



<https://samforeman.me/talks/hpc-user-forum/slides.html>



LLMs at ANL

<https://samforeman.me/talks/hpc-user-forum/slides.html>



Overview

- **AuroraGPT**: *General purpose scientific foundation model*
 - Model Sizes: {7B, 70B, ~ 1T}
- Broadly trained on general corpora of:
 - scientific text
 - structured data
- Multi-lingual: English, Japanese, French, German, Spanish, Italian
- Safe, with a focus on:
 - trustworthiness, safety, robustness, privacy, machine ethics
- Multi-modal:
 - images ■ sequences
 - tables ■ time-series
 - equations ■ graphs
 - proofs ■ fields

🔍 Details

- Llama Style Architecture:
 - trained using: [argonne-lcf/Megatron-DeepSpeed](#)
- Performant training implementations of Llama2 architecture for
 - Aurora, Polaris, Cerebras? and SambaNova?
- Workflow to capture:
 - snapshots, checkpoints
 - loss curves
 - scaling / performance data
- Training runs for AuroraGPT-7B
 - Baseline (Dolma) @ Aurora and @ Polaris (twins for debugging)
 - Baseline + SciencePUBs @ Aurora
- Trained raw models 🎉 delivered to post-pretraining team
 - AuroraGPT-7B-A, AuroraGPT-7B-P, AuroraGPT-7B-S
(A=[aurora](#), P=[Polaris](#), S=[science](#))



Why?

- **For Science!**
- Data-{sets, pipelines} for preparing scientific training data
- Infrastructure to {train, eval, deploy} LLMs for science
 - Comparative analysis across: {models, tasks, languages, contexts, ...}
- Augment text data from the web with:
 - full text papers
 - structured scientific data¹
- Safety-driven, publicly-visible, open-source approach:
 - Distribution of research grade artifacts (models, checkpoints, etc.)
 - International collaborations on AGI for science

1. Can be much more difficult than text (or even image) data

Teams: Who Does What?

1. Planning

2. Data Prep

- Accumulate 20+ T tokens of high-quality scientific text and structured data

3. Models / Training¹

- Train (entirely from scratch) a series of models on publicly available data

4. Evaluation

- Skills, trustworthiness, safety, robustness, privacy, machine ethics

1. Co-led by: Venkat Vishwanath, Sam Foreman

5. Post-Training

- Fine-tuning, alignment

6. Inference

- Model serving, API development / public-facing web services

7. Distribution

- Licensing, generating and distributing artifacts for public consumption

8. Communication



Aurora: System Overview

- 166 Racks
- 10,624 Nodes
- 21,248 CPUs
- 63,744 GPUs
- 84,992 NICs
- 8 PB HBM
- 10 PB DDR5c





Large Language Models

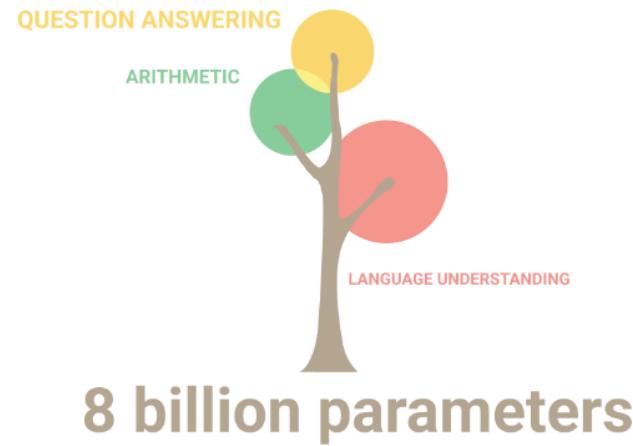


Figure 1: Large Language Models have (LLM)s have taken the NLP community **world** by storm¹.

1. [Hannibal046/Awesome-LLM](https://github.com/Hannibal046/Awesome-LLM)

Training LLMs

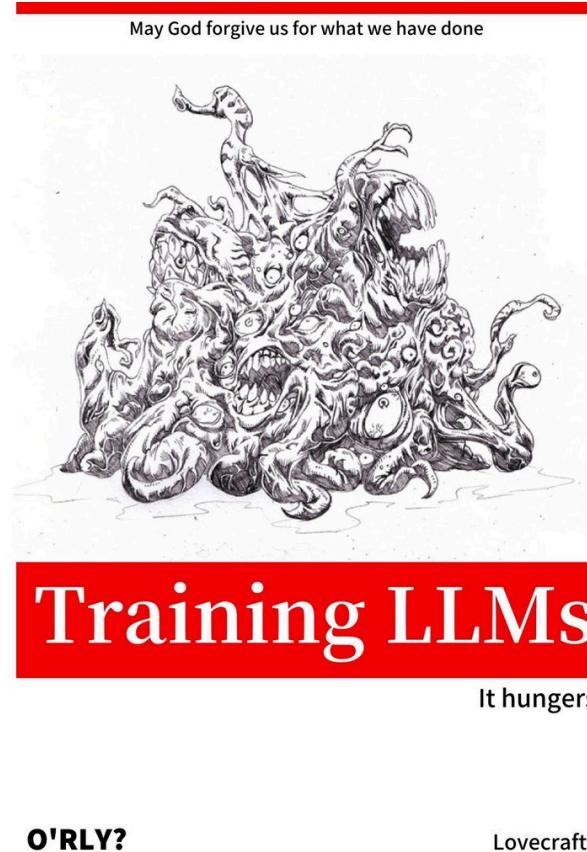


Figure 2: It's hungry!

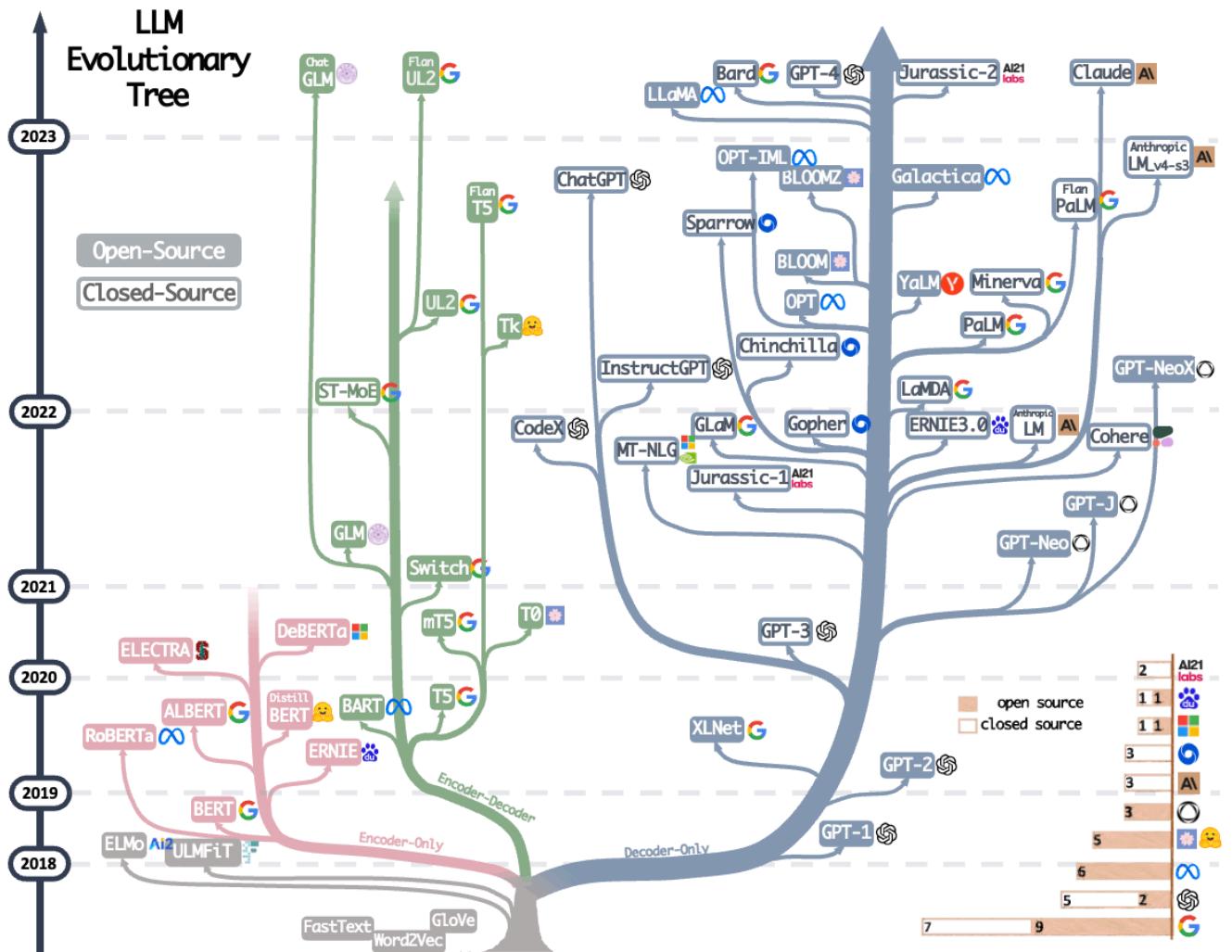


Figure 3: Visualization from Yang et al. (2023)



Pre-training



Fine-Tuning

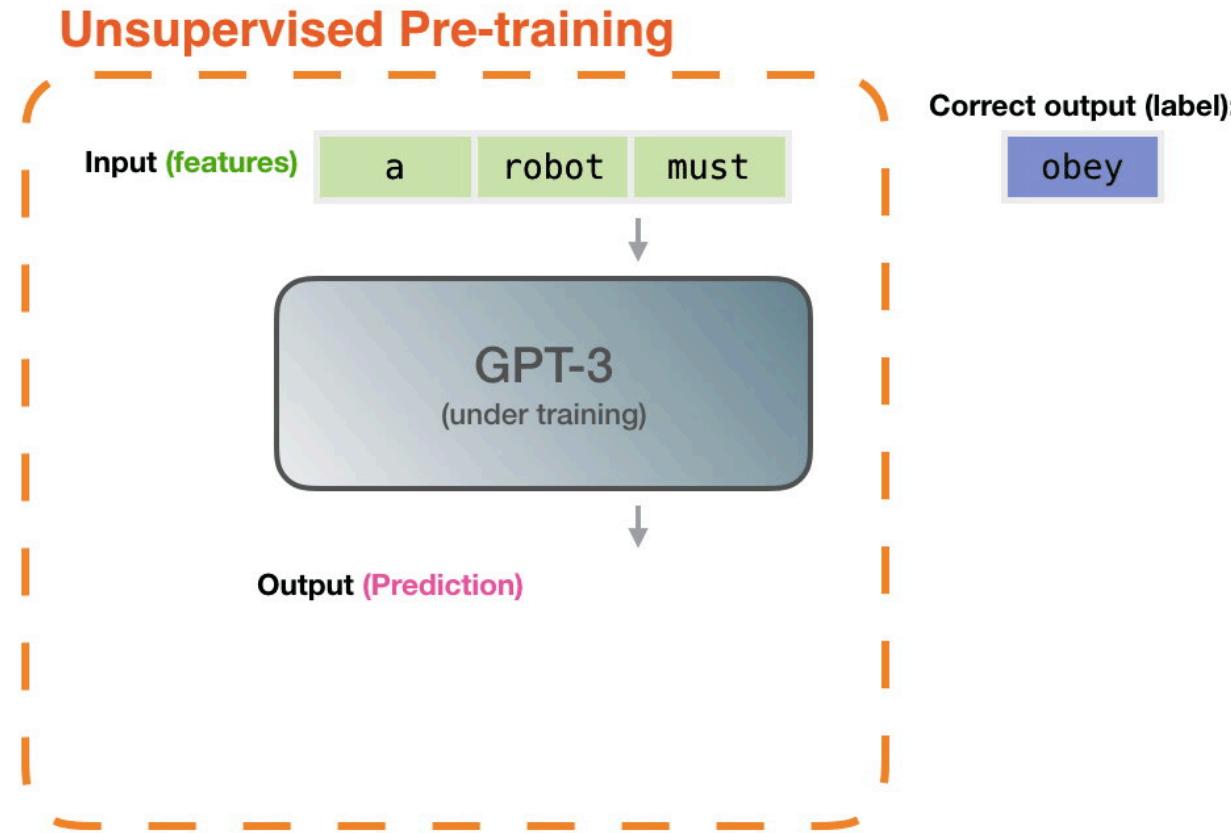


Figure 4: **Pre-training:** Virtually all of the compute used during pretraining phase

Figure from [The Illustrated Transformer](#)



Model Training

✓ Want

- Training runs at scale to be:
 - Efficient
 - Stable
 - Reproducible

✗ Difficulties

- LLMs take a *long time* to train
- Stability particularly important
 - Failures are **expensive**
 - stragglers common at scale
- Jobs are:
 - **fragile**
 - one bad worker can crash job



Looooooooooooong Sequence Lengths

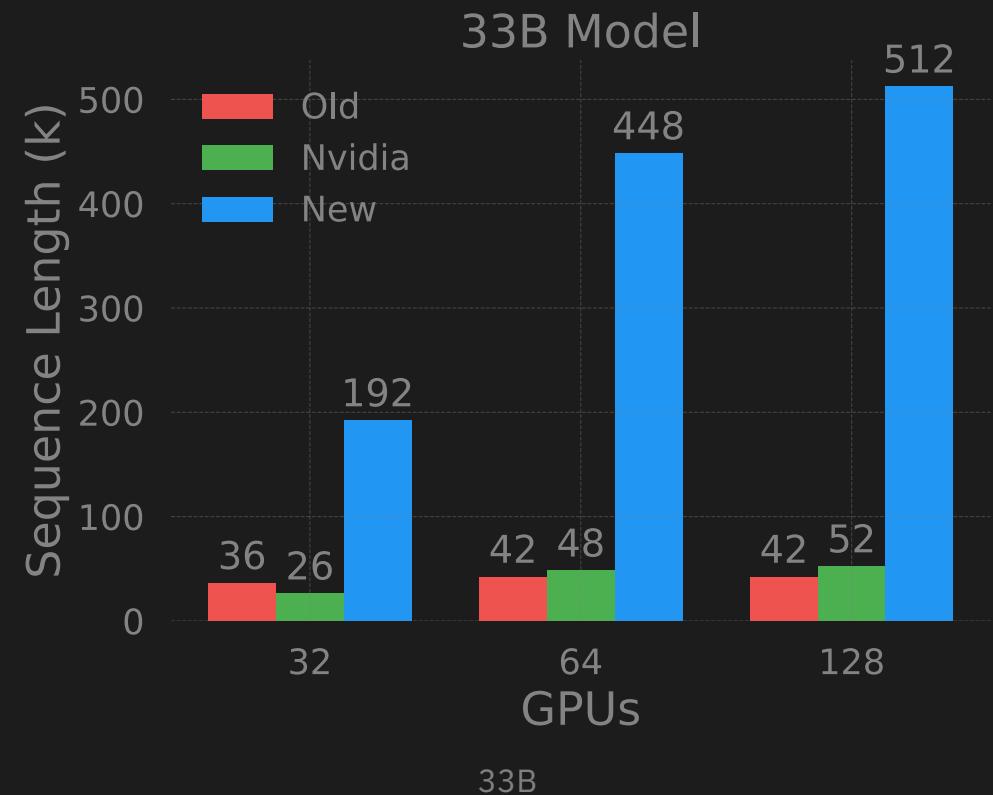
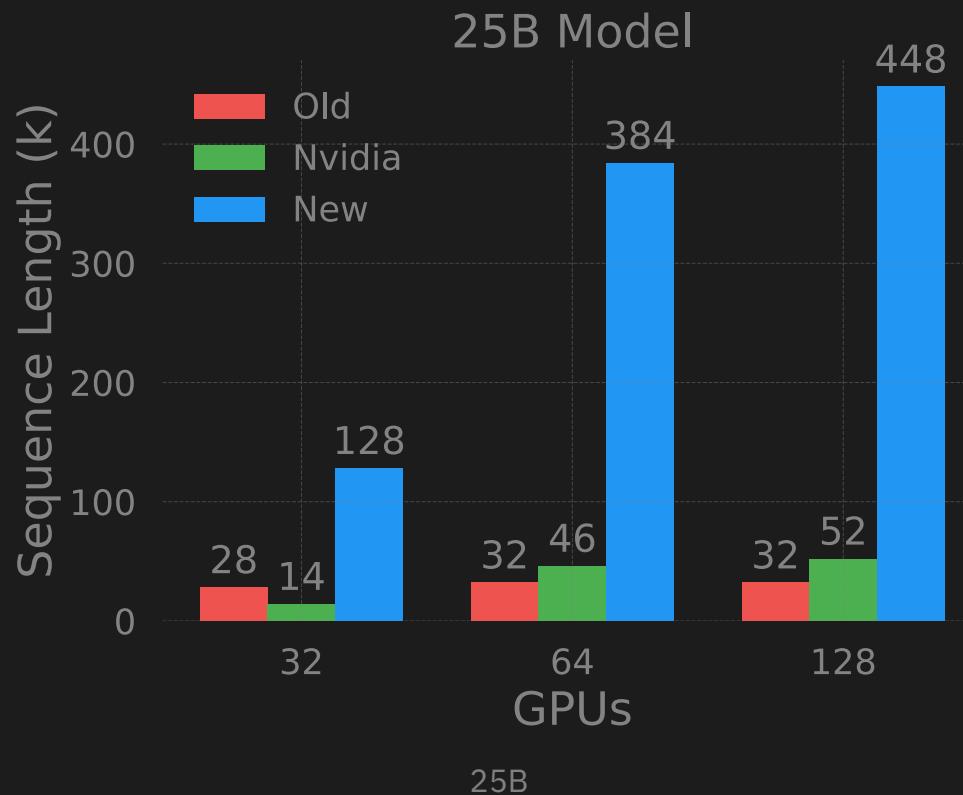
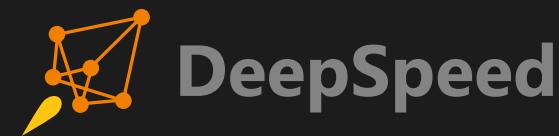


Figure 6: Maximum (achievable) SEQ_LEN for both 25B and 33B models



Evaluation

<https://samforeman.me/talks/hpc-user-forum/slides.html>

Checkpoints

```
1 from typing import Optional
2 import os
3 from pathlib import Path
4
5 from transformers import LlamaForCausalLM, AutoTokenizer
6
7 tokenizer = AutoTokenizer.from_pretrained("meta-llama/Llama-2-7B-hf")
8
9 def load_model(ckpt_dir) -> LlamaForCausalLM:
10     return LlamaForCausalLM.from_pretrained(ckpt_dir)
11
12 def eval_model(model, max_length: int, prompt: str) -> str:
13     return (
14         tokenizer.batch_decode(
15             model.generate(
16                 **tokenizer(prompt, return_tensors="pt"),
17                 max_length=max_length,
18             ),
19             clean_up_tokenization_spaces=True,
20             skip_special_tokens=True,
21             )[0]
22     )
23
24 def load_and_eval_model_from_checkpoint(
25     step: int,
26     max_length: int = 64,
```



Model Evaluations

7000 12000 17000 22000 32000 40000

Tokens: 88B

```
1 >>> print(load_checkpoint(7000))
2 Loading model from checkpoint at global step: 7000
3 "What is it like in there?"
4 """
5 I'm not sure if it's a good idea to use a different name for the same thing,
6 but I'm sure it's a good idea to use a different name for the same thing.
7 I'm not sure if it's a good idea to use a different name for the same thing,
8 but I'm sure it's a good idea to use a different name for the same thing.
9 I'm not sure if it's a good idea to use a different name for the same thing,
10 but I'm sure it
11 """
```



Links

-  [samforeman.me:](https://samforeman.me)
 -  [Talks:](#)
 - [HPC User Forum \[slides\]](https://samforeman.me/talks/hpc-user-forum/slides.html)
 -  [Repos:](#)
 -  [argonne-lcf/Megatron-DeepSpeed](https://github.com/argonne-lcf/Megatron-DeepSpeed)
For only the largest of large language models.
 -  [saforem2/ezpz](https://github.com/saforem2/ezpz)
Train your model across any number of arbitrary devices, ezpz.



References

- See also:
 - [PyTorch Distributed Overview](#)
 - [Distributed Data Parallel – PyTorch master documentation](#)
 - [Efficient Training on Multiple GPUs](#)
 - [Getting Started - DeepSpeed](#)
- See my slides on:
 - [Parallel Training Techniques](#) for additional details
 - [saforem2/11m-lunch-talk \(slides\)](#)

🙏 Thank you!

- Organizers
- ALCF Data Science & Operations
- Feel free to reach out!



🙏 Acknowledgements

This research used resources of the Argonne Leadership Computing Facility, which is a DOE Office of Science User Facility supported under Contract DE-AC02-06CH11357.



Bibliography

- Refs:

- Wei et al. (2022)

Wei, Jason, Yi Tay, Rishi Bommasani, Colin Raffel, Barret Zoph, Sebastian Borgeaud, Dani Yogatama, et al. 2022.
“Emergent Abilities of Large Language Models.” <https://arxiv.org/abs/2206.07682>.

Yang, Jingfeng, Hongye Jin, Ruixiang Tang, Xiaotian Han, Qizhang Feng, Haoming Jiang, Bing Yin, and Xia Hu. 2023.
“Harnessing the Power of LLMs in Practice: A Survey on ChatGPT and Beyond.”
<https://arxiv.org/abs/2304.13712>.