Status Update of LLM Training in Japan

Multicore World XII
2025/2/19

Institute of SCIENCE TOKYO

Supercomputing Research Center
Rio Yokota

Training Japanese LLMs

Fugaku-LLM

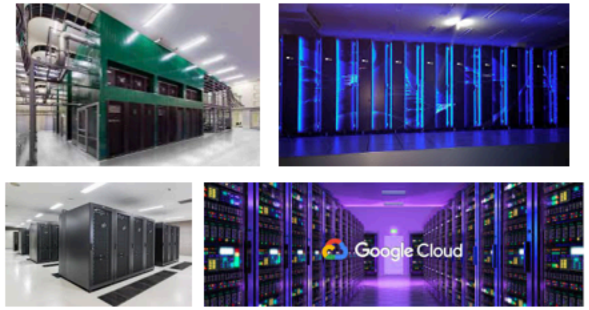
Members:
Tokyo Tech., RIKEN, Fujitsu, CyberAgent
Tohoku U., Nagoya U. Kotoba Tech,

System:
Fugaku (50,000,000 A64FX hours)

Model:
GPT 13B

Framework:
Megatron-DeepSpeed

LLM-jp



Members:
NII,++

System:
MDX (600,000 A100 hours)
ABCI (900,000 A100 hours)
GCP (? ,000,000 H100 hours)
TSUBAME4.0 (720,000 H100 hours)

Model:
GPT 1.3B, 13B, 175B
Llama2 172B

Framework:
Megatron-DeepSpeed, Megatron-LM

Swallow

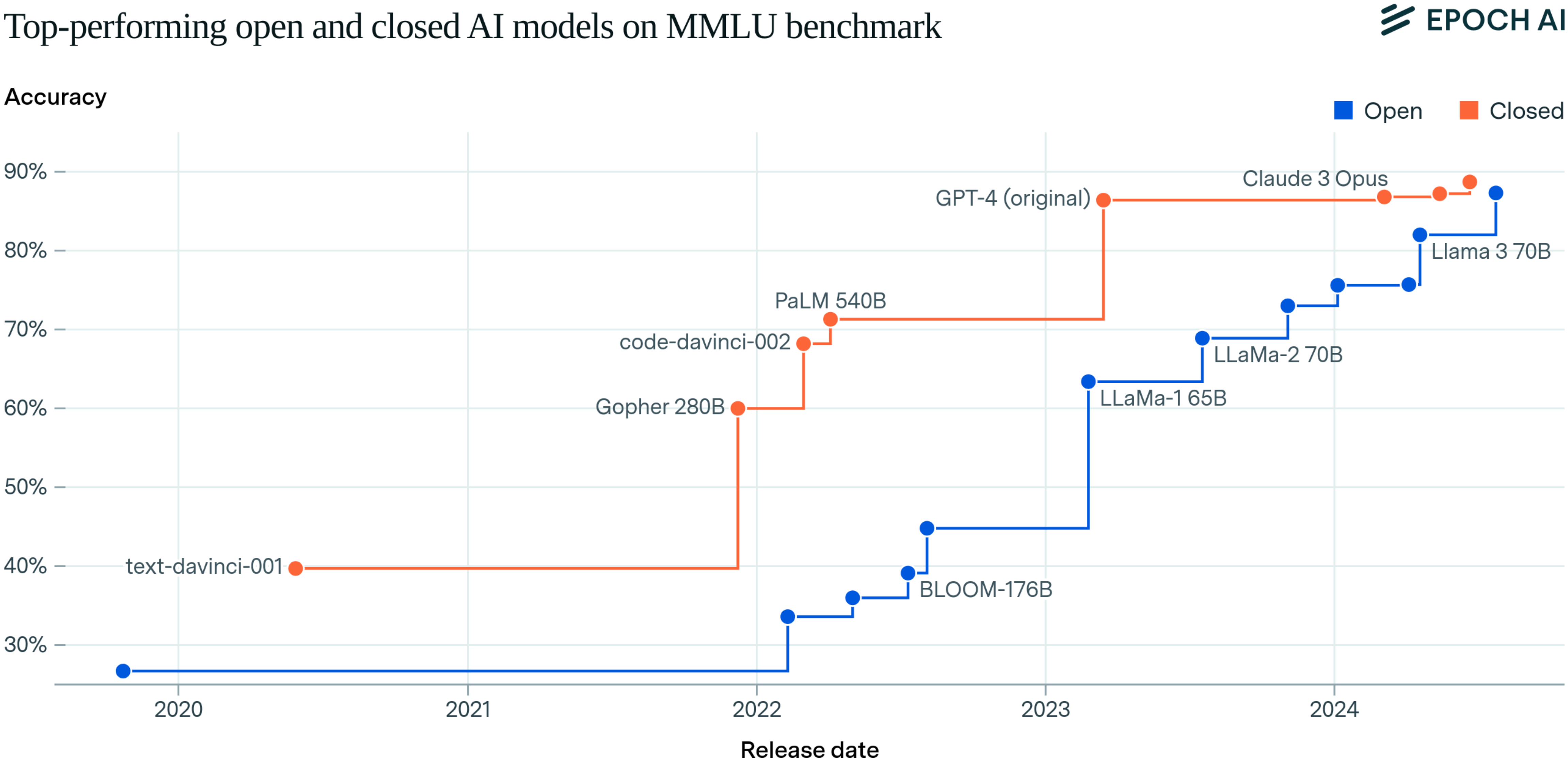
Members:
Tokyo Tech., AIST

System:
ABCI (350,000 A100 hours)

Model:
Llama2 7B, 13B, 70B
Mistral, Mixtral 7B
Llama3 8B, 70B

Framework:
Megatron-LM

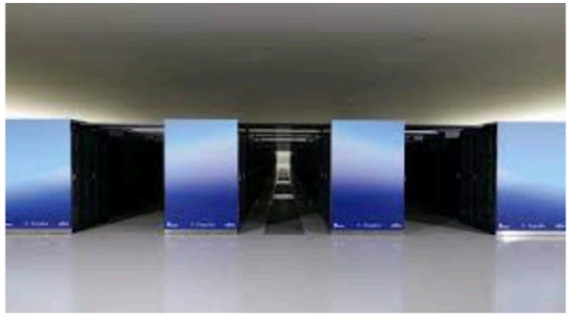
Closed Models vs Open Models



epoch.ai

Japanese LLMs

Fugaku-LLM



Members:
Tokyo Tech., RIKEN, Fujitsu, CyberAgent
Tohoku U., Nagoya U. Kotoba Tech,

System:
Fugaku (50,000,000 A64FX hours)

Model:
GPT 13B

Framework:
Megatron-DeepSpeed

LLM-jp

Members:
NII,++

System:
MDX (600,000 A100 hours)
ABCI (900,000 A100 hours)
GCP (? ,000,000 H100 hours)
TSUBAME4.0 (720,000 H100 hours)

Model:
GPT 1.3B, 13B, 175B
Llama2 172B

Framework:
Megatron-DeepSpeed, Megatron-LM

Swallow

Members:
Tokyo Tech., AIST

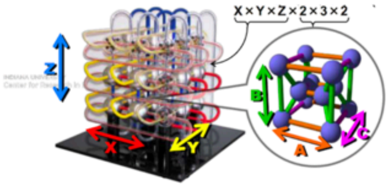
System:
ABCI (350,000 A100 hours)

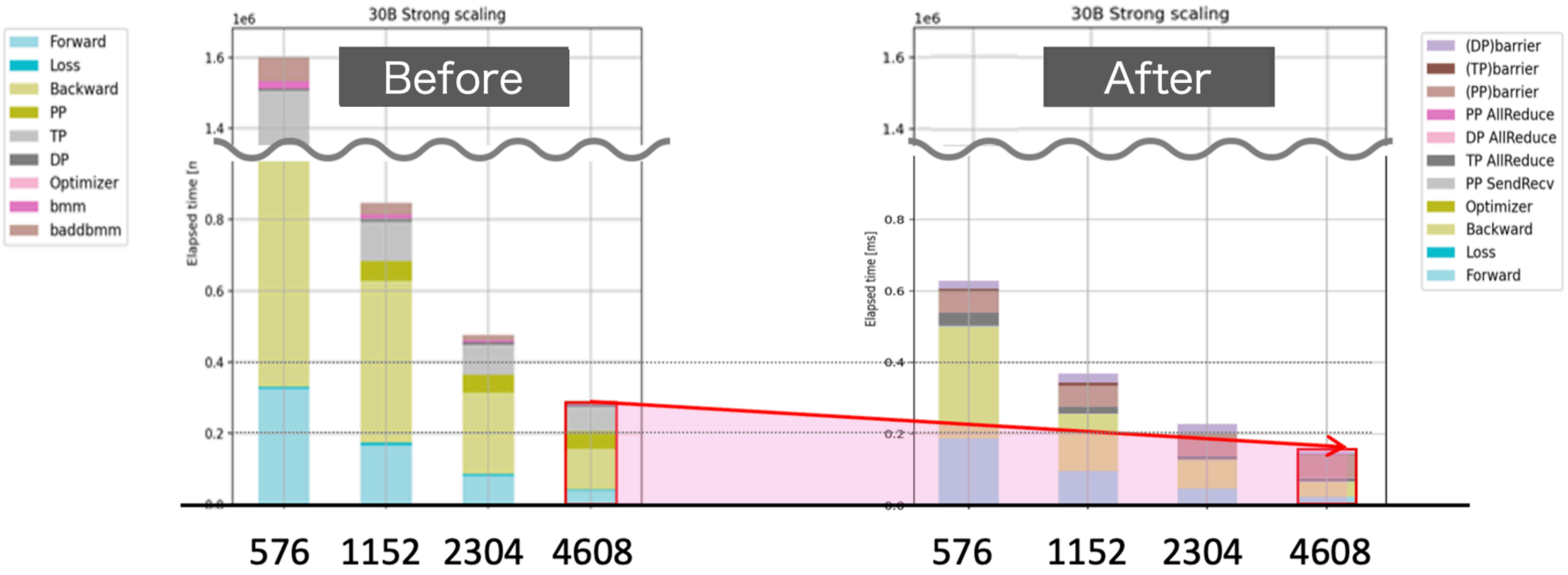
Model:
Llama2 7B, 13B, 70B
Mistral, Mixtral 7B
Llama3 8B, 70B

Framework:
Megatron-LM

Training LLMs on Fugaku (with A64fx CPU)

- We accelerated the GEMM operations 6x by optimizing batched operations for irregular sizes
- We accelerated the AllReduce communication 3x by using rank mapping and uTofu RDMA
- The resulting throughput was 1.0 TFLOP/s/node (theoretical peak is 6.8 TFLOP/s)
- Using 13,824 nodes of Fugaku, we achieved only around 14 PFLOP/s
- 32GB memory per node + global batch size limitations turns this into a strong scaling problem

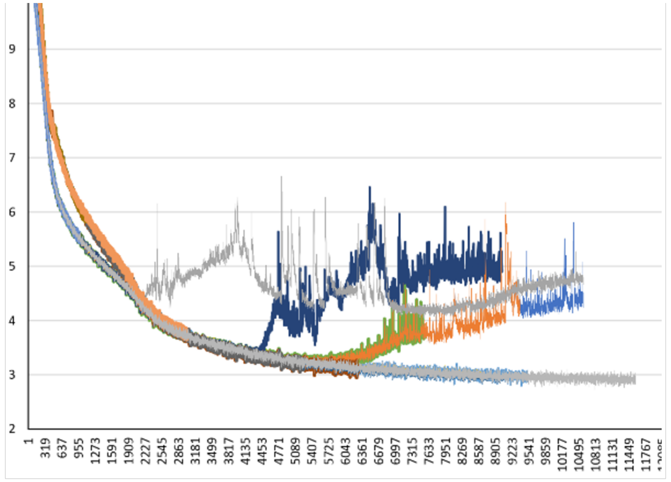




This is what happens if you use an original software stack

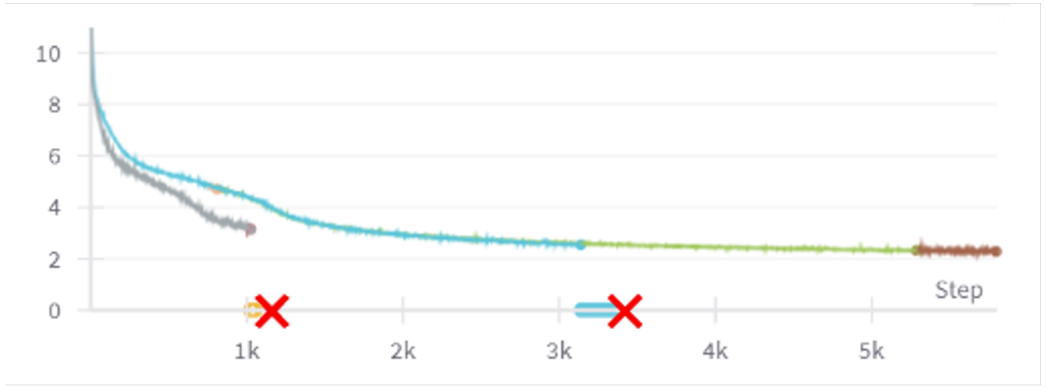
Initial port of Megatron-DeepSpeed to Fugaku caused instabilities

- This happened only when we used Tensor Parallelism
- This was caused by some random seeds going out of sync



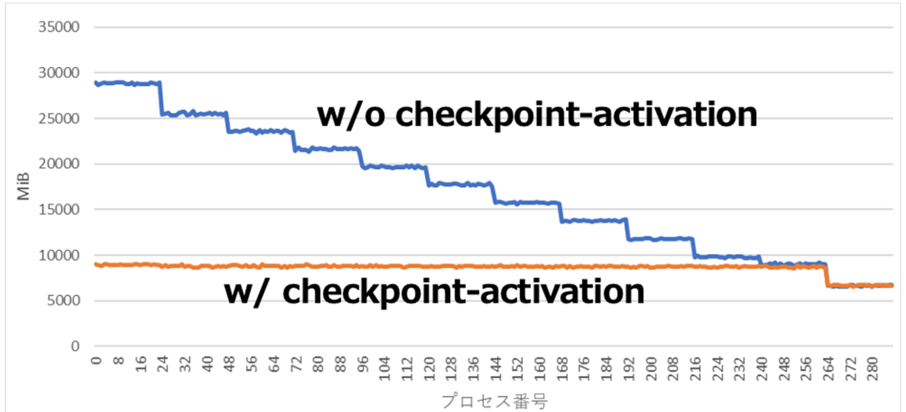
The loss suddenly became NaN

- This happened suddenly after many iterations
- This was caused by the custom tanh kernels we introduced



Abnormal memory consumption

- We initially observed abnormal about of memory usage
- This was solved by turning on the activation checkpointing



Japanese LLMs

Fugaku-LLM

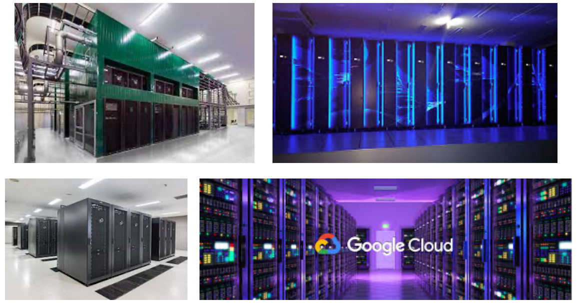
Members:
Tokyo Tech., RIKEN, Fujitsu, CyberAgent
Tohoku U., Nagoya U. Kotoba Tech,

System:
Fugaku (50,000,000 A64FX hours)

Model:
GPT 13B

Framework:
Megatron-DeepSpeed

LLM-jp



Members:
NII,++

System:
MDX (600,000 A100 hours)
ABCI (900,000 A100 hours)
GCP (？,000,000 H100 hours)
TSUBAME4.0 (720,000 H100 hours)

Model:
GPT 1.3B, 13B, 175B
Llama2 172B

Framework:
Megatron-DeepSpeed, Megatron-LM

Swallow

Members:
Tokyo Tech., AIST

System:
ABCI (350,000 A100 hours)

Model:
Llama2 7B, 13B, 70B
Mistral, Mixtral 7B
Llama3 8B, 70B

Framework:
Megatron-LM

LLM-JP

Japanese corpora:
Wikipedia: 1.4B tokens (1.3M documents)
mC4: 136B tokens (75M documents)
Common Crawl: 380B tokens (300M documents)
NDL WARP: 250B tokens (160M URL → 50M PDF → 39M documents)
(JST J-STAGE: 3B tokens)

English corpora:
Wikipedia: 5.1B tokens
Pile: 176B tokens
Stack: 148B tokens
SlimPajama: 627B tokens
RefinedWeb: 600B tokens
Dolma: 3T tokens
FineWeb: 15B tokens

Groups:
1. Data (crawling)
2. Data (cleaning)
3. Data (papers/books)
4. Architectures
5. Pre-training
6. Instruct/Fine-tuning
7. Evaluation
8. Safety

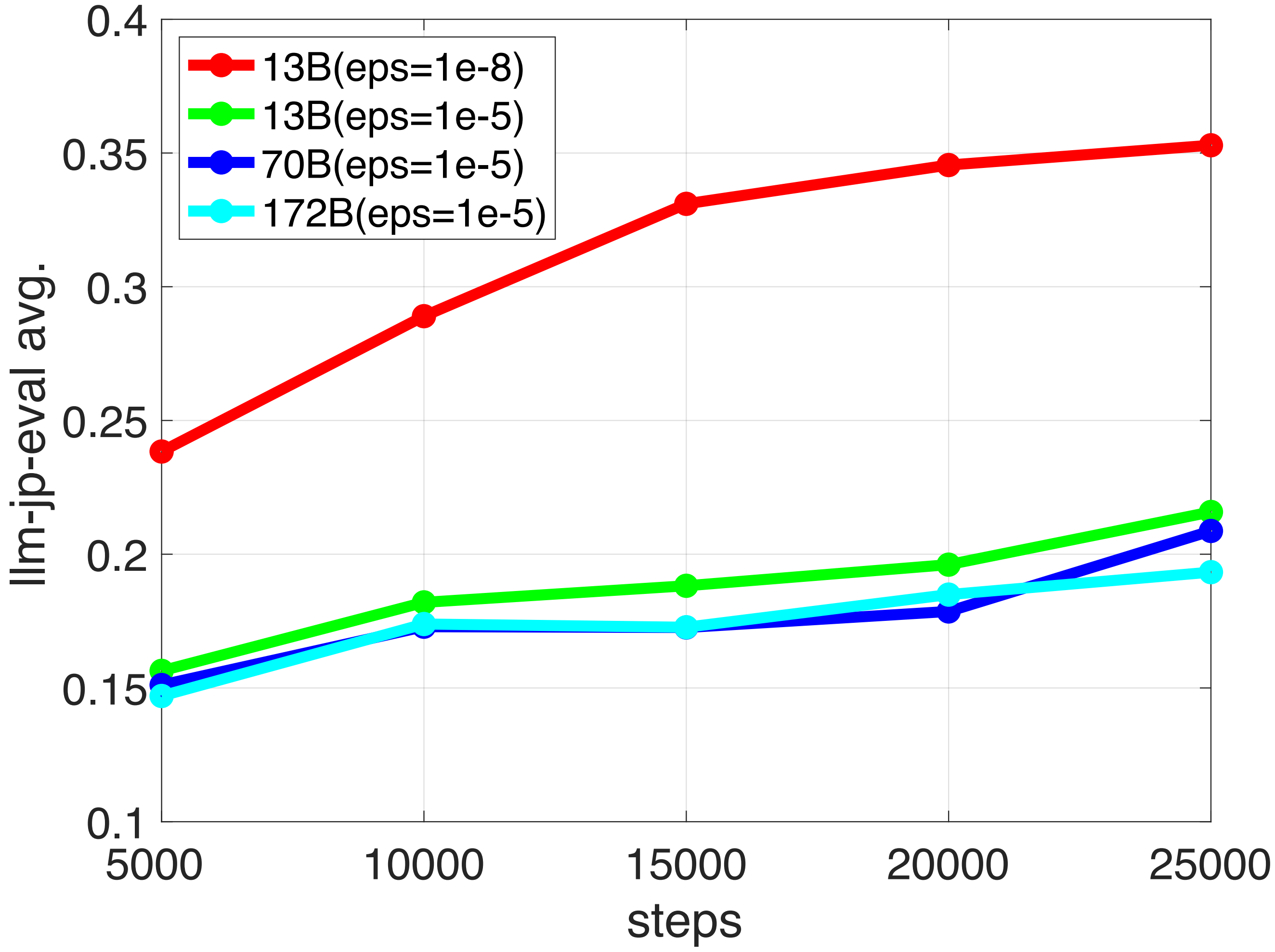
Computational resource:
MDX (600K A100 hours)
ABCI (900K A100 hours)
GCP (?M H100 hours)
TSUBAME4.0 (720K H100 hours)

Universities:
The University of Tokyo (Imaizumi, Ozeki, Kawahara, Tsuruoka, Baba, Matsuo, Miyao, Yanaka, Yoshinaga, Hanaoka, Kawazoe, Kodera, Taura), Tohoku University (Inui, Suzuki, Sakaguchi), Tokyo Institute of Technology (Okazaki, Arase, Yokota, Endo, Okumura), Waseda University (Kawahara), Ochanomizu University (Kobayashi), Nagoya University (Takeda, Sasano), Kyoto University (Kurohashi), Osaka University (Onizuka), Hokkaido University (Rafal), Tsukuba (Ochiai), Ochanomizu University (Kobayashi), Sophia University (Fukazawa), UEC (Yanai), Hitotsubashi University (Keyaki), Tokyo Metropolitan University (Hirasawa), Musashino University (Watanabe), Keio University (Ohara), Nara Institute of Science and Technology (Aramaki, Watanabe), Kyushu Institute of Technology (Okita), OIST (Yamada)

National Laboratories:
RIKEN AIP, RIKEN CCS, RIKEN GRP, AIST, NII, NICT, JST, JAMSTEC

Industry:
Microsoft Japan, AWS Japan, NVIDIA Japan, Intel, IBM Japan, Sakana AI, Stability AI Japan, SB Institutions, LINE/Yahoo, Sony, DeNA, Toshiba, Fujitsu, NTT, NTT Communications, KDDI, Toyota, Turing, Preferred Networks, Cyberagent, ELYZA, OmronScinicX, Studio Ousia, Precision, ZENKIGEN, Legalscape, Miraihonyaku, Megagon Labs, Stockmark, Matsuri Technologies, First Accounting, Baobab, Polaris.ai, Money Forward, Mercari, Asteras, Pasco, Rakuten, Lightblue, GMO, Advance Soft, Laboro.AI, Algomatic, Brainpad, IHI, Mizuho Bank, Retrieva, Fixstars, neoAI, and many more.

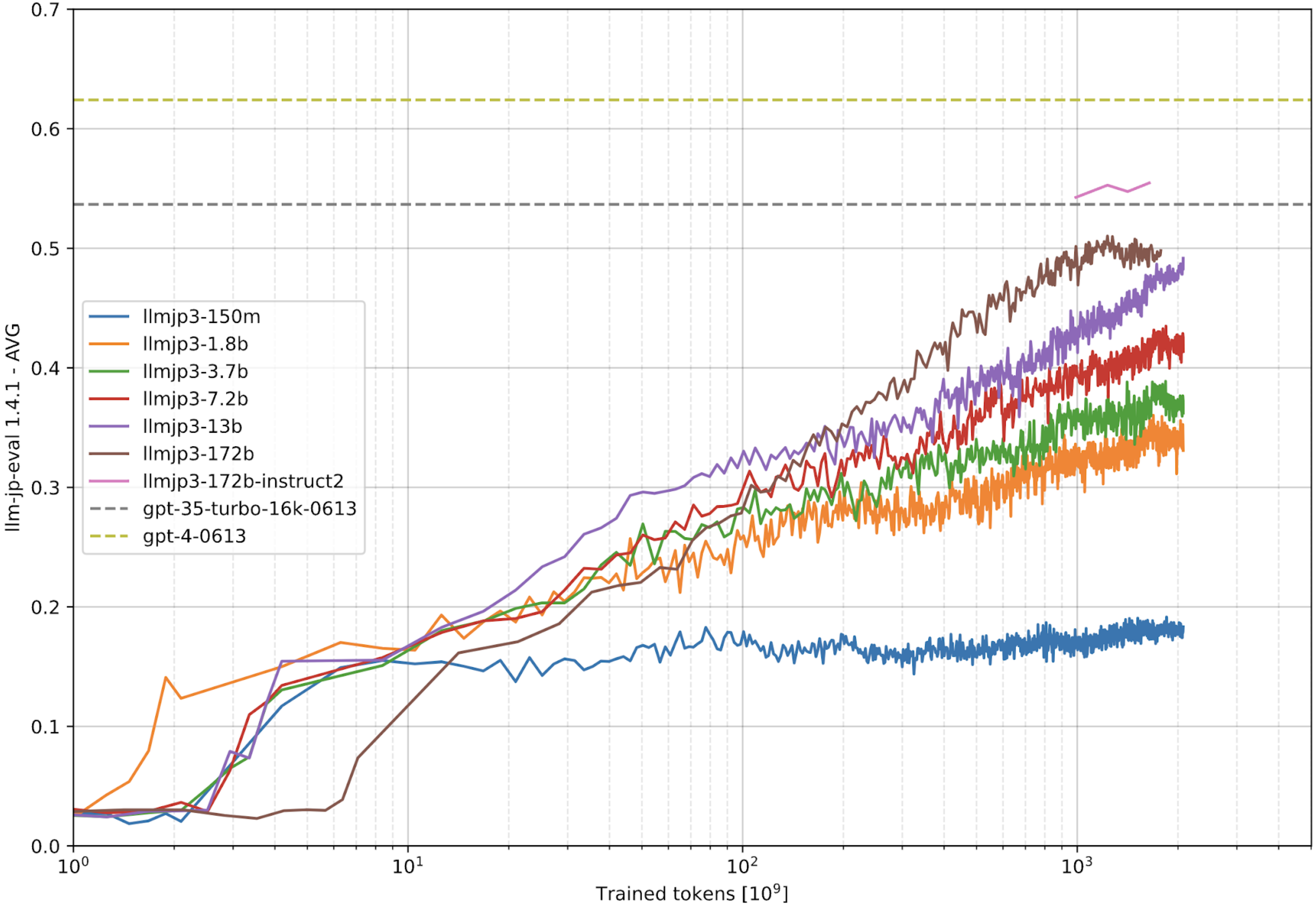
Adam eps=1e-5?



Things we changed:
GPT → Llama2
・pre-norm
・RMS norm
・scaled embedding
・z-loss
LR (minLR)：6e-5 (1e-6) → 1e-4 (1e-5)
LR warm up：3433 → 2000
Adam eps：1e-8 → 1e-5
Init. STD：0.005 → 0.02
Seq. length：2048 → 4096
Batch size：1536 → 1728

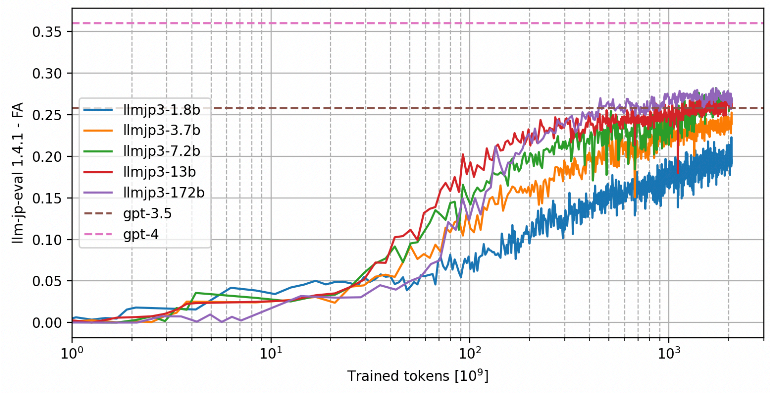
Hyperparameters. We trained using the AdamW optimizer (Loshchilov and Hutter, 2017), with $\beta_1 = 0.9$, $\beta_2 = 0.95$, eps $= 10^{-5}$. We use a cosine learning rate schedule, with warmup of 2000 steps, and decay final learning rate down to 10% of the peak learning rate. We use a weight decay of 0.1 and gradient clipping of 1.0. Figure 5 (a) shows the training loss for Llama 2 with these hyperparameters.

172B parameter run saturating?

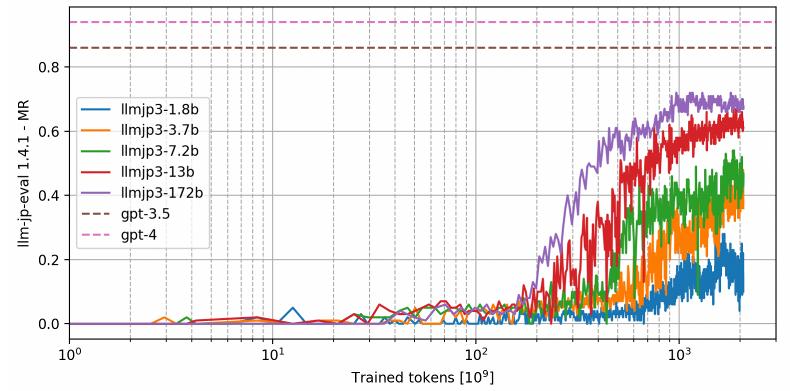


Looking at each task separately

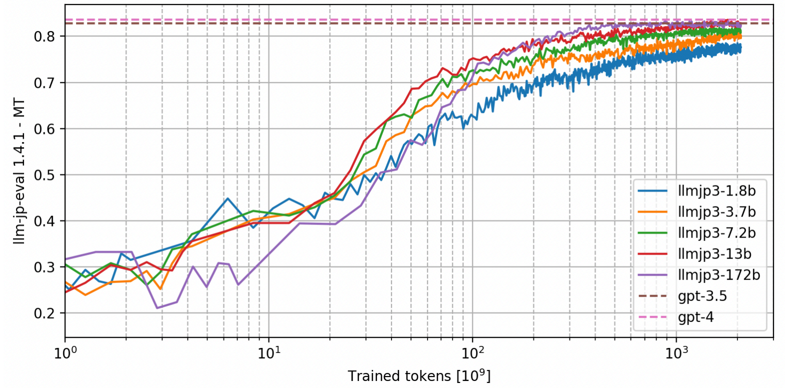
Fundamental Analysis



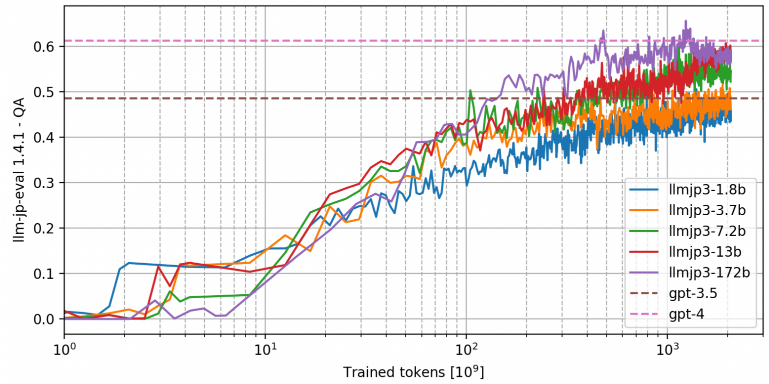
Mathematical Reasoning



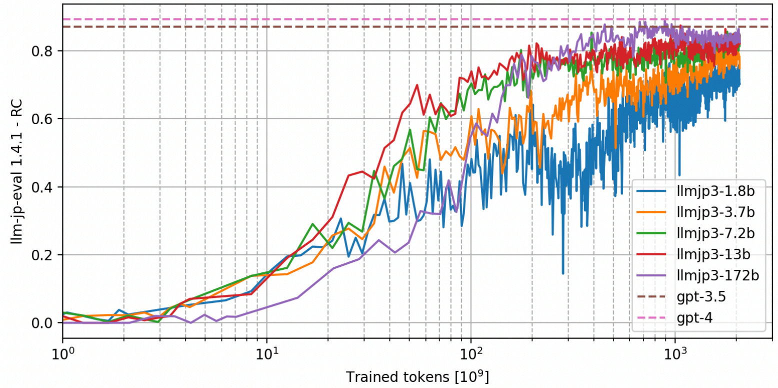
Machine Translation



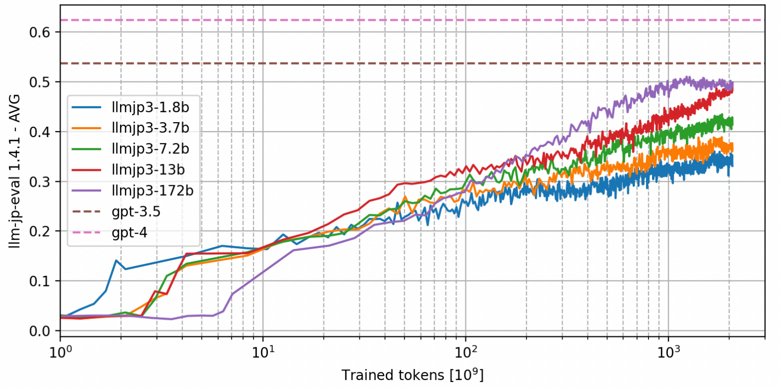
Question Answering



Reading Comprehension

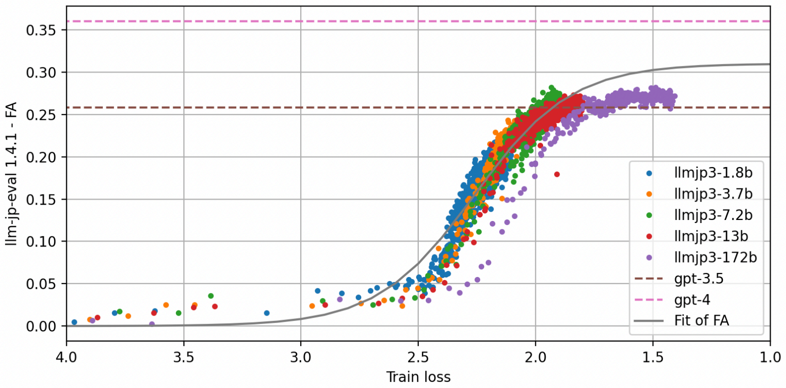


Average of 11 tasks

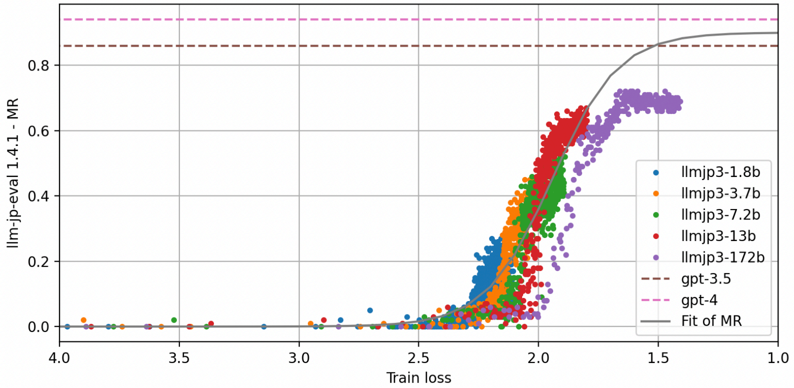


Looking at Loss vs Eval Score

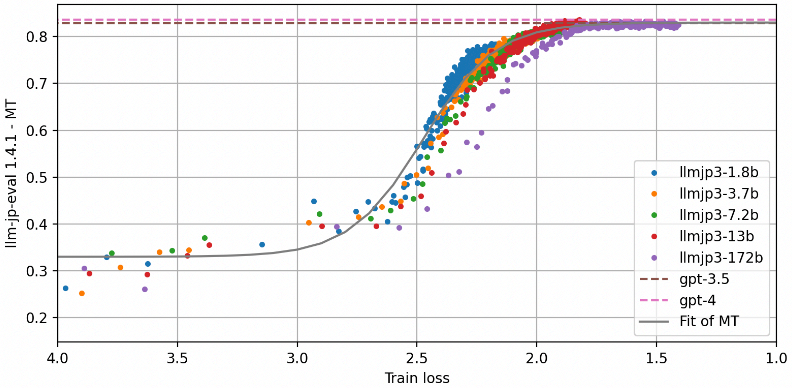
Fundamental Analysis



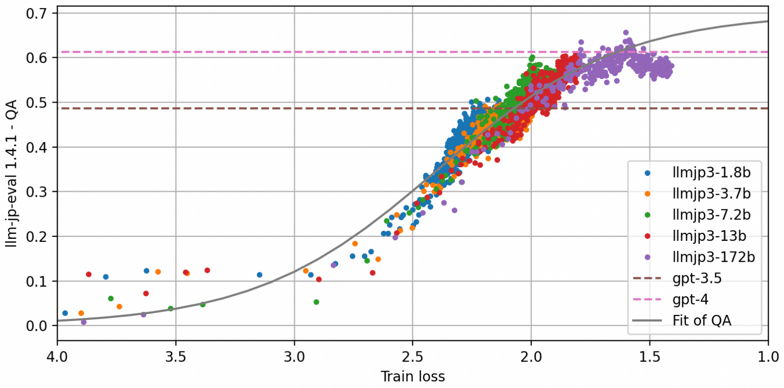
Mathematical Reasoning



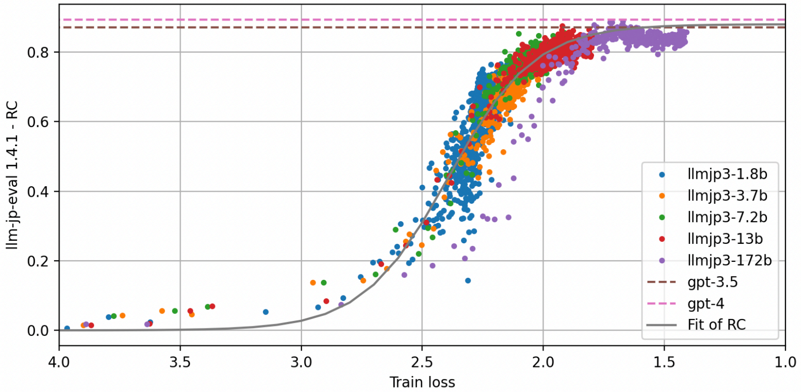
Machine Translation



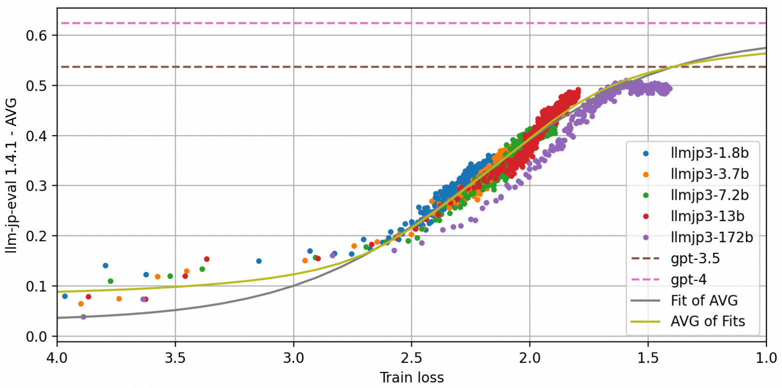
Question Answering



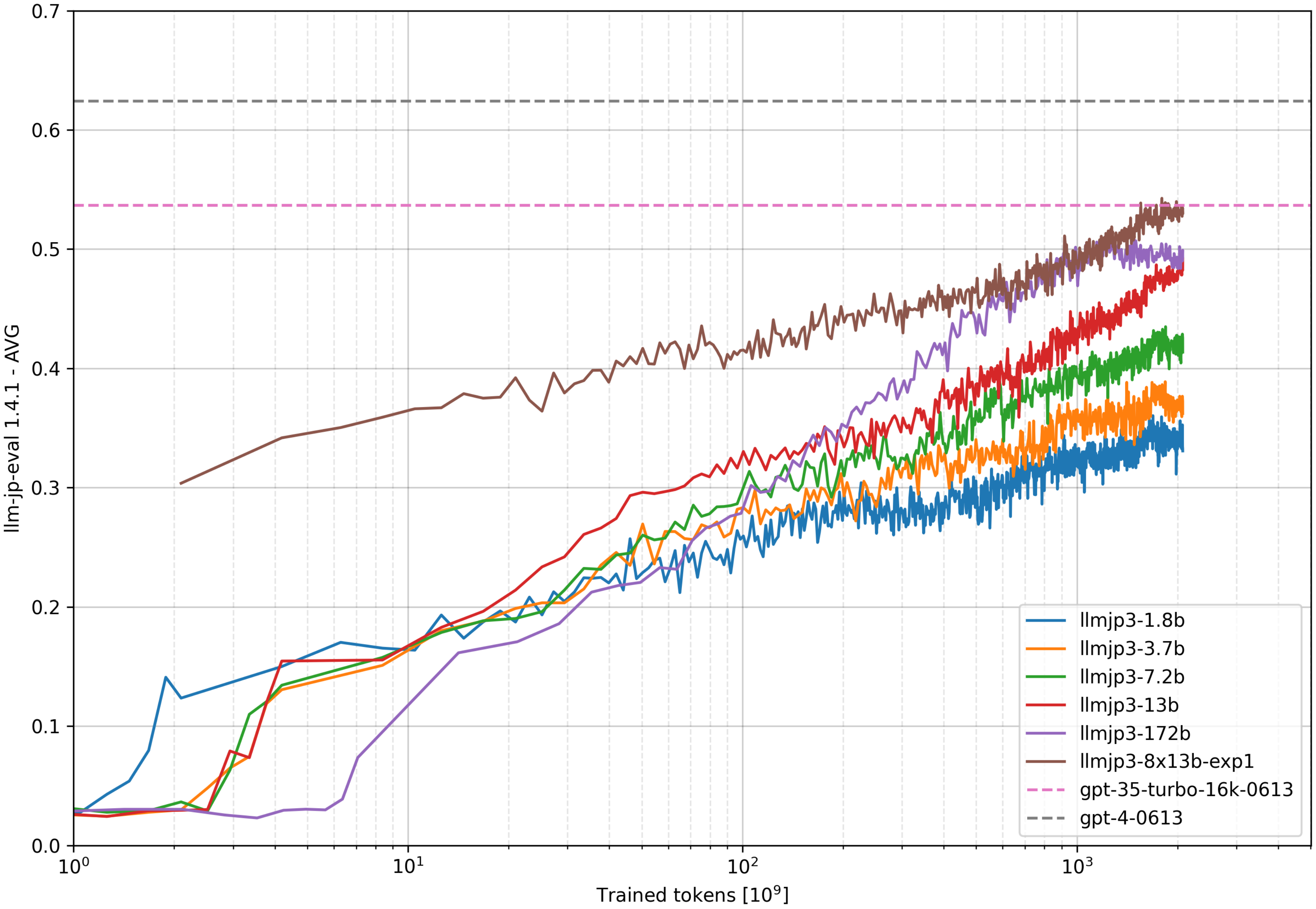
Reading Comprehension



Average of 11 tasks



8x13B MoE model did not saturate for some reason



Japanese LLMs

Fugaku-LLM

Members:
Tokyo Tech., RIKEN, Fujitsu, CyberAgent
Tohoku U., Nagoya U. Kotoba Tech,

System:
Fugaku (50,000,000 A64FX hours)

Model:
GPT 13B

Framework:
Megatron-DeepSpeed

LLM-jp

Members:
NII,++

System:
MDX (600,000 A100 hours)
ABCI (900,000 A100 hours)
GCP (？,000,000 H100 hours)
TSUBAME4.0 (720,000 H100 hours)

Model:
GPT 1.3B, 13B, 175B
Llama2 172B

Framework:
Megatron-DeepSpeed, Megatron-LM

Swallow

Members:
Tokyo Tech., AIST

System:
ABCI (350,000 A100 hours)

Model:
Llama2 7B, 13B, 70B
Mistral, Mixtral 7B
Llama3 8B, 70B

Framework:
Megatron-LM

Continual Pre-training vs Pre-training from Scratch

https://medium.com/@lars.chr.wiik/gpt-4o-vs-gpt-4-vs-gemini-1-5-performance-analysis-6bd207a2c580

Continual Pre-training

Advantages

- Leverages all the training data used to train the original model

Disadvantages

- Unclear what data the model was trained on

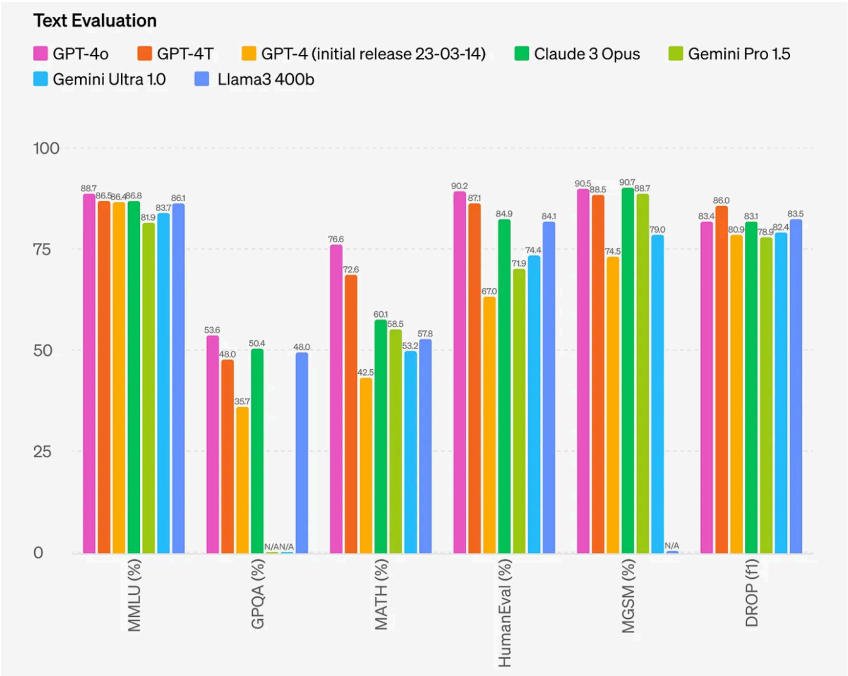
Pre-training from Scratch

Advantages

- Total control over what data the model is trained on

Disadvantages

- Need enormous data and computer resources



Llama3-400B is pretty competitive with GPT-4 on MMLU and DROP

Qwen2-72B is even better than Llama3-70B

	Qwen2-72B	Llama3-70B	Mixtral-8x22B
MMLU	84.2	79.5	77.8
MMLU-Pro	55.6	52.8	49.5
GPQA	37.9	36.3	34.3
TheoremQA	43.1	32.3	35.9
BBH	82.4	81.0	78.9
HumanEval	64.6	48.2	46.3
MBPP	76.9	70.4	71.7
MultiPL-E	59.6	46.3	46.7
GSM8K	89.5	83.0	83.7
MATH	51.1	42.5	41.7
C-Eval	91.0	65.2	54.6
CMMLU	90.1	67.2	53.4
Multi-Exam	76.6	70.0	63.5
Multi-Understanding	80.7	79.9	77.7
Multi-Mathematics	76.0	67.1	62.9

Open-source models will always be trailing not so far behind closed models

How to leverage these models and adapt them to novel languages / modalities is something worth investigating

https://www.reddit.com/r/singularity/comments/1d9mi13/new_alibabas_llm_qwen_2_72b_surpasses_llama_3_70b/

Continual Training on Japanese datasets

English

Characters not in the vocabulary are broken down into UTF-8 bytes, consuming as many as three tokens per character

Tokens	Characters
22	114

```
Characters not in the vocabulary are broken down into UTF-8 bytes,
 consuming as many as three tokens per character
```

Japanese

語彙に含まれない文字はUTF-8のバイト列に分解され1文字が3トークン程度も消費することとなる

Tokens	Characters
60	47

```
����に��まれない����はUTF-8のバイト��に�����され1����が3トークン�����も������す
ることとなる
```

Language	Tokens
English	1x
Japanese	3x
Chinese	3x
Korean	5x

Data filtering

Step 1	**Downloading WARC files**	63,352,266,406 pages in Common Crawl
Step 2	**Rapid Japanese detection**	This step reduces processing time for Steps 3 and 4
Step 3	**Text extraction**	Extract text from HTML (Trafilatura)
Step 4	**Precise Japanese detection**	2,686,080,919 Japanese pages extracted
Step 5	**Quality filtering**	Find high-quality text based on several rules
Step 6	**Deduplication**	Remove duplicated text (to avoid overfitting)
Step 7	**Filtering by hostnames**	Remove pages that may be unuseful to LLMs
Step 8	**Normalizing punctuations**	Normalize Japanese punctuations into “、” and “。”
Step 9	**Removing footersd**	Remove footers that were left at Step 3

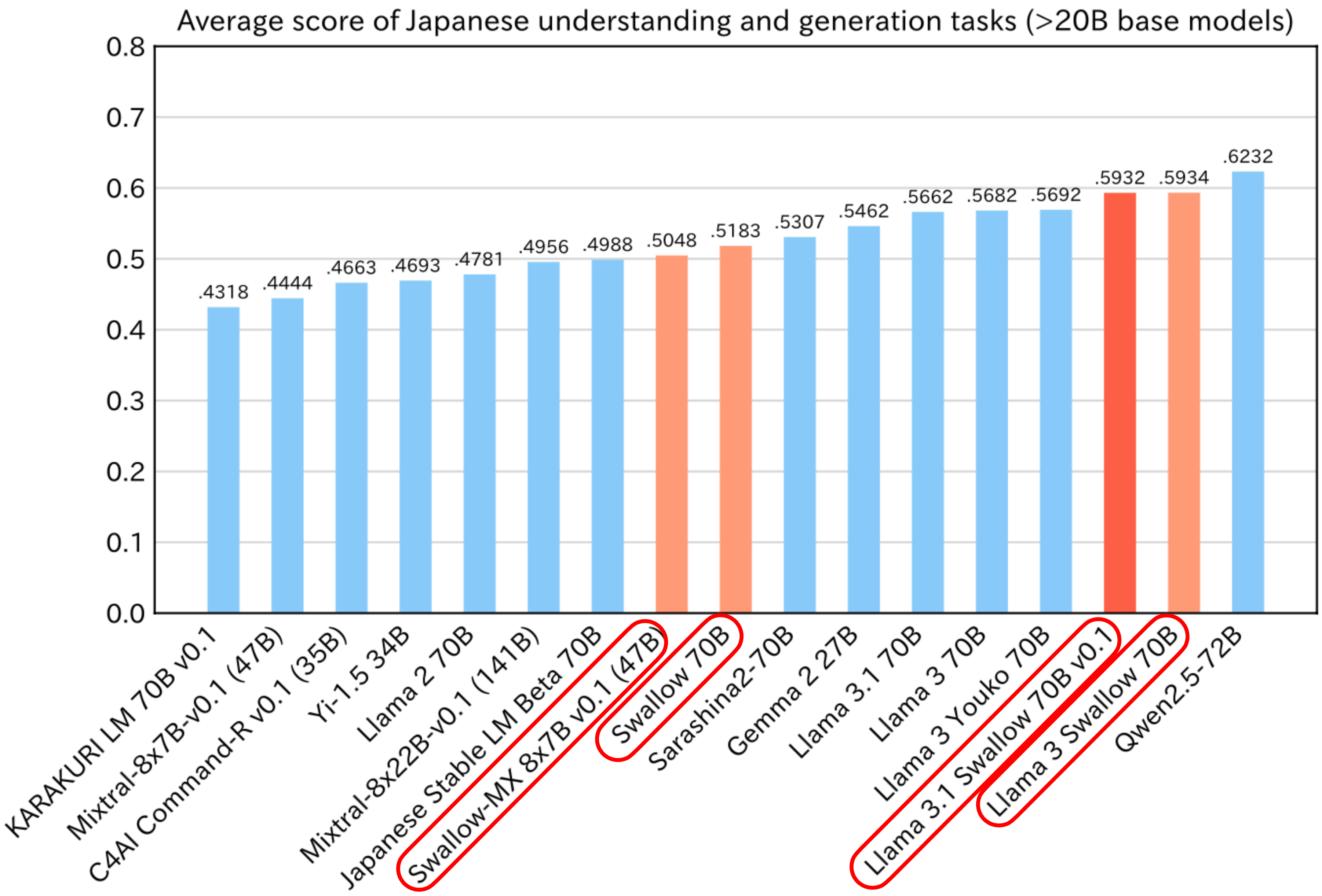
Swallow

・Continual pre-training from open models

・Create original Japanese dataset from CommonCrawl

・Currently #1 among LLMs trained in Japan

・Qwen2.5 is a little better than Swallow



Where do we go from here?

Fugaku-LLM

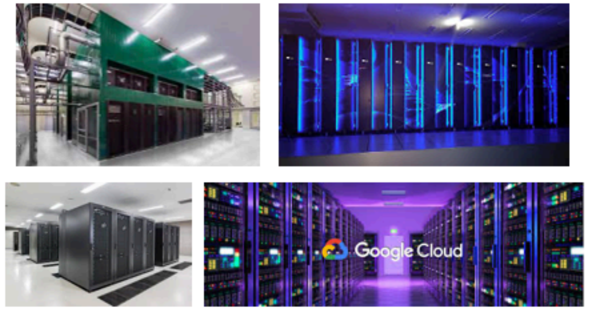
Members:
Tokyo Tech., RIKEN, Fujitsu, CyberAgent
Tohoku U., Nagoya U. Kotoba Tech,

System:
Fugaku (50,000,000 A64FX hours)

Model:
GPT 13B

Framework:
Megatron-DeepSpeed

LLM-jp



Members:
NII,++

System:
MDX (600,000 A100 hours)
ABCI (900,000 A100 hours)
GCP (? ,000,000 H100 hours)
TSUBAME4.0 (720,000 H100 hours)

Model:
GPT 1.3B, 13B, 175B
Llama2 172B

Framework:
Megatron-DeepSpeed, Megatron-LM

Swallow

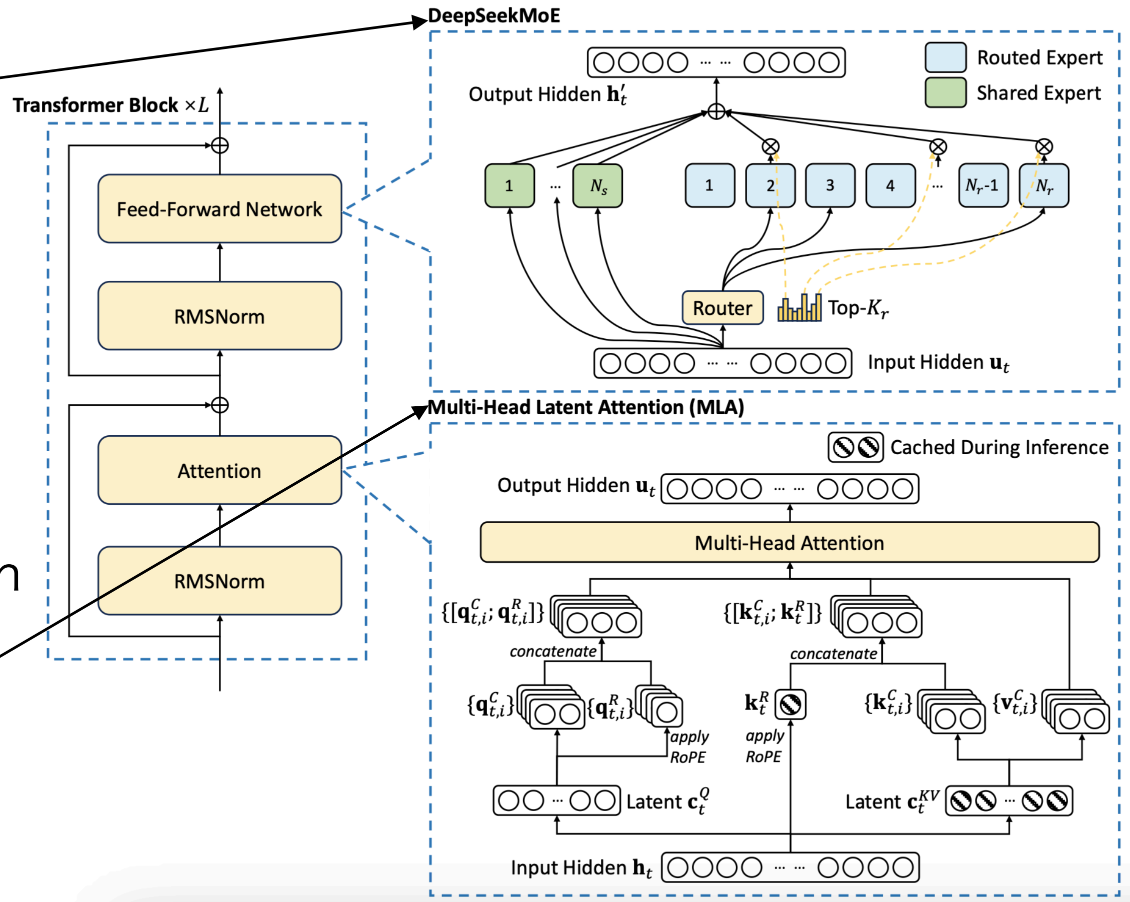
Members:
Tokyo Tech., AIST

System:
ABCI (350,000 A100 hours)

Model:
Llama2 7B, 13B, 70B
Mistral, Mixtral 7B
Llama3 8B, 70B

Framework:
Megatron-LM

HPC Aspects of DeepSeek



Mixture of Experts (MoE)

It is common to use 8 experts

→DeepSeek used 256 experts

671B parameters (37B active) 18x reduction in FLOPs

Making Expert Parallelism Scalable

Expert Parallelism (EP) results in an AlltoAll communication

Current Megatron-LM: 600 → 300 TFLOP/s/GPU

→Combine EP with Pipeline Parallelism (PP)

→Limit the number of SMs used for communication

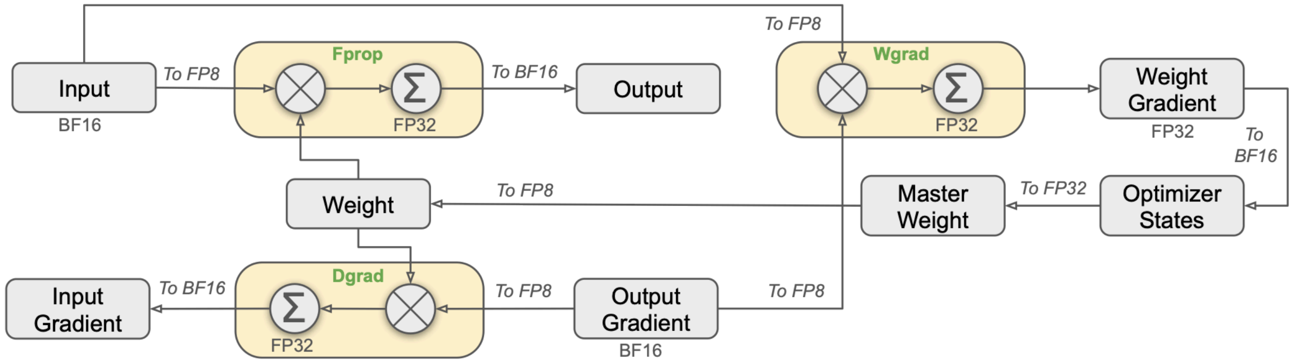
Multi-Head Latent Attention (MHLA)

60% reduction in FLOPs

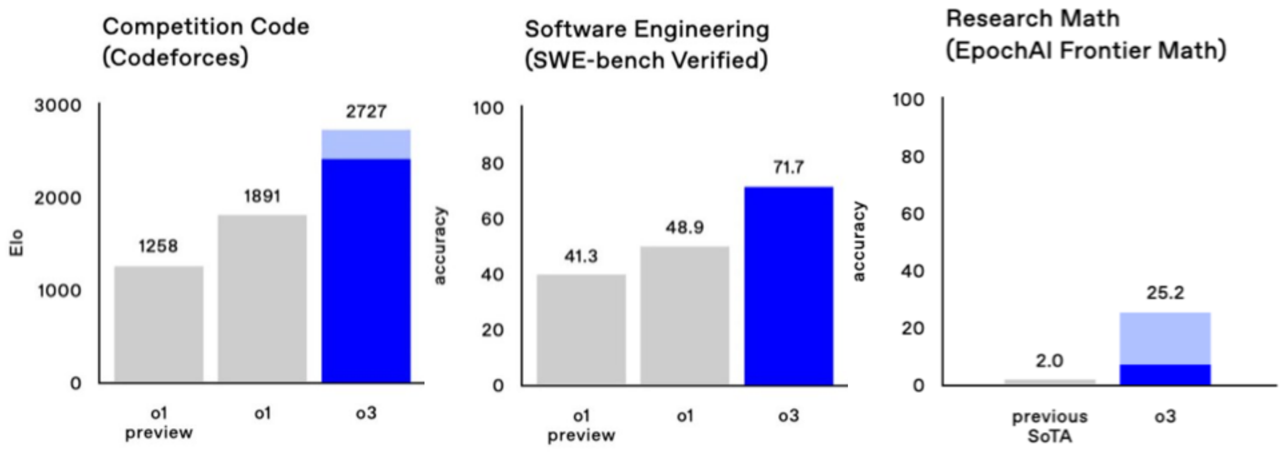
FP8 training on YOLO run

Using FP8 in the largest runs are still risky

→You don't get the theoretical 2x speedup



Things are advancing very fast



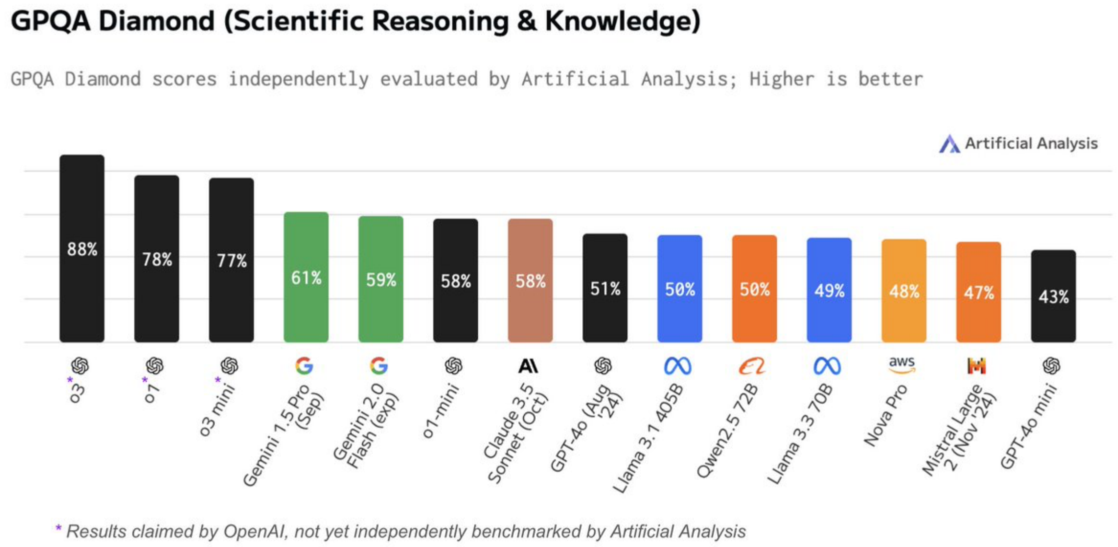
Prime field continuous extensions

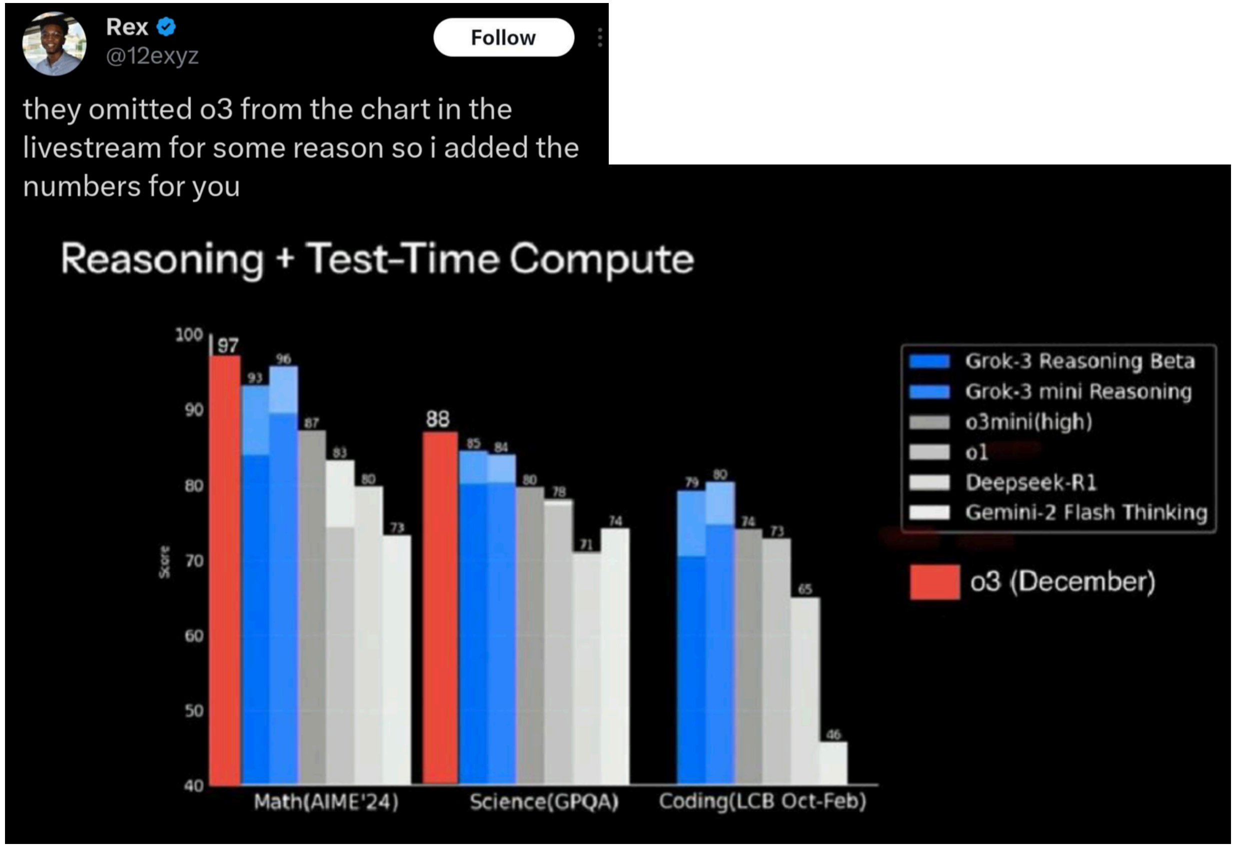
Problem Solution

Let a_n for $n \in \mathbb{Z}$ be the sequence of integers satisfying the recurrence formula

$$a_n = 198130309625\,a_{n-1} + 354973292077\,a_{n-2} - 427761277677\,a_{n-3} + 370639957\,a_{n-4}$$

with initial conditions $a_i = i$ for $0 \leq i \leq 3$. Find the smallest prime $p \equiv 4 \pmod{7}$ for which the function $\mathbb{Z} \to \mathbb{Z}$ given by $n \mapsto a_n$ can be extended to a continuous function on $\mathbb{Z}_p$.



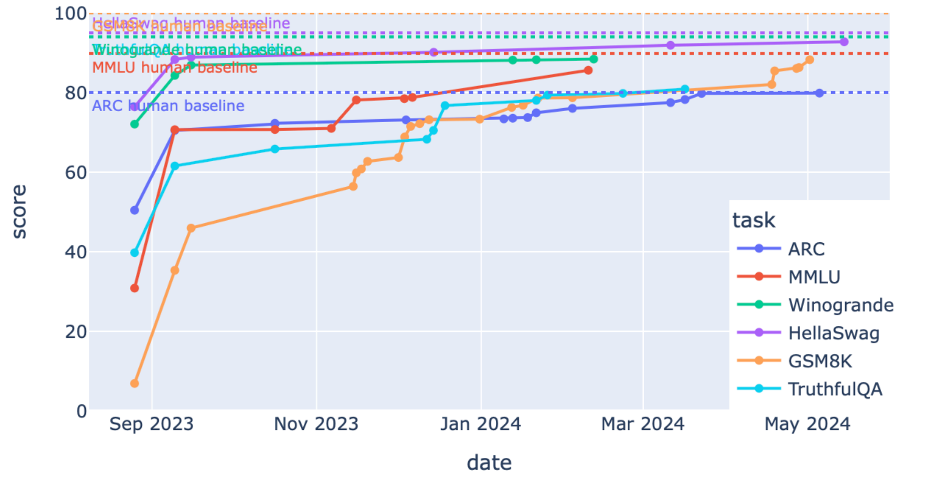


Implications of these advancements

Benchmarks should have an expiration date

We need more difficult benchmarks →

→Our models are only as good as our benchmarks



AI for Science can benefit from these reasoning models

Test-time scaling is crucial for “AI for Science”

→With the excellent math and coding skills on the horizon, giant leaps in science will become possible

→Don’t need data, only compute

Removing the restrictions of old deep learning

Classify → Segment/Search → Generate → Interact

Do you really need to pre-train on that data?

Pre-training → Fine-tuning → In-context learning → Deep Research

Change in the way AI interacts with data

Data goes into the AI model → AI agent goes into the data