Scaling Language Models

Antoine Bosselut

EPFL

nlp lab

Announcements

- **No Lecture Tomorrow!**
- **Assignment 3:** Due Sunday, 21/04/2024 at 11:59 PM
  - Office Hours: **tomorrow**, Thursday 2 PM
- **Assignment 1:** Grading Feedback Session Tomorrow at 2 PM
- **Course Project:** Kickoff!
  - Data Packages were released. If you haven't received them yet, contact us.

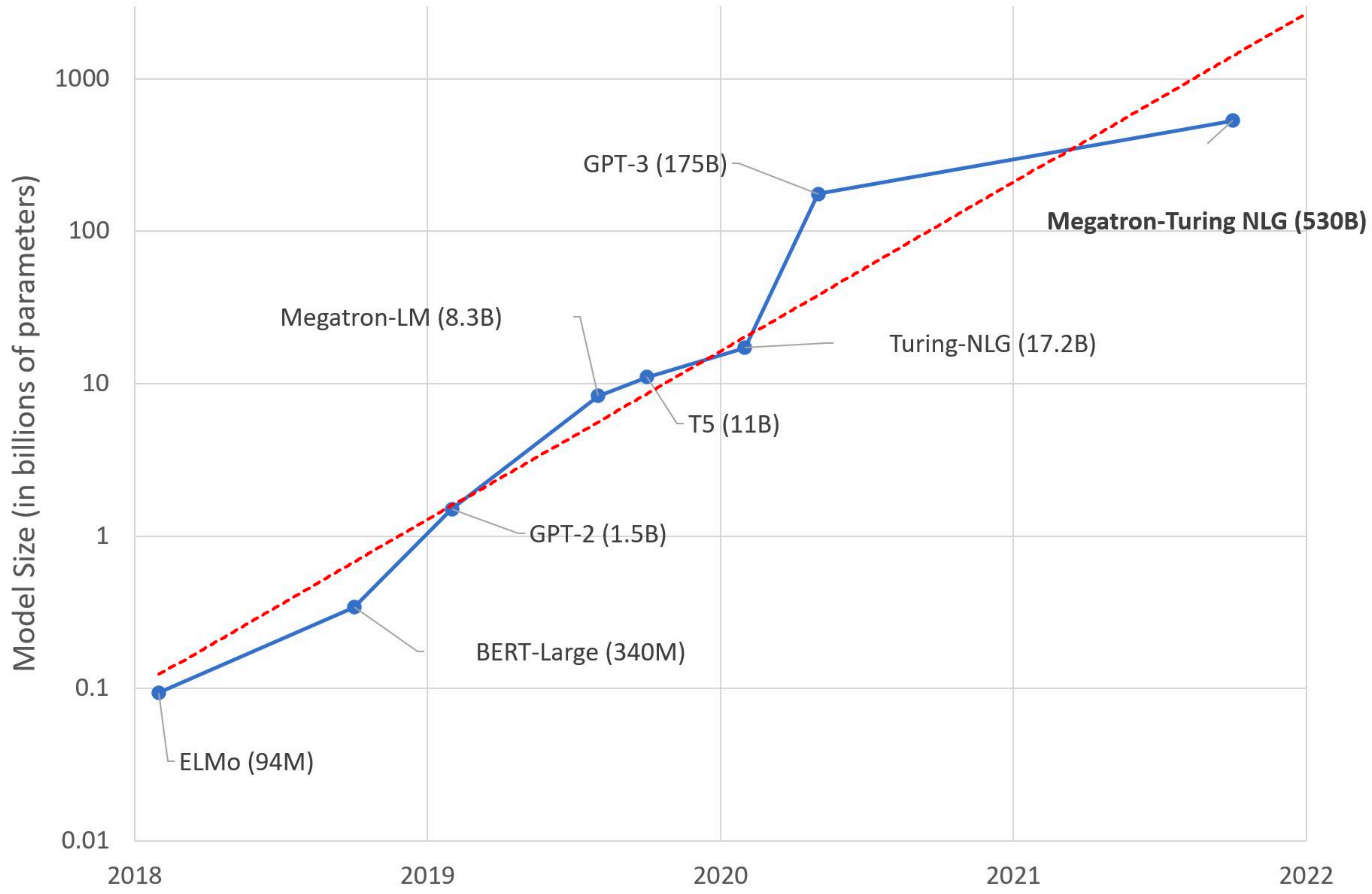
Next few days: Project Sign-ups

- **To-Dos:**
  - **URGENT:** Look over the Project Description
  - **URGENT:** Sign up for project repository
  - Look through README in project repository for details on milestone submission
  - Get started early! **Milestone 1 due May 5th!**

Today's Outline

- **Lecture**
  - **Quick Recap:** Scale
  - **Managing scale when training:** Scaling laws
  - **Managing scale when deploying:** Model Compression (pre-recorded video)
    - how can we make LLMs more compute- and memory-efficient for deployment ?

Language Model Scaling



Larger models

More data

More compute

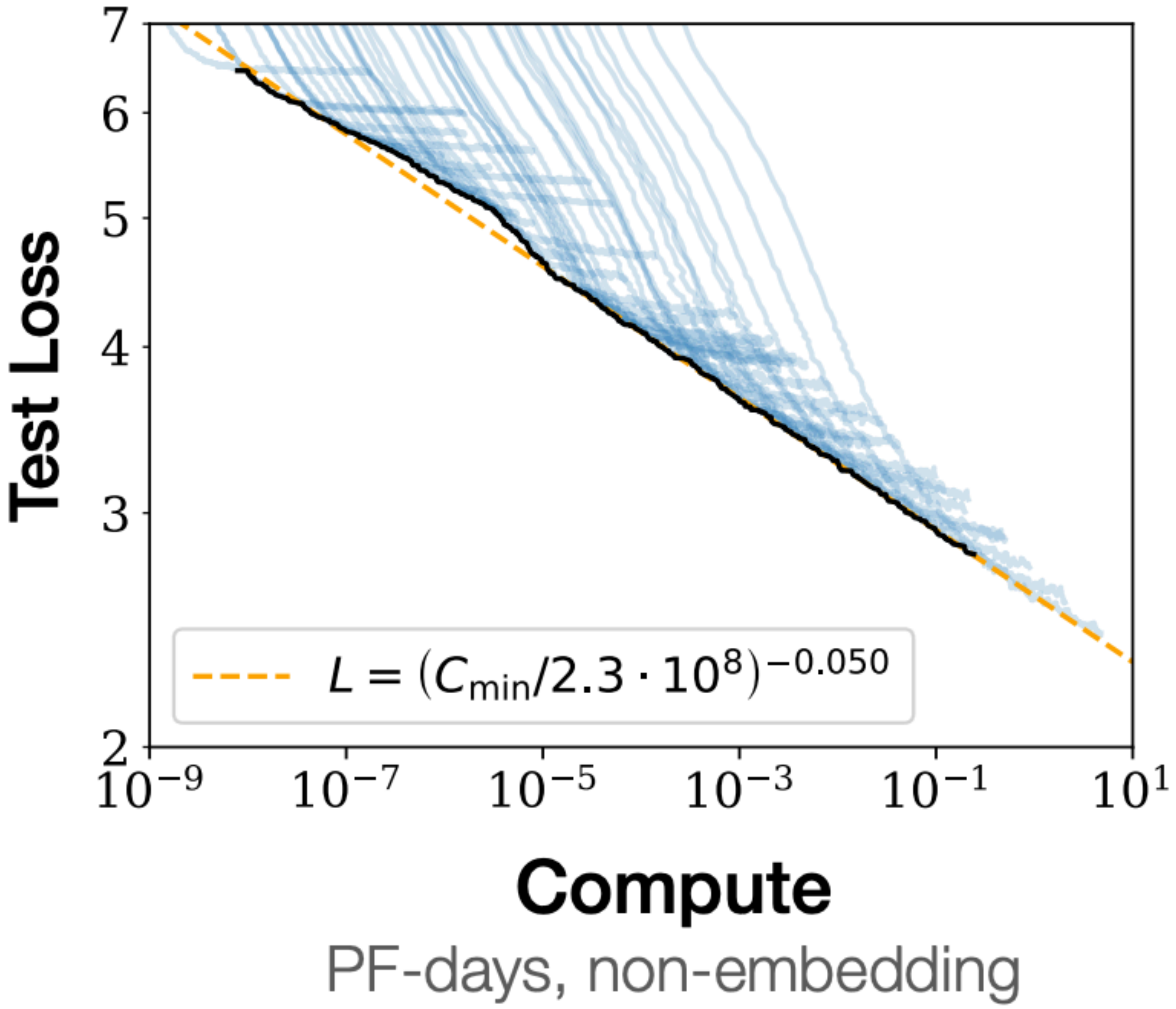
More 

Every part of the model scales!

Model Name	n_{params}	n_{layers}	d_{model}	n_{heads}	d_{head}	Batch Size	Learning Rate
GPT-3 Small	125M	12	768	12	64	0.5M	6.0×10^{-4}
GPT-3 Medium	350M	24	1024	16	64	0.5M	3.0×10^{-4}
GPT-3 Large	760M	24	1536	16	96	0.5M	2.5×10^{-4}
GPT-3 XL	1.3B	24	2048	24	128	1M	2.0×10^{-4}
GPT-3 2.7B	2.7B	32	2560	32	80	1M	1.6×10^{-4}
GPT-3 6.7B	6.7B	32	4096	32	128	2M	1.2×10^{-4}
GPT-3 13B	13.0B	40	5140	40	128	2M	1.0×10^{-4}
GPT-3 175B or “GPT-3”	175.0B	96	12288	96	128	3.2M	0.6×10^{-4}

- Trained on 570 GB of Common Crawl data
- **How?** Used cluster provided by Microsoft

Why scale?



- Last week, we talked about benefits of scaling in terms of **emergence**
- Practically, training for longer also leads to lower test loss
- Larger models can reach lower test losses

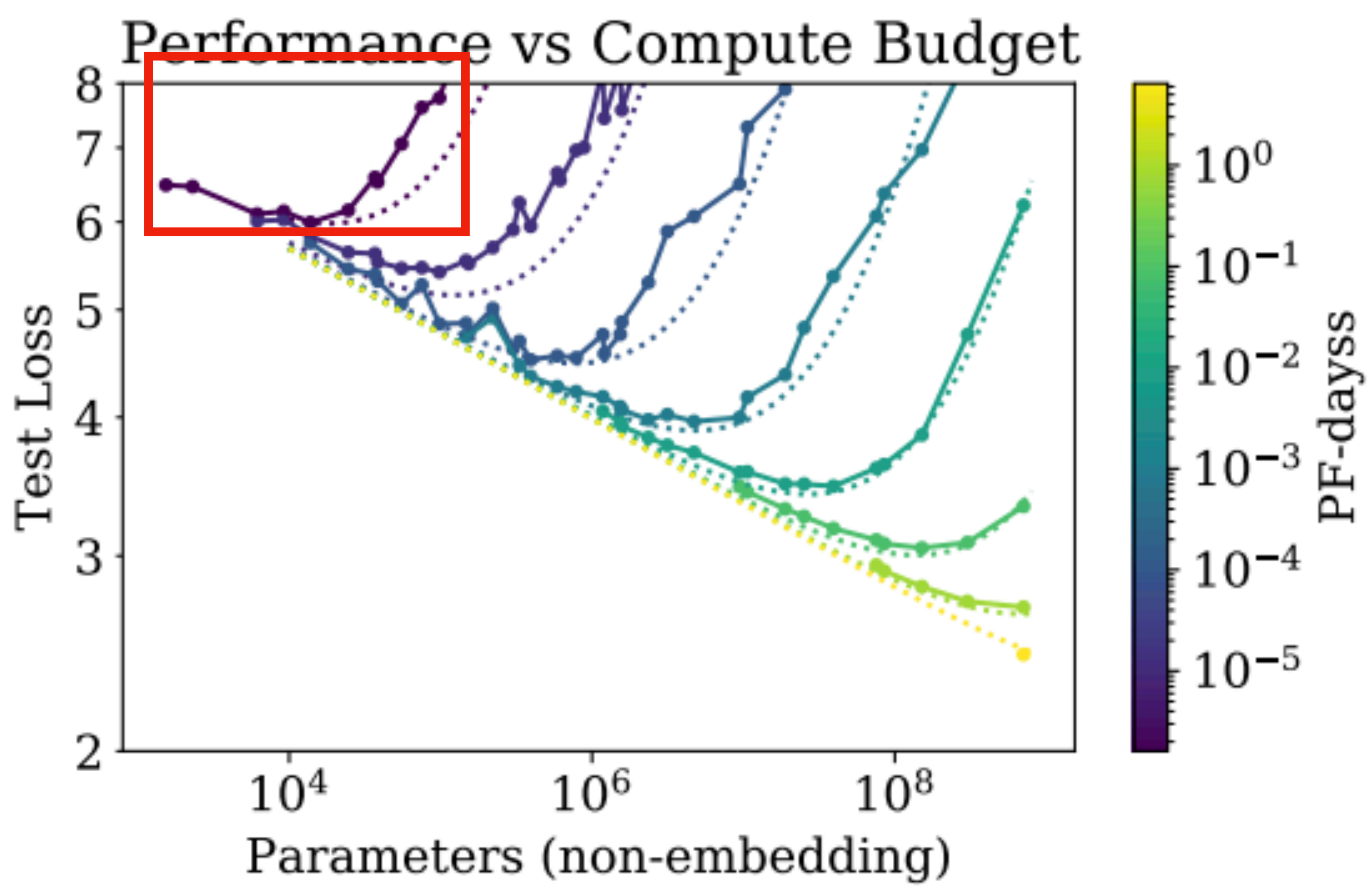
Kaplan et al. (2020)

What should we scale?

Model size, dataset size, compute budget

**Given a compute budget, how big of a model can we train?
and how big of a dataset should we train it on?**

Impact of compute budget



Dotted lines estimate these curves. Need to predict for larger models!

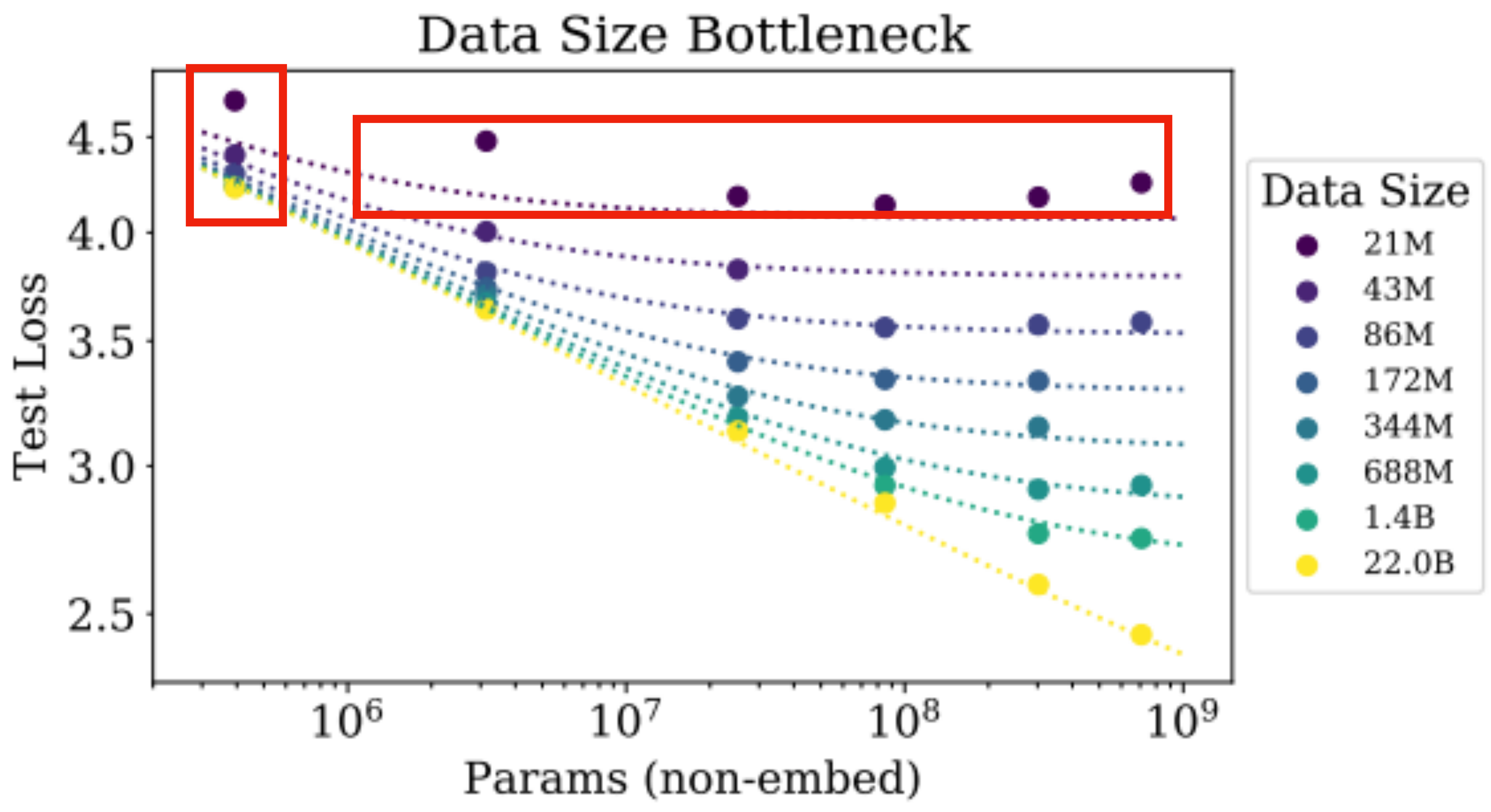
- For a fixed compute budget, there is an optimal number of parameters that we can train
- Having **too large** a model for **too small** a compute budget does NOT let the model learn
  - Model doesn't see enough examples during training
- Having **too small** a model for too large a compute budget is also bad
  - Repeated computation isn't helpful if the model has no capacity to encode additional information

Kaplan et al. (2020)

Consideration

- With a fixed compute budget (in FLOP-days), we have two costs:
  - Number of floating point operations needed to train on a single example (model size)
  - Number of total examples we will train on
- **How should we trade off these two costs?**
  - Which should we prioritise?

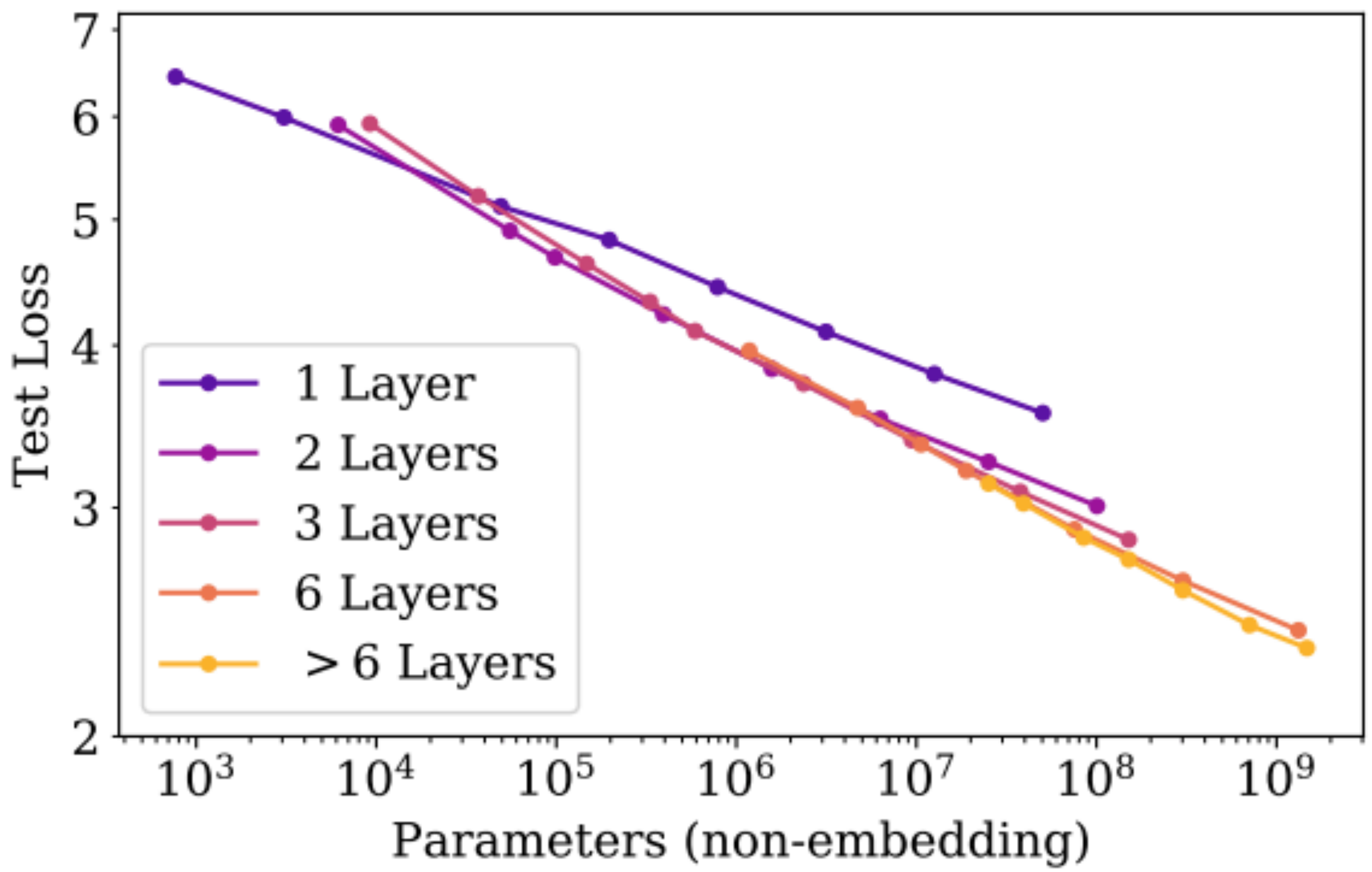
Model-Data Trade-off

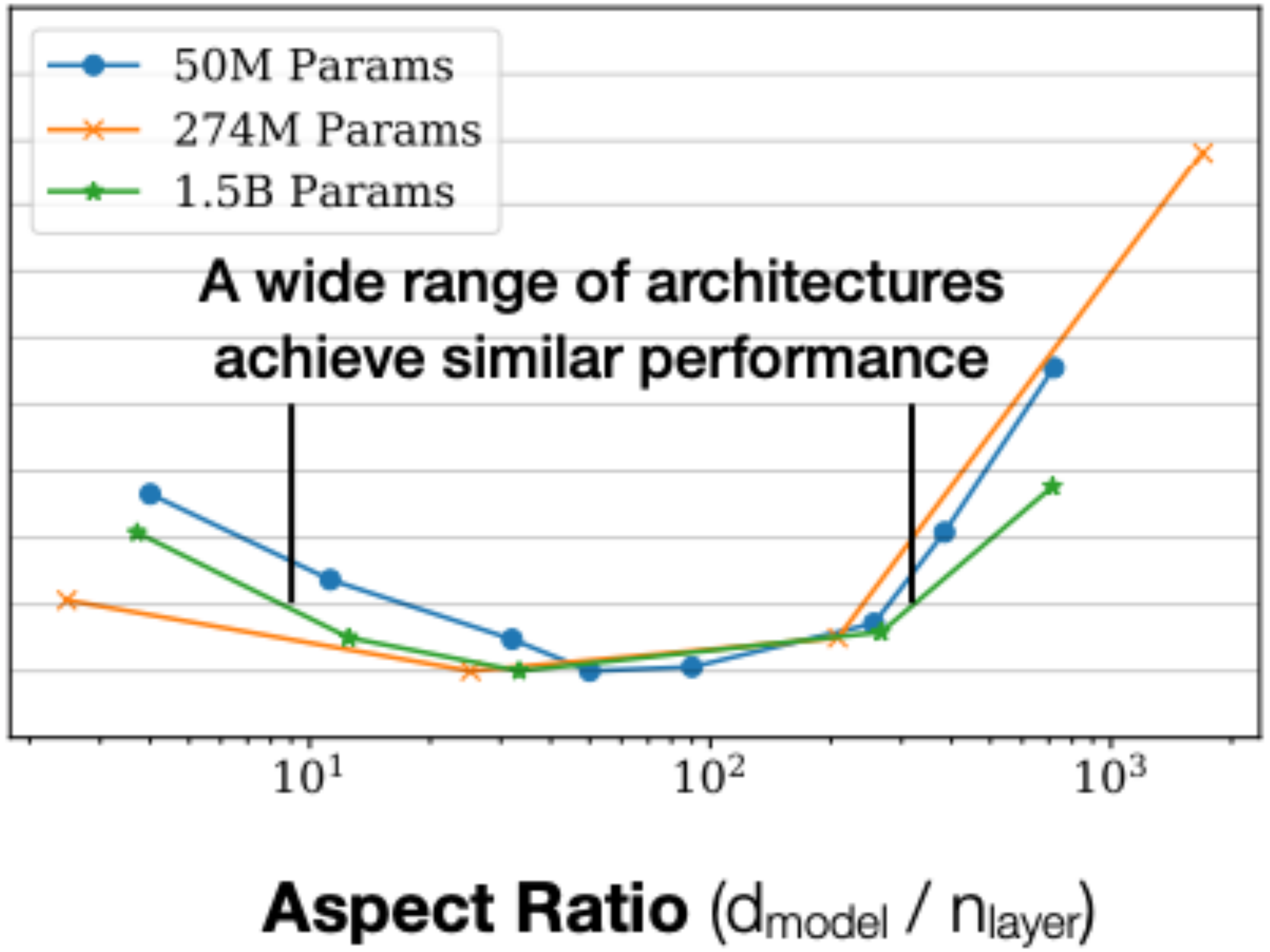


- Larger models benefit more from larger datasets
- **Smaller models saturate**
  - Only so much capacity to learn!
- At some point, larger models don't benefit more from same-sized data
- Model size needs to be scaled jointly with data size

Kaplan et al. (2020)

Other Cool Findings

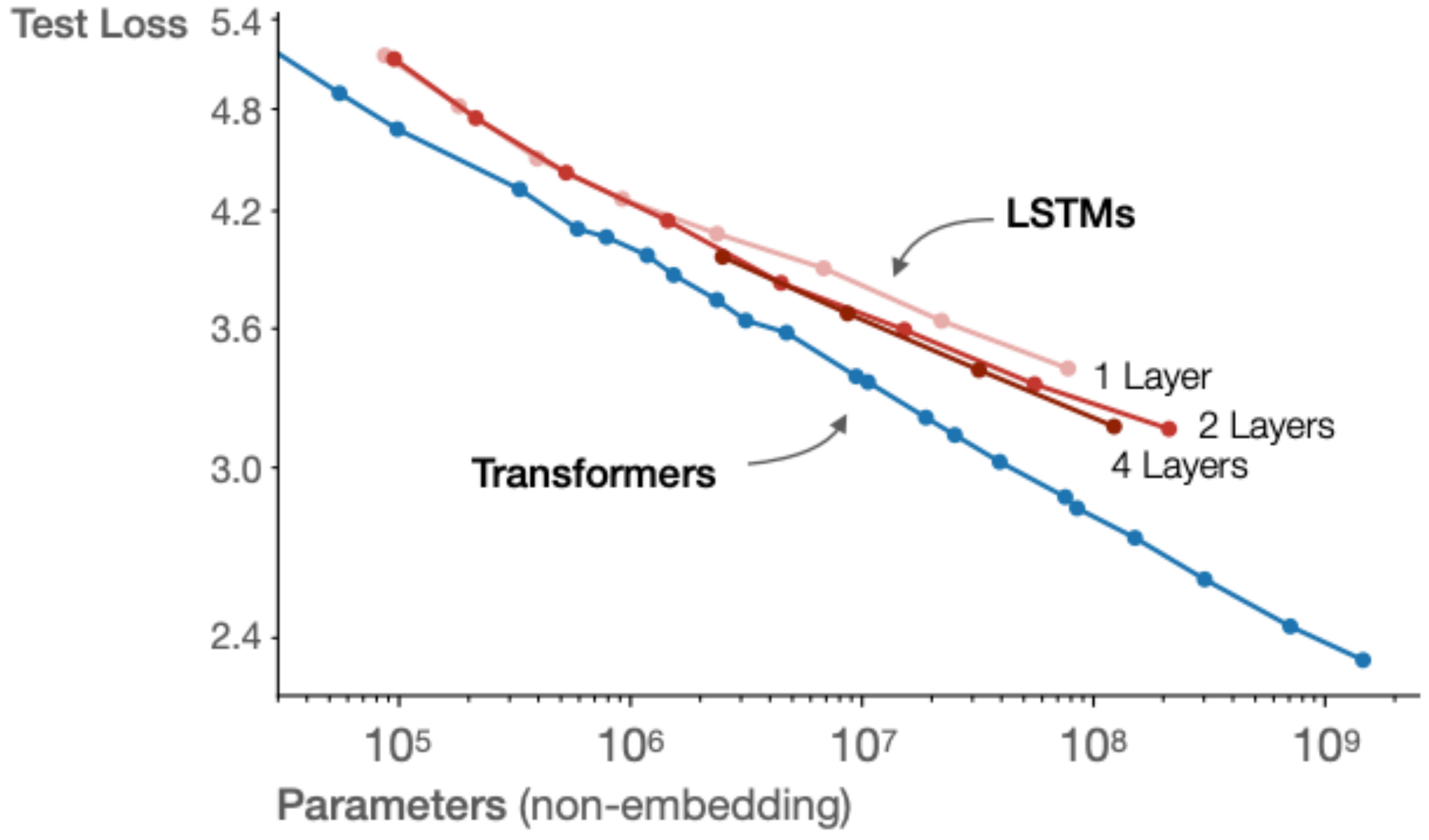




No need to make models terribly deep

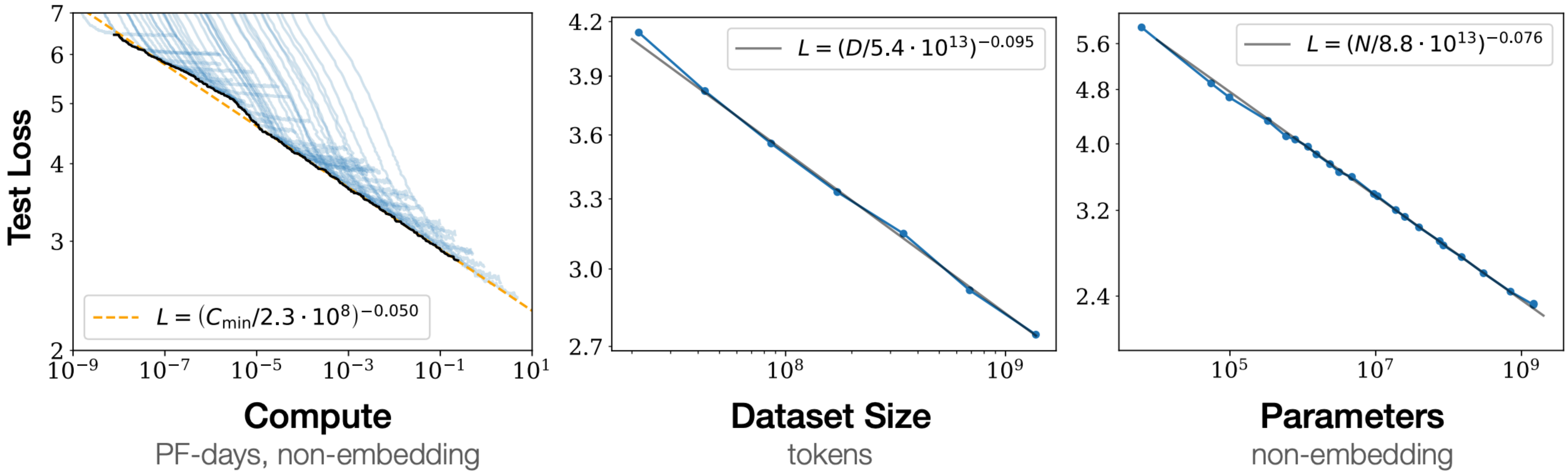
Multiple ratios of depth vs. width (aka embedding size) are fine

Other cool findings



- LSTMs also follow scaling laws, benefitting from increased scale
- They scale less efficiently than transformers, though
- They still have trouble modelling long-term dependencies (>100 tokens)

To scale up: estimate model, data, compute



- Assuming no bottlenecks, expected test loss has power law relationship with each variable
- From smaller models, we can estimate how much compute, data, and model size is needed to achieve a particular test loss

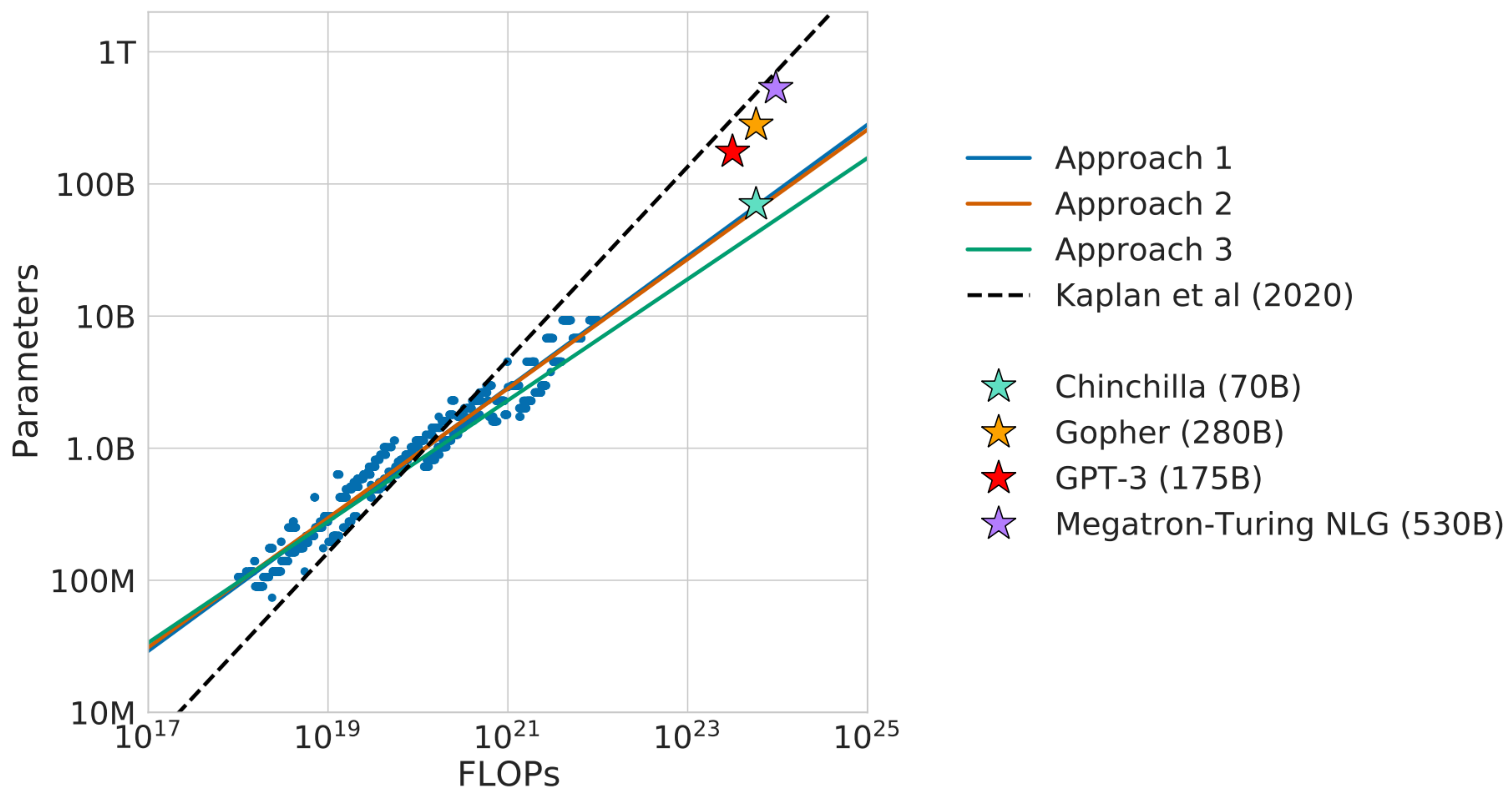
Kaplan et al. (2020)

Model Scaling in the last two years

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

What happens if we get these estimates wrong?

Oops!



- Chinchilla authors founds that original works on model scaling had poorly estimated power laws
- New estimates showed that a 4x smaller model should be used for the compute budget
- Trained Gopher (280B) before finding this out!

Hoffmann et al. (2020)

Model Scaling in the last few years

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

Chinchilla gets better performance than all of the above models on most common NLP benchmarks!

Smaller model, trained on much more data!

Hoffmann et al. (2020)

Should we train the largest model that will converge given the data and compute we have ?

Not necessarily ! Why not ?

Inference cost!

Importance of Inference

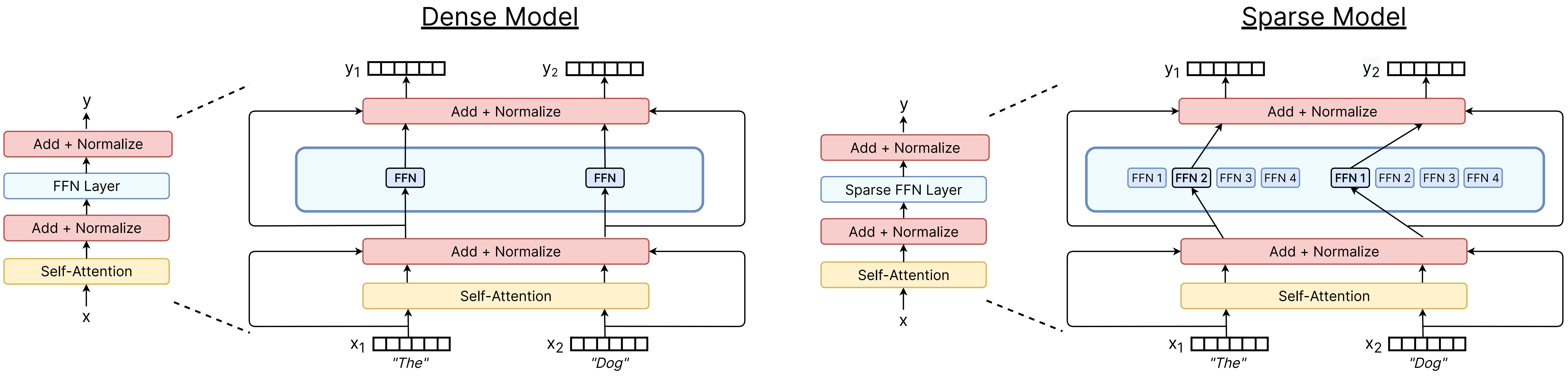
	GPU Type	GPU Power consumption	GPU-hours	Total power consumption
OPT-175B	A100-80GB	400W	809,472	356 MWh
BLOOM-175B	A100-80GB	400W	1,082,880	475 MWh
LLaMA-7B	A100-80GB	400W	82,432	36 MWh
LLaMA-13B	A100-80GB	400W	135,168	59 MWh
LLaMA-33B	A100-80GB	400W	530,432	233 MWh
LLaMA-65B	A100-80GB	400W	1,022,362	449 MWh

- Scaling laws helps estimate dataset and model size for a given *training compute budget*
  - Ignores, the compute *inference budget*
  - How much should a single query cost ?
  - Training cost is amortised; inference cost is constant
- LLaMa authors showed that training smaller models (7B) on more data (1T tokens) continued to improve them
- Worse performance than 65B model, but much cheaper for inference (10x!)

Touvron et al. (2023)

How can we reduce inference cost while still keeping model capacity high ?

Mixture-of-Experts



- Initialise multiple **FFNs** in the transformer block
- Initialise routing function that selects an **FFN** that the out of self-attention should be routed to
  - Input can be routed to multiple **FFNs** (i.e., Top-K routing), but top-2 is common
- Model can have more parameters as number of "experts" increases, but inference cost per example remains the same

GPT-4 rumoured to be mixture-of-experts architecture

Fedus et al. (2022)

Recap

- Scale is necessary to achieve many of the emergent breakthroughs of the last few years
  - in-context learning, chain-of-thought reasoning, instruction learning
- Training at scale is very expensive
  - Potentially, months of training = millions of $$$$
- Scaling laws let us estimate the optimal model and dataset sizes for a fixed compute budget, so that we only have to do the training once!
- While scaling laws suggests we should train the largest model possible, downstream *inference cost* is important to consider as well

References

- Kaplan, J., McCandlish, S., Henighan, T., Brown, T. B., Chess, B., Child, R., ... & Amodei, D. (2020). Scaling laws for neural language models. arXiv preprint arXiv:2001.08361.
- Hoffmann, J., Borgeaud, S., Mensch, A., Buchatskaya, E., Cai, T., Rutherford, E., ... & Sifre, L. (2022). Training compute-optimal large language models. arXiv preprint arXiv:2203.15556.