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

XXX [*]
Tsinghua University

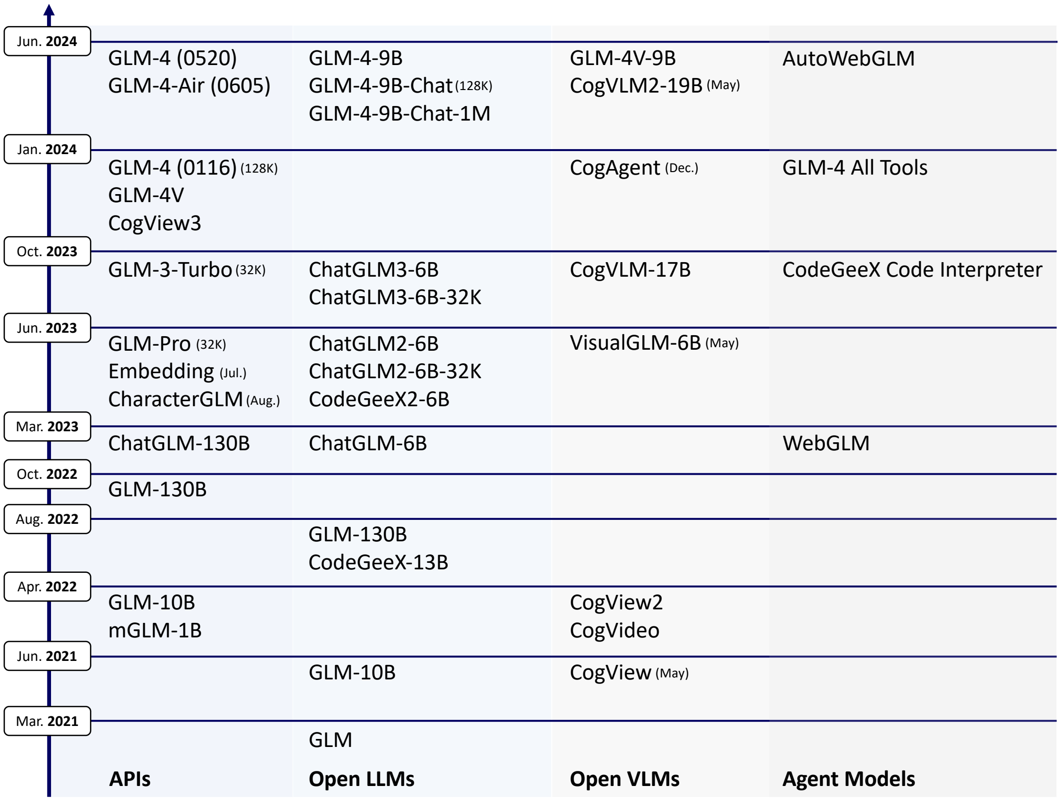
Abstract

We introduce ChatGLM, an evolving family of large language models that we have been developing over time. This report primarily focuses on the GLM-4 language series, which includes GLM-4, GLM-4-Air, and GLM-4-9B. To date, the GLM-4 models are pre-trained on ten trillions of tokens mostly in Chinese and English, along with a small set of corpus from 24 languages, and aligned primarily for Chinese and English usage. The high-quality alignment is achieved via a multi-stage post-training process, which involves supervised fine-tuning and learning from human feedback. The GLM-4 All Tools model is further aligned to understand user intent and autonomously decide when and which tool(s) to use—including web browser, Python interpreter, text-to-image model, and user-defined functions—to effectively complete complex tasks. Over the course, we have open-sourced a series of models, including ChatGLM-6B (three generations), GLM-4-9B (128K, 1M), GLM-4V-9B, WebGLM, and CodeGeeX, attracting over 10 million downloads on Hugging face in the year 2023 alone. The open models can be accessed through `https://github.com/THUDM` and `https://huggingface.co/THUDM`.

---

[*] The original paper is located at `https://arxiv.org/abs/2406.12793`. This version has been modified by LuYF-Lemon-love ⟨ `luyanfeng_nlp@qq.com` ⟩ for personal study.

Preprint. Under review.



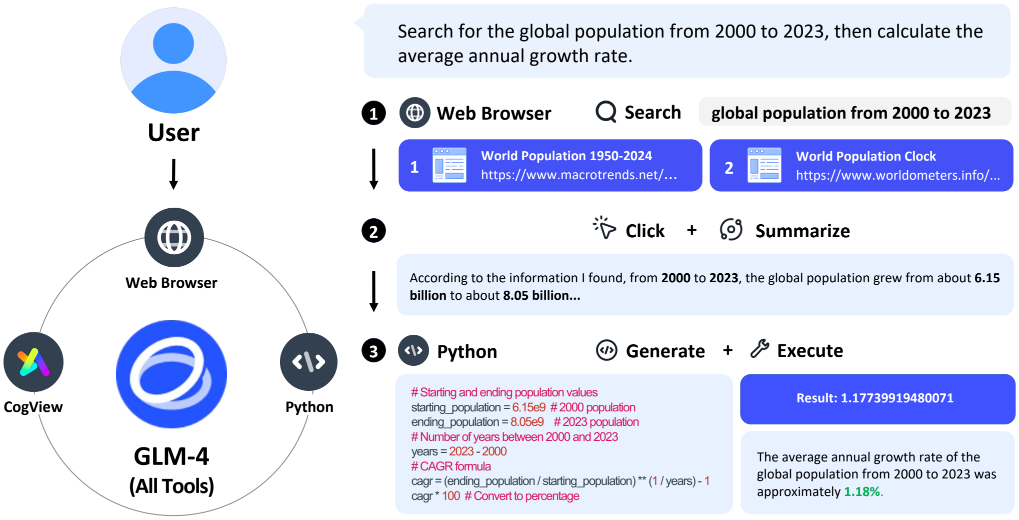
	APIs	Open LLMs	Open VLMs	Agent Models
Jun. 2024	GLM-4 (0520) GLM-4-Air (0605)	GLM-4-9B GLM-4-9B-Chat (128K) GLM-4-9B-Chat-1M	GLM-4V-9B CogVLM2-19B (May)	AutoWebGLM
Jan. 2024	GLM-4 (0116) (128K) GLM-4V CogView3		CogAgent (Dec.)	GLM-4 All Tools
Oct. 2023	GLM-3-Turbo (32K)	ChatGLM3-6B ChatGLM3-6B-32K	CogVLM-17B	CodeGeeX Code Interpreter
Jun. 2023	GLM-Pro (32K) Embedding (Jul.) CharacterGLM (Aug.)	ChatGLM2-6B ChatGLM2-6B-32K CodeGeeX2-6B	VisualGLM-6B (May)	
Mar. 2023	ChatGLM-130B	ChatGLM-6B		WebGLM
Oct. 2022	GLM-130B			
Aug. 2022		GLM-130B CodeGeeX-13B		
Apr. 2022	GLM-10B mGLM-1B		CogView2 CogVideo	
Jun. 2021		GLM-10B	CogView (May)	
Mar. 2021		GLM		

Figure 1: The timeline of the GLM family of language, code, vision, and agent models. The focus of this report is primarily on the language models, i.e., ChatGLM. The APIs are publicly available at `https://bigmodel.cn` and open models can be accessed through `https://github.com/THUDM`.

1 Introduction

ChatGLM-6B was pre-trained on approximately one trillion tokens of Chinese and English corpus with a context length of 2,048 (2K), supplemented mostly by SFT. Released in June, ChatGLM2-6B was pre-trained and aligned with more and better data, leading to substantial improvements over its predecessor, including a 23% improvement on MMLU, 571% on GSM8K, and 60% on BBH. By adopting the FlashAttention technique [6], its context length was extended to 32K. Additionally, the integration of Multi-Query Attention [27] contributed to a 42% increase in inference speed. Taking this further, our 2nd generation code model CodeGeeX2-6B was developed by pre-training on an additional 600 billion code tokens. It demonstrates Pass@1 improvements over the initial generation, CodeGeeX-13B [49], with increases of 57% in Python, 71% in C++, 54% in Java, 83% in JavaScript, and 56% in Go as measured by HumanEval-X. By further realizing more diverse training datasets, more sufficient training steps, and more optimized training strategies, ChatGLM3-6B topped 42 benchmarks across semantics, mathematics, reasoning, code, and knowledge. Starting from this generation, ChatGLM also supports function call and code interpreter, as well as complex agent tasks [7; 29; 43]. In the course of these developments, we also developed models with 1.5B, 3B, 12B, 32B, 66B, and 130B parameters, allowing us to validate our observations and establish our own scaling laws.

With all the lessons learned and experiences accumulated, we kicked off the training of GLM-4. The first cutoff checkpoint then underwent a multi-stage post-training process (e.g., SFT, RLHF, safety alignment) with a focus on the Chinese and English language for now. Subsequently, it was developed into two distinct versions: GLM-4 and GLM-4 All Tools, both supporting a 128K context length. Since Jan. 16, 2024, GLM-4 (0116) has been made available through the GLM-4 API at `https://bigmodel.cn`, and GLM-4 All Tools is accessible via the website `https://chatglm.cn` and mobile applications that support the creation of one's own agent—GLMs. The



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

global population from 2000 to 2023

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

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

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

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

Result: 1.17739919480071

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

Figure 2: An Illustrative Example of GLM-4 All Tools.

latest models are GLM-4 (0520) and GLM-4-Air (0605) with an upgrade on both pre-training and alignment. GLM-4-Air achieves comparable performance to GLM-4 (0116) with lower latency and inference cost. Evaluations of GLM-4 were performed on a variety of language benchmarks. These evaluations assess GLM-4's general abilities in English, instruction following in both English and Chinese, and alignment, long-context, and agent capacities in Chinese.

First, on the most commonly-used English academic benchmarks—MMLU, GSM8K, MATH, BBH, GPQA, and HumanEval, GLM-4 0520 achieves performance closely comparable to that of GPT-4 0613 [21] and Gemini 1.5 Pro [33]. For example, it scores 83.3 vs. 86.4 and 83.7 on MMLU, respectively. Second, according to IFEval [52], GLM-4's instruction following capacities on both prompt and instruction levels are approximately as effective as GPT-4-Turbo in both English and Chinese. Third, in terms of Chinese language alignment, GLM-4 outperforms GPT-4 and matches GPT-4-Turbo across eight dimensions in AlignBench [16]. Finally, for long-context tasks, the GLM-4 (128K) model matches the performance level of GPT-4 Turbo and Claude 3 Opus as measured by LongBench-Chat [1], i.e., 87.3 vs. 87.2 and 87.7, respectively.

The GLM-4 All Tools model is specifically aligned to better understand user intent and autonomously select the most appropriate tool(s) for task completion. For example, it can access online information via a web browser in a multi-round manner, use the Python interpreter to solve math problems, leverage a text-to-image model to generate images, and call user-defined functions. Figure 2 shows an illustrative example of GLM-4 All Tools with a web browser and Python Interpreter for addressing the user query of "Search for the global population from 2000 to 2023, then calculate the average annual growth rate". Our first-hand test shows that it not only matches but often surpasses the capabilities of GPT-4 All Tools for common tasks.

Following our three generations of open ChatGLM-6B models, we also openly released the GLM-4-9B (128K and 1M context length) model. GLM-4-9B is pre-trained on approximately ten trillion tokens of multilingual corpus with a context length of 8192 (8K) and post-trained with the same pipeline and data used for GLM-4 (0520). With less training compute, it outperforms Llama-3-8B [19] and supports all the functionality of All Tools in GLM-4. We also provide an experimental model GLM-4-9B-Chat-1M with 1 million (1M) context length (about 2 million Chinese characters). Table 1 shows the performance of the three generations of ChatGLM-6B models and GLM-4-9B, illustrating the progressive improvements of ChatGLM over time.

Figure 3 summarizes the major improvements and features from GLM-130B to GLM-4 All Tools. Throughout this journey, we have also contributed to the open development of the code LLMs (CodeGeeX [49]) as well as visual language models for image understanding (CogVLM [37] and CogAgent [12]) and text-to-image generation (CogView [8; 50]). The open models and data can be accessed via `https://github.com/THUDM` and `https://huggingface.co/THUDM`.

Table 1: Performance of Open ChatGLM-6B, ChatGLM2-6B, ChatGLM3-6B, and GLM-4-9B.

Language	Dataset	ChatGLM-6B (2023-03-14)	ChatGLM2-6B (2023-06-25)	ChatGLM3-6B-Base (2023-10-27)	GLM-4-9B (2024-06-05)
English	GSM8K	1.5	25.9	72.3	84.0
	MATH	3.1	6.9	25.7	30.4
	BBH	0.0	29.2	66.1	76.3
	MMLU	25.2	45.2	61.4	74.7
	GPQA	-	-	26.8	34.3
	HumanEval	0.0	9.8	58.5	70.1
	BoolQ	51.8	79.0	87.9	89.6
	CommonSenseQA	20.5	65.4	86.5	90.7
	HellaSwag	30.4	57.0	79.7	82.6
	PIQA	65.7	69.6	80.1	79.1
	DROP	3.9	25.6	70.9	77.2
Chinese	C-Eval	23.7	51.7	69.0	77.1
	CMMLU	25.3	50.0	67.5	75.1
	GAOKAO-Bench	26.8	46.4	67.3	74.5
	C3	35.1	58.6	73.9	77.2

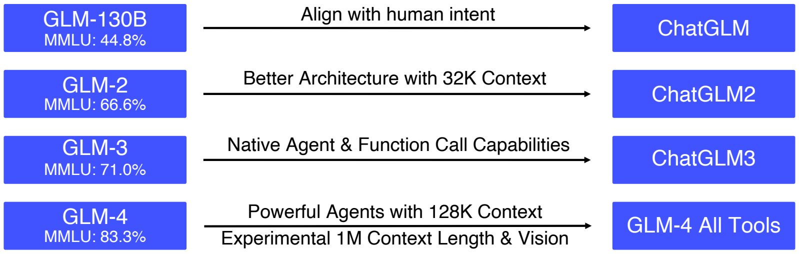

GLM-130B
MMLU: 44.8%
Align with human intent
ChatGLM
GLM-2
MMLU: 66.6%
Better Architecture with 32K Context
ChatGLM2
GLM-3
MMLU: 71.0%
Native Agent & Function Call Capabilities
ChatGLM3
GLM-4
MMLU: 83.3%
Powerful Agents with 128K Context
Experimental 1M Context Length & Vision
GLM-4 All Tools

Figure 3: From GLM-130B to ChatGLM to ChatGLM2/3 to GLM-4 All Tools.

2 ChatGLM Techniques

In this section, we cover both the pre-training and post-training techniques we adopted and developed in ChatGLM, including model architecture, pre-training data, alignment, and All Tools. We have detailed technical reports introducing each of the major techniques we used to reach GLM-4.

Pre-Training Data. Our pre-training corpus consists of multilingual (mostly English and Chinese) documents from a mixture of different sources, including webpages, Wikipedia, books, code, and papers. The data processing pipeline mainly includes three stages: deduplication, filtering, and tokenization. The deduplication stage improves data diversity by removing duplicated or similar documents, with both exact deduplication and fuzzy deduplication. The filtering stage improves data quality by removing noisy documents that contain offensive language, placeholder text, source code, etc. The tokenization stage converts the text into a sequence of tokens for further processing. The number of tokens in the pre-training data directly affects model training speed. To optimize this aspect, we employ the byte-level byte pair encoding (BPE) algorithm [26] to separately learn the Chinese and multilingual tokens merge them with the tokens of the cl100k_base tokenizer in tiktoken [20] into a unified vocabulary with a size of 150,000. In the final training set, we reweight different sources to increase the ratios of high-quality and educational sources like books and Wikipedia. To this end, the pre-training corpus consists of around ten trillion tokens.

Throughout the four generations of ChatGLM development, our findings align with existing studies [51]: data quality and diversity are crucial for building effective LLMs. Despite the empirical lessons and insights gained, we have to date yet to identify a fundamental principle that could guide the processes of data collection, cleaning, and selection.

Architecture. The GLM family of LLMs is built on Transformer [35]. In GLM-130B [46], we explored various options to stabilize its pre-training by taking into account the hardware constraints we faced at the time. Specifically, GLM-130B leveraged DeepNorm [36] as the layer normalization strategy and used Rotary Positional Encoding (RoPE) [31] as well as the Gated Linear Unit [28] with GeLU [11] activation function in FFNs. Throughout our exploration, we have investigated different strategies to enhance model performance and inference efficiency. The recent GLM-4 model adopts the following architecture design choices.

- **No Bias Except QKV**: To increase training speed, we have removed all bias terms with the exception of the biases in Query, Key, and Value (QKV) of the attention layers. In doing so, we observed a slight improvement in length extrapolation.
- **RMSNorm and SwiGLU**: We have adopted RMSNorm and SwiGLU to replace LayerNorm and ReLU, respectively. These two strategies have been observed with better model performance.
- **Rotary positional embeddings (RoPE)**: We have extended the RoPE to a two-dimensional form to accommodate the 2D positional encoding in GLM.
- **Group Query Attention (GQA)**: We have replaced Multi-Head Attention (MHA) with Group Query Attention (GQA) to cut down on the KV cache size during inference. Given GQA uses fewer parameters than MHA, we increased the FFN parameter count to maintain the same model size, i.e., setting d_{ffn} to 10/3 of the hidden size.

The context length of our models was extended from 2K (ChatGLM), to 32K (ChatGLM2 and ChatGLM3), and to 128K and 1M (GLM-4). This expansion was achieved not only through context extension—position encoding extension [24; 4] and continual training [39] on long text—but also long context alignment, enabling GLM-4 to effectively handle long contexts (Cf [1] for technical details).

Alignment. Pre-training builds the foundation of LLMs while post-training [22] further refines these models to align with human preferences, such as understanding human intents, following instructions, and facilitating multi-turn dialogues. For GLM-4, the alignment is mostly achieved with supervised fine-tuning (SFT) and reinforcement learning from human feedback (RLHF) [13]. In SFT, we find that authentic human prompts and interactions instead of template-based or model-generated responses are vital to the alignment quality. While SFT largely aligns the base models with human preferences, RLHF can further help mitigate issues of response rejection, safety, mixture of bilingual tokens generated, and multi-turn coherence among others.

For the first generation of models (ChatGLM-6B and ChatGLM-130B), the prompt-response pairs were mostly annotated by the model developers. For later models, the alignment data is a combination of in-house annotating data and proprietary data acquired from third parties, subject to relatively strict quality control measures. Similar to existing practice [34], annotators are instructed to score model responses from several dimensions, including safety, factuality, relevance, helpfulness, and human preferences.

ChatGLM Techniques. Throughout the development of ChatGLM, we have introduced and will publish techniques that are used to enhance its performance.

- **Emergent Abilities of LLMs [9]**: We examined the relationship between pre-training loss and performance on downstream tasks and found that with the same pre-training loss, LLMs of different model sizes and training tokens generate the same downtream performance. We also find that on some tasks (such as MMLU and GSM8K), the performance improves beyond random chance only when the pre-training loss falls below a certain threshold. We thus redefine emergent abilities as those exhibited by models with lower pre-training losses [9].
- **LongAlign [1]**: To extend LLMs' context window size, we proposed LongAlign—a comprehensive recipe for long context alignment. It enables GLM-4 to process long context texts (up to 128K tokens) with performance comparable to that of Claude 2 and GPT-4 Turbo (1106).

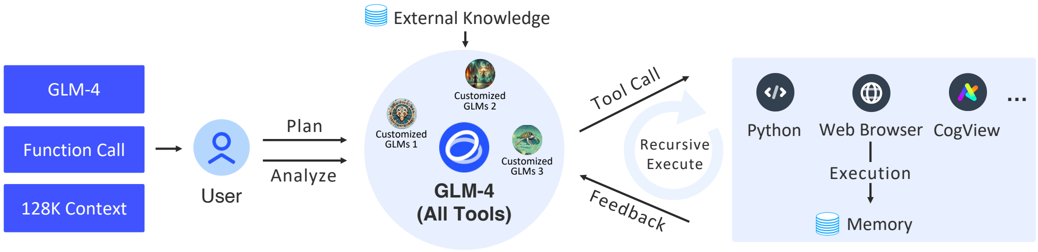


Figure 4: The overall pipeline of GLM-4 All Tools and customized GLMs (agents).

- **ChatGLM-Math [40]**: For the improvement of math problem solving in LLMs, we introduced ChatGLM-Math that leverages self-critique rather than external models or manual annotations for data selection.
- **ChatGLM-RLHF [13]**: To align LLMs with human feedback, we introduced ChatGLM-RLHF—our practices of applying PPO and DPO into LLMs.
- **Self-Contrast [17]**: To avoid the need for expensive human preference feedback data, we developed a feedback-free alignment strategy Self-Contrast. It utilizes the target LLM itself to self-generate massive negative samples for its RLHF alignment.
- **AgentTuning [45]**: To improve LLMs' agent capabilities, we developed the AgentTurning framework with the AgentInstruct instruction-tuning dataset that includes high-quality interaction trajectories between agents and environment.
- **APAR [15]**: To improve the inference speed of LLMs for responses with hierarchical structures, we presented an auto-parallel auto-regressive (APAR) generation approach. It leverages instruct tuning to train LLMs to plan their (parallel) generation process and execute APAR generation.
- **Benchmarks**: We also developed several open LLM benchmarks, including AgentBench [18] for evaluating LLMs as agents, LongBench [2] for evaluating the long context handling performance of LLMs, AlignBench [1] to measure the alignment quality of ChatGLM with Chinese language content, HumanEval-X [49] to evaluate HumanEval [3] problems in programming languages beyond Python, as well as NaturalCodeBench (NCB) to measure models' capacities to solve practical programming tasks.

GLM-4 All Tools. The latest ChatGLM models are GLM-4 and GLM-4 All Tools, both of which were trained and aligned by using the techniques above. GLM-4 All Tools is a model version further aligned to support intelligent agents and related tasks. It can autonomously understand user intent, plan complex instructions, and call one or multiple tools (e.g., Web browser, Python interpreter, and the text-to-image model) to complete complex tasks. Figure 4 presents the overall pipeline of the GLM-4 All Tools system. When a user issues a complex request, the model analyzes the task and plan the solving process step by step. If it determines that it cannot complete the task independently, it will sequentially call one or multiple external tools, utilizing their intermediate feedback and result to help solve the task.

Built on the GLM-4's all-tools capabilities, we also developed the GLMs application platform that allows users to create and customize their own agents for specific tasks. The GLMs support not only the embedded Python interpreter, Web browser, text-to-image model but also user-defined functions, APIs, and external knowledge bases to more effectively address user needs.

3 GLM-4 Capabilities

We examine the capabilities of the GLM-4 model from diverse perspectives, including the base capacity on academic benchmarks, code problem-solving, agent abilities in English, and instruction following, long context for both Chinese and English, as well as alignment in Chinese. As mentioned, GLM-4 was pre-trained mostly in Chinese and English and aligned predominantly to Chinese. In this section, we report results primarily for the latest GLM-4 version, i.e., GLM-4 (0520) and GLM-4-Air (0605), as GLM-4 (0520) is slightly better than its original 0116 version across the evaluated benchmarks. During evaluation, both GLM-4 and GLM-4-Air are deployed with BFloat16 precision.

For baselines, we present results for GPT-4 (0603), GPT-4 Turbo (1106, 2024-04-09), Claude 2, Claude 3 Opus, and Gemini 1.5 Pro, all of which were extracted from the corresponding technical reports or tested through their public APIs.

Overall, GLM-4 gets close to the state-of-the-art models (GPT-4-Turbo, Gemini 1.5 Pro, and Claude 3 Opus) in terms of standard benchmarks, as well as instruction following, long context, code problem-solving, and agent abilities in English environment. For Chinese alignment, it generates strong performance against SOTA models across various domains, such as fundamental language ability, advanced Chinese understanding, professional knowledge, and open-ended questions. In summary, GLM-4 is among the best in terms of Chinese language tasks. It also demonstrates comparable performance to GPT-4 and Claude 3 Opus in Chinese math and logic reasoning capabilities though it lags behind GPT-4 Turbo.

3.1 Evaluation of Academic Benchmarks

To evaluate the general performance of the base model, we select six commonly-used benchmarks spanning knowledge, math, reasoning, commonsense, and coding:

- MMLU [10]: Multi-choice questions collected from various examinations including mathematics, history, computer science, and more. We present all answers to the model and ask it to choose the letter of the answer.
- GSM8K [5]: 8,500 grade school math word problems (1,000 in the test set) that require the model to solve real-life situational problems using mathematical concepts. We use chain-of-thought prompting [38] for this benchmark.
- MATH: 12,500 challenging competition-level mathematics problems (5,000 in the test set). We use chain-of-thought prompting [38] for this benchmark.
- BBH [32]: A suite of 23 challenging BIG-Bench [30] tasks. We use chain-of-thought prompting [38] for this benchmark.
- GPQA [25]: A graduate-level multi-choice benchmark in biology, chemistry, and physics.
- HumanEval [3]: a coding benchmark that measures correctness of synthetic functions with automatic test-case checking.

We compare the performance of GLM-4 with the original GPT-4 [21]. The results are shown in Table 2. We can observe that GLM-4 achieves 96.3% of GPT-4's accuracy on MMLU, and outperforms GPT-4 on other benchmarks. Overall, the base capacity of GLM-4 approaches that of GPT-4-Turbo and Claude 3 Opus.

Table 2: GLM-4 performance on academic benchmarks.

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

3.2 Evaluation of Instruction Following

We assess the proficiency of GLM-4 in following instructions with the recently-introduced IFEval dataset [52]. The dataset comprises 541 prompts derived from 25 distinct instructions that are verifiable through explicit criteria (e.g., "end your email with: P.S. I do like the cake" can be verified via string matching). We adhere to the methodologies outlined by [52] to calculate prompt-level and instruction-level accuracy in both strict mode and loose mode. To further evaluate the models performance on following instructions in Chinese, we translate the original prompts into Chinese, omitted instructions that are not applicable in Chinese (such as capitalization), and adjust the scoring scripts to accommodate Chinese data.

Table 3: GLM-4 performance on IFEval [52], an LLM instruction following benchmark. 'L' stands for 'Loose' and 'S' stands for 'Strict'. 'P' stands for 'Prompt' and 'I' stands for 'Instruction'.

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

The English and Chinese sections in Table 3 show results in both English and Chinese, respectively. In loose mode, GLM-4 matches instruction-level accuracy achieved by GPT-4 Turbo in both English and Chinese. In strict mode, GLM-4 achieves 99.0% and 98.6% of instruction-level accuracy of GPT-4 Turbo (2024-04-09) in English and Chinese, respectively.

3.3 Evaluation of Alignment

AlignBench [16] provides an automatic LLMs-as-Judge method to benchmark the alignment of LLMs in Chinese context. It consists 683 queries spanning 8 different categories, and judges model responses using a GPT-4 based multidimensional rule-calibrated pointwise reference-based scoring method. We evaluate on AlignBench-v1.1, which more carefully improves the reference generation quality, especially by complementing human-collected evidences from webpages with urls for knowledge-requiring problems that takes up 66.5% of total queries. On this version, almost all LLMs achieve lower scores than they do in the previous AlignBench more or less.

Results are shown in Table 4. GLM-4 outperforms GPT-4 Turbo, Claude 3 Opus, and Gemini 1.5 Pro in general, achieves the highest overall score among the baselines. Especially on Chinese Logic Reasoning and Language Understanding dimensions, GLM-4 significantly outperforms all other powerful models. These results demonstrate its strong grasping of Chinese language and knowledge.

The current performance gap between GLM-4 and GPT-4 Turbo (2024-04-09) mostly lies in the Mathematics dimension. We have been employing techniques introduced in ChatGLM-Math [40] such as self-critique to continuously enhance GLM models' reasoning capabilities.

3.4 Evaluation of Long Context Handling Abilities

To obtain the performance of GLM-4 on long text tasks, we carry out evaluations on LongBench-Chat [1], a benchmark set with contextual lengths ranging from 10-100k, encompassing a wide range of long text scenarios frequently utilized by users, such as document Q&A, summarizing, and coding. In our quest to provide a more detailed comparison of the performance of GLM-4

Table 4: GLM-4 performance on AlignBench [16], an LLM benchmark for alignment in Chinese.

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

in different languages, we also segregate LongBench-Chat according to language. This yields two distinct portions: Chinese and English. We have accordingly supplied the results for both segments separately, offering a fine-grained overview of GLM-4's cross-linguistic capabilities.

Regarding the specific evaluation settings, we score the outputs of each model based on GPT-4, adopting a few-shot strategy within LongBench-Chat. Moreover, given our objective to minimize score variations and to reach a more reliable statistical conclusion, we conduct repeated evaluations. Subsequently, we compute the average from these multiple evaluations to ensure that the final performance metric reflects a thorough understanding of how GLM-4 behaves under diverse conditions.

Table 5: GLM-4 performance on LongBench-Chat [2].

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

Table 5 features the results obtained from our experiments. It can be clearly observed that the performance of GLM-4 aligns with that of GPT-4 Turbo and Claude 3 Opus on English prompts, and it can outperform the best of them on Chinese prompts.

3.5 Evaluation of Coding on Real-world User Prompts

While HumanEval [3] has been widely adopted for evaluating code generation, most of its problems can be categorized to introductory algorithms. However, in practice, real users ask complicated questions for production purposes, which are usually far beyond the scope of HumanEval. Additionally, previous works have reported HumanEval-contaminated training data [21; 14; 42] in their own or other LLMs, making the results on HumanEval relatively less trustful than before.

As a result, beside HumanEval we evaluate GLM-4 on NaturalCodeBench (NCB) [47], a challenging bilingual coding benchmark derived from natural user prompts to mirror the complexity of real-world coding missions. Results are shown in Table 6. It shows that GLM-4 has a close coding performance to Claude 3 Opus in practical scenarios. While there is still some gaps to GPT-4 models, considering GLM-4 bilingually balanced nature, there is quite much potential to improve its performance on NCB via better training strategies and data curation in our following iterations.

Table 6: GLM-4 performance on NaturalCodeBench (NCB) [47], a benchmark with real coding prompts in two programming languages (Python and Java) for English and Chinese.

Model	Python (en)	Java (en)	Python (zh)	Java (zh)	Overall
GPT-4 (0613)	55.7	51.1	53.4	51.1	52.8
GPT-4 Turbo (1106)	51.9	55.0	47.3	51.9	51.5
GPT-4 Turbo (2024-04-09)	57.5	52.3	53.1	52.3	53.8
Claude 2	34.4	36.6	33.6	32.8	34.4
Claude 3 Opus	48.9	48.9	45.0	50.4	48.3
Gemini 1.5 Pro	45.0	39.7	41.5	43.1	42.3
GLM-4-9B-Chat	33.9	29.8	30.8	34.4	32.2
GLM-4-Air (0605)	40.8	39.7	43.1	39.7	40.8
GLM-4 (0520)	51.6	42.8	45.4	48.9	47.1

3.6 Evaluation of Function Call

To evaluate the performance of GLM models on function call, we carry out evaluations on Berkeley Function Call Leaderboard [41], a benchmark with 2k question-function-answer pairs. The benchmark evaluates model capacity on calling functions in three categories: evaluation by Abstract Syntax Tree (AST), evaluation by executing APIs, and relevance detection. The first category compares the model output functions against function documents and possible answers with AST analysis. The second category checks for response correctness by executing the generated function calls. Relevance detection evaluates the model's capacity on recognizing functions that are not suitable to answer the user's question. The results are shown in Table 7. We can observe that the function-call capacity of GLM-4 (0520) aligns with that of GPT-4 Turbo (2024-04-09), while GLM-4-9B-Chat significantly outperforms Llama-3-8B-Instruct. Another observation is that the overall accuracy does not improve with model sizes, while GLM-4-9B-Chat can even outperform GLM-4-Air. On the other hand, we observe that the performance on execution summary, which evaluates the execution results of real-world APIs, improves smoothly with model sizes.

Table 7: GLM performance on the Berkeley Function Call Leaderboard.

Model	AST Summary	Exec Summary	Relevance	Overall
Llama-3-8B-Instruct	59.25	70.01	45.83	58.88
GPT-4 Turbo (2024-04-09)	82.14	78.61	88.75	81.24
GPT-4o (2024-05-13)	85.23	80.37	81.25	82.94
ChatGLM3-6B	62.18	69.78	5.42	57.88
GLM-4-9B-Chat	80.26	84.40	87.92	81.00
GLM-4-Air (0605)	84.34	85.93	68.33	80.94
GLM-4 (0520)	82.59	87.78	84.17	81.76

3.7 Evaluation of Agent Abilities

It is widely observed that LLMs are capable to serve as intelligent agents in versatile environments and contexts [23; 44], known as LLMs-as-Agents [18]. As a result, we evaluate GLM-4 together with other comparison LLMs on AgentBench [18], a comprehensive agentic benchmark for text-based LLMs across an array of practical environments, including code-based, game-based, and web-based contexts. Specifically, we evaluate on 7 out of 8 AgentBench environments except for Digital Card Game, which takes much longer time to interact with. Overall scores are calculated using the original per-dataset weights provided in AgentBench [18].

Table 8: GLM-4 performance on AgentBench [18].

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

The results are presented in Table 8. As it shows, GLM-4 models present quite impressive performance on agent tasks, with the GLM-4-Air's comparable and GLM-4's outperforming scores to GPT-4 Turbo and Claude 3 Opus. In terms of specific environments, we find GLM-4 series performing especially well on Database, House-Holding, and Web Shopping tasks, while still demonstrating a gap to GPT-4 series on Operating System, Knowledge Graph, and Lateral Thinking Puzzles. The gap indicates that there is still some room for GLM-4 to improve its performance on code-related agentic tasks and highly interactive language tasks.

3.8 Evaluation of All Tools

GLM-4 is further aligned to support intelligent agents and user self-configured GLMs functionalities on `https://chatglm.cn`, and the resultant model is GLM-4 All Tools. As mentioned, GLM-4 All Tools can complete complex tasks by autonomously understanding user intent, planing step-by-step instructions, and calling multiple tools, including web browser, Python interpreter, and the text-to-image model (e.g., CogView3 [50]. Table 9 shows that GLM-4 All Tools (Web) can generate similar performance on Python Interpreter for solving math problems, browser for information seeking, compared to ChatGPT-4 (Web), respectively.

Table 9: Performance of GLM-4 All Tools.

		GLM-4 All Tools (Web, 0116)	GPT-4 (Web, 0110)
Python Interpreter	GSM8K	91.59	92.72
	MATH	63.60	65.00
	Math23K	88.50	88.40
Browser	Information Seeking	78.08	67.12

4 Safety and Risks

We are committed to ensuring that GLM-4 operates as a safe, responsible, and unbiased model. In addition to addressing common ethical and fairness concerns, we carefully assess and mitigate potential harms that the model may pose to users in real-world scenarios.

Risk Mitigation. We carefully clean data in the pre-training stage by removing text containing sensitive keywords and web pages from a pre-defined blacklist. In the alignment phase, we evaluate each training sample for safety and remove any that pose potential risks. Harmlessness is also an important criteria for preference alignment when comparing multiple model outputs.

Table 10: GLM-4 performance on SafetyBench [48], compared to GPT-4 models and Claude 3 Opus.

	Ethics & Morality	Illegal Activities	Mental Health	Offens-iveness	Physical Health	Privacy & Property	Unfairness & Bias	Overall
GPT-4 (0613)	92.7	93.3	93.0	87.7	96.7	91.3	73.3	89.7
GPT-4 Turbo (1106)	91.0	92.0	93.0	86.0	92.0	88.7	74.3	88.1
GPT-4 Turbo (2024-04-09)	90.3	91.3	91.7	85.3	92.0	89.3	75.0	87.9
Claude 3 Opus	92.7	91.7	92.7	86.3	94.7	88.7	66.0	87.5
GLM-4 (0520)	92.3	91.3	93.3	86.3	92.3	88.6	66.0	87.2

We have a red team that constantly challenges the model with tricky questions that tend to cause unsafe answers. We collect all harmful question-answer pairs from GLM-4 and improve them with human annotations for further model alignment.

Safety Evaluation. We evaluate the GLM-4 model on the SafetyBench [48] dataset, which assesses the capability of each model from 7 dimensions: Ethics and Morality (unethical behaviors), Illegal Activities (basic knowledge of law), Mental Health (adverse impacts on mental health), Offensiveness (offensive behaviors), Physical Health (dangerous behaviors that can cause physical harms), Privacy and Property (privacy breach or property loss), Unfairness and Bias. We evaluate different models on the Chinese subset of SafetyBench, which is created by removing highly sensitive questions that tend to be blocked, to mitigate interference from different API safety policies.

Table 10 shows the safety results of GLM-4 and SOTA models. On most dimensions GLM-4 (0520) shows competitive safety performance, and overall it achieves comparable performance with Claude 3 Opus. GLM-4 slightly falls behind the GPT-4 family of models, especially on the Physical Health dimension, which demands robust common sense knowledge about the physical world to avoid potential risks. More efforts have been put into this direction to develop a more capable and safe GLM model.

5 Conclusion

In this report, we introduce the ChatGLM family of large language models from GLM-130B to GLM-4 (All Tools). Over the past one and half years, we have made great progress in understanding various perspectives of large language models from our first-hand experiences. With the development of each model generation, the team has learned and applied more effective and more efficient strategies for both model pre-training and alignment. The recent ChatGLM models—GLM-4 (0116, 0520), GLM-4-Air (0605), and GLM-4 All Tools—demonstrate significant advancements in understanding and executing complex tasks by autonomously employing external tools and functions. These GLM-4 models have achieved performance on par with, and in some cases superior to, state-of-the-art models such as GPT-4 Turbo, Claude 3 Opus, and Gemini 1.5 Pro, particularly in handling tasks relevant to the Chinese language. In addition, we are committed to promoting accessibility and safety of LLMs through open releasing our model weights and techniques developed throughout this journey. Our open models, including language, code, and vision models, have attracted over 10 million downloads on Hugging Face in the year 2023 alone. Currently, we are working on more capable models with everything we have learned to date. In the future, we will continue democratizing cutting-edge LLM technologies through open sourcing, and push the boundary of model capabilities towards the mission of teaching machines to think like humans.

Acknowledgement. We would like to thank all the data annotators, infra operating staffs, collaborators, and partners as well as everyone at Zhipu AI and Tsinghua University not explicitly mentioned in the report who have provided support, feedback, and contributed to ChatGLM. We would also like to thank Yuxuan Zhang and Wei Jia from Zhipu AI as well as the teams at Hugging Face, Mod-

elScope, WiseModel, and others for their help on the open-sourcing efforts of the GLM family of models.

References

[1] Y. Bai, X. Lv, J. Zhang, Y. He, J. Qi, L. Hou, J. Tang, Y. Dong, and J. Li. Longalign: A recipe for long context alignment of large language models, 2024.

[2] Y. Bai, X. Lv, J. Zhang, H. Lyu, J. Tang, Z. Huang, Z. Du, X. Liu, A. Zeng, L. Hou, Y. Dong, J. Tang, and J. Li. Longbench: A bilingual, multitask benchmark for long context understanding, 2023.

[3] M. Chen, J. Tworek, H. Jun, Q. Yuan, H. P. de Oliveira Pinto, J. Kaplan, H. Edwards, Y. Burda, N. Joseph, G. Brockman, A. Ray, R. Puri, G. Krueger, M. Petrov, H. Khlaaf, G. Sastry, P. Mishkin, B. Chan, S. Gray, N. Ryder, M. Pavlov, A. Power, L. Kaiser, M. Bavarian, C. Winter, P. Tillet, F. P. Such, D. Cummings, M. Plappert, F. Chantzis, E. Barnes, A. Herbert-Voss, W. H. Guss, A. Nichol, A. Paino, N. Tezak, J. Tang, I. Babuschkin, S. Balaji, S. Jain, W. Saunders, C. Hesse, A. N. Carr, J. Leike, J. Achiam, V. Misra, E. Morikawa, A. Radford, M. Knight, M. Brundage, M. Murati, K. Mayer, P. Welinder, B. McGrew, D. Amodei, S. McCandlish, I. Sutskever, and W. Zaremba. Evaluating large language models trained on code. CoRR, abs/2107.03374, 2021.

[4] S. Chen, S. Wong, L. Chen, and Y. Tian. Extending context window of large language models via positional interpolation. arXiv preprint arXiv:2306.15595, 2023.

[5] K. Cobbe, V. Kosaraju, M. Bavarian, M. Chen, H. Jun, L. Kaiser, M. Plappert, J. Tworek, J. Hilton, R. Nakano, C. Hesse, and J. Schulman. Training verifiers to solve math word problems. CoRR, abs/2110.14168, 2021.

[6] T. Dao, D. Fu, S. Ermon, A. Rudra, and C. Ré. Flashattention: Fast and memory-efficient exact attention with io-awareness. Advances in Neural Information Processing Systems, 35:16344–16359, 2022.

[7] X. Deng, Y. Gu, B. Zheng, S. Chen, S. Stevens, B. Wang, H. Sun, and Y. Su. Mind2web: Towards a generalist agent for the web. arXiv preprint arXiv:2306.06070, 2023.

[8] M. Ding, Z. Yang, W. Hong, W. Zheng, C. Zhou, D. Yin, J. Lin, X. Zou, Z. Shao, H. Yang, and J. Tang. Cogview: Mastering text-to-image generation via transformers, 2021.

[9] Z. Du, A. Zeng, Y. Dong, and J. Tang. Understanding emergent abilities of language models from the loss perspective, 2024.

[10] D. Hendrycks, C. Burns, S. Basart, A. Zou, M. Mazeika, D. Song, and J. Steinhardt. Measuring massive multitask language understanding. In International Conference on Learning Representations, 2021.

[11] D. Hendrycks and K. Gimpel. Gaussian error linear units (gelus). arXiv preprint arXiv:1606.08415, 2016.

[12] W. Hong, W. Wang, Q. Lv, J. Xu, W. Yu, J. Ji, Y. Wang, Z. Wang, Y. Zhang, J. Li, B. Xu, Y. Dong, M. Ding, and J. Tang. Cogagent: A visual language model for gui agents, 2023.

[13] Z. Hou, Y. Niu, Z. Du, X. Zhang, X. Liu, A. Zeng, Q. Zheng, M. Huang, H. Wang, J. Tang, and Y. Dong. Chatglm-rlhf: Practices of aligning large language models with human feedback, 2024.

[14] Y. Li, S. Bubeck, R. Eldan, A. D. Giorno, S. Gunasekar, and Y. T. Lee. Textbooks are all you need ii: phi-1.5 technical report, 2023.

[15] M. Liu, A. Zeng, B. Wang, P. Zhang, J. Tang, and Y. Dong. Apar: Llms can do auto-parallel auto-regressive decoding. ArXiv, abs/2401.06761, 2024.

[16] X. Liu, X. Lei, S. Wang, Y. Huang, Z. Feng, B. Wen, J. Cheng, P. Ke, Y. Xu, W. L. Tam, X. Zhang, L. Sun, H. Wang, J. Zhang, M. Huang, Y. Dong, and J. Tang. Alignbench: Benchmarking chinese alignment of large language models, 2023.

[17] X. Liu, X. Song, Y. Dong, and J. Tang. Extensive self-contrast enables feedback-free language model alignment, 2024.

[18] X. Liu, H. Yu, H. Zhang, Y. Xu, X. Lei, H. Lai, Y. Gu, H. Ding, K. Men, K. Yang, S. Zhang, X. Deng, A. Zeng, Z. Du, C. Zhang, S. Shen, T. Zhang, Y. Su, H. Sun, M. Huang, Y. Dong, and J. Tang. Agentbench: Evaluating llms as agents, 2023.

[19] Meta. Introducing meta llama 3: The most capable openly available llm to date. `https://ai.meta.com/blog/meta-llama-3/`, 2024.

[20] OpenAI. tiktoken. `https://github.com/openai/tiktoken`, 2023.

[21] R. OpenAI. Gpt-4 technical report. arXiv, pages 2303–08774, 2023.

[22] L. Ouyang, J. Wu, X. Jiang, D. Almeida, C. Wainwright, P. Mishkin, C. Zhang, S. Agarwal, K. Slama, A. Ray, et al. Training language models to follow instructions with human feedback. Advances in Neural Information Processing Systems, 35:27730–27744, 2022.

[23] J. S. Park, J. O'Brien, C. J. Cai, M. R. Morris, P. Liang, and M. S. Bernstein. Generative agents: Interactive simulacra of human behavior. In Proceedings of the 36th Annual ACM Symposium on User Interface Software and Technology, pages 1–22, 2023.

[24] O. Press, N. Smith, and M. Lewis. Train short, test long: Attention with linear biases enables input length extrapolation. In International Conference on Learning Representations, 2022.

[25] D. Rein, B. L. Hou, A. C. Stickland, J. Petty, R. Y. Pang, J. Dirani, J. Michael, and S. R. Bowman. GPQA: A graduate-level google-proof q&a benchmark. CoRR, abs/2311.12022, 2023.

[26] R. Sennrich, B. Haddow, and A. Birch. Neural machine translation of rare words with subword units. In Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 1715–1725, Berlin, Germany, 2016. Association for Computational Linguistics.

[27] N. Shazeer. Fast transformer decoding: One write-head is all you need. arXiv preprint arXiv:1911.02150, 2019.

[28] N. Shazeer. Glu variants improve transformer, 2020.

[29] M. Shridhar, X. Yuan, M.-A. Cote, Y. Bisk, A. Trischler, and M. Hausknecht. Alfworld: Aligning text and embodied environments for interactive learning. In International Conference on Learning Representations, 2020.

[30] A. Srivastava, A. Rastogi, A. Rao, A. A. M. Shoeb, A. Abid, A. Fisch, A. R. Brown, A. Santoro, A. Gupta, A. Garriga-Alonso, A. Kluska, A. Lewkowycz, A. Agarwal, A. Power, A. Ray, A. Warstadt, A. W. Kocurek, A. Safaya, A. Tazarv, A. Xiang, A. Parrish, A. Nie, A. Hussain, A. Askell, A. Dsouza, A. Rahane, A. S. Iyer, A. Andreassen, A. Santilli, A. Stuhlmüller, A. M. Dai, A. La, A. K. Lampinen, A. Zou, A. Jiang, A. Chen, A. Vuong, A. Gupta, A. Gottardi, A. Norelli, A. Venkatesh, A. Gholamidavoodi, A. Tabassum, A. Menezes, A. Kirubarajan, A. Mullokandov, A. Sabharwal, A. Herrick, A. Efrat, A. Erdem, A. Karakas, and et al. Beyond the imitation game: Quantifying and extrapolating the capabilities of language models. CoRR, abs/2206.04615, 2022.

[31] J. Su, Y. Lu, S. Pan, A. Murtadha, B. Wen, and Y. Liu. Roformer: Enhanced transformer with rotary position embedding. arXiv preprint arXiv:2104.09864, 2021.

[32] M. Suzgun, N. Scales, N. Schärli, S. Gehrmann, Y. Tay, H. W. Chung, A. Chowdhery, Q. V. Le, E. H. Chi, D. Zhou, and J. Wei. Challenging big-bench tasks and whether chain-of-thought can solve them. In A. Rogers, J. L. Boyd-Graber, and N. Okazaki, editors, Findings of the Association for Computational Linguistics: ACL 2023, Toronto, Canada, July 9-14, 2023, pages 13003–13051. Association for Computational Linguistics, 2023.

[33] G. Team, R. Anil, S. Borgeaud, Y. Wu, J.-B. Alayrac, J. Yu, R. Soricut, J. Schalkwyk, A. M. Dai, A. Hauth, K. Millican, D. Silver, S. Petrov, M. Johnson, I. Antonoglou, J. Schrittwieser, A. Glaese, J. Chen, E. Pitler, T. Lillicrap, A. Lazaridou, O. Firat, J. Molloy, M. Isard, P. R. Barham, T. Hennigan, B. Lee, F. Viola, M. Reynolds, Y. Xu, R. Doherty, E. Collins, C. Meyer, E. Rutherford, E. Moreira, K. Ayoub, M. Goel, G. Tucker, E. Piqueras, M. Krikun, I. Barr, N. Savinov, I. Danihelka, B. Roelofs, A. White, A. Andreassen, T. von Glehn, L. Yagati, M. Kazemi, L. Gonzalez, M. Khalman, J. Sygnowski, A. Frechette, C. Smith, L. Culp, L. Proleev, Y. Luan, X. Chen, J. Lottes, N. Schucher, F. Lebron, A. Rrustemi, N. Clay, P. Crone, T. Kocisky, J. Zhao, B. Perz, D. Yu, H. Howard, A. Bloniarz, J. W. Rae, H. Lu, L. Sifre, M. Maggioni, F. Alcober, D. Garrette, M. Barnes, S. Thakoor, J. Austin, G. Barth-Maron, W. Wong, R. Joshi, R. Chaabouni, D. Fatiha, A. Ahuja, R. Liu, Y. Li, S. Cogan, J. Chen, C. Jia, C. Gu, Q. Zhang, J. Grimstad, A. J. Hartman, M. Chadwick, G. S. Tomar, X. Garcia, E. Senter, E. Taropa, T. S. Pillai, J. Devlin, M. Laskin, D. de Las Casas, D. Valter, C. Tao, L. Blanco, A. P. Badia, D. Reitter, M. Chen, J. Brennan, C. Rivera, S. Brin, S. Iqbal, G. Surita, J. Labanowski, A. Rao, S. Winkler, E. Parisotto, Y. Gu, K. Olszewska, Y. Zhang, R. Addanki, A. Miech, A. Louis, L. E. Shafey, D. Teplyashin, G. Brown, E. Catt, N. Attaluri, J. Balaguer, J. Xiang, P. Wang, Z. Ashwood, A. Briukhov, A. Webson, S. Ganapathy, S. Sanghavi, A. Kannan, M.-W. Chang, A. Stjerngren, J. Djolonga, Y. Sun, A. Bapna, M. Aitchison, P. Pejman, H. Michalewski, T. Yu, C. Wang, J. Love, J. Ahn, D. Bloxwich, K. Han, P. Humphreys, T. Sellam, J. Bradbury, V. Godbole, S. Samangooei, B. Damoc, A. Kaskasoli, S. M. R. Arnold, V. Vasudevan, S. Agrawal, J. Riesa, D. Lepikhin, R. Tanburn, S. Srinivasan, H. Lim, S. Hodkinson, P. Shyam, J. Ferret, S. Hand, A. Garg, T. L. Paine, J. Li, Y. Li, M. Giang, A. Neitz, Z. Abbas, S. York, M. Reid, E. Cole, A. Chowdhery, D. Das, D. Rogoziska, V. Nikolaev, P. Sprechmann, Z. Nado, L. Zilka, F. Prost, L. He, M. Monteiro, G. Mishra, C. Welty, J. Newlan, D. Jia, M. Allamanis, C. H. Hu, R. de Liedekerke, J. Gilmer, C. Saroufim, S. Rijhwani, S. Hou, D. Shrivastava, A. Baddepudi, A. Goldin, A. Ozturel, A. Cassirer, Y. Xu, D. Sohn, D. Sachan, R. K. Amplayo, C. Swanson, D. Petrova, S. Narayan, A. Guez, S. Brahma, J. Landon, M. Patel, R. Zhao, K. Villela, L. Wang, W. Jia, M. Rahtz, M. Giménez, L. Yeung, H. Lin, J. Keeling, P. Georgiev, D. Mincu, B. Wu, S. Haykal, R. Saputro, K. Vodrahalli, J. Qin, Z. Cankara, A. Sharma, N. Fernando, W. Hawkins, B. Neyshabur, S. Kim, A. Hutter, P. Agrawal, A. Castro-Ros, G. van den Driessche, T. Wang, F. Yang, S. yiin Chang, P. Komarek, R. McIlroy, M. Lui, G. Zhang, W. Farhan, M. Sharman, P. Natsev, P. Michel, Y. Cheng, Y. Bansal, S. Qiao, K. Cao, S. Shakeri, C. Butterfield, J. Chung, P. K. Rubenstein, S. Agrawal, A. Mensch, K. Soparkar, K. Lenc, T. Chung, A. Pope, L. Maggiore, J. Kay, P. Jhakra, S. Wang, J. Maynez, M. Phuong, T. Tobin, A. Tacchetti, M. Trebacz, K. Robinson, Y. Katariya, S. Riedel, P. Bailey, K. Xiao, N. Ghelani, L. Aroyo, A. Slone, N. Houlsby, X. Xiong, Z. Yang, E. Gribovskaya, J. Adler, M. Wirth, L. Lee, M. Li, T. Kagohara, J. Pavagadhi, S. Bridgers, A. Bortsova, S. Ghemawat, Z. Ahmed, T. Liu, R. Powell, V. Bolina, M. Iinuma, P. Zablotskaia, J. Besley, D.-W. Chung, T. Dozat, R. Comanescu, X. Si, J. Greer, G. Su, M. Polacek, R. L. Kaufman, S. Tokumine, H. Hu, E. Buchatskaya, Y. Miao, M. Elhawaty, A. Siddhant, N. Tomasev, J. Xing, C. Greer, H. Miller, S. Ashraf, A. Roy, Z. Zhang, A. Ma, A. Filos, M. Besta, R. Blevins, T. Klimenko, C.-K. Yeh, S. Changpinyo, J. Mu, O. Chang, M. Pajarskas, C. Muir, V. Cohen, C. L. Lan, K. Haridasan, A. Marathe, S. Hansen, S. Douglas, R. Samuel, M. Wang, S. Austin, C. Lan, J. Jiang, J. Chiu, J. A. Lorenzo, L. L. Sjösund, S. Cevey, Z. Gleicher, T. Avrahami, A. Boral, H. Srinivasan, V. Selo, R. May, K. Aisopos, L. Hussenot, L. B. Soares, K. Baumli, M. B. Chang, A. Recasens, B. Caine, A. Pritzel, F. Pavetic, F. Pardo, A. Gergely, J. Frye, V. Ramasesh, D. Horgan, K. Badola, N. Kassner, S. Roy, E. Dyer, V. Campos, A. Tomala, Y. Tang, D. E. Badawy, E. White, B. Mustafa, O. Lang, A. Jindal, S. Vikram, Z. Gong, S. Caelles, R. Hemsley, G. Thornton, F. Feng, W. Stokowiec, C. Zheng, P. Thacker, Çalar Ünlü, Z. Zhang, M. Saleh, J. Svensson, M. Bileschi, P. Patil, A. Anand, R. Ring, K. Tsihlas, A. Vezer, M. Selvi, T. Shevlane, M. Rodriguez, T. Kwiatkowski, S. Daruki, K. Rong, A. Dafoe, N. FitzGerald, K. Gu-Lemberg, M. Khan, L. A. Hendricks, M. Pellat, V. Feinberg, J. Cobon-Kerr, T. Sainath, M. Rauh, S. H. Hashemi, R. Ives, Y. Hasson, Y. Li, E. Noland, Y. Cao, N. Byrd, L. Hou, Q. Wang, T. Sottiaux, M. Paganini, J.-B. Lespiau, A. Moufarek, S. Hassan, K. Shivakumar, J. van Amersfoort, A. Mandhane, P. Joshi, A. Goyal, M. Tung, A. Brock, H. Sheahan, V. Misra, C. Li, N. Rakievi, M. Dehghani, F. Liu, S. Mittal, J. Oh, S. Noury, E. Sezener, F. Huot, M. Lamm, N. D. Cao, C. Chen, G. Elsayed, E. Chi, M. Mahdieh, I. Tenney, N. Hua, I. Petrychenko, P. Kane, D. Scandinaro, R. Jain, J. Uesato, R. Datta, A. Sadovsky, O. Bunyan, D. Rabiej, S. Wu, J. Zhang, G. Vasudevan, E. Leurent, M. Alnahlawi, I. Georgescu, N. Wei,

I. Zheng, B. Chan, P. G. Rabinovitch, P. Stanczyk, Y. Zhang, D. Steiner, S. Naskar, M. Azzam, M. Johnson, A. Paszke, C.-C. Chiu, J. S. Elias, A. Mohiuddin, F. Muhammad, J. Miao, A. Lee, N. Vieillard, S. Potluri, J. Park, E. Davoodi, J. Zhang, J. Stanway, D. Garmon, A. Karmarkar, Z. Dong, J. Lee, A. Kumar, L. Zhou, J. Evens, W. Isaac, Z. Chen, J. Jia, A. Levskaya, Z. Zhu, C. Gorgolewski, P. Grabowski, Y. Mao, A. Magni, K. Yao, J. Snaider, N. Casagrande, P. Suganthan, E. Palmer, G. Irving, E. Loper, M. Faruqui, I. Arkatkar, N. Chen, I. Shafran, M. Fink, A. Castaño, I. Giannoumis, W. Kim, M. Rybiski, A. Sreevatsa, J. Prendki, D. Soergel, A. Goedeckemeyer, W. Gierke, M. Jafari, M. Gaba, J. Wiesner, D. G. Wright, Y. Wei, H. Vashisht, Y. Kulizhskaya, J. Hoover, M. Le, L. Li, C. Iwuanyanwu, L. Liu, K. Ramirez, A. Khorlin, A. Cui, T. LIN, M. Georgiev, M. Wu, R. Aguilar, K. Pallo, A. Chakladar, A. Repina, X. Wu, T. van der Weide, P. Ponnapalli, C. Kaplan, J. Simsa, S. Li, O. Dousse, F. Yang, J. Piper, N. Ie, M. Lui, R. Pasumarthi, N. Lintz, A. Vijayakumar, L. N. Thiet, D. Andor, P. Valenzuela, C. Paduraru, D. Peng, K. Lee, S. Zhang, S. Greene, D. D. Nguyen, P. Kurylowicz, S. Velury, S. Krause, C. Hardin, L. Dixon, L. Janzer, K. Choo, Z. Feng, B. Zhang, A. Singhal, T. Latkar, M. Zhang, Q. Le, E. A. Abellan, D. Du, D. McKinnon, N. Antropova, T. Bolukbasi, O. Keller, D. Reid, D. Finchelstein, M. A. Raad, R. Crocker, P. Hawkins, R. Dadashi, C. Gaffney, S. Lall, K. Franko, E. Filonov, A. Bulanova, R. Leblond, V. Yadav, S. Chung, H. Askham, L. C. Cobo, K. Xu, F. Fischer, J. Xu, C. Sorokin, C. Alberti, C.-C. Lin, C. Evans, H. Zhou, A. Dimitriev, H. Forbes, D. Banarse, Z. Tung, J. Liu, M. Omernick, C. Bishop, C. Kumar, R. Sterneck, R. Foley, R. Jain, S. Mishra, J. Xia, T. Bos, G. Cideron, E. Amid, F. Piccinno, X. Wang, P. Banzal, P. Gurita, H. Noga, P. Shah, D. J. Mankowitz, A. Polozov, N. Kushman, V. Krakovna, S. Brown, M. Bateni, D. Duan, V. Firoiu, M. Thotakuri, T. Natan, A. Mohananey, M. Geist, S. Mudgal, S. Girgin, H. Li, J. Ye, O. Roval, R. Tojo, M. Kwong, J. Lee-Thorp, C. Yew, Q. Yuan, S. Bagri, D. Sinopalnikov, S. Ramos, J. Mellor, A. Sharma, A. Severyn, J. Lai, K. Wu, H.-T. Cheng, D. Miller, N. Sonnerat, D. Vnukov, R. Greig, J. Beattie, E. Caveness, L. Bai, J. Eisenschlos, A. Korchemniy, T. Tsai, M. Jasarevic, W. Kong, P. Dao, Z. Zheng, F. Liu, F. Yang, R. Zhu, M. Geller, T. H. Teh, J. Sanmiya, E. Gladchenko, N. Trdin, A. Sozanschi, D. Toyama, E. Rosen, S. Tavakkol, L. Xue, C. Elkind, O. Woodman, J. Carpenter, G. Papamakarios, R. Kemp, S. Kafle, T. Grunina, R. Sinha, A. Talbert, A. Goyal, D. Wu, D. Owusu-Afriyie, C. Du, C. Thornton, J. Pont-Tuset, P. Narayana, J. Li, S. Fatehi, J. Wieting, O. Ajmeri, B. Uria, T. Zhu, Y. Ko, L. Knight, A. Héliou, N. Niu, S. Gu, C. Pang, D. Tran, Y. Li, N. Levine, A. Stolovich, N. Kalb, R. Santamaria-Fernandez, S. Goenka, W. Yustalim, R. Strudel, A. Elqursh, B. Lakshminarayanan, C. Deck, S. Upadhyay, H. Lee, M. Dusenberry, Z. Li, X. Wang, K. Levin, R. Hoffmann, D. Holtmann-Rice, O. Bachem, S. Yue, S. Arora, E. Malmi, D. Mirylenka, Q. Tan, C. Koh, S. H. Yeganeh, S. Põder, S. Zheng, F. Pongetti, M. Tariq, Y. Sun, L. Ionita, M. Seyedhosseini, P. Tafti, R. Kotikalapudi, Z. Liu, A. Gulati, J. Liu, X. Ye, B. Chrzaszcz, L. Wang, N. Sethi, T. Li, B. Brown, S. Singh, W. Fan, A. Parisi, J. Stanton, C. Kuang, V. Koverkathu, C. A. Choquette-Choo, Y. Li, T. Lu, A. Ittycheriah, P. Shroff, P. Sun, M. Varadarajan, S. Bahargam, R. Willoughby, D. Gaddy, I. Dasgupta, G. Desjardins, M. Cornero, B. Robenek, B. Mittal, B. Albrecht, A. Shenoy, F. Moiseev, H. Jacobsson, A. Ghaffarkhah, M. Rivière, A. Walton, C. Crepy, A. Parrish, Y. Liu, Z. Zhou, C. Farabet, C. Radebaugh, P. Srinivasan, C. van der Salm, A. Fidjeland, S. Scellato, E. Latorre-Chimoto, H. Klimczak-Pluciska, D. Bridson, D. de Cesare, T. Hudson, P. Mendolicchio, L. Walker, A. Morris, I. Penchev, M. Mauger, A. Guseynov, A. Reid, S. Odoom, L. Loher, V. Cotruta, M. Yenugula, D. Grewe, A. Petrushkina, T. Duerig, A. Sanchez, S. Yadlowsky, A. Shen, A. Globerson, A. Kurzrok, L. Webb, S. Dua, D. Li, P. Lahoti, S. Bhupatiraju, D. Hurt, H. Qureshi, A. Agarwal, T. Shani, M. Eyal, A. Khare, S. R. Belle, L. Wang, C. Tekur, M. S. Kale, J. Wei, R. Sang, B. Saeta, T. Liechty, Y. Sun, Y. Zhao, S. Lee, P. Nayak, D. Fritz, M. R. Vuyyuru, J. Aslanides, N. Vyas, M. Wicke, X. Ma, T. Bilal, E. Eltyshev, D. Balle, N. Martin, H. Cate, J. Manyika, K. Amiri, Y. Kim, X. Xiong, K. Kang, F. Luisier, N. Tripuraneni, D. Madras, M. Guo, A. Waters, O. Wang, J. Ainslie, J. Baldridge, H. Zhang, G. Pruthi, J. Bauer, F. Yang, R. Mansour, J. Gelman, Y. Xu, G. Polovets, J. Liu, H. Cai, W. Chen, X. Sheng, E. Xue, S. Ozair, A. Yu, C. Angermueller, X. Li, W. Wang, J. Wiesinger, E. Koukoumidis, Y. Tian, A. Iyer, M. Gurumurthy, M. Goldenson, P. Shah, M. Blake, H. Yu, A. Urbanowicz, J. Palomaki, C. Fernando, K. Brooks, K. Durden, H. Mehta, N. Momchev, E. Rahimtoroghi, M. Georgaki, A. Raul, S. Ruder, M. Redshaw, J. Lee, K. Jalan, D. Li, G. Perng, B. Hechtman, P. Schuh, M. Nasr, M. Chen, K. Milan, V. Mikulik, T. Strohman, J. Franco, T. Green, D. Hassabis, K. Kavukcuoglu, J. Dean, and O. Vinyals. Gemini: A family of highly capable multimodal models, 2023.

[34] H. Touvron, L. Martin, K. Stone, P. Albert, A. Almahairi, Y. Babaei, N. Bashlykov, S. Batra, P. Bhargava, S. Bhosale, D. Bikel, L. Blecher, C. C. Ferrer, M. Chen, G. Cucurull, D. Esiobu, J. Fernandes, J. Fu, W. Fu, B. Fuller, C. Gao, V. Goswami, N. Goyal, A. Hartshorn, S. Hosseini, R. Hou, H. Inan, M. Kardas, V. Kerkez, M. Khabsa, I. Kloumann, A. Korenev, P. S. Koura, M.-A. Lachaux, T. Lavril, J. Lee, D. Liskovich, Y. Lu, Y. Mao, X. Martinet, T. Mihaylov, P. Mishra, I. Molybog, Y. Nie, A. Poulton, J. Reizenstein, R. Rungta, K. Saladi, A. Schelten, R. Silva, E. M. Smith, R. Subramanian, X. E. Tan, B. Tang, R. Taylor, A. Williams, J. X. Kuan, P. Xu, Z. Yan, I. Zarov, Y. Zhang, A. Fan, M. Kambadur, S. Narang, A. Rodriguez, R. Stojnic, S. Edunov, and T. Scialom. Llama 2: Open foundation and fine-tuned chat models, 2023.

[35] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin. Attention is all you need, 2023.

[36] H. Wang, S. Ma, L. Dong, S. Huang, D. Zhang, and F. Wei. Deepnet: Scaling transformers to 1,000 layers, 2022.

[37] W. Wang, Q. Lv, W. Yu, W. Hong, J. Qi, Y. Wang, J. Ji, Z. Yang, L. Zhao, X. Song, J. Xu, B. Xu, J. Li, Y. Dong, M. Ding, and J. Tang. Cogvlm: Visual expert for pretrained language models, 2023.

[38] J. Wei, X. Wang, D. Schuurmans, M. Bosma, B. Ichter, F. Xia, E. H. Chi, Q. V. Le, and D. Zhou. Chain-of-thought prompting elicits reasoning in large language models. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh, editors, Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022, 2022.

[39] W. Xiong, J. Liu, I. Molybog, H. Zhang, P. Bhargava, R. Hou, L. Martin, R. Rungta, K. A. Sankararaman, B. Oguz, et al. Effective long-context scaling of foundation models. arXiv preprint arXiv:2309.16039, 2023.

[40] Y. Xu, X. Liu, X. Liu, Z. Hou, Y. Li, X. Zhang, Z. Wang, A. Zeng, Z. Du, W. Zhao, J. Tang, and Y. Dong. Chatglm-math: Improving math problem-solving in large language models with a self-critique pipeline, 2024.

[41] F. Yan, H. Mao, C. C.-J. Ji, T. Zhang, S. G. Patil, I. Stoica, and J. E. Gonzalez. Berkeley function calling leaderboard. 2024.

[42] S. Yang, W.-L. Chiang, L. Zheng, J. E. Gonzalez, and I. Stoica. Rethinking benchmark and contamination for language models with rephrased samples. arXiv preprint arXiv:2311.04850, 2023.

[43] S. Yao, H. Chen, J. Yang, and K. Narasimhan. Webshop: Towards scalable real-world web interaction with grounded language agents. Advances in Neural Information Processing Systems, 35:20744–20757, 2022.

[44] S. Yao, J. Zhao, D. Yu, N. Du, I. Shafran, K. Narasimhan, and Y. Cao. React: Synergizing reasoning and acting in language models. arXiv preprint arXiv:2210.03629, 2022.

[45] A. Zeng, M. Liu, R. Lu, B. Wang, X. Liu, Y. Dong, and J. Tang. Agenttuning: Enabling generalized agent abilities for llms, 2023.

[46] A. Zeng, X. Liu, Z. Du, Z. Wang, H. Lai, M. Ding, Z. Yang, Y. Xu, W. Zheng, X. Xia, et al. Glm-130b: An open bilingual pre-trained model. arXiv preprint arXiv:2210.02414, 2022.

[47] S. Zhang, H. Zhao, X. Liu, Q. Zheng, Z. Qi, X. Gu, X. Zhang, Y. Dong, and J. Tang. Naturalcodebench: Examining coding performance mismatch on humaneval and natural user prompts. arXiv preprint arXiv:2405.04520, 2024.

[48] Z. Zhang, L. Lei, L. Wu, R. Sun, Y. Huang, C. Long, X. Liu, X. Lei, J. Tang, and M. Huang. Safetybench: Evaluating the safety of large language models with multiple choice questions. arXiv preprint arXiv:2309.07045, 2023.

[49] Q. Zheng, X. Xia, X. Zou, Y. Dong, S. Wang, Y. Xue, Z. Wang, L. Shen, A. Wang, Y. Li, T. Su, Z. Yang, and J. Tang. Codegeex: A pre-trained model for code generation with multilingual evaluations on humaneval-x, 2023.

[50] W. Zheng, J. Teng, Z. Yang, W. Wang, J. Chen, X. Gu, Y. Dong, M. Ding, and J. Tang. Cogview3: Finer and faster text-to-image generation via relay diffusion, 2024.

[51] C. Zhou, P. Liu, P. Xu, S. Iyer, J. Sun, Y. Mao, X. Ma, A. Efrat, P. Yu, L. Yu, S. Zhang, G. Ghosh, M. Lewis, L. Zettlemoyer, and O. Levy. Lima: Less is more for alignment, 2023.

[52] J. Zhou, T. Lu, S. Mishra, S. Brahma, S. Basu, Y. Luan, D. Zhou, and L. Hou. Instruction-following evaluation for large language models. arXiv preprint arXiv:2311.07911, 2023.