Tele-FLM Technical Report

Xiang Li[1†], Yiqun Yao[1†], Xin Jiang[1†], Xuezhi Fang[1†], Chao Wang[2†], Xinzhang Liu[2†], Zihan Wang[2], Yu Zhao[2], Xin Wang[2], Yuyao Huang[2], Shuangyong Song[2], Yongxiang Li[2], Zheng Zhang[1], Bo Zhao[1], Aixin Sun[3], Yequan Wang[1*], Zhongjiang He[2*], Zhongyuan Wang[1], Xuelong Li[2], Tiejun Huang[1]

[1]Beijing Academy of Artificial Intelligence, Beijing, China
[2]Institute of Artificial Intelligence (TeleAI), China Telecom Corp Ltd, China
[3]School of Computer Science and Engineering, Nanyang Technological University, Singapore

Abstract

Large language models (LLMs) have showcased profound capabilities in language understanding and generation, facilitating a wide array of applications. However, there is a notable paucity of detailed, open-sourced methodologies on efficiently scaling LLMs beyond 50 billion parameters with minimum trial-and-error cost and computational resources. In this report, we introduce Tele-FLM (aka FLM-2), a 52B open-sourced multilingual large language model that features a stable, efficient pre-training paradigm and enhanced factual judgment capabilities. Tele-FLM demonstrates superior multilingual language modeling abilities, measured by BPB on textual corpus. Besides, in both English and Chinese foundation model evaluation, it is comparable to strong open-sourced models that involve larger pre-training FLOPs, such as Llama2-70B and DeepSeek-67B. In addition to the model weights, we share the core designs, engineering practices, and training details, which we expect to benefit both the academic and industrial communities.

1 Introduction

Large Language Models (LLMs) have been considered a remarkable approach for unsupervised learning, utilizing extensive data to achieve significant advancements. Large models based on decoder-only Transformers [64; 43] have demonstrated strong abilities on language understanding, generation, and in-context learning [10], *et al.*. Through downstream supervised fine-tuning (SFT) and task-specific alignments (*e.g.,* Reinforcement Learning from Human Feedback, RLHF) [41], LLMs have led to significant progress in the development of dialogue assistant applications with their human-level multi-turn interaction capabilities [40]. Furthermore, LLMs have demonstrated complex cognitive abilities as reflected by code interpretation and completion [37], mathematical problem-solving [35], logical reasoning [69], and agent-like actions [9]. Recently, LLMs have also shown potential to facilitate a unified sequence-to-sequence modeling paradigm for multimodal learning by treating image, video, and audio signals all as token sequences [57; 30]. This positions LLMs as pivotal for progress towards Artificial General Intelligence (AGI) [11].

Inspired by the superior performances of proprietary applications [40; 6], a plethora of open-sourced LLMs has been publicly available for both the English [60; 61; 42; 27; 58] and Chinese [71; 5; 7; 33] communities. The open-sourced models typically vary in size from 7B to 70B parameters, with their performances improving with model sizes and training FLOPs, which is described as scaling laws [29; 23]. Open LLMs can be classified into foundation language models, SFT models, and RLHF models.

[†]Indicates equal contribution.
[*]Corresponding authors.

Despite the growing prevalence and impressive evaluation performances, the high computational cost remains the major challenge in LLM development. In this study, we focus on alleviating the excessive computation by establishing a model-producing pipeline that streamlines the hyperparameter searching process, minimizes trial-and-error, and reduces restarts in training. For instance, the Llama technical report [60] assumed the use of around 2,048 A100 GPUs for 5 months, while a single Llama-65B training trial spanned only 21 days, constituting only 14% of the total GPU time. It indicates that open-source endeavors of pre-training LLMs may undergo redundant trial-and-error cycles that may consume enormous computational resources. In contrast, in this work, we reduce the total time cost due to restarts and trial-and-error to negligible levels. We believe that sharing our detailed techniques, engineering practices, and training dynamics [20], especially for LLMs exceeding the 50B scale, could benefit the community as well as contribute to green AI.

In this report, we introduce Tele-FLM (aka FLM-2), an open multilingual LLM with 52 billion parameters, which is pre-trained from scratch on a 2.0 trillion token corpus comprising texts from English, Chinese, and various other languages. Tele-FLM inherits and extends the low carbon techniques and fact-enhancing pre-training objectives from the FLM family [33]. The training of Tele-FLM has encountered no instability issue except hardware failures through the completed 2T tokens, and remains ongoing for more data. In addition to the model checkpoints, we release the details of data composition, model architecture, hyperparameter searching, and the full pre-training dynamics.

We evaluate Tele-FLM across multiple English and Chinese benchmarks. Regarding English language modeling, Tele-FLM has better Bits-Per-Byte (BPB) than Llama2-70B [61], demonstrating strong compression capabilities. The model also achieves lower BPB than Llama3-70B [2] and Qwen1.5-72B [5] on Chinese corpora, showcasing its multilingual nature. With fewer English training tokens and smaller models, Tele-FLM matches Llama-65B and is comparable to Llama2-70B in English foundation model evaluation. As for Chinese foundation model evaluation, Tele-FLM matches the overall performance of larger multilingual models trained with a similar amount of data (*e.g.,* DeepSeek-67B [7]). On certain tasks, it surpasses larger models trained with significantly more data (*e.g.,* Qwen1.5-72B).

The remainder of this report is structured as follows: Section 2 delves into the specifics of pre-training data processing. Section 3 details our model architecture, tokenizer, infrastructures, training techniques, and hyperparameters. In Section 4, we illustrate the pre-training dynamics and conduct BPB-based evaluation and analysis. Benchmark evaluation in both English and Chinese are provided in Section 5. Section 6 discusses some common issues and lessons learned. Section 7 reviews related literature. We conclude our work and look to the future in Section 8.

2 Pre-training Data

Our training dataset comprises a variety of domains, as detailed in Table 1. We build a custom pipeline on spark cluster for massive data processing and apply custom functions to each subset. The pipeline includes text extraction from HTML/WARC, cleaning and paragraph-level deduplication with heuristic rules, model-based quality filtering and document-level deduplication with MinHash [8] algorithm. We obtain 2T tokens after all the procedures, and the distribution ratio between English and Chinese data is roughly 2:1. We incorporate more English data because of its higher quality, especially regarding the WebText domain. Additionally, in line with the methodology of GPT-4, we collected some instruct data and incorporated it into our pre-training data after removing the test sets of common datasets using the strict n-gram-based method. We deliberately avoid "training on the test set" or any other benchmark-oriented trick.

WebText. CommonCrawl[1] is often considered to be a repository containing diverse human experience and rich knowledge (especially long-tail knowledge). However, the high-quality sources in CommonCrawl are primarily concentrated in the English segment, with the Chinese content exhibiting relatively lower information density and quality. We use the latest CommonCrawl dumps from RedPajama [15] and incorporate WudaoCorpora [77] and similar Chinese-specific datasets together to form a large web-text dataset. We apply custom heuristic rules and a FastText [28] classifier to

[1] `https://commoncrawl.org/`.

Table 1: Pre-training data. For each subset of our 2T pre-training tokens, we detail the language, the sampling proportion, the number of epochs completed during training, and the disk size.

Domain	Language	Sampling Prop.	Epochs	Disk Size
WebText	en, zh	75.21%	1.0	5.9 TB
Code	code, zh	9.81%	1.0	528.1 GB
Book	en, zh	7.17%	0.8	647.6 GB
WorldKnowledge	multi., en, zh	2.87%	2.5	67.5 GB
QA	en, zh	2.12%	1.0	159.2 GB
AcademicPaper	en	0.99%	1.0	54.4 GB
Profession-Law	zh	1.04%	1.0	84.2 GB
Profession-Math	math	0.62%	2.0	6.1 GB
Profession-Patent	zh	0.14%	1.0	10.4 GB
Profession-Medical	zh	0.02%	1.0	1.2 GB
ClassicalChinese	zh	0.02%	2.5	0.5 GB

filter out low-quality content, cross-deduplicate for each language, and up-sample/down-sample each subset with regard to data quality. The ratio of English to Chinese is approximately 2:1.

Code. We incorporate multiple Github-like code datasets and post-process it to filter-out low quality and duplicated content. Simultaneously, we carefully assembled and curated a well-formed markdown dataset comprising Chinese technical articles.

Book. We collect books from various sources in both English and Chinese, such as Redpajama [15] and Gutenberg[2], among others. We develop a series of cleaning steps to remove redundant formatting, garbled text, formula errors, duplicated paragraphs, and other unwanted content from the books. After interleaved deduplication on document level, we finally obtain a high-quality book dataset. The ratio of English to Chinese is nearly 1:1.

WorldKnowledge. To enrich the model's knowledge base and common sense, we add Wikipedia dumps[3] from 2024 period to our training set, covering 22 languages: bg, ca, cs, da, de, en, es, fr, hr, hu, it, ja, nl, pl, pt, ro, ru, sl, sr, sv, uk, zh. We first process these dumps via Online Language Modelling Dataset Pipeline [59] to clean up format; then a meticulous multi-lingual cleaning function is applied to remove reference and subsequent content, which tend to be irrelevant to the main text.

QA. We use StackExchange dataset provided by RedPajama-Data [15]. Furthermore, similar Chinese datasets are collected and incorporated into the training after filtering out those QA pairs with low information content. The ratio of English to Chinese in this subset is roughly 1:2.

AcademicPaper. We use arxiv dataset collected and processed by RedPajama-Data. This dataset is processed following a Llama-like procedure, which mainly focuses on clearing useless or redundant formats for better language modeling.

Profession. To enhance the model's capacity in various professional fields, we decide to include some specific domains in our dataset, including medical, law, patent, and math. Some subsets are from open-source data, such as Wanjuan-Patent [21] and MathGLM [74]. We post-process each subset independently to address formatting issues, private information disclosure, *et al.*.

ClassicalChinese. In order to improve the model's understanding of traditional Chinese culture and its capability in classical Chinese, we carefully collect classic Chinese ancient books and poetry. These materials are more credible than those found in web texts; therefore, we assign them a larger weight during sampling.

3 Pre-training Details

3.1 Model Architecture

We adapt the architecture of FLM-101B [33] as a backbone with several modifications. FLM-101B follows the standard GPT-style decoder-only transformer architecture [43], with pre-normalization

[2] `https://www.gutenberg.org/`.

[3] `https://dumps.wikimedia.org/`.

Table 2: Detailed model architecture. The model configuration of Tele-FLM$_{\mu P}$ is a reduced version of Tele-FLM with a smaller hidden size.

Models	Layer Num	Attention Heads	Hidden Size	FFN Hidden Size	Vocab Size	Context Length	Params Size (M)
Tele-FLM	64	64	8,192	21,824	80,000	4,096	52,850
Tele-FLM$_{\mu P}$	64	4	512	1,344	80,000	4,096	283

Table 3: Tokenizer compression ratio. Tokenizer Compression Ratio is defined as the ratio of token length to the original UTF-8 text length. Smaller values indicate better compression. We report the compression ratios of GPT-4, Llama1/2, Llama3, and Tele-FLM on various domains in our training set, as well as the weighted average.

Tokenizer	Vocab Size	Compression Rate						
		English	Chinese	Classical Chinese	Code	Multilingual	Mathematical	Weighted Avg.
GPT-4	100k	0.221	0.420	0.478	0.267	0.303	0.508	0.291
Llama1/2	32k	0.262	0.515	0.558	0.367	0.314	0.974	0.356
Llama3	128k	0.220	0.294	0.353	0.267	0.274	0.508	0.251
Tele-FLM	80k	0.248	0.235	0.307	0.363	0.340	0.965	0.261

and adds a LayerNorm to the last layer's output. Meanwhile, we apply scalar multipliers to: (1) the output of the word embedding layer and (2) the final output hidden states before softmax. We leave these multipliers tunable in pre-training to control the numerical flow. For example, the output multiplier may benefit training by modulating the entropy of the vocabulary distribution.

Building on FLM-101B, we further optimize the model structure for Tele-FLM. Specifically, We use RMSNorm [80] for normalization and SwiGLU [50] for the activation function. We roll back to use Rotary Positional Embedding (RoPE) [53] without Extrapolatable Position Embedding (xPos) [55], untie the embedding layer with language modeling head, and disable linear bias in the attention and all MLP modules. One mini version named Tele-FLM$_{\mu P}$ is used to search hyper-parameters here. Table 2 details the architecture of both Tele-FLM and Tele-FLM$_{\mu P}$.

3.2 Tokenizer

The key to training a text tokenizer is to make a better trade-off between compression ratio and vocabulary size. English-focused tokenizers like GPT-4 or previous Llama series often underperform in compressing Chinese text. In order to guarantee Tele-FLM's text compression ratio within Chinese while maintaining performance under multilingual setting, we train a tokenizer that aligns closely with the pre-training data distribution. We sample 12 million diverse text samples from our pre-training dataset as the tokenizer's training dataset, including multilingual texts with a primary focus on Chinese and English, code snippets, classical Chinese literature, and mathematical content. We train the tokenizer with Byte-level BPE (BBPE) algorithm [65].

Table 3 details the tokenizers of Tele-FLM, GPT-4, and the Llama family. The tokenizer of Tele-FLM outperforms GPT-4 and Llama series in both Chinese and Classical Chinese and is comparable with their performances in English, code, and multilingual content. In math, our tokenizer aligns with Llama2 while slightly trailing GPT-4. Overall, Tele-FLM tokenizer showcases a superior compression ratio for Chinese text and satisfactory performance in English. While slightly behind Llama3, Tele-FLM outperforms other approaches on average compression ratio by a large margin.

3.3 Cluster Hardware

Tele-FLM is trained on a cluster of 112 A800 SXM4 GPU servers, each with 8 NVLink A800 GPUs and 2TB of RAM. The nodes have heterogeneous CPU architectures: 96 nodes with Intel 8358 (128× 2.60GHz) CPUs and 16 nodes with AMD 7643 (96× 2.30GHz) CPUs. All nodes are interconnected via InfiniBand (IB). The training process lasts around two months, including downtime due to unexpected factors. As a comparison of infrastructures, Llama3 [2] is pre-trained on at least 49,152 Nvidia H100 GPUs (in contrast to our 896× A800). Meta also claims to have

the equivalent of 600k H100 GPUs for future computing power[4]. With this significant gap in total resources, computational efficiency and success rate are critical for average entities.

3.4 Parallelism

Tele-FLM utilizes 3D parallel training, combining the prevailing methodologies: data parallelism, tensor parallelism, and pipeline parallelism.

Data parallelism [63] is a well-established distributed training method, in which the samples in a batch are partitioned and distributed across multiple devices and processed simultaneously. No inter-device communication is involved in the forward and backward computation, while the gradient is aggregated at the end of each step.

Tensor parallelism [51] splits specific neural network tensors across multiple devices and computes via inter-device communication. In Tele-FLM training, tensor parallelism is mainly applied to the attention and feed-forward modules.

Excessive use of tensor parallelism may escalate GPU communication overheads and reduce the training speed. To alleviate this, we integrate pipeline parallelism [39] that partitions the model at the layer level.

3D parallelism incorporates these parallel approaches, prioritizing allocation of tensor parallelism groups with higher communication overheads to the same node, thereby maximizing intra-node communication and minimizing inter-node communication. The parallel training setup for Tele-FLM is a mixture of 4 tensor parallel, 2 pipeline parallel, and 112 data parallel.

Additionally, we partition inputs to the Transformer's LayerNorm and Dropout layers along the sequence length dimension with sequence parallelism [31], yielding further GPU computational and memory savings. Furthermore, we utilize Distributed Optimizer module from Megetron-LM[5] [46] with optimization. This optimizer further reduces GPU memory consumption by partitioning optimizer states with larger memory footprints across the data parallel dimension.

3.5 Hyperparameter Search

Effective hyperparameter tuning may accelerate the loss reduction and ensure convergence, making it crucial for model training. However, the high cost of training large models often renders exhaustive grid searches impractical. Hence, we employ μP [73] for optimal parameter search. The Tensor Programs theories [72; 36] reveal universal relations in the training dynamics across a series of models, with their widths approaching infinity. For certain hyperparameter classes, this leads to a parameterized mapping for their optimal values between small and large widths. Generally, under μP transfer, wider models will consistently achieve lower loss than narrower ones when trained on identical data [73]. Consequently, if a narrow model converges, its wider counterparts will always converge.

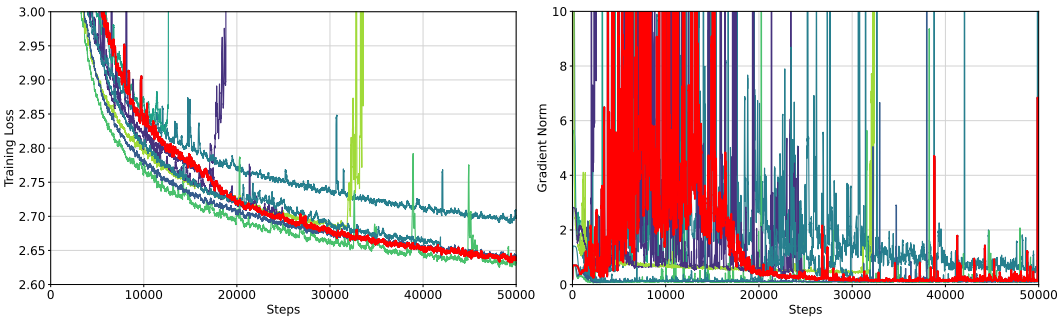
Based on this approach, we set a small model, namely Tele-FLM$_{\mu\text{P}}$, for grid search purpose. As demonstrated in Table 2, this small model's architecture is different from Tele-FLM only in width. With a fixed layer number of 64 and attention head dimension of 128, we reduce the hidden size to 512. This modification results in 4 attention heads and a feed-forward hidden size of 1344. Due to its smaller size, Tele-FLM$_{\mu\text{P}}$ allows for significantly more experimental runs within fixed time and resource constraints.

We search 7 hyperparameters: Learning Rate for vector-like and matrix-like weights, the Minimum Learning Rate at the end of the schedule, the initialization Standard Deviation for vector-like and matrix-like weights, the scaling factor for the embedding layer (namely Input Mult), and the scaling factor for the output hidden state in the final layer (namely Output Mult). For the definitions of vector/matrix-like weights and the μP transferring formula we apply, please refer to [75] and [73]. We use truncated normal distribution for model initialization.

Figure 1 illustrates the loss and gradient norm dynamics of 9 hyperparameter combinations for the grid search, which are selected based on our prior knowledge of model configurations. We choose

[4] `https://www.instagram.com/reel/C2QARHJR1sZ/?hl=en`.

[5] `https://github.com/NVIDIA/Megatron-LM`.



(a) Loss curves for grid search. **(b)** Gradient norm curves for grid search.

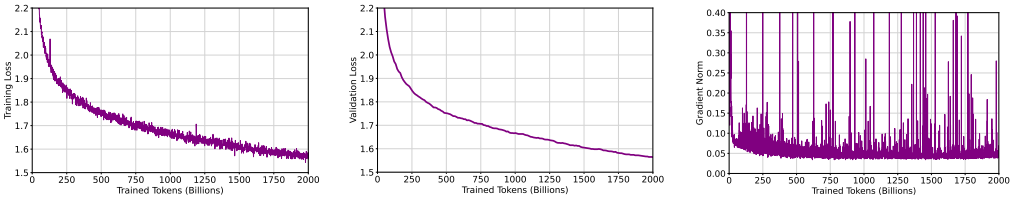
Figure 1: Experimental curves of hyperparameter search based on μP.

Table 4: Tele-FLM Training Hyperparameters.

Searched Hyperparameters		Non-Searched Hyperparameters	
Learning Rate	1.5e-4	LR Schedule Type	cosine
Matrix Learning Rate	1.5e-4	LR Schedule (tokens)	2.5T
Minimum Learning Rate	1.5e-5	Warmup Step	2,000
Standard Deviation	4e-3	Clip Grad	1.0
Matrix Standard Deviation	4.242e-3	Weight Decay	0.0
Input Mult	1.0	Batch Size (tokens)	5,505,024
Output Mult	3.125e-2	RoPE Theta	10,000

the hyperparameters represented by the red line for final training after assessing the rate of loss decrease, trend stability, and gradient norm stability. Using μP, we derive the optimal hyperparameter configuration for the final 52B model based on this searched result, which is detailed in Table 4. A more fine-grained search can be conducted with expanded time and budgets.

4 Loss Dynamics and BPB Evaluation



(a) Training loss curve. **(b)** Validation loss curve. **(c)** Training gradient norm curve.

Figure 2: Pre-training curves for Tele-FLM w.r.t. amount of data in billion tokens.

We present the curves for training and validation loss and gradient norm on our pre-training data distribution in Figure 2. Figure 2a shows that the training process of Tele-FLM succeeds with a single, stable run without any divergence. This result is predictable with our μP hyperparameter search mentioned above. Figure 2b indicates that the loss curve generalizes well to validation data without saturation or overfitting. Figure 2c presents the gradient norm. We observe that the reduction in language modeling loss translates well into improvements on downstream tasks.

Language modeling is compression [16]. Evaluation metrics related to language perplexity (PPL) are well-known to be closely connected to compression ratio. Moreover, these metrics usually exhibit more stable scaling behavior, making them an authentic foundation of downstream task performance (which is usually measured by more complex and nonlinear metrics [48]). For PPL-related evaluation, we use Bits-Per-Byte (BPB) [38; 18] as our metric, which considers both per-token loss and the

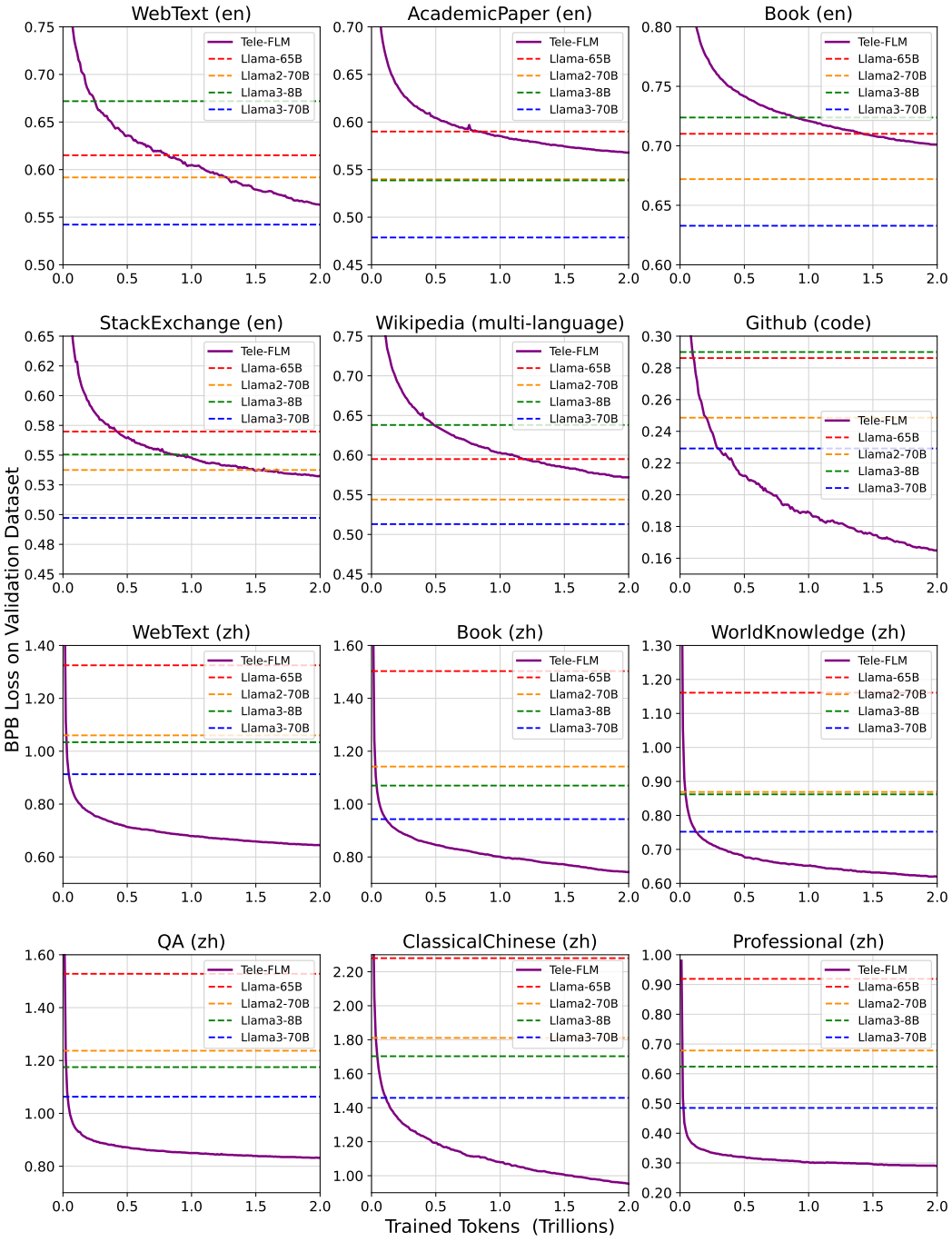


Figure 3: BPB curves of Tele-FLM on representative English (en), Chinese (zh), multi-language, and code validation datasets, compared with Llama series.

influence of domains and tokenizers. Specifically, on a test corpus in a certain domain, if the total loss is close, a model that tokenizes with a better compression ratio is preferred by the BPB metric.

For the English language, we break down the BPB evaluation into 6 different domains, represented by validation datasets from WebText[6], Github, Wikipedia, Books, ArXiv, and StackExchange, respectively. We compare with different versions of Llama, including Llama-65B, Llama2-70B, Llama3-8B, and Llama3-70B [2], to analyze how well Tele-FLM fits to compress English data.

[6]We use text from CommonCrawl and C4, which approximately represent the same source (broad web data).

Table 5: BPB of Tele-FLM, Llama family models, and Qwen1.5-72B on English datasets. BPB is computed for 6 dataset categories, with weighted sum results based on Llama [60] and Tele-FLM training data configurations. The best results are in boldface and second-best underlined.

	Model	WebText	Github	Wikipedia	Book	ArXiv	StackExchange	Weighted Sum L-Prop.[1]	Weighted Sum F-Prop.[2]
Loss	Llama-65B	1.650	0.543	1.297	1.791	1.205	1.293	1.572	1.485
	Llama2-70B	1.588	0.471	1.198	1.695	1.103	1.220	1.506	1.418
	Llama3-70B	1.729	0.597	1.300	1.886	1.042	1.388	1.642	1.556
	Qwen1.5-72B	1.996	0.592	1.433	2.107	1.111	1.393	1.878	1.773
	Tele-FLM (52B)	1.598	0.314	1.163	1.843	1.153	1.193	1.512	1.411
BPB	Llama-65B	0.615	0.286	0.595	0.710	0.590	0.570	0.602	0.574
	Llama2-70B	0.592	0.249	0.544	0.672	0.540	0.538	0.576	0.547
	Llama3-70B	**0.542**	0.229	**0.513**	**0.633**	**0.479**	**0.497**	**0.528**	**0.502**
	Qwen1.5-72B	0.642	0.234	0.601	0.717	0.521	0.515	0.620	0.586
	Tele-FLM (52B)	0.562	**0.164**	0.570	0.700	0.567	0.531	0.550	0.516

[1] L-Prop. (Llama [60] Proportion): 82% : 4.5% : 4.5% : 4.5% : 2.5% : 2.0%.
[2] F-Prop. (Tele-FLM Proportion): 75.17% : 13.48% : 3.56% : 5.26% : 1.46% : 1.07%.

Table 6: BPB of Tele-FLM, Llama family models and Qwen1.5-72B, on Chinese datasets. BPB is computed for 7 dataset categories, with direct average and weighted sum results based on Tele-FLM training data distributions.

	Models	WebText	Code	Book	World Knowledge	QA	Classical Chinese	Professional	Direct Average	Weighted[1] Sum
Loss	Llama-65B	1.773	1.236	2.029	1.586	2.076	2.819	1.215	1.819	1.782
	Llama2-70B	1.419	1.019	1.542	1.189	1.681	2.233	0.896	1.426	1.414
	Llama3-70B	2.152	1.264	2.210	1.722	2.568	2.844	1.109	1.981	2.114
	Qwen1.5-72B	2.260	1.405	2.520	1.751	2.888	2.748	0.908	2.069	2.243
	Tele-FLM (52B)	1.923	1.096	2.135	1.612	2.530	2.144	0.846	1.755	1.913
BPB	Llama-65B	1.325	0.744	1.503	1.161	1.528	2.280	0.919	1.351	1.326
	Llama2-70B	1.060	0.614	1.142	0.869	1.237	1.811	0.678	1.059	1.052
	Llama3-70B	0.913	0.498	0.943	0.752	1.063	1.458	0.485	0.873	0.897
	Qwen1.5-72B	0.759	0.537	0.871	0.663	0.951	1.237	0.329	0.764	0.759
	Tele-FLM (52B)	**0.643**	**0.478**	**0.741**	**0.619**	**0.831**	**0.949**	**0.290**	**0.650**	**0.646**

[1] Tele-FLM training set Proportion: 76.60% : 1.91% : 11.61% : 1.44% : 4.50% : 0.07% : 3.87%.

Figure 3 illustrates the BPB trends w.r.t. to the amount of our pre-training data (in trillion tokens). As training progresses, Tele-FLM surpasses Llama2-70B on WebText, Github, and StackExchange, outperforming Llama-65B and Llama3-8B on almost all datasets, demonstrating strong foundation abilities in English. Numerical results are presented in Table 5. Regarding the weighted sum of BPB, Tele-FLM outperforms Llama-65B, Llama2-70B, Qwen1.5-72B, and Llama3-8B on both Tele-FLM and Llama [60] weighting proportions. Note that Llama3-8B is trained on more than 15T tokens, and these results may indicate that scaling up the model size is still important, despite the rapid growth of the total amount of training data.

Similarly to English, we compute BPB across 7 domains with the corresponding Chinese validation data, namely WebText, Code, Book, World Knowledge, QA, Classical Chinese, and Professional. Results are visualized in Figure 3 (with "zh" suffix). Specific scores are provided in Table 6. On all these validation corpora, Tele-FLM demonstrates lower BPB than Qwen1.5-72B and the latest Llama3-70B model. Thus, we conclude that our foundation model achieves strong compression performance for Chinese without sacrificing its English language modeling abilities, and vice versa.

5 Benchmark Evaluations

5.1 English: Open LLM, HumanEval, and BBH

Benchmarks. We evaluate Tele-FLM on three public and widely-used English benchmarks: Open LLM Leaderboard[7], HumanEval [12], and BIG-Bench Hard [52].

[7] `https://huggingface.co/spaces/HuggingFaceH4/open_llm_leaderboard`.

- Open LLM Leaderboard is hosted on Huggingface and includes 6 key tasks to measure a model's performance on a variety of areas, such as commonsense inference, knowledge capacity, truthfulness, and maths. We report our model's results with the official evaluation tools (Language Model Evaluation Harness [19]). For the baseline models, we pick the results directly from the Open LLM Leaderboard.
- HumanEval, introduced by OpenAI, tends to evaluate the code generation ability of language models by measuring functional correctness of docstring-prompted output. We choose the pass@5 metric as a trade-off between representing model capability and the evaluation speed.
- Big-Bench Hard is derived from the BIG-Bench benchmark, a diverse evaluation suite that focuses on tasks believed to be beyond the capabilities of current language models. The Big-Bench-Hard, containing 23 challenging tasks, is specifically chosen to represent areas where language models did not surpass average human-rater performance, according to prior evaluations [56].

Table 7: Performance of Tele-FLM and baselines on English benchmarks.

Model	Average	ARC 25-shot	HellaSwag 10-shot	MMLU 5-shot	TruthfulQA zero-shot	WinoGrande 5-shot	GSM8K 5-shot	HumanEval zero-shot	BBH 3-shot
Llama2-70B	63.39	67.32	87.33	69.83	44.92	83.74	54.06	46.95	52.94
Llama2-13B	50.29	59.39	82.13	55.77	37.38	76.64	22.82	28.66	39.52
Llama-65B	56.98	63.48	86.09	63.93	43.43	82.56	37.23	33.54	45.54
Llama-13B	46.20	56.23	80.93	47.67	39.48	76.24	7.58	23.78	37.72
Tele-FLM (52B)	56.60	59.47	82.25	64.00	43.09	79.40	45.19	34.76	44.60

Results. Table 7 compares Tele-FLM to the Llama series. With 52B parameters and around 1.3T English pre-training tokens, Tele-FLM matches the overall performance of Llama-65B, which is trained on approximately 1.4T tokens. Regarding the nature of different subtasks, Tele-FLM shows advantages over Llama-65B on GSM8K [14] and HumanEval, which focus on reasoning capabilities, but performs slightly worse on some tasks that rely more heavily on knowledge. This disadvantage can potentially be mitigated with more pre-training data consumed. Besides, Tele-FLM achieves $> 90\%$ of the performances of Llama2-70B, which is larger in size and trained on a 2T token corpus.

5.2 Chinese: OpenCompass

Benchmarks. To measure the Chinese language and knowledge capabilities of our model, we conduct an evaluation using the OpenCompass[8] toolkit. Specifically, we choose the following tasks to evaluate the model's performance in multiple aspects: C-Eval [26] and CMMLU [32] (multi-subject knowledge), C3 [54] (reading comprehension), CHID [82] (Chinese culture and language understanding), and CSL [34] (keyword recognition).

Results. Table 8 shows evaluation results on Chinese benchmarks. On average, Tele-FLM achieves significantly higher scores than GPT-3.5 and comparable to GPT-4 and DeepSeek-67B [7], reaching 84% of Qwen1.5-72B's performance [5]. Note that Qwen1.5-72B is larger in size and trained with up to 3T tokens. On CHID and CSL, Tele-FLM shows leading performance among all the models compared. Interestingly, CHID is very specific to Chinese culture, while CSL comes from the scientific domain. This indicates Tele-FLM's potential to both quickly adapt to a specific language and benefit from general knowledge presented in different languages.

5.3 Evolution of Performance during Training

We automatically track the evaluation scores on sampled validation data for 8 of the evaluation benchmarks, as depicted in Figure 4. We observe that for all the tasks, evaluation score improves as pre-training and validation loss/BPB decreases. For knowledge-oriented English benchmarks, including ARC [13], HellaSwag [78], Winogrande [3], and MMLU [22], the performances increase smoothly with more data, which is intuitive regarding the task nature. For reasoning-oriented tasks including GSM8K and BBH, we observe a sharper increase, which indicates these tasks have more complex metrics and could possibly demonstrate emergent abilities. CMMLU is a knowledge-oriented Chinese benchmark. The sharper increase in CMMLU indicates that our Chinese training data is far from saturating, and further improvement can be expected with the ongoing training process.

[8] `https://opencompass.org.cn/home`.

Table 8: Performance of Tele-FLM and baselines on Chinese benchmarks. The results of Qwen1.5-72B and our Tele-FLM are locally computed with the OpenCompass toolkit, while other results are picked from OpenCompass leaderboard.

Model	Average	C-Eval	CMMLU	C3	CHID	CSL
GPT-4	76.64	69.90	71.00	95.10	82.20	65.00
GPT-3.5	61.86	52.50	53.90	85.60	60.40	56.90
Qwen1.5-72B	80.45	83.72	83.09	81.86	91.09	62.50
Qwen-72B	83.00	83.30	83.60	95.80	91.10	61.20
DeepSeek-67B	73.46	66.90	70.40	77.80	89.10	63.10
Tele-FLM (52B)	71.13	65.48	66.98	66.25	92.57	64.38

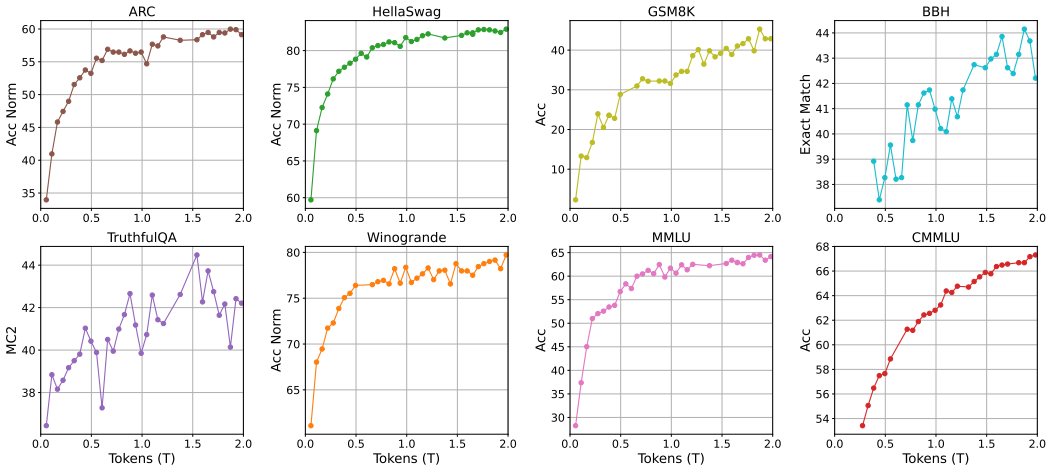


Figure 4: Evolution of performance evaluated by Language Model Evaluation Harness during training. Note that we sampled 20% examples for Hellswag and 30% examples for MMLU considering the time cost.

6 Lessons Learned

Lesson on Pre-training Data. We have the following observations in Tele-FLM's pre-training process. First, as is widely known, both quality and quantity of the data are critical for pre-training; however, when there should be a trade-off between quality and quantity, data quality might be prioritized. For our project, an English-Chinese data ratio of 2:1 works better than 1:1, likely because the average quality of the Chinese Web data we have is relatively low. Second, changing the data distribution midway sometimes leads to changes in gradient norm curves and potential divergence, while maintaining a fixed distribution is more stable. Another advantage of maintaining a fixed data distribution is that it allows for safer early-stop of the μP experiments. To conclude, the data processing should be as complete as possible before the pre-training starts.

Lesson on Hyperparameter Search. We observe that μP-based methods [73; 75] are effective and efficient in searching for the best hyperparameters and predicting the behaviors of the final large models. Specifically, prior experiences and the open-sourced learning rates are good starting points for hyperparameter search. Nevertheless, initialization standard deviation and output multipliers have more significant influences than commonly known.

Lesson on Loss Dynamics. First, the slope of the loss curve typically flattens after 500B tokens. Therefore, training should be restarted promptly if early loss values are unsatisfactory. Second, random loss spikes are common and acceptable if the gradient norm curve looks normal. We observe that our model recovers from all the spikes in the pre-training process, unlike the early open-sourced endeavors [81; 4; 79]. We speculate that modern Llama-like structures, especially those with non-bias designs and truncated normal initialization, combined with effective hyperparameter search, provide decent robustness against loss spikes. Another type of spike corresponds to consistent loss increases, which can be identified early with μP and avoided before the training begins.

Lesson on Gradient Norm. The early gradient norm curves are not strong indicators of training stability. In hyperparameter search, we observe divergence following various gradient curve patterns, yet with higher divergence probabilities associated with continuously increasing gradient trends.

7 Related Work

The idea of large foundation models originates from unsupervised pre-training with Transformer-based [64] architectures. Well-known examples of early foundation models include Bert [17], GPT-2 [43], and T5 [45]. GPT-3 [10] increases the model size to 175B and observes decent few-shot and zero-shot reasoning capabilities, which encourages a series of efforts to scale up foundation models [81; 47; 4; 79]. Research on scaling laws [29; 23; 24; 75] sheds light on the predictable trends of model performance when the parameter number increases. On the other hand, other works explore the emergent abilities [68; 67; 48] and their relationships to evaluation metrics and task nature.

The Llama series [60; 61; 2] is well-known for its contributions to open-sourced large language models, and is widely regarded as a strong baseline for foundation model evaluation. Falcon [42] explores data processing of publicly available pre-training corpora. Mistral [27] and Gemma [58] release 7B-scaled models that are trained with more data and incorporated with advanced designs. For the Chinese community, Qwen [5], Baichuan [71], Yi [76], and DeepSeek [7] represent efforts in multilingual foundation model pre-training and open-sourcing. FLM-101B [33] studies methodologies for training large foundation models under limited budgets.

InstructGPT [41] establishes the paradigm of aligning large foundation models with human preferences. Widely used approaches include supervised fine-tuning (SFT) [66; 70] and Reinforcement Learning from Human Feedback (RLHF) [49], among others [44]. Aligning techniques turn foundation models into dialogue agents, which form the core of AI assistants in commercial use. Closed-source dialogue agents are represented by GPT-4 [40], Claude [6], Grok [1], and Gemini [57]. Open-sourced chat models include Zephyr [62] and ChatGLM [25], among the large number of human-aligned versions of the open foundation models mentioned above.

8 Conclusions and Future Work

In this report, we introduce Tele-FLM, an open multilingual foundation model. With 52B parameters and 2T training tokens, Tele-FLM matches the performance of larger models trained with more data, in both multilingual language modeling capabilities and benchmark evaluations. The pre-training procedure of Tele-FLM features a high success rate and low carbon footprint. We open-source the model weights as well as technical details and training dynamics. We hope this work will catalyze the growth of open-sourced LLM communities and reduce the trial-and-error cycles to train LLMs with more than 50B parameters. Note that although efforts are made to filter out harmful contents in the training data, such kind of outputs could still potentially be elicited from the released model, which does not represent the opinions of the authors or entities involved.

For future work, we plan to continue enhancing the capabilities of Tele-FLM to facilitate broader application, as well as to develop efficient training techniques to explore the unmanned deep space of larger-scaled dense models.

Acknowledgments

This work is supported by the National Science and Technology Major Project (No. 2022ZD0116300) and the National Science Foundation of China (No. 62106249). We would like to thank Boya Wu, Li Du, Quanyue Ma, Hanyu Zhao, Shiyu Wu and Kaipeng Jia for their help on data, Hailong Qian, Jinglong Li, Taojia Liu, Junjie Wang, Yuanlin Cai, Jiahao Guo, Quan Zhao, Xuwei Yang, Hanxiao Qu, Yan Tian, and Kailong Xie for their help on computational resources, and all other colleagues' strong support for this project.

References

[1] Grok. `https://x.ai/blog`.

[2] Introducing meta llama 3: The most capable openly available llm to date. https://ai.meta.com/blog/meta-llama-3/.

[3] Winogrande: An adversarial winograd schema challenge at scale. 2019.

[4] Rohan Anil, Andrew M. Dai, Orhan Firat, Melvin Johnson, Dmitry Lepikhin, Alexandre Passos, Siamak Shakeri, Emanuel Taropa, Paige Bailey, Zhifeng Chen, Eric Chu, Jonathan H. Clark, Laurent El Shafey, Yanping Huang, Kathy Meier-Hellstern, Gaurav Mishra, Erica Moreira, Mark Omernick, Kevin Robinson, Sebastian Ruder, Yi Tay, Kefan Xiao, Yuanzhong Xu, Yujing Zhang, Gustavo Hernández Ábrego, Junwhan Ahn, Jacob Austin, Paul Barham, Jan A. Botha, James Bradbury, Siddhartha Brahma, Kevin Brooks, Michele Catasta, Yong Cheng, Colin Cherry, Christopher A. Choquette-Choo, Aakanksha Chowdhery, Clément Crepy, Shachi Dave, Mostafa Dehghani, Sunipa Dev, Jacob Devlin, Mark Díaz, Nan Du, Ethan Dyer, Vladimir Feinberg, Fangxiaoyu Feng, Vlad Fienber, Markus Freitag, Xavier Garcia, Sebastian Gehrmann, Lucas Gonzalez, and et al. Palm 2 technical report. CoRR, abs/2305.10403, 2023.

[5] Jinze Bai, Shuai Bai, Yunfei Chu, Zeyu Cui, Kai Dang, Xiaodong Deng, Yang Fan, Wenbin Ge, Yu Han, Fei Huang, et al. Qwen technical report. arXiv preprint arXiv:2309.16609, 2023.

[6] Yuntao Bai, Andy Jones, Kamal Ndousse, Amanda Askell, Anna Chen, Nova DasSarma, Dawn Drain, Stanislav Fort, Deep Ganguli, Tom Henighan, Nicholas Joseph, Saurav Kadavath, Jackson Kernion, Tom Conerly, Sheer El Showk, Nelson Elhage, Zac Hatfield-Dodds, Danny Hernandez, Tristan Hume, Scott Johnston, Shauna Kravec, Liane Lovitt, Neel Nanda, Catherine Olsson, Dario Amodei, Tom B. Brown, Jack Clark, Sam McCandlish, Chris Olah, Benjamin Mann, and Jared Kaplan. Training a helpful and harmless assistant with reinforcement learning from human feedback. CoRR, abs/2204.05862, 2022.

[7] Xiao Bi, Deli Chen, Guanting Chen, Shanhuang Chen, Damai Dai, Chengqi Deng, Honghui Ding, Kai Dong, Qiushi Du, Zhe Fu, et al. Deepseek llm: Scaling open-source language models with longtermism. arXiv preprint arXiv:2401.02954, 2024.

[8] Andrei Z Broder. On the resemblance and containment of documents. In Proceedings. Compression and Complexity of SEQUENCES 1997 (Cat. No. 97TB100171), pages 21–29. IEEE, 1997.

[9] Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, Xi Chen, Krzysztof Choromanski, Tianli Ding, Danny Driess, Avinava Dubey, Chelsea Finn, et al. Rt-2: Vision-language-action models transfer web knowledge to robotic control. arXiv preprint arXiv:2307.15818, 2023.

[10] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. Advances in neural information processing systems, 33:1877–1901, 2020.

[11] Sébastien Bubeck, Varun Chandrasekaran, Ronen Eldan, Johannes Gehrke, Eric Horvitz, Ece Kamar, Peter Lee, Yin Tat Lee, Yuanzhi Li, Scott M. Lundberg, Harsha Nori, Hamid Palangi, Marco Túlio Ribeiro, and Yi Zhang. Sparks of artificial general intelligence: Early experiments with GPT-4. CoRR, abs/2303.12712, 2023.

[12] Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde de Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, et al. Evaluating large language models trained on code. arXiv preprint arXiv:2107.03374, 2021.

[13] Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. Think you have solved question answering? try arc, the AI2 reasoning challenge. CoRR, abs/1803.05457, 2018.

[14] Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, et al. Training verifiers to solve math word problems. arXiv preprint arXiv:2110.14168, 2021.

[15] Together Computer. Redpajama: an open dataset for training large language models, 2023.

[16] Gregoire Deletang, Anian Ruoss, Paul-Ambroise Duquenne, Elliot Catt, Tim Genewein, Christopher Mattern, Jordi Grau-Moya, Li Kevin Wenliang, Matthew Aitchison, Laurent Orseau, et al. Language modeling is compression. In The Twelfth International Conference on Learning Representations, 2023.

[17] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: pre-training of deep bidirectional transformers for language understanding. In Jill Burstein, Christy Doran, and Thamar Solorio, editors, Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers), pages 4171–4186. Association for Computational Linguistics, 2019.

[18] Leo Gao, Stella Biderman, Sid Black, Laurence Golding, Travis Hoppe, Charles Foster, Jason Phang, Horace He, Anish Thite, Noa Nabeshima, et al. The pile: An 800gb dataset of diverse text for language modeling. arXiv preprint arXiv:2101.00027, 2020.

[19] Leo Gao, Jonathan Tow, Baber Abbasi, Stella Biderman, Sid Black, Anthony DiPofi, Charles Foster, Laurence Golding, Jeffrey Hsu, Alain Le Noac'h, Haonan Li, Kyle McDonell, Niklas Muennighoff, Chris Ociepa, Jason Phang, Laria Reynolds, Hailey Schoelkopf, Aviya Skowron, Lintang Sutawika, Eric Tang, Anish Thite, Ben Wang, Kevin Wang, and Andy Zou. A framework for few-shot language model evaluation, 12 2023.

[20] Dirk Groeneveld, Iz Beltagy, Pete Walsh, Akshita Bhagia, Rodney Kinney, Oyvind Tafjord, Ananya Harsh Jha, Hamish Ivison, Ian Magnusson, Yizhong Wang, et al. Olmo: Accelerating the science of language models. arXiv preprint arXiv:2402.00838, 2024.

[21] Conghui He, Zhenjiang Jin, Chao Xu, Jiantao Qiu, Bin Wang, Wei Li, Hang Yan, Jiaqi Wang, and Dahua Lin. Wanjuan: A comprehensive multimodal dataset for advancing english and chinese large models, 2023.

[22] Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. Measuring massive multitask language understanding. In 9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021. OpenReview.net, 2021.

[23] Tom Henighan, Jared Kaplan, Mor Katz, Mark Chen, Christopher Hesse, Jacob Jackson, Heewoo Jun, Tom B. Brown, Prafulla Dhariwal, Scott Gray, Chris Hallacy, Benjamin Mann, Alec Radford, Aditya Ramesh, Nick Ryder, Daniel M. Ziegler, John Schulman, Dario Amodei, and Sam McCandlish. Scaling laws for autoregressive generative modeling. CoRR, abs/2010.14701, 2020.

[24] Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, Diego de Las Casas, Lisa Anne Hendricks, Johannes Welbl, Aidan Clark, Tom Hennigan, Eric Noland, Katherine Millican, George van den Driessche, Bogdan Damoc, Aurelia Guy, Simon Osindero, Karen Simonyan, Erich Elsen, Oriol Vinyals, Jack W. Rae, and Laurent Sifre. An empirical analysis of compute-optimal large language model training. In NeurIPS, 2022.

[25] Zhenyu Hou, Yiin Niu, Zhengxiao Du, Xiaohan Zhang, Xiao Liu, Aohan Zeng, Qinkai Zheng, Minlie Huang, Hongning Wang, Jie Tang, et al. Chatglm-rlhf: Practices of aligning large language models with human feedback. arXiv preprint arXiv:2404.00934, 2024.

[26] Yuzhen Huang, Yuzhuo Bai, Zhihao Zhu, Junlei Zhang, Jinghan Zhang, Tangjun Su, Junteng Liu, Chuancheng Lv, Yikai Zhang, Jiayi Lei, Yao Fu, Maosong Sun, and Junxian He. C-eval: A multi-level multi-discipline chinese evaluation suite for foundation models. CoRR, abs/2305.08322, 2023.

[27] Albert Q Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, et al. Mistral 7b. arXiv preprint arXiv:2310.06825, 2023.

[28] Armand Joulin, Edouard Grave, Piotr Bojanowski, Matthijs Douze, Hérve Jégou, and Tomas Mikolov. Fasttext.zip: Compressing text classification models. arXiv preprint arXiv:1612.03651, 2016.

[29] Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B. Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. Scaling laws for neural language models. CoRR, abs/2001.08361, 2020.

[30] Dan Kondratyuk, Lijun Yu, Xiuye Gu, José Lezama, Jonathan Huang, Rachel Hornung, Hartwig Adam, Hassan Akbari, Yair Alon, Vighnesh Birodkar, et al. Videopoet: A large language model for zero-shot video generation. arXiv preprint arXiv:2312.14125, 2023.

[31] Vijay Korthikanti, Jared Casper, Sangkug Lym, Lawrence McAfee, Michael Andersch, Mohammad Shoeybi, and Bryan Catanzaro. Reducing activation recomputation in large transformer models, 2022.

[32] Haonan Li, Yixuan Zhang, Fajri Koto, Yifei Yang, Hai Zhao, Yeyun Gong, Nan Duan, and Timothy Baldwin. Cmmlu: Measuring massive multitask language understanding in chinese. arXiv preprint arXiv:2306.09212, 2023.

[33] Xiang Li, Yiqun Yao, Xin Jiang, Xuezhi Fang, Xuying Meng, Siqi Fan, Peng Han, Jing Li, Li Du, Bowen Qin, et al. Flm-101b: An open llm and how to train it with $100 k budget. arXiv preprint arXiv:2309.03852, 2023.

[34] Yudong Li, Yuqing Zhang, Zhe Zhao, Linlin Shen, Weijie Liu, Weiquan Mao, and Hui Zhang. CSL: A large-scale Chinese scientific literature dataset. In Nicoletta Calzolari, Chu-Ren Huang, Hansaem Kim, James Pustejovsky, Leo Wanner, Key-Sun Choi, Pum-Mo Ryu, Hsin-Hsi Chen, Lucia Donatelli, Heng Ji, Sadao Kurohashi, Patrizia Paggio, Nianwen Xue, Seokhwan Kim, Younggyun Hahm, Zhong He, Tony Kyungil Lee, Enrico Santus, Francis Bond, and Seung-Hoon Na, editors, Proceedings of the 29th International Conference on Computational Linguistics, pages 3917–3923, Gyeongju, Republic of Korea, October 2022. International Committee on Computational Linguistics.

[35] Hunter Lightman, Vineet Kosaraju, Yura Burda, Harrison Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. Let's verify step by step. CoRR, abs/2305.20050, 2023.

[36] Etai Littwin and Greg Yang. Adaptive optimization in the ∞-width limit. In The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023. OpenReview.net, 2023.

[37] Ziyang Luo, Can Xu, Pu Zhao, Qingfeng Sun, Xiubo Geng, Wenxiang Hu, Chongyang Tao, Jing Ma, Qingwei Lin, and Daxin Jiang. Wizardcoder: Empowering code large language models with evol-instruct. arXiv preprint arXiv:2306.08568, 2023.

[38] Ian Magnusson, Akshita Bhagia, Valentin Hofmann, Luca Soldaini, Ananya Harsh Jha, Oyvind Tafjord, Dustin Schwenk, Evan Pete Walsh, Yanai Elazar, Kyle Lo, et al. Paloma: A benchmark for evaluating language model fit. arXiv preprint arXiv:2312.10523, 2023.

[39] Deepak Narayanan, Mohammad Shoeybi, Jared Casper, Patrick LeGresley, Mostofa Patwary, Vijay Korthikanti, Dmitri Vainbrand, Prethvi Kashinkunti, Julie Bernauer, Bryan Catanzaro, Amar Phanishayee, and Matei Zaharia. Efficient large-scale language model training on GPU clusters. CoRR, abs/2104.04473, 2021.

[40] OpenAI. GPT-4 technical report. CoRR, abs/2303.08774, 2023.

[41] Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul F. Christiano, Jan Leike, and Ryan Lowe. Training language models to follow instructions with human feedback. In NeurIPS, 2022.

[42] Guilherme Penedo, Quentin Malartic, Daniel Hesslow, Ruxandra Cojocaru, Alessandro Cappelli, Hamza Alobeidli, Baptiste Pannier, Ebtesam Almazrouei, and Julien Launay. The refinedweb dataset for falcon llm: outperforming curated corpora with web data, and web data only. arXiv preprint arXiv:2306.01116, 2023.

[43] Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. 2019.

[44] Rafael Rafailov, Archit Sharma, Eric Mitchell, Stefano Ermon, Christopher D. Manning, and Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model. CoRR, abs/2305.18290, 2023.

[45] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. J. Mach. Learn. Res., 21:140:1–140:67, 2020.

[46] Samyam Rajbhandari, Jeff Rasley, Olatunji Ruwase, and Yuxiong He. Zero: Memory optimization towards training A trillion parameter models. CoRR, abs/1910.02054, 2019.

[47] Teven Le Scao, Angela Fan, Christopher Akiki, Ellie Pavlick, Suzana Ilic, Daniel Hesslow, Roman Castagné, Alexandra Sasha Luccioni, François Yvon, Matthias Gallé, Jonathan Tow, Alexander M. Rush, Stella Biderman, Albert Webson, Pawan Sasanka Ammanamanchi, Thomas Wang, Benoît Sagot, Niklas Muennighoff, Albert Villanova del Moral, Olatunji Ruwase, Rachel Bawden, Stas Bekman, Angelina McMillan-Major, Iz Beltagy, Huu Nguyen, Lucile Saulnier, Samson Tan, Pedro Ortiz Suarez, Victor Sanh, Hugo Laurençon, Yacine Jernite, Julien Launay, Margaret Mitchell, Colin Raffel, Aaron Gokaslan, Adi Simhi, Aitor Soroa, Alham Fikri Aji, Amit Alfassy, Anna Rogers, Ariel Kreisberg Nitzav, Canwen Xu, Chenghao Mou, Chris Emezue, Christopher Klamm, Colin Leong, Daniel van Strien, David Ifeoluwa Adelani, and et al. BLOOM: A 176b-parameter open-access multilingual language model. CoRR, abs/2211.05100, 2022.

[48] Rylan Schaeffer, Brando Miranda, and Sanmi Koyejo. Are emergent abilities of large language models a mirage? Advances in Neural Information Processing Systems, 36, 2024.

[49] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. CoRR, abs/1707.06347, 2017.

[50] Noam Shazeer. GLU variants improve transformer. CoRR, abs/2002.05202, 2020.

[51] Mohammad Shoeybi, Mostofa Patwary, Raul Puri, Patrick LeGresley, Jared Casper, and Bryan Catanzaro. Megatron-lm: Training multi-billion parameter language models using model parallelism. CoRR, abs/1909.08053, 2019.

[52] Aarohi Srivastava, Abhinav Rastogi, Abhishek Rao, Abu Awal Md Shoeb, Abubakar Abid, Adam Fisch, Adam R Brown, Adam Santoro, Aditya Gupta, Adrià Garriga-Alonso, et al. Beyond the imitation game: Quantifying and extrapolating the capabilities of language models. arXiv preprint arXiv:2206.04615, 2022.

[53] Jianlin Su, Yu Lu, Shengfeng Pan, Bo Wen, and Yunfeng Liu. Roformer: Enhanced transformer with rotary position embedding. CoRR, abs/2104.09864, 2021.

[54] Kai Sun, Dian Yu, Dong Yu, and Claire Cardie. Investigating prior knowledge for challenging chinese machine reading comprehension. Transactions of the Association for Computational Linguistics, 2020.

[55] Yutao Sun, Li Dong, Barun Patra, Shuming Ma, Shaohan Huang, Alon Benhaim, Vishrav Chaudhary, Xia Song, and Furu Wei. A length-extrapolatable transformer. In Anna Rogers, Jordan L. Boyd-Graber, and Naoaki Okazaki, editors, Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2023, Toronto, Canada, July 9-14, 2023, pages 14590–14604. Association for Computational Linguistics, 2023.

[56] Mirac Suzgun, Nathan Scales, Nathanael Schärli, Sebastian Gehrmann, Yi Tay, Hyung Won Chung, Aakanksha Chowdhery, Quoc V Le, Ed H Chi, Denny Zhou, et al. Challenging big-bench tasks and whether chain-of-thought can solve them. arXiv preprint arXiv:2210.09261, 2022.

[57] Gemini Team, Rohan Anil, Sebastian Borgeaud, Yonghui Wu, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M Dai, Anja Hauth, et al. Gemini: a family of highly capable multimodal models. arXiv preprint arXiv:2312.11805, 2023.

[58] Gemma Team, Thomas Mesnard, Cassidy Hardin, Robert Dadashi, Surya Bhupatiraju, Shreya Pathak, Laurent Sifre, Morgane Rivière, Mihir Sanjay Kale, Juliette Love, et al. Gemma: Open models based on gemini research and technology. arXiv preprint arXiv:2403.08295, 2024.

[59] Tristan Thrush, Helen Ngo, Nathan Lambert, and Douwe Kiela. Online language modelling data pipeline. `https://github.com/huggingface/olm-datasets`, 2022.

[60] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurélien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. Llama: Open and efficient foundation language models. CoRR, abs/2302.13971, 2023.

[61] Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton-Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurélien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. Llama 2: Open foundation and fine-tuned chat models. CoRR, abs/2307.09288, 2023.

[62] Lewis Tunstall, Edward Beeching, Nathan Lambert, Nazneen Rajani, Kashif Rasul, Younes Belkada, Shengyi Huang, Leandro von Werra, Clémentine Fourrier, Nathan Habib, et al. Zephyr: Direct distillation of lm alignment. arXiv preprint arXiv:2310.16944, 2023.

[63] Leslie G. Valiant. A bridging model for parallel computation. Commun. ACM, 33(8):103–111, aug 1990.

[64] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. Advances in neural information processing systems, 30, 2017.

[65] Changhan Wang, Kyunghyun Cho, and Jiatao Gu. Neural machine translation with byte-level subwords. CoRR, abs/1909.03341, 2019.

[66] Yizhong Wang, Yeganeh Kordi, Swaroop Mishra, Alisa Liu, Noah A. Smith, Daniel Khashabi, and Hannaneh Hajishirzi. Self-instruct: Aligning language models with self-generated instructions. In Anna Rogers, Jordan L. Boyd-Graber, and Naoaki Okazaki, editors, Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2023, Toronto, Canada, July 9-14, 2023, pages 13484–13508. Association for Computational Linguistics, 2023.

[67] Jason Wei, Najoung Kim, Yi Tay, and Quoc V Le. Inverse scaling can become u-shaped. arXiv preprint arXiv:2211.02011, 2022.

[68] Jason Wei, Yi Tay, Rishi Bommasani, Colin Raffel, Barret Zoph, Sebastian Borgeaud, Dani Yogatama, Maarten Bosma, Denny Zhou, Donald Metzler, Ed H. Chi, Tatsunori Hashimoto, Oriol Vinyals, Percy Liang, Jeff Dean, and William Fedus. Emergent abilities of large language models. Trans. Mach. Learn. Res., 2022, 2022.

[69] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed H. Chi, Quoc V. Le, and Denny Zhou. Chain-of-thought prompting elicits reasoning in large language models. In NeurIPS, 2022.

[70] Can Xu, Qingfeng Sun, Kai Zheng, Xiubo Geng, Pu Zhao, Jiazhan Feng, Chongyang Tao, and Daxin Jiang. Wizardlm: Empowering large language models to follow complex instructions. CoRR, abs/2304.12244, 2023.

[71] Aiyuan Yang, Bin Xiao, Bingning Wang, Borong Zhang, Ce Bian, Chao Yin, Chenxu Lv, Da Pan, Dian Wang, Dong Yan, et al. Baichuan 2: Open large-scale language models. arXiv preprint arXiv:2309.10305, 2023.

[72] Greg Yang and Edward J. Hu. Tensor programs IV: feature learning in infinite-width neural networks. In Marina Meila and Tong Zhang, editors, Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event, volume 139 of Proceedings of Machine Learning Research, pages 11727–11737. PMLR, 2021.

[73] Greg Yang, Edward J. Hu, Igor Babuschkin, Szymon Sidor, Xiaodong Liu, David Farhi, Nick Ryder, Jakub Pachocki, Weizhu Chen, and Jianfeng Gao. Tuning large neural networks via zero-shot hyperparameter transfer. In Marc'Aurelio Ranzato, Alina Beygelzimer, Yann N. Dauphin, Percy Liang, and Jennifer Wortman Vaughan, editors, Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual, pages 17084–17097, 2021.

[74] Zhen Yang, Ming Ding, Qingsong Lv, Zhihuan Jiang, Zehai He, Yuyi Guo, Jinfeng Bai, and Jie Tang. Gpt can solve mathematical problems without a calculator. arXiv preprint arXiv:2309.03241, 2023.

[75] Yiqun Yao, Siqi fan, Xiusheng Huang, Xuezhi Fang, Xiang Li, Ziyi Ni, Xin Jiang, Xuying Meng, Peng Han, Shuo Shang, Kang Liu, Aixin Sun, and Yequan Wang. nanolm: an affordable llm pre-training benchmark via accurate loss prediction across scales. CoRR, abs/2304.06875, 2023.

[76] Alex Young, Bei Chen, Chao Li, Chengen Huang, Ge Zhang, Guanwei Zhang, Heng Li, Jiangcheng Zhu, Jianqun Chen, Jing Chang, et al. Yi: Open foundation models by 01. ai. arXiv preprint arXiv:2403.04652, 2024.

[77] Sha Yuan, Hanyu Zhao, Zhengxiao Du, Ming Ding, Xiao Liu, Yukuo Cen, Xu Zou, Zhilin Yang, and Jie Tang. Wudaocorpora: A super large-scale chinese corpora for pre-training language models. AI Open, 2:65–68, 2021.

[78] Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. Hellaswag: Can a machine really finish your sentence? In Anna Korhonen, David R. Traum, and Lluís Màrquez, editors, Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers, pages 4791–4800. Association for Computational Linguistics, 2019.

[79] Aohan Zeng, Xiao Liu, Zhengxiao Du, Zihan Wang, Hanyu Lai, Ming Ding, Zhuoyi Yang, Yifan Xu, Wendi Zheng, Xiao Xia, Weng Lam Tam, Zixuan Ma, Yufei Xue, Jidong Zhai, Wenguang Chen, Zhiyuan Liu, Peng Zhang, Yuxiao Dong, and Jie Tang. GLM-130B: an open bilingual pre-trained model. In The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023. OpenReview.net, 2023.

[80] Biao Zhang and Rico Sennrich. Root mean square layer normalization. CoRR, abs/1910.07467, 2019.

[81] Susan Zhang, Stephen Roller, Naman Goyal, Mikel Artetxe, Moya Chen, Shuohui Chen, Christopher Dewan, Mona T. Diab, Xian Li, Xi Victoria Lin, Todor Mihaylov, Myle Ott, Sam Shleifer, Kurt Shuster, Daniel Simig, Punit Singh Koura, Anjali Sridhar, Tianlu Wang, and Luke Zettlemoyer. OPT: open pre-trained transformer language models. CoRR, abs/2205.01068, 2022.

[82] Chujie Zheng, Minlie Huang, and Aixin Sun. ChID: A large-scale Chinese IDiom dataset for cloze test. In Anna Korhonen, David Traum, and Lluís Màrquez, editors, Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, pages 778–787, Florence, Italy, July 2019. Association for Computational Linguistics.