Lecture 21: Massive models

CS 182/282A ("Deep Learning")

2022/04/13

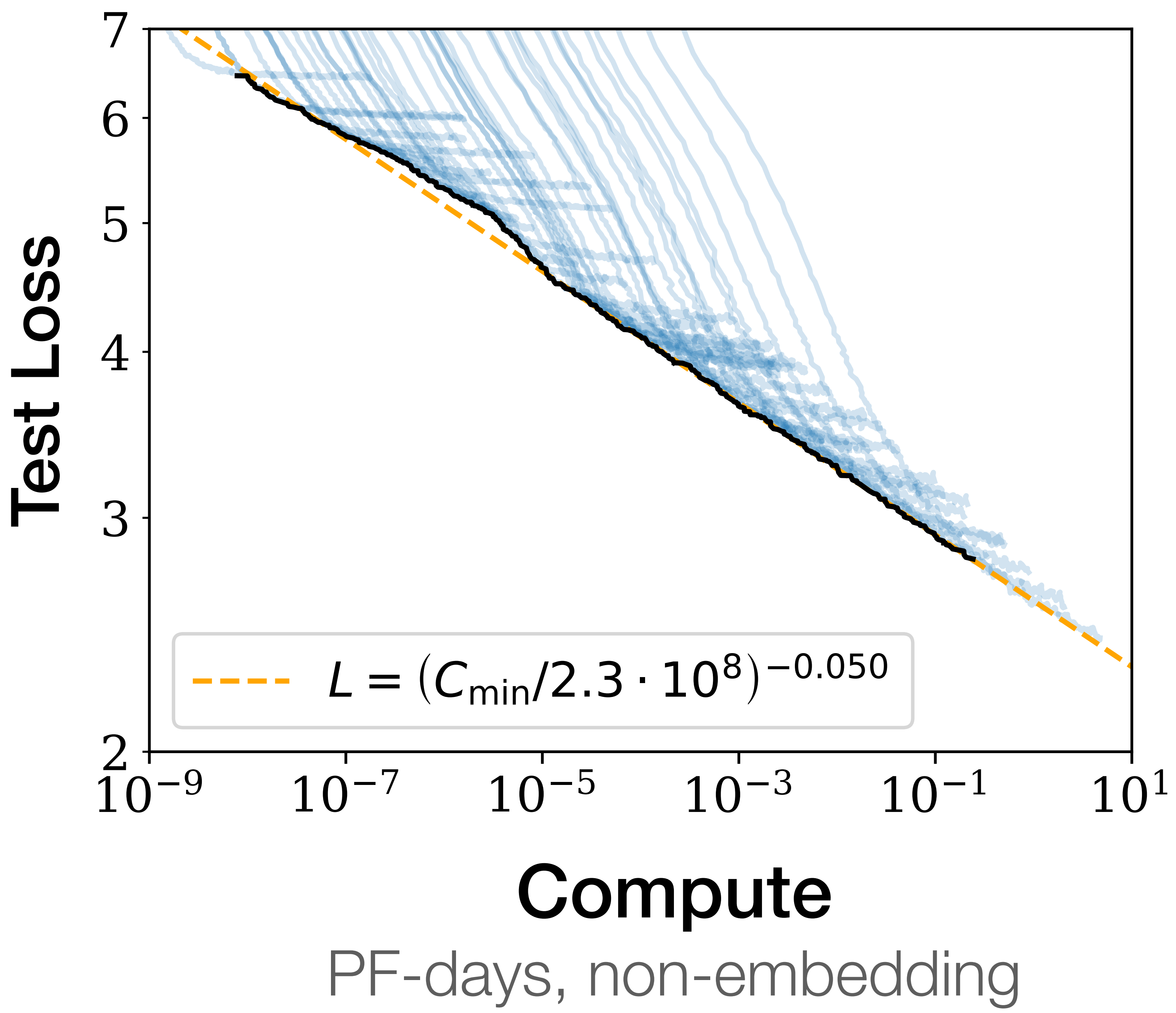
Today’s lecture

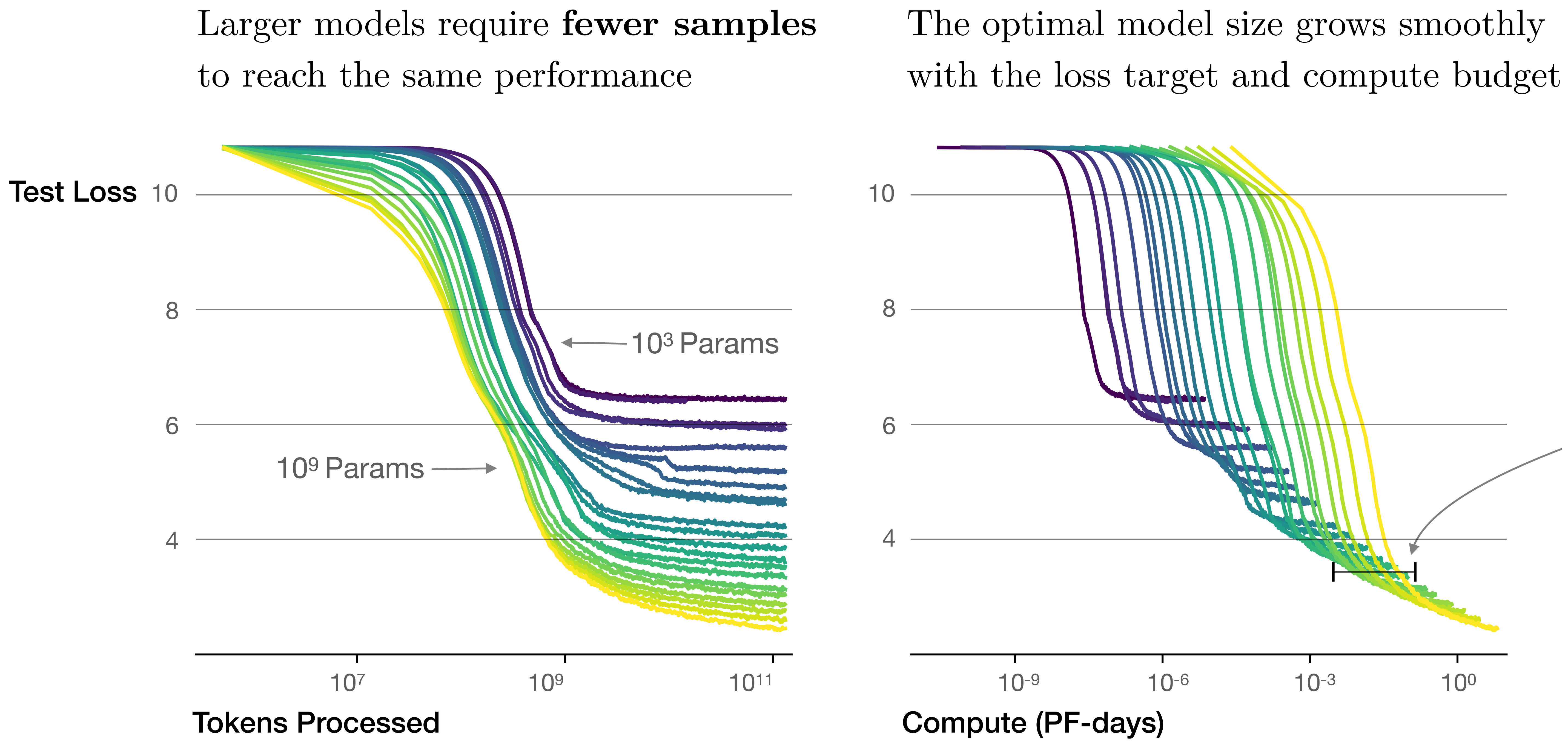
- Today, we take a tour of the current landscape of massive models
- There are only a handful of models with >100B parameters, all of them (as far as I know) are transformer decoder language models
- We will go over the high level details of four of these models
- A natural question is whether moving in this direction is the right way to go; we will study this question theoretically, via **scaling laws**, as well as practically
- Finally, we will review some applications and current limitations of these models

Scaling laws

Scaling laws for neural language models

Kaplan et al, 2020





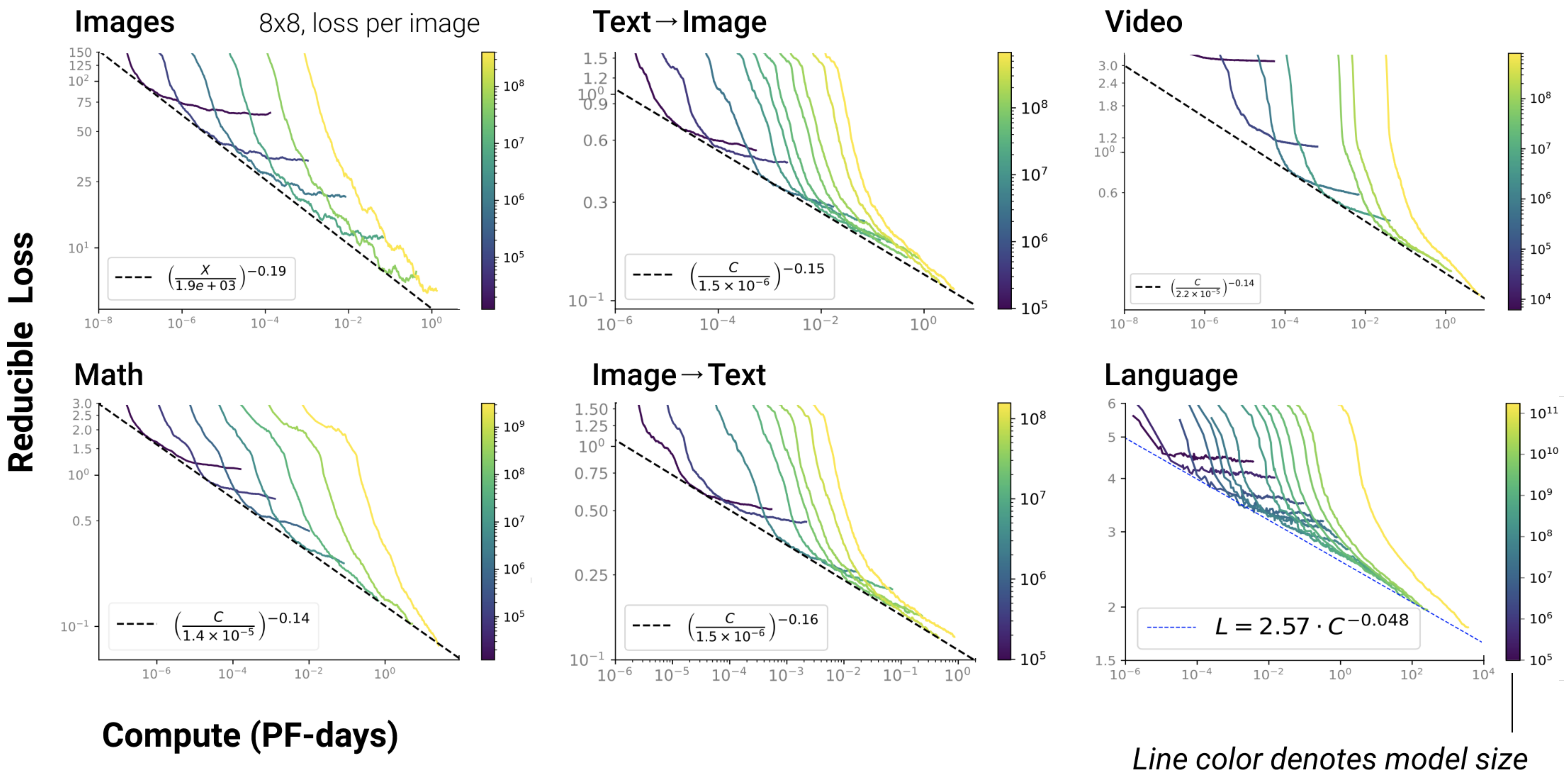
Scaling laws for neural language models

Kaplan et al, 2020

- These scaling laws hold for over six orders of magnitude for amount of available compute and model size
- Model size and dataset size need to be scaled together, but not equally — roughly, $8\times$ model size increase requires only $5\times$ dataset size increase
  - This point is disputed by some other work that says equal scaling is best
- Larger models require fewer data points and optimization steps to reach the same performance as smaller models
- For a fixed compute budget, the best performance is obtained by training large models and stopping well short of convergence

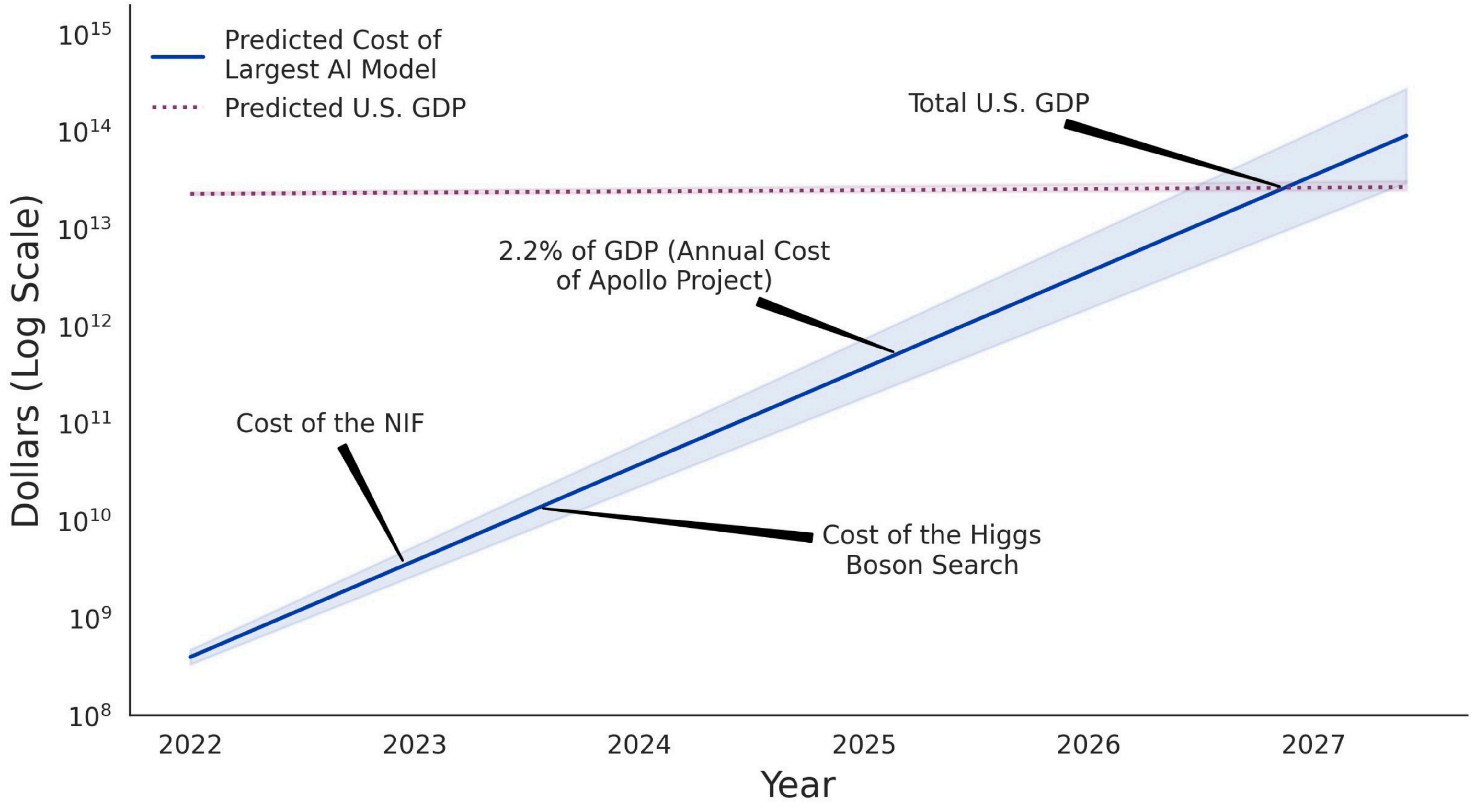
Scaling laws for autoregressive generative modeling

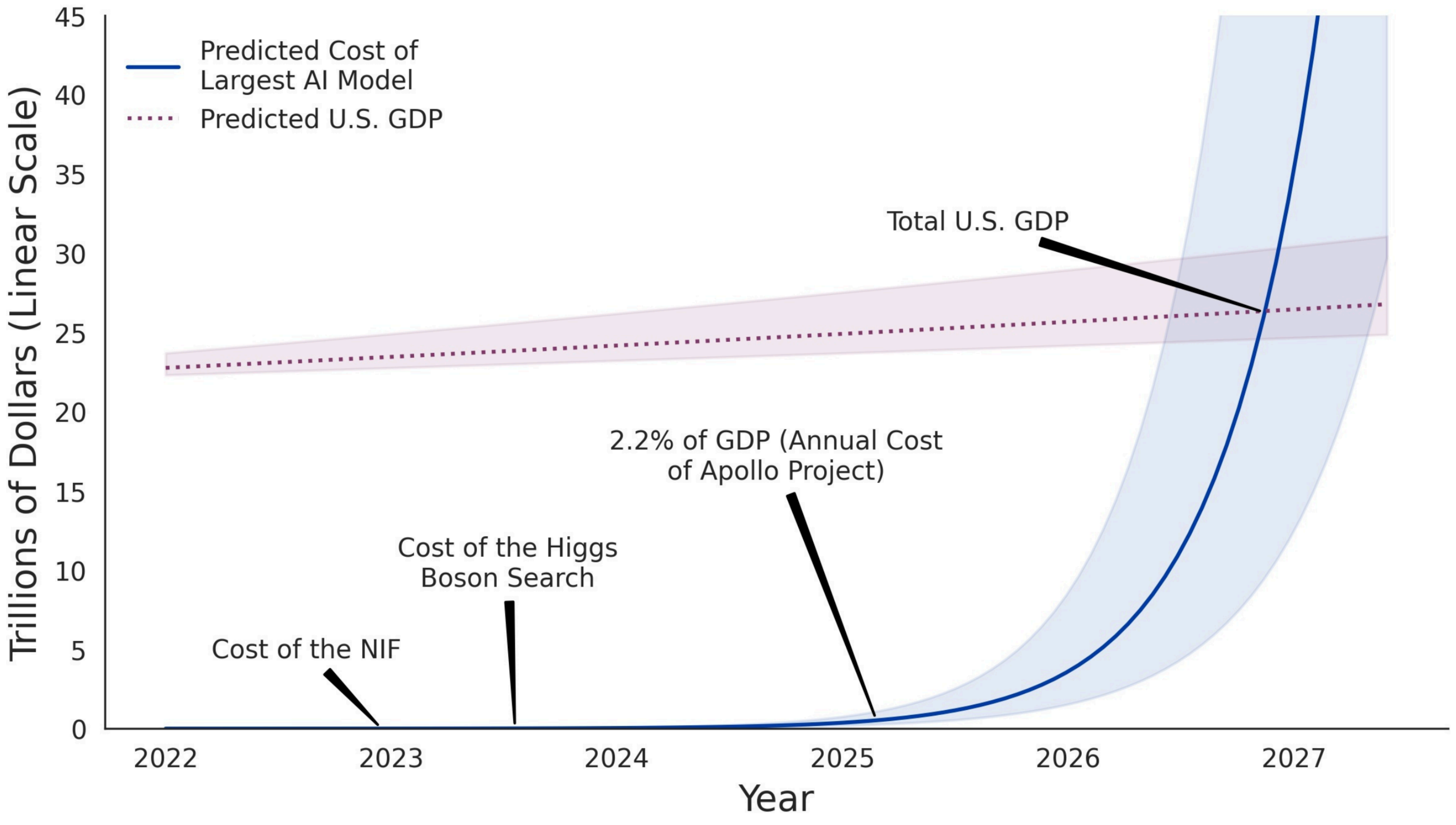
Henighan et al, 2020



Economic infeasibility

https://cset.georgetown.edu/wp-content/uploads/AI-and-Compute-How-Much-Longer-Can-Computing-Power-Drive-Artificial-Intelligence-Progress_v2.pdf





Massive text models

Scaling models also scales author lists

Language Models are Few-Shot Learners

Tom B. Brown[*] Benjamin Mann[*] Nick Ryder[*] Melanie Subbiah[*]
Jared Kaplan[†] Prafulla Dhariwal Arvind Neelakantan Pranav Shyam Girish Sastry
Amanda Askell Sandhini Agarwal Ariel Herbert-Voss Gretchen Krueger Tom Henighan
Rewon Child Aditya Ramesh Daniel M. Ziegler Jeffrey Wu Clemens Winter
Christopher Hesse Mark Chen Eric Sigler Mateusz Litwin Scott Gray
Benjamin Chess Jack Clark Christopher Berner
Sam McCandlish Alec Radford Ilya Sutskever Dario Amodei

Scaling Language Models: Methods, Analysis & Insights from Training *Gopher*

Jack W. Rae, Sebastian Borgeaud, Trevor Cai, Katie Millican, Jordan Hoffmann, Francis Song, John Aslanides, Sarah Henderson, Roman Ring, Susannah Young, Eliza Rutherford, Tom Hennigan, Jacob Menick, Albin Cassirer, Richard Powell, George van den Driessche, Lisa Anne Hendricks, Maribeth Rauh, Po-Sen Huang, Amelia Glaese, Johannes Welbl, Sumanth Dathathri, Saffron Huang, Jonathan Uesato, John Mellor, Irina Higgins, Antonia Creswell, Nat McAleese, Amy Wu, Erich Elsen, Siddhant Jayakumar, Elena Buchatskaya, David Budden, Esme Sutherland, Karen Simonyan, Michela Paganini, Laurent Sifre, Lena Martens, Xiang Lorraine Li, Adhiguna Kuncoro, Aida Nematzadeh, Elena Gribovskaya, Domenic Donato, Angeliki Lazaridou, Arthur Mensch, Jean-Baptiste Lespiau, Maria Tsimpoukelli, Nikolai Grigorev, Doug Fritz, Thibault Sottiaux, Mantas Pajarskas, Toby Pohlen, Zhitao Gong, Daniel Toyama, Cyprien de Masson d'Autume, Yujia Li, Tayfun Terzi, Vladimir Mikulik, Igor Babuschkin, Aidan Clark, Diego de Las Casas, Aurelia Guy, Chris Jones, James Bradbury, Matthew Johnson, Blake Hechtman, Laura Weidinger, Iason Gabriel, William Isaac, Ed Lockhart, Simon Osindero, Laura Rimell, Chris Dyer, Oriol Vinyals, Kareem Ayoub, Jeff Stanway, Lorrayne Bennett, Demis Hassabis, Koray Kavukcuoglu and Geoffrey Irving

Using DeepSpeed and Megatron to Train Megatron-Turing NLG 530B, A Large-Scale Generative Language Model

Shaden Smith[§,†], Mostofa Patwary[§,‡], Brandon Norick[†], Patrick LeGresley[‡], Samyam Rajbhandari[†], Jared Casper[‡], Zhun Liu[†], Shrimai Prabhumoye[‡], George Zerveas[*†], Vijay Korthikanti[‡], Elton Zhang[†], Rewon Child[‡], Reza Yazdani Aminabadi[†], Julie Bernauer[‡], Xia Song[†], Mohammad Shoeybi[‡], Yuxiong He[†], Michael Houston[‡], Saurabh Tiwary[†], and Bryan Catanzaro[‡]

PaLM: Scaling Language Modeling with Pathways

Aakanksha Chowdhery[*] Sharan Narang[*] Jacob Devlin[*]
Maarten Bosma Gaurav Mishra Adam Roberts Paul Barham
Hyung Won Chung Charles Sutton Sebastian Gehrmann Parker Schuh Kensen Shi
Sasha Tsvyashchenko Joshua Maynez Abhishek Rao[†] Parker Barnes Yi Tay
Noam Shazeer[‡] Vinodkumar Prabhakaran Emily Reif Nan Du Ben Hutchinson
Reiner Pope James Bradbury Jacob Austin Michael Isard Guy Gur-Ari
Pengcheng Yin Toju Duke Anselm Levskaya Sanjay Ghemawat Sunipa Dev
Henryk Michalewski Xavier Garcia Vedant Misra Kevin Robinson Liam Fedus
Denny Zhou Daphne Ippolito David Luan[‡] Hyeontaek Lim Barret Zoph
Alexander Spiridonov Ryan Sepassi David Dohan Shivani Agrawal Mark Omernick
Andrew M. Dai Thanumalayan Sankaranarayana Pillai Marie Pellat Aitor Lewkowycz
Erica Moreira Rewon Child Oleksandr Polozov[†] Katherine Lee Zongwei Zhou
Xuezhi Wang Brennan Saeta Mark Diaz Orhan Firat Michele Catasta[†] Jason Wei
Kathy Meier-Hellstern Douglas Eck Jeff Dean Slav Petrov Noah Fiedel

GPT-3

Brown et al, 2020

- 175B parameter transformer decoder, combined training set is ~500B tokens (though the model is only actually trained for 300B tokens)

Model Name	n_{params}	n_{layers}	d_{model}	n_{heads}	d_{head}	Batch Size	Learning Rate
GPT-3 Sma							0×10^{-4}
GPT-3 Me							0×10^{-4}
GPT-3 Lar							5×10^{-4}
GPT-3 XL							0×10^{-4}
GPT-3 2.7I							6×10^{-4}
GPT-3 6.7I							2×10^{-4}
GPT-3 13B							0×10^{-4}
GPT-3 175B or "GPT-3"	175.0B	96	12288	96	128	3.2M	0.6×10^{-4}

Dataset	Quantity (tokens)	Weight in training mix	Epochs elapsed when training for 300B tokens
Common Crawl (filtered)	410 billion	60%	0.44
WebText2	19 billion	22%	2.9
Books1	12 billion	8%	1.9
Books2	55 billion	8%	0.43
Wikipedia	3 billion	3%	3.4

GPT-3

Brown et al, 2020

- GPT-3 demonstrates impressive *few-shot* learning performance, though there is still room for significant improvement

Zero-shot

The model predicts the answer given only a natural language description of the task. No gradient updates are performed.

`Translate English to French:`	*task description*
`cheese =>`	*prompt*

One-shot

In addition to the task description, the model sees a single example of the task. No gradient updates are performed.

`Translate English to French:`	*task description*
`sea otter => loutre de mer`	*example*
`cheese =>`	*prompt*

Few-shot

In addition to the task description, the model sees a few examples of the task. No gradient updates are performed.

`Translate English to French:`	*task description*
`sea otter => loutre de mer`	*examples*
`peppermint => menthe poivrée`	
`plush girafe => girafe peluche`	
`cheese =>`	*prompt*

Gopher

Rae et al, 2021

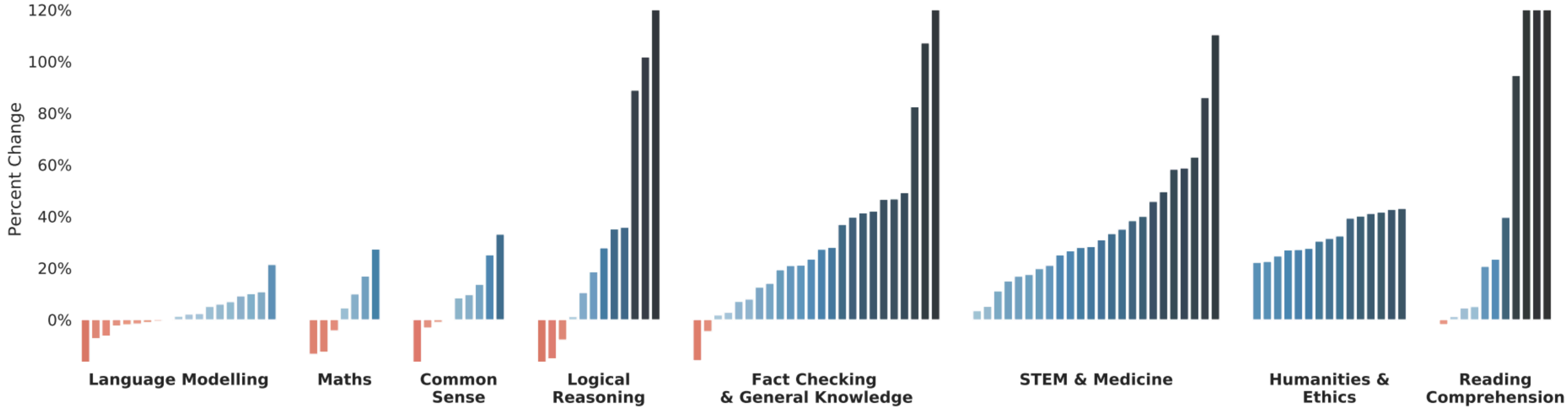
- Gopher is a 280B parameter transformer decoder
- The training set has over 2T tokens, the model is still only trained for 300B tokens

	Disk Size	Documents	Tokens	Sampling proportion
MassiveWeb	1.9 TB	604M	506B	48%
Books	2.1 TB	4M	560B	27%
C4	0.75 TB	361M	182B	10%
News	2.7 TB	1.1B	676B	10%
GitHub	3.1 TB	142M	422B	3%
Wikipedia	0.001 TB	6M	4B	2%

Model						**Batch Size**
44M						0.25M
117M						0.25M
417M						0.25M
1.4B						0.25M
7.1B						2M
Gopher 280B	80	128	128	16,384	4×10^{-5}	3M → 6M

Gopher

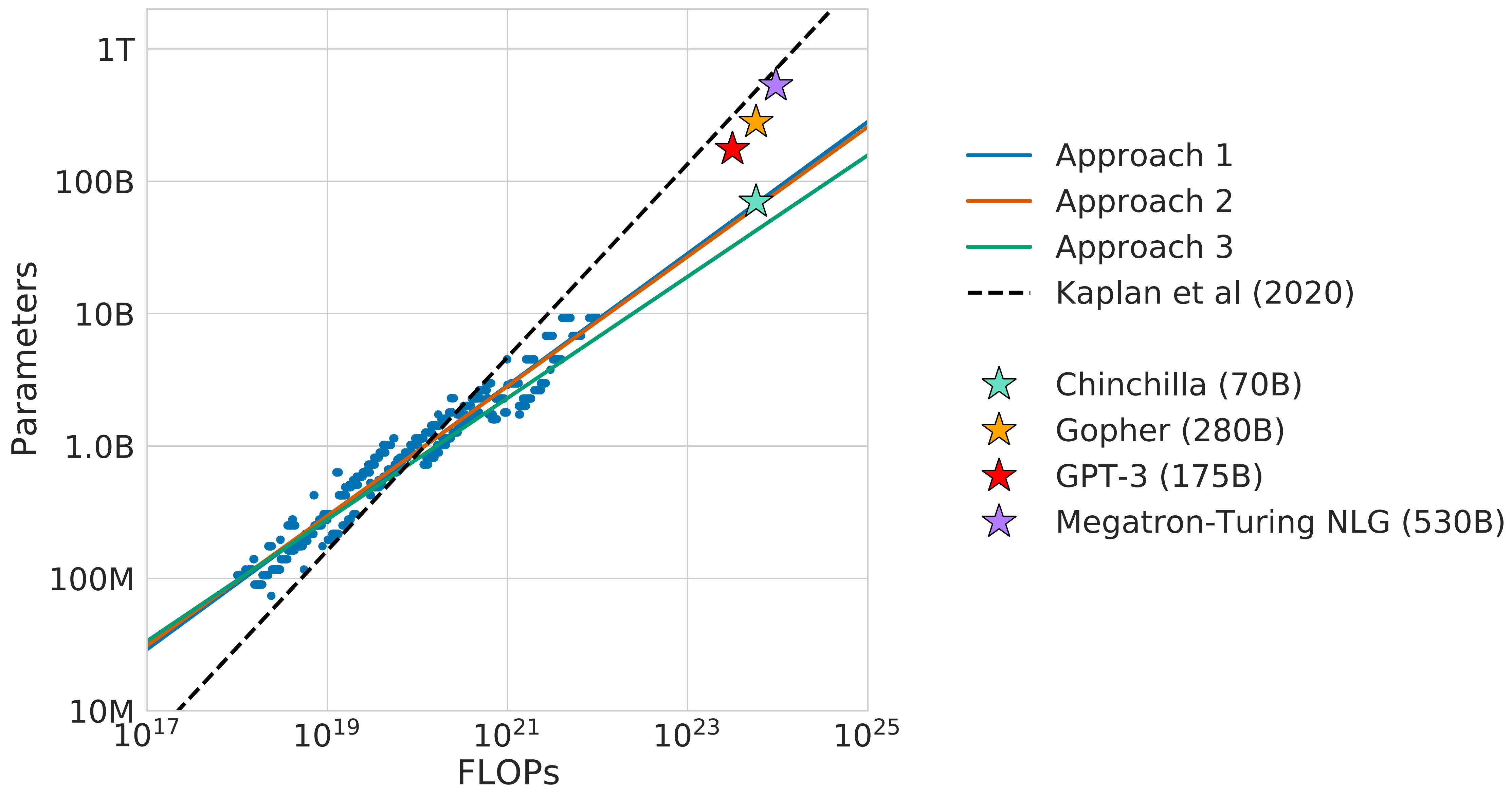
Rae et al, 2021



Chinchilla: a “smaller Gopher”

Hoffmann et al, 2022

- Chinchilla considers varying hyperparameters (primarily, learning rate schedule) that Kaplan et al held fixed, which leads to different scaling conclusions
- In particular, they advocate that model and data size scaling should be **equal**



Chinchilla performance

On Massive Multitask Language Understanding (MMLU)

- The MMLU benchmark contains exam questions from 57 academic subjects, ranging from elementary to professional level difficulty

Task	*Chinchilla*	*Gopher*	Task	*Chinchilla*	*Gopher*
abstract_algebra	31.0	25.0	anatomy	70.4	56.3
astronomy	73.0	65.8	business_ethics	72.0	70.0
clinical_knowledge	75.1	67.2	college_biology	79.9	70.8
college_chemistry	51.0	45.0	college_computer_science	51.0	49.0
college_mathematics	32.0	37.0	college_medicine	66.5	60.1
college_physics	46.1	34.3	computer_security	76.0	65.0
conceptual_physics	67.2	49.4	econometrics	38.6	43.0
electrical_engineering	62.1	60.0	elementary_mathematics	41.5	33.6
formal_logic	33.3	35.7	global_facts	39.0	38.0
high_school_biology	80.3	71.3	high_school_chemistry	58.1	47.8
high_school_computer_science	58.0	54.0	high_school_european_history	78.8	72.1
high_school_geography	86.4	76.8	high_school_gov_and_politics	91.2	83.9
high_school_macroeconomics	70.5	65.1	high_school_mathematics	31.9	23.7
high_school_microeconomics	77.7	66.4	high_school_physics	36.4	33.8
high_school_psychology	86.6	81.8	high_school_statistics	58.8	50.0
high_school_us_history	83.3	78.9	high_school_world_history	85.2	75.1
human_aging	77.6	66.4	human_sexuality	86.3	67.2
international_law	90.9	77.7	jurisprudence	79.6	71.3
logical_fallacies	80.4	72.4	machine_learning	41.1	41.1
management	82.5	77.7	marketing	89.7	83.3
medical_genetics	69.0	69.0	miscellaneous	84.5	75.7
moral_disputes	77.5	66.8	moral_scenarios	36.5	40.2
nutrition	77.1	69.9	philosophy	79.4	68.8
prehistory	81.2	67.6	professional_accounting	52.1	44.3
professional_law	56.5	44.5	professional_medicine	75.4	64.0
professional_psychology	75.7	68.1	public_relations	73.6	71.8
security_studies	75.9	64.9	sociology	91.0	84.1
us_foreign_policy	92.0	81.0	virology	53.6	47.0
world_religions	87.7	84.2			

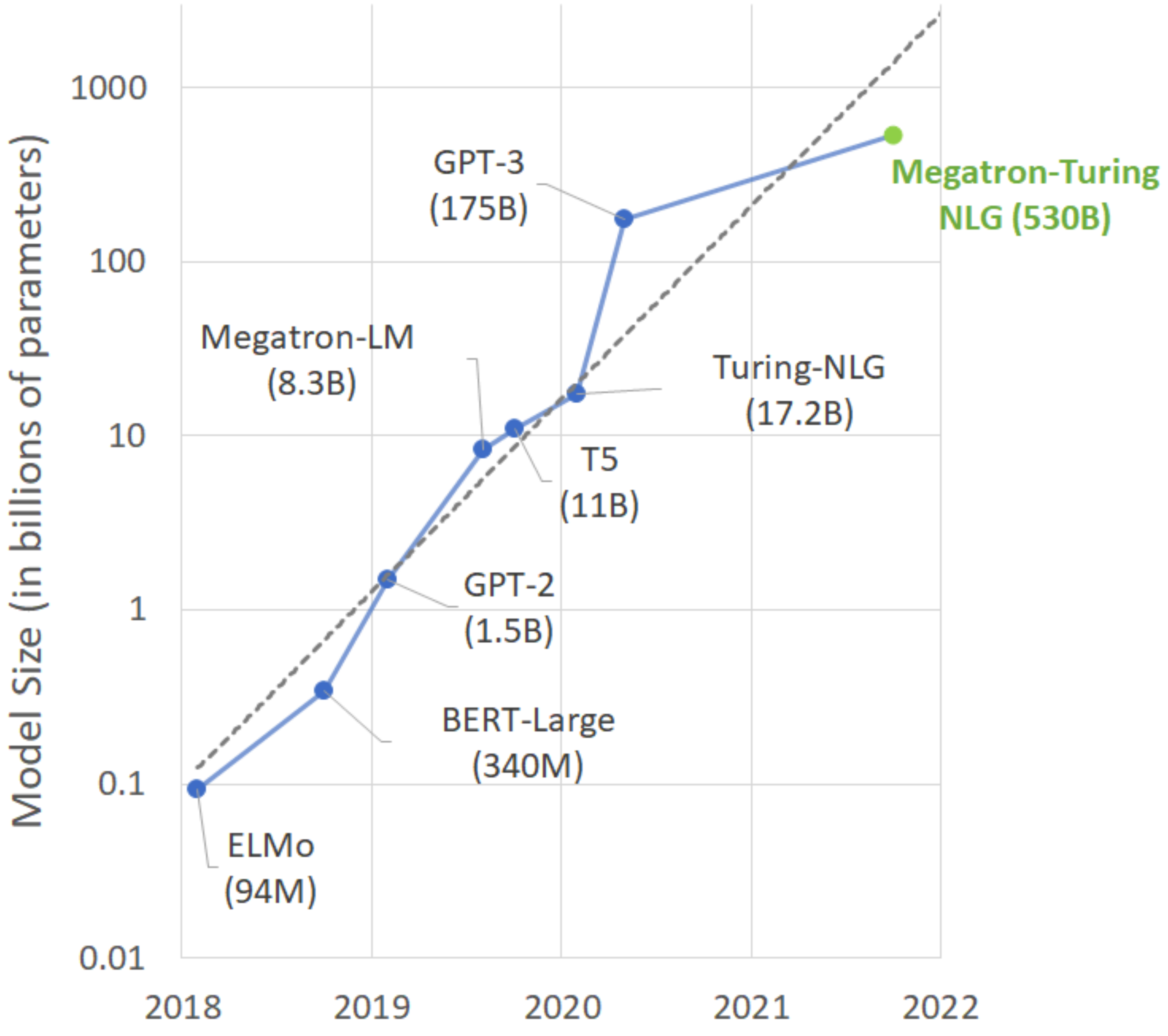
Random	25.0%
Average human rater	34.5%
GPT-3 5-shot	43.9%
Gopher 5-shot	60.0%
***Chinchilla* 5-shot**	**67.6%**
Average human expert performance	*89.8%*
June 2022 Forecast	57.1%
June 2023 Forecast	63.4%

Megatron-Turing NLG

Smith et al, 2022

- MT-NLG is a 530B parameter transformer decoder
- Training set is (a puny) 339B tokens, training is done with 270B tokens

Dataset	Tokens (billion)	Weights (%)	Epochs
Books3	25.7	14.3	1.5
OpenWebText2	14.8	19.3	3.6
Stack Exchange	11.6	5.7	1.4
PubMed Abstracts	4.4	2.9	1.8
Wikipedia	4.2	4.8	3.2
Gutenberg (PG-19)	2.7	0.9	0.9
BookCorpus2	1.5	1.0	1.8
NIH ExPorter	0.3	0.2	1.8
ArXiv	20.8	1.4	0.2
GitHub	24.3	1.6	0.2
Pile-CC	49.8	9.4	0.5
CC-2020-50	68.7	13.0	0.5
CC-2021-04	82.6	15.7	0.5
Realnews	21.9	9.0	1.1
CC-Stories	5.3	0.9	0.5



PaLM

Chowdhery et al, 2022

- PaLM is a 540B parameter (yes, you guessed it) transformer decoder
- Training set: 780B tokens; training: one full epoch!

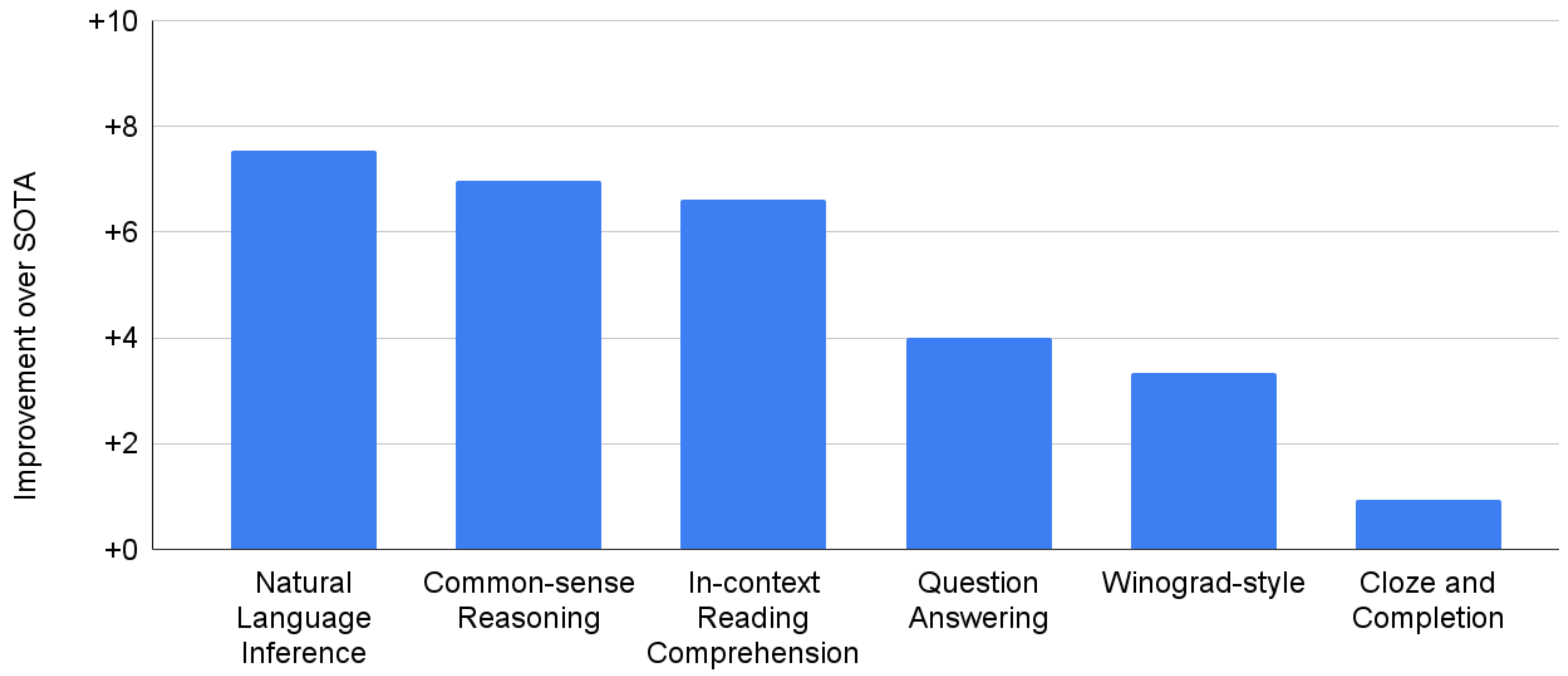
Model	Layers	# of Heads	d_{model}	# of Parameters (in billions)	Batch Size
PaLM 8B	32	16	4096	8.63	$256 \rightarrow 512$
PaLM 62B	64	32	8192	62.50	$512 \rightarrow 1024$
PaLM 540B	118	48	18432	540.35	$512 \rightarrow 1024 \rightarrow 2048$

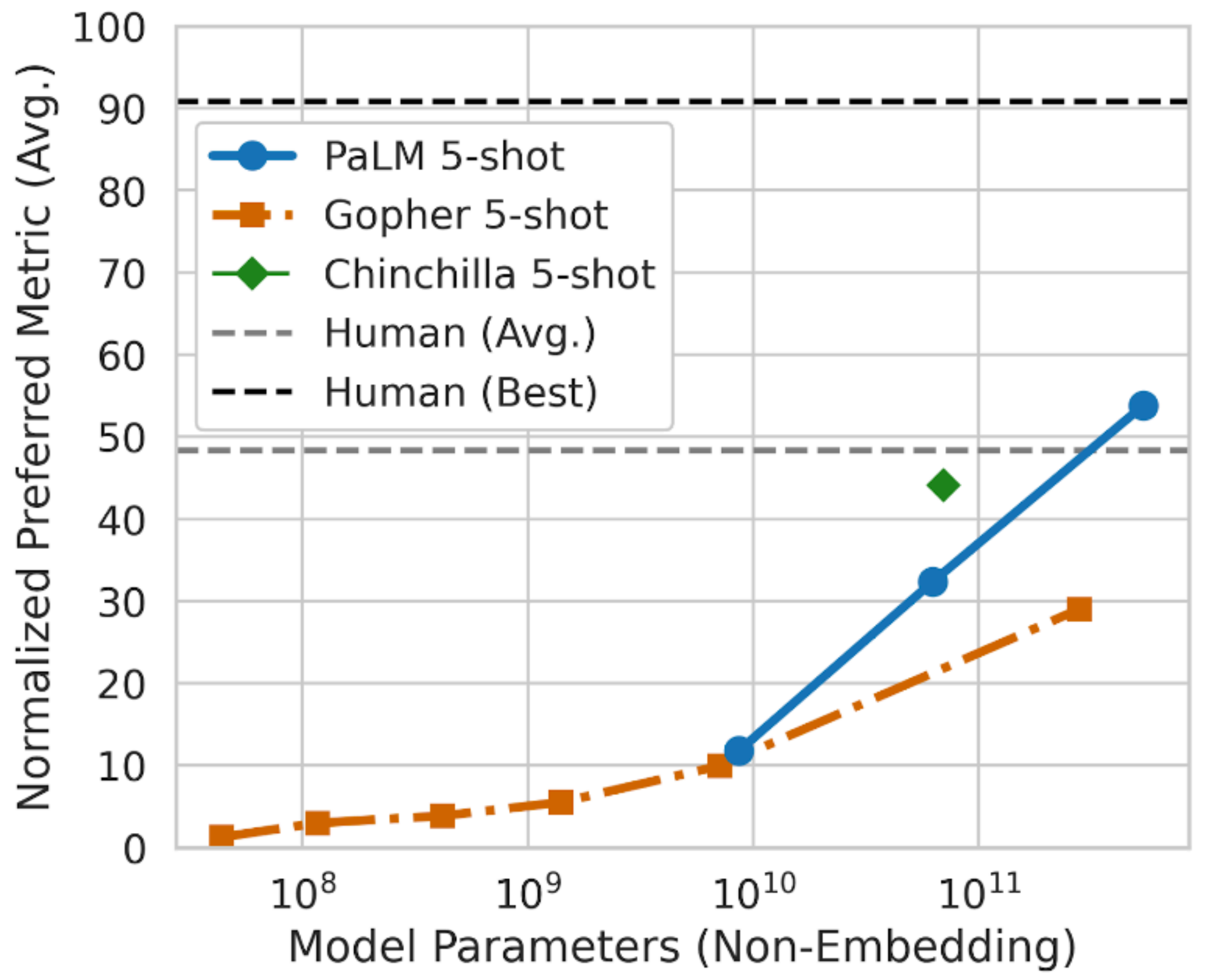
Total dataset size = 780 billion tokens	
Data source	Proportion of data
Social media conversations (multilingual)	50%
Filtered webpages (multilingual)	27%
Books (English)	13%
GitHub (code)	5%
Wikipedia (multilingual)	4%
News (English)	1%

PaLM

Chowdhery et al, 2022

- PaLM improves across a number of natural language tasks, including the recently proposed **BIG-Bench** (right), a recently proposed benchmark of >200 tasks designed for evaluating large language models





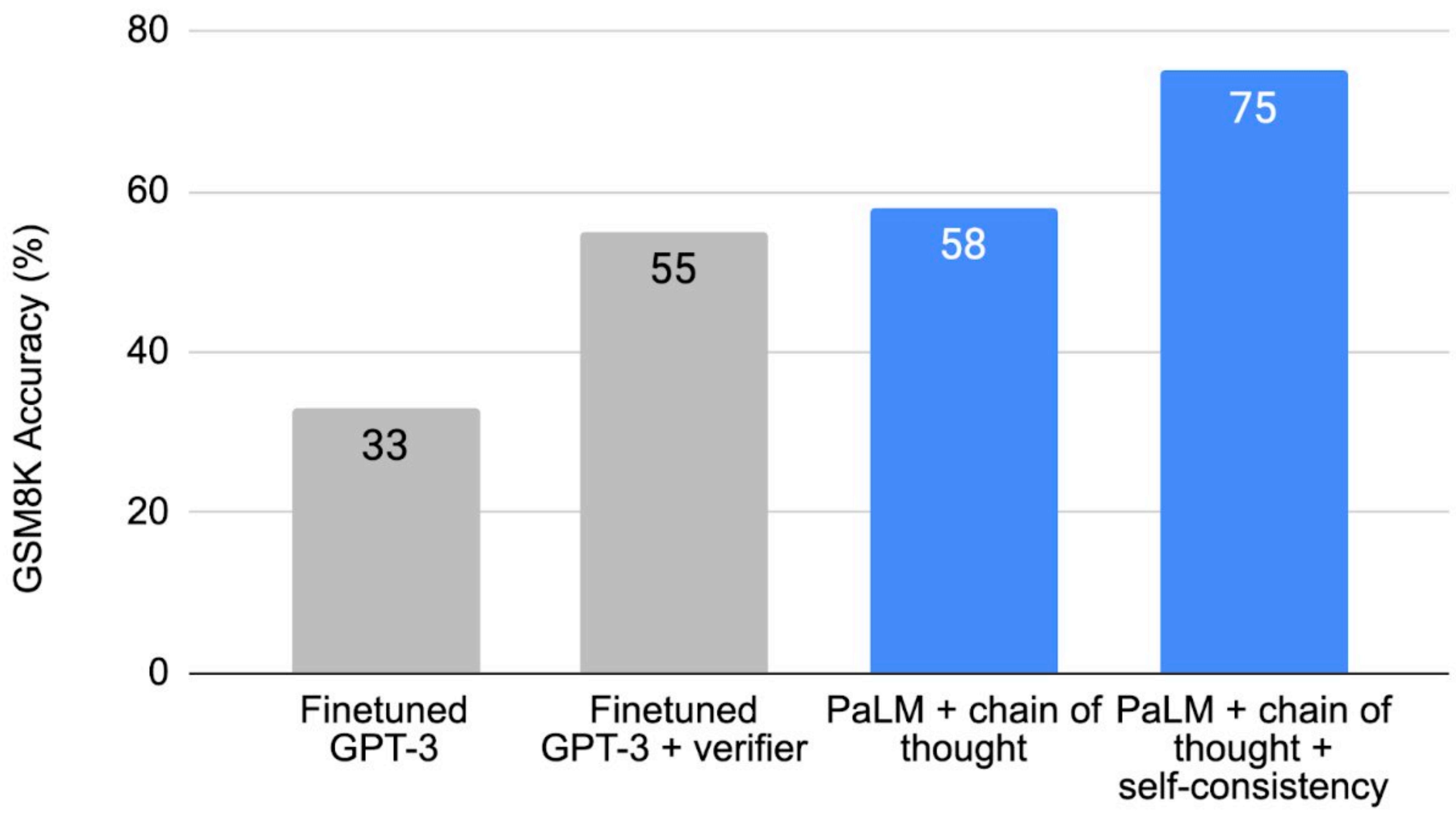
Applications of massive models

Few-shot learning via prompting

- GPT-3's **prompting** strategy has also been adopted by other large models: provide some number of examples of the task within the model's input itself
  - Often, the number of examples is less than 10
- **Prompt engineering** is now an important part of getting large models to perform their best, and this often requires some fiddling to get right
- PaLM sometimes uses **chain-of-thought prompting**, which not only prompts with the correct answer but also the process by which that answer is reached
  - This significantly improves performance for some more complex tasks

PaLM + chain-of-thought + self-consistency

- Further combined with **self-consistency** (sampling multiple answers and picking the most consistent answer), PaLM + chain-of-thought results in substantial improvements on eighth grade arithmetic word problems (GSM8K)



PaLM can explain jokes with chain-of-thought prompting

Explaining a joke

Prompt

```
Explain this joke:

Joke: Did you see that Google just hired an eloquent whale for
their TPU team? It showed them how to communicate between two
different pods!
```

Model Response

```
Prediction: TPUs are a type of computer chip that Google uses
for deep learning. A "pod" is a group of TPUs. A "pod" is also a
group of whales. The joke is that the whale is able to
communicate between two groups of whales, but the speaker is
pretending that the whale is able to communicate between two
groups of TPUs.
```

“Medium-shot” learning with fine tuning

- **Fine tuning** usually refers to updating the model via gradient based optimization with a small dataset (hundreds or thousands of data points)
- With the size of these models, even this can be impractical or even infeasible
  - GPT-3 offers fine tuning as part of the OpenAI API
- When possible, fine tuning still outperforms prompting significantly

Model	BoolQ	CB	CoPA	MultiRC	Record	RTE	WiC	WSC
Few-shot	89.1	89.3	95	86.3/-	92.9/-	81.2	64.6	89.5
Finetuned	92.2	100/100	100	90.1/69.2	94.0/94.6	95.7	78.8	100

Table 7: Results on SuperGLUE dev set comparing PaLM-540B few-shot and finetuned.

Specializing to code: Codex and AlphaCode

- Codex is a 12B parameter model that starts from a (smaller) GPT-3 and fine tunes on 159GB of code from Github
- This model is what powers Github Copilot: https://copilot.github.com/
- AlphaCode scales up capabilities to competition level coding, achieving performance comparable to the median competitor
- Scaling up model size (41B) and dataset size (715GB), changing the model architecture to be an encoder-decoder, fine tuning on competition code, and sampling/filtering many candidate solutions all help in scaling to this level

Limitations of (current) massive models

Challenge tasks

- A number of tasks still elude the largest models and may be beyond the reach of simply making the models even bigger
  - Challenge sets designed to “stress test” models, e.g., ANLI, still have significant room for improvement, and scaling up is making slow progress
  - Hard tasks, such as generating solutions for high school math competition problems, also have very low accuracy even for the largest models
- This is even after these models are trained / fine tuned with more text/code/math than a human will ever see in their lifetime, so it seems like something is missing

Potential harms and biases

- Papers about large models now typically come with a *model card* describing its details and intended uses, along with some analysis about potential harms
- For example, GPT-3 was analyzed for gender, race, and religion biases, and this shed light on its predispositions that (unfortunately) seem in line with its training
    - This analysis has also been carried out for the three other models mentioned
- Analysis on Gopher (and, to an extent, PaLM) demonstrates that large models, when given a *toxic* prompt, are more likely to generate toxic continuations
- These concerns, and more, have to be carefully studied and mitigated before deploying such models into sensitive applications

Summary

- Massive models represent another potential paradigm shift within machine learning: pretraining + fine tuning was one such shift over the last ~10 years, but now perhaps we don’t even need to do fine tuning anymore!
- In some ways, this may be more accessible (fewer data/expertise requirements); in other ways, this may be less accessible (compute requirements, privatization)
- Massive models still have problems in which they struggle, and they are primarily language models at the moment, but this may all change over the next few years
- As these models continue to proliferate, careful auditing of the potential benefits vs. potential harms will be needed to truly understand their full impact