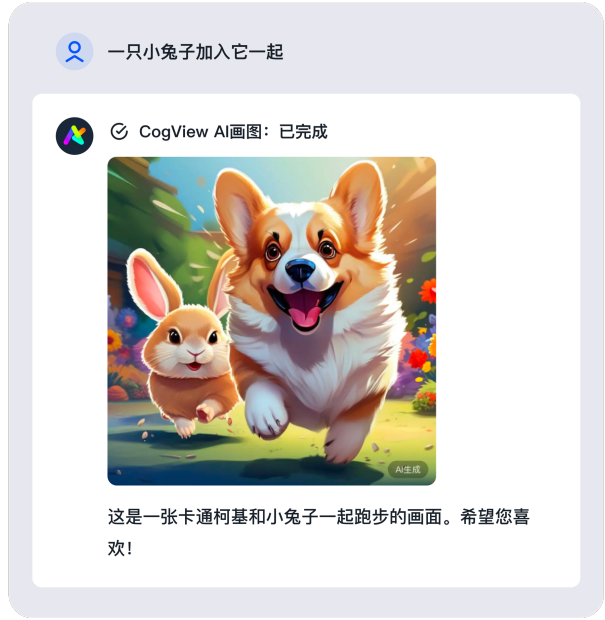
```python
import matplotlib.pyplot as plt
# GDP data for the years 2007 to 2022
years = ...
gdp_values = ...
# Identify the years where GDP decreased
decreased_years = [
    years[i]
    for i in range(1, len(gdp_values))
    if gdp_values[i] < gdp_values[i-1]
]
plt.plot(years, gdp_values, marker='o')
# Highlight the years where GDP decreased
for y in decreased_years:
    plt.plot(y, gdp_values[years.index(y)], 'ro')
...
plt.show()
```

GLM-4 All Tools

- **融合多种工具的 Agent 形态，无需手动指定工具，自动触发**
  - Browser, Code Interpreter, **Text2Image (CogView3)**







Zheng et al., **CogView3**: Finer and Faster Text-to-Image Generation via Relay Diffusion. arXiv:2403:05121
Yang et al., **Inf-DiT**: Upsampling Any-Resolution Image with Memory-Efficient Diffusion Transformer. arXiv:2405.04312

Zheng et al., **CogView3**: Finer and Faster Text-to-Image Generation via Relay Diffusion. arXiv:2403:05121

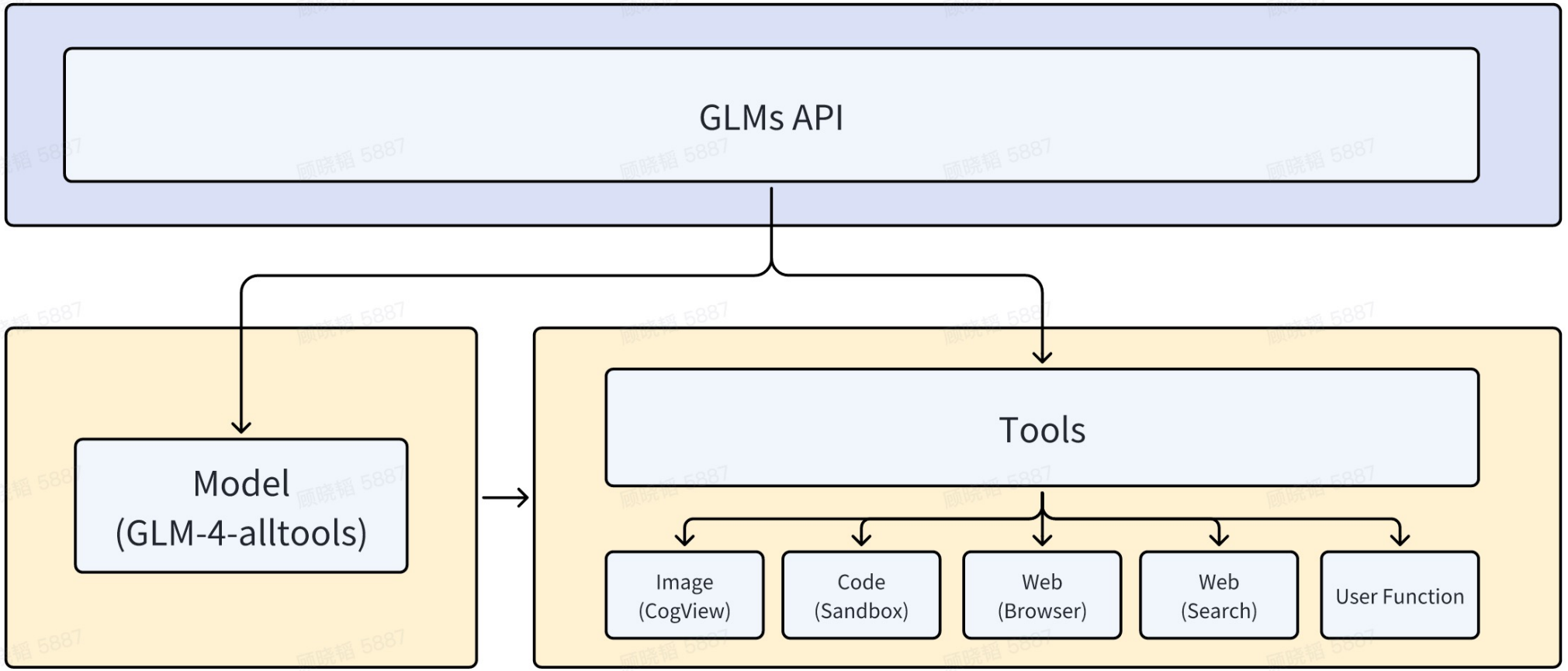
Yang et al., **Inf-DiT**: Upsampling Any-Resolution Image with Memory-Efficient Diffusion Transformer. arXiv:2405.04312

智谱清言：智能体

智谱清言 chatglm.cn | APPs on iOS, Android, Windows, Mac

智谱清言：智能体APIs

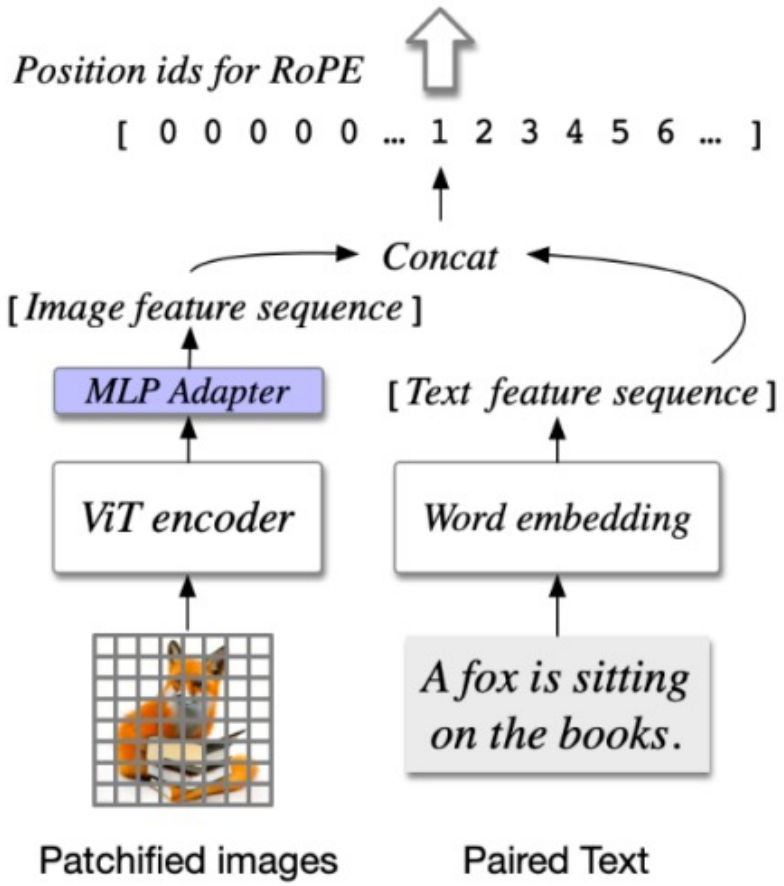
零代码创建智能体及智能体API



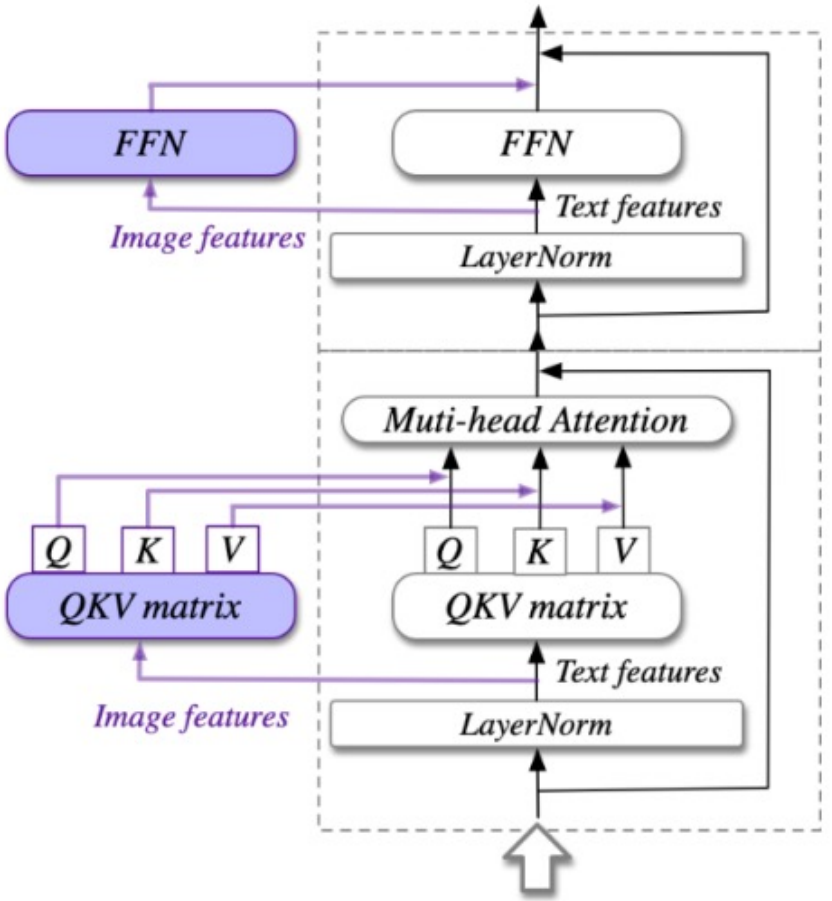
LLM/VLM for Building GUI agents?

CogVLM: Vision Language Models

开源



(a) The input of visual language model



(b) The visual expert built on the language model

Q1: What is funny thing in this image?
翻译：图中有趣的点是什么？

CogVLM: **The funny thing in this image is that a mobile phone is being charged with an unconventional USB interface, which looks like the top of a VGA cable.** This design is quite unusual and might be considered humorous because it combines the functionality of a charging cable with the appearance of an old video game accessory.
翻译：**这幅图片中有趣的是，一部手机正在通过一个非常规的USB接口充电，这个接口看起来像VGA线的顶部。**这种设计相当不寻常，可能会被认为是幽默的，因为它将充电线的功能与旧视频游戏配件的外观相结合。

MiniGPT4: The image shows a smartphone with a USB cable plugged into it. The phone appears to be charging. The image is not funny.
翻译：这幅图片展示了一部智能手机，手机上插有一个USB线。手机似乎正在充电。这幅图片并不好笑。

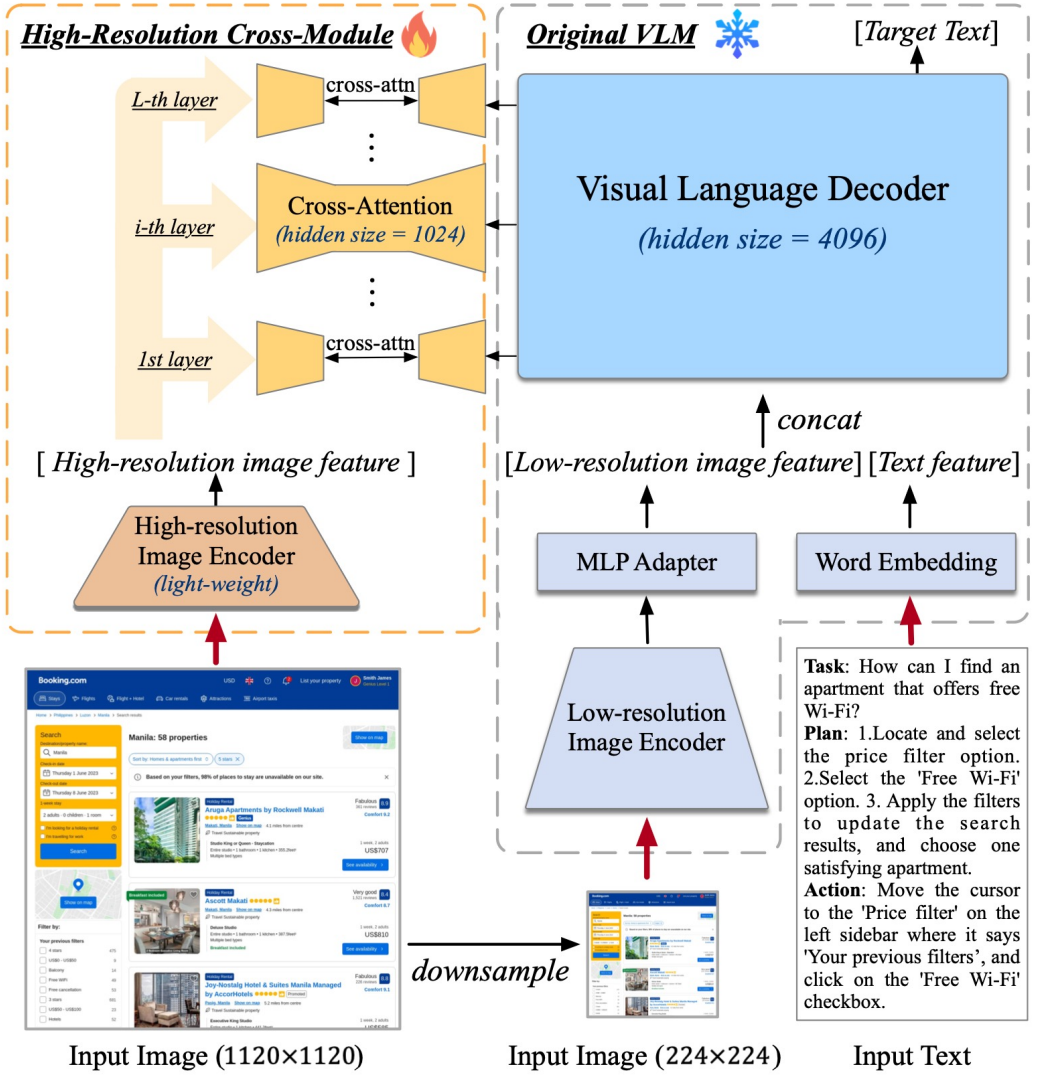
LLava-v1.5: The funny thing in this image is that the cell phone is plugged into a charger, but the charger is upside down. This is unusual because chargers are typically used with the correct orientation, with the plug facing downwards to connect to the device. The upside-down charger adds a humorous element to the scene.
翻译：这幅图片中有趣的是，手机插在充电器上，但充电器是倒放的。这很不寻常，因为充电器通常都是正确方向使用的，插头朝下以连接到设备。这个颠倒的充电器为场景增添了幽默元素。

Wang et al. CogVLM: visual experts for large language models. Submitted **https://github.com/THUDM**

CogAgent: A Visual Language Model for GUI Agents 开源

- High resolution (1120*1120 pixel) by adding a efficient high-resolution cross-module
- Visual agent ability: Web, smartphone app…

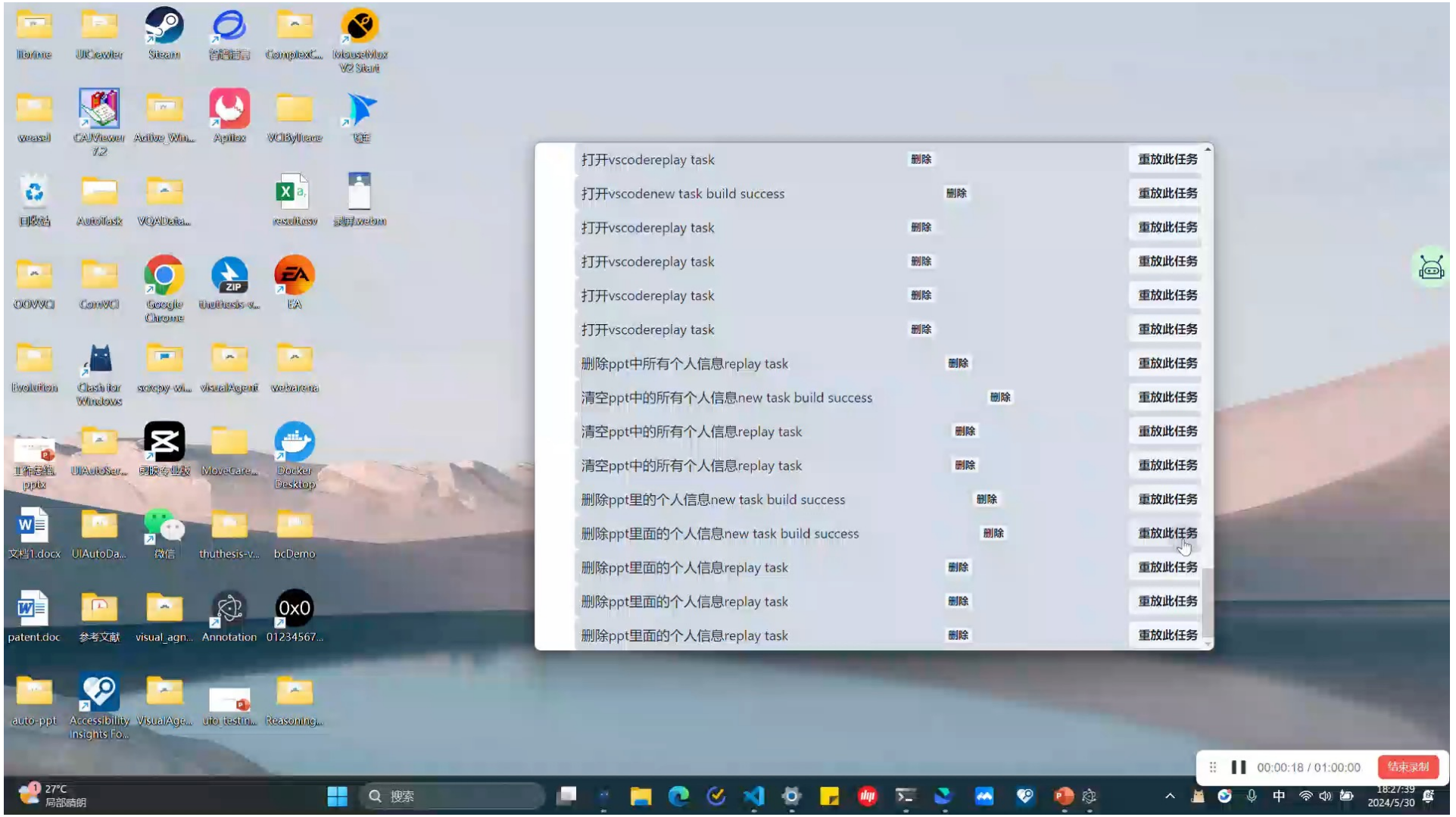
- Achieves SOTA on 5 text-rich and 4 general VQA benchmarks
- Will release soon



Hong et al. CogAgent: A Visual Language Model for GUI Agents. CVPR'24 https://github.com/THUDM/CogVLM

CogAgent

请删除ppt中个人信息



Hong et al. CogAgent: A Visual Language Model for GUI Agents. CVPR'24 https://github.com/THUDM/CogVLM

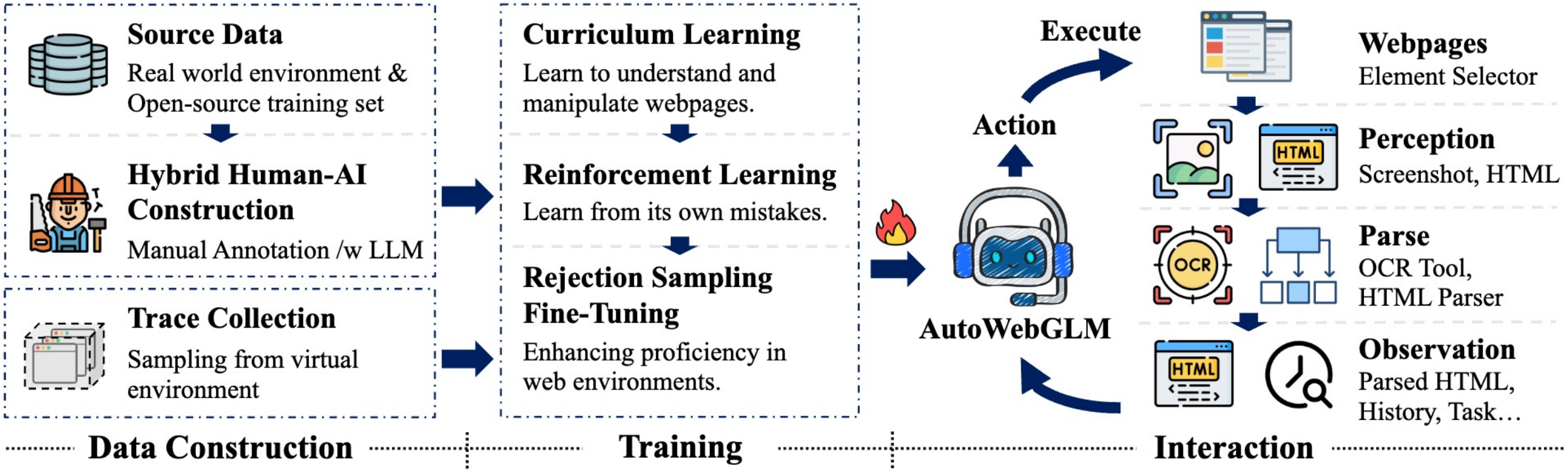
CogAgent

任务：1. CogAgent日程设定；2. 邮件回复；3. github star

Hong et al. CogAgent: A Visual Language Model for GUI Agents. CVPR'24 https://github.com/THUDM/CogVLM

AutoWebGLM

开源

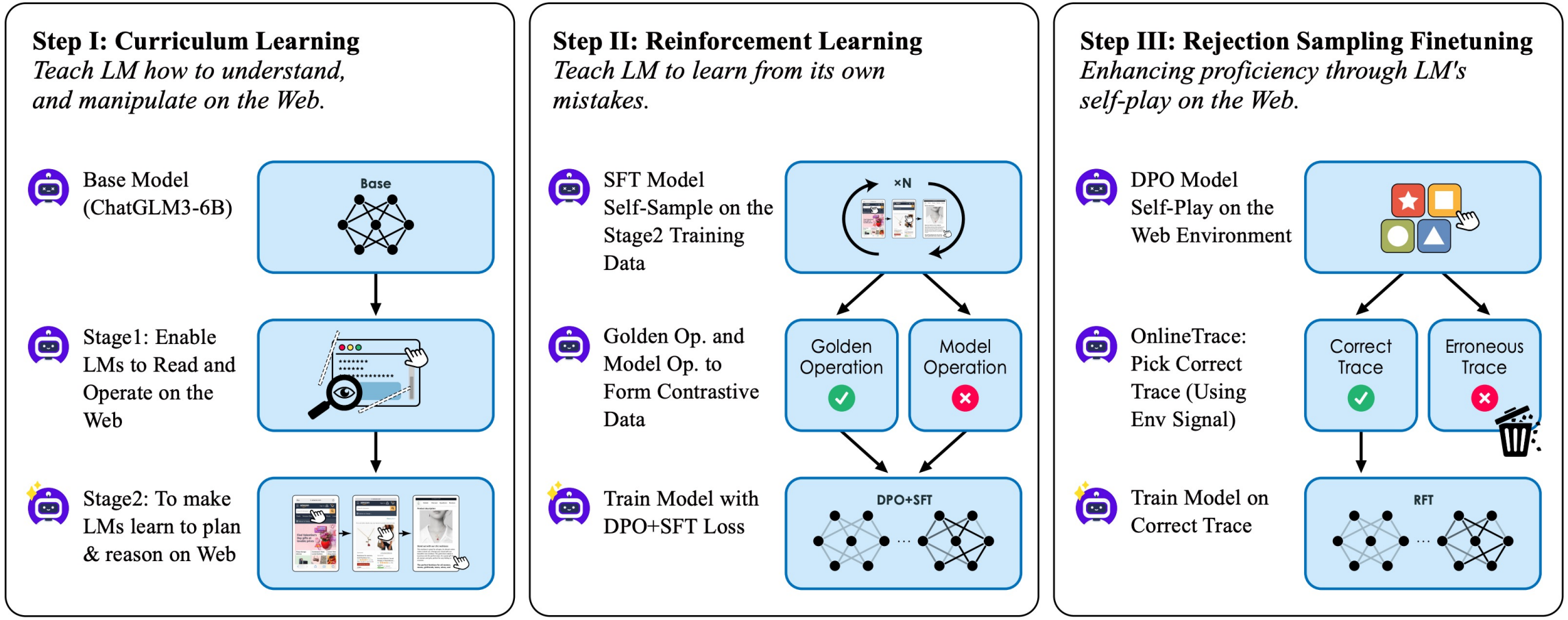


1. Hanyu Lai, Xiao Liu, et al. **AutoWebGLM**: A Large Language Model-based Web Navigating Agent. KDD'24

AutoWebGLM——训练框架

开源

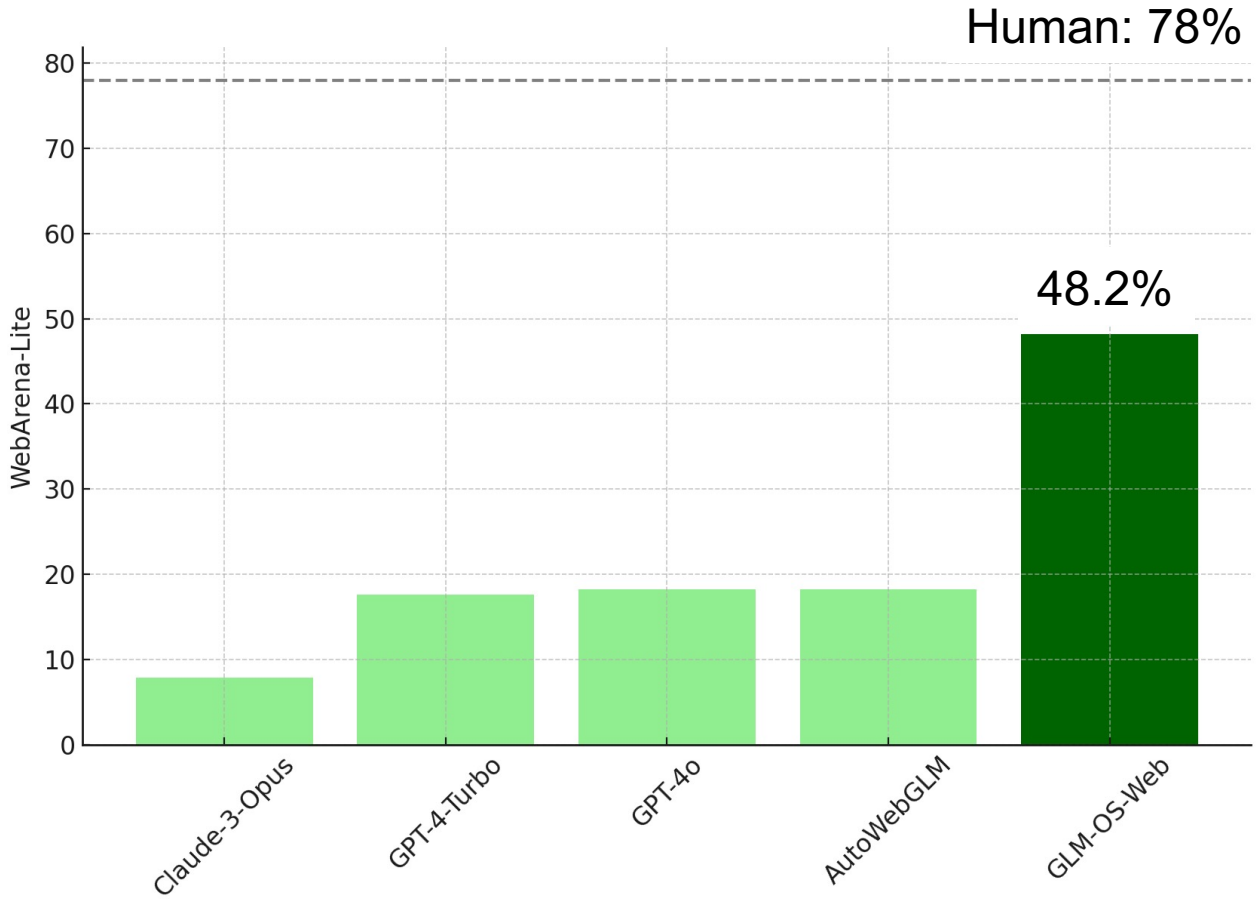
课程学习 -> DPO 偏好学习 -> 拒绝采样微调



1. Hanyu Lai, Xiao Liu, et al. **AutoWebGLM**: A Large Language Model-based Web Navigating Agent. KDD'24

AutoWebGLM (Android)

为我筛选出价格在100元到300元，同时包邮的女性钱包产品



1. Hanyu Lai, Xiao Liu, et al. **AutoWebGLM**: A Large Language Model-based Web Navigating Agent. KDD'24

AutoWebGLM (Android)

打开量子位的最新推送，给此文章写一段英文摘要

故宫和颐和园哪个离我更近

请在我的订单中找到我最新下单的外卖，再来一单

1. Hanyu Lai, Xiao Liu, et al. **AutoWebGLM**: A Large Language Model-based Web Navigating Agent. KDD'24

From GLM-130B to **GLM-4 All Tools**

GLM-130B
MMLU: 44.8%
Aug. 2022

Align with human intent → **ChatGLM** (Mar. 2023)

GLM-2
MMLU: 66.6%

Better Architecture with **32K Context** → **ChatGLM2** (Jun. 2023)

GLM-3
MMLU: 71.0%

Native **Agent** & **Function Call** Capabilities → **ChatGLM3** (Oct. 2023)

GLM-4
MMLU: 81.7%

• LongAlign • AgentTuning → **GLM-4 （All Tools）** (Jan. 2024)

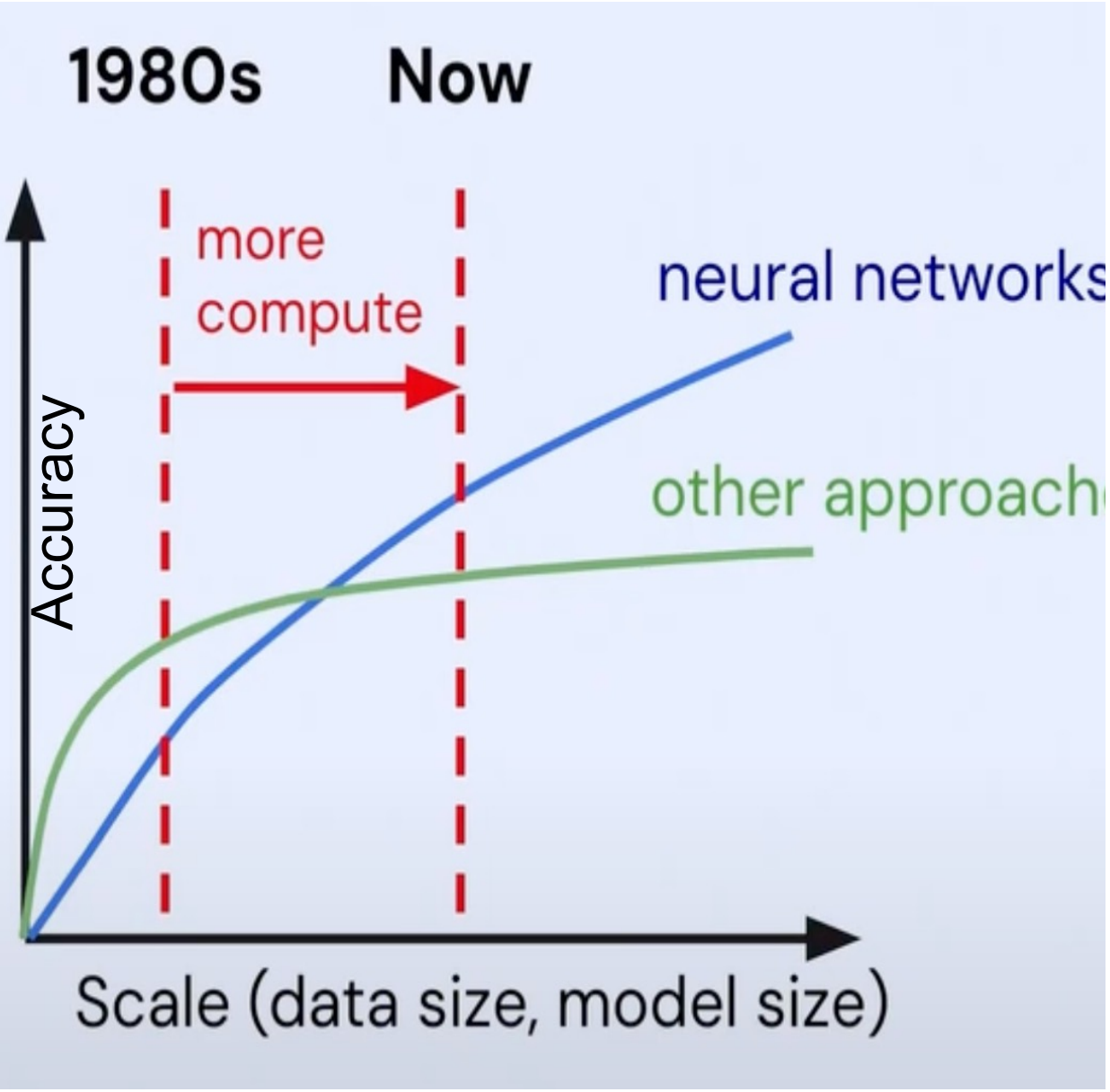
CogVLM | CogAgent | AutoWebGLM

GLM-4 9B （1M | Open） (Jun. 2024)

Benchmarks

- **对齐：** AlignBench (ACL'24)
  - evaluating Chinese alignment for language and reasoning
- **智能体：** AgentBench (ICLR'24)
  - evaluating LLMs' agent capacities
- **长文本：** LongBench (ACL'24)
  - evaluating long context handling
- **工程代码：** NaturalCodeBench (ACL'24 Findings)
  - evaluating real code problem solving
- **多语言代码：** HumanEval-X (KDD'23)
  - evaluating code generation beyond Python
- **安全：** SafetyBench (ACL'24)
  - evaluating LLM ethics and safety

Scaling 的尽头?



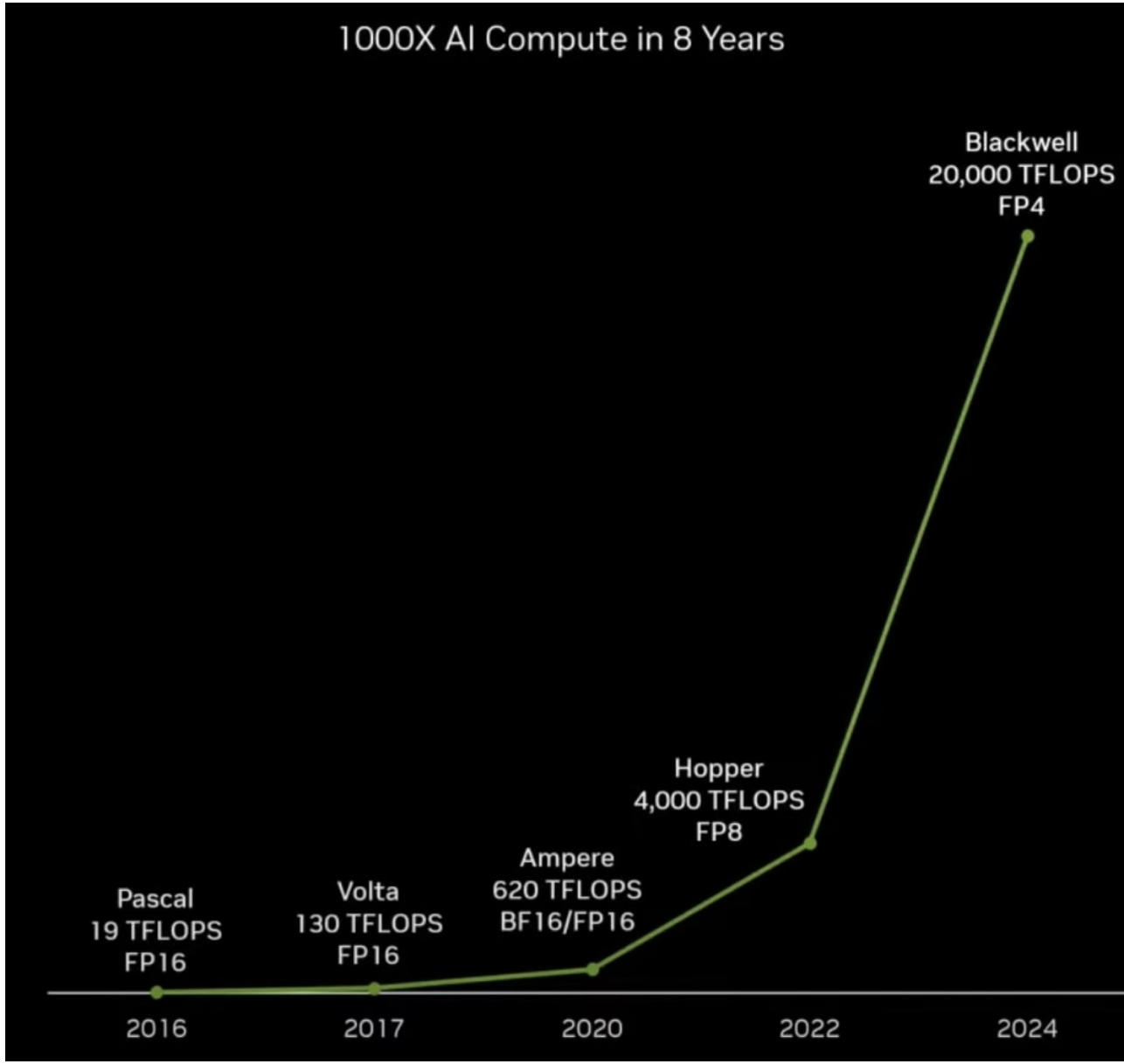


Figure credit: google | nvidia

相关文献

1. Team GLM et al. **ChatGLM**: A Family of Large Language Models from GLM-130B to GLM-4 All Tools. arXiv:2406.12793
2. Hanyu Lai, Xiao Liu, et al. **AutoWebGLM**: A Large Language Model-based Web Navigating Agent. KDD'24
3. Wenyi Hong, et al. **CogAgent**: A Visual Language Model for GUI Agents. CVPR'24.
4. Zeng et al. **AgentTuning**: Enabling Generalized Agent Abilities for LLMs. arXiv: 2310.12823
5. Liu e al. **AgentBench**: Evaluating LLMs as Agents. ICLR 2024. arXiv: 2308.03688
6. Yushi Bai et al. **LongBench**: A Bilingual, Multitask Benchmark for Long Context Understanding. ACL 24
7. Wang et al. **CogVLM**: Visual Expert for Pretrained Language Models. arXiv: 2311.03079
8. Qinkai Zheng, Xiao Xia, et al. **CodeGeeX**: A Pre-Trained Model for Code Generation with Multilingual Benchmarking on HumanEval-X. KDD'23.
9. Xiao Liu, Hanyu Lai, Yu Hao, Yifan Xu, Aohan Zeng, Zhengxiao Du, Peng Zhang, Yuxiao Dong, and Jie Tang. **WebGLM**: Towards An Efficient Web-enhanced Question Answering System with Human Preference. KDD'23.
10. Aohan Zeng, Xiao Liu, Zhengxiao Du, et al. **GLM-130B**: An Open Bilingual Pre-trained Model. ICLR'23.

谢谢大家！

https://github.com/THUDM