

Prompt-based learning

CS685 Spring 2022

Advanced Natural Language Processing

Mohit Iyyer

College of Information and Computer Sciences
University of Massachusetts Amherst

Many slides from Tu Vu

The language model “scaling wars”!

ELMo: 93M params, 2-layer biLSTM

BERT-base: 110M params, 12-layer Transformer

BERT-large: 340M params, 24-layer Transformer

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}

The language model “scaling wars”!

ELMo: 93M params, 2-layer biLSTM

BERT-base: 110M params, 12-layer Transformer

BERT-large: 340M params, 24-layer Transformer

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	5120	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}

The language model “scaling wars”!

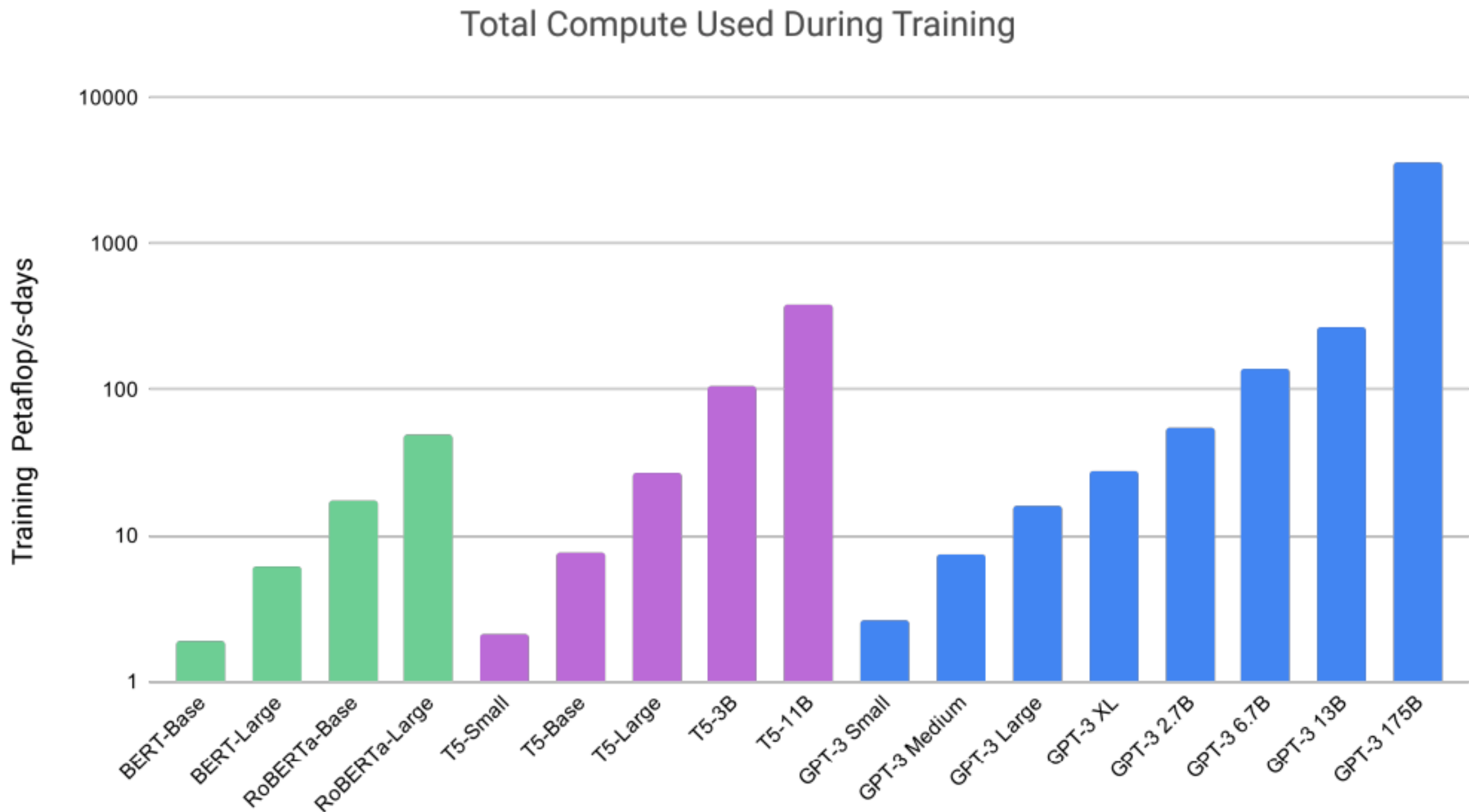
ELMo: 1B training tokens

BERT: 3.3B training tokens

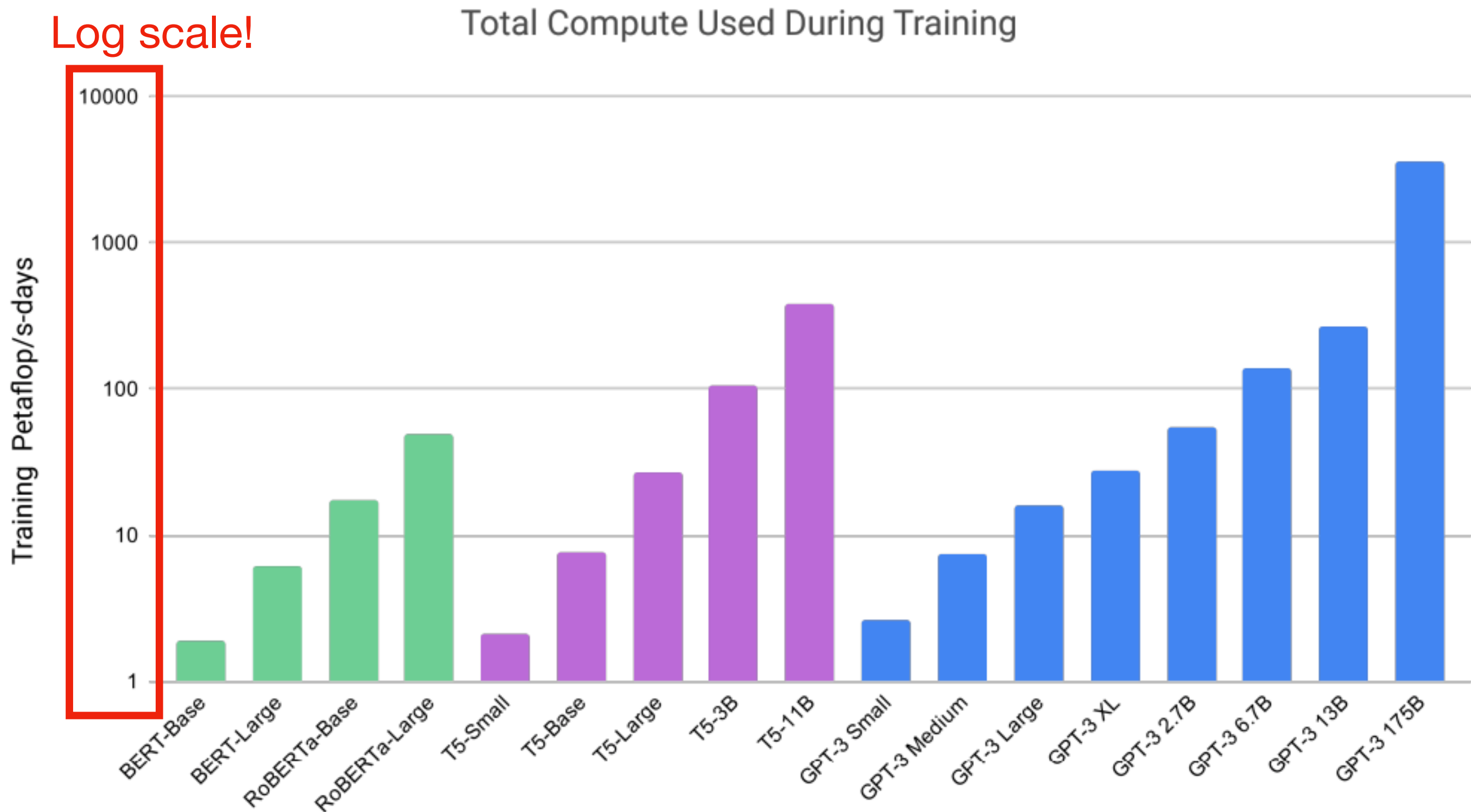
RoBERTa: ~30B training tokens

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

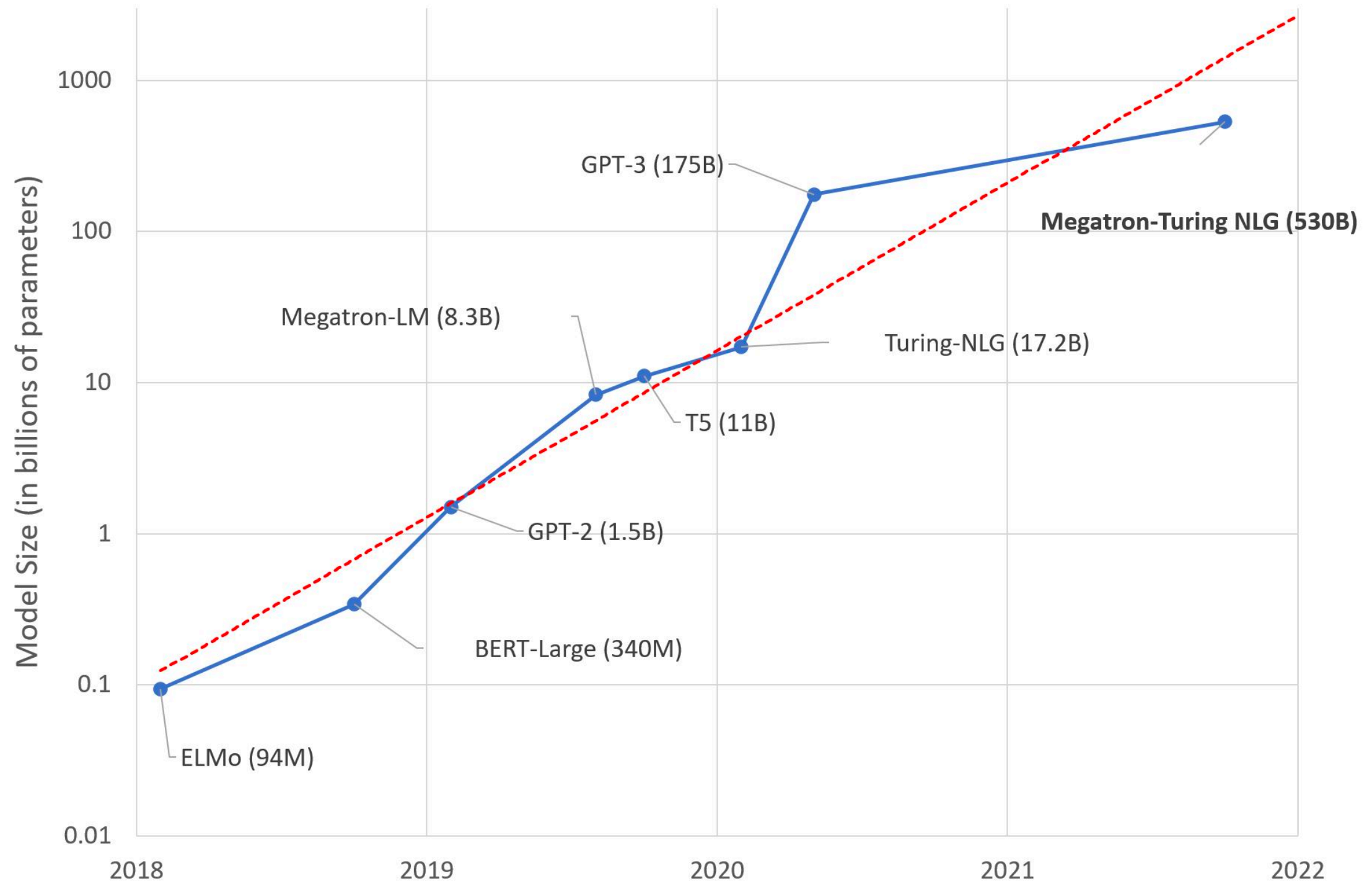
The language model “scaling wars”!



The language model “scaling wars”!



A new 530B param model was released late last year



so... what does all of this scaling buy us?

Traditional fine-tuning (not used for GPT-3)

Fine-tuning

The model is trained via repeated gradient updates using a large corpus of example tasks.



Downstream
training data

Downstream
test data

Zero-shot

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

1	Translate English to French:	← <i>task description</i>
2	cheese =>	← <i>prompt</i>

without seeing any
parallel data.

Zero-shot

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



The diagram shows a light blue rounded rectangle containing two lines of text. The first line is '1 Translate English to French:' and the second line is '2 cheese =>'. To the right of the rectangle, there are two arrows pointing left. The top arrow points to the first line and is labeled 'task description'. The bottom arrow points to the second line and is labeled 'prompt'.

```
1 Translate English to French: ← task description
2 cheese => ..... ← prompt
```

No fine-tuning!!! Literally just take a pretrained LM and give it the following prefix:

“Translate English to French: cheese =>”

One-shot

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



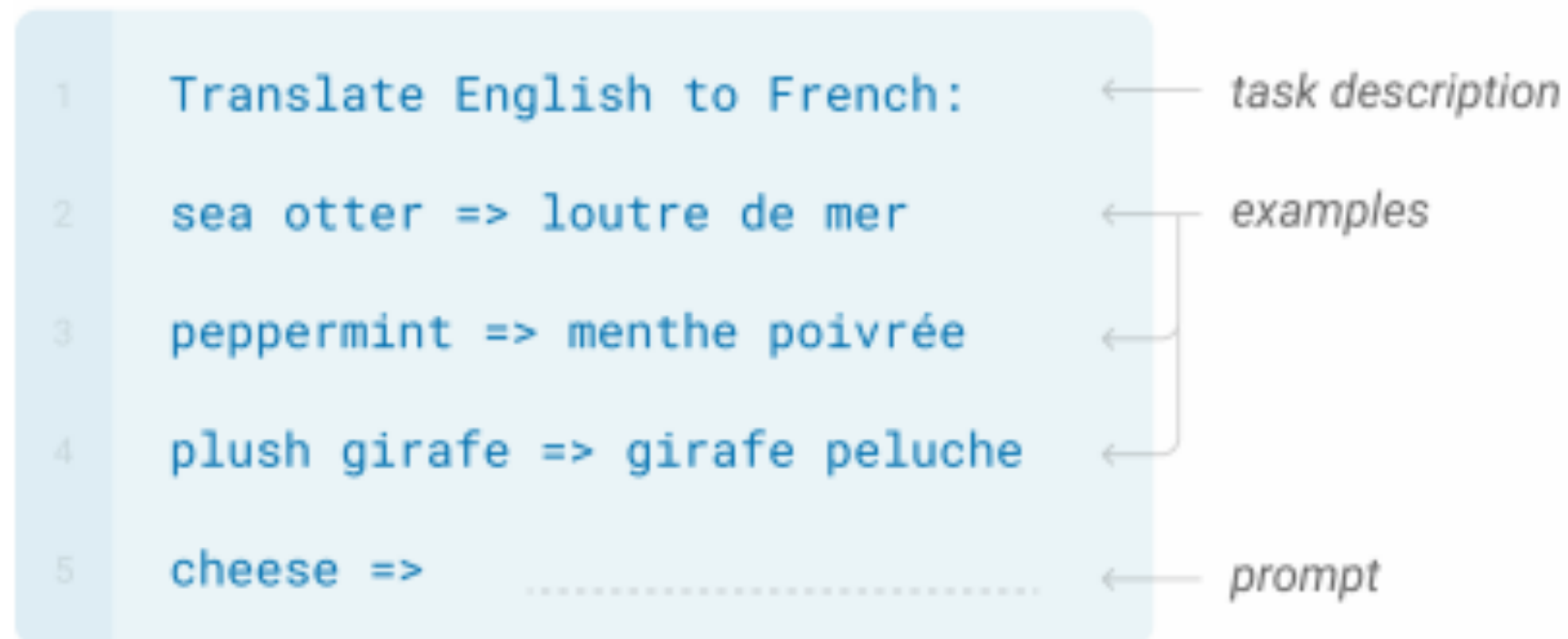
give one example, maybe not sure what is an arrow is

No fine-tuning!!! Literally just take a pretrained LM and give it the following prefix:

“Translate English to French: sea otter => loutre de mer, cheese =>”

Few-shot

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



no gradient being computed, using decoding algorithm. There is no training at all

No fine-tuning!!! Literally just take a pretrained LM and give it the following prefix:

“Translate English to French: sea otter => loutre de mer, peppermint => ... (few more examples), cheese =>”

Max of 100 examples fed into the prefix in this way

Example

How does this new paradigm
compare to “pretrain + finetune”?

TriviaQA

Question

Miami Beach in Florida borders which ocean?

What was the occupation of Lovely Rita according to the song by the Beatles

Who was Poopdeck Pappys most famous son?

The Nazi regime was Germany's Third Reich; which was the first Reich?

At which English racecourse did two horses collapse and die in the parade ring due to electrocution, in February 2011?

Which type of hat takes its name from an 1894 novel by George Du Maurier where the title character has the surname O'Ferrall ?

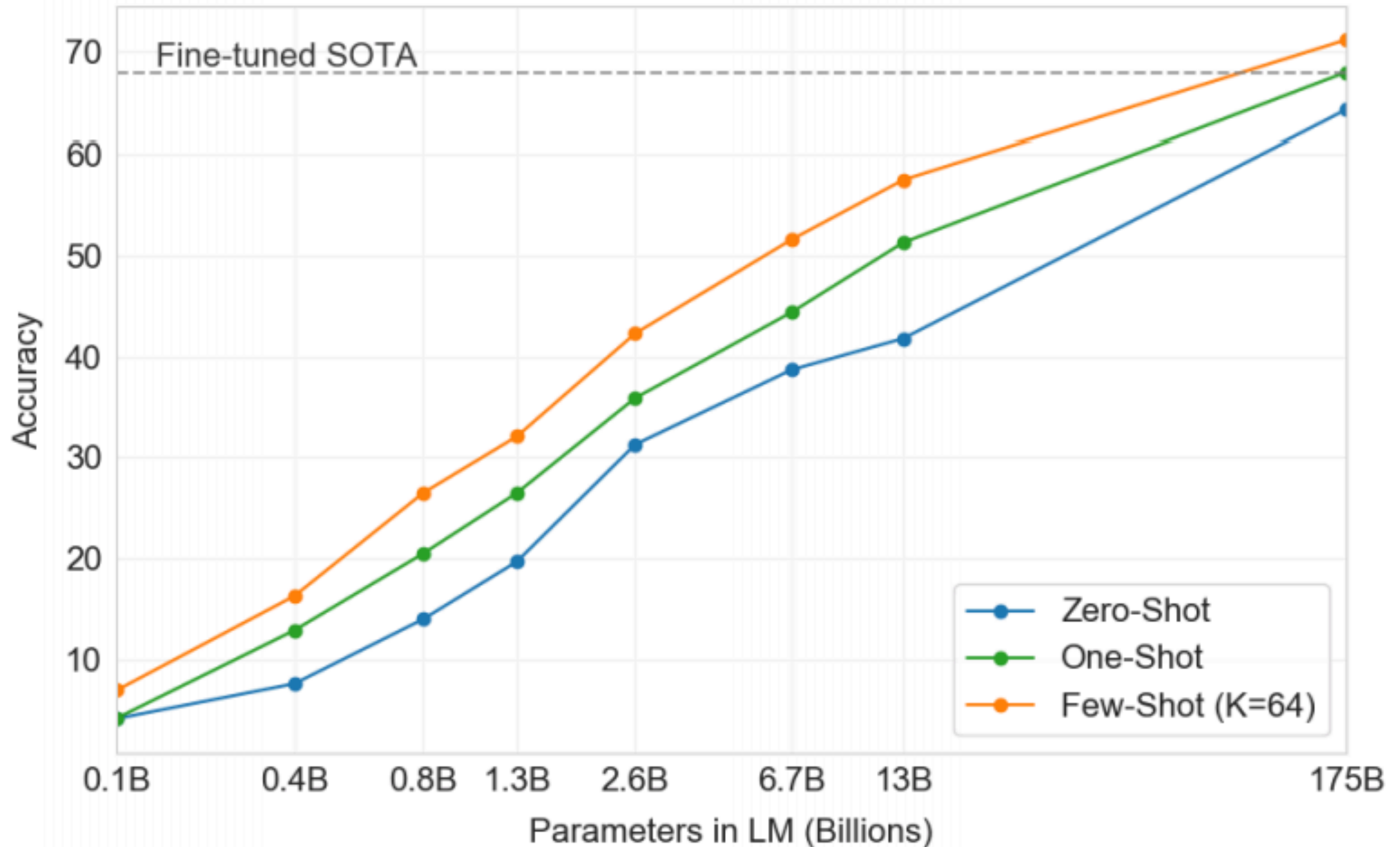
What was the Elephant Man's real name?

trivia question. people will have to know the context

giant pretrain model with no gradient update is actually do better than a fine-tuned model

Bert, Roberta. fine tune to solve the training task.

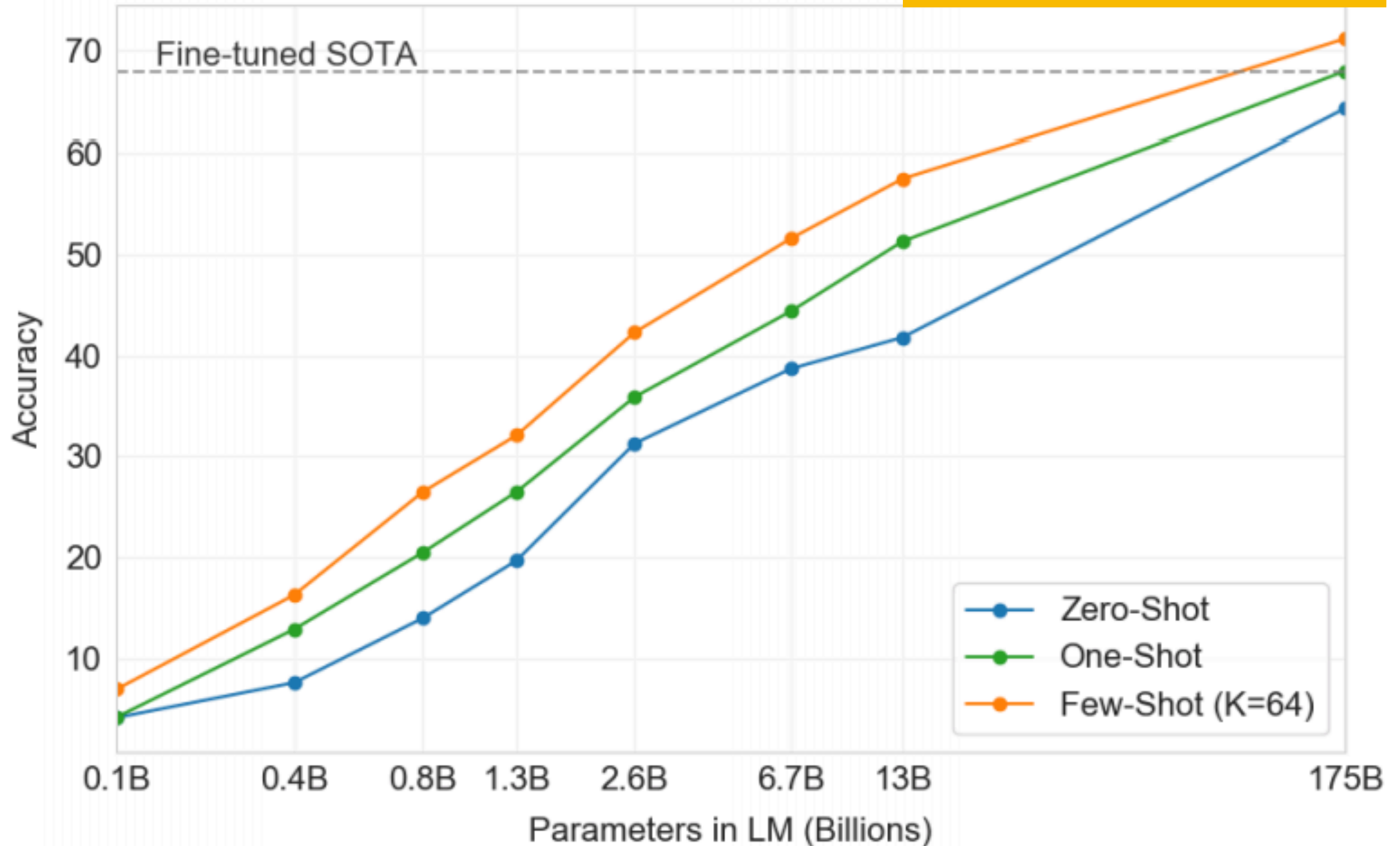
TriviaQA



it is able to memorize, memorization ability work well with trivia task. as we scale up, it gets better to memorize.

TriviaQA

What does this mean?

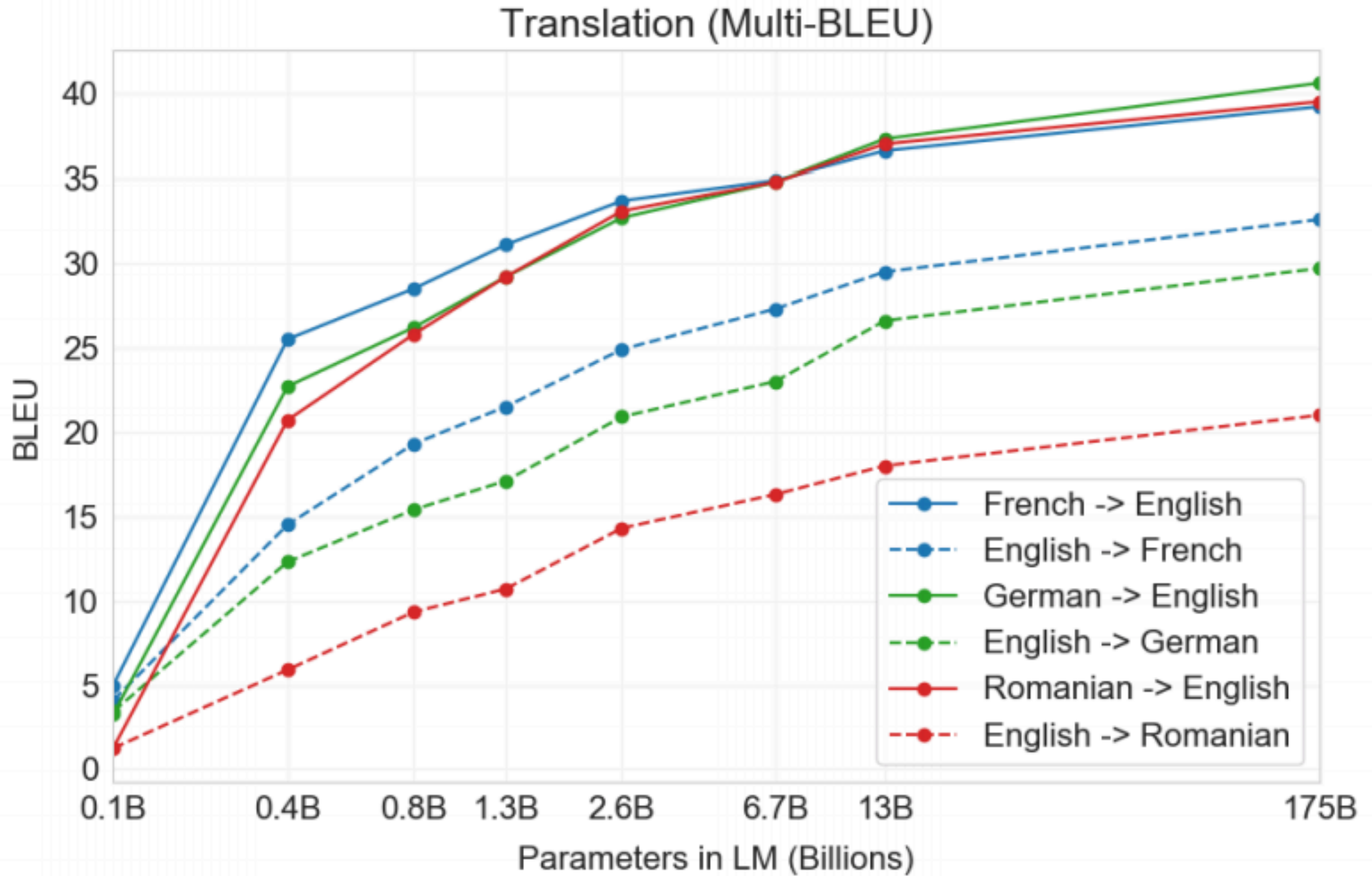


What about translation? (7% of
GPT3's training data is in
languages other than English)

BLEU Score, a measure of how good of a translation.

Setting	En→Fr	Fr→En	En→De	De→En	En→Ro	Ro→En
SOTA (Supervised)	45.6^a	35.0 ^b	41.2^c	40.2 ^d	38.5^e	39.9^e
XLM [LC19]	33.4	33.3	26.4	34.3	33.3	31.8
MASS [STQ ⁺ 19]	<u>37.5</u>	34.9	28.3	35.2	<u>35.2</u>	33.1
mBART [LGG ⁺ 20]	-	-	<u>29.8</u>	34.0	<u>35.0</u>	30.5
GPT-3 Zero-Shot	25.2	21.2	24.6	27.2	14.1	19.9
GPT-3 One-Shot	28.3	33.7	26.2	30.4	20.6	38.6
GPT-3 Few-Shot	32.6	<u>39.2</u>	29.7	<u>40.6</u>	21.0	<u>39.5</u>

Improvements haven't plateaued!

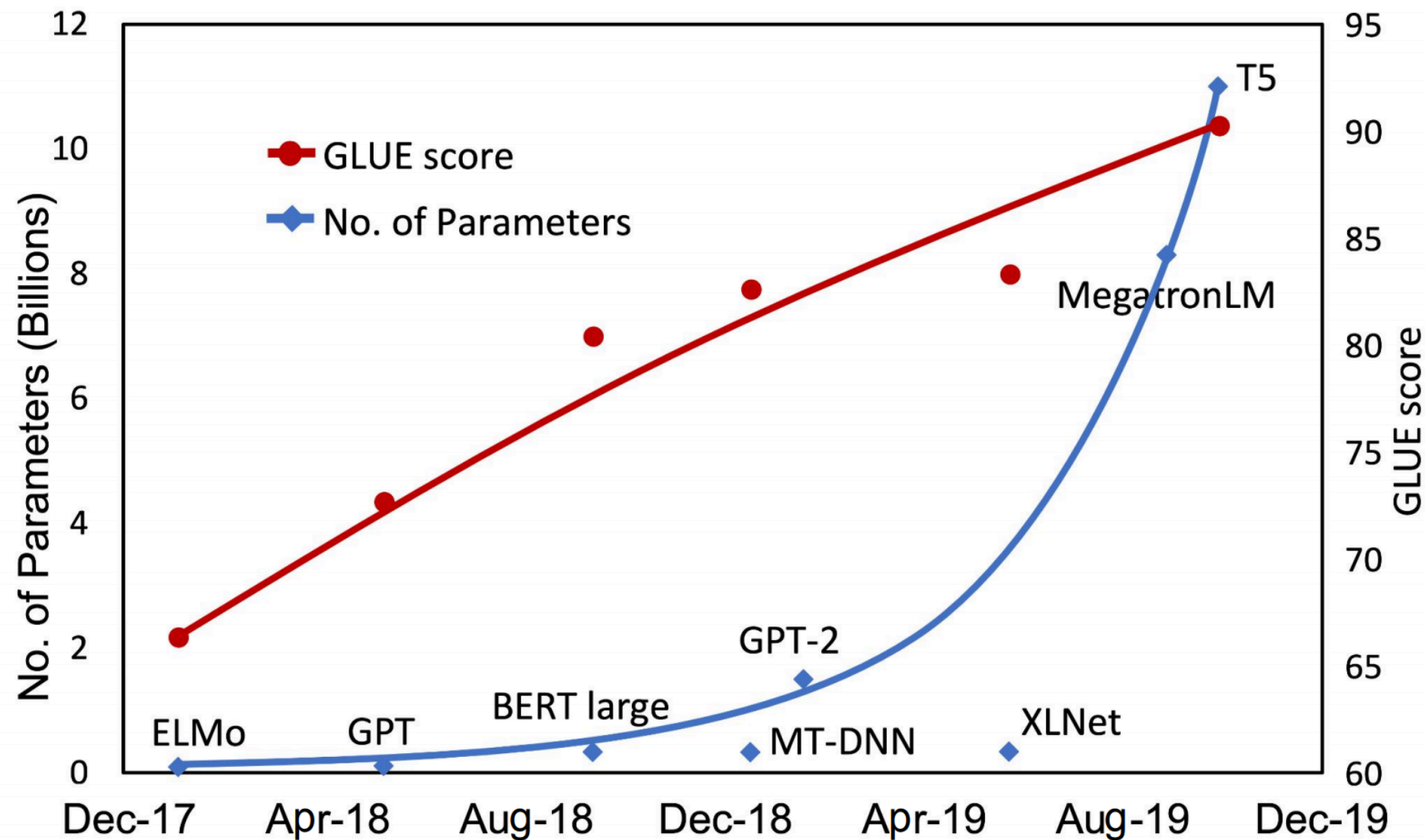


What about reading
comprehension QA?

Setting	CoQA	DROP	QuAC	SQuADv2	RACE-h	RACE-m
Fine-tuned SOTA	90.7^a	89.1^b	74.4^c	93.0^d	90.0^e	93.1^e
GPT-3 Zero-Shot	81.5	23.6	41.5	59.5	45.5	58.4
GPT-3 One-Shot	84.0	34.3	43.3	65.4	45.9	57.4
GPT-3 Few-Shot	85.0	36.5	44.3	69.8	46.8	58.1

Struggles on “harder” datasets

Scaling up the model size is one of the most important ingredients for achieving the best performance

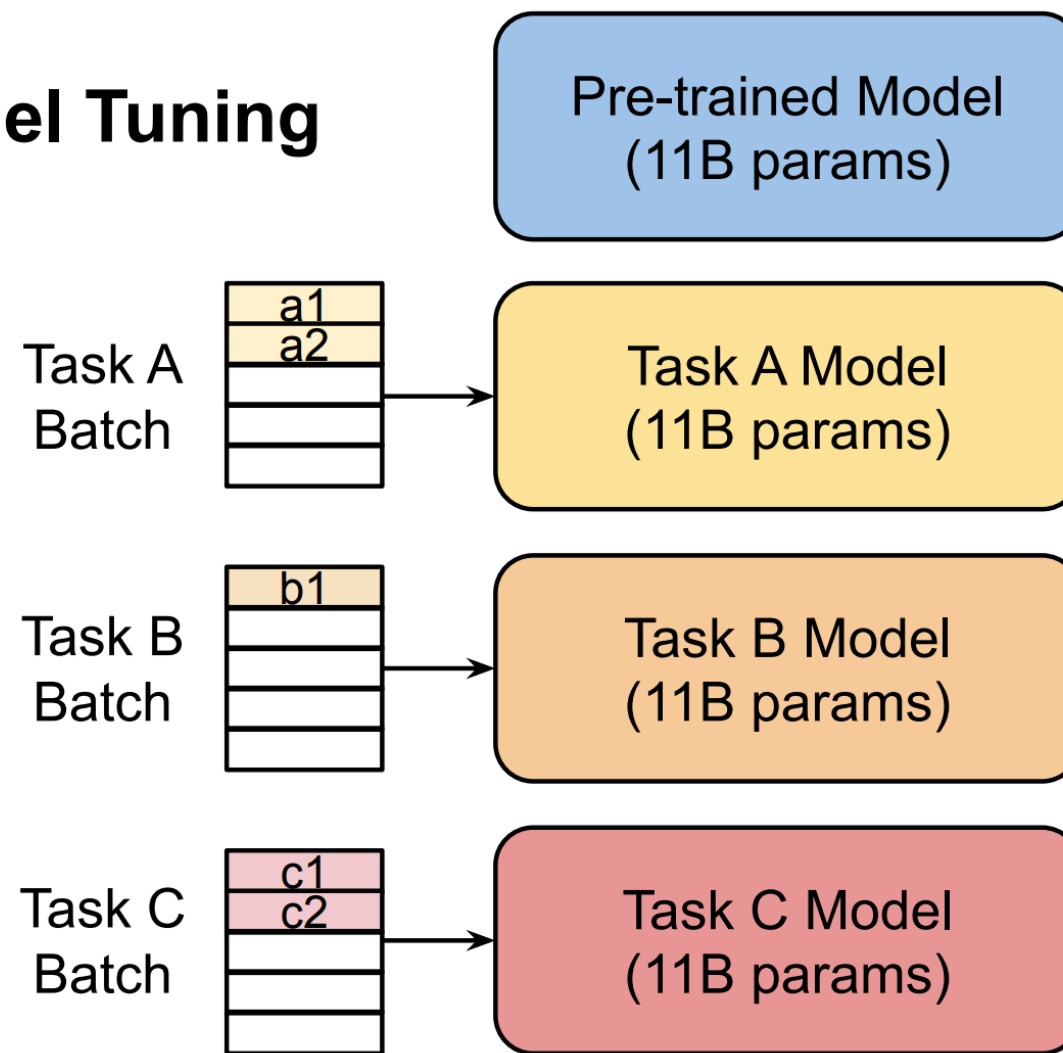


[Ahmet and Abdullah., 2021](#)

GLUE benchmark on NLP classification task. GLUE has been going up as we increase the number of parameters.

Practical challenges: large-scale models are costly to share and serve

Model Tuning



[Lester et al., 2021](#)

Language model prompting to the rescue!

GPT-3 ([Brown et al., 2020](#)): In-context learning

- **natural language instruction** and/or **a few task demonstrations** → **output**

“Translate English to German:” That is good → Das
is gut

- *no* gradient updates or fine-tuning

Sub-optimal and sensitive discrete/hard prompts

Discrete/hard prompts

- natural language instructions/task descriptions

Problems

- requiring domain expertise/understanding of the model's inner workings
- performance still lags far behind SotA model tuning results
- sub-optimal and sensitive
 - prompts that humans consider reasonable is not necessarily effective for language models ([Liu et al., 2021](#))
 - pre-trained language models are sensitive to the choice of prompts ([Zhao et al., 2021](#))

Sub-optimal and sensitive discrete/hard prompts (cont.)

Prompt	P@1
[X] is located in [Y]. (<i>original</i>)	31.29
[X] is located in which country or state? [Y].	19.78
[X] is located in which country? [Y].	31.40
[X] is located in which country? In [Y].	51.08

Table 1. Case study on LAMA-TREx P17 with bert-base-cased. A single-word change in prompts could yield a drastic difference.

[Liu et al., 2021](#)

Shifting from discrete/hard to continuous/soft prompts

Progress in prompt-based learning

- manual prompt design ([Brown et al., 2020](#); [Schick and Schutze, 2021a,b](#))
- mining and paraphrasing based methods to automatically augment the prompt sets ([Jiang et al., 2020](#))
- gradient-based search for improved discrete/hard prompts ([Shin et al., 2020](#))
- automatic prompt generation using a separate generative language model (i.e., T5) ([Gao et al., 2020](#))
- learning continuous/soft prompts ([Liu et al., 2021](#); [Li and Liang., 2021](#); [Qin and Eisner., 2021](#); [Lester et al., 2021](#))

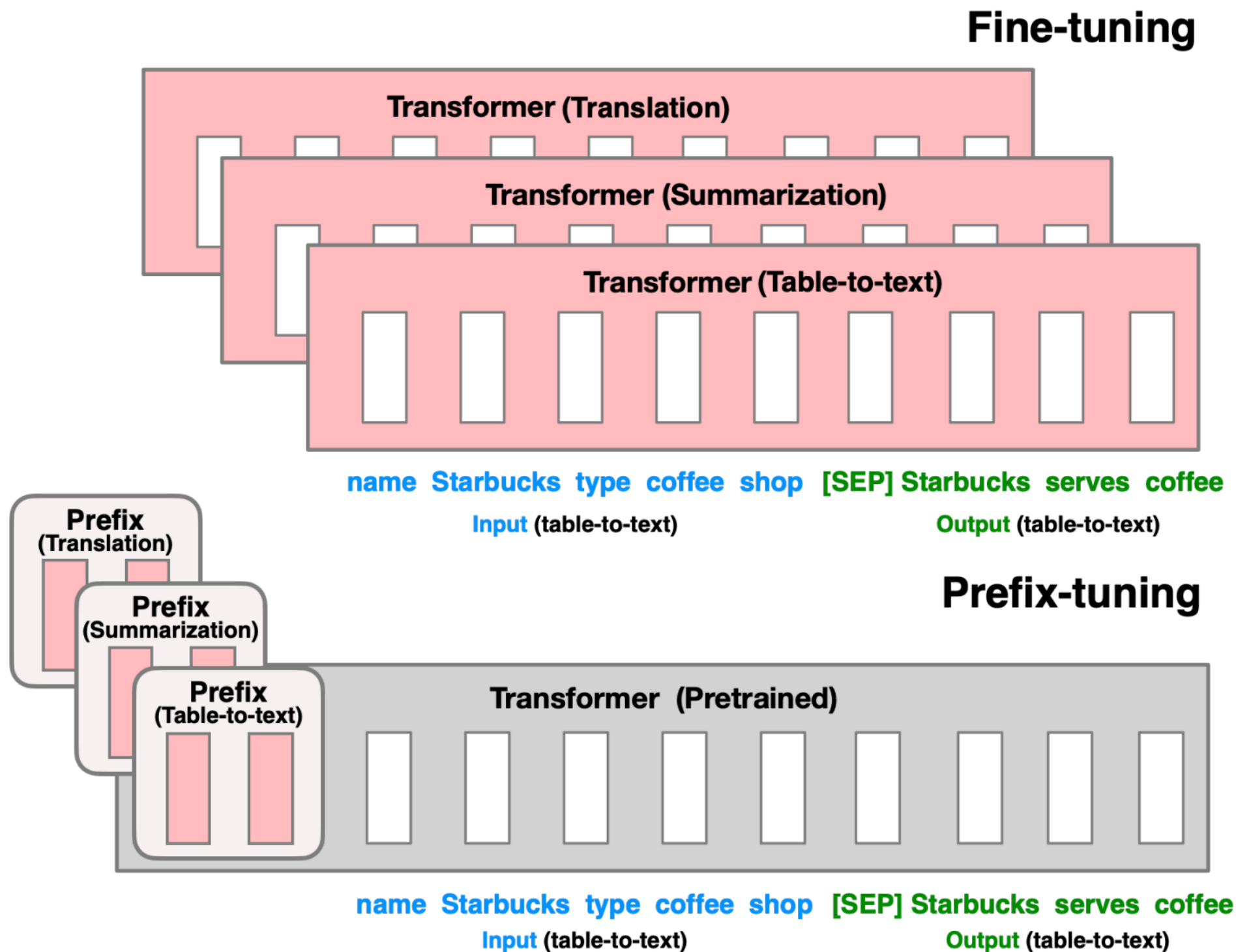
Continuous/soft prompts

- additional learnable parameters injected into the model

It remains unclear how to learn continuous/soft prompts effectively?

- **P-tuning** ([Liu et al., 2021](#)): encode dependencies between prompt tokens using a BiLSTM network
- **P-tuning** ([Liu et al., 2021](#)), **Prefix Tuning** ([Li and Liang., 2021](#)): inject prompts at different positions of the input / model
- **P-tuning** ([Liu et al., 2021](#)): use mixed prompt initialization strategies
- **Soft Prompts** ([Qin and Eisner., 2021](#)): use ensemble methods, e.g., mixture-of-experts

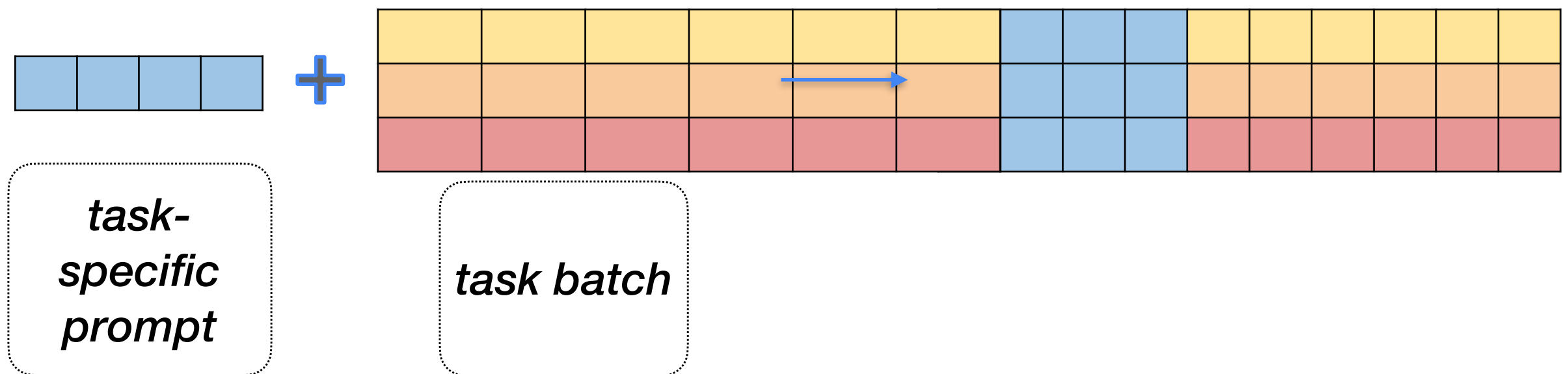
Prefix tuning (Li & Liang, ACL 2021)



Prompt Tuning idea ([Lester et al., 2021](#))

What is a prompt in Prompt Tuning?

- a sequence of additional task-specific tunable tokens prepended to the input text



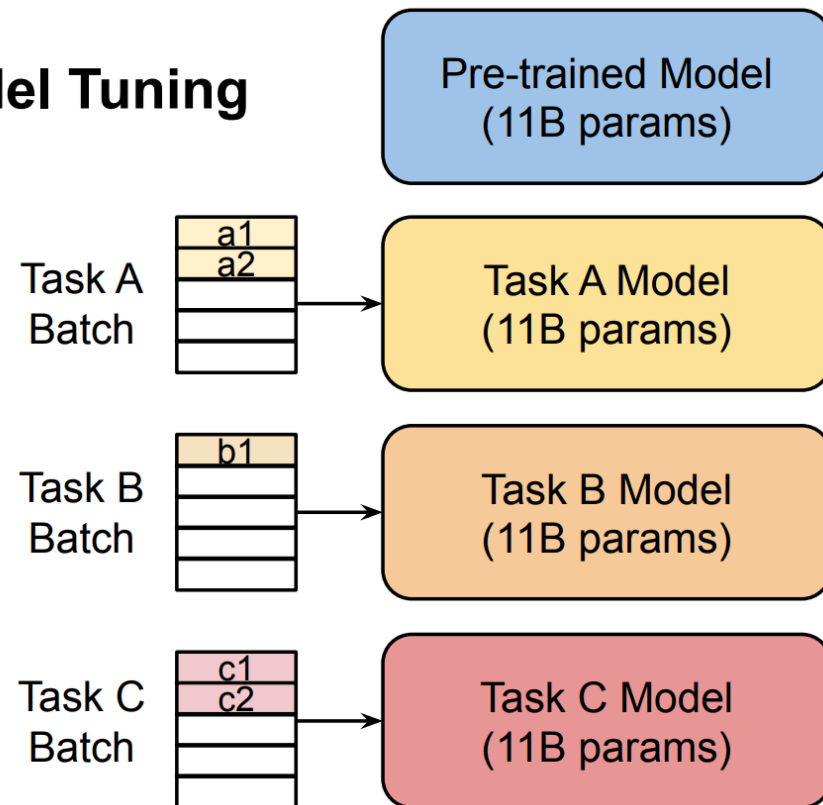
there are not words. Prepending some learned embedding to the model. (Embedding is initialized to a random vectors.)

we can initialize differently

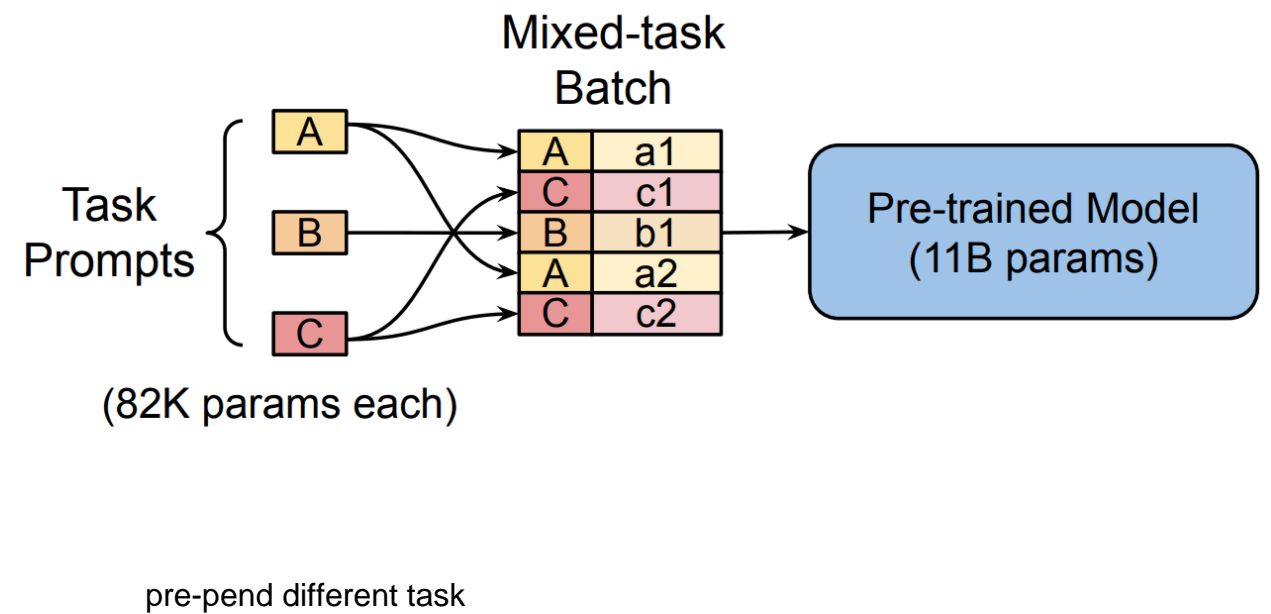
iPad

Parameter-efficient Prompt Tuning

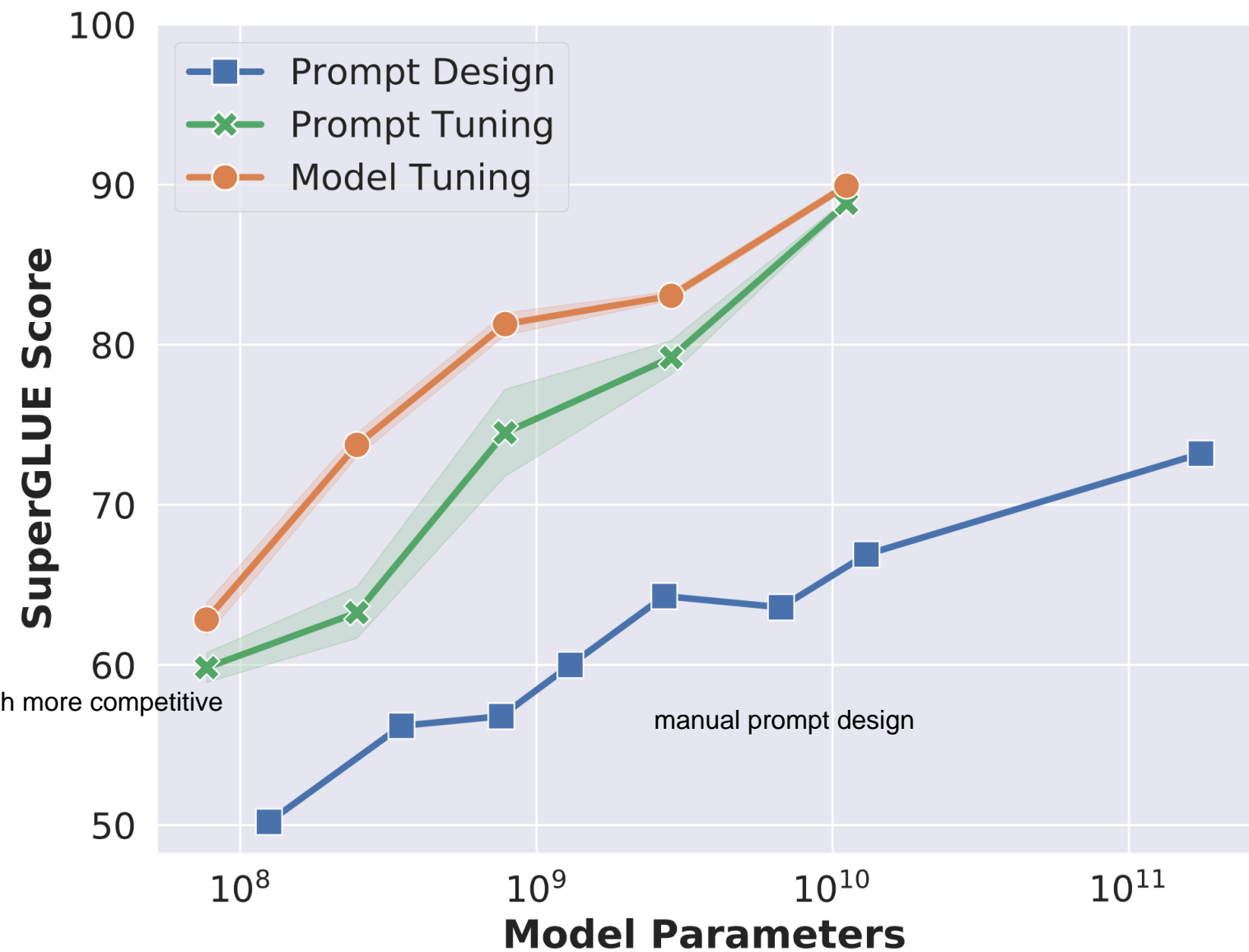
Model Tuning



Prompt Tuning



Prompt Tuning becomes more competitive with scale



more complex task than GLUE

as your base model becomes bigger, it become much more competitive

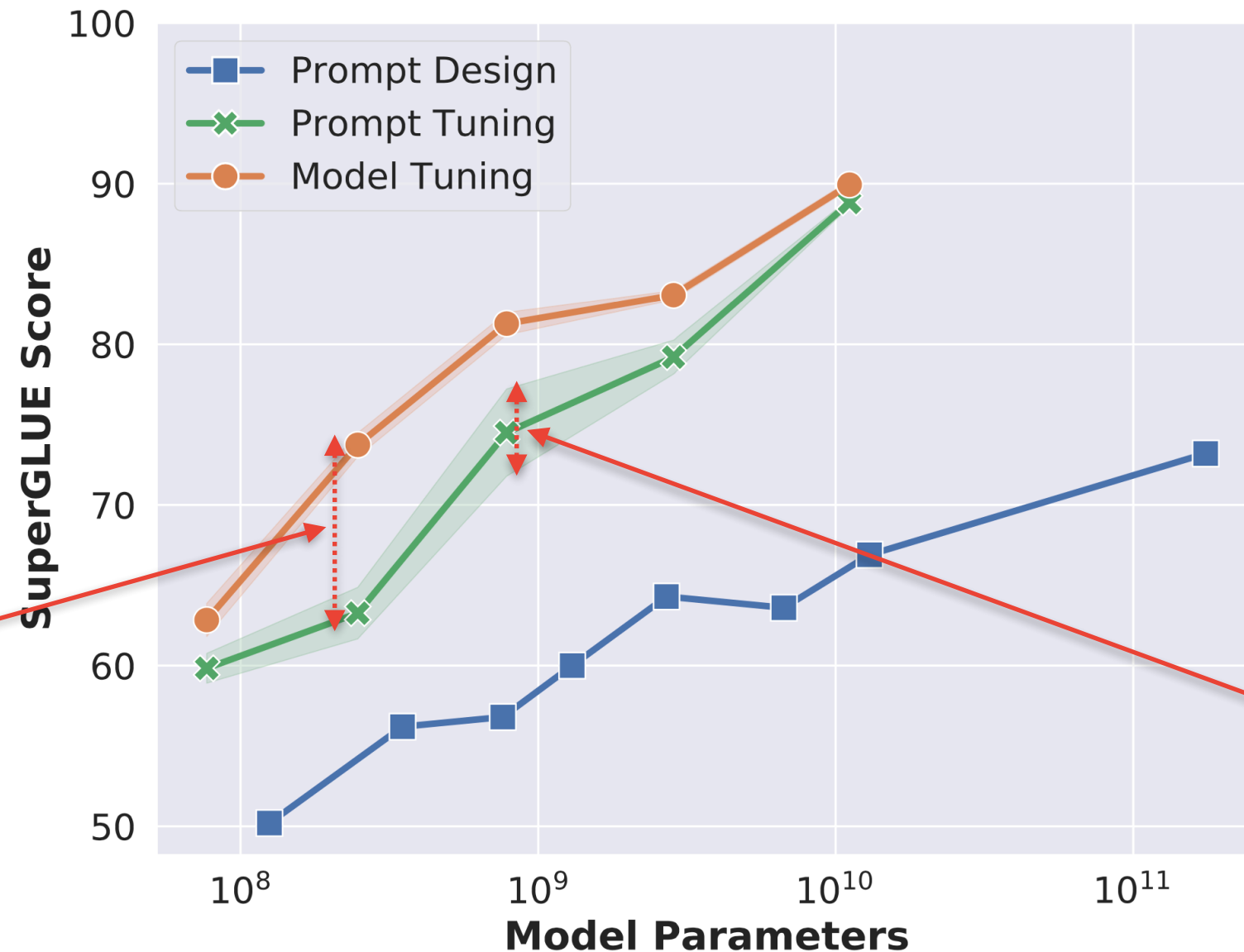
able to memorize more, and generalize more

manual prompt design

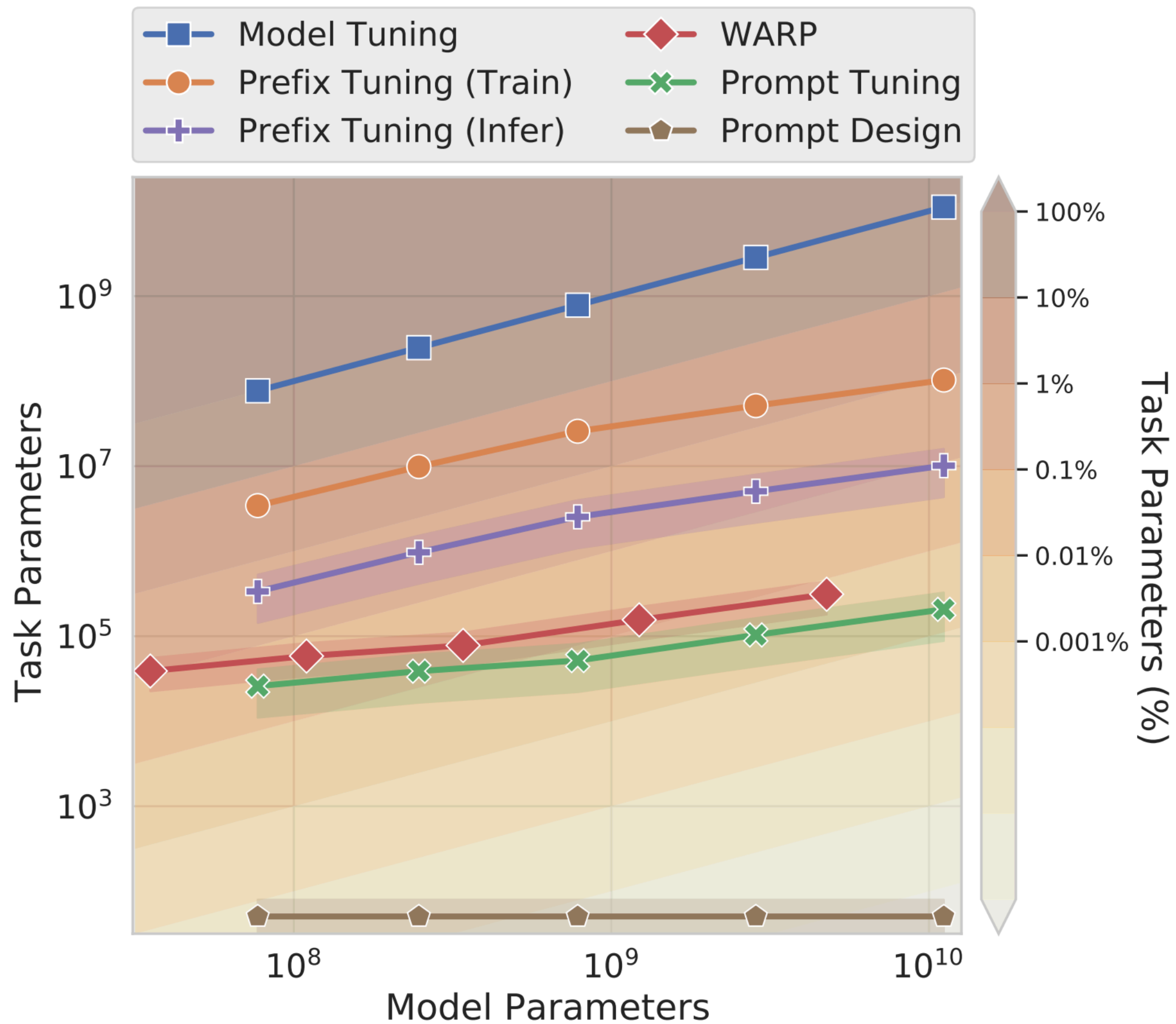
Room for improving Prompt Tuning

[Lester et al., 2021](#)

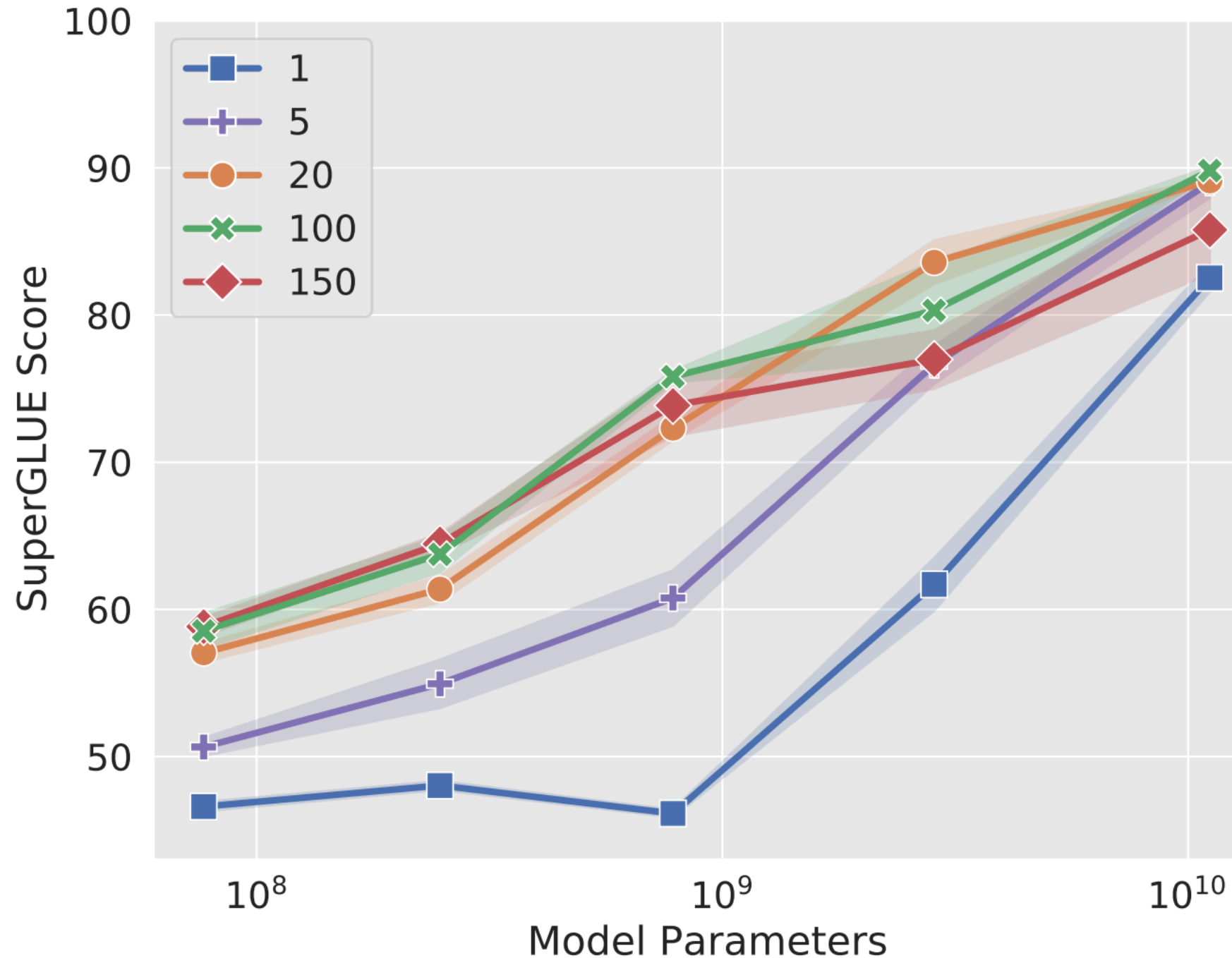
performance



stability

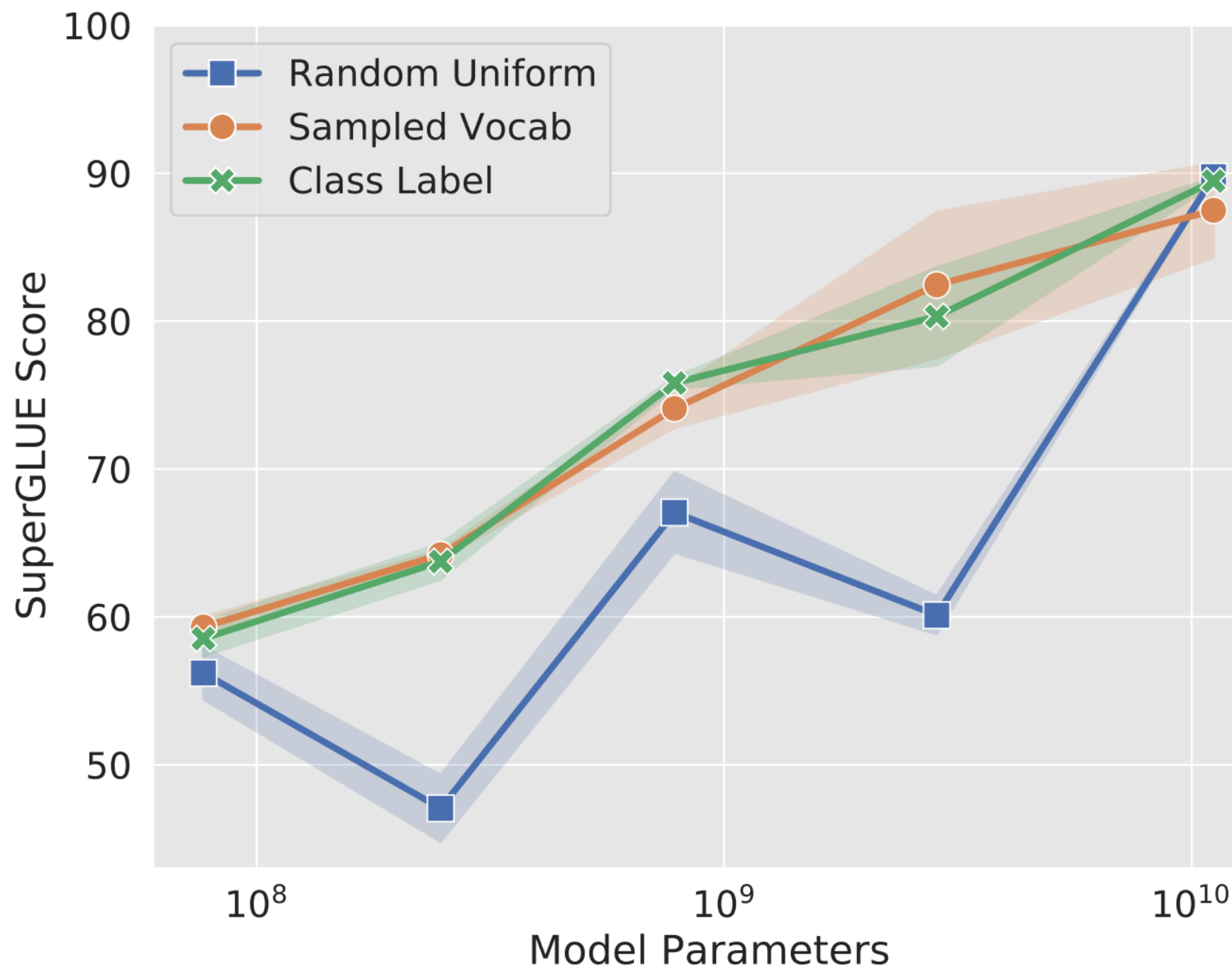


Prompt length matters less with larger pretrained LMs



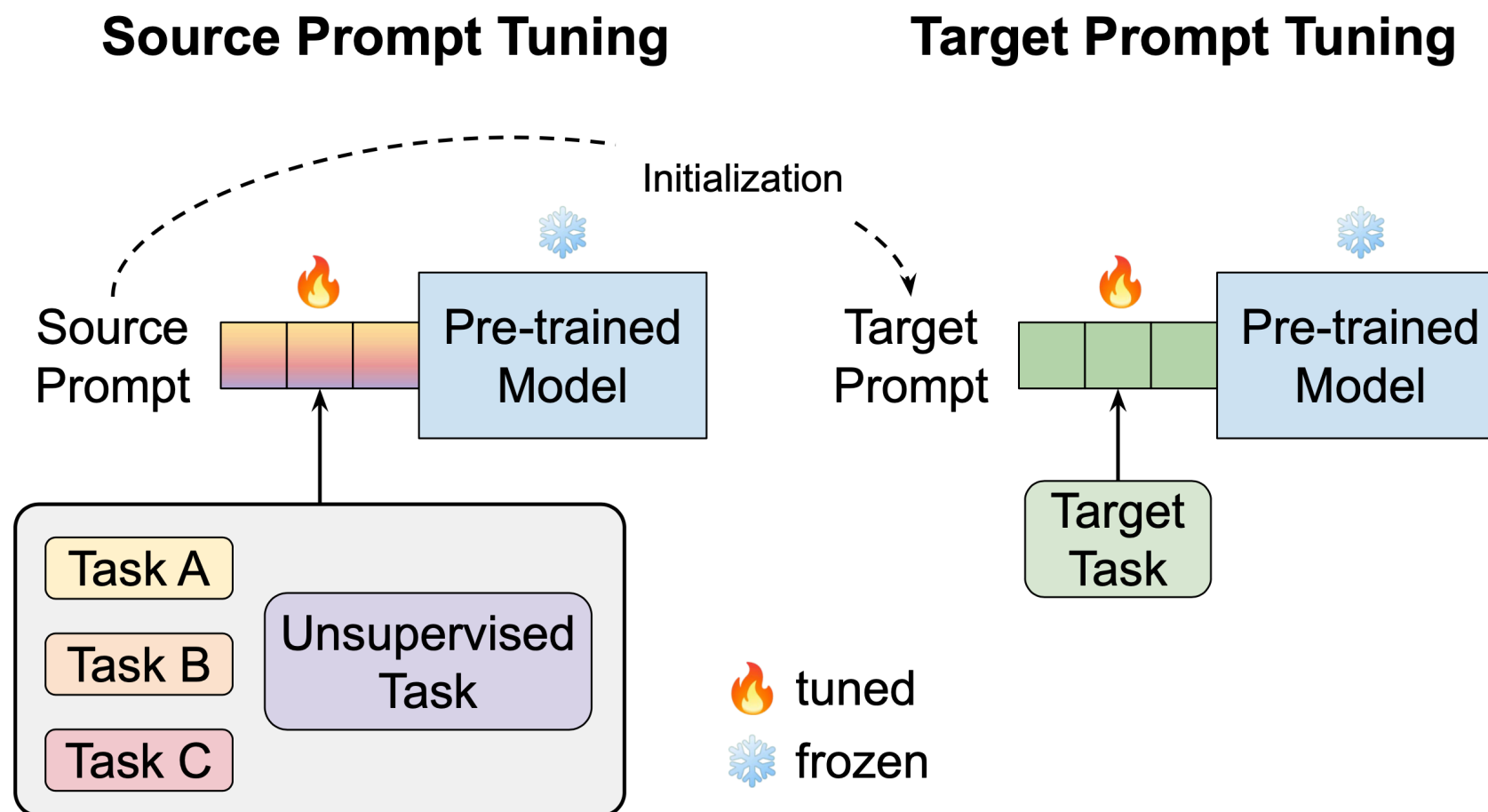
can do better by cleverly initialize rather than randomly initialize

Prompt initialization matters less with larger pretrained LMs



pre-train those embedding on a different task, and then you fine-tune that on a sentiment analysis.
can close the gap of model tuning and prompt tuning.

Prompt *pretraining*: the SPoT approach



We learn a single generic source prompt on one or more source tasks, which is then used to initialize the prompt for each target task.

Google