Scaling laws for LLMs

CMSC 848O

Seminar on long-context language models

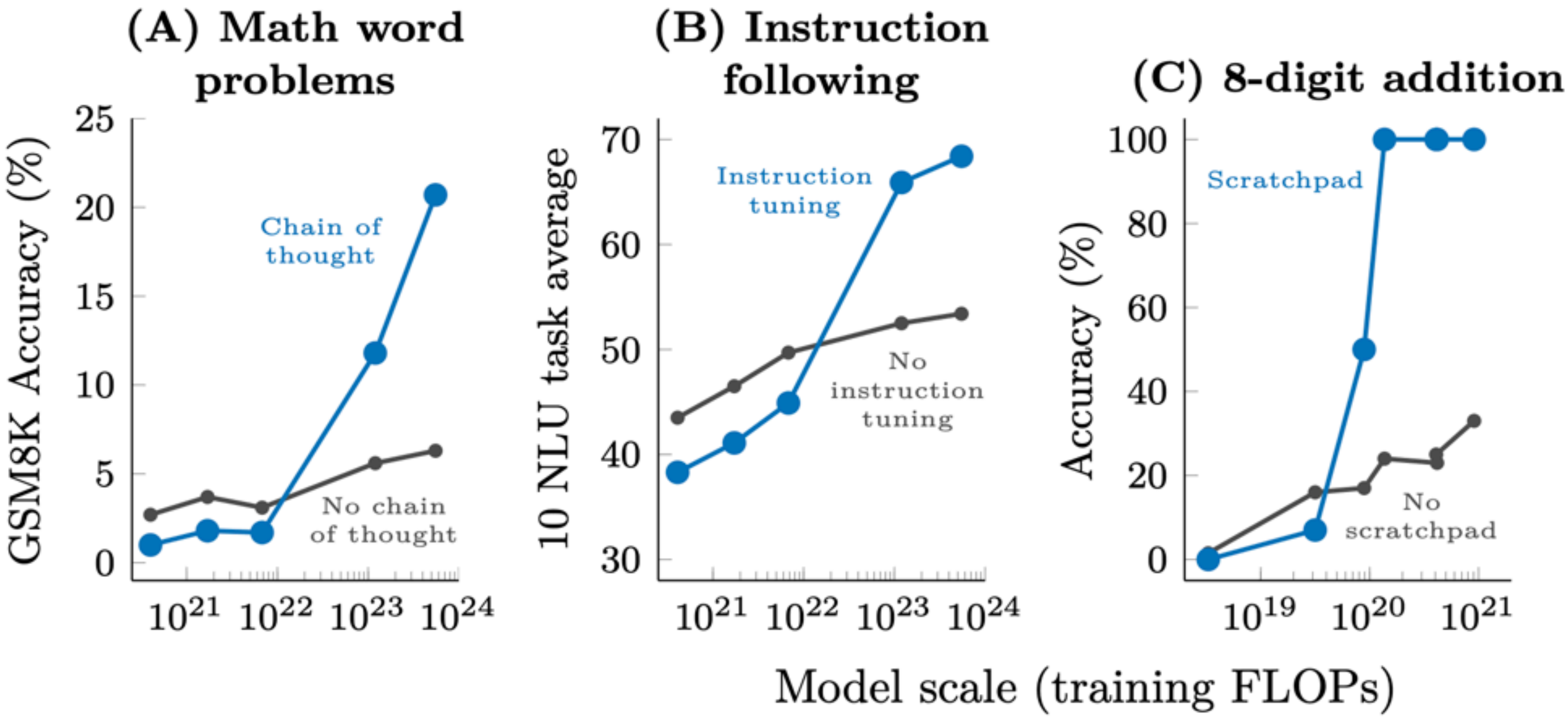
Mohit Iyyer

What do bigger LMs buy us?

- “In-context” learning, chain-of-thought prompting, instruction following, more memorized knowledge and patterns from the training data, etc

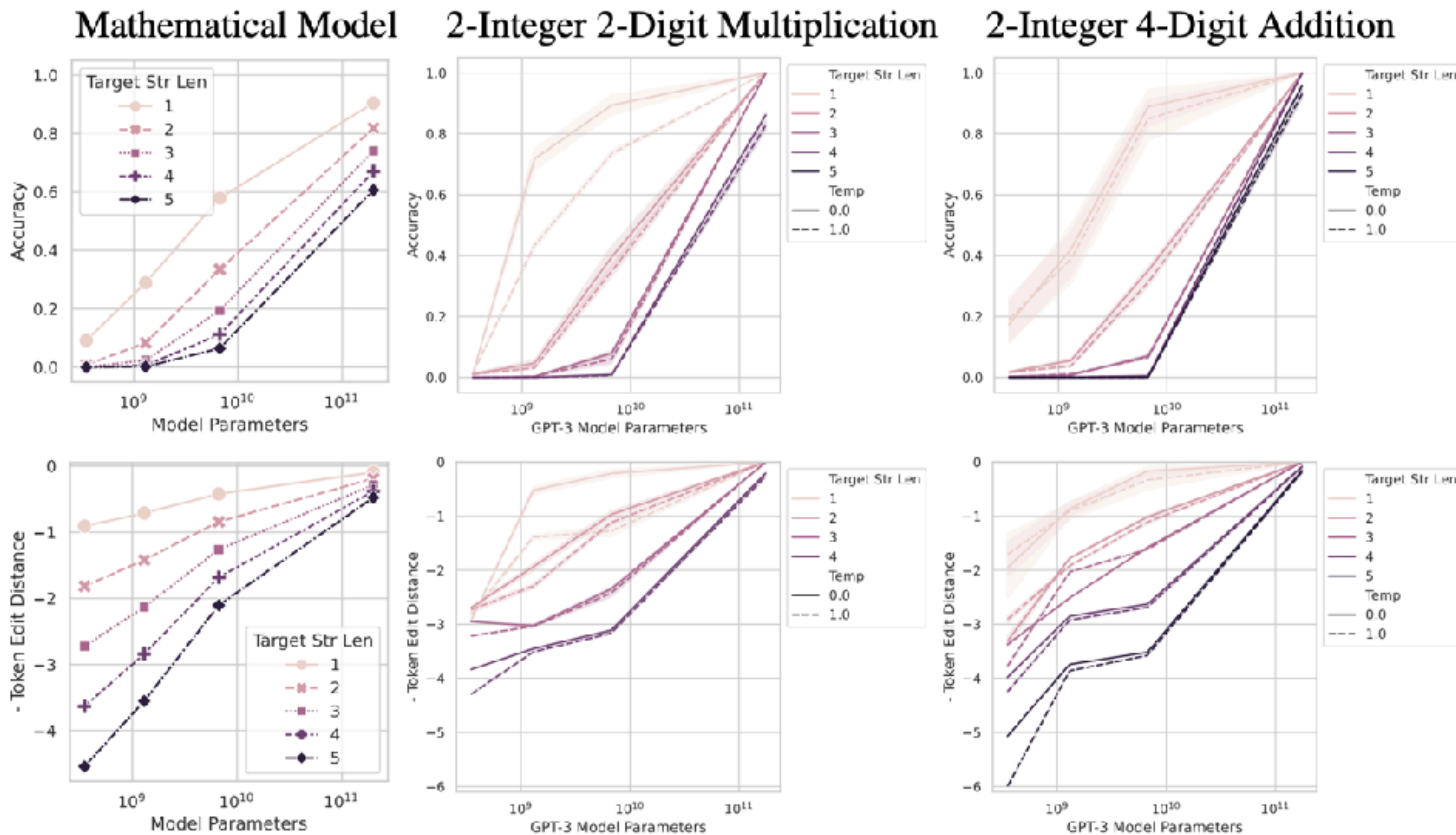
- Broadly, **“emergent properties”**, which may only appear with larger LMs but not smaller ones

“The ability to perform a task via few-shot prompting is emergent when a model has random performance until a certain scale, after which performance increases to well-above random.” (Wei et al., 2022)



Emergent Abilities of Large Language Models, Wei et al., TMLR 2022

Are “emergent properties” really emergent?



Are Emergent Abilities of Large Language Models a Mirage?, Schaeffer et al., *NeurIPS 2023*

What can we scale?

- Model size
- Dataset size
- Amount of total compute used during training (e.g., number of training steps)

Given a fixed compute budget, what is the optimal model size and training dataset size for training a Transformer LM?

Let's say you can use one GPU for one day

- Would you train a 5 million parameter LM on 100 books?
- What about a 500 million parameter LM on one book?
- Or a 100k parameter LM on 5k books?

Observations from Kaplan et al., 2020

- Performance depends strongly on scale (model params, data size, and compute used for training), weakly on model shape (e.g., depth, width)

- Perf vs scale can be modeled with power laws

- Perf improves most if model size and dataset size are scaled up together. Increasing one while keeping the other fixed leads to diminishing returns

- Larger models are more sample efficient than smaller models, take fewer steps / data points to reach same loss

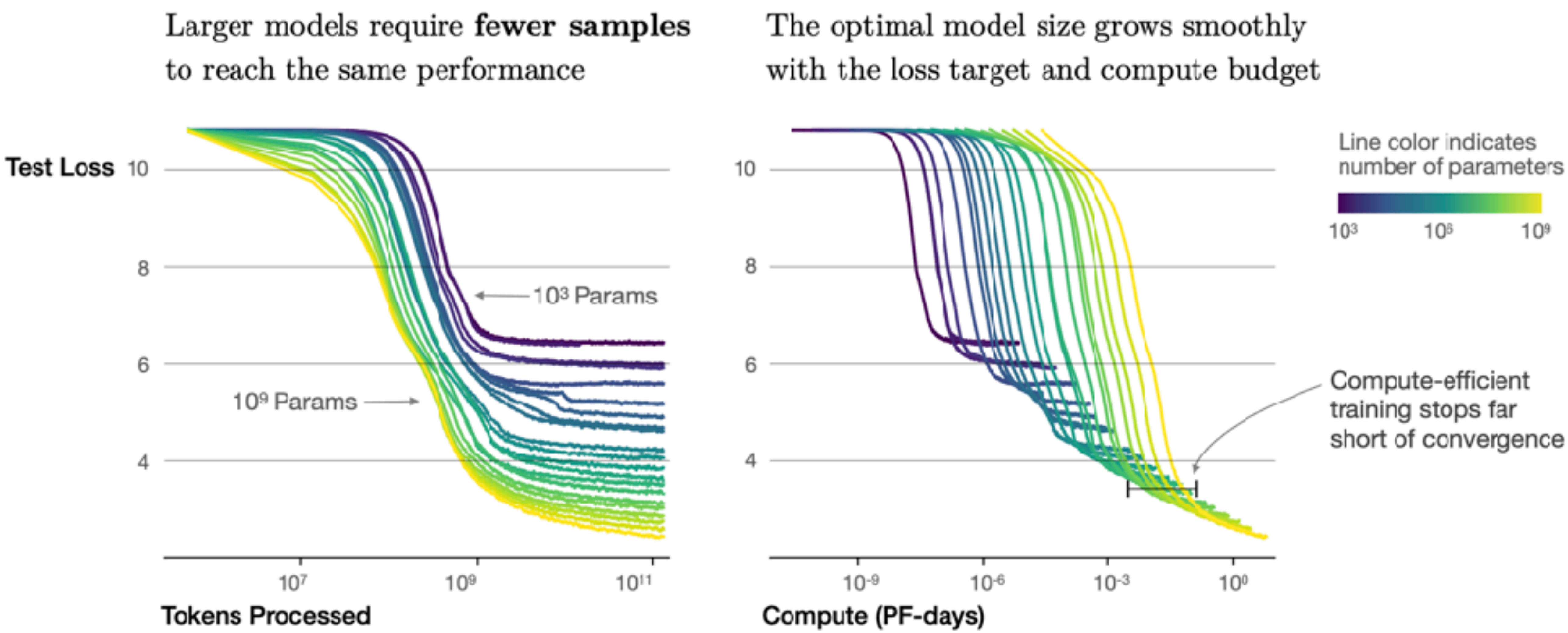


Figure 2 We show a series of language model training runs, with models ranging in size from 10^3 to 10^9 parameters (excluding embeddings).

Issues with Kaplan laws

- Used same learning rate schedule for all training runs, regardless of how many training tokens / batches!
- This schedule needs to be adjusted based on the number of training steps; otherwise, it can impair performance
- The resulting “scaling laws” from Kaplan et al., are flawed because of this!

Chinchilla (Hoffmann et al., 2022)

Quick takeaways

- **Kaplan et al., 2020**: if you're able to increase your compute budget, you should prioritize increasing model size over data size
  - With a 10x compute increase, you should increase model size by 5x and data size by 2x
  - With a 100x compute increase, model size 25x and data 4x
- **Hoffmann et al., 2022**: you should increase model and data size at the same rate
  - With a 10x compute increase, you should increase both model size and data size by 3.1x
  - With a 100x compute increase, both model and data size 10x

Given a fixed compute budget, what is the optimal model size and training dataset size for training a Transformer LM?

Model	Size (# Parameters)	Training Tokens
LaMDA (Thoppilan et al., 2022)	137 Billion	168 Billion
GPT-3 (Brown et al., 2020)	175 Billion	300 Billion
Jurassic (Lieber et al., 2021)	178 Billion	300 Billion
Gopher (Rae et al., 2021)	280 Billion	300 Billion
MT-NLG 530B (Smith et al., 2022)	530 Billion	270 Billion
Chinchilla	70 Billion	1.4 Trillion

- N – the number of model parameters, *excluding all vocabulary and positional embeddings*
- $C \approx 6NBS$ – an estimate of the total non-embedding training compute, where B is the batch size, and S is the number of training steps (ie parameter updates). We quote numerical values in PF-days, where one PF-day $= 10^{15} \times 24 \times 3600 = 8.64 \times 10^{19}$ floating point operations.

TLDR: Chinchilla says to train a compute-optimal model, you should use ~20 tokens for every parameter

TLDR: Chinchilla says to train a compute-optimal model, you should use ~20 tokens for every parameter

However, most modern models are *overtrained* by this definition

TLDR: Chinchilla says to train a compute-optimal model, you should use ~20 tokens for every parameter

Model	# Params	# Training Tokens	Ratio
Chinchilla	70B	1.4T	20 tokens / param
Llama 3	70B	14T	200 tokens/param
Phi-3	3.8B	3.3T	875 tokens/param
Llama 3	8B	14T	1875 tokens/ param

What about the *type* of data?

What about the *type* of data?

- The internet contains a huge amount of text, but it’s extremely noisy! Copyrighted text (e.g. published books) are much higher-quality, but is it legal to train on them?

- What is the impact of *repeated* data?
  - Repeated data can lead to severe degradation in performance (Brown et al., 2022)
    - *“For instance, performance of an 800M parameter model can be degraded to that of a 2x smaller model (400M params) by repeating 0.1% of the data 100 times, despite the other 90% of the training tokens remaining unique.”*
  - Repeated data is helpful (Taylor et al., 2022; Galactica)
    - *“We train the models for 450 billion tokens, or approximately 4.25 epochs. We find that performance continues to improve on validation set, in-domain and out-of-domain benchmarks with multiple repeats of the corpus.”*
    - *“We note the implication that the "tokens → ∞" focus of current LLM projects may be overemphasised versus the importance of filtering the corpus for quality.”*

Google Books Search

Books of the world, stand up and be counted! All 129,864,880 of you.

Thursday, August 05, 2010 at 8:26 AM

Posted by Leonid Taycher, software engineer

When you are part of a company that is trying to digitize all the books in the world, the first question you often get is: “Just how many books are out there?”

http://booksearch.blogspot.com/2010/08/books-of-world-stand-up-and-be-counted.html