The **GLM** LLMs: From GLM-130B to ChatGLM to xTrimoPGLM

Yuxiao Dong
Knowledge Engineering Group (KEG)
Computer Science, Tsinghua University

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

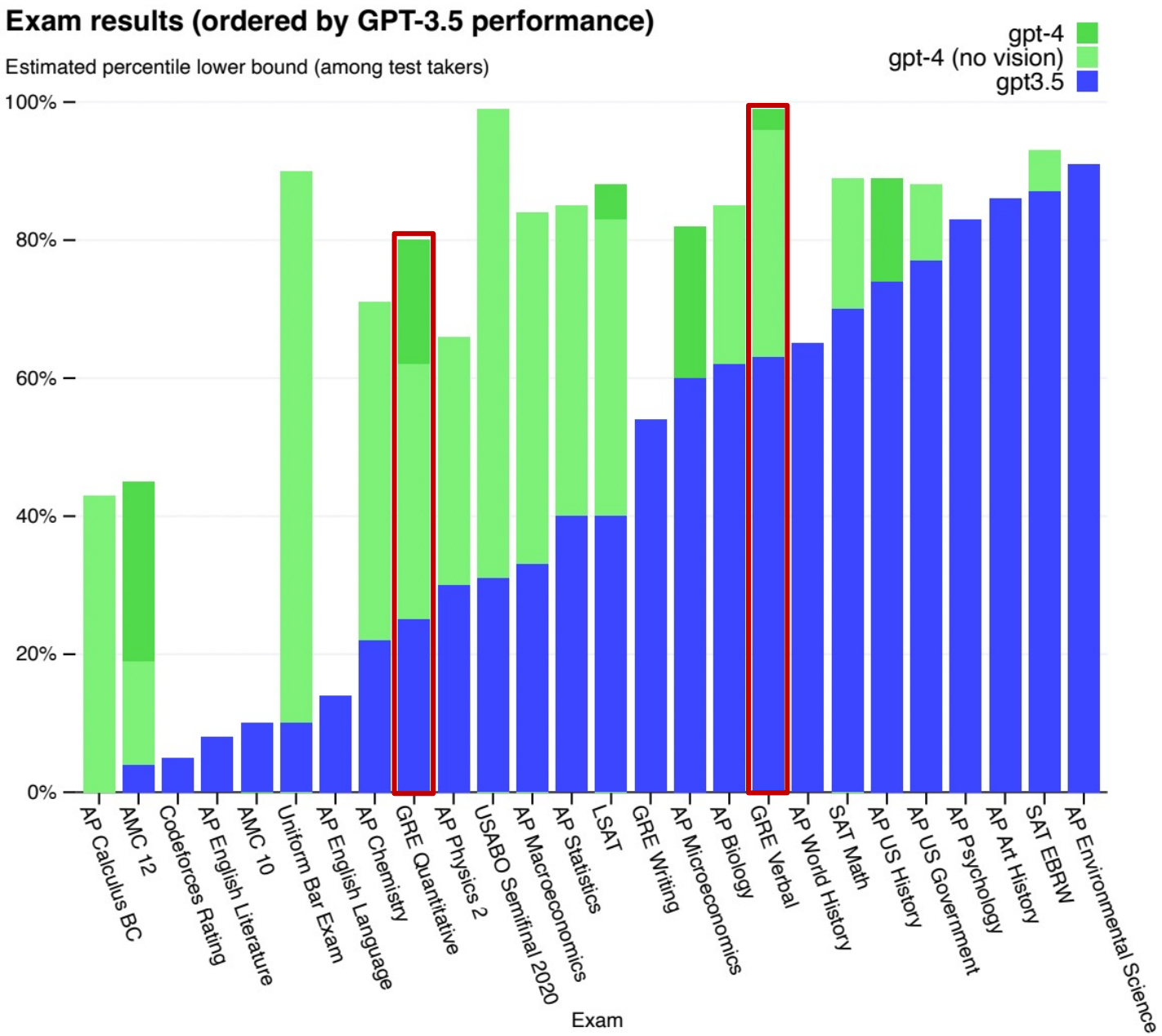
Joint Work with






and many others at **Tsinghua** and **Zhipu.AI**

2023.3.14 GPT-4



OpenAI. GPT-4 Technical Report

2023.3.14 GPT-4

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

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

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.

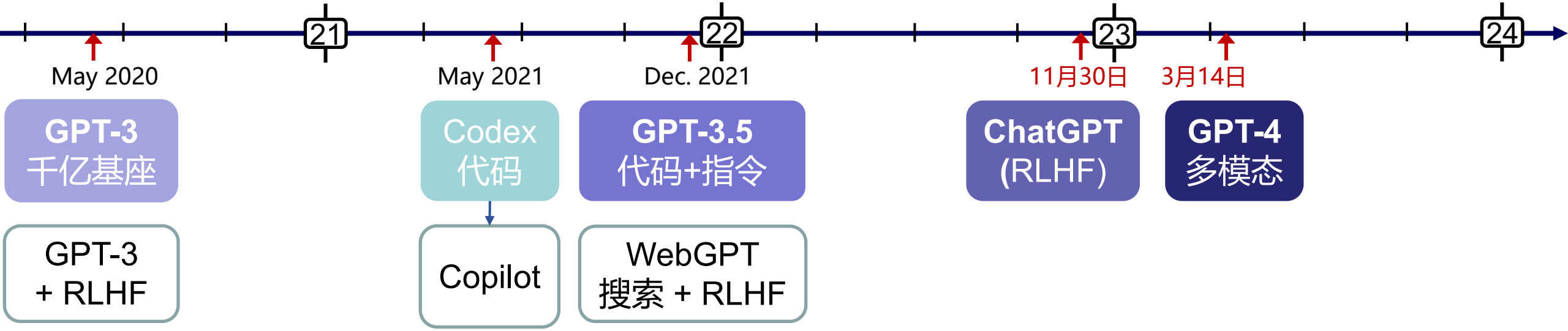
OpenAI. GPT-4 Technical Report

OpenAI GPTs

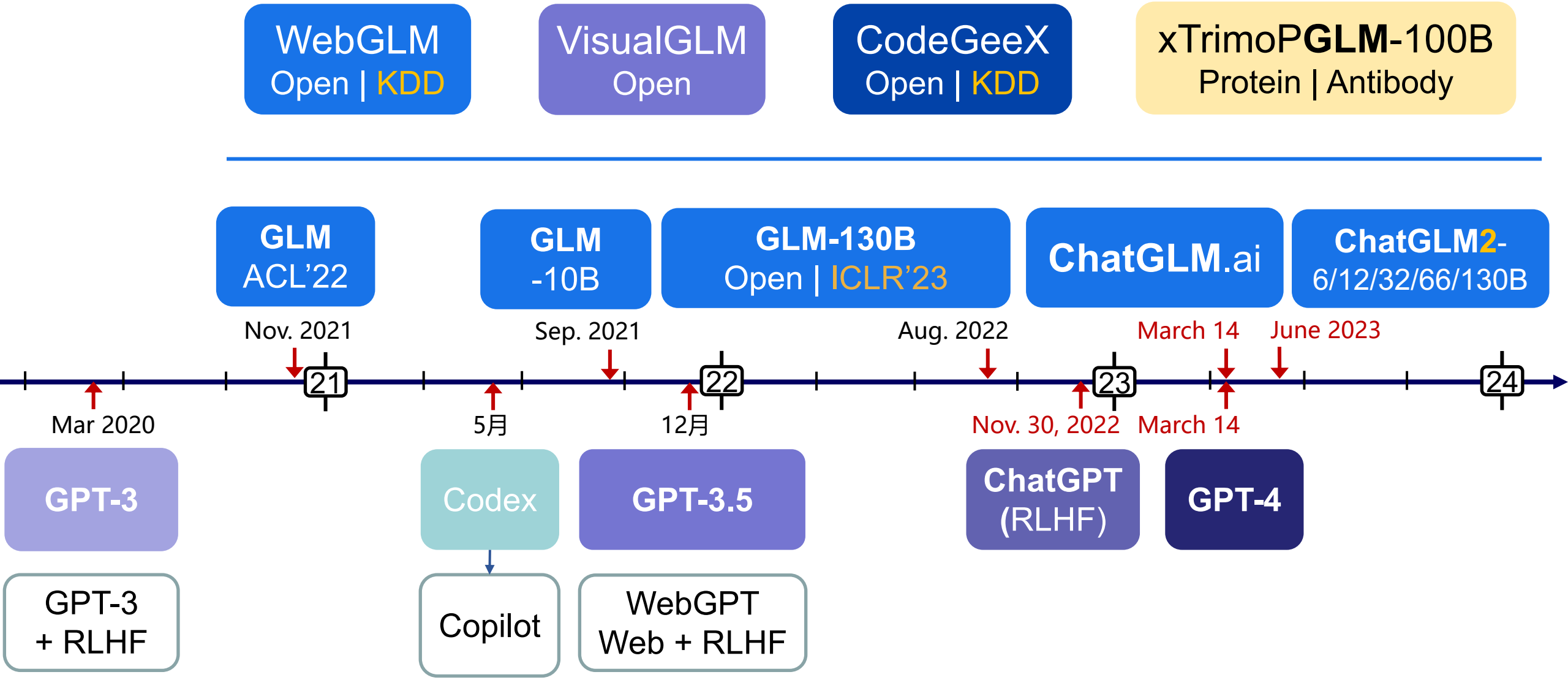
1. 100B-scale base model

3. RLHF

2. SFT / Instructions



Tsinghua & Zhipu.AI GLMs



OpenAI GPTs

100B-scale Pre-Training

OpenAI

- GPT-3 175B

[2020]

Microsoft NVIDIA

- Megatron-Turing-530B

[2021]

Google

- LaMDA 137B
- PaLM 540B

[2021~2022]

DeepMind

- Chinchilla 70B
- Gopher 260B

[2021~2022]

清華大學 Tsinghua University ZHIPU·AI

- GLM-130B

[2022]

Meta

- OPT 175B

[2022]

BigScience

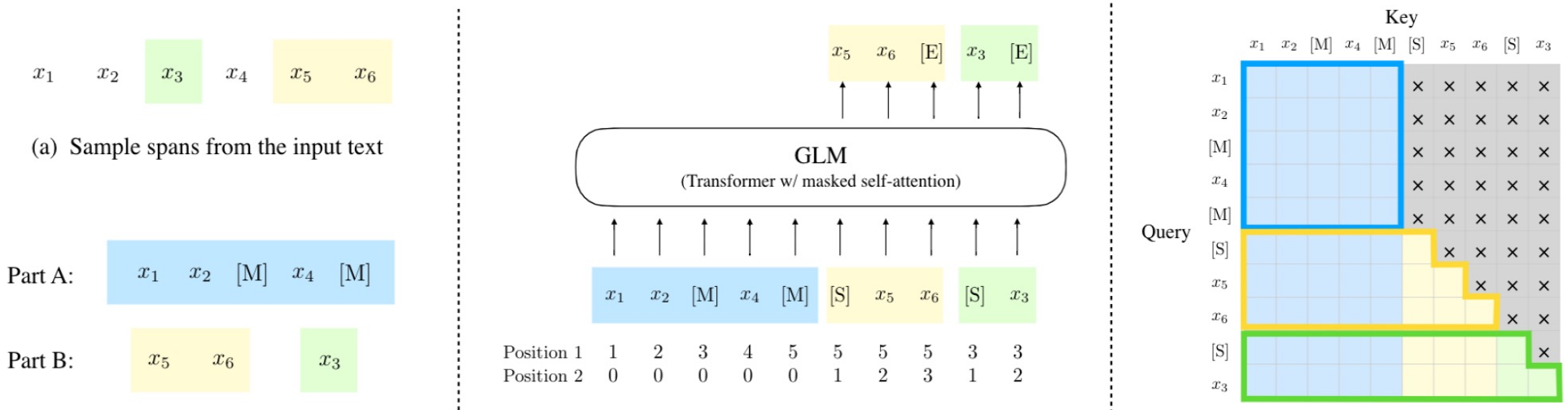
- BLOOM 176B

[2022]

GLM-130B • **NVIDIA** • **DCU** • **Ascend 910** • **Sunway**	200B CN 200B EN (2022.07)	Transformer 130B	**GLM** Autoregressive Blank Infilling
	DATA	NN	Pre-Training Obj
GPT-3 davinci • **NVIDIA**	300B EN (2020.05)	Transformer 175B	**GPT** Autoregressive

General Language Model (GLM)

Framework	NLU	Cond. Gen.	Uncond. Gen.
Autoregressive (GPT)	—	—	√
Autoencoding (BERT)	√	×	×
Encoder-Decoder (T5)	—	√	—
Autoregressive Blank-Infilling (GLM)	√	√	√



Du and Qian et al. All NLP Tasks are Generation Tasks. ACL'22. arxiv: 2103.10360

General Language Model (GLM)

1. Du, et al. “GLM: General Language Model Pre-training with Autoregressive Blanking Infilling.”, ACL, 2022

Text Understanding

- Better than BERT、T5、RoBERTa

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
$\text{BERT}_{\text{Base}}$	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**
$\text{BERT}_{\text{Large}}$	76.3/75.6	69.0	64.4	73.6	80.1	**71.0**	94.8/92.9	71.9/24.1	72.0
$\text{UniLM}_{\text{Large}}$	80.0/79.1	72.0	65.4	76.5	80.5	69.7	91.0/91.1	77.2/38.2	74.1
$\text{GLM}_{\text{Large}}$	81.7/81.1	76.0	**81.7**	74.0	**82.1**	68.5	96.1/94.6	**77.1/36.3**	77.0
$\text{GLM}_{\text{Large}}$ (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
$\text{BART}_{\text{Large}}$ *	88.3/87.8	60.0	65.4	84.5	84.3	69.0	90.5/92.9	81.8/48.0	76.0
$\text{RoBERTa}_{\text{Large}}$ *	89.0/88.4	**90.0**	63.5	87.0	**86.1**	**72.6**	96.1/94.6	**84.4/52.9**	81.5
$\text{GLM}_{\text{RoBERTa}}$	**89.6/89.0**	82.0	83.7	**87.7**	84.7	71.2	**98.7/98.2**	82.4/50.1	**82.9**

1. Du, et al. “GLM: General Language Model Pre-training with Autoregressive Blanking Infilling.”, ACL, 2022

Text Generation

- The most important thing is that **one model can do all the things**

Table 3. Results on Gigaword abstractive summarization

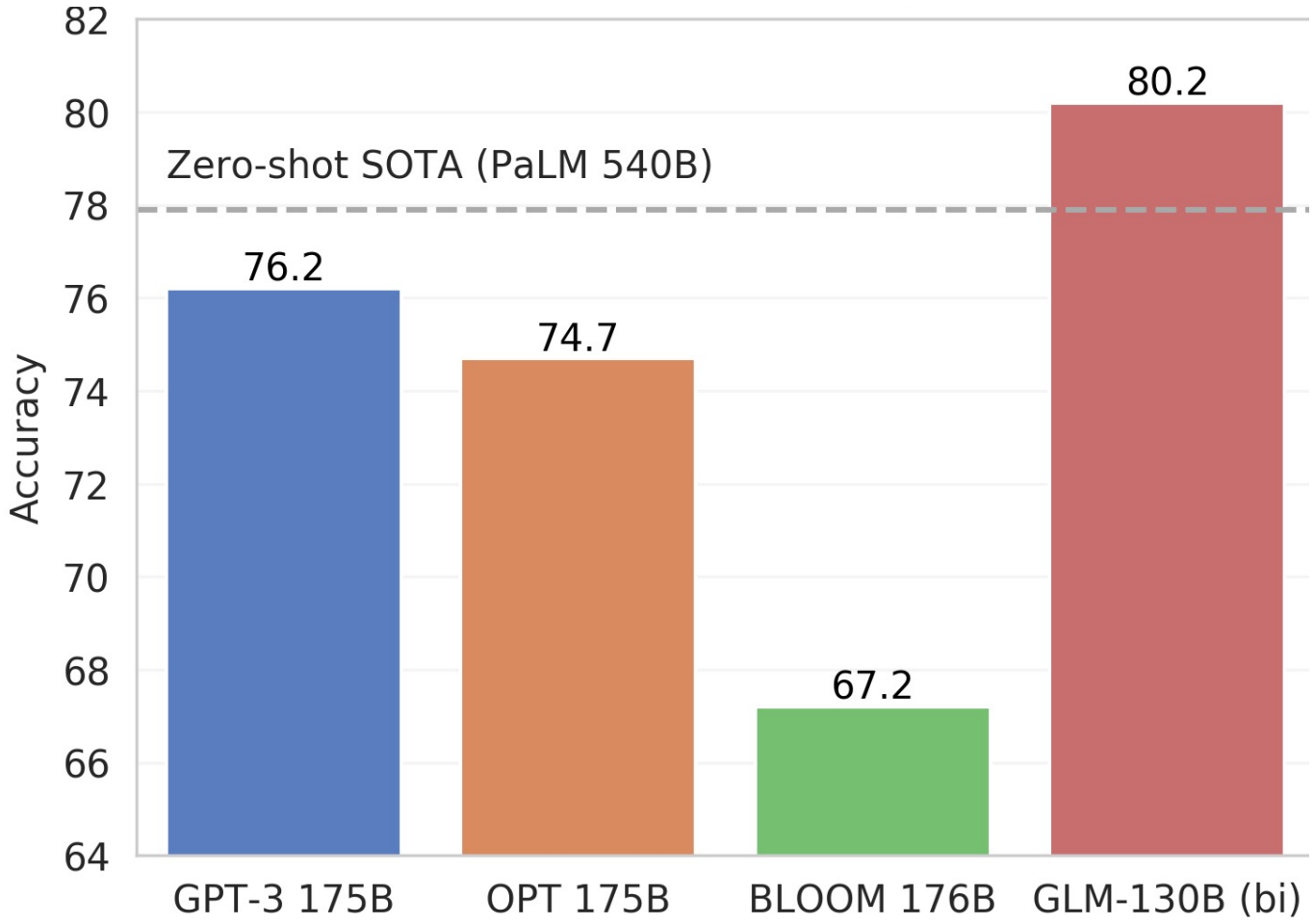
Model	RG-1	RG-2	RG-L
MASS	37.7	18.5	34.9
$\text{UniLM}_{\text{Large}}$	38.5	19.5	35.8
$\text{GLM}_{\text{Large}}$	38.6	19.7	36.0
$\text{GLM}_{\text{Large}}$ (multi-task)	38.5	19.4	35.8
GLM_{410M} (multi-task)	**38.9**	**20.0**	**36.2**

Table 4. Zero-shot language modeling results.

Model	Lambada (Accuracy)	BookWiki (Perplexity)
$\text{GLM}_{\text{Large}}$ (uni)	0.0	> 100
$\text{GLM}_{\text{Large}}$ (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**
$\text{GLM}_{\text{Large}}$ (bi)	10.6	> 100
$\text{GLM}_{\text{Large}}$ (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**
$\text{GPT}_{\text{Large}}$ (uni)	50.1	14.4

1. Du, et al. “GLM: General Language Model Pre-training with Autoregressive Blanking Infilling.”, ACL, 2022

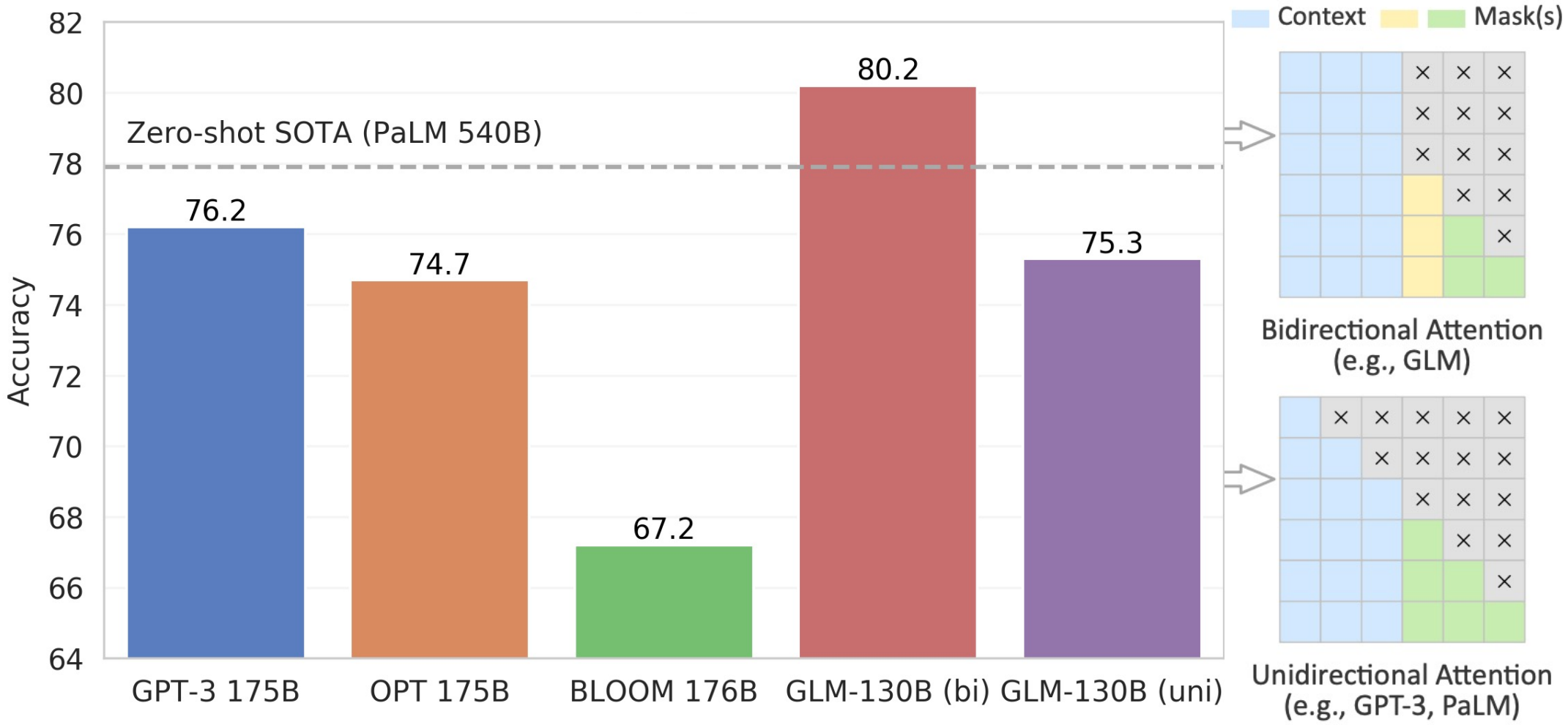
GLM?



LAMBADA

Zeng, Liu, et al. GLM-130B: An Open Bilingual Pre-trained Model. ICLR'23

GLM?



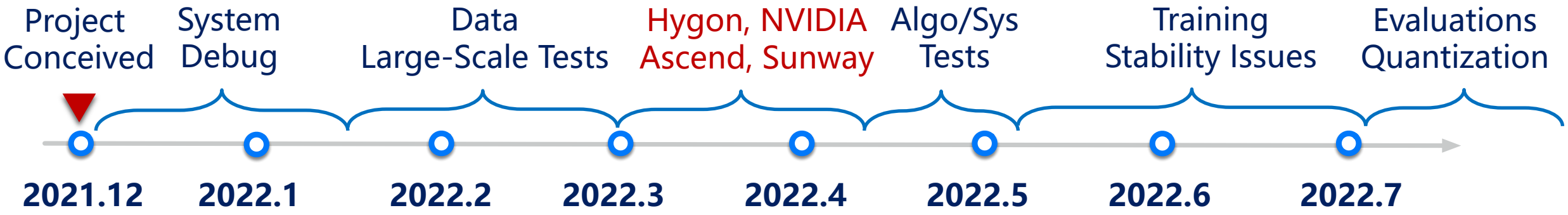
LAMBADA

Zeng, Liu, et al. GLM-130B: An Open Bilingual Pre-trained Model. ICLR'23

GLM-130B Training

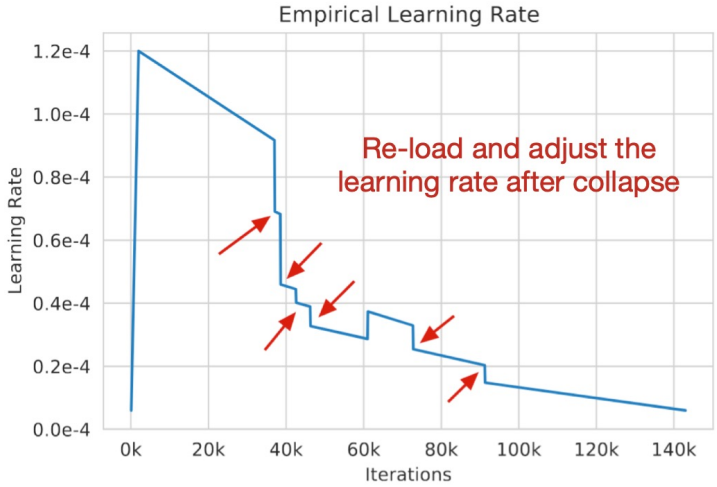
9 months have witnessed numerous technical challenges

- **Engineering:** How to train 100B-scale models from scratch?
  - Hygon DCU, NVIDIA A100, Ascend 910, Sunway
  - Frequent & random hardware failures, Megatron-DeepSpeed 3D pipeline, CUDA kernel efficiency, GPU memory overflow, 10K+ threads TCP init & comms…
- **Algorithm:** How to stabilize the training of 100B-scale models?
  - The gradient norms of embeddings, Post-LN / Pre-LN stability, dataloader state seeds, computation precision in Softmax / Attention



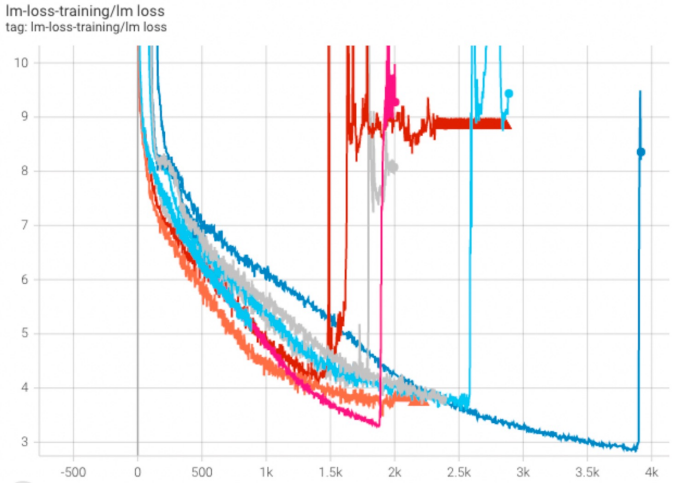
Training Stability of 100B-Scale Models

- Tradeoff: Stability (Slow) or Efficiency (Instable)
- Existing Solutions
  - **OPT-175B:** manually adjust LR & skip data when collapses (performance drop)
  - **BLOOM 176B**: embedding norm & BF16 (performance drop, few platform)



(a) OPT 175B's experiments

(b) BLOOM 176B's experiments



(c) GLM 130B's experiments

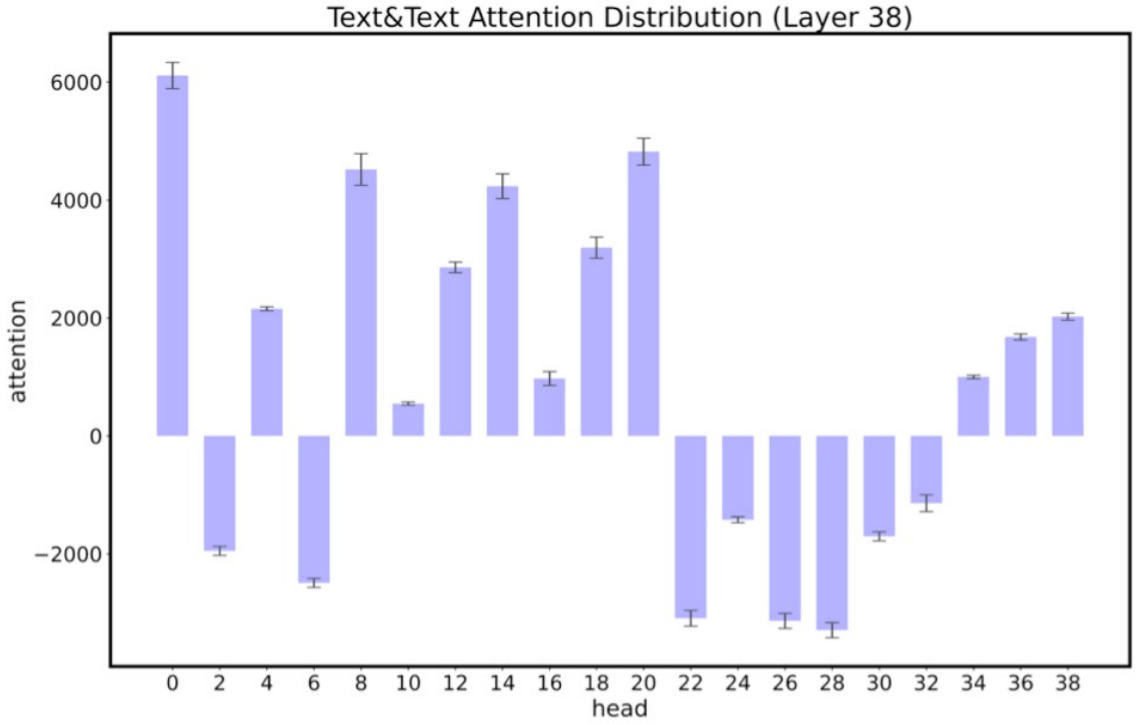
Sources: OPT / BLOOM / GLM-130B

GLM-130B: Training Stability

- Attention score: Softmax in 32 to avoid overflow

$$\mathrm{softmax}\left(\frac{\boldsymbol{Q}_i\boldsymbol{K}_i^\top}{\sqrt{d}}\right) = \mathrm{softmax}\left(\left(\frac{\boldsymbol{Q}_i\boldsymbol{K}_i^\top}{\alpha\sqrt{d}} - \max\left(\frac{\boldsymbol{Q}_i\boldsymbol{K}_i^\top}{\alpha\sqrt{d}}\right)\right)\times\alpha\right) = \mathrm{FP16}\left(\mathrm{softmax}\left(\mathrm{FP32}\left(\frac{\boldsymbol{Q}_i\boldsymbol{K}_i^\top}{\alpha\sqrt{d}}\right)\times\alpha\right)\right)$$

Attention scores grow large --- exceeding FP16's range



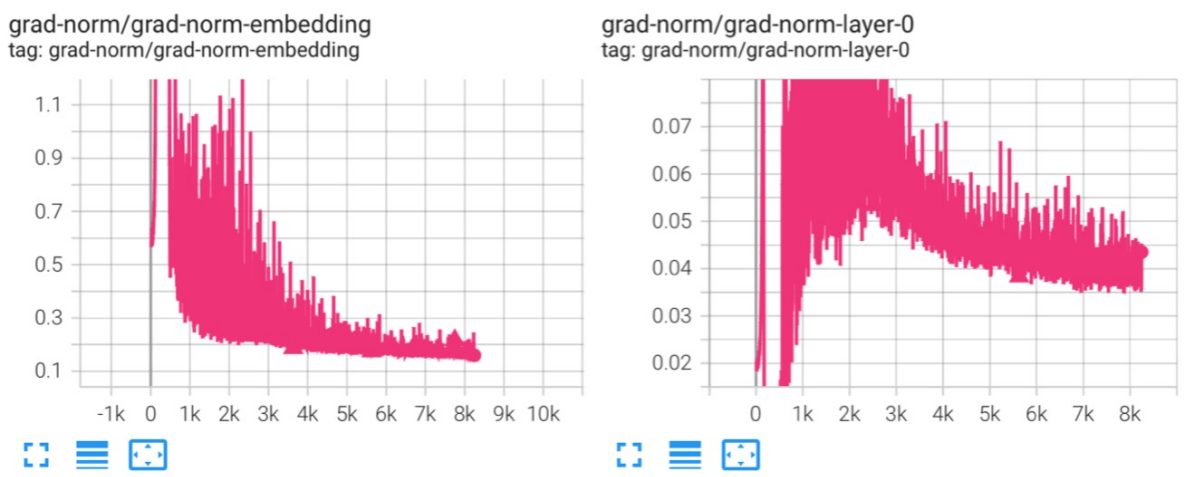
Zeng, Liu, et al. GLM-130B: An Open Bilingual Pre-trained Model. ICLR'23

GLM-130B: Training Stability

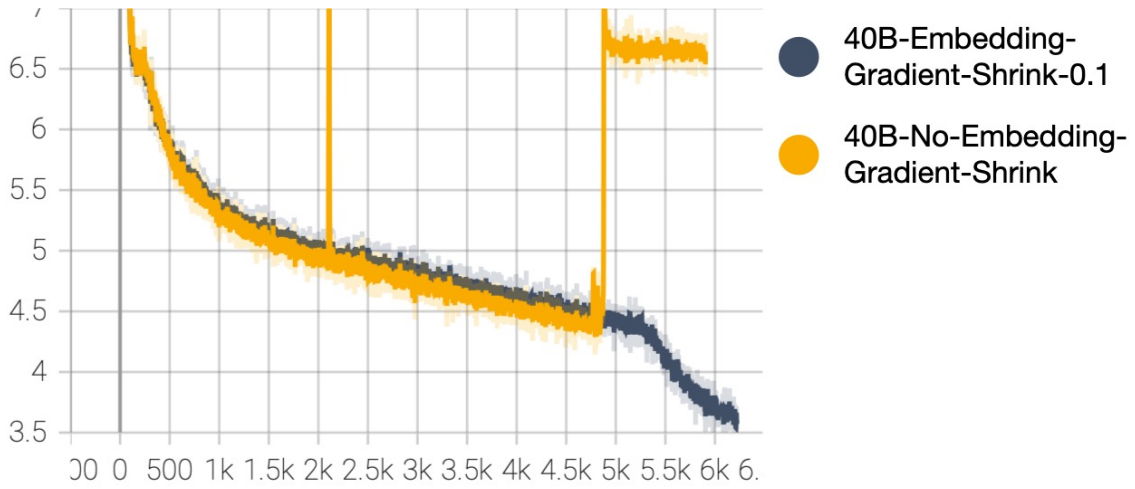
◻ Embedding Layer Gradient Shrink (EGS)

```
word_embedding = word_embedding * alpha +
                 word_embedding .detach() * (1 - alpha)
```

Embedding Layer gradients can be magnitudes larger than others



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

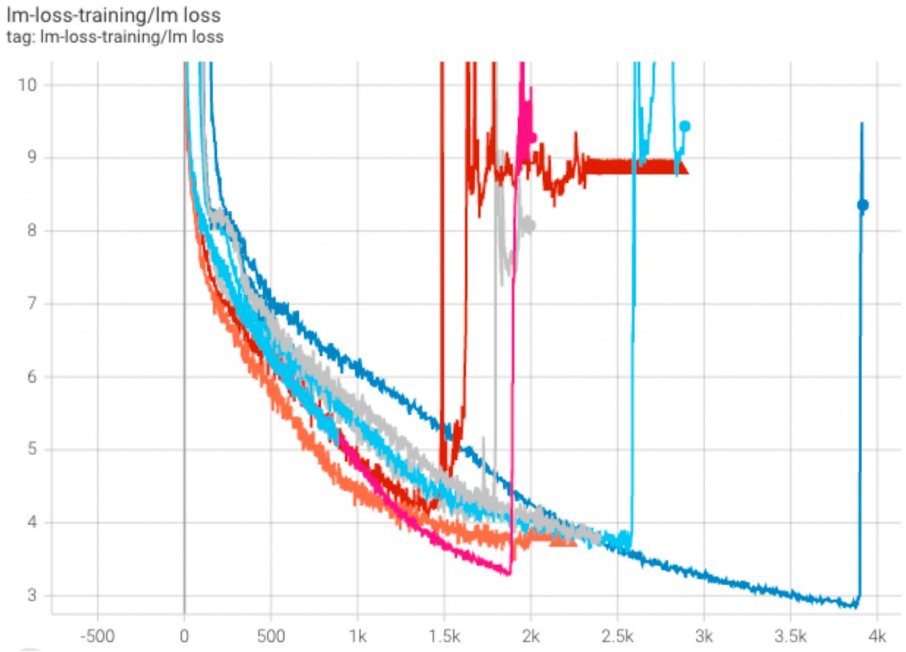


(b) Training loss curves of GLM-40B with and without gradient shrink

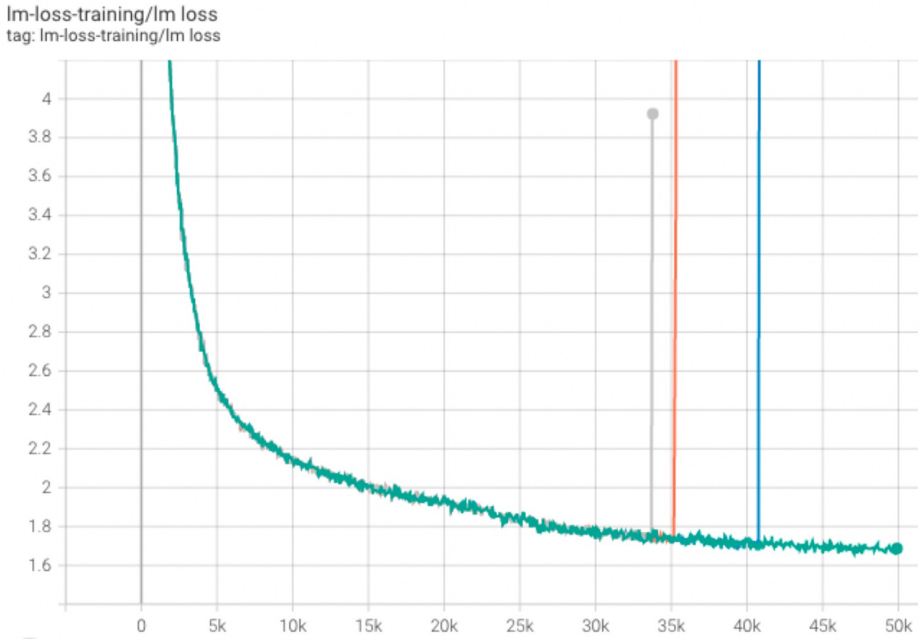
Zeng, Liu, et al. GLM-130B: An Open Bilingual Pre-trained Model. ICLR'23

GLM-130B: Training Stability

- The final training run of GLM-130B



(c) GLM 130B's experiments



(d) GLM 130B's real training

Zeng, Liu, et al. GLM-130B: An Open Bilingual Pre-trained Model. ICLR'23

GLM-130B Training Lessons

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.*

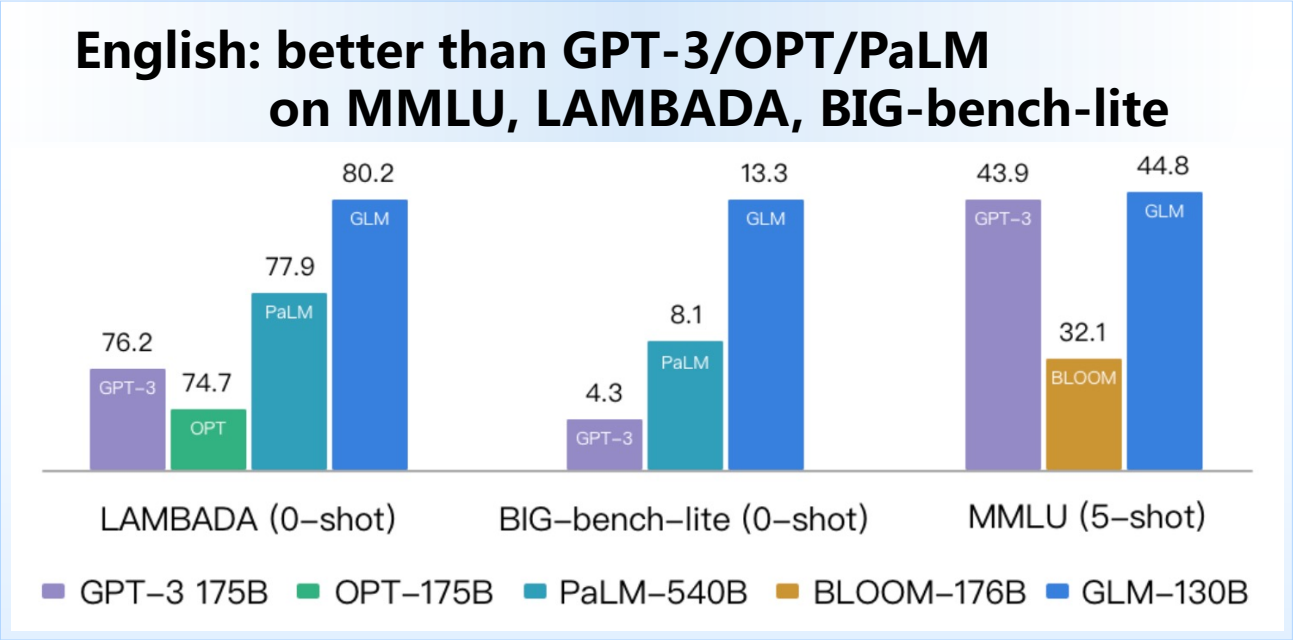
2022.4
- Optimize A100 kernel's computing efficiency *=> A100 kernels prefer square-shaped inputs, and seq_len=2,048 is optimal for our hidden-state dimension (12,288)*
- Inability to converge due to large gradient norms (170+) of input embeddings *=> Try embedding norm and gradient shrink, which turn 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*

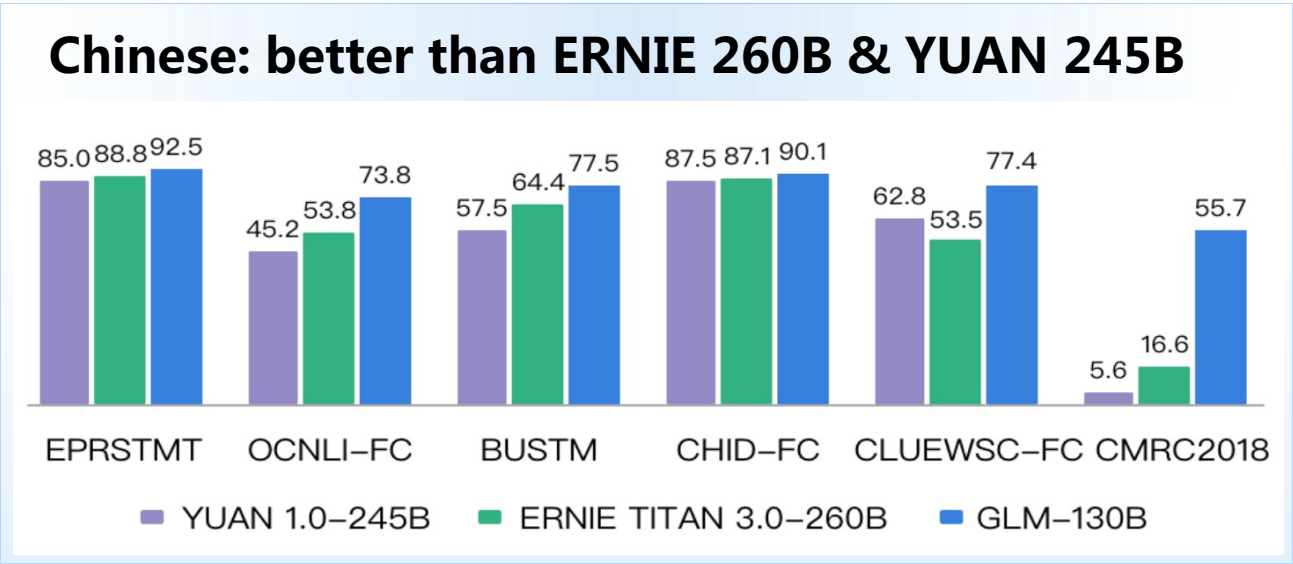
2022.5-6
- Implement a RoPE cuda operator in C++ *=> See unexpected precision errors and finally have it abandoned*
- Sandwich-LN still disconverges *=> 1) Reducing learning rate does not help; 2) Using Hinge cross-entropy becomes slower and harms performance; 3) Shifting to DeepNorm still disconverges*
- 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 fo fine-tuning with reasonable performance*

- Zeng, Liu, et al. GLM-130B: An Open Bilingual Pre-trained Model. ICLR'23
- https://github.com/THUDM/GLM-130B

GLM-130B





Aug., 2022-Mar. 2023, research use requests from ~1000 orgs in 70 countries

- Google
- Microsoft
- Facebook
- Stanford
- MIT
- UC Berkely
- CMU
- Harvard
- Princeton
- Yale
- Cornell
- UIUC
- Cambridge
- Oxford

- Huawei
- Alibaba
- Tencent
- Baidu
- Meituan
- Bytedance
- Didi
- Xiaoice
- Xiaodu
- Xiaomi
- Xiaopeng
- Youdao
- Face++
- Ping An Cap

- Peking U.
- Zhejiang U.
- Shanghai JT U.
- Fudan U.
- USTC
- U of CAS
- Wuhan U.
- Naikai U.
- Hongkong U.
- CUHK
- HKUST
- BAAI
- Zhejiang Lab
- Shanghai AI Lab

GLM-130B in HELM

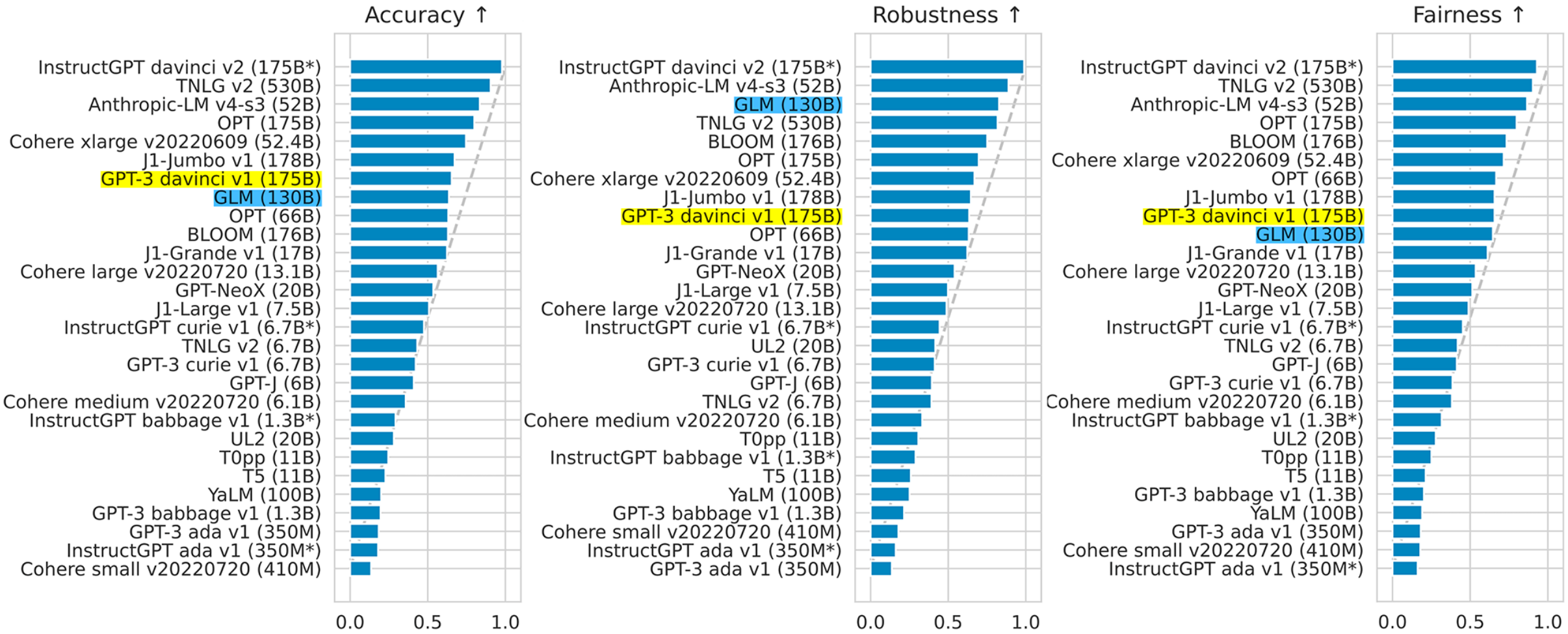
Stanford's Holistic Evaluation of Language Models (HELM, Nov. 2022)

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	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
Anthropic-LM v4-s3 (52B)	Anthropic	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
BLOOM (176B)	BigScience	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
T0++ (11B)	BigScience	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
Cohere xlarge v20220609 (52.4B)	Cohere	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
Cohere large v20220720 (13.1B)[58]	Cohere	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
Cohere medium v20220720 (6.1B)	Cohere	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
Cohere small v20220720 (410M)[59]	Cohere	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
GPT-J (6B)	EleutherAI	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
GPT-NeoX (20B)	EleutherAI	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
T5 (11B)	Google	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
UL2 (20B)	Google	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
OPT (66B)	Meta	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
OPT (175B)	Meta	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
TNLG v2 (6.7B)	Microsoft/NVIDIA	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
TNLG v2 (530B)	Microsoft/NVIDIA	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
GPT-3 davinci v1 (175B)	OpenAI	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
GPT-3 curie v1 (6.7B)	OpenAI	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
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

https://crfm.stanford.edu/helm, **2023.0308**

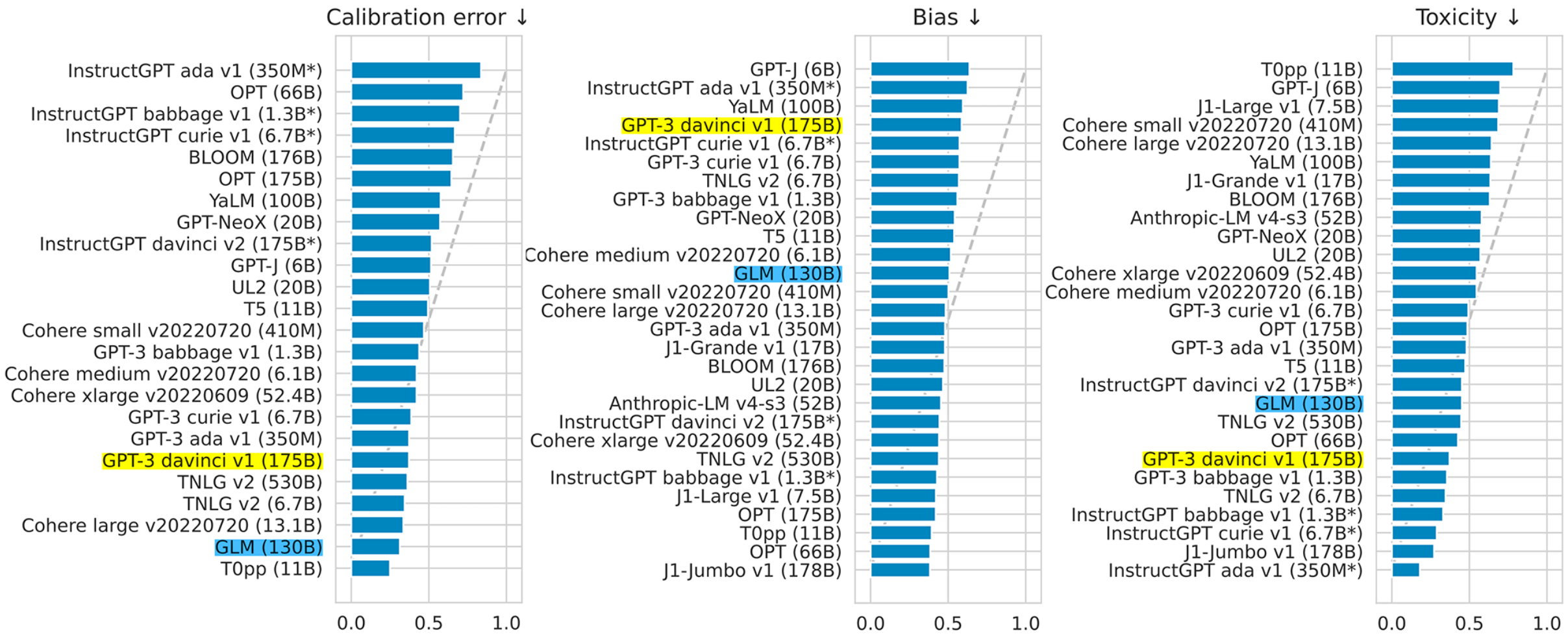
Liang et al., Holistic Evaluation of Language Models. arXiv: **2211.09110**

GLM-130B in HELM



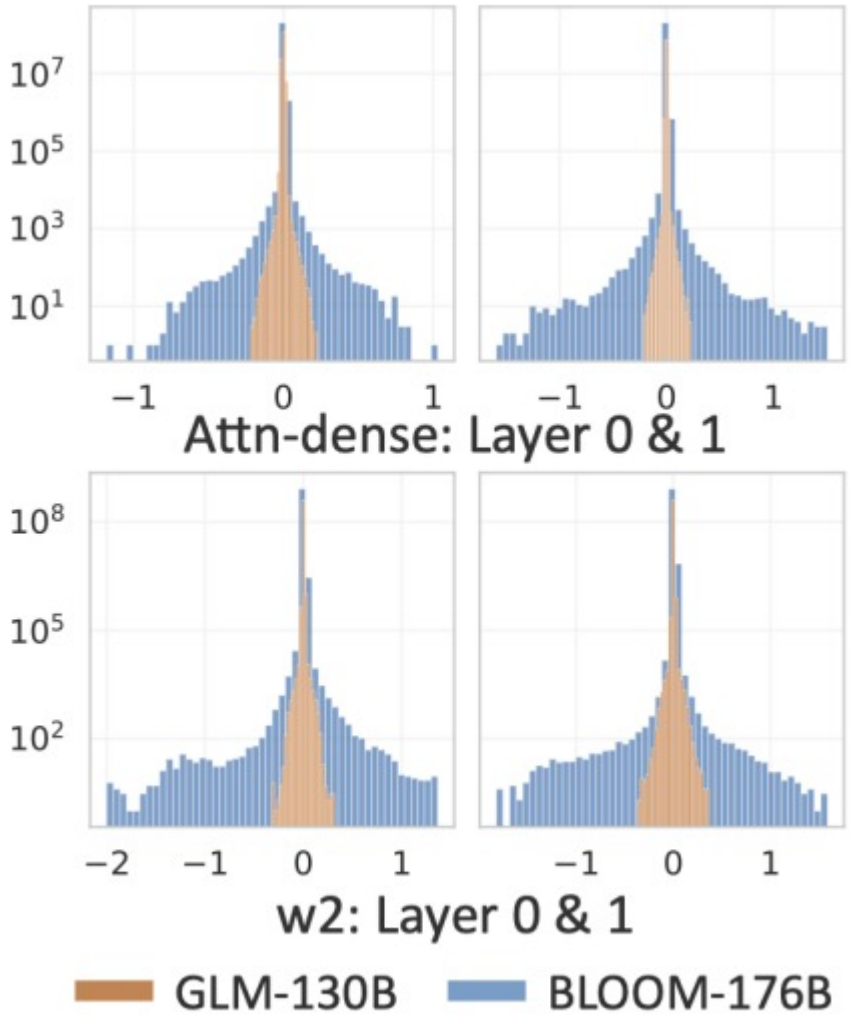
1.Liang et al., Holistic Evaluation of Language Models. arXiv: **2211.09110**

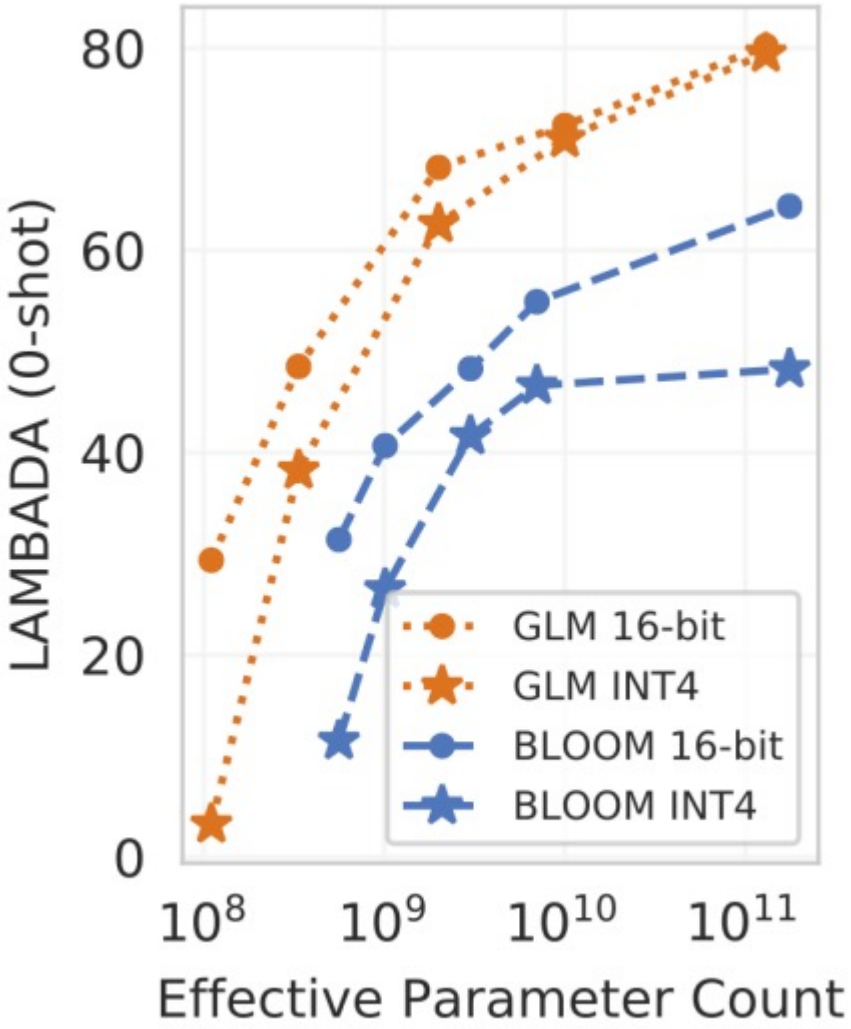
GLM-130B in HELM



1.Liang et al., Holistic Evaluation of Language Models. arXiv: **2211.09110**

INT4 Quantization for RTX 3090s/2080s





GLM's INT4 Weight Quantization Scaling Law

INT4 Quantization for RTX 3090s/2080s

- GLM-130B INT4 Quant. w/o perform. degradation

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

Zeng, Liu, et al. GLM-130B: An Open Bilingual Pre-trained Model. ICLR'23

GLM-130B

	Backbone	Training Objective	Quantization	Acceleration	Cross-Platform
GPT3-175B	GPT	SSL Only	—	—	NVIDIA
OPT-175B	GPT	SSL Only	INT8	Megatron	NVIDIA
BLOOM-176B	GPT	SSL Only	INT8	Megatron	NVIDIA
GLM-130B	**GLM**	SSL & **MIP**	• INT8 • **INT4**	**Faster Transformer**	• NVIDIA • **Hygon DCU** • **Ascend 910** • **Sunway**
Effects	**Performance Impro:** • Big-bench-lite: +5.2% • LAMBADA: +2.3% • CLUE: +24.3% • FewCLUE: +12.8%		**Affordable Serving:** It saves 75% GPU mem in inference; It can infer on **3090x4** / 2080x8	**Fast Inference:** 7-8.4 faster than Pytorch; 2.5 faster than Megatron	**Diverse Supports:** It supports more diverse adoptions of LLMs

Tsinghua & Zhipu.AI GLMs

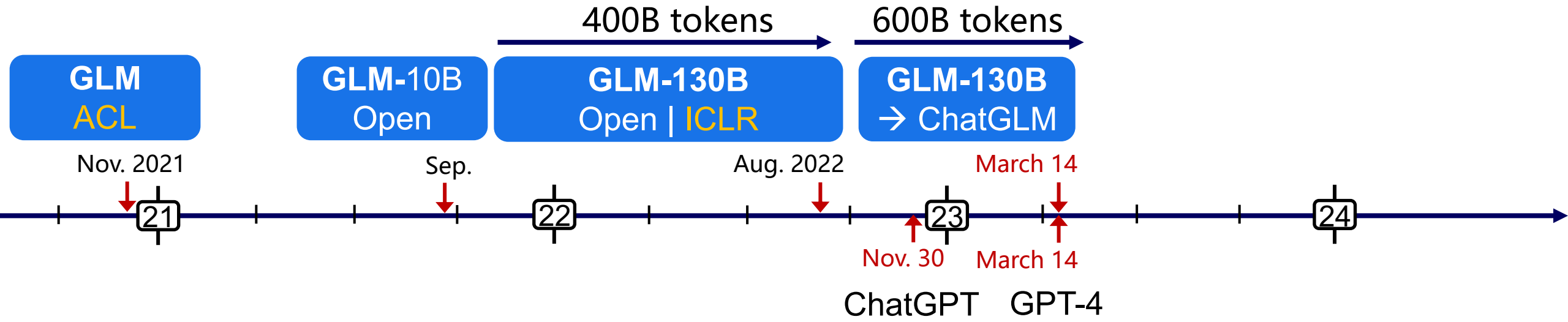
thomas i. liao @ ICML
@ThomasILiao

Models announced today:
- @OpenAI GPT4
- @AnthropicAI Claude
- @thukeg ChatGLM
- @GoogleAI Med-PaLM 2

Patrick Loeber
@patloeber

What a week for AI 🚀

- @OpenAI GPT-4
- @AnthropicAI Claude
- @thukeg ChatGLM
- @GoogleAI PaLM API & MakerSuite
- @AssemblyAI Conformer-1
- Stanford Alpaca



chatglm.cn

ChatGLM 内测版

我们一直在路上

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 >

screenshot at 2023.05.28

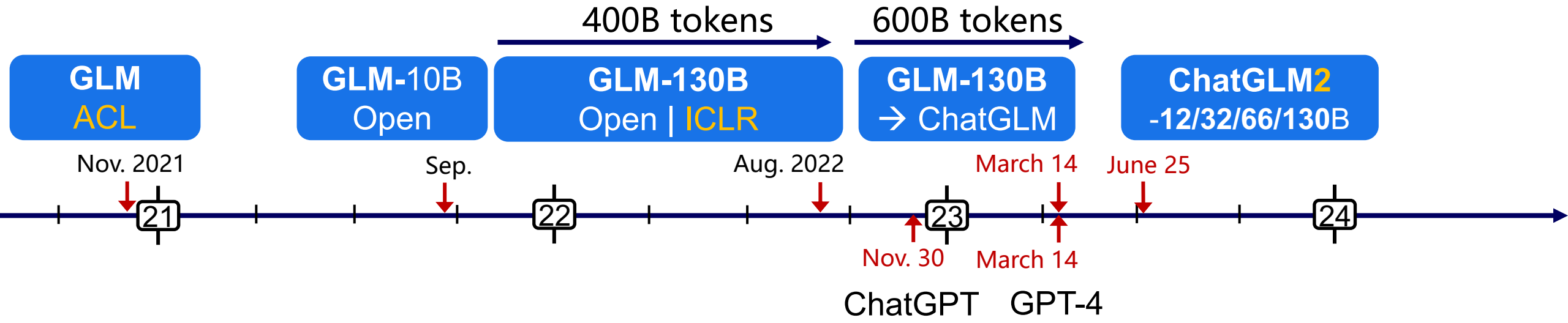
Z-Bench by ZhenFund

(The leaderboard is **later** than the models evaluated)

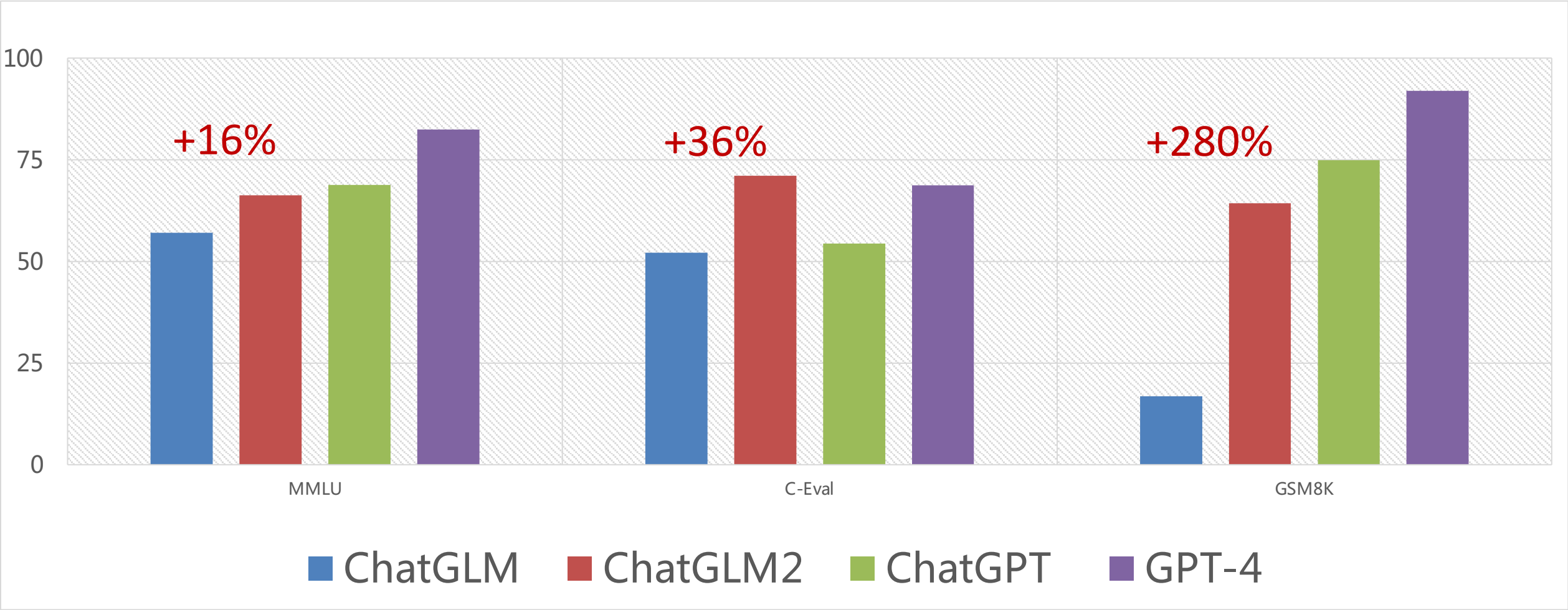
	GPT3.5	GPT4	国内大厂	ChatGLM
Basics	59/63	60/63	26/63	40/63
Advanced	120/209	159/209	41/209	74(+5)/209
Vertical	20/39	26/39	11/39	11(+5)/39
Sum:	199/311	245/311	78/311	125(+10)/311
100%	64.0	78.8	25.1	40.2(+3)

2023.03.20

Tsinghua & Zhipu.AI GLMs



ChatGLM-130B vs. ChatGLM2



chatglm.ai

ChatGLM 内测版

我们一直在路上

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 >

screenshot at 2023.05.28

chatglm.ai

ChatGLM 内测

创建新对话

分析品牌营销领域头部的创新者，并列...
2023-07-05 20:38:58

帮忙写个给猫咪的道歉信，原因是我忘...
2023-07-05 20:37:58

一个包子蒸熟要10分钟，如果我有4个蒸...
2023-06-29 16:44:16

如何建设全国统一大市场
2023-06-29 16:22:59

孙悟空有没有火烧赤壁
2023-06-29 15:55:32

一个包子蒸熟要10分钟，如果我有4个蒸...
2023-06-29 15:55:10

帮我写一个日文公文，内容为今日金价...
2023-06-29 15:53:06

一个包子蒸熟要10分钟，如果我有4个蒸...
2023-06-29 15:51:51

孙悟空有没有火烧赤壁
2023-06-29 15:30:05

您好，我是ChatGLM

我是一个中英双语大模型，通过万亿字符的文本与代码预训练，结合有监督微调，我具备丰富的能力，包括通用问答、多轮对话、角色扮演、文本生成、代码生成等功能。我能为各种场景下的需求提供多维度、多形式的解决方案，期待与您共同探索更多有趣的玩法。

试试以下例子：

工作总结

帮我写一份年终工作总结，夸夸我的优点并展望明年。

道歉信

帮忙写个给猫咪的道歉信，原因是我忘了给它铲屎。

正则

请提供一段JS正则代码，确认是否是真实手机号

提出你的想法

灵感大全

全部 创作 学习 职场 编程 cosplay 教育 公文 论文

广告文案
请帮我为宝宝乐安抚奶嘴生成3个条广告文案，每条字数50字以内

即兴发言
以保护环境为主题，帮我生成一份在科技发展论坛上的即兴发言

开题报告
依据我的研究方向：大模型与传统NLP，帮我输出一份开题报告

JD生成
我想招聘一个产品经理,工作内容是类chatgpt对话产品数据分析，请帮我生成一个招聘JD

语言翻译
你是一名资深的翻译，请将条条大路通罗马翻译成英文

写作文
请以荷塘月色为题目，帮我撰写一份五年级600字的作文

辩题立论
请帮我基于愚公应该搬家这个观点的持方生成一份立论

历史名人
现在你是孔子，模仿他说话的语气和风格与我对话

辅助决策
我正在决策是继续读研还是先参加工作，给我一个利弊清单，帮我决定我应该如何选择

简历润色
依据我的工作内容H5开发、小程序优化帮助我润色简历中的项目经验

视频脚本
帮我生成一个展现北京历史文化和科技发展的短视频拍摄脚本

专业计划
你是一个NBA资深教练，请为我制定一个篮球基本功15天计划

演讲稿
请生成演讲稿来介绍一款演讲稿生成工具的功能和使用方法。

PPT大纲
帮我生成一份户外运动发展趋势的调研汇报PPT大纲，并提供每

ChatGLM：常识

孙悟空有没有火烧赤壁

孙悟空是中国古代小说《西游记》中的重要人物，他是一位神通广大的猴子，经过修行后成为了斗战胜佛。在小说中，孙悟空并没有参与火烧赤壁的战斗。

火烧赤壁是中国古代历史上著名的战役之一，发生在公元 208 年，是刘备、孙权联合对抗曹操的军队而形成的三方对峙局面。在战役中，孙刘联军采取水战，用火攻来攻破曹军。曹操的军队在火势蔓延之下，陷入混乱，最终溃败。

虽然孙悟空没有在火烧赤壁中战斗，但他在小说中参与了其他许多重要的战役和冒险，最终成为了斗战胜佛，成为了中国文学和艺术中的重要形象之一。

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

复制

ChatGLM：常识

我的姥姥的闺女是我的什么人

你姥姥的闺女是你的妈妈或者姨妈。

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

复制

我奶奶的闺女呢

你奶奶的闺女是你的姑姑。

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

复制

奶奶的儿子呢

你奶奶的儿子是你的父亲或者叔叔。

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：

```python
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：

```python
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。

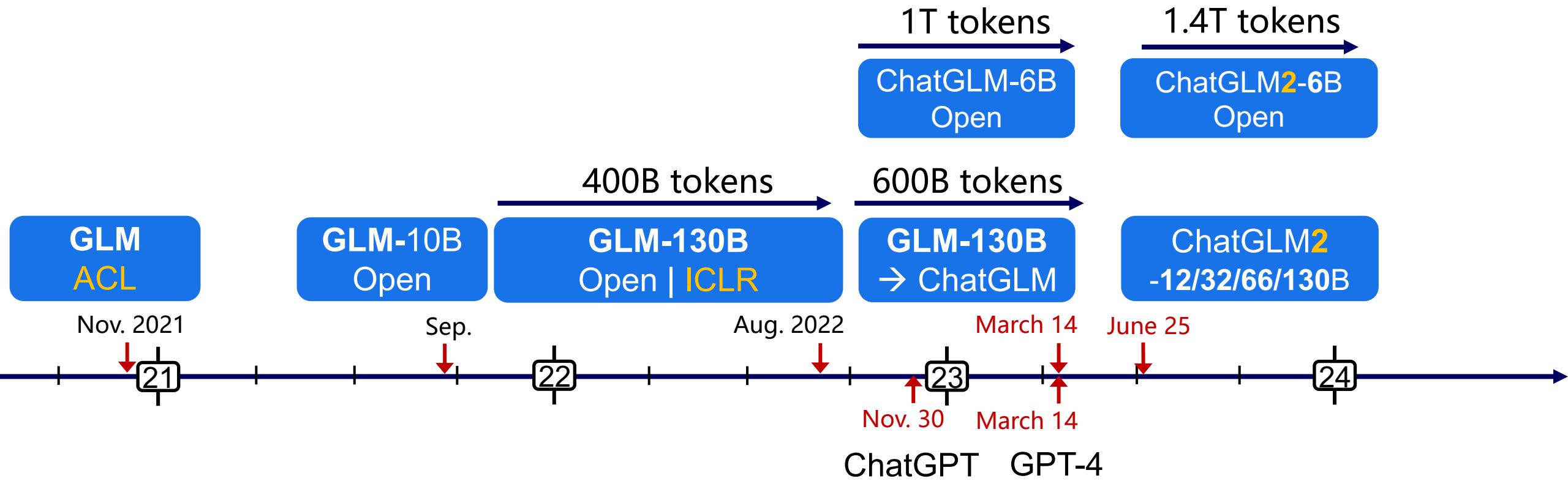
C-Eval

(The leaderboard is **earlier** than the models evaluated except GPT-4/ChatGPT/Claude)

#	Model	Creator	Submission Date	Avg ▾	Avg(Hard)	STEM	Social Science	Humanities	Others
0	ChatGLM2	Tsinghua & Zhipu.AI	2023/6/25	71.1	50	64.4	81.6	73.7	71.3
1	GPT-4*	OpenAI	2023/5/15	68.7	54.9	67.1	77.6	64.5	67.8
2	AiLMe-100B v2	APUS	2023/7/25	67.7	55.3	65.4	72.3	71.2	64
3	SageGPT-V0.2	4Paradigm	2023/7/25	66.6	61.1	67.9	76.6	66.9	54.9
4	SenseChat	SenseTime	2023/6/20	66.1	45.1	58	78.4	67.2	68.8
5	InternLM	SenseTime & Shanghai AI Laboratory (equal contribution)	2023/6/1	62.7	46	58.1	76.7	64.6	56.4
6	ChatGLM2-12B	Tsinghua & Zhipu.AI	2023/7/26	61.6	42	55.4	73.7	64.2	59.4
7	UniGPT	Unisound	2023/7/26	60.3	46.4	57.7	69.3	58	59
8	Instruct-DLM-v2	DeepLang AI	2023/7/2	56.8	37.4	50.3	71.1	59.1	53.4
9	GS-LLM-Alpha	共生矩阵科技（深圳）有限公司	2023/7/26	55.6	35.8	47	69.7	61.8	52
10	DFM2.0	AISpeech & SJTU	2023/7/10	55.4	38.3	47.5	64.6	58.7	58.2
11	ChatGPT*	OpenAI	2023/5/15	54.4	41.4	52.9	61.8	50.9	53.6
12	Claude-v1.3*	Anthropic	2023/5/15	54.2	39	51.9	61.7	52.1	53.7

Screenshot 2023, 0801

Tsinghua & Zhipu.AI GLMs

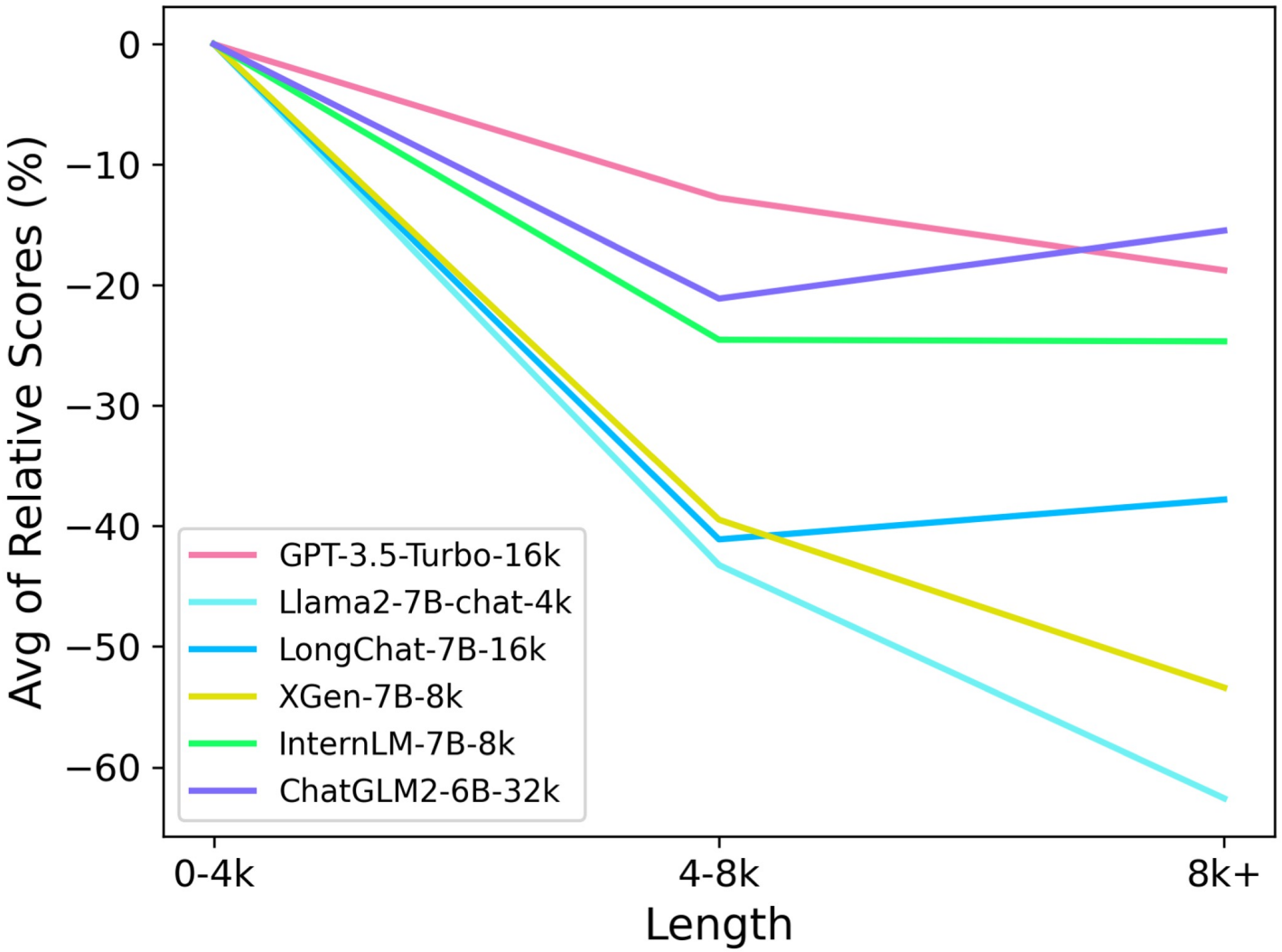


ChatGLM-6B → ChatGLM2-6B

- **Performance:**
  - **1.4T** bilingual tokens and human preference alignment training;
  - MMLU（+18%）、CEval（+33%）、GSM8K（+572%）、BBH（+60%）;
- **Context Length:**
  - Context length of the base model: **2K → 32K;**
  - Context length of the chat model: **8K**
- **Inference:**
  - The inference speed increases by **42%**;
  - W/ INT4 quantization, 6G GPU mem. supports dialogue length from **1K to 8K**;
- **Open license**:
  - The model weight is completely open for academic research;
  - Free commercial use;

https://github.com/THUDM/ChatGLM2-6B

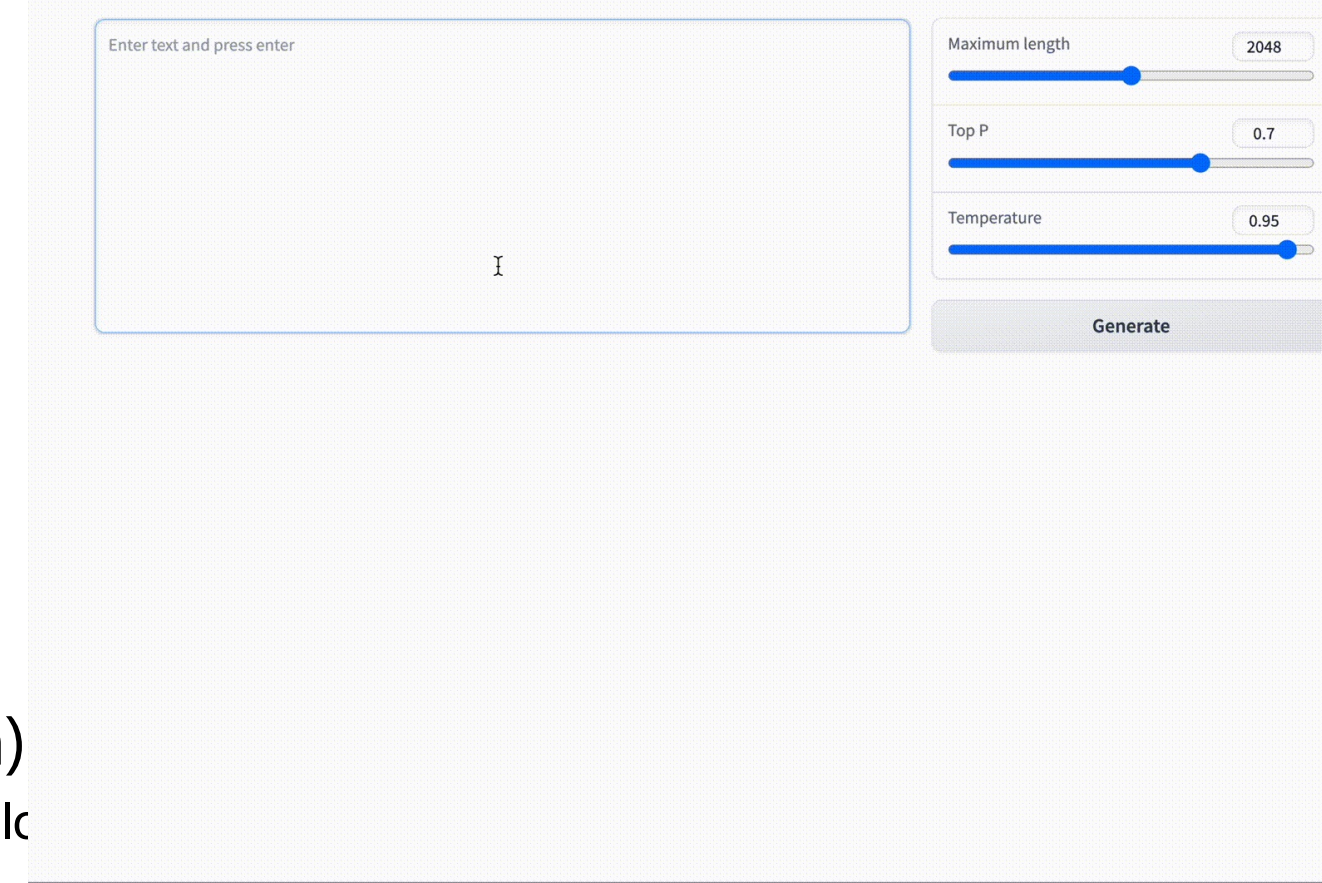
ChatGLM2-6B-32K



https://huggingface.co/THUDM/chatglm2-6b-32k

Install ChatGLM-6B/ChatGLM2-6B Locally

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 lo



ChatGLM-6B → ChatGLM2-6B

- **ChatGLM-6B**
  - 6B parameters
  - 1T tokens training data
  - 6G GPU mem (INT4)
- Mar. 14, 2023, open-soured model
- Mar. 16, 2023, **#1** on GitHub Trending
- Mar. 18-30, **#1** on HF Trending
- Jun. 23, 2023, **3M** downloads in HF

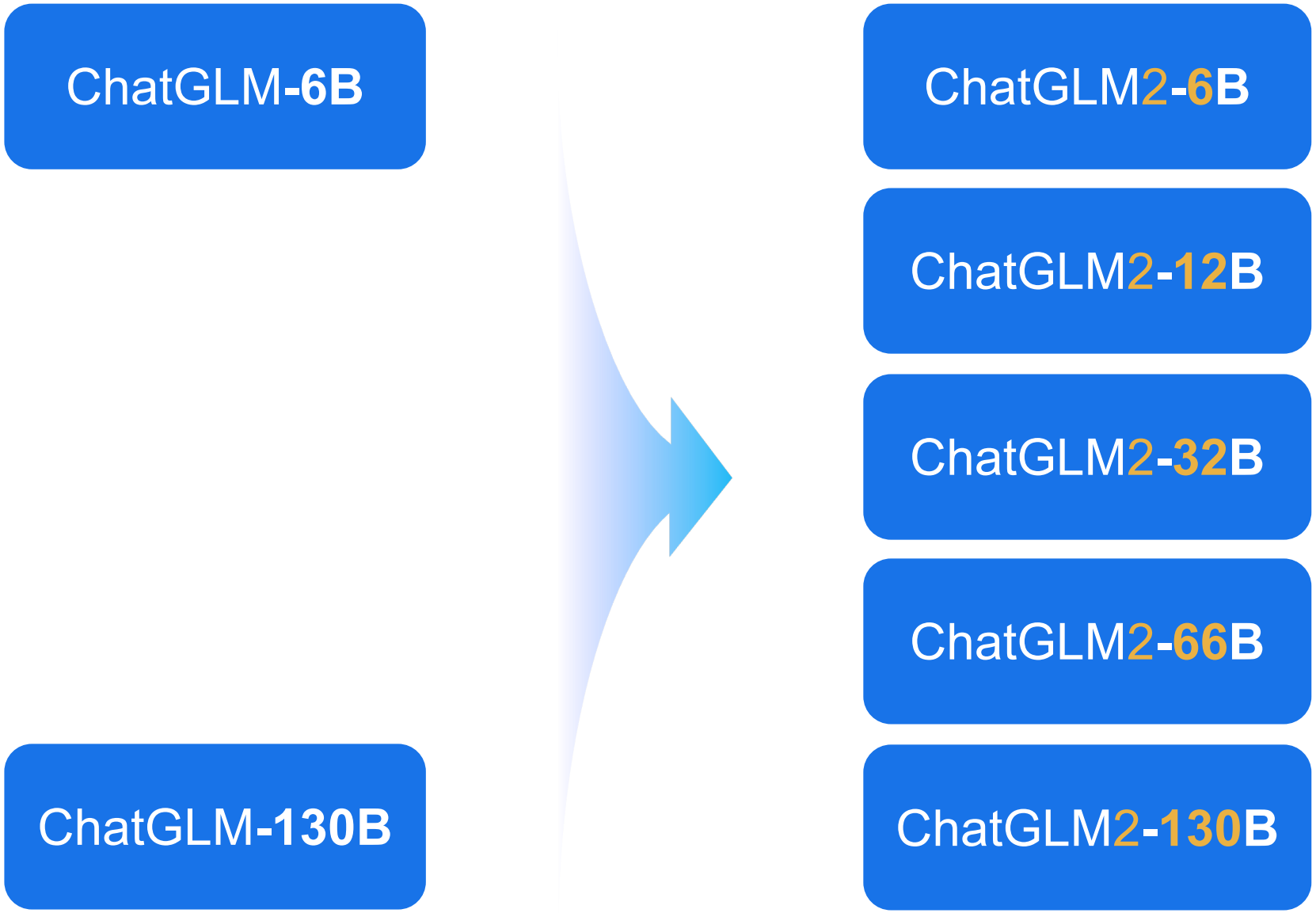
  30k stars on GitHub

- **ChatGLM2-6B**
  - 6B parameters
  - 1.4T tokens training data
  - 6G GPU mem (INT4)
- Jun. 25, 2023, open-soured model
- Jun. 27, 2023, **#2** on GitHub Trending
- Jun. 26-Jul. 8, **#1** on HF Trending
- Aug. 8, 2023, **1.8M** downloads in HF

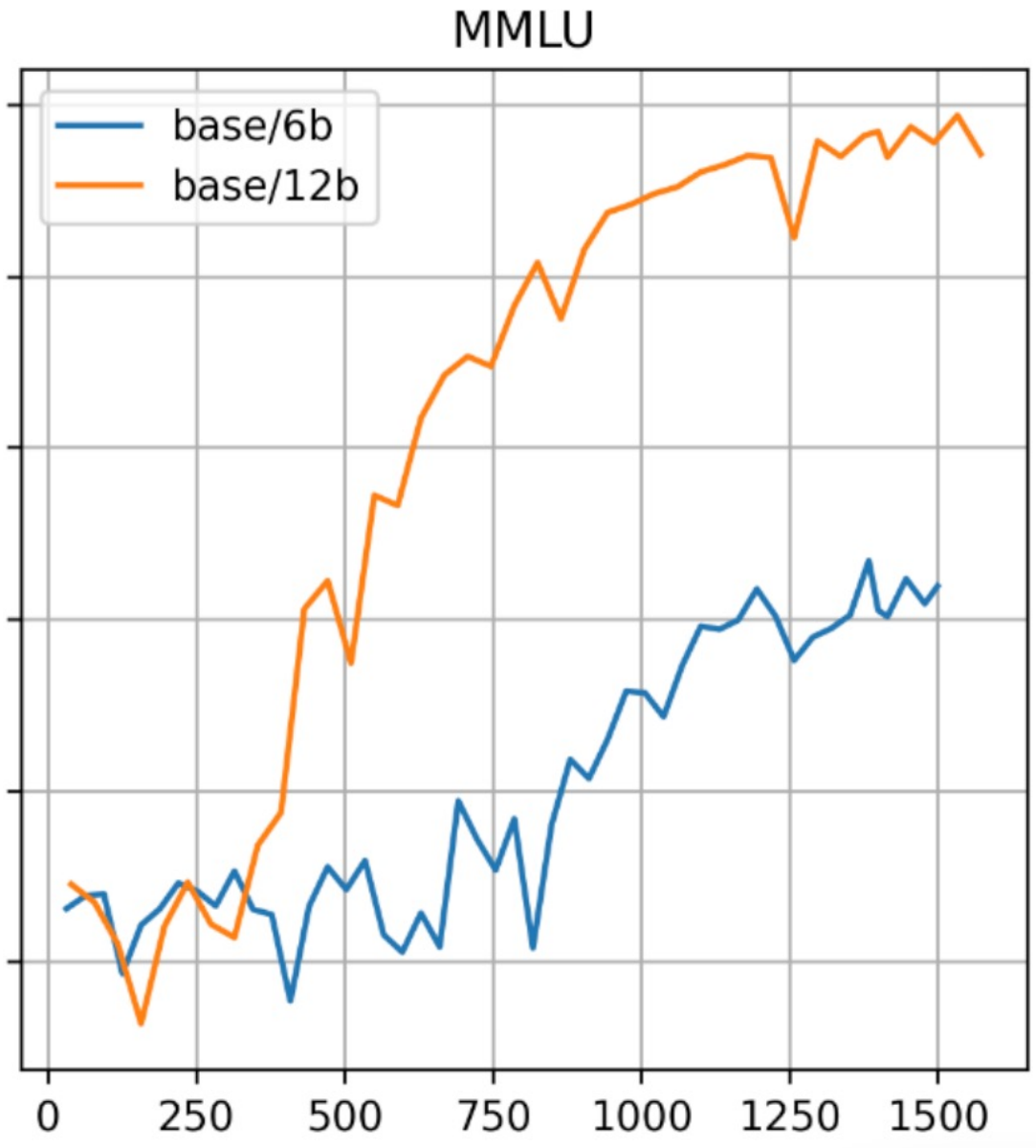
  10k stars in GitHub

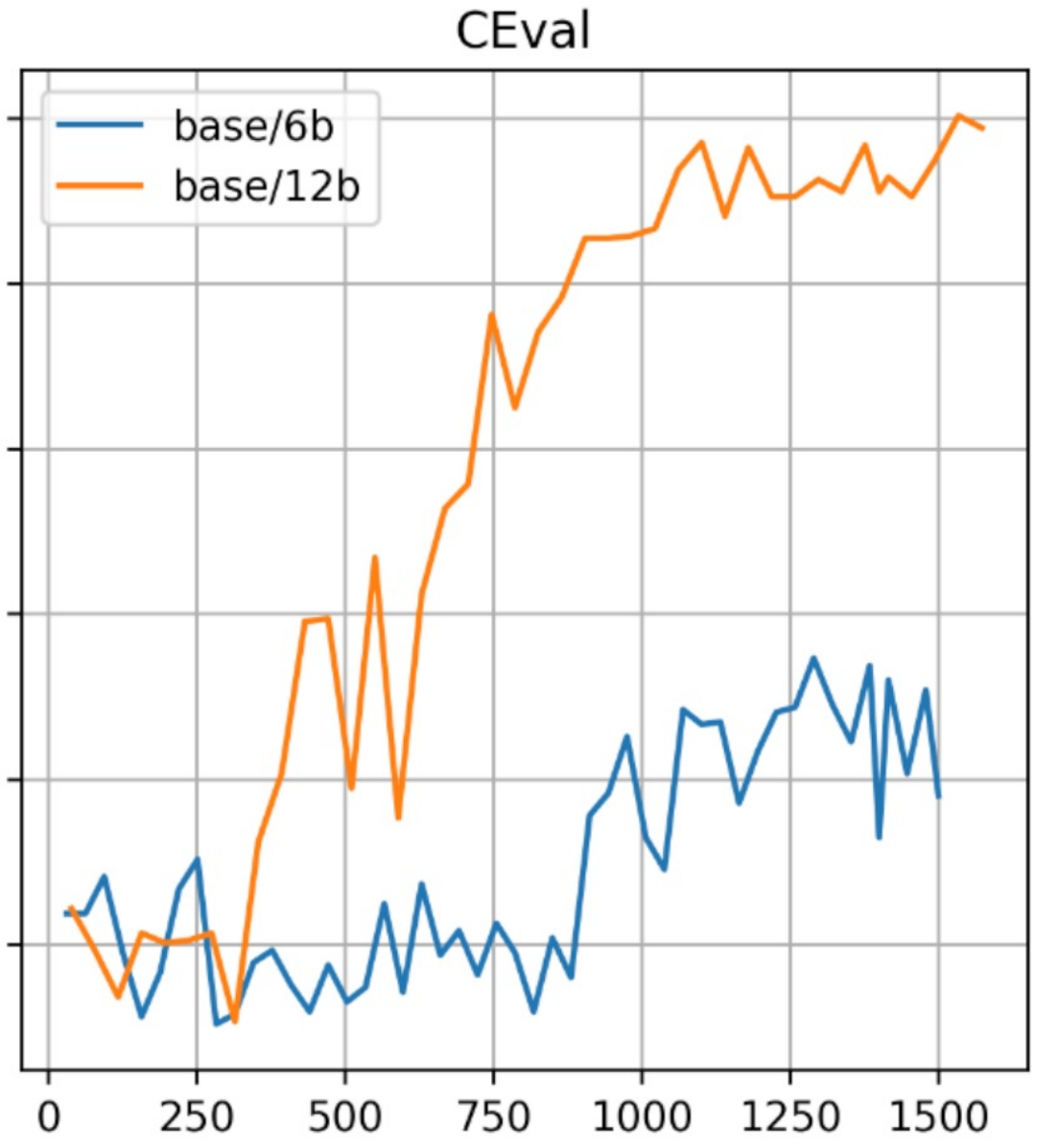
https://huggingface.co/THUDM

ChatGLM2-6B


ChatGLM-6B
ChatGLM-130B
ChatGLM2-6B
ChatGLM2-12B
ChatGLM2-32B
ChatGLM2-66B
ChatGLM2-130B

ChatGLM2-6B vs. ChatGLM2-12B





Tsinghua & Zhipu.AI GLMs

WebGLM
Open | KDD

ChatGLM
Open | ACL | ICLR

WebGLM = GLM + Search

Why do people try to maintain eye contact while communicating?

Why do people try to maintain eye contact while communicating?

People try to maintain eye contact while communicating because it is a form of body language which can help improve communication and create a positive impression. Eye contact is a non-verbal form of communication which can help show that you are actively listening and paying attention [1]. It can also make others more likely to like and respect you, making it an important tool for building and maintaining relationships [2]. Furthermore, a lack of eye contact can negatively impact communication, making misunderstandings more likely to occur. [3]

Like Dislike

[1] Why Eye Contact is Important in Communication

Eye contact is a type of body language that is extremely important during communication and conversation. Sometimes, our eyes and body language speak even more than words. Keeping eye contact with the person you are talking to shows that you are actively listening and paying attention.

[2] Why Eye Contact is Important during Conversation?

Liu et al. WebGLM: Towards An Efficient Web-enhanced Question Answering System with Human Preference. KDD'23

Tsinghua & Zhipu.AI GLMs

WebGLM
Open | KDD

VisualGLM
Open

ChatGLM
Open | ACL | ICLR

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。

https://github.com/THUDM/VisualGLM-6B

Tsinghua & Zhipu.AI GLMs

WebGLM
Open | KDD

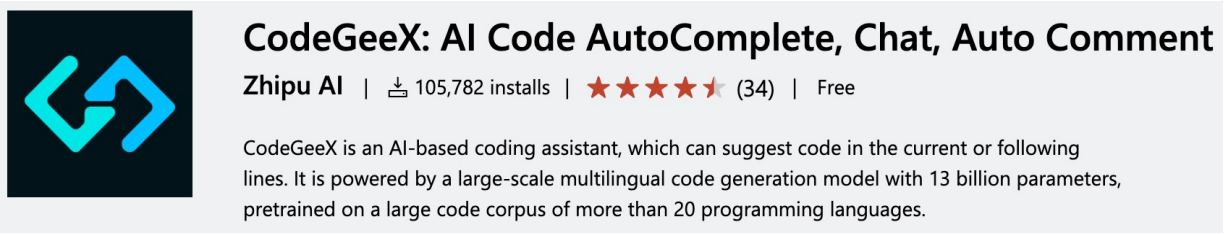
VisualGLM
Open

CodeGeeX
Open | KDD

ChatGLM
Open | ACL | ICLR

CodeGeeX2-6B

- **Performance**
  - **ChatGLM2-6B (1.4T text tokens) + 600B code tokens**
  - Python +57%, C++ +71%, Java +54%, JavaScript +83%, Go +56%, Rust +321%
- **Inference**
  - Inference speed 3X up from CodeGeeX-13B to CodeGeeX2-6B
- **AI Coding Assistant**
  - "CodeGeeX" plugin/extension on VS Code, Jetbrains for 100+ languages
  - "Ask CodeGeeX": code generation, code summarization, code translation, debugging, and comment generation, project-level generation



- **10M lines of code generated per day**
- Tens of thousands daily active users

https://github.com/THUDM/CodeGeeX2

CodeGeeX2-6B on HumanEval

HumanEval (Pass@1,10,100)

Model	Pass@1	Pass@10	Pass@100
CodeGen-16B-multi	19.2	34.6	55.2
CodeGeeX-13B	22.9	39.6	60.9
Codex-12B	28.8	46.8	72.3
CodeT5Plus-16B-mono	30.9	51.6	76.7
Code-Cushman-001	33.5	54.3	77.4
LLaMA-65B	23.7	-	79.3
LLaMA2-70B	29.9	-	-
CodeGen2.5-7B-mono	33.4	58.4	82.7
StarCoder-15B	33.2	61.0	84.7
CodeGeeX2-6B	**35.9**	**62.6**	**88.3**

Zheng et al., CodeGeeX: A Pre-Trained Model for Code Generation with Multilingual Evaluations on HumanEval-X. KDD’23

CodeGeeX2-6B on HumanEval-X

HumanEval-X (Pass@1)

Model	Python	C++	Java	JavaScript	Go	Rust	Overall
CodeGen-16B-multi	19.2	18.1	15.0	18.4	13.0	1.8	14.2
CodeGeeX-13B	22.9	17.1	20.0	17.6	14.4	4.3	16.0
Replit-code-v1-3B	22.0	20.1	20.1	20.1	12.2	8.6	17.2
CodeGen2.5-7B-multi	30.6	24.3	29.0	27.5	18.9	**20.1**	25.1
StarCoder-15B	35.5	28.2	**31.5**	**33.2**	21.3	17.8	27.9
CodeGeeX2-6B	**35.9**	**29.3**	30.8	32.2	**22.5**	18.1	**28.1**

> `n=20, t=0.2, top_p=0.95` for **Pass@1**.

Zheng et al., CodeGeeX: A Pre-Trained Model for Code Generation with Multilingual Evaluations on HumanEval-X. KDD'23

Tsinghua & Zhipu.AI GLMs

WebGLM
Open | KDD

VisualGLM
Open

CodeGeeX
Open | KDD

xTrimoP**GLM**-100B
Protein | Antibody

ChatGLM
Open | ACL | ICLR

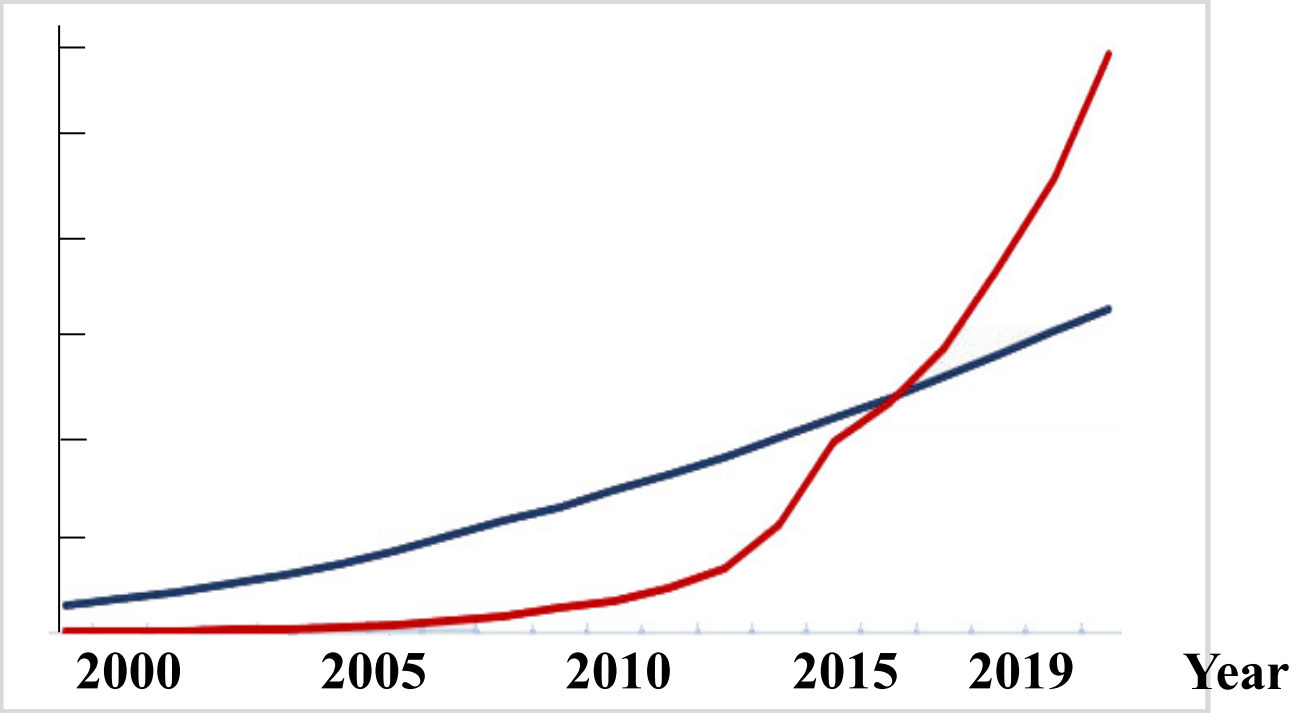
Protein Pre-training?

- "In general, the amino-acid sequence of a protein determines the 3D shape of a protein." [Anfinsen et al., 1950]
- Protein sequences are constrained to adopt particular 3D structures optimized for accomplishing particular functions. [Elnaggar et al., TPAMI'21]
- These constraints mirror the rules of grammar and meaning in NLP.

1D Sequence → 3D Structure → Function

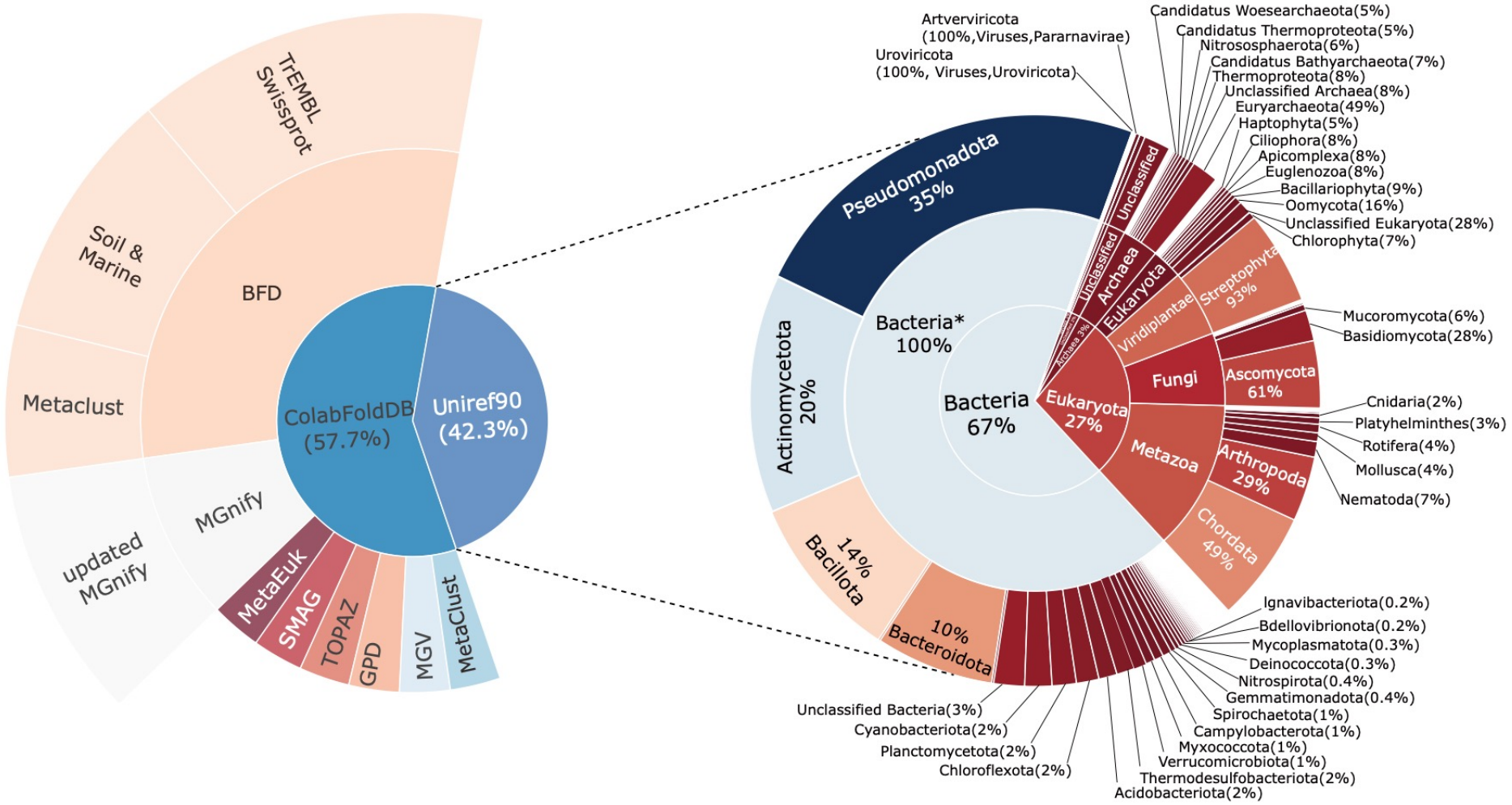
Protein Pre-training?

known protein sequences grows in the **exponential** order.
Known protein structures grows in the **linear** order.



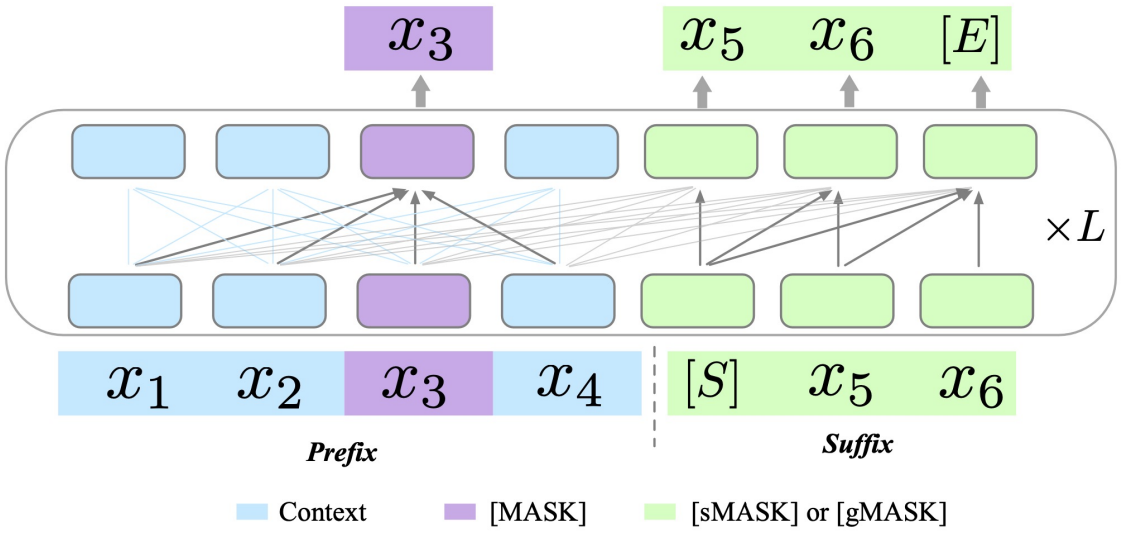
- There only have 200-300K experimental-determined protein structures, while owning billion-scale protein sequences.

Protein Data



- Data (500M seqs, 150B toks) = **2** * U90 (300M seqs) + **1** * CFDB (200M seqs)
  - ESM2 is trained on 65M seqs sampled from U50 and U90.
- During the pre-training phase, **1T** tokens are consumed.

Protein Pre-training



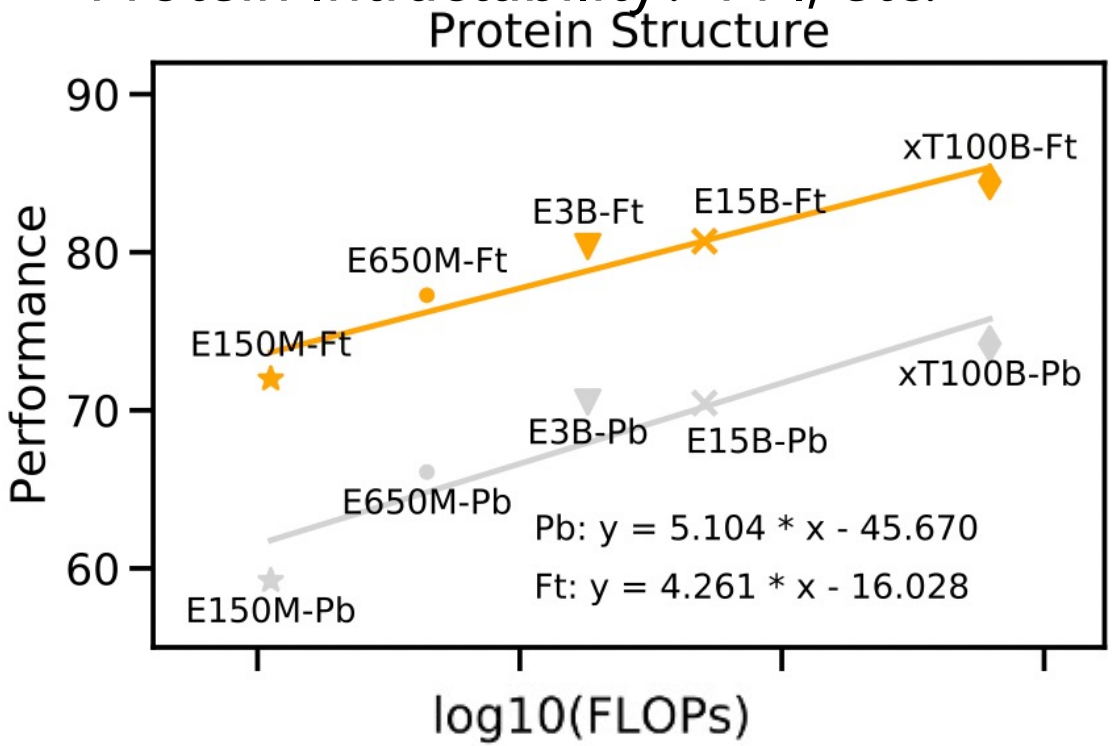
- **[MASK]:**
  - xTrimoPGLM acts like BERT.
- **[sMASK] OR [gMASK]:**
  - xTrimoPGLM acts like GPT.

Downstream Task	**Autoenc.**	**Autoreg.**	**Enc.-Dec.**	**GLM**	Example
Protein Understanding	✓	×	✓	✓	Contact Prediction
Protein Generation	×	✓	—	✓	Antibody Re-design
Representatives	ESM-1b[6], ESM2[7], Pro.BERT[22]	ProGen[9], ProGen2[13], ProtGPT2[23]	ProtTrans[8], Ankh[24]	xTrimoPGLM	/

Chen et al. xTrimoPGLM: Unified 100B-Scale Pre-trained Transformer for Deciphering the Language of Protein. bioRxiv, 2023

Protein Understanding

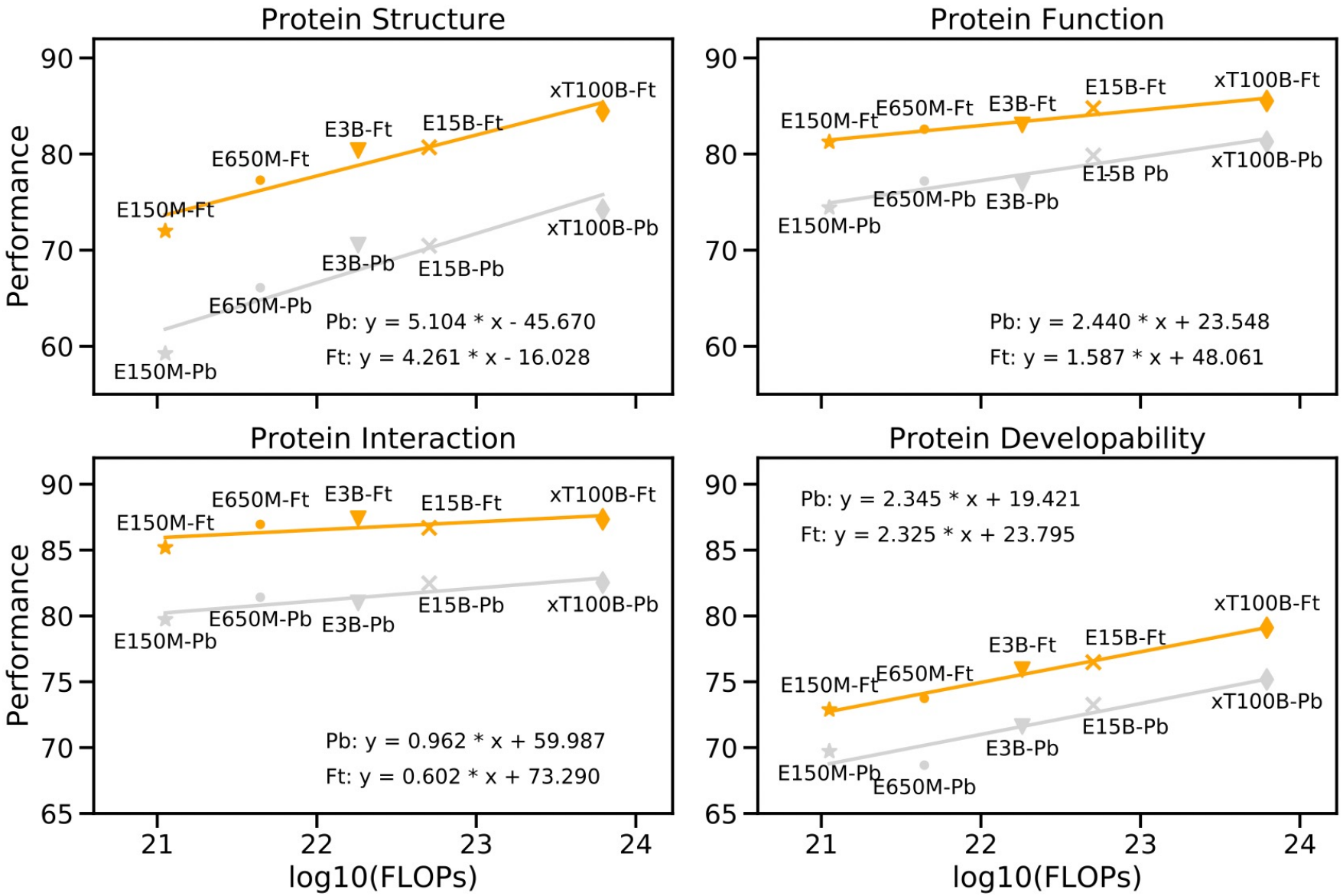
- Evaluation: 15 tasks covering four types of protein understanding tasks.
  - Protein Structure: Contact Prediction, SSR, etc.
  - Protein Function： Anti-resistance Prediction, etc.
  - Protein Developability： Stability Prediction & Solubility Prediction, etc.
  - Protein Intractability： PPI, etc.



- SOTA on 13/15 tasks
- Universally good on four types of tasks
- Scaling laws & Emergent ability

Chen et al. xTrimoPGLM: Unified 100B-Scale Pre-trained Transformer for Deciphering the Language of Protein. bioRxiv, 2023

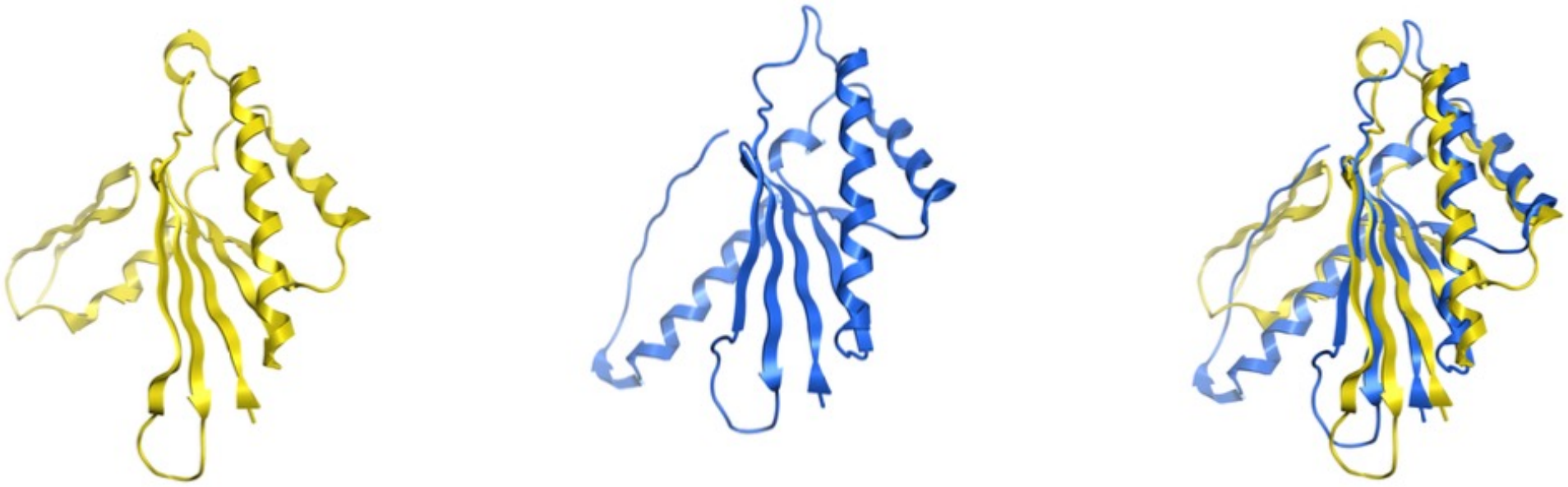
Protein Understanding



Chen et al. xTrimoPGLM: Unified 100B-Scale Pre-trained Transformer for Deciphering the Language of Protein. bioRxiv, 2023

Protein Design

- Empirical analysis of the structures (predicted by AF2) from generated sequences (Yellow for generated sequences while blue for natural sequences.)



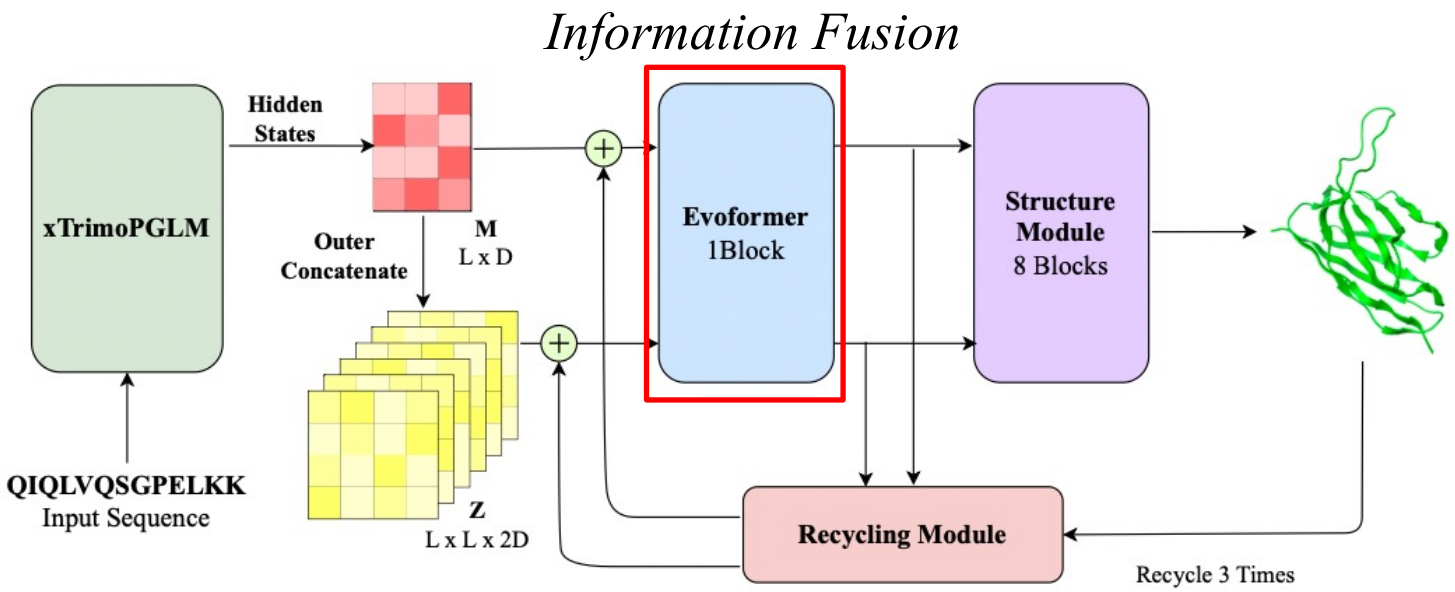
Case-1: TM-score=0.735, sequence identity=25.1%

xTrimoPGLM can generate structurally similar but low sequence identity novel protein sequences when compared to natural ones, which is meaningful for drug synthesis.

Chen et al. xTrimoPGLM: Unified 100B-Scale Pre-trained Transformer for Deciphering the Language of Protein. bioRxiv, 2023

xTrimoPGLM-Ab

- xTrimoPGLM-Ab: pre-trained on antibodies
  - xTrimoPGLM-Ab performs SOTA on the wide range of related-tasks.
  - Taking, antibody structural predictions, for example.



- Alphafold2:
  - 64 evoformer layer + MSA search + template search
- xTrimoPGLM-AbFold:
  - 1 evoformer layer + no MSA search + no template search

Chen et al. xTrimoPGLM: Unified 100B-Scale Pre-trained Transformer for Deciphering the Language of Protein. bioRxiv, 2023

Antibody Structure Prediction

Single Chain

Model	RMSD↓	TM-SCORE↑	HEAVY CHAIN RMSD↓			LIGHT CHAIN RMSD↓		
			H1	H2	H3	L1	L2	L3
AlphaFold2	1.225	0.951	1.254	1.091	2.826	0.89	0.723	1.329
OmegaFold	1.337	0.946	1.418	1.183	3.246	0.860	0.598	1.360
ESMFold	1.421	0.943	1.464	1.320	3.409	1.048	0.679	1.520
IgFold	1.261	0.945	1.324	1.126	2.998	0.948	0.589	1.318
xTrimoAbFold	1.089	0.958	1.176	0.912	2.472	0.811	**0.566**	1.038
xTrimoPGLM-AbFold	**0.9823** ±0.007	**0.961** ±0.001	**1.089** ±0.012	**0.866** ±0.011	**2.230** ±0.04	**0.779** ±0.017	0.573 ±0.008	**0.937** ±0.014

Double Chains

	RMSD↓	TM-SCORE↑	DOCKQ ↑	INFERENCE TIME ↓
ZDock	10.982	0.596	0.108	34h
ClusPro	5.899	0.792	0.404	1.3h
EquiDock	18.293	0.559	0.032	2m
HDOCK	2.032	0.926	0.705	3.2h
AlphaFold-Multimer	1.325	0.962	0.765	56.6h (original) 55m (faster MSA)
xTrimoPGLM-AbFold (evo 1)	1.304	0.962	0.762	**32s**
xTrimoPGLM-AbFold (evo 16)	**1.234**	**0.966**	**0.770**	82s

6,300X

103X

Chen et al. xTrimoPGLM: Unified 100B-Scale Pre-trained Transformer for Deciphering the Language of Protein. bioRxiv, 2023

Tsinghua & Zhipu.AI GLMs

WebGLM
Open | KDD

VisualGLM
Open

CodeGeeX
Open | KDD

xTrimoP**GLM**-100B
Protein | Antibody

ChatGLM
Open | ACL | ICLR

Open LLM Research

AgentBench Public

A Comprehensive Benchmark to Evaluate LLMs as Agents

C++ 279 10

GLM-130B Public

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

Python 6.9k 545

CodeGeeX2 Public

CodeGeeX2: A More Powerful Multilingual Code Generation Model

Python 1.9k 133

VisualGLM-6B Public

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

Python 3.2k 324

ChatGLM-6B Public

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

Python 33.1k 4.5k

ChatGLM2-6B Public

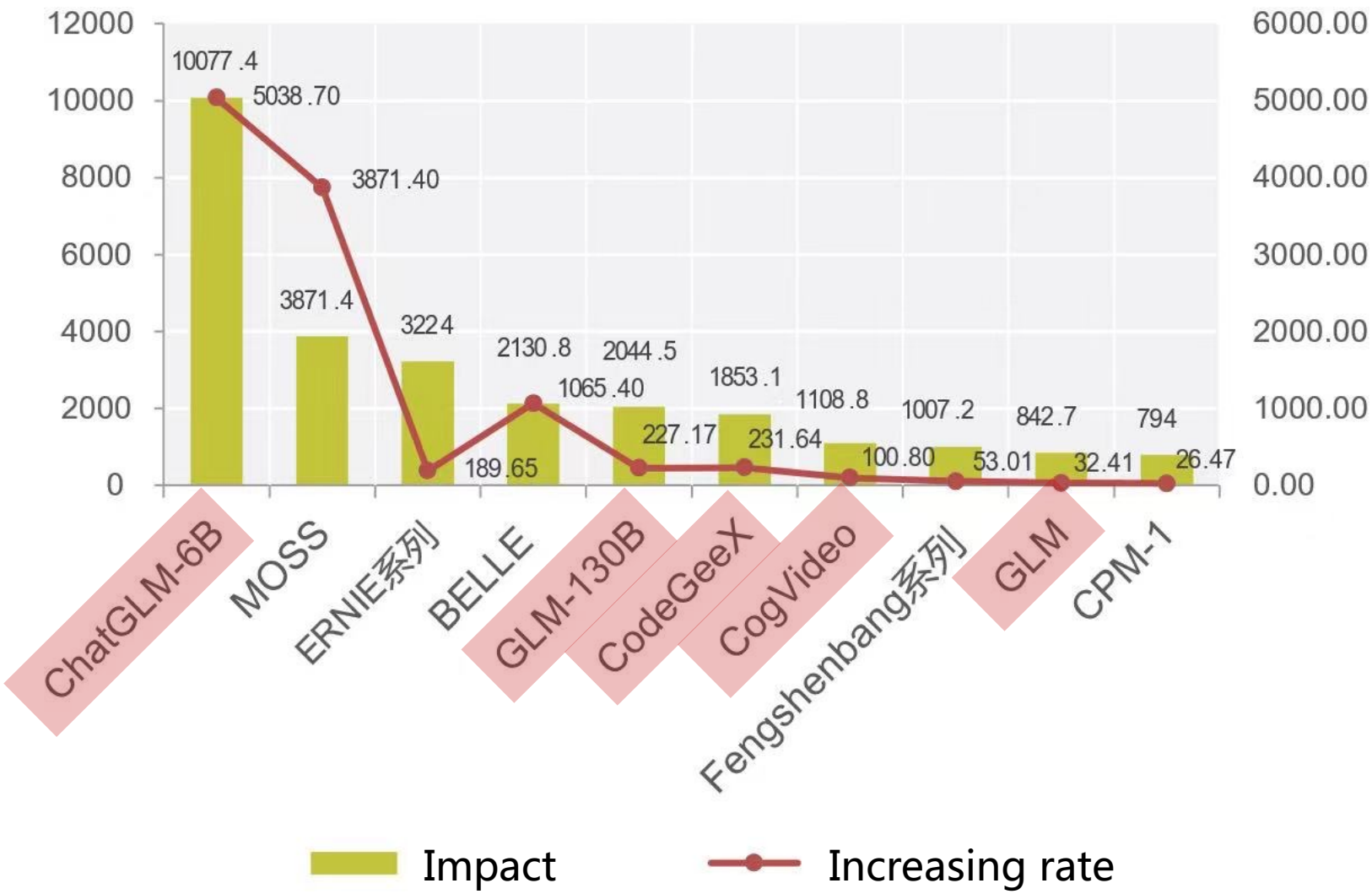
ChatGLM2-6B: An Open Bilingual Chat LLM | 开源双语对话语言模型

Python 11.1k 1.8k

https://github.com/THUDM

Open LLM Research

- ChatGLM-6B becomes the **most impactful open** LLM in China, according to the gov.
- 5 in Top 10 include
  - ChatGLM-6B
  - GLM-130B
  - CodeGeeX
  - CogVideo
  - GLM



Next

- **In-context Learning**
  - Strengthen ICLs
    - A sufficiently-trained LLM could be more powerful than we ever thought
  - Unlock new ICLs
- **Interactions with Env.**
  - With users
  - With environments, e.g., Web, tools.
- **Self Improvements**
  - self-instruct etc.

Thank You!

- **Techniques (major):**
  - Tsinghua Knowledge Engineering Group（KEG）
  - Zhipu.AI

- **GPUs (major):**

ZHIPU · AI

Thank You!

https://**chatglm.ai**

https://**github.com/THUDM**

https://**huggingface.co/THUDM**

@thukeg