An extended version of this report is coming soon.

LLM360 K2-65B: Scaling Up Fully Transparent Open-Source LLMs

Zhengzhong Liu[12], **Bowen Tan**[3], **Hongyi Wang**[37], **Willie Neiswanger**[4], **Tianhua Tao**[5], **Haonan Li**[1], **Fajri Koto**[1], **Yuqi Wang**[2], **Suqi Sun**[2], **Omkar Pangarkar**[2], **Richard Fan**[2], **Yi Gu**[6], **Victor Miller**[2], **Liqun Ma**[1], **Liping Tang**[1], **Nikhil Ranjan**[1], **Yonghao Zhuang**[3], **Guowei He**[1], **Renxi Wang**[1], **Mingkai Deng**[3], **Robin Algayres**[1], **Yuanzhi Li**[1], **Zhiqiang Shen**[1], **Preslav Nakov**[1], **Eric Xing**[13]

[1]MBZUAI [2]Petuum, Inc. [3]Carnegie Mellon University [4]University of Southern California [5]University of Illinois Urbana-Champaign [6]University of California San Diego [7]Rutgers University

team@llm360.ai

Abstract

In this paper, we present **LLM360 K2-65B**, the most powerful fully transparent open-source large language model (LLM) released to date. K2 is a 65 billion parameter LLM, which follows best practices for reproducibility from the LLM360 project. Despite numerous efforts to develop and release open-source LLMs, full transparency around the training process still remains limited. This is especially true for large-scale models, as prior releases have typically been limited to around 7 billion parameters. While processes and lessons for training large-scale models are valuable to the community, few exist. For instance, what are the challenges and training dynamics when training an LLM with tens of billions of parameters? Are these unique to larger models? To this end, we pre-train K2 from scratch on a carefully curated dataset of 1.4 trillion tokens, mixed from web crawls, high-quality textbooks, publications, domain-specific knowledge, and programming code. We find that K2 outperforms LLaMA-65B and is comparable to Llama2-70B, yet it requires significantly fewer FLOPS and tokens for training. We detail our findings during K2's development, as well as the challenges encountered and their resolutions. We envision that K2 can serve both as a strong base model for product development with more flexibility and also help researchers dive deeper into LLM pretraining behavior at a large model parameter scale.

🤗	Checkpoints:	huggingface.co/LLM360/K2
	Code:	github.com/llm360
	W&B Logs:	wandb.ai/llm360/K2
🤗	Data Sequence:	huggingface.co/datasets/LLM360/K2Datasets
🤗	K2 Chat:	huggingface.co/LLM360/K2-Chat
	K2 Suites:	www.llm360.ai/#two
🤗	Prompt Gallery:	huggingface.co/spaces/LLM360/k2-gallery
🤗	Evaluation Gallery:	huggingface.co/spaces/LLM360/k2-eval-gallery

A technical report of the LLM 360 project.

1 Introduction

Over the past year, the LLM360 project has released a set of *fully open-source* and reproducible large language models (LLMs) ranging from English models to code-generating models, as well as their instruction-tuned and chat variants (Liu et al., 2023b). We are pleased to see multiple other dedicated teams pursuing a similar endeavor to release transparent, open-source LLMs (Groeneveld et al., 2024; Shen et al., 2024; Snowflake, 2024; Biderman et al., 2023; Zheng et al., 2024). However, until now the full pretraining details of most powerful recent LLMs are still mysterious, as virtually all prior fully-reproducible open-source LLM releases have remained at a relatively smaller scale ($\leq$12B parameters) and thus lag far behind the performance and model quality of many mainstream LLMs, such as LLaMA-65B and Llama2-70B (Touvron et al., 2023a,b).

In this technical report, we present **LLM360 K2-65B**, the most powerful fully-reproducible open-source LLM released to date, as the newest member of the LLM360 model family. K2 is a 65 billion parameter large language model trained completely from scratch on a mixture of web crawl data, high-quality textbooks, domain-specific data, and programming code (1.4 trillion tokens in total). To the best of our knowledge, **K2 is the very first fully open-source LLM of this size**. We follow best practices of the LLM360 project to release a comprehensive set of pretraining details for K2 (Liu et al., 2023b), including all pretraining and finetuning code, training algorithm and model details (*e.g.,* hyperparameters, schedules, architecture, and designs), all logs and metrics collected during training, all intermediate model checkpoints saved during training, and the exact pretraining data used.

Performance-wise, K2 significantly outperforms LLaMA-65B, and rivals Llama2-70B on various standard benchmarks (*e.g.,* GSM8K, HumanEval, etc. (Cobbe et al., 2021; Chen et al., 2021)), while using a much smaller pretraining corpus; specifically, K2 demonstrates a roughly 35% reduction in FLOPS in comparison with Llama2-70B. In addition, K2 demonstrates significantly better math reasoning and coding abilities than Llama2-70B, as well as greater proficiency in the medical domain.

We envision that the release of K2 will further enable the community to further develop impactful models. K2 can also help researchers and practitioners who are interested in studying the behaviors of large-scale LLMs. For users who would like to build applications, or create small-scale LLMs through knowledge distillation (*e.g.,* for deployment on mobile and embedded systems), K2 offers a more flexible license than many recent large-parameter models.

K2 Released Artifacts. The artifacts released for K2 largely follow the LLM360 full-transparency approach (Liu et al., 2023b). In this release, the major artifacts include:

- **Code**: All code, including for training, finetuning, and data preparation.
- **Model checkpoints**: 140 intermediate model checkpoints, evenly spaced for stage 1 (120 checkpoints)[1] and stage 2 (20 checkpoints) respectively.
- **Data**: The exact training data sequence, split into chunks that correspond to the checkpoints.
- **Logs**: The Weights & Biases training logs, evaluation logs, and system logs.
- **Finetunes**: An instruction tuned model, K2-Chat, and the finetuning datasets.

We further organize and release a few resources incorporating feedback from the community.

K2 Galleries. The exact output of the language models during evaluation and prompting carries useful information about the model. For instance, the perplexity values/scores of each choice in multiple-choice questions offer a more-nuanced understanding than final accuracy alone. To make the evaluation transparent, as suggested by Biderman et al. (2024), we release all evaluation prompts, hyperparameters, and outputs. We follow the Bloom Book[2] approach and release the model output as two K2 galleries. First, the *K2 Prompt Gallery*[3] contains all K2 checkpoints' outputs on a curated list of prompts, allowing one to intuitively compare and understand the development of a model over

[1] We saved and numbered 360 checkpoints, but only uploaded 120 so far, since each can take >100GB space.

[2] https://huggingface.co/spaces/bigscience/bloom-book

[3] https://huggingface.co/spaces/LLM360/k2-gallery

the course of pretraining. Second, the *K2 Evaluation Gallery*[4] is designed similarly and contains the raw evaluation outputs (e.g., perplexity of each option, generated text) for the benchmark tasks. The Evaluation Gallery helps in understanding the model's development across various abilities and provides a more-comprehensive view on the benchmarks.

K2 Suites. The LLM360 project has released a variety of artifacts, but users sometimes have difficulty finding the right resources. To address this, we have organized the artifacts into K2 Suites. Each suite contains materials organized for a particular scenario.

- The **Research Suite** contains artifacts for researchers to explore and conduct research on LLMs, such as training dynamics and AI safety. This suite features items like intermediate checkpoints and the aligned data sequence.

  We further present two **K2 loss spikes**. During the training, we encountered two major loss spikes. In our final run, we always restarted the training from a checkpoint prior to the spike. However, loss spikes are relatively rare, in order to facilitate research on the spike phenomenon, we allowed the training to run for a few steps after the spikes occurred. We release the checkpoints obtained this way in separate model repositories. We discuss our preliminary findings of the spikes at §3.1.

- The **Pretraining Suite** organizes the artifacts useful for reproducing or extending our pretraining process. Specifically, we release (1) the full data preparation recipe to recreate the data sequence used for training; (2) the pretraining code for training a model similar to LLM360 model architectures; (3) the evaluation code and instructions to measure the model performance; and (4) the model analysis approaches to understand the states of the model. We further provide the intermediate model outputs and evaluation results as references, for pretrainers to validate the progress against our trajectory.

- The **Developer Suite** contains artifacts useful for developing applications, such as fine-tuning and deployment, given the pretrained models. This includes code for fine-tuning and inference, and tutorials describing how to deploy fine-tuning and inference, and conduct evaluation on these models.

2 Modeling

The model architecture of K2 largely follows the architecture of the LLaMA-65B model (Touvron et al., 2023a). It has 80 transformer layers each with a hidden dimension of 8192 and 64 attention heads. We did not choose to use Group Query Attention to simplify the model architecture for research purposes. More detailed information is shown in Table 1.

Hyperparameter	Value
Layers	80
Hidden Size	8192
Intermediate Size (in MLPs)	22016
RMSNorm ϵ	$1e^{-5}$
Embedding Positions	32032
Vocab Size	32018

Table 1: A subset of the model architecture & hyperparameters used in K2.

Our tokenizer is also based on LLaMA's, and, following Starcoder (Li et al., 2023), we add 18 code-related special tokens, e.g., `<jupyter_code>`, and `<fim_suffix>`, to accomondate Github data such as jupyter notebooks and issues, as well as fill-in-the-middle code pretraining (Bavarian et al., 2022). The model has 32032 embedding positions with a smaller vocabulary size of 32018, which encourages users to add other special tokens creatively in various downstream applications without a great deal of additional effort.

3 Training

We briefly describe our training procedures in this section, including both the pretraining stage and finetuning stage. The exact details are also released in our training code repository.[5]

[4] https://huggingface.co/spaces/LLM360/k2-eval-gallery
[5] https://github.com/LLM360/k2-train

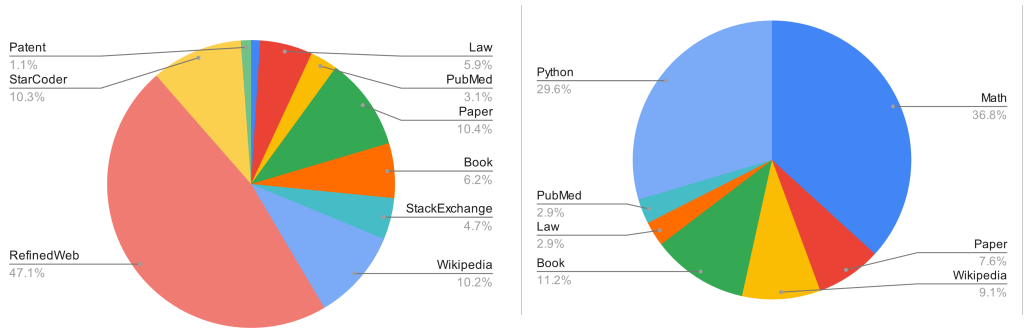

Figure 1: Data mix of pretraining stage 1 (**left**) and stage 2 (**right**). In stage 1, *Paper* data includes ArXiv from RedPajama (Together Computer, 2023) and S2ORC (Lo et al., 2020). USPTO (Gao et al., 2020) and Pile-of-law (Henderson* et al., 2022) are used as *Patent* and *Law* domain texts, respectively. In stage 2, SimpleWiki from Dolma (Soldaini et al., 2024) is added into *Wikipedia*. *Math* data includes Algebraic-Stack (Azerbayev et al., 2023) and Open-Web-Math (Paster et al., 2023). *Paper* data consists of ArXiv (Together Computer, 2023), S2ORC (Lo et al., 2020), and PES2O (Soldaini & Lo, 2023).

3.1 Pretraining

2-Stage Training. We apply a two-stage pretraining for K2, similar to Tao et al. (2024), except that we propose a major first stage, and a second stage with fewer tokens. Specifically, stage 1 is designed to establish the model's basic language ability, which is trained on 1.4T tokens from various sources including web text, academic papers, books, code, as well as math, medical, and legal documents. Following LLaMA-65B settings, the data in stage 1 is packed into samples with context length 2048. Stage 2 is designed to further enhance the model's generation ability (such as arithmetic and coding), as well as expanding the context length. Hence we sample more data that has longer sequences, such as papers and books. In this stage, 69.3B tokens are used to extend the context length to 8192 and enhance K2's math reasoning and coding abilities. The detailed data mix of both stages is shown in Figure. 1. To keep a reasonable proportion of subsets, we repeat or truncate some subsets. For example, USPTO is repeated three times while only a half of Starcoder is used.[6]

Data Sampling. We process all the data into 360 and 20 data chunks separately in stage 1 and 2, with a checkpoint saved after each data chunk, resulting into 380[7] K2 checkpoints along the training process. Notebly, in every stage, we make sure every subset of training data is evenly distributed into all data chunks, in order to reduce the variance of data sampling.

Optimization. In both stages, we use AdamW optimizer with $\beta_1 = 0.9$, $\beta_2 = 0.95$, and a gradient clipping of 1.0 and weight decay of 0.1. The batch size for both stages is 4M tokens. Stage 1 and 2 has 2000 and 500 warmup steps, respectively. In stage 1, we uses a cosine learning rate from 1.5×10^{-4} to 1.5×10^{-5}. In stage 2, the learning rate is linearly decayed from 10^{-4} to 0.

Infrastructure. K2 is pre-trained on a cluster consisting of 480 Nvidia A100 80GB GPUs hosted on Nvidia's NGC cloud. Our LLM pre-training framework is adapted from Megatron-LM (Shoeybi et al., 2019; Narayanan et al., 2021). We use a carefully tuned parallelism strategy that combines data, tensor-model, and pipeline parallelism. More specifically, we use 8-way tensor-model parallelism, 4-way pipeline parallelism, and 15-way data parallelism (such that $8 \times 4 \times 15 = 480$ GPUs). The parallelism strategy tuning procedure follows and is inspired by the heuristics and methods introduced in (Narayanan et al., 2021; Zheng et al., 2022; Li et al., 2022).[8] We also enable sequence parallelism

[6]For detailed subset-wise repeat or truncation, see: https://github.com/LLM360/k2-data-prep/blob/master/gather.py#99-L110.

[7]As mentioned above, only 120 are shared for stage 1 due to the large size of model checkpoints.

[8]In our stage 1, the global batch size is 2040, and we use a micro-batch size of 4 per data parallel GPU group. Thus, the number of micro-batches is $2040/15/4 = 34$, which is much larger compared to the number

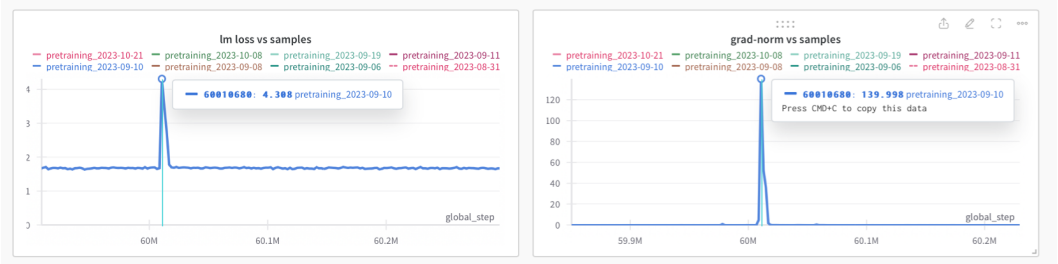

Figure 2: An example of *benign spikes* during pretraining.

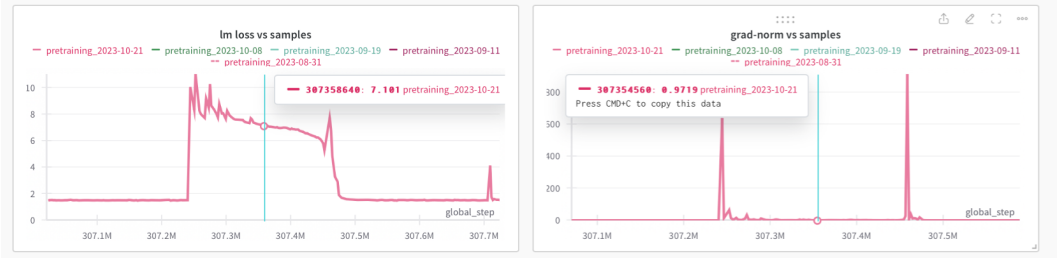

Figure 3: An example of *malignant spikes* during pretraining.

to further reduce the GPU memory cost (Korthikanti et al., 2023). BF16 mixed precision and FlashAttention-2 are enabled to speed up the training (Dao, 2023).

Observation: Loss Spikes. Similar to several other prior works involving pretraining, we observe many spikes in our loss curve, and find that some of them significantly influence training while others do not. Specifically, we find that loss spikes usually come with large gradient norms; considering we have gradient clipping of 1.0 in the optimization, the updates on the model at those steps can be minor, resulting in an insignificant effect on the model's performance. However, sometimes the spike can last for more than 100 steps, where some gradient norms are small inside this span. In our evaluation, such long loss spikes are destructive, and we roll back the training to bypass those spikes. We refer to these as *malignant spikes* (while we refer to spikes that have an insignificant effect as *benign spikes*); see Figures 2 and 3 for examples. Two major malignant spikes were observed during pretraining and we recorded both incidents as artifacts for the community to study.

3.2 Instruction Finetuning

We also release a finetuned model of K2: K2-Chat. Our data for finetuning primarily consists of 1M chat samples from OpenHermes-2.5 (Teknium, 2023) and 3M samples from FLAN (Longpre et al., 2023), as well as 300K other variously collected QA samples. We adopt the Do-Not-Answer dataset (Wang et al., 2024), and further collect a few UAE culture related data, resulting in 2700 alignment samples, to discourage our model from delivering harmful or inappropriate responses such as those involving crime and displaying toxic behavior. We pack all of the chat data into 8K-token samples with the following template for the sake of model serving:

```
{system_prompt}<|endofsystemprompt|>
<|beginofuser|>{user_instruction}<|beginofsystem|>{model_response}<|endoftext|>
...
<|endofchat|>
```

where `<|endofsystemprompt|>`, `<|beginofuser|>`, `<|beginofsystem|>`, and `<|endofchat|>` are newly added special tokens.

of pipeline stages (which is four). Therefore, the pipeline bubble in our hybrid parallelism strategy is negligible by design.

4 Evaluation

We conduct evaluation on a wide range of benchmarks to measure the model performance, mainly sourcing from LM-Evaluation-Harness and BigCode-Evaluation-Harness (Gao et al., 2023; Ben Allal et al., 2022). The benchmark spans over a variety of aspects in natural language, including **Reasoning**: Hellaswag (Zellers et al., 2019), ARC (Clark et al., 2018), Winogrande (Sakaguchi et al., 2021), PIQA (Bisk et al., 2020), BBH-COT (Suzgun et al., 2022), LogiQA2.0 (Liu et al., 2023a); **Question Answering**: OpenBookQA (Mihaylov et al., 2018), RACE (Lai et al., 2017); **General Knowledge**: MMLU (Hendrycks et al., 2021); **Math**: GSM8K (Cobbe et al., 2021), MathQA (Amini et al., 2019); **Truthfulness**: TruthfulQA (Lin et al., 2021); **Biases**: CrowS-Paris (Nangia et al., 2020). We also evaluate on **Coding**: HumanEval (Chen et al., 2021), MBPP (Austin et al., 2021); and **Medical**: MedQA (Jin et al., 2020), MedMCQA (Pal et al., 2022), PubMedQA (Jin et al., 2019). More Evaluation details can be found in the Appendix A.

Tokens Trained	K2-65B 1.4T	LLaMA-65B 1.4T	Llama2-70B 2T	Falcon-40B 1T	Falcon-180B 3.5T
			Natural Language Benchmarks		
MMLU (0-shot)	64.8	59.7	65.4	53.4	65.7
RACE (0-shot)	40.6	41.8	42.7	40.0	41.1
HellaSwag (10-shot)	85.5	85.9	86.9	85.3	89.0
PIQA (5-shot)	84.6	83.9	84.3	84.8	87.1
ARC-easy (5-shot)	86.4	86.9	88.2	85.6	89.6
ARC-challenge (25-shot)	64.8	63.2	67.2	61.9	69.5
OpenBookQA (5-shot)	49.2	50.0	52.4	49.0	52.2
Winogrande (5-shot)	77.0	77.2	77.7	76.6	86.6
TruthfulQA (0-shot)	40.8	42.6	44.9	41.7	45.6
CrowS-Pairs (0-shot)	74.0	72.9	73.5	75.9	68.4
GSM8K (5-shot)	50.2	47.0	52.6	22.7	56.6
MathQA (5-shot)	39.0	38.0	39.5	35.0	42.3
LogiQA2.0 (0-shot)	34.6	37.0	37.3	30.1	33.6
BBH CoT (0-shot)	64.6	58.5	66.7	41.2	62.0
			Code Benchmarks †		
HumanEval (pass@1)	32.0	22.8	30.0	0.00	35.4*
HumanEval (pass@10)	48.2	36.0	42.1	0.00	-
MBPP (pass@1)	25.7	21.5	21.2	3.20	42.1
MBPP (pass@10)	51.0	34.8	44.4	18.2	55.0
			Domain Specific (Medical)		
MedQA (0-shot)	53.7	46.2	56.2	40.8	58.4
MedMCQA (5-shot)	56.0	46.9	51.8	41.9	**56.1**
PubMedQA (0-shot)	78.6	76.4	74.4	76.0	74.2
			Overall Average Score		
Avg Score	57.20	53.77	57.11	45.87	-

Table 2: Evaluation results of 21 benchmark tasks for K2. We follow common settings for most of the evaluation metrics. We conduct MMLU with 0-shot for faster evaluation. The scores for the referenced models are evaluated with our evaluation code. We compare K2 with models trained with similar architectures and scales, including LLaMa-65B, Llama2-70B (Touvron et al., 2023a,b), Falcon-40B and Falcon-180B (Almazrouei et al., 2023).
† Coding evaluation scores are sensitive to detailed settings. For the Falcon series, we currently adopt the scores reported by Almazrouei et al. (2023), marked by *. We omit the ones (-) that have potential discrepancies between our numbers and previously reported ones.

Base Model Performance. The performance of the final K2 model is shown in Table 2. Compared with models of similar architecture and token sizes, K2 exhibits strong performance across the board, especially in generation benchmarks, such as coding tasks. Notably, K2 remains competitive with other models even if the model size is smaller, or trained on less tokens.

Model Size	K2-Chat 65B	Qwen1.5-Chat 72B	DeepSeek-Chat 67B	Llama2-Chat 70B	Llama3-Instruct 70B
			Natural Language Benchmarks		
MMLU (0-shot)	63.5	76.9	72.0	61.1	78.6
RACE (0-shot)	46.1	38.1	46.3	44.0	47.0
HellaSwag (10-shot)	81.7	86.3	87.0	85.9	85.6
PIQA (5-shot)	82.3	82.4	85.8	81.8	85.0
ARC-easy (5-shot)	84.6	87.1	89.9	85.5	89.8
ARC-challenge (25-shot)	61.3	67.7	68.5	65.3	72.0
OpenBookQA (5-shot)	48.0	47.8	52.2	47.2	55.2
Winogrande (5-shot)	79.5	80.2	85.7	75.1	76.1
TruthfulQA (0-shot)	44.7	63.9	55.9	52.8	61.9
CrowS-Pairs (0-shot)	64.2	65.7	73.9	71.9	71.1
GSM8K (5-shot)	60.7	30.6	47.0	48.4	91.2
MathQA (5-shot)	44.8	49.6	44.2	38.0	67.4
LogiQA2.0 (0-shot)	38.0	39.8	42.7	37.7	41.5
BBH CoT (0-shot)	64.9	29.4	73.7	63.0	45.6
			Code Benchmarks		
HumanEval (pass@1)	47.9	40.4	59.0	30.7	41.8
HumanEval (pass@10)	64.6	54.3	75.0	41.5	56.7
MBPP (pass@1)	48.4	51.1	58.2	31.4	18.9
MBPP (pass@10)	60.0	61.2	70.6	39.2	36.4
			Domain Specific (Medical)		
MedQA (0-shot)	53.6	65.2	61.4	50.0	76.4
MedMCQA (5-shot)	51.3	62.7	56.7	44.8	71.0
PubMedQA (0-shot)	75.0	79.2	79.0	76.8	79.6
			Overall Average Score		
Avg Score	60.24	59.98	65.93	55.81	64.22

Table 3: Evaluation results of 21 benchmark tasks for the finetuned K2, with the same settings for K2 base models. We compare the model with more recent instruction tuned models: Qwen1.5 (Bai et al., 2023), DeepSeek (DeepSeek-AI, 2024), Llama2 (Touvron et al., 2023b) and Llama3 (Meta AI, 2024). The number of training tokens for some of the models are not publicly available.

In Figure 4 we show the model's performance developed over the course of training. Compared with our prior experiments such as Amber (Liu et al., 2023b), we find that large models show performance improvement in very early stage. For example, the evaluation scores of MMLU for Amber and Olmo-7B (Groeneveld et al., 2024) both struggle to improve over the random baseline. In K2 we observed a sharp MMLU score improvement at the early stage of the training. This may be due to that larger models are much better at memorizing facts, contributing to the high scores of MMLU. Furthermore, we found that some metrics, such as OpenBookQA, fluctuate over the course of training, which may indicate that they are not suitable to be used as indicators of model performance.

Finetuned Models Performance. Comparing the performance of the finetuned models can be tricky, due to the different settings of finetuning and the potential of data leakage. Nevertheless, we compare the finetuned K2-Chat model with other more recent models in Table 3 and K2-Chat still performs relatively well. K2-Chat scores lower than Llama3-Instruct on many benchmarks, but K2 is only trained of 1.4 trillion tokens, as comparison to Llama3's 15 trillion pretraining tokens.

5 Power Consumption and Carbon Footprint

It has become a widespread concern regarding the power consumption and carbon emissions of LLM pre-training (and development in general), as well as its impact on the environment, as studied in previous literature (Strubell et al., 2019; Patterson et al., 2021; Wu et al., 2022; Dodge et al., 2022; Touvron et al., 2023a; Groeneveld et al., 2024). We estimate the total energy consumed and carbon released while pre-training K2 by calculating the total power consumption required for training and then multiplying it by the carbon emission intensity of the power grid where the model was trained.

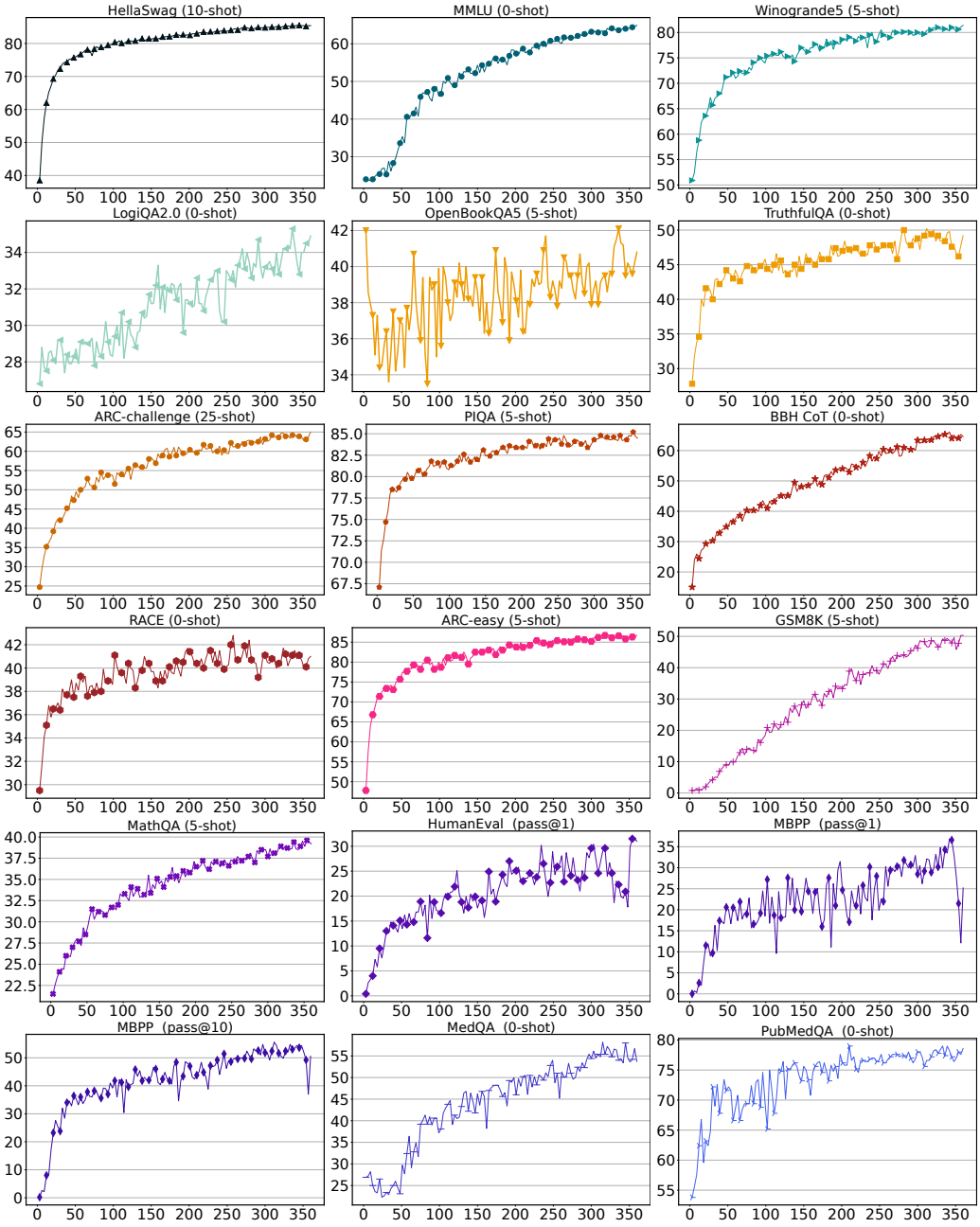

Figure 4: Evaluation metrics for K2 during the full pre-training process. The x-axis indicates the K2 checkpoint (120 plotted) during pretraining, and the y-axis indicates the evaluation score. Several metrics grows smoothly. Metrics like MBPP fluctuates significantly during traing. In fact, our final checkpoint scores 10 points lower than the highest score.

We would like to note that we only aim to provide a rough estimation of power consumption and carbon footprint for K2 pre-training.

During K2 pre-training, each A100 GPU consumes around 0.34 kW consistently. We used 480 A100s, and the entire pre-training lasted for around 100 days. Following the power consumption calculation proposed in LLaMA (Touvron et al., 2023a), *i.e.,*

$$\text{Wh} = \text{GPU-hours} \times (\text{GPU power consumption}) \times \text{PUE},$$

we set the Power Usage Effectiveness (PUE) at 1.1 following (Touvron et al., 2023a; Groeneveld et al., 2024). Thus, overall, **the power consumption of K2 base model pre-training is 430.8**

MWh. We used 30 extra days to deal with the loss spikes, which caused an additional 129.3 MWh of power consumption. The fine-tuning process of K2 used 240 A100s for five days. Thus, the power consumption at this stage is 10.8 MWh.

For carbon footprint, we follow (Touvron et al., 2023a), *i.e.,*

$$\text{tCO}_2\text{eq} = \text{MWh} \times 0.385.$$

Thus, K2 pre-training, loss spike overheads, and fine-tuning cost 165.9, 49.8, 4.2 tCO_2eq respectively.

Pretraining a LLM from scratch creates a large amount of power consumption and carbon emissions. Yet we believe our fully open source approach can reduce unnecessary repeated work and allow future scientific research of LLMs to be carried out with much smaller environmental impact.

6 Open Source Approach

The LLM360 team is dedicated to advancing the frontier of open source by actively collaborating with the community to explore and implement best practices.

Since our initial launch, we have gained valuable feedback from the community. In this release, we have reorganized numerous artifacts and designed improved methods to share information with the community. Additionally, we have further studied licenses and open-source requirements with the community, and seek to find appropriate definition of Open Source for Artificial Intelligence, with organizations such as the Open Source Initiative (OSI).

The mission of LLM360 aligns well with the community's values. For example, we are pleased to see that the current OSI Open Source AI definition includes the provision that open source systems need to allow the freedom to "study how the system works and inspect its components."[9] One of LLM360's goals is to further demystify large language models (LLMs) and we invite the community to join us in opening the black box.

6.1 License

LLM360 is committed to facilitating an open and collaborative environment for innovation. To ensure this, we have chosen to release our code and model weights under the Apache 2.0 license, without any additional clauses that restrict the use of the models' outputs.

We also release the exact data sequence used during training to simplify research and promote reproducibility. The K2 dataset is released under the Open Data Commons Attribution License (ODC-By), which governs the rights over the curated dataset, not the contents of the underlying data.

We understand the risks associated with open-source models. However, we believe that the final model we release does not add additional risks to the field, especially since there exist open-weight models, such as Llama 3, which offer better performance. Open-source releases of larger models will enable researchers to study the security and safety issues associated with models of this scale. We will continue to explore the right approach for open source and open science around LLMs.

7 Conclusion and Future Work

K2 is the first-ever fully open-source LLM at the scale of 65 billion parameters and is powerful in terms of its performance. We use the LLM360 approach to open the black box of LLM pretraining at a large model parameter scale. We envision that K2 can serve both as a strong base model for product development (*e.g.,* building chatbots, virtual assistants, programming assistants, etc.) with more flexibility, and also can help researchers dive deeper into LLM pretraining behavior and the learning dynamics at a larger model scale.

The LLM360 team is actively developing our next generation of LLMs at a variety of different scales while exploring novel model architectural designs, *e.g.,* mixture-of-expert (MoE) as well as efficient and scalable attention mechanisms (Fedus et al., 2022). We are committed to continually pushing the boundaries of LLMs through this open-source effort.

[9] https://hackmd.io/@opensourceinitiative/osaid-0-0-8

References

Ebtesam Almazrouei, Hamza Alobeidli, Abdulaziz Alshamsi, Alessandro Cappelli, Ruxandra Cojocaru, Mérouane Debbah, Étienne Goffinet, Daniel Hesslow, Julien Launay, Quentin Malartic, Daniele Mazzotta, Badreddine Noune, Baptiste Pannier, and Guilherme Penedo. The falcon series of open language models, 2023.

Aida Amini, Saadia Gabriel, Peter Lin, Rik Koncel-Kedziorski, Yejin Choi, and Hannaneh Hajishirzi. Mathqa: Towards interpretable math word problem solving with operation-based formalisms, 2019.

Jacob Austin, Augustus Odena, Maxwell Nye, Maarten Bosma, Henryk Michalewski, David Dohan, Ellen Jiang, Carrie Cai, Michael Terry, Quoc Le, et al. Program synthesis with large language models. *arXiv preprint arXiv:2108.07732*, 2021.

Zhangir Azerbayev, Hailey Schoelkopf, Keiran Paster, Marco Dos Santos, Stephen McAleer, Albert Q. Jiang, Jia Deng, Stella Biderman, and Sean Welleck. Llemma: An open language model for mathematics, 2023.

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.

Mohammad Bavarian, Heewoo Jun, Nikolas Tezak, John Schulman, Christine McLeavey, Jerry Tworek, and Mark Chen. Efficient training of language models to fill in the middle. *arXiv preprint arXiv:2207.14255*, 2022.

Loubna Ben Allal, Niklas Muennighoff, Logesh Kumar Umapathi, Ben Lipkin, and Leandro von Werra. A framework for the evaluation of code generation models. https://github.com/bigcode-project/bigcode-evaluation-harness, 2022.

Stella Biderman, Hailey Schoelkopf, Quentin Gregory Anthony, Herbie Bradley, Kyle O'Brien, Eric Hallahan, Mohammad Aflah Khan, Shivanshu Purohit, USVSN Sai Prashanth, Edward Raff, et al. Pythia: A suite for analyzing large language models across training and scaling. In *International Conference on Machine Learning*, pp. 2397–2430. PMLR, 2023.

Stella Biderman, Hailey Schoelkopf, Lintang Sutawika, Leo Gao, Jonathan Tow, Baber Abbasi, Alham Fikri Aji, Pawan Sasanka Ammanamanchi, Sidney Black, Jordan Clive, Anthony DiPofi, Julen Etxaniz, Benjamin Fattori, Jessica Zosa Forde, Charles Foster, Jeffrey Hsu, Mimansa Jaiswal, Wilson Y. Lee, Haonan Li, Charles Lovering, Niklas Muennighoff, Ellie Pavlick, Jason Phang, Aviya Skowron, Samson Tan, Xiangru Tang, Kevin A. Wang, Genta Indra Winata, François Yvon, and Andy Zou. Lessons from the trenches on reproducible evaluation of language models, 2024.

Yonatan Bisk, Rowan Zellers, Ronan Le Bras, Jianfeng Gao, and Yejin Choi. Piqa: Reasoning about physical commonsense in natural language. In *Thirty-Fourth AAAI Conference on Artificial Intelligence*, 2020.

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.

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. *arXiv:1803.05457v1*, 2018.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*, 2021.

Tri Dao. Flashattention-2: Faster attention with better parallelism and work partitioning. *arXiv preprint arXiv:2307.08691*, 2023.

DeepSeek-AI. Deepseek llm: Scaling open-source language models with longtermism. *arXiv preprint arXiv:2401.02954*, 2024. URL https://github.com/deepseek-ai/DeepSeek-LLM.

Jesse Dodge, Taylor Prewitt, Remi Tachet des Combes, Erika Odmark, Roy Schwartz, Emma Strubell, Alexandra Sasha Luccioni, Noah A Smith, Nicole DeCario, and Will Buchanan. Measuring the carbon intensity of ai in cloud instances. In *Proceedings of the 2022 ACM conference on fairness, accountability, and transparency*, pp. 1877–1894, 2022.

William Fedus, Barret Zoph, and Noam Shazeer. Switch transformers: Scaling to trillion parameter models with simple and efficient sparsity. *Journal of Machine Learning Research*, 23(120):1–39, 2022.

Leo Gao, Stella Biderman, Sid Black, Laurence Golding, Travis Hoppe, Charles Foster, Jason Phang, Horace He, Anish Thite, Noa Nabeshima, Shawn Presser, and Connor Leahy. The pile: An 800gb dataset of diverse text for language modeling, 2020.

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. URL https://zenodo.org/records/10256836.

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.

Peter Henderson*, Mark S. Krass*, Lucia Zheng, Neel Guha, Christopher D. Manning, Dan Jurafsky, and Daniel E. Ho. Pile of law: Learning responsible data filtering from the law and a 256gb open-source legal dataset, 2022. URL https://arxiv.org/abs/2207.00220.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. Measuring massive multitask language understanding. *Proceedings of the International Conference on Learning Representations (ICLR)*, 2021.

Di Jin, Eileen Pan, Nassim Oufattole, Wei-Hung Weng, Hanyi Fang, and Peter Szolovits. What disease does this patient have? a large-scale open domain question answering dataset from medical exams. *arXiv preprint arXiv:2009.13081*, 2020.

Qiao Jin, Bhuwan Dhingra, Zhengping Liu, William Cohen, and Xinghua Lu. Pubmedqa: A dataset for biomedical research question answering. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pp. 2567–2577, 2019.

Vijay Anand Korthikanti, Jared Casper, Sangkug Lym, Lawrence McAfee, Michael Andersch, Mohammad Shoeybi, and Bryan Catanzaro. Reducing activation recomputation in large transformer models. *Proceedings of Machine Learning and Systems*, 5, 2023.

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph E. Gonzalez, Hao Zhang, and Ion Stoica. Efficient memory management for large language model serving with pagedattention. In *Proceedings of the ACM SIGOPS 29th Symposium on Operating Systems Principles*, 2023.

Guokun Lai, Qizhe Xie, Hanxiao Liu, Yiming Yang, and Eduard Hovy. Race: Large-scale reading comprehension dataset from examinations. *arXiv preprint arXiv:1704.04683*, 2017.

Dacheng Li, Hongyi Wang, Eric Xing, and Hao Zhang. Amp: Automatically finding model parallel strategies with heterogeneity awareness. *Advances in Neural Information Processing Systems*, 35: 6630–6639, 2022.

Raymond Li, Loubna Ben Allal, Yangtian Zi, Niklas Muennighoff, Denis Kocetkov, Chenghao Mou, Marc Marone, Christopher Akiki, Jia Li, Jenny Chim, et al. Starcoder: may the source be with you! *arXiv preprint arXiv:2305.06161*, 2023.

Lizhi Lin, Honglin Mu, Zenan Zhai, Minghan Wang, Yuxia Wang, Renxi Wang, Junjie Gao, Yixuan Zhang, Wanxiang Che, Timothy Baldwin, Xudong Han, and Haonan Li. Against the achilles' heel: A survey on red teaming for generative models, 2024.

Stephanie Lin, Jacob Hilton, and Owain Evans. Truthfulqa: Measuring how models mimic human falsehoods, 2021.

Hanmeng Liu, Jian Liu, Leyang Cui, Zhiyang Teng, Nan Duan, Ming Zhou, and Yue Zhang. Logiqa 2.0—an improved dataset for logical reasoning in natural language understanding. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 31:2947–2962, 2023a. doi: 10.1109/TASLP.2023.3293046.

Zhengzhong Liu, Aurick Qiao, Willie Neiswanger, Hongyi Wang, Bowen Tan, Tianhua Tao, Junbo Li, Yuqi Wang, Suqi Sun, Omkar Pangarkar, et al. Llm360: Towards fully transparent open-source llms. *arXiv preprint arXiv:2312.06550*, 2023b.

Kyle Lo, Lucy Lu Wang, Mark Neumann, Rodney Michael Kinney, and Daniel S. Weld. S2orc: The semantic scholar open research corpus. In *Annual Meeting of the Association for Computational Linguistics*, 2020. URL https://api.semanticscholar.org/CorpusID:215416146.

Shayne Longpre, Le Hou, Tu Vu, Albert Webson, Hyung Won Chung, Yi Tay, Denny Zhou, Quoc V Le, Barret Zoph, Jason Wei, et al. The flan collection: Designing data and methods for effective instruction tuning. *arXiv preprint arXiv:2301.13688*, 2023.

Meta AI. Introducing meta llama 3: The most capable openly available llm to date, 2024. URL https://ai.meta.com/blog/meta-llama-3/.

Todor Mihaylov, Peter Clark, Tushar Khot, and Ashish Sabharwal. Can a suit of armor conduct electricity? a new dataset for open book question answering. In *EMNLP*, 2018.

Nikita Nangia, Clara Vania, Rasika Bhalerao, and Samuel R. Bowman. CrowS-Pairs: A Challenge Dataset for Measuring Social Biases in Masked Language Models. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing*, Online, November 2020. Association for Computational Linguistics.

Deepak Narayanan, Mohammad Shoeybi, Jared Casper, Patrick LeGresley, Mostofa Patwary, Vijay Korthikanti, Dmitri Vainbrand, Prethvi Kashinkunti, Julie Bernauer, Bryan Catanzaro, et al. Efficient large-scale language model training on gpu clusters using megatron-lm. In *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis*, pp. 1–15, 2021.

Ankit Pal, Logesh Kumar Umapathi, and Malaikannan Sankarasubbu. Medmcqa: A large-scale multi-subject multi-choice dataset for medical domain question answering. In Gerardo Flores, George H Chen, Tom Pollard, Joyce C Ho, and Tristan Naumann (eds.), *Proceedings of the Conference on Health, Inference, and Learning*, volume 174 of *Proceedings of Machine Learning Research*, pp. 248–260. PMLR, 07–08 Apr 2022. URL https://proceedings.mlr.press/v174/pal22a.html.

Keiran Paster, Marco Dos Santos, Zhangir Azerbayev, and Jimmy Ba. Openwebmath: An open dataset of high-quality mathematical web text, 2023.

David Patterson, Joseph Gonzalez, Quoc Le, Chen Liang, Lluis-Miquel Munguia, Daniel Rothchild, David So, Maud Texier, and Jeff Dean. Carbon emissions and large neural network training. *arXiv preprint arXiv:2104.10350*, 2021.

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.

Chen Qian, Jie Zhang, Wei Yao, Dongrui Liu, Zhenfei Yin, Yu Qiao, Yong Liu, and Jing Shao. Towards tracing trustworthiness dynamics: Revisiting pre-training period of large language models, 2024.

Keisuke Sakaguchi, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. Winogrande: an adversarial winograd schema challenge at scale. *Commun. ACM*, 64(9):99–106, aug 2021. ISSN 0001-0782. doi: 10.1145/3474381. URL https://doi.org/10.1145/3474381.

Yikang Shen, Zhen Guo, Tianle Cai, and Zengyi Qin. Jetmoe: Reaching llama2 performance with 0.1 m dollars. *arXiv preprint arXiv:2404.07413*, 2024.

Mohammad Shoeybi, Mostofa Patwary, Raul Puri, Patrick LeGresley, Jared Casper, and Bryan Catanzaro. Megatron-lm: Training multi-billion parameter language models using model parallelism. *arXiv preprint arXiv:1909.08053*, 2019.

AI Research Team Snowflake. Snowflake arctic: The best llm for enterprise ai - efficiently intelligent, truly open, 2024. URL https://www.snowflake.com/blog/arctic-open-efficient-foundation-language-models-snowflake/. Accessed on May 28, 2024.

Luca Soldaini and Kyle Lo. peS2o (Pretraining Efficiently on S2ORC) Dataset. Technical report, Allen Institute for AI, 2023. ODC-By, https://github.com/allenai/pes2o.

Luca Soldaini, Rodney Kinney, Akshita Bhagia, Dustin Schwenk, David Atkinson, Russell Authur, Ben Bogin, Khyathi Chandu, Jennifer Dumas, Yanai Elazar, Valentin Hofmann, Ananya Harsh Jha, Sachin Kumar, Li Lucy, Xinxi Lyu, Nathan Lambert, Ian Magnusson, Jacob Morrison, Niklas

Muennighoff, Aakanksha Naik, Crystal Nam, Matthew E. Peters, Abhilasha Ravichander, Kyle Richardson, Zejiang Shen, Emma Strubell, Nishant Subramani, Oyvind Tafjord, Pete Walsh, Luke Zettlemoyer, Noah A. Smith, Hannaneh Hajishirzi, Iz Beltagy, Dirk Groeneveld, Jesse Dodge, and Kyle Lo. Dolma: an open corpus of three trillion tokens for language model pretraining research, 2024.

Emma Strubell, Ananya Ganesh, and Andrew Mccallum. Energy and policy considerations for deep learning in nlp. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 3645–3650, 2019.

Mirac Suzgun, Nathan Scales, Nathanael Schärli, Sebastian Gehrmann, Yi Tay, Hyung Won Chung, Aakanksha Chowdhery, Quoc V. Le, Ed H. Chi, Denny Zhou, and Jason Wei. Challenging big-bench tasks and whether chain-of-thought can solve them, 2022.

Tianhua Tao, Junbo Li, Bowen Tan, Hongyi Wang, William Marshall, Bhargav M Kanakiya, Joel Hestness, Natalia Vassilieva, Zhiqiang Shen, Eric P. Xing, and Zhengzhong Liu, 2024. URL https://huggingface.co/LLM360/CrystalCoder.

Teknium. Openhermes 2.5: An open dataset of synthetic data for generalist llm assistants, 2023. URL https://huggingface.co/datasets/teknium/OpenHermes-2.5.

Together Computer. Redpajama: an open dataset for training large language models, October 2023. URL https://github.com/togethercomputer/RedPajama-Data.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023a.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023b.

Yuxia Wang, Haonan Li, Xudong Han, Preslav Nakov, and Timothy Baldwin. Do-not-answer: Evaluating safeguards in LLMs. In Yvette Graham and Matthew Purver (eds.), *Findings of the Association for Computational Linguistics: EACL 2024*, pp. 896–911, St. Julian's, Malta, March 2024. Association for Computational Linguistics. URL https://aclanthology.org/2024.findings-eacl.61.

Carole-Jean Wu, Ramya Raghavendra, Udit Gupta, Bilge Acun, Newsha Ardalani, Kiwan Maeng, Gloria Chang, Fiona Aga, Jinshi Huang, Charles Bai, et al. Sustainable ai: Environmental implications, challenges and opportunities. *Proceedings of Machine Learning and Systems*, 4: 795–813, 2022.

Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. Hellaswag: Can a machine really finish your sentence? In *Annual Meeting of the Association for Computational Linguistics*, 2019. URL https://api.semanticscholar.org/CorpusID:159041722.

Lianmin Zheng, Zhuohan Li, Hao Zhang, Yonghao Zhuang, Zhifeng Chen, Yanping Huang, Yida Wang, Yuanzhong Xu, Danyang Zhuo, Eric P Xing, et al. Alpa: Automating inter-and {Intra-Operator} parallelism for distributed deep learning. In *16th USENIX Symposium on Operating Systems Design and Implementation (OSDI 22)*, pp. 559–578, 2022.

Tianyu Zheng, Ge Zhang, Tianhao Shen, Xueling Liu, Bill Yuchen Lin, Jie Fu, Wenhu Chen, and Xiang Yue. Opencodeinterpreter: Integrating code generation with execution and refinement. *arXiv preprint arXiv:2402.14658*, 2024.

A Evaluation Details

The evaluation is done on an HPC cluster where each node has 4x Nvidia A100 80GB GPUs. We use lm-evaluation-harness (Gao et al., 2023) v0.4.0 as our evaluation framework, which is the latest version at the time. To speed up the evaluation, we use bf16 precision with the vLLM (Kwon et al., 2023) inference technique for all models. Specifically, the hyperparameters are as shown in Table 4, the task specific parameters are detailed in Table 5.

Parameter	Value
tensor parallel size	4
data type	bf16
gpu memory utilization	0.8
data parallel size	1
batch size	auto

Table 4: Evaluation hyperparameters applied to all models and tasks.

Note that for the code generation tasks, the bigcode-evaluation-harness (Ben Allal et al., 2022) only supports models that can be loaded on a single GPU. Therefore, we use the lm-evaluation-harness framework (as specified above) to run the code generation tasks with a temperature of 0.2 and top_p of 0.95.

Task	N shot	Task type	Metric
MMLU (Hendrycks et al., 2021)	0	multiple-choice	acc,none
RACE (Lai et al., 2017)	0	multiple-choice	acc,none
HellaSwag (Zellers et al., 2019)	10	multiple-choice	acc_norm,none
PIQA (Bisk et al., 2020)	5	multiple-choice	acc_norm,none
ARC-easy (Clark et al., 2018)	5	multiple-choice	acc_norm,none
ARC-challenge (Clark et al., 2018)	25	multiple-choice	acc_norm,none
OpenBookQA (Mihaylov et al., 2018)	5	multiple-choice	acc_norm,none
Winogrande (Sakaguchi et al., 2021)	5	multiple-choice	acc,none
TruthfulQA (Lin et al., 2021)	0	multiple-choice	acc,none
CrowS-Pairs (Nangia et al., 2020)	0	multiple-choice	acc_norm,none
GSM8K (Cobbe et al., 2021)	5	generation	exact_match,get-answer
MathQA (Amini et al., 2019)	5	multiple-choice	acc_norm,none
LogiQA2.0 (Liu et al., 2023a)	0	multiple-choice	acc_norm,none
BBH CoT (Suzgun et al., 2022)	0	generation	exact_match,get-answer
HumanEval (Chen et al., 2021)	0	generation	pass@1 & pass@10
MBPP (Austin et al., 2021)	0	generation	pass@1 & pass@10
MedQA (Jin et al., 2020)	0	multiple-choice	acc_norm,none
MedMCQA (Pal et al., 2022)	5	multiple-choice	acc_norm,none
PubMedQA (Jin et al., 2019)	0	multiple-choice	acc,none

Table 5: Tasks and their settings for evaluation. Note that we conduct MMLU in a 0-shot setting for faster inference speed.

Responsible Research

The LLM360 project was created with the mission to open-source and share knowledge about large language models to foster transparency, trust, and collaborative research. While large language models have demonstrated promise in advancing numerous domains throughout commercial and academic settings, the technology is still relatively poorly understood. Due to the significant capital requirements to training and experimentation with LLMs, many learnings in the space happen behind closed doors. The lack of knowledge transfer has negative effects for the ecosystem on the whole, as advances are limited to small groups. To fully realize the potential that large language models can deliver, we believe that the core tenets of transparency, trust, and collaboration are paramount to the long term success of the field.

For each model released under LLM360, we will release the datasets, data preparation scripts, training code, numerous intermediate checkpoints, all evaluation and system logs, and complete analysis performed during training. We prioritize publicly available datasets such as The RedPajama (Together Computer, 2023) and Refined Web (Penedo et al., 2023) and existing architectures and conventions such as LLaMA (Touvron et al., 2023a) to make our resource relevant and easy to access. By providing the listed artifacts, we hope to promote the reproducibility for all our work to encourage additional research.

Datasets are expensive to curate and are a major competitive advantage for training performant models. By making all data available, our models are fully auditable. We provide clarity on all pretraining sources, the ethical manner in which data was sourced, and the actual data. Releasing checkpoints from the entire training process enables fine grained research into training dynamics (Qian et al., 2024) which would otherwise be restricted to those with the financial resources to pretrain models. We believe that the future should only be constrained by our creativity, not man-made hurdles, and hope that access to our artifacts motivates others to pursue their own creative research unhindered.

Ethical Use We openly release our scores for the K2 models on safety evaluations such as TruthfulQA. These scores educate users on the potential risks that using our models may introduce when generating text. Additionally, we gather our data from reputable sources and apply standard filtering to remove harmful data, as well as conducting red teaming (Lin et al., 2024) to our model, but we cannot guarantee the outputs of our models will be completely safe. All users should conduct their own testing before adopting our models.

K2 models are also trained with coding abilities. When using code generated from large language models, users should always review the output before submitting it into their codebase. Generated code may introduce issues such as insecure code which cannot be eliminated from the model. Users should perform their own safety testing and code reviews before deploying applications.

Risks and Mitigation Since there are many comparable and more performant open weight model released publicly, we believe the release of K2 does not add additional risks on the misuse of powerful base models. Nevertheless, we align the instruction tuned model K2-Chat by finetuning on datasets such as Do-Not-Answer (Wang et al., 2024), as well as culture knowledge of the United Arab Emirates.