Prompt-based learning

Many slides from Tu Vu, M Iyyer, G. Neubig

Recommended Reading

Pre-train, Prompt, and Predict: A Systematic Survey of Prompting Methods in Natural Language Processing

Pengfei Liu
Carnegie Mellon University
pliu3@cs.cmu.edu

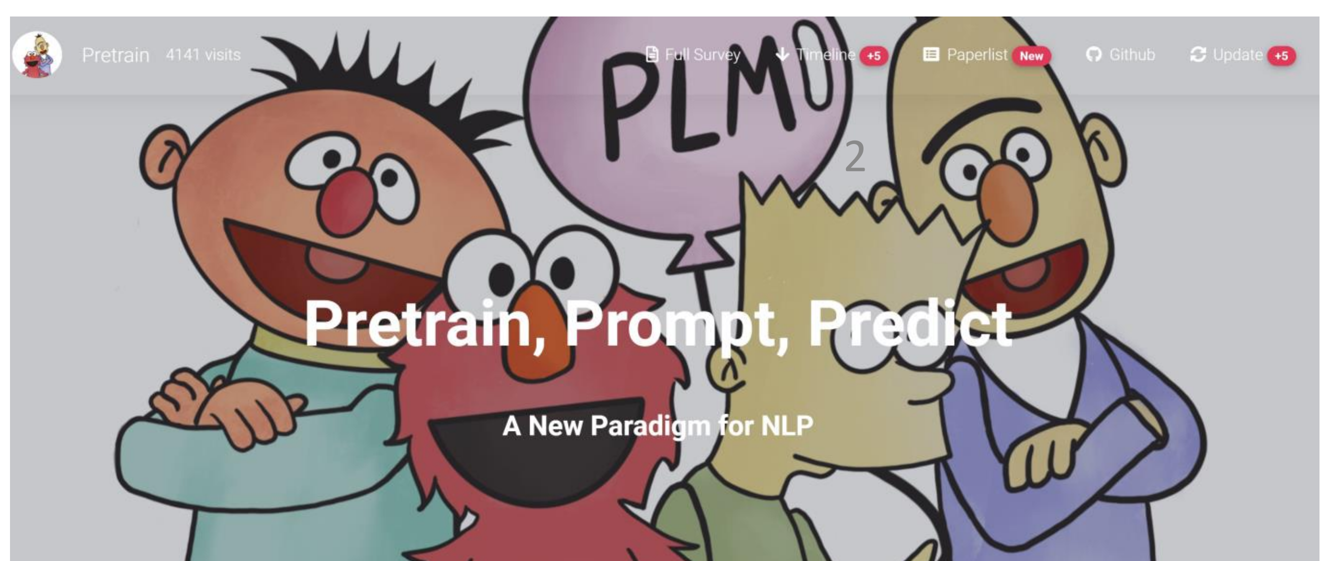
Weizhe Yuan
Carnegie Mellon University
weizhey@cs.cmu.edu

Jinlan Fu
National University of Singapore
jinlanjonna@gmail.com

Zhengbao Jiang
Carnegie Mellon University
zhengbaj@cs.cmu.edu

Hiroaki Hayashi
Carnegie Mellon University
hiroakih@cs.cmu.edu

Graham Neubig
Carnegie Mellon University
gneubig@cs.cmu.edu



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	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”!

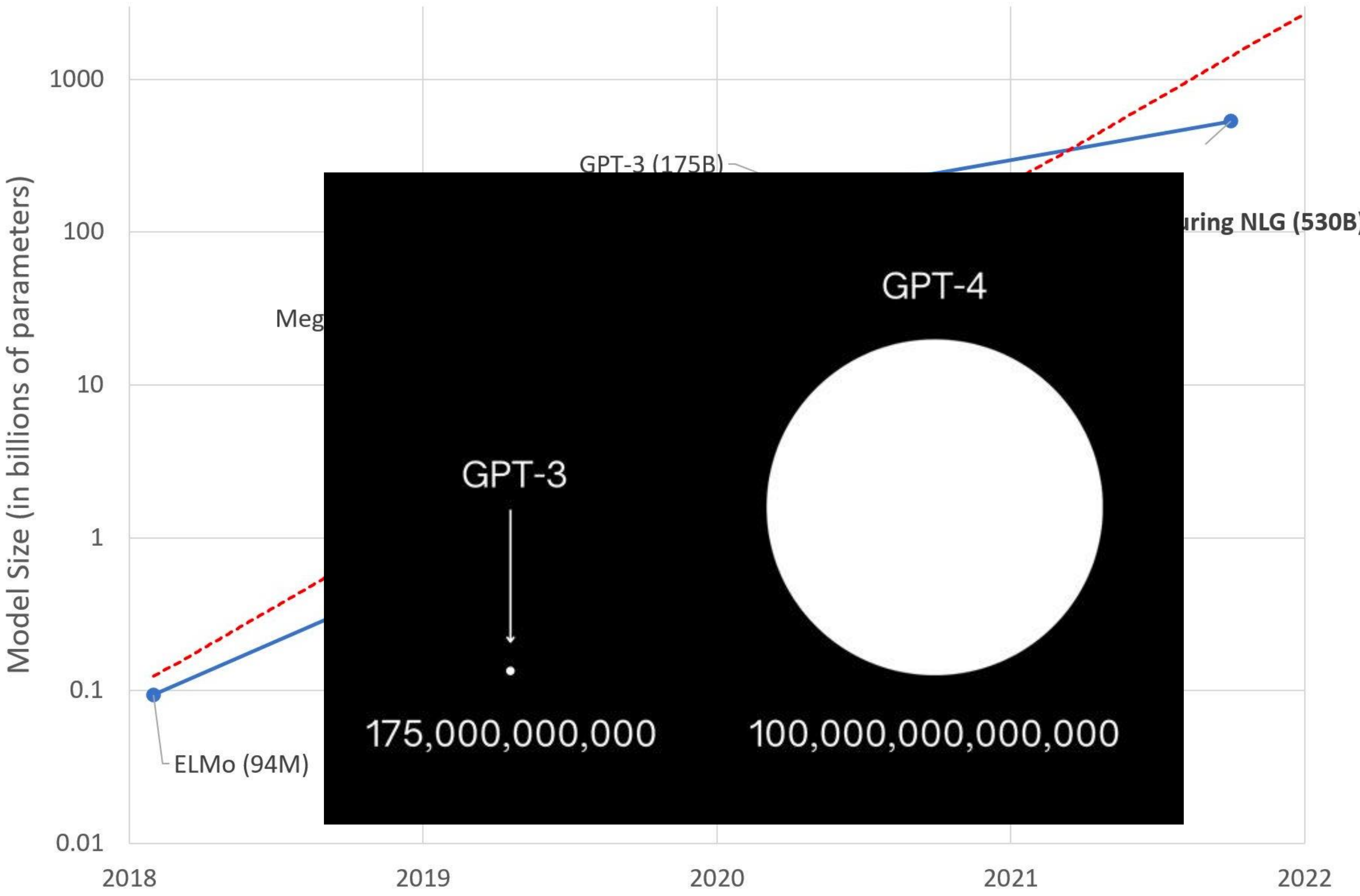
ELMo: 1B training tokens

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

A new 530B param model was released late last year



So… what does all of this scaling buy us?

GPT-3

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

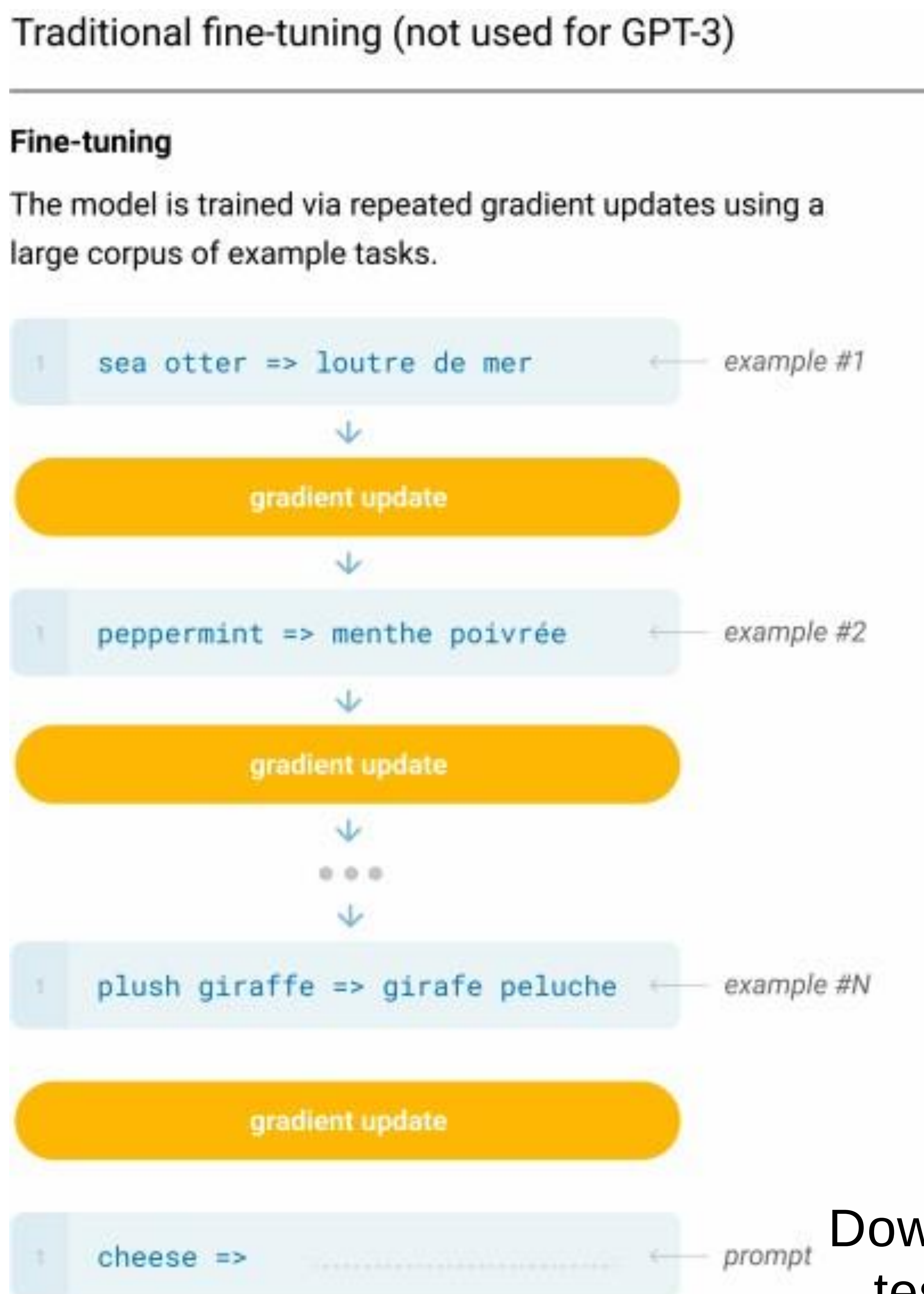
Amanda Askell Sandhini Agarwal Ariel Herbert-Voss Gretchen Krueger Tom Henighan

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



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.

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

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:        ← 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 =>"

Why "=>" ? What is the optimal prompt?

One-shot

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

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 =>”

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

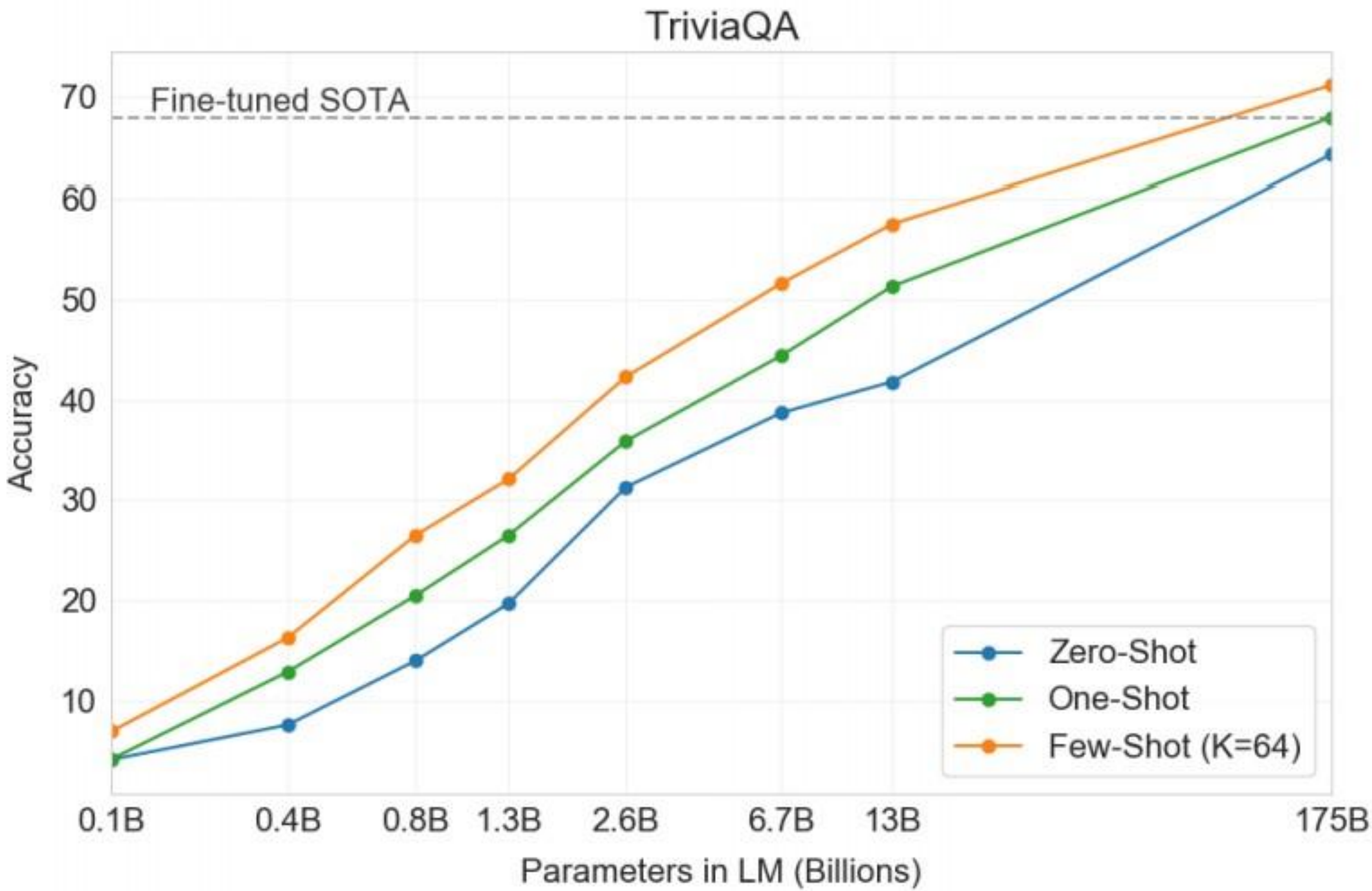
Example

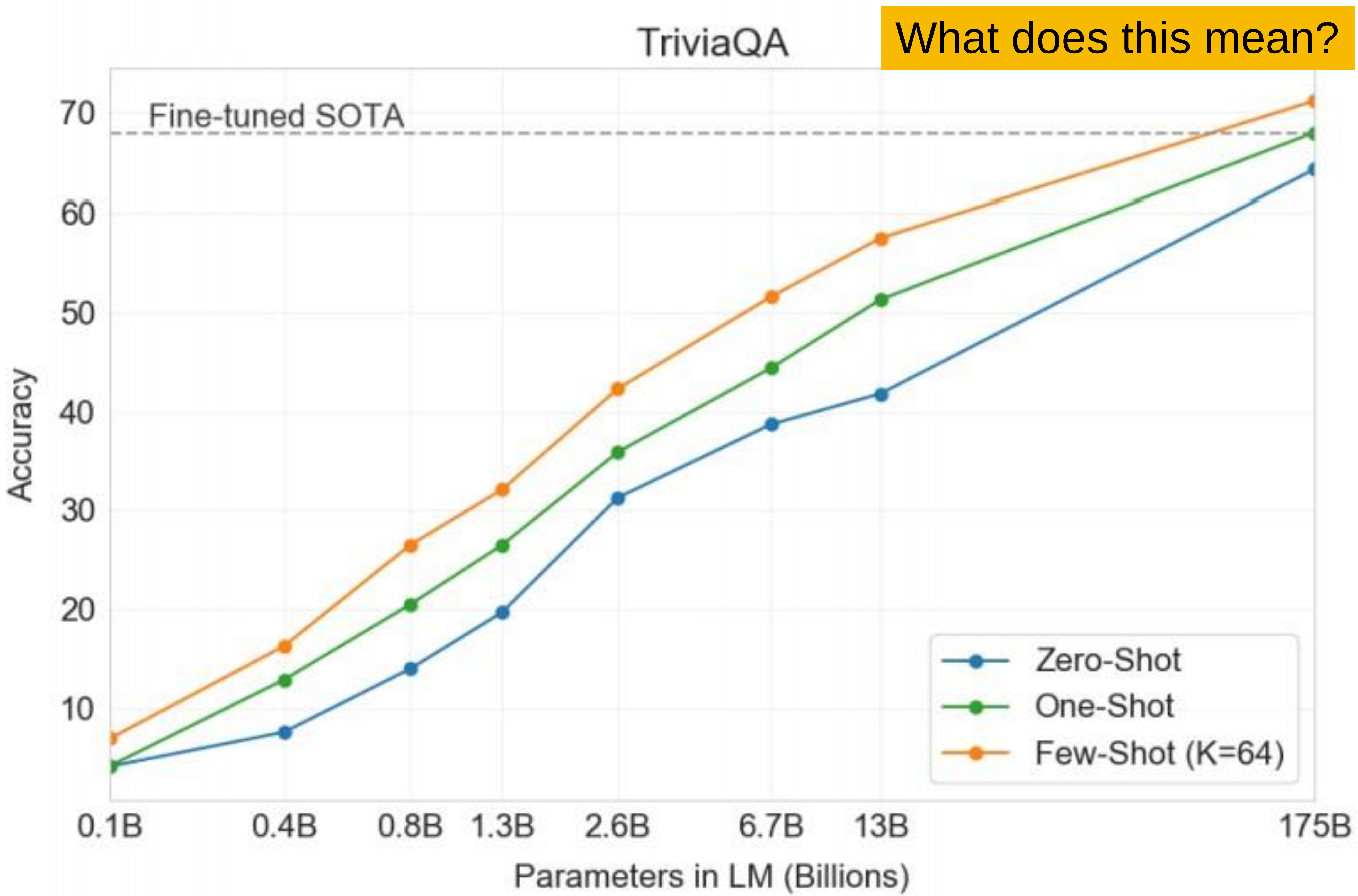
https://platform.openai.com/playground

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?


TriviaQA
Fine-tuned SOTA
Accuracy
70
60
50
40
30
20
10
0.1B
0.4B
0.8B
1.3B
2.6B
6.7B
13B
175B
Parameters in LM (Billions)
Zero-Shot
One-Shot
Few-Shot (K=64)

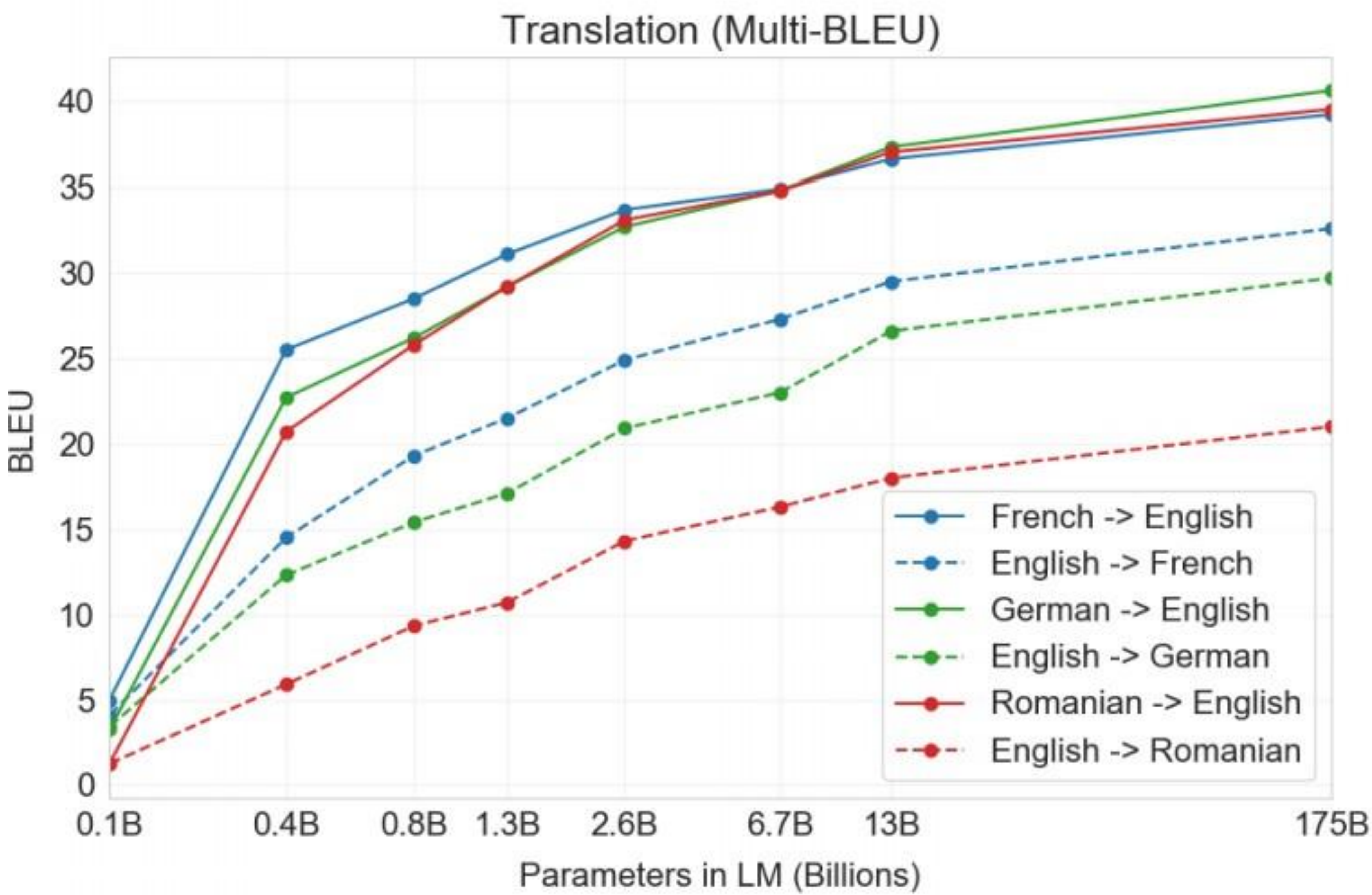


TriviaQA
What does this mean?
Fine-tuned SOTA
Accuracy
Parameters in LM (Billions)
0.1B
0.4B
0.8B
1.3B
2.6B
6.7B
13B
175B
70
60
50
40
30
20
10
Zero-Shot
One-Shot
Few-Shot (K=64)

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

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	35.0	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!

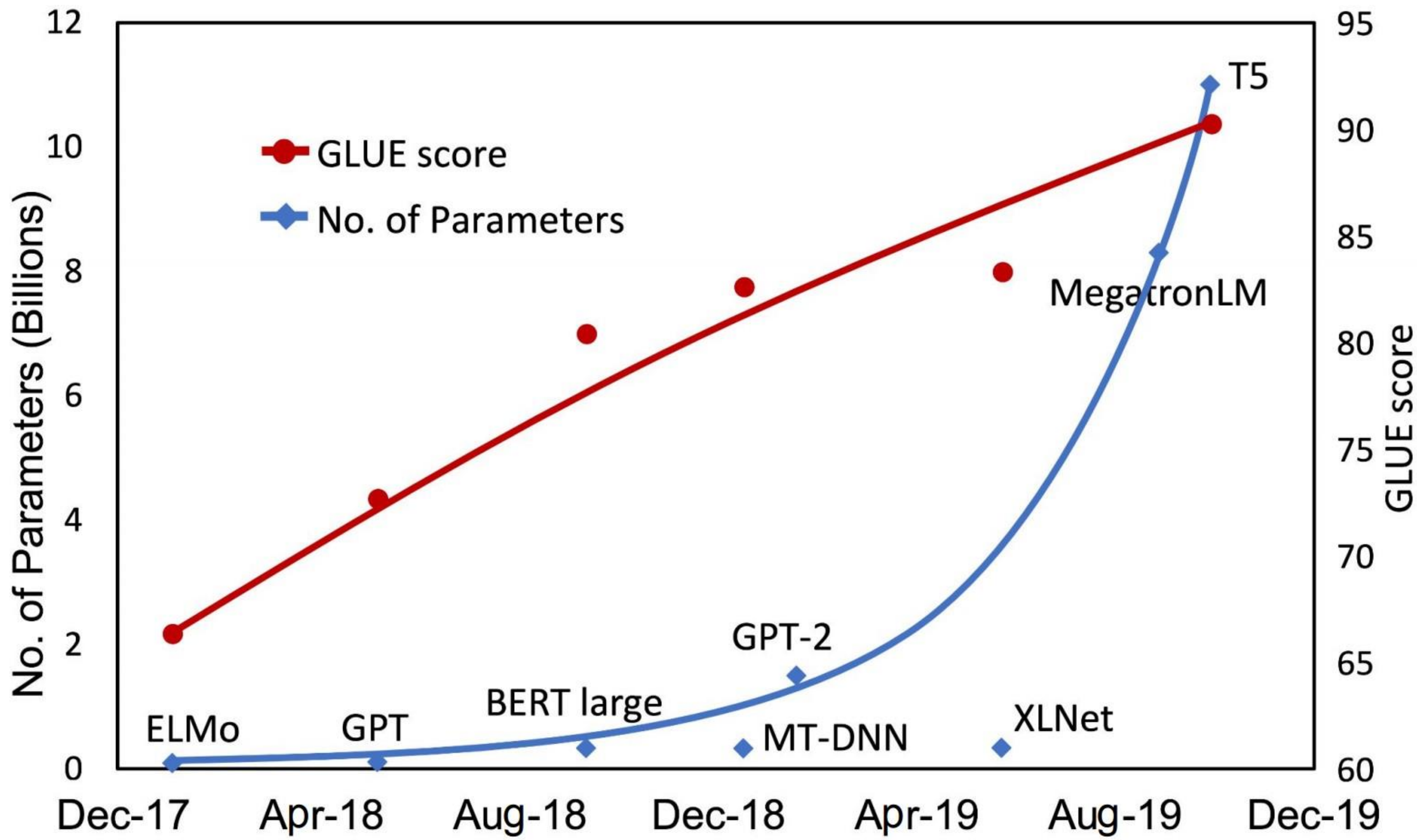


What about reading comprehension QA?

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

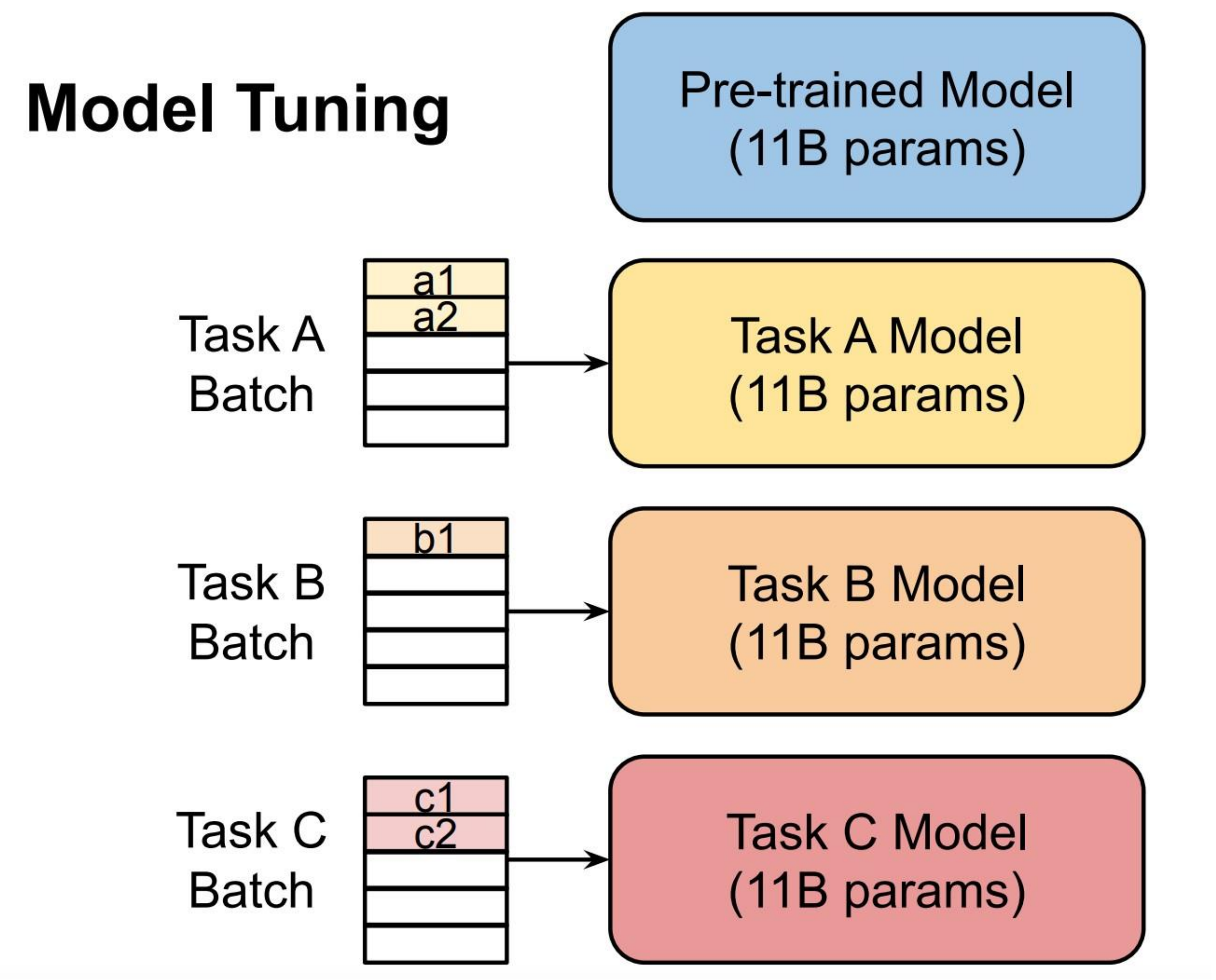
Struggles on “harder” datasets

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



[Ahmet and Abdullah., 2021]

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



[Lester et al., 2021]

Language model prompting to the rescue!

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

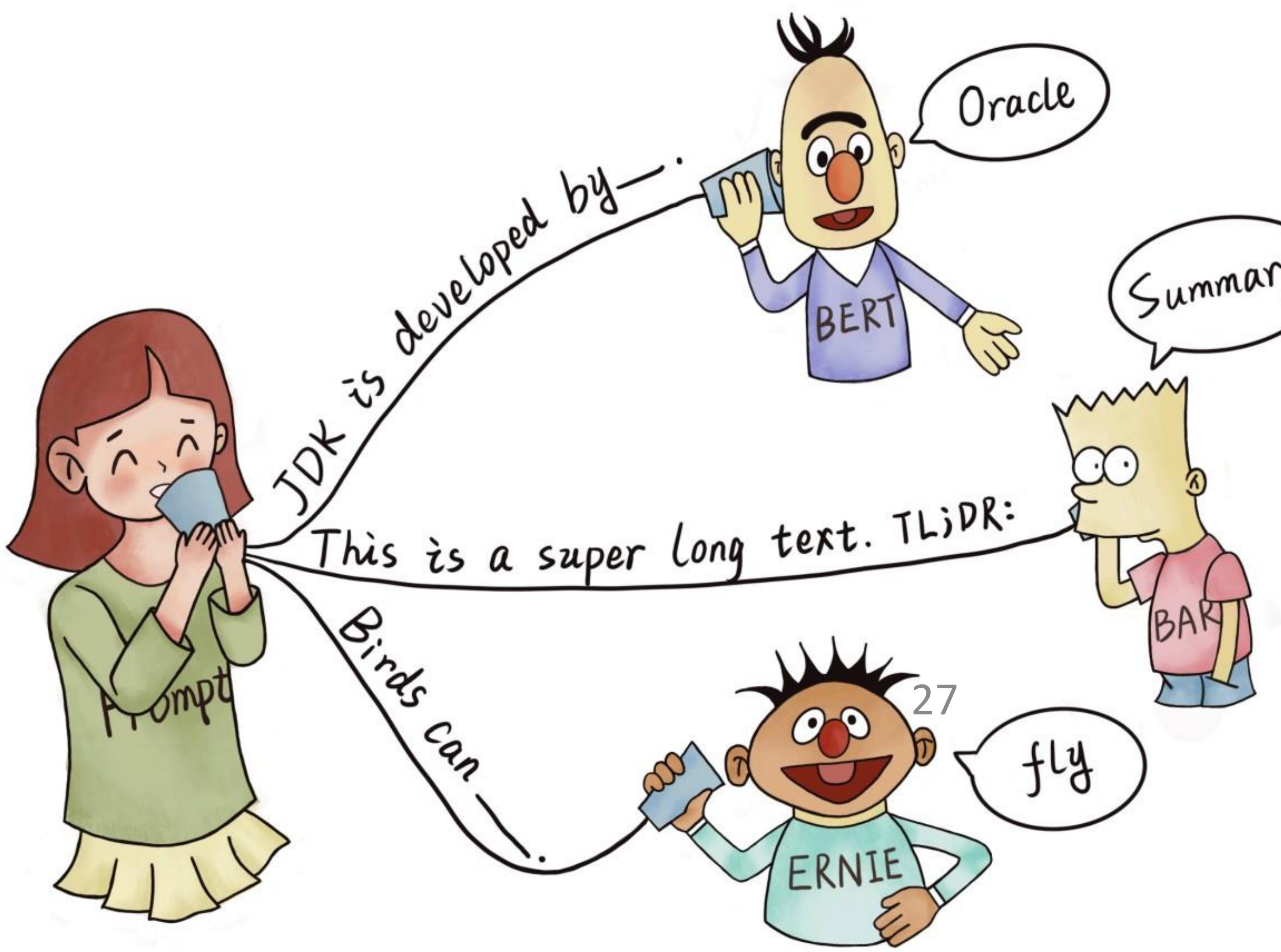
- **natural language instruction** and/or **a few task demonstrations** ⟶ **output**

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

- *no* gradient updates or fine-tuning

What is Prompting？

Encouraging a pre-trained model to make particular predictions by providing a "prompt" specifying the task to be done.



Terminologies and Notations

Name	Notation	Example	Description
Input	$\boldsymbol{x}$	I love this movie.	One or multiple texts
Output	$\boldsymbol{y}$	++ (very positive)	Output label or text
Prompting Function	$f_{\text{prompt}}(\boldsymbol{x})$	`[X]` Overall, it was a `[Z]` movie.	A function that converts the input into a specific form by inserting the input $\boldsymbol{x}$ and adding a slot `[Z]` where answer $\boldsymbol{z}$ may be filled later.
Prompt	$\boldsymbol{x}'$	I love this movie. Overall, it was a `[Z]` movie.	A text where `[X]` is instantiated by input $\boldsymbol{x}$ but answer slot `[Z]` is not.
Filled Prompt	$f_{\text{fill}}(\boldsymbol{x}', \boldsymbol{z})$	I love this movie. Overall, it was a bad movie.	A prompt where slot `[Z]` is filled with any answer.
Answered Prompt	$f_{\text{fill}}(\boldsymbol{x}', \boldsymbol{z}^*)$	I love this movie. Overall, it was a good movie.	A prompt where slot `[Z]` is filled with a true answer.
Answer	$\boldsymbol{z}$	"good", "fantastic", "boring"	A token, phrase, or sentence that fills `[Z]`

Terminology and notation of prompting methods. $\boldsymbol{z}^*$ represents answers that correspond to true output $\boldsymbol{y}^*$.

What is the general workflow of Prompting?

- Prompt Addition
- Answer Prediction
- Answer-Label Mapping

Prompt Addition

- **Prompt Addition**: Given input x, we transform it into prompt x' through two steps:
  - Define a template with two slots, one for input [x], and one for the answer [z]
  - Fill in the input slot [x]

Example: Sentiment Classification

Input: x = “I love this movie”

Template: [x] Overall, it was a [z] movie

Prompting: x’ = “I love this movie. Overall it was a [z] movie.”

Answer Prediction

- Answer Prediction: Given a prompt, predict the answer [z]
  - Fill in [z]

Example

Input: x = "I love this movie"

Template: [x] Overall, it was a [z] movie

Prompting: x' = "I love this movie. Overall it was a [z] movie."

Predicting: x' = "I love this movie. Overall it was a fantastic movie."

Mapping

- Mapping: Given an answer, map it into a class label

Example

Input: x = “I love this movie”

Template: [x] Overall, it was a [z] movie

Prompting: x’ = “I love this movie. Overall it was a [z] movie.”

Predicting: x’ = “I love this movie. Overall it was a fantastic movie.”

Mapping: fantastic => Positive

Types of Prompts

- Prompt: **I love this movie. Overall it was a [z] movie**

- Filled Prompt: **I love this movie. Overall it was a boring movie**
- Answered Prompt: **I love this movie. Overall it was a fantastic movie**

- Prefix Prompt: **I love this movie. Overall this movie is [z]**
- Cloze Prompt: **I love this movie. Overall it was a [z] movie**

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]. *(original)*	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)

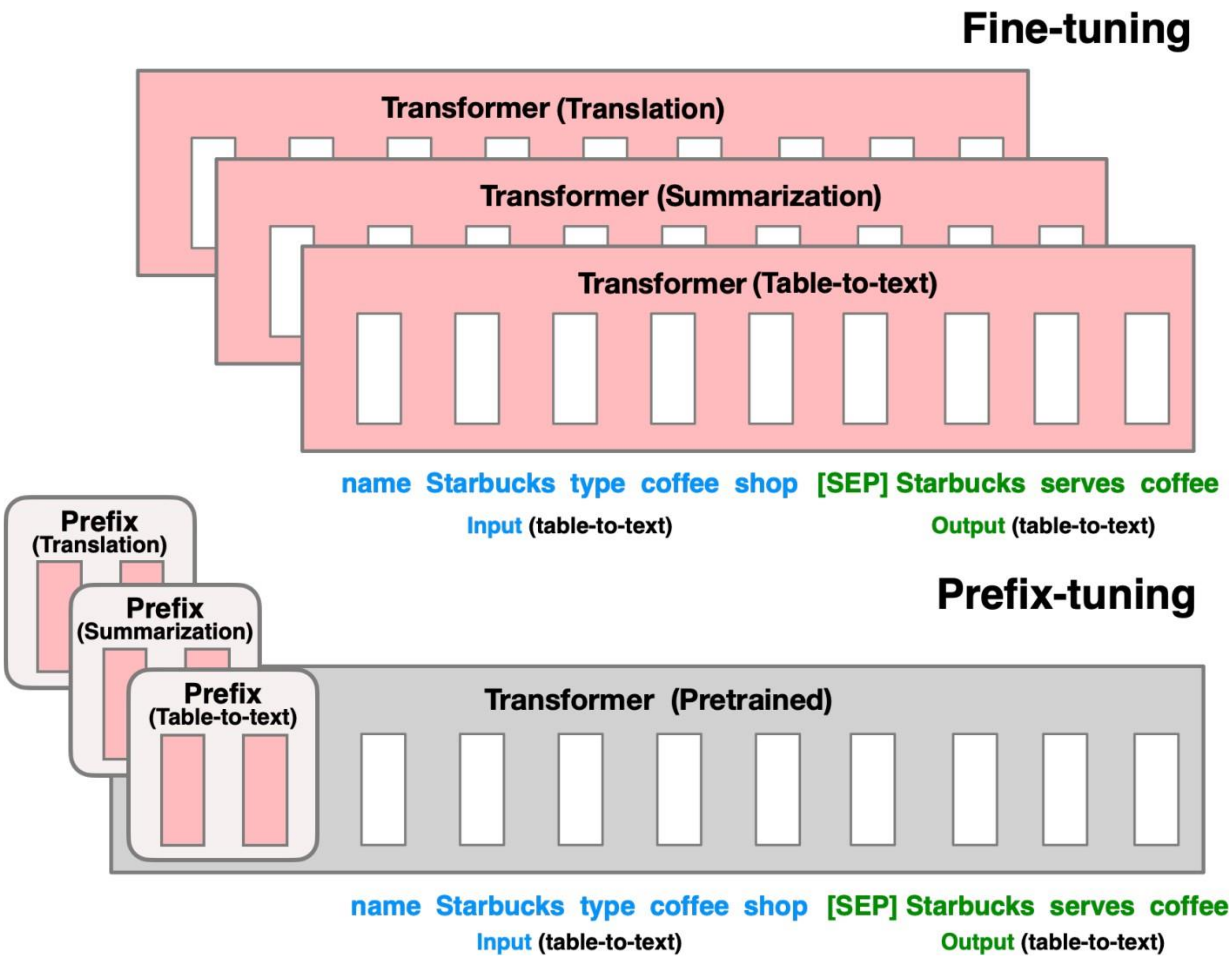
Continuous/soft prompts

- 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

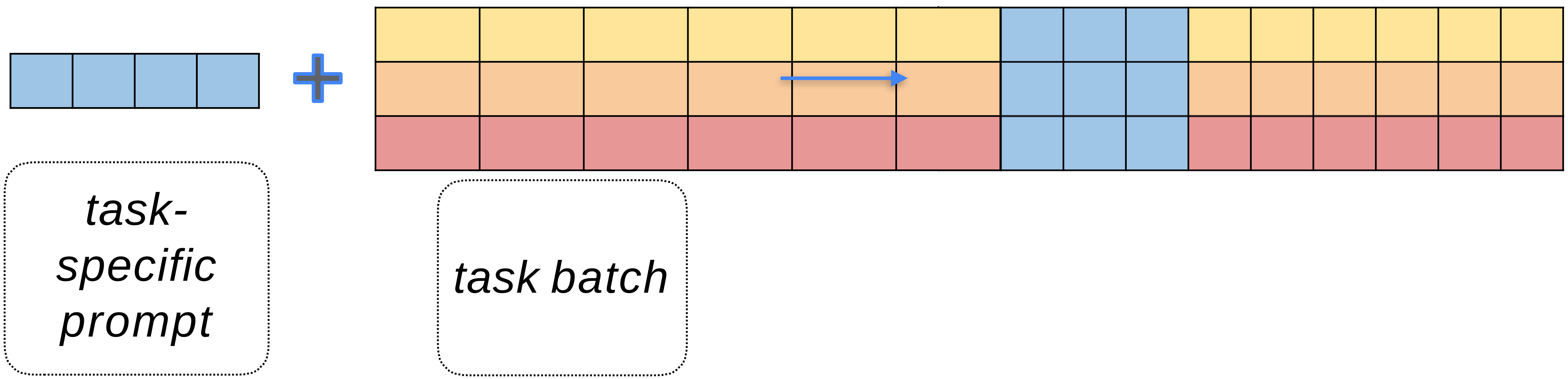
Prefix tuning (Li & Liang, ACL 2021)



Prompt Tuning idea (Lester et al., 2021)

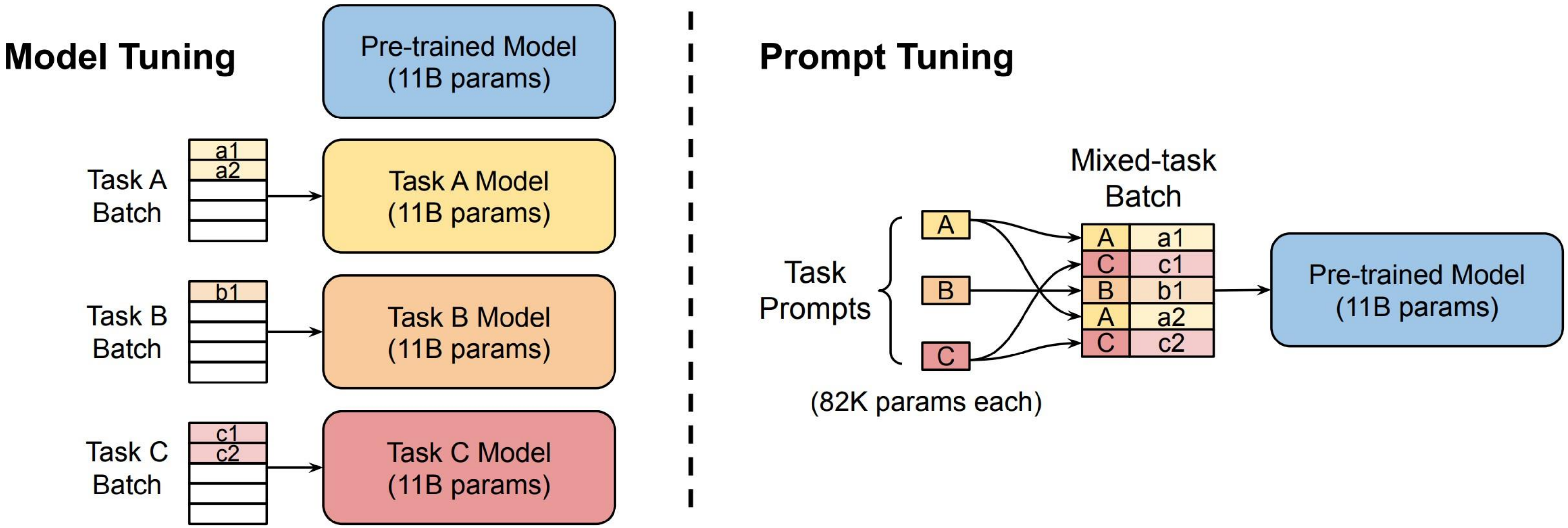
What is a prompt in Prompt Tuning?

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

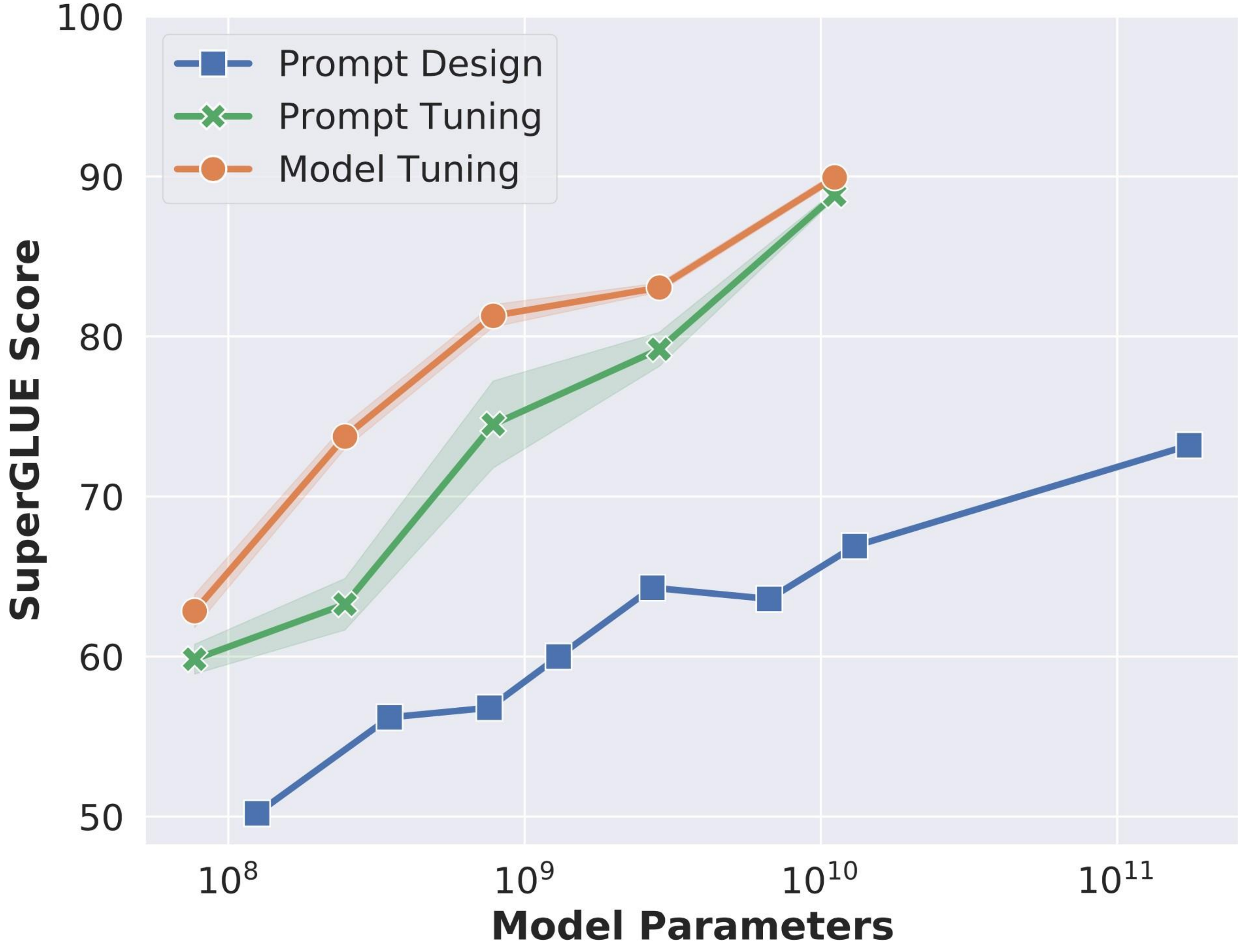


Whiteboard

Parameter-efficient Prompt Tuning

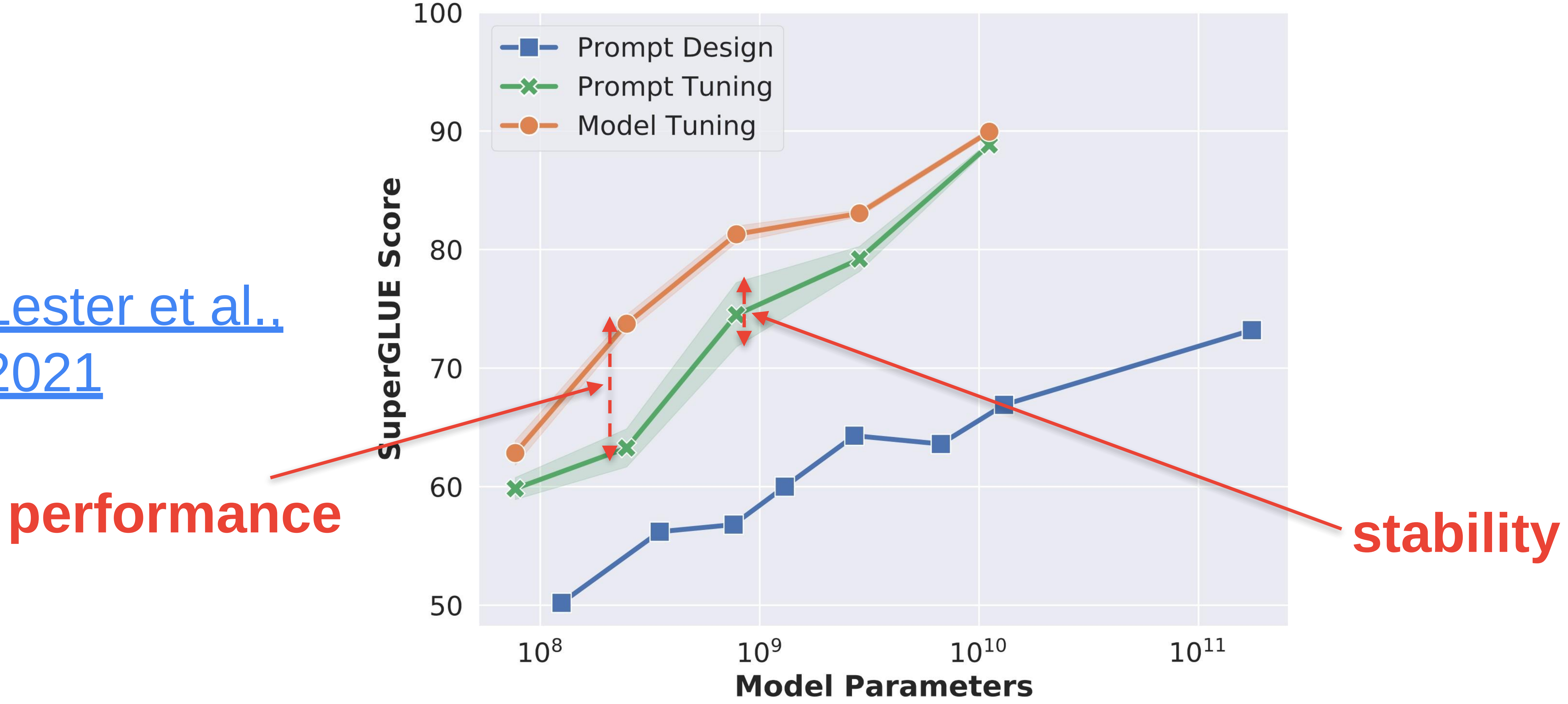


Prompt Tuning becomes more competitive with scale

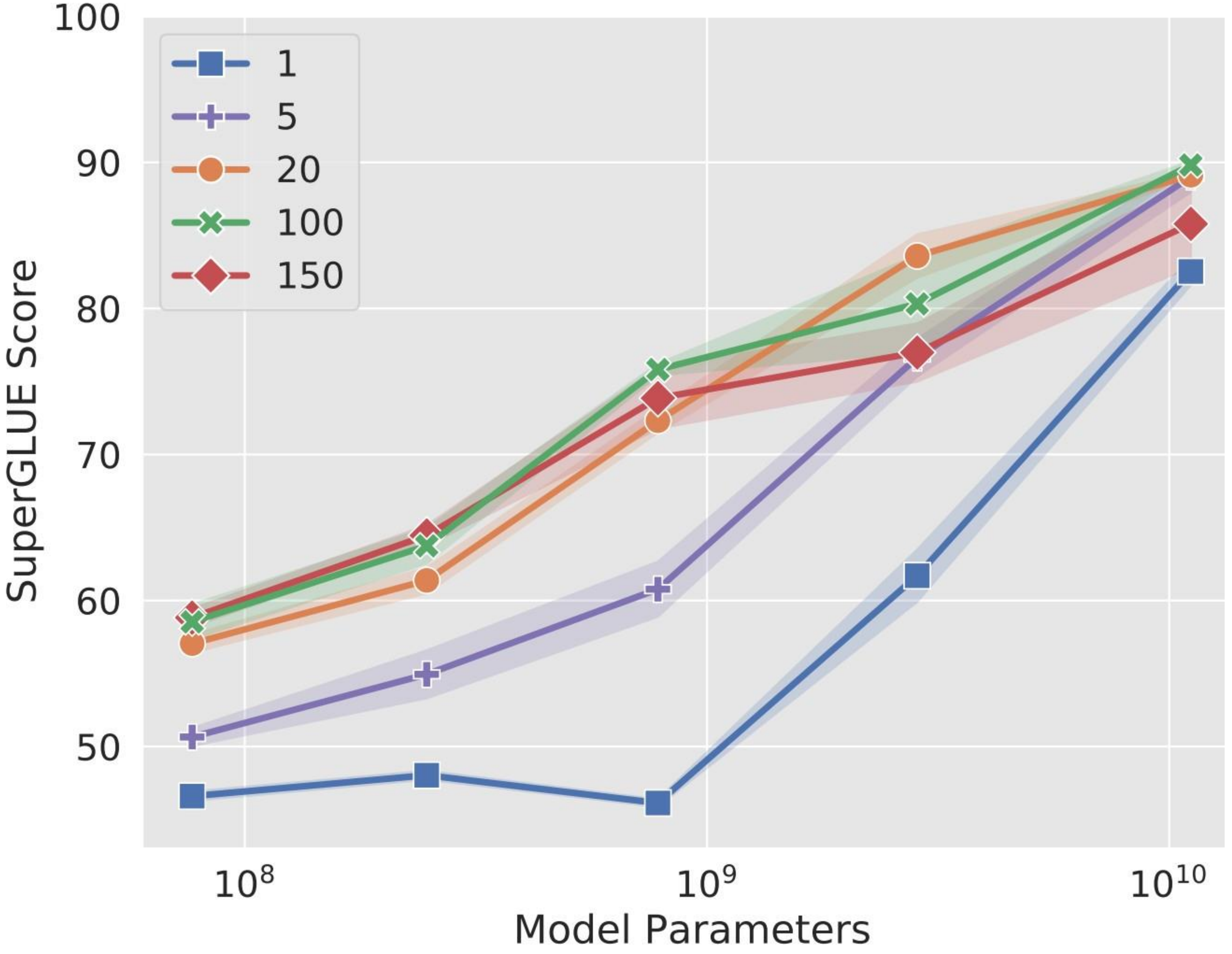


Room for improving Prompt Tuning

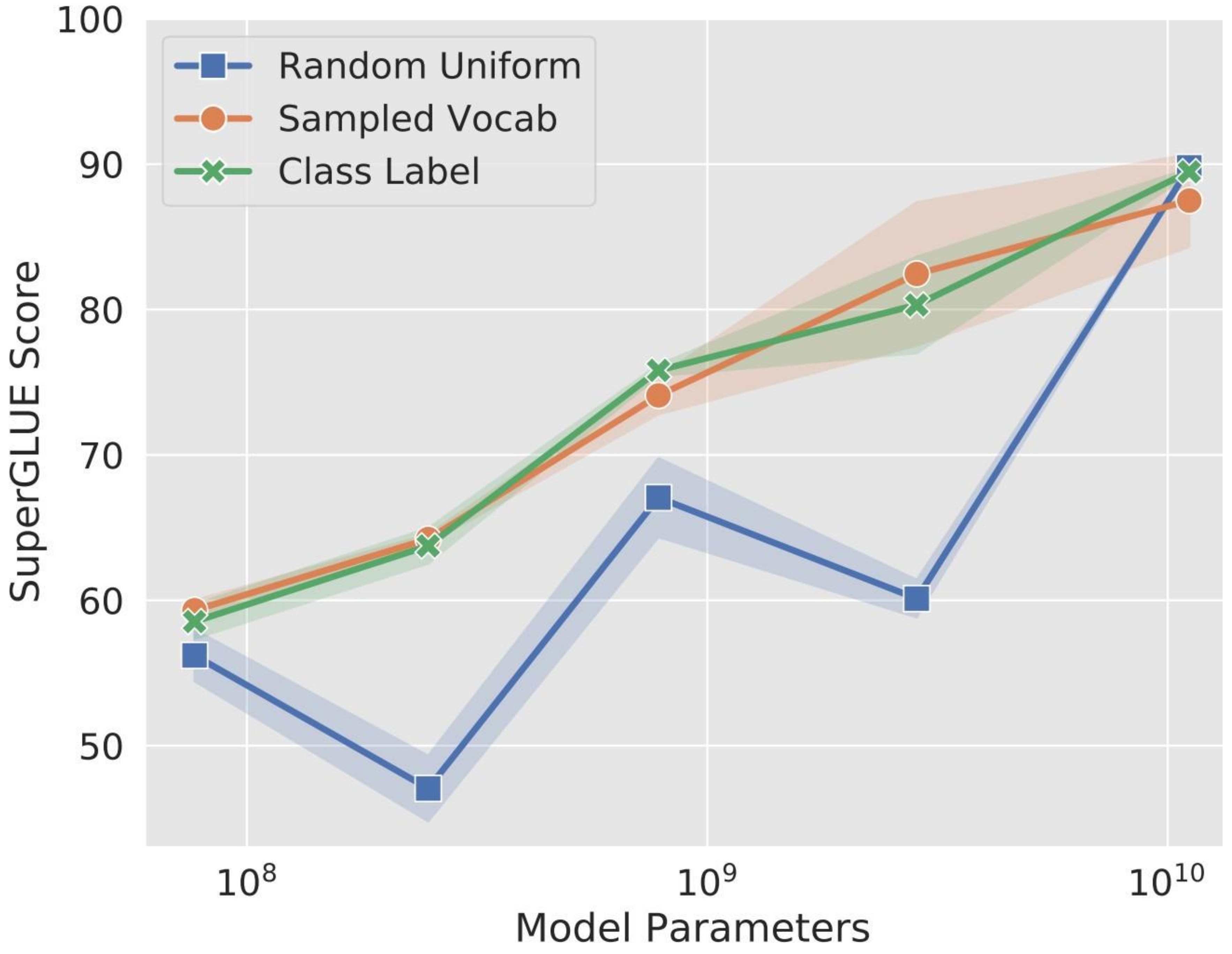
[Lester et al., 2021]



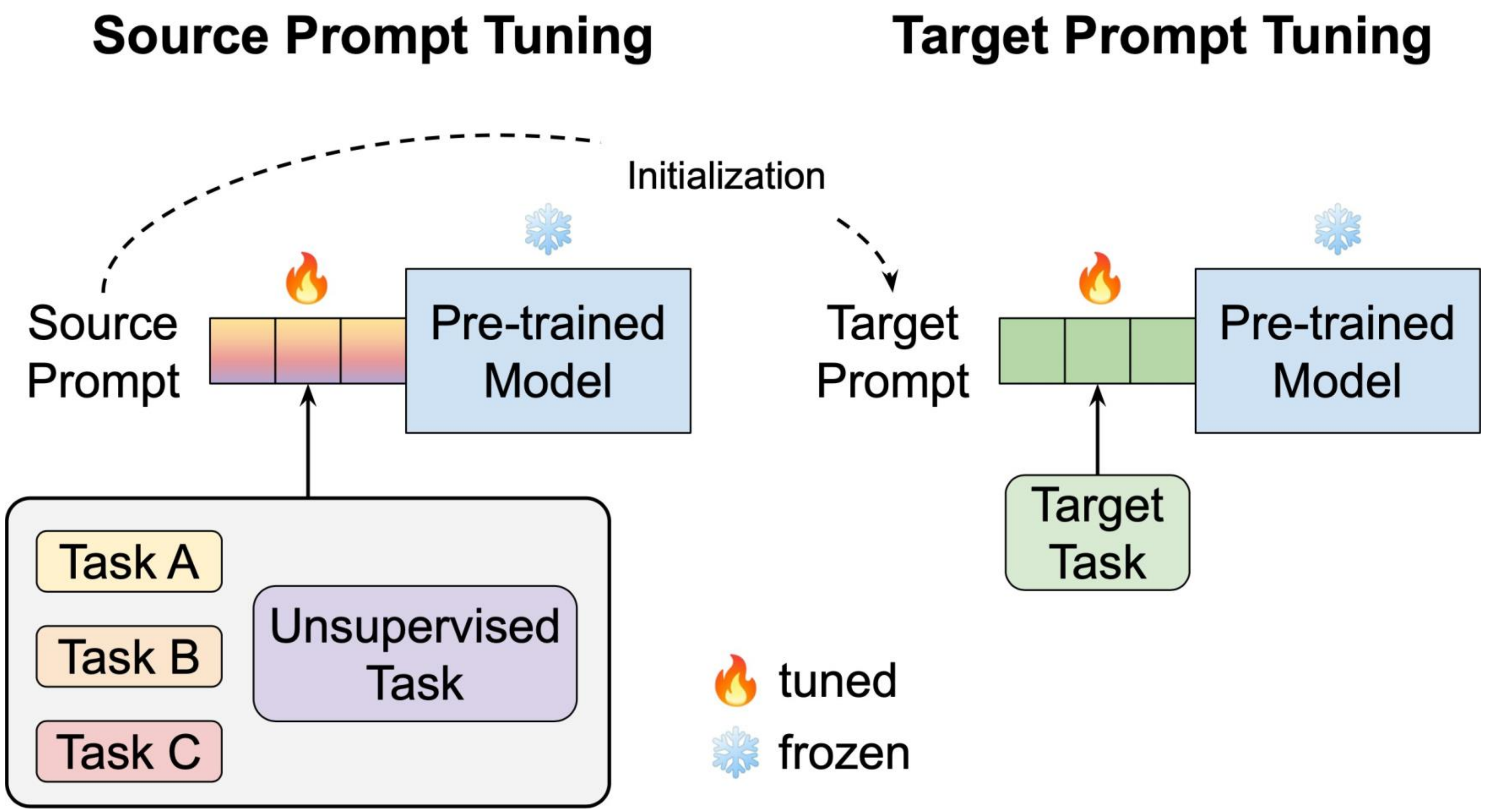
Prompt length matters less with larger pretrained LMs



Prompt initialization matters less with larger pretrained LMs



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

SPoT: Better Frozen Model Adaptation through Soft Prompt Transfer. Vu et al., ACL 2022