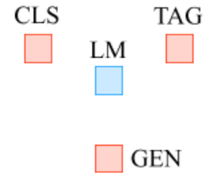
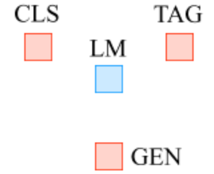
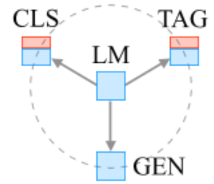
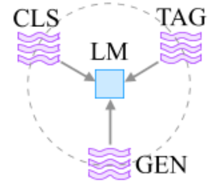
Continuous Prompting Methods

汪杰

Four paradigms in NLP

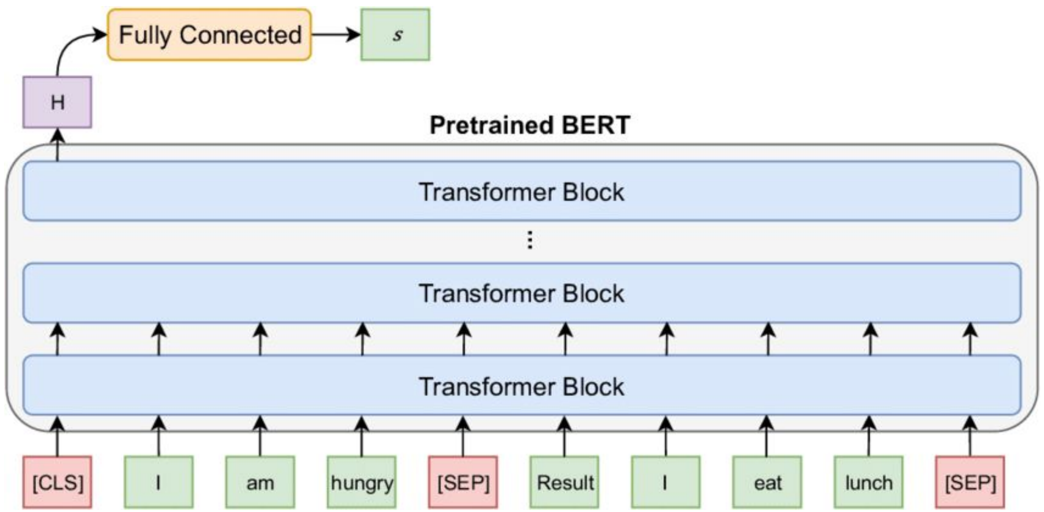
- 特征工程
- 架构工程
- 目标工程
- 提示工程

Paradigm	Engineering	Task Relation
a. Fully Supervised Learning (Non-Neural Network)	Features (e.g. word identity, part-of-speech, sentence length)	
b. Fully Supervised Learning (Neural Network)	Architecture (e.g. convolutional, recurrent, self-attentional)	
c. Pre-train, Fine-tune	Objective (e.g. masked language modeling, next sentence prediction)	
d. Pre-train, Prompt, Predict	Prompt (e.g. cloze, prefix)	

Paper: Pre-train, Prompt, and Predict: A Systematic Survey of Prompting Methods in Natural Language Processing, 2021

Fine Tuning

- Pretrain a language model on task
- Attach a small task specific layer
- Fine-tune the weights of full NN by propagating gradients on a downstream task



Paper: BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding, 2019 (BERT)

Parameter-efficient Fine tuning

- With standard fine-tuning, we need to make a **new copy** of the model for each task.
- In the extreme case of a different model per user, we could never store 1000 different full models.
- If we fine tuned a **subset of the parameters** for each task, we could alleviate storage costs. This is **parameter-efficiency**.

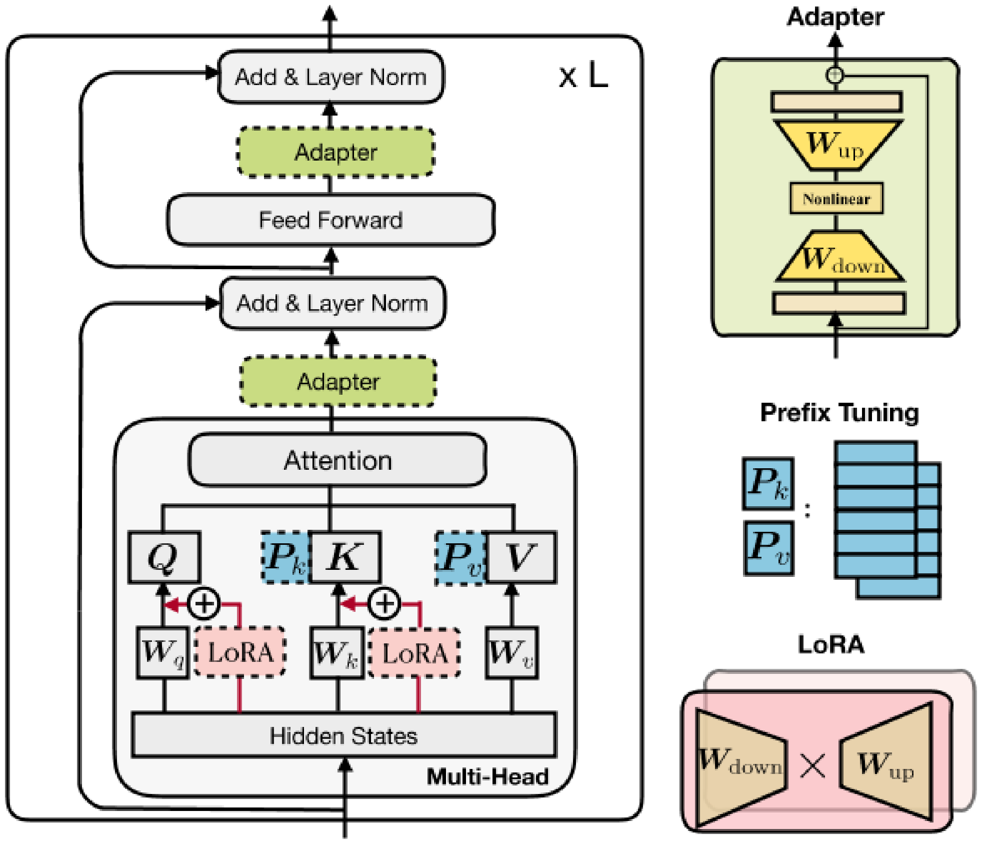
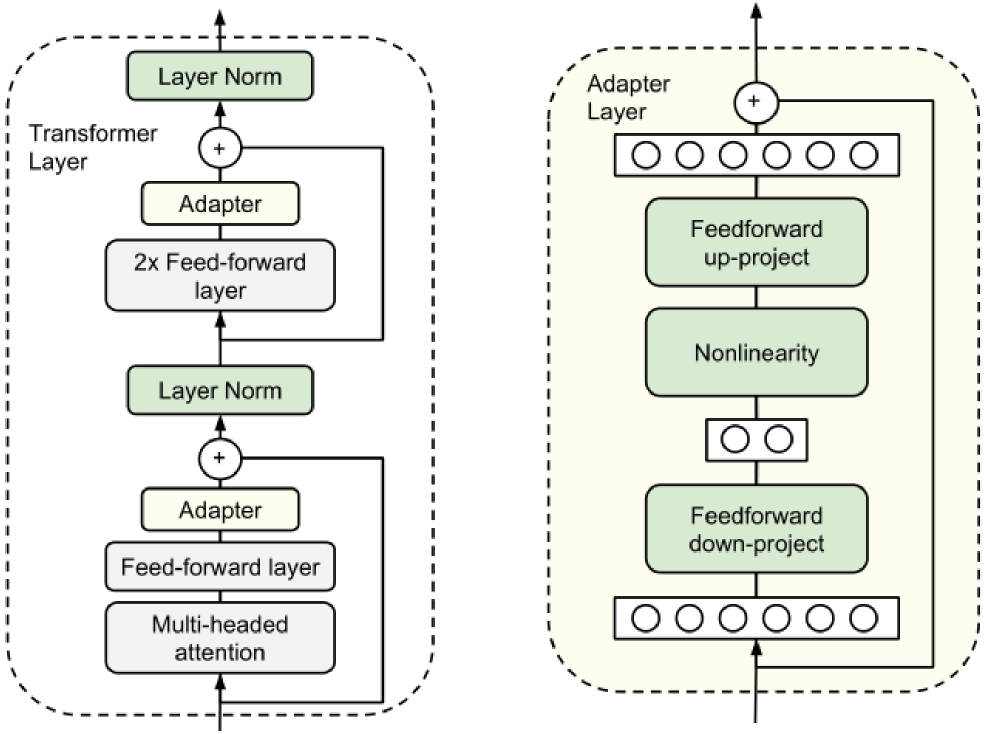


Image Source: Towards a Unified View of Parameter-Efficient Transfer Learning, 2021

Adapter Fine Tuning

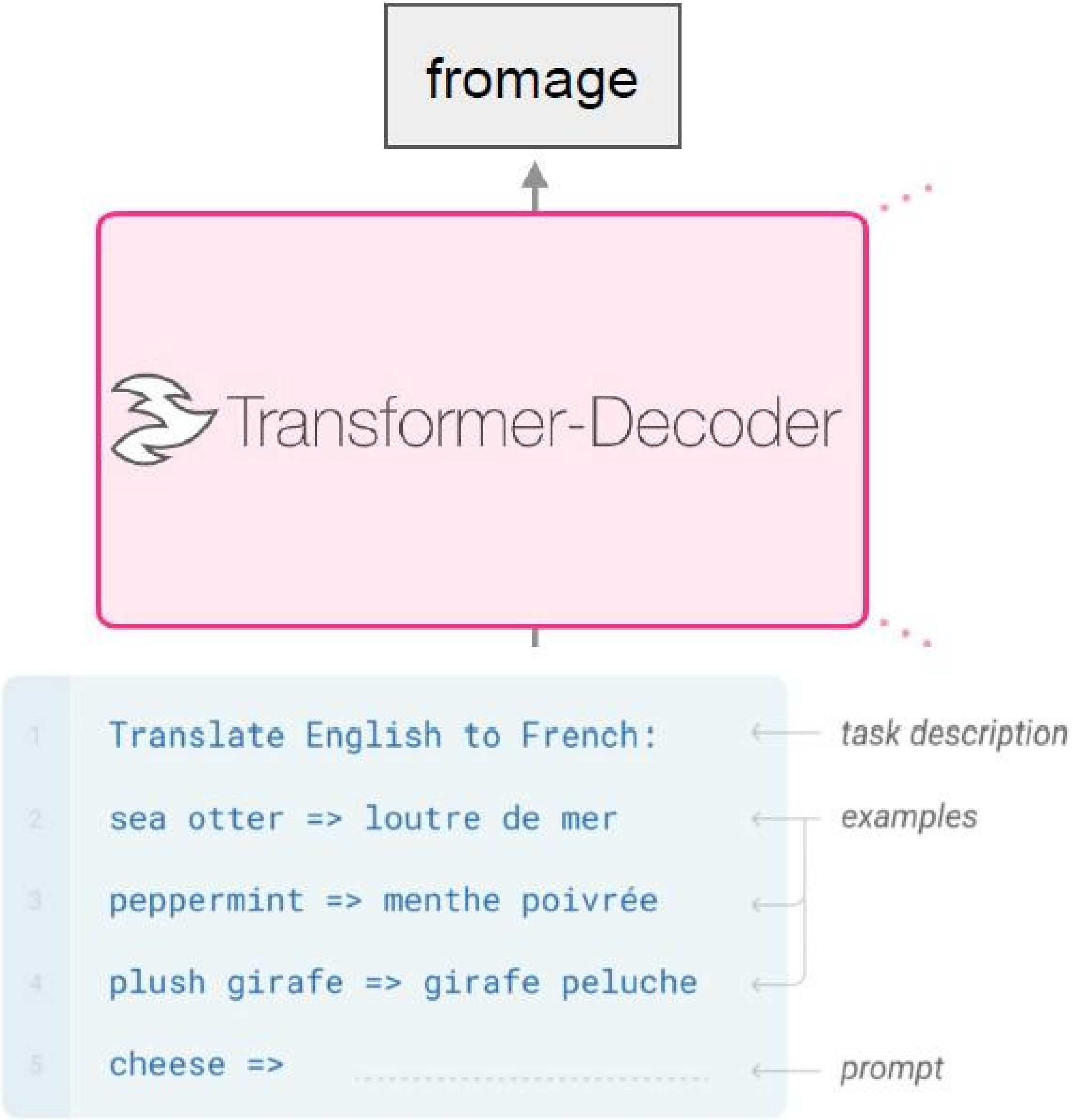
- They add **adapter layers** in between the transformer layers of a large model.
- During fine-tuning, they **fix the original model parameters** and only tune the adapter layers.
- No need to store a full model for each task, only the adapter params.
- **3.6%** of parameters needed!



Paper: Parameter-Efficient Transfer Learning for NLP, 2019

In-context Learning

- Pretrain a language model on task (LM)
- Manually design a "prompt" that demonstrates how to formulate a task as a generation task.
- No need to update the model weights at all!



Paper: Language Models are Few-Shot Learners, 2020 (GPT3)

Chain of Thought ~~(AI 鼓励师)~~

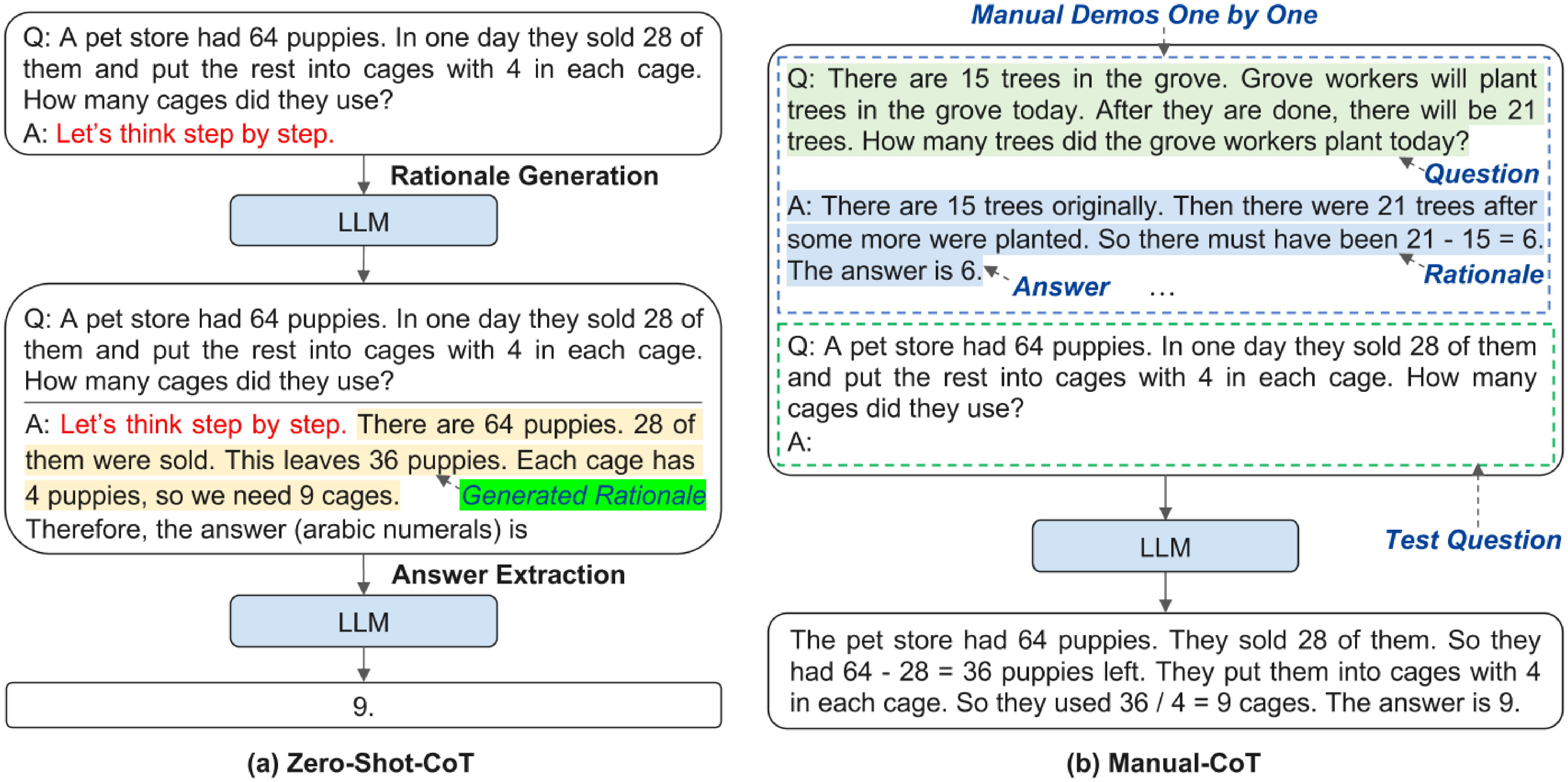
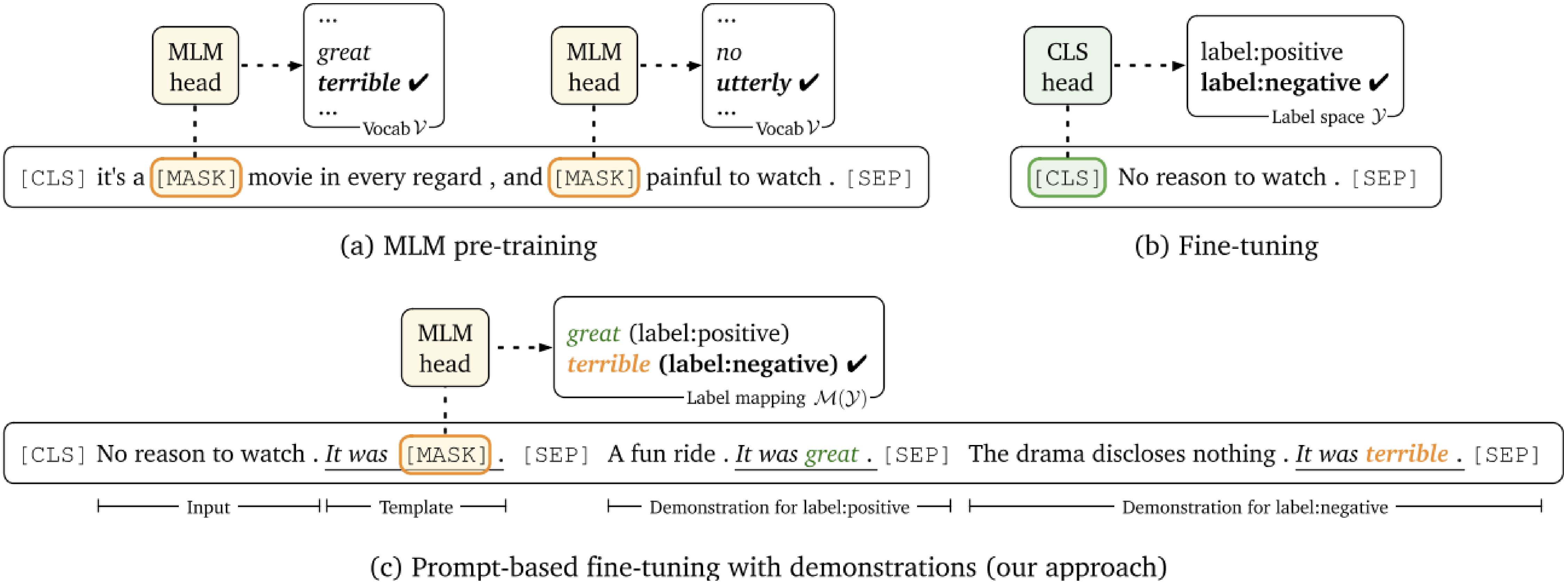


Image Source: Automatic Chain of Thought Prompting in Large Language Models, 2022

Discrete prompting



Paper: Making Pre-trained Language Models Better Few-shot Learners, 2021 (LM-BFF)

Continuous prompting / Soft prompting

Motivation: Do we really need discrete words in the prompts?

- For prompt design (GPT3), the discrete prompt is optimized manually.
    - **Error-prone** and requires engineering.
- Optimization in discrete space is hard! (LM-BFF)
- What if we can optimize the prompt in the continuous embedding space?
- This would sacrifice interpretability but would be **easier to optimize**.
- Much less parameters to tune (parameter-efficient).

Prefix-Tuning: Optimizing Continuous Prompts for Generation

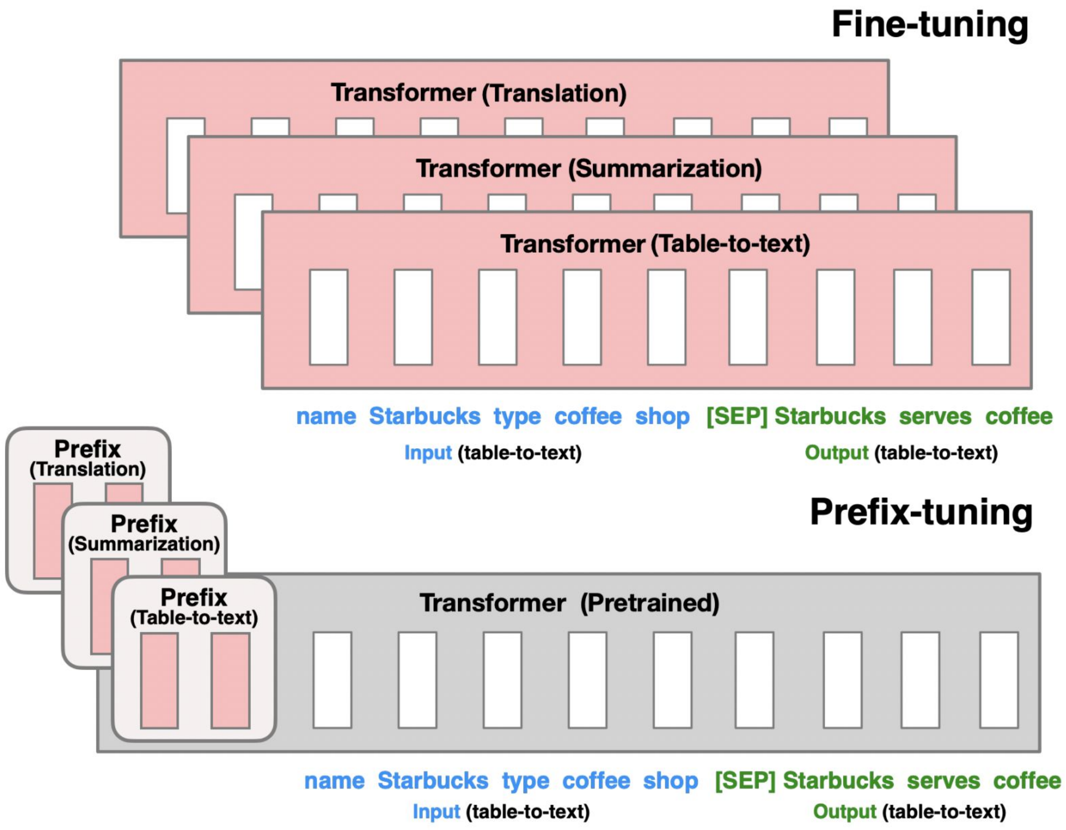
Xiang Lisa Li
Stanford University
xlisali@stanford.edu

Percy Liang
Stanford University
pliang@cs.stanford.edu

Published in ACL 2021, link

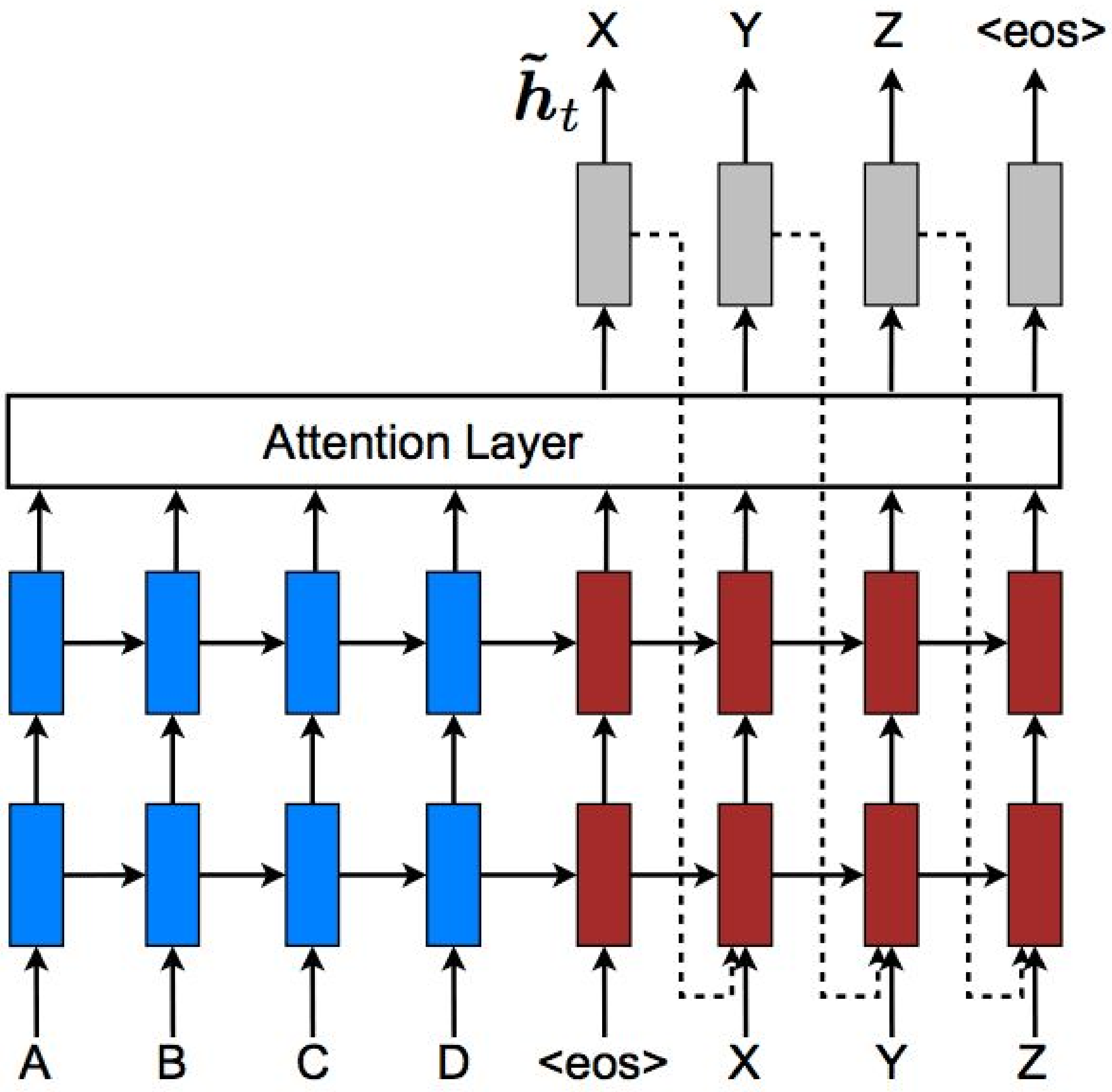
Methodology: Prefix-tuning

- Rather than designing a prompt manually, we can learn an optimal prefix for each task.
- Only ~0.1% of parameters need to be tuned! (adapter is 3.6%)

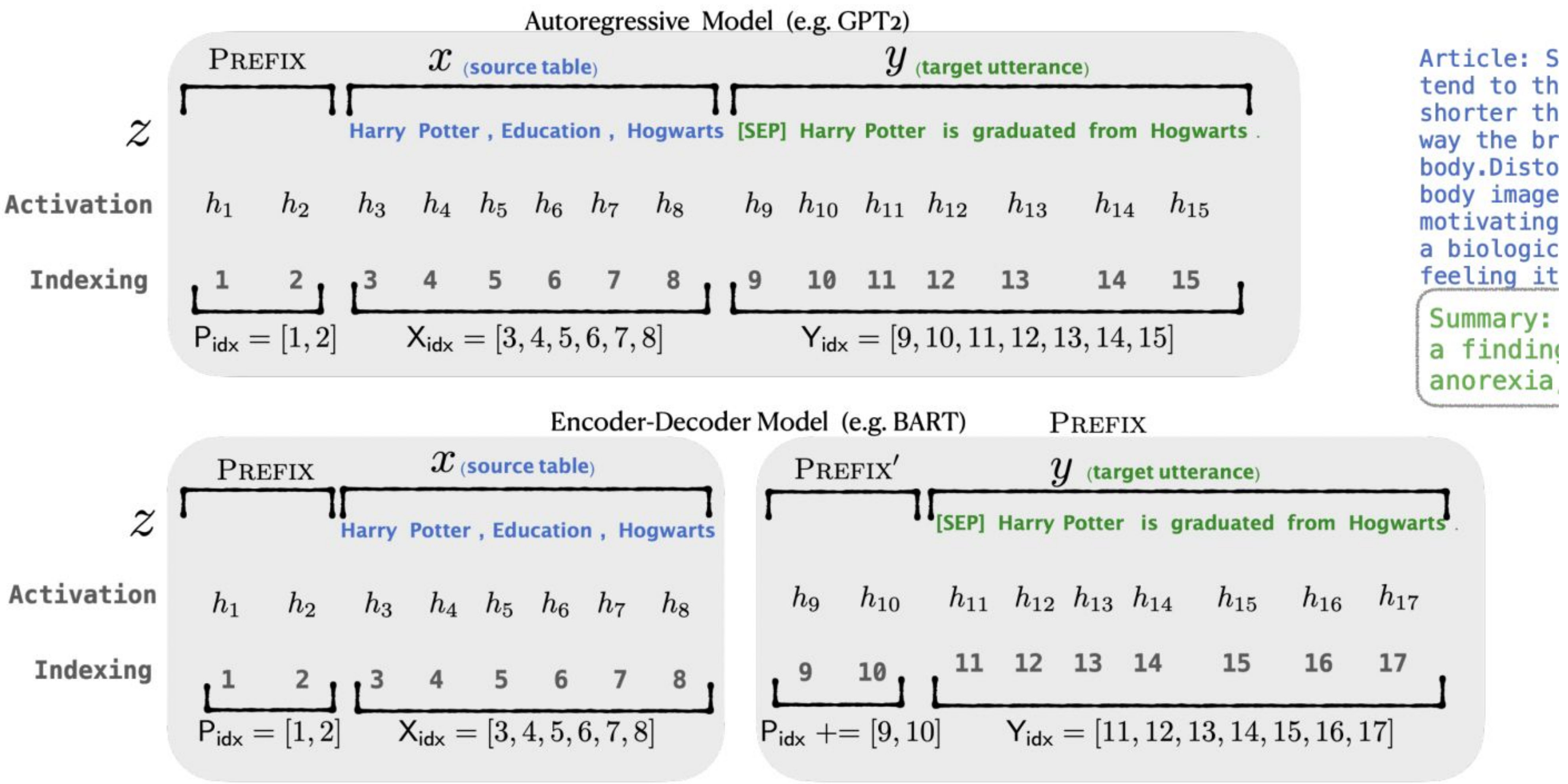


Methodology: Hidden State Tuning (Deep prompting)

- It's not just the prompts in the prefix that get tuned, it is also the hidden representations of later layer.
- Reparametrize using MLP to make training more stable.
  $P_\theta[i,:] = MLP_\theta(P'_\theta[i,:])$



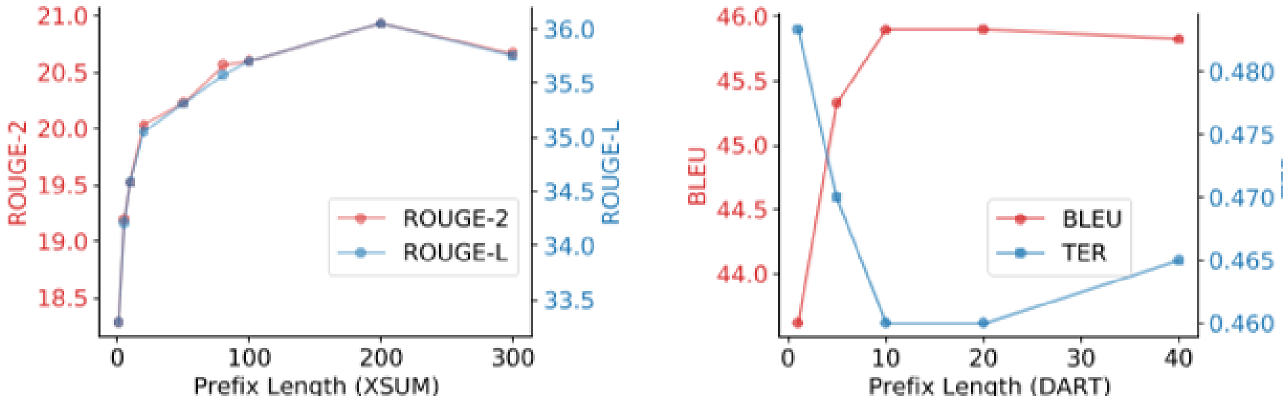
Methodology: Encoder-decoder Models



Ablation: Prefix Length

- Results:
  - table-to-text: 10
  - summarization: 200
- Overfitting is observed in longer prefix.

- Note: prefixes have a negligible impact on inference speed.



Compare to Prompt Length

Ablation: Embedding-only

- Embedding-only tuning is very sensitive to the learning rate and initialization.
- Tuning only the embedding layer is not enough expressivity to match prefix-tuning performance.
    - Future works (prompt tuning, etc.) disapprove this finding.
    - "All models are wrong, but some are useful" -- George Box
- Expressive power: discrete prompting < embedding-only ablation < prefix-tuning

	E2E				
	BLEU	NIST	MET	ROUGE	CIDEr
PREFIX	69.7	8.81	46.1	71.4	2.49
Embedding-only: EMB-{PrefixLength}					
EMB-1	48.1	3.33	32.1	60.2	1.10
EMB-10	62.2	6.70	38.6	66.4	1.75
EMB-20	61.9	7.11	39.3	65.6	1.85
Infix-tuning: INFIX-{PrefixLength}					
INFIX-1	67.9	8.63	45.8	69.4	2.42
INFIX-10	67.2	8.48	45.8	69.9	2.40
INFIX-20	66.7	8.47	45.8	70.0	2.42

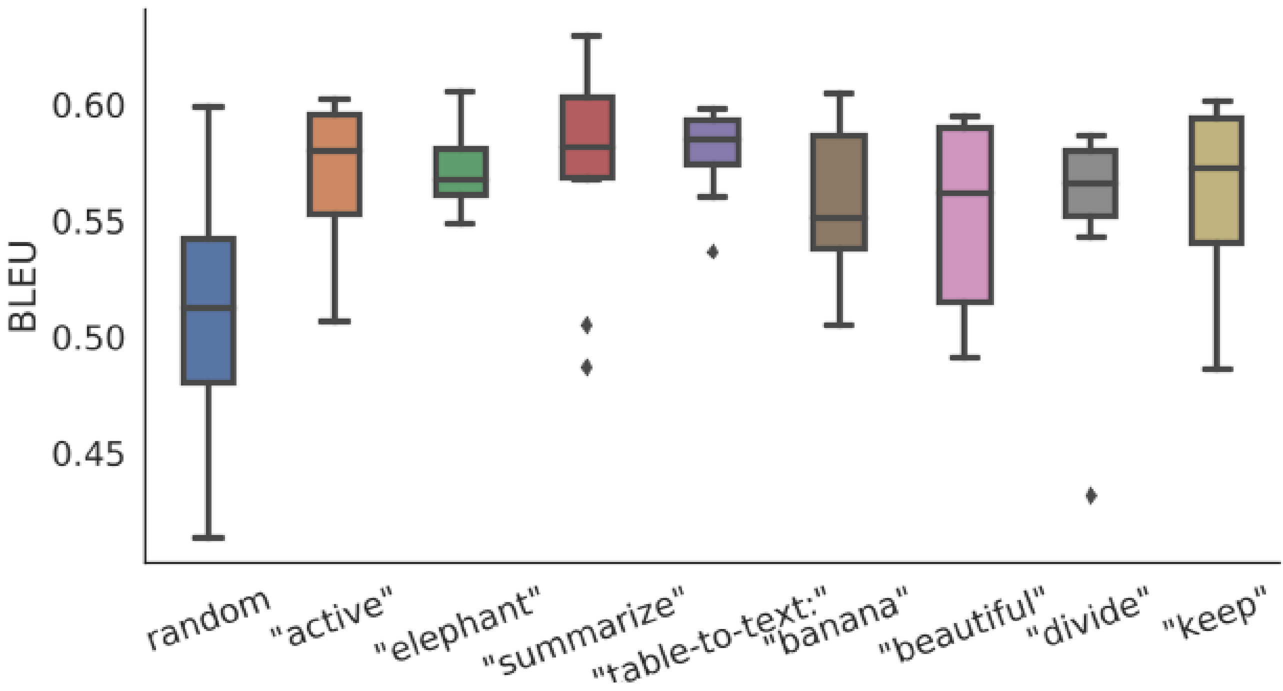
Ablation: Prefix vs Infix

- $[PREFIX; x; y]$ vs $[x; INFIX; y]$
- Infix tuning is worse than prefix tuning, since input embeddings cannot attend to infix.

	E2E BLEU	NIST	MET	ROUGE	CIDEr
PREFIX	69.7	8.81	46.1	71.4	2.49
Embedding-only: EMB-{PrefixLength}					
EMB-1	48.1	3.33	32.1	60.2	1.10
EMB-10	62.2	6.70	38.6	66.4	1.75
EMB-20	61.9	7.11	39.3	65.6	1.85
Infix-tuning: INFIX-{PrefixLength}					
INFIX-1	67.9	8.63	45.8	69.4	2.42
INFIX-10	67.2	8.48	45.8	69.9	2.40
INFIX-20	66.7	8.47	45.8	70.0	2.42

Ablation: Initialization

- Initializing randomly performs poorly and has high variance.
- It's better to initialize with words in the LM's vocabulary.
- It's even better to initialize with task specific words (summarize / table-to-text)



Compare to Prompt Initialization

Task: Table-to-Text

- Given a table, generate the information that the table contains in natural language.
- Datasets:
    - E2E
    - WebNLG
    - DART

Table-to-text Example

Table: name[Clowns] customer-rating[1 out of 5] eatType[coffee shop] food[Chinese] area[riverside] near[Clare Hall]

Textual Description: Clowns is a coffee shop in the riverside area near Clare Hall that has a rating 1 out of 5 . They serve Chinese food .

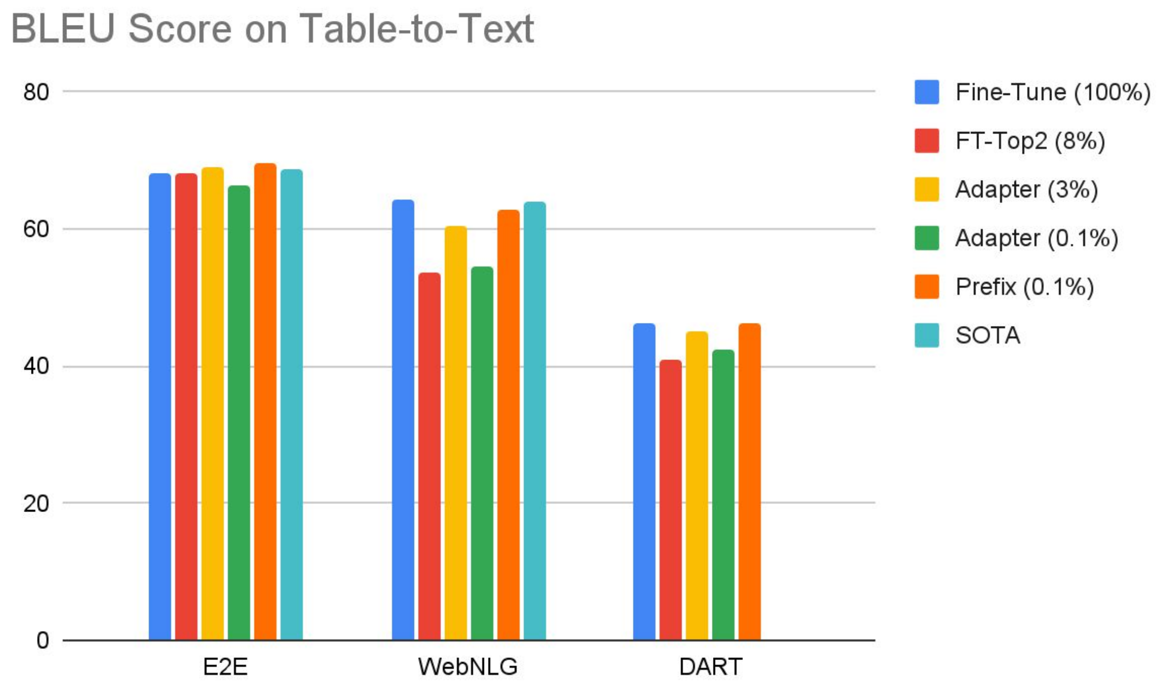
Task: Summarization

- Given a longer passage, generate a few summary sentences.
- Dataset: XSUM

SUMMARY: *A man and a child have been killed after a light aircraft made an emergency landing on a beach in Portugal.*

DOCUMENT: Authorities said the incident took place on Sao Joao beach in Caparica, south-west of Lisbon.
The National Maritime Authority said a middle-aged man and a young girl died after they were unable to avoid the plane.
[*6 sentences with 139 words are abbreviated from here.*]
Other reports said the victims had been sunbathing when the plane made its emergency landing.
[*Another 4 sentences with 67 words are abbreviated from here.*]
Video footage from the scene carried by local broadcasters showed a small recreational plane parked on the sand, apparently intact and surrounded by beachgoers and emergency workers.
[*Last 2 sentences with 19 words are abbreviated.*]

Results: Table to Text / Summarization

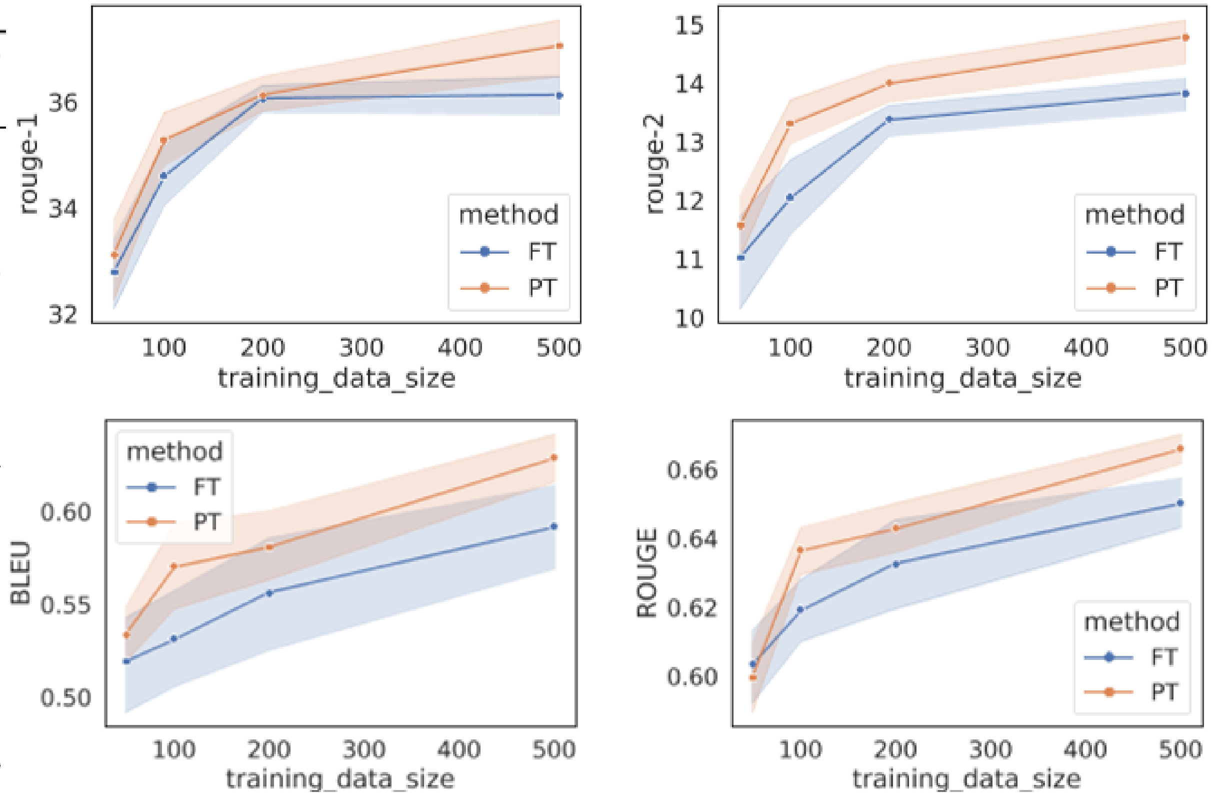


	R-1 ↑	R-2 ↑	R-L ↑
FINE-TUNE(Lewis et al., 2020)	45.14	22.27	37.25
PREFIX(2%)	43.80	20.93	36.05
PREFIX(0.1%)	42.92	20.03	35.05

- Come close to fine tuning!

Results: Low-data Setting

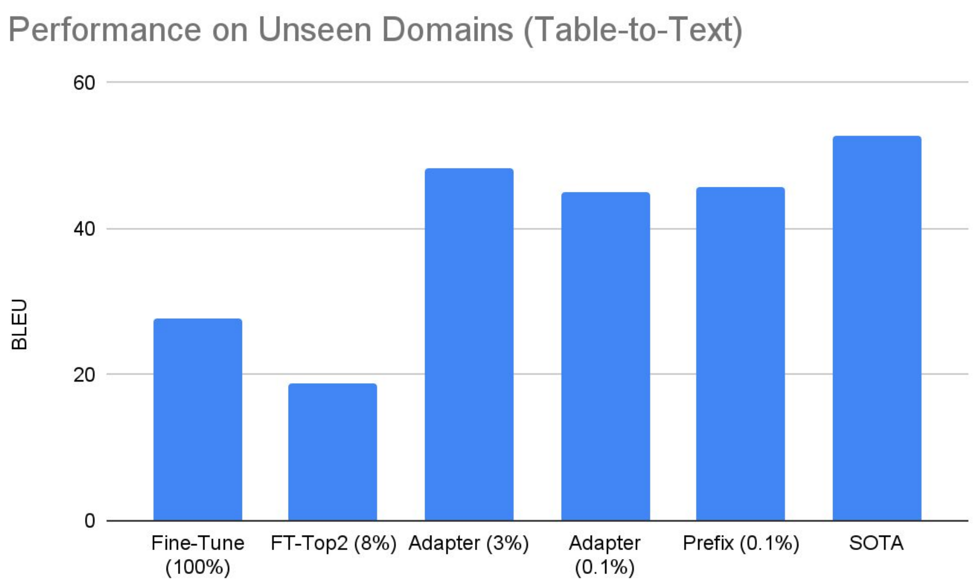
Source	name : The Eagle | type : coffee shop | food : Chinese | price : cheap | customer rating : average | area : riverside | family friendly : no | near : Burger King
Prefix (50)	The Eagle is a cheap Chinese coffee shop located near Burger King.
Prefix (100)	The Eagle is a cheap coffee shop located in the riverside near Burger King. It has average customer ratings.
Prefix (200)	The Eagle is a cheap Chinese coffee shop located in the riverside area near Burger King. It has average customer ratings.
Prefix (500)	The Eagle is a coffee shop that serves Chinese food. It is located in the riverside area near Burger King. It has an average customer rating and is not family friendly.
FT (50)	The Eagle coffee shop is located in the riverside area near Burger King.
FT (100)	The Eagle is a cheap coffee shop near Burger King in the riverside area. It has a low customer rating and is not family friendly.
FT (200)	The Eagle is a cheap Chinese coffee shop with a low customer rating. It is located near Burger King in the riverside area.
FT (500)	The Eagle is a cheap Chinese coffee shop with average customer ratings. It is located in the riverside area near Burger King.



- Prefix tuning outperforms fine-tuning when training data size is low.

Results: Transfer Learning

- Prefix tuning (and also Adapter) outperform fine-tuning on generalization to different domains!
- Why? Author: "Since the language models are pretrained on general purpose corpus, **preserving the LM parameters** might help generalization to domains unseen during training."



The Power of Scale for Parameter-Efficient Prompt Tuning

Brian Lester* **Rami Al-Rfou** **Noah Constant**
Google Research
{brianlester,rmyeid,nconstant}@google.com

Published in EMNLP 2021, link

Methodology: Prompt-tuning

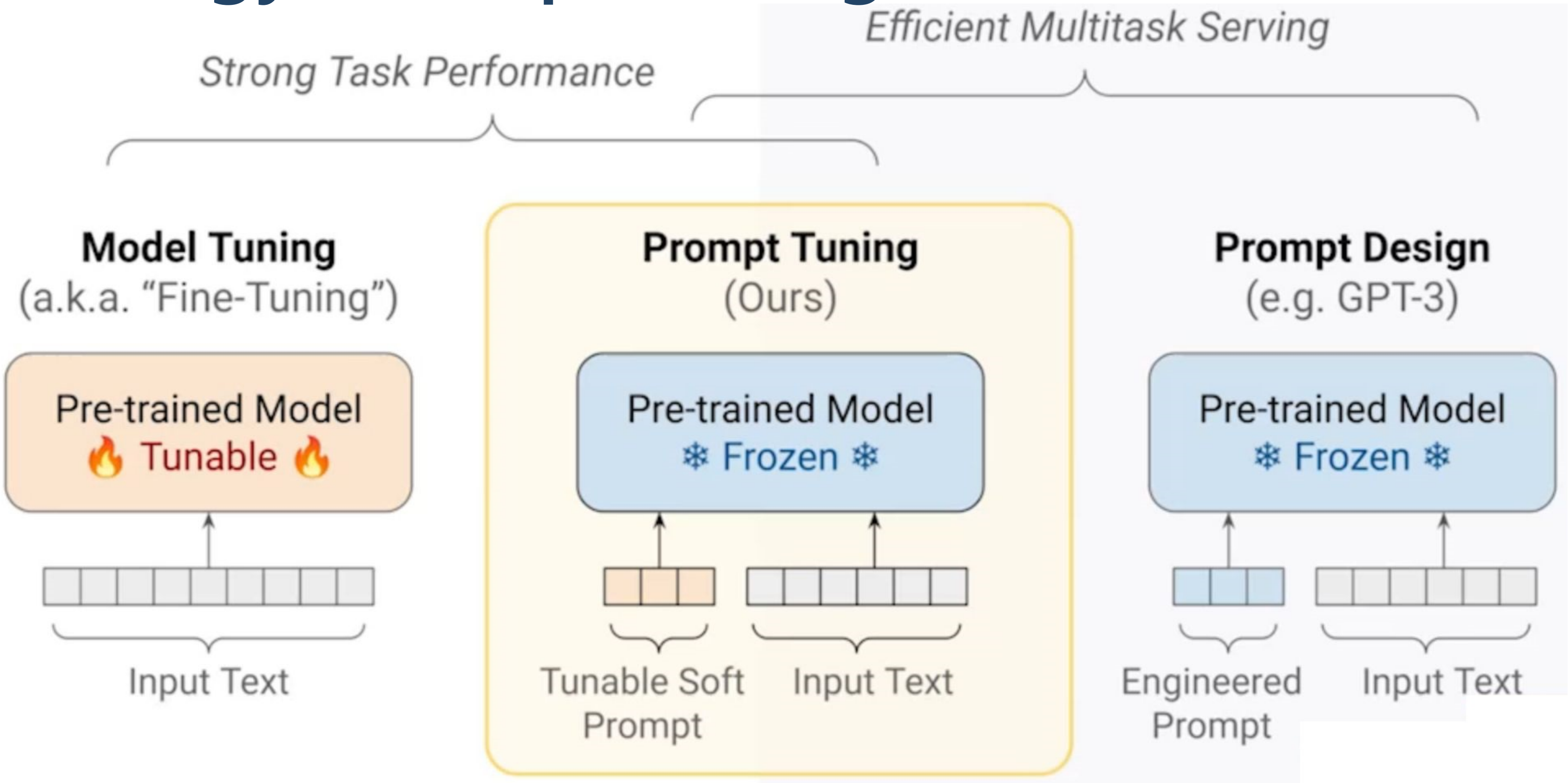


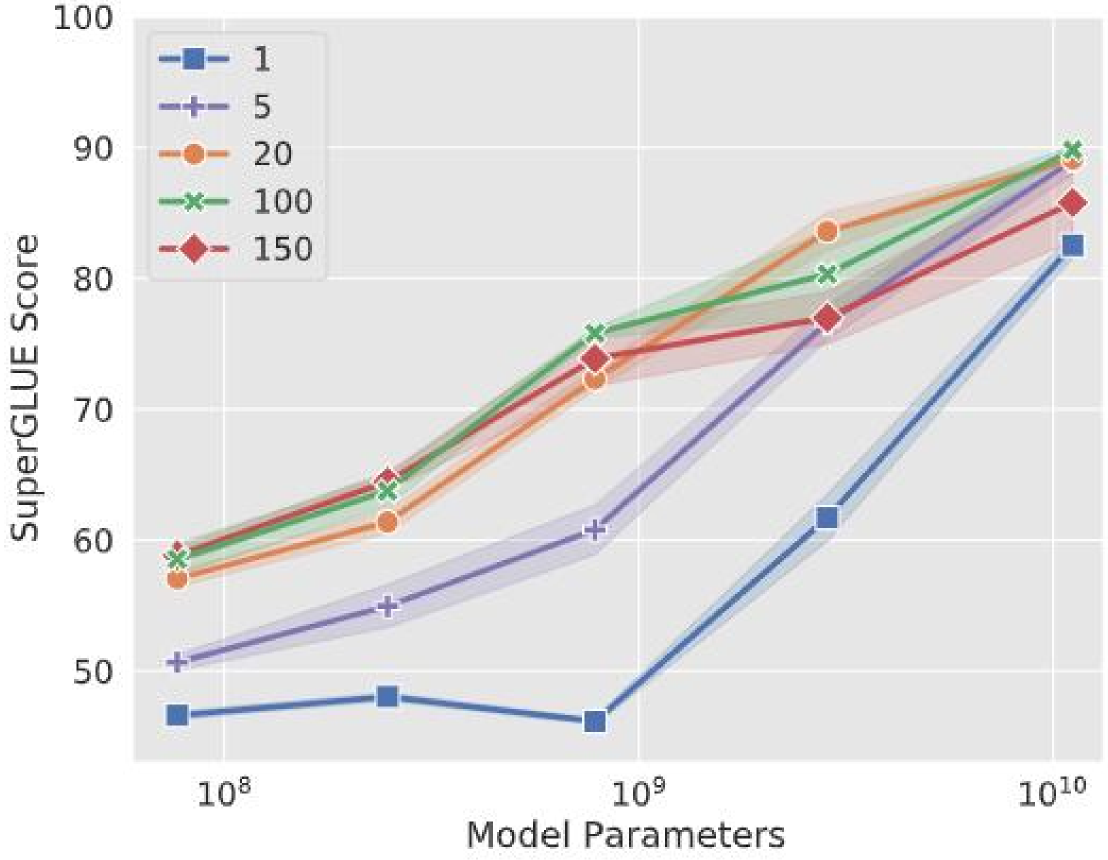
Image Credit: Brian Lester

Experiment Setup

- Model backbone: pre-trained T5 models (Small, Base, Large, XL, XXL)
- Ablations:
  - Prompt Length
  - Prompt Initialization
  - Pre-training Objective
- Benchmark: SuperGLUE (a collection of eight challenging English language understanding tasks)
  - Each prompt trains on a single task
  - Each dataset is translated into a text-to-text format

Ablation: Prompt Length

- Increasing prompt length beyond a single token is critical to achieve good performance.
- The XXL model still gives strong results with a single-token prompt: **The larger the model, the less conditioning signal is needed**
- Longer than 100 tokens is bad (mildly).

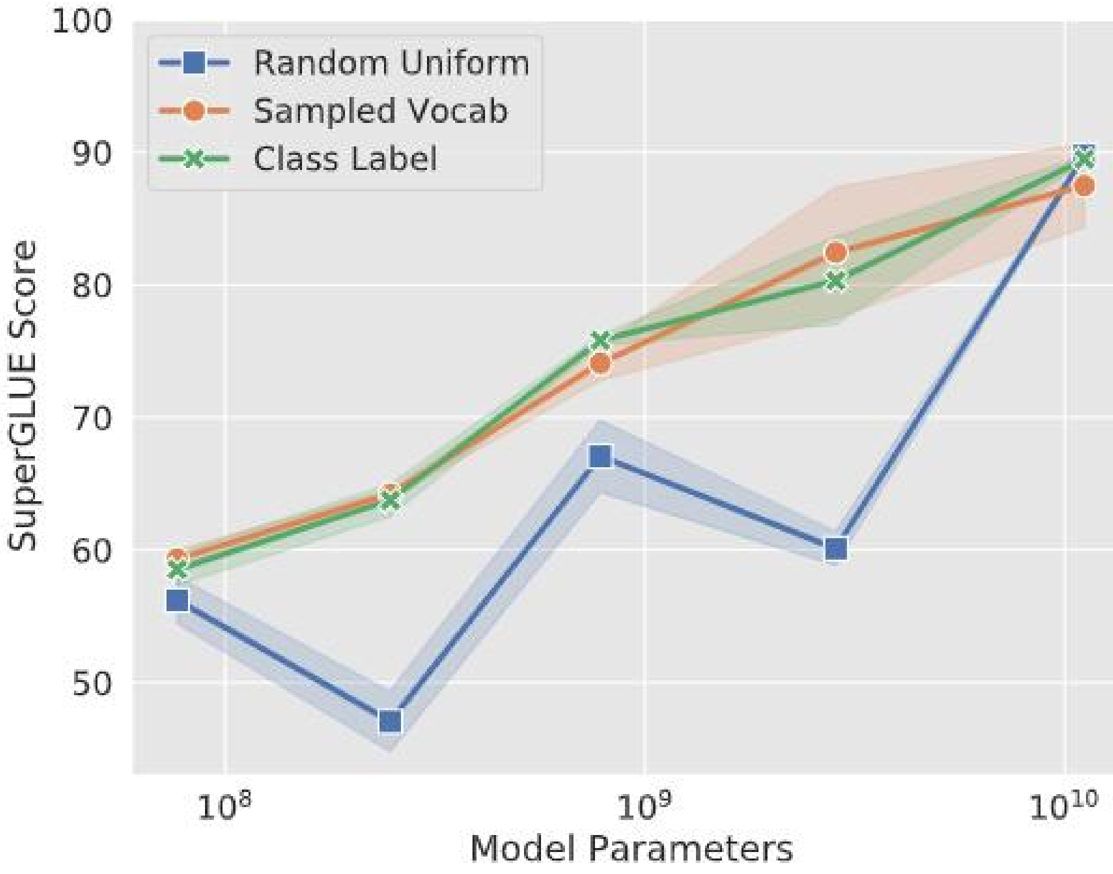


(a) Prompt length

Compare to Prefix Length

Ablation: Initialization

- **Random initialization**: sample uniformly from $[-0.5, 0.5]$
- **Sampled vocabulary**: 5000 most "common" tokens in SentencePiece
- **Class label**
  - Average the token embeddings if multiple
  - Fall back to sampled vocab if run out of class labels

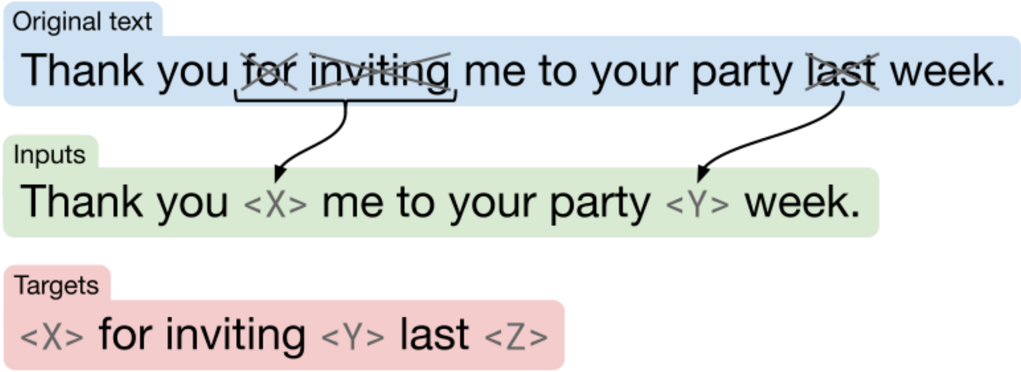


(b) Prompt initialization

Compare to Prefix Initialization

T5 Pre-training Objective

- Randomly sample and then drop out 15% of tokens in the input sequence.
- Consecutive spans of dropped-out tokens are replaced by a single **sentinel token**.
- Predict **only** dropped-out tokens and corresponding sentinel tokens
  - Motivation: to reduce the computational cost of pre-training

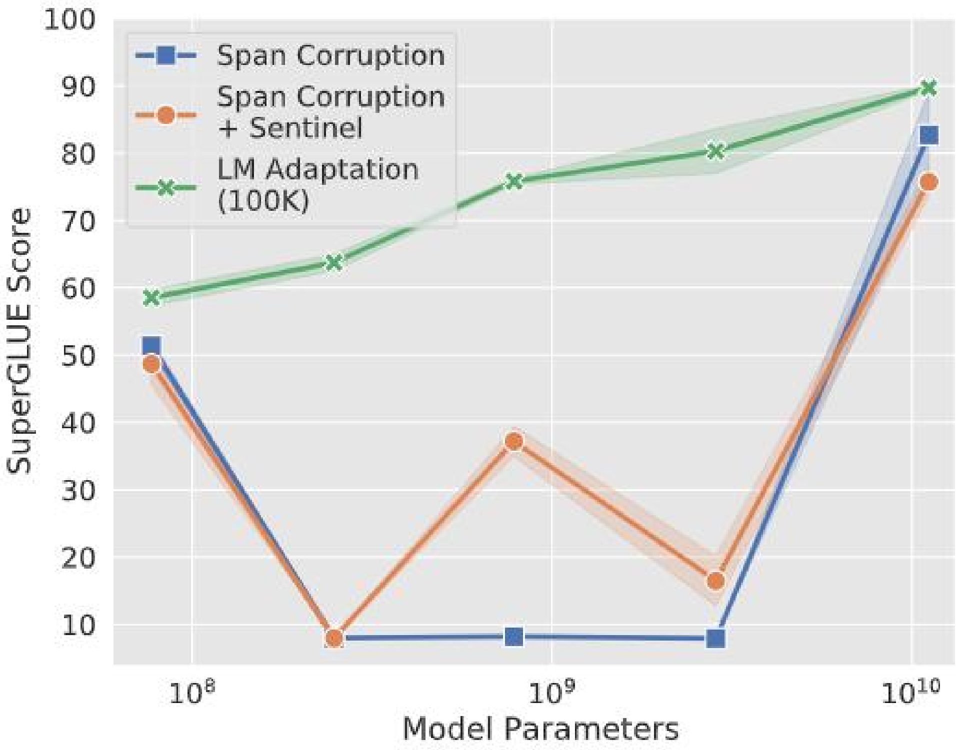


Paper: Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer (T5)

Ablation:
Pre-training Objective

Motivation: T5 has never been asked to predict truly natural targets (free of sentinel tokens).

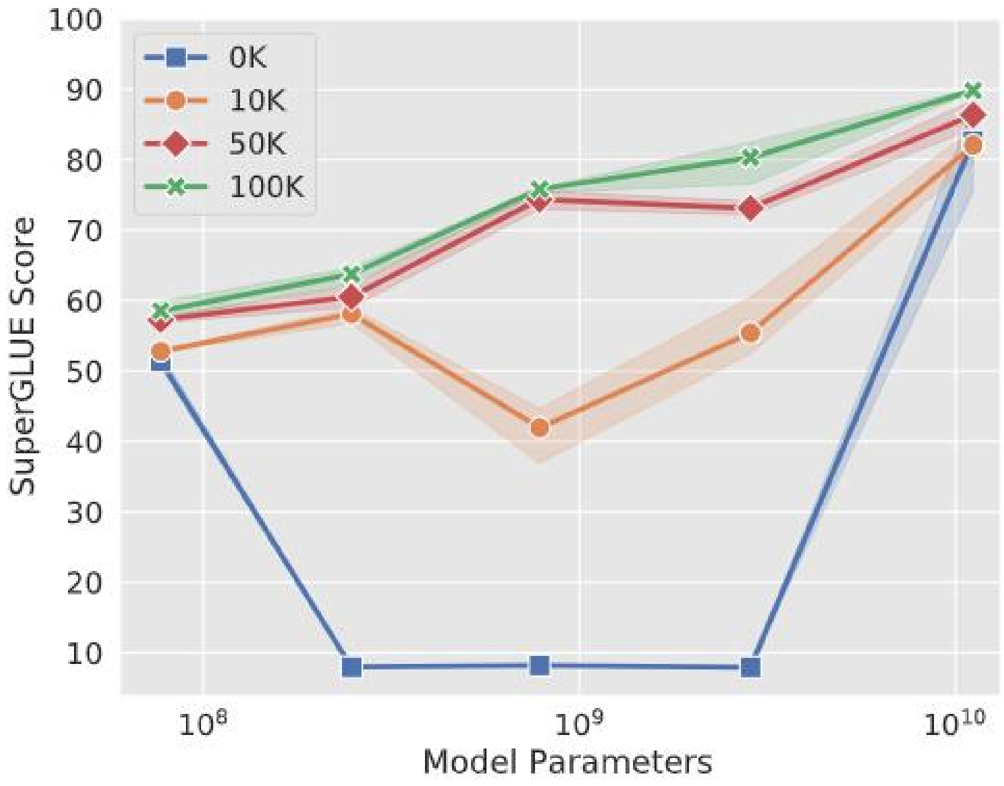
- **Span Corruption**: use pre-trained T5 off-the-shelf
- **Span Corruption + Sentinel**: use the same model, but prepend a sentinel before downstream targets (as a workaround).
- **LM Adaptation**: continue pre-training using the "LM" objective
  - Adaptation happens **only once**



(c) Pre-training method

Ablation: LM Adaption

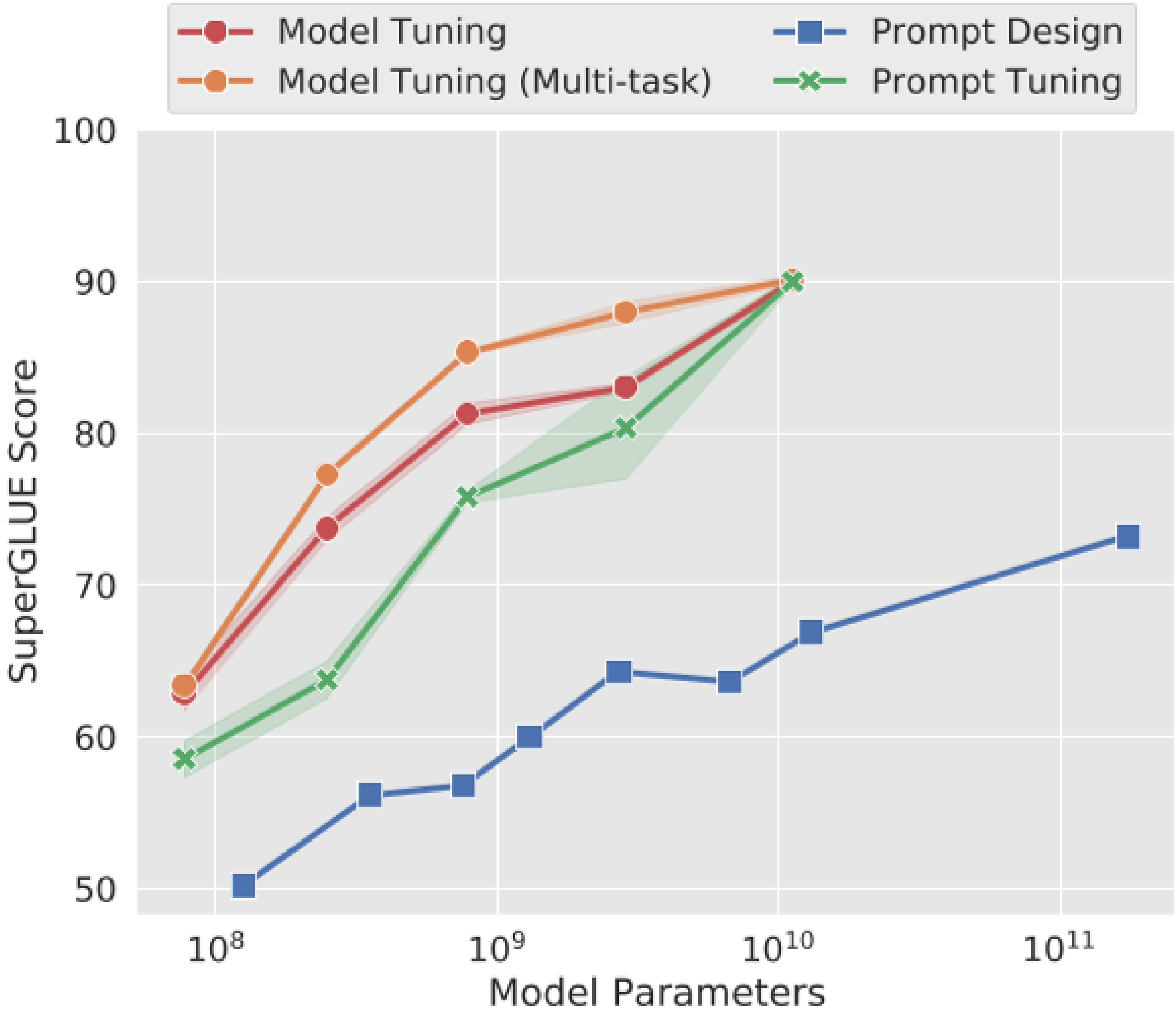
- Longer adaptation provides additional gains, up to 100K steps.
- At the largest model size, the gains from adaptation are small
  - ***The Power of Scale***
- Observation: mid-sized models **never** output legal class labels and thus score 0%
  - copying subspans from the input
  - predicting an empty string



(d) LM adaptation steps

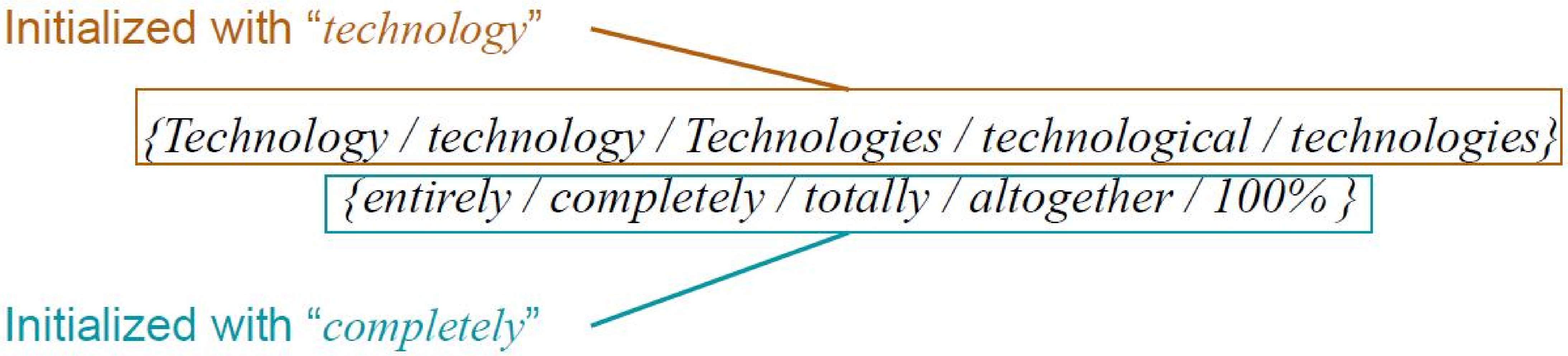
Closing the Gap

- Prompt tuning becomes more competitive with model tuning as scale increases
- Prompt tuning beats GPT-3 prompt design by a large margin



Interpretability

- Compute the nearest neighbors (cosine distance) to each soft prompt from the frozen model's vocabulary.
- Observation:
  - The top-5 nearest neighbors form tight semantic clusters.
  - The class labels persist through training.
- Related work: Prompt Waywardness



Towards a Unified View of Parameter-Efficient Transfer Learning

Junxian He*
Carnegie Mellon University
junxianh@cs.cmu.edu

Chunting Zhou*
Carnegie Mellon University
chuntinz@cs.cmu.edu

Xuezhe Ma
University of Southern California
xuezhema@isi.edu

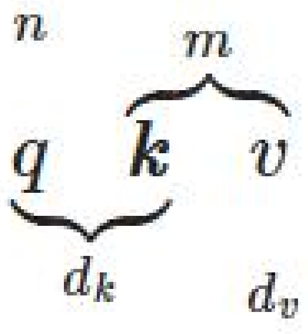
Taylor Berg-Kirkpatrick
UC San Diego
tberg@eng.ucsd.edu

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

Published in ICLR 2022, link

Transformer Recap: Attention

- Dot-Product: $Q_{n \times d_k} K^T_{m \times d_k} \in \mathbb{R}^{n \times m}$
- Softmax normalization (by query/row): $\mathrm{softmax}(\frac{QK^T}{\sqrt{d_k}}) \triangleq W \in \mathbb{R}^{n \times m}$
- Weighted sum of $V = \begin{bmatrix} - & v_1 & - \\ - & v_2 & - \\ - & \cdots & - \\ - & v_m & - \end{bmatrix}$:
  - For query i: $\sum_{j=1}^{m} W_{ij} v_j = W_i V \in \mathbb{R}^{1 \times d_v}$
  - For all queries: $W_{n \times m} V_{m \times d_v} \in \mathbb{R}^{n \times d_v}$
- Scaled Dot-Product Attention:



$$\mathrm{Attention}(Q_{n \times d_k}, K_{m \times d_k}, V_{m \times d_v}) = \mathrm{softmax}(\frac{QK^T}{\sqrt{d_k}}) V \in \mathbb{R}^{n \times d_v}$$

Transformer Recap: Multi-head Attention

$$\mathrm{head_i} = \mathrm{Attention}(Q_{n\times d_{model}} W^{Q}_{d_{model}\times d_k}, K_{m\times d_{model}} W^{K}_{d_{model}\times d_k}, V_{m\times d_{model}} W^{V}_{d_{model}\times d_v}) \in \mathbb{R}^{n\times d_v}$$

$$\mathrm{head} = \mathrm{Concat}(\mathrm{head_1}, \cdots, \mathrm{head_h}) \in \mathbb{R}^{n\times d_v\cdot N_h}$$

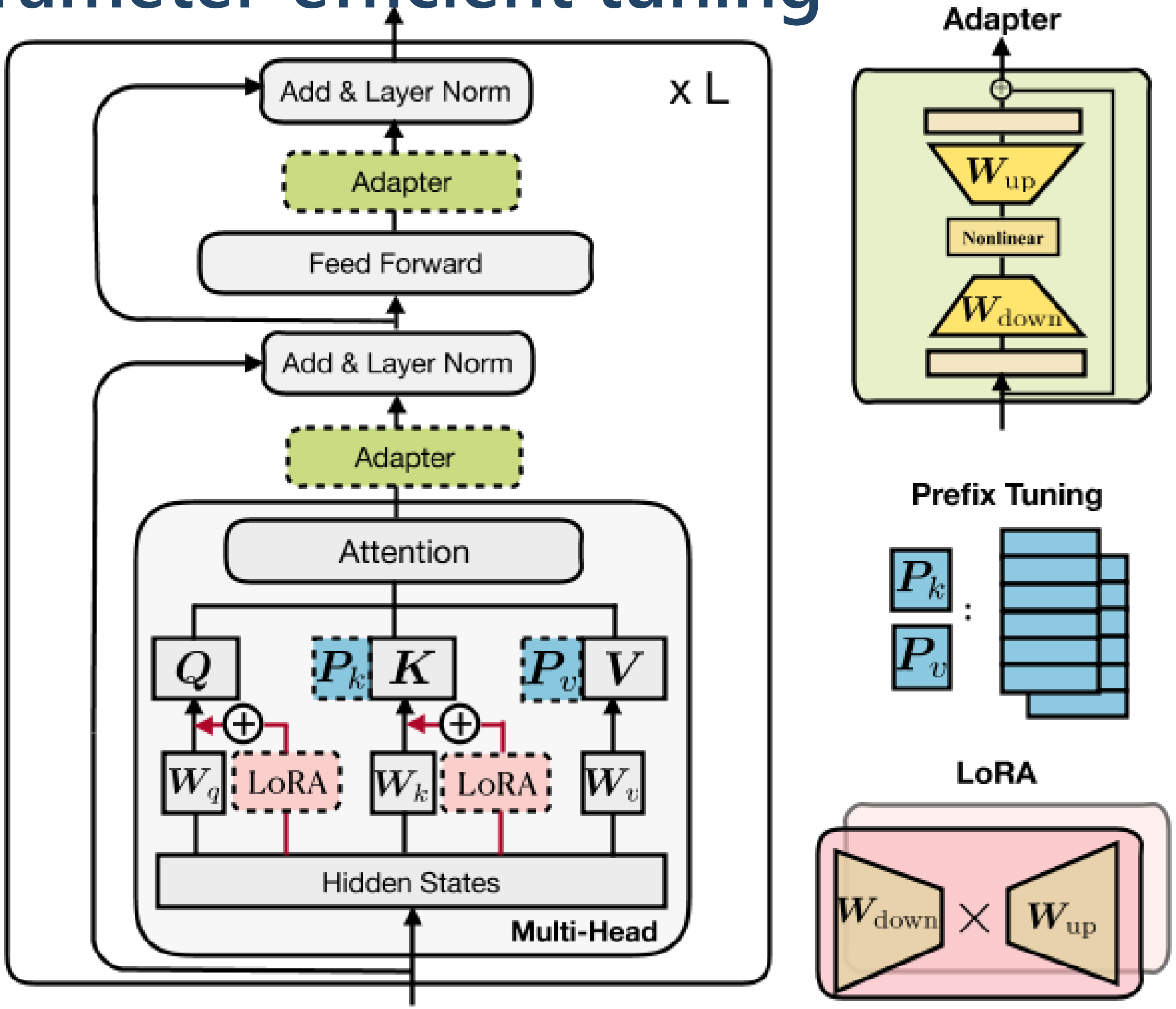
$$\mathrm{MHA}(Q_{n\times d_{model}}, K_{m\times d_{model}}, V_{m\times d_{model}}) = \mathrm{head}_{n\times d_v\cdot N_h} W^{O}_{d_v\cdot h\times d_{model}} \in \mathbb{R}^{n\times d_{model}}$$

where, $d_k = d_v = \dfrac{d_{model}}{N_h} \triangleq d_h$

For query $x \in \mathbb{R}^{d_{model}}$ on sequence $C \in \mathbb{R}^{m\times d_{model}}$:

$$\mathrm{head_i} = \mathrm{Attention}\left(xW^{Q}, CW^{K}, CW^{V}\right)$$

Recap: Parameter-efficient tuning



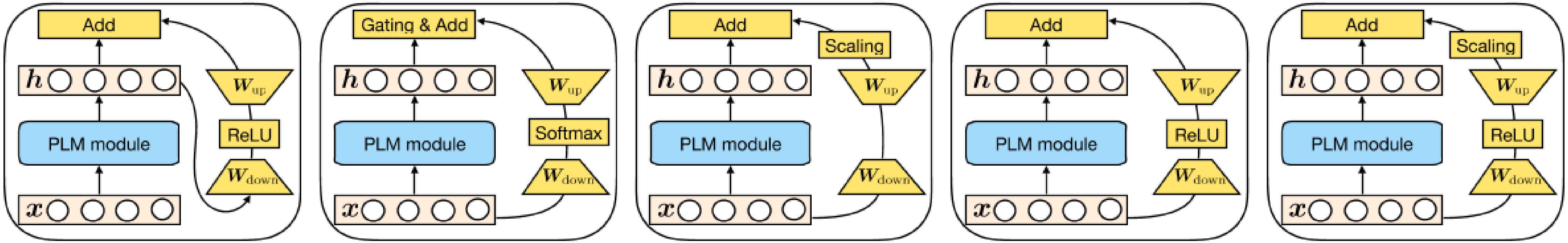
Rethinking Prefix

$$\begin{aligned}
\text{head}_{\text{prefix}} &= \text{Attn}\left(Q, \begin{bmatrix} P_k \\ K \end{bmatrix}, \begin{bmatrix} P_v \\ V \end{bmatrix}\right) \\
&= \text{softmax}\left(Q \begin{bmatrix} P_k \\ K \end{bmatrix}^T\right) \begin{bmatrix} P_v \\ V \end{bmatrix} \qquad \left(\text{ignore } \frac{1}{\sqrt{d}} \text{ for ease of notation}\right) \\
&= \text{softmax}\left(\begin{bmatrix} QP_k^T & QK^T \end{bmatrix}\right) \begin{bmatrix} P_v \\ V \end{bmatrix} \\
&= (1 - \lambda(Q))\text{softmax}\left(QK^T\right) V + \lambda(Q)\text{softmax}\left(QP_k^T\right) P_v \\
&= (1 - \lambda(Q))\underbrace{\text{Attn}(Q, K, V)}_{\text{standard attention}} + \lambda(Q)\underbrace{\text{Attn}(Q, P_k, P_v)}_{\text{independent of } K, V}
\end{aligned}$$

$$\lambda(Q) = \frac{\sum_i \exp\left(QP_k^T\right)_i}{\sum_i \exp\left(QP_k^T\right)_i + \sum_j \exp\left(QK^T\right)_j}$$

A Unified View

- Adapter: $h \leftarrow h + \mathrm{ReLU}\left(hW_{\text{down}}\right)W_{\text{up}}$
- Prefix: $h \leftarrow (1-\lambda(x))h + \lambda(x)\mathrm{softmax}\left(xW_1\right)W_2$
  - $W_1 \triangleq W^Q P_k^T, W_2 \triangleq P_v$
- LoRA: $h \leftarrow h + s \cdot xW_{\text{down}}W_{\text{up}}$

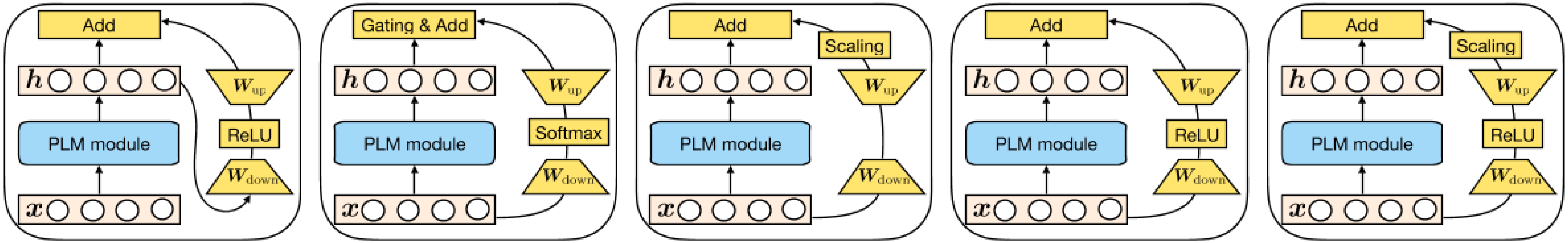


(a) Adapter (b) Prefix Tuning (c) LoRA (d) Parallel Adapter (e) Scaled PA

Figure 3: Graphical illustration of existing methods and the proposed variants. “PLM module” represents a

The Unified Framework

Method	$\Delta \boldsymbol{h}$ functional form	insertion form	modified representation	composition function
Existing Methods				
Prefix Tuning	$\mathrm{softmax}(\boldsymbol{x}\boldsymbol{W}_q\boldsymbol{P}_k^\top)\boldsymbol{P}_v$	parallel	head attn	$\boldsymbol{h} \leftarrow (1-\lambda)\boldsymbol{h} + \lambda\Delta\boldsymbol{h}$
Adapter	$\mathrm{ReLU}(\boldsymbol{h}\boldsymbol{W}_{\text{down}})\boldsymbol{W}_{\text{up}}$	sequential	ffn/attn	$\boldsymbol{h} \leftarrow \boldsymbol{h} + \Delta\boldsymbol{h}$
LoRA	$\boldsymbol{x}\boldsymbol{W}_{\text{down}}\boldsymbol{W}_{\text{up}}$	parallel	attn key/val	$\boldsymbol{h} \leftarrow \boldsymbol{h} + s \cdot \Delta\boldsymbol{h}$
Proposed Variants				
Parallel adapter	$\mathrm{ReLU}(\boldsymbol{h}\boldsymbol{W}_{\text{down}})\boldsymbol{W}_{\text{up}}$	parallel	ffn/attn	$\boldsymbol{h} \leftarrow \boldsymbol{h} + \Delta\boldsymbol{h}$
Muti-head parallel adapter	$\mathrm{ReLU}(\boldsymbol{h}\boldsymbol{W}_{\text{down}})\boldsymbol{W}_{\text{up}}$	parallel	head attn	$\boldsymbol{h} \leftarrow \boldsymbol{h} + \Delta\boldsymbol{h}$
Scaled parallel adapter	$\mathrm{ReLU}(\boldsymbol{h}\boldsymbol{W}_{\text{down}})\boldsymbol{W}_{\text{up}}$	parallel	ffn/attn	$\boldsymbol{h} \leftarrow \boldsymbol{h} + s \cdot \Delta\boldsymbol{h}$



(a) Adapter (b) Prefix Tuning (c) LoRA (d) Parallel Adapter (e) Scaled PA

Figure 3: Graphical illustration of existing methods and the proposed variants. "PLM module" represents a

Experiment Setup

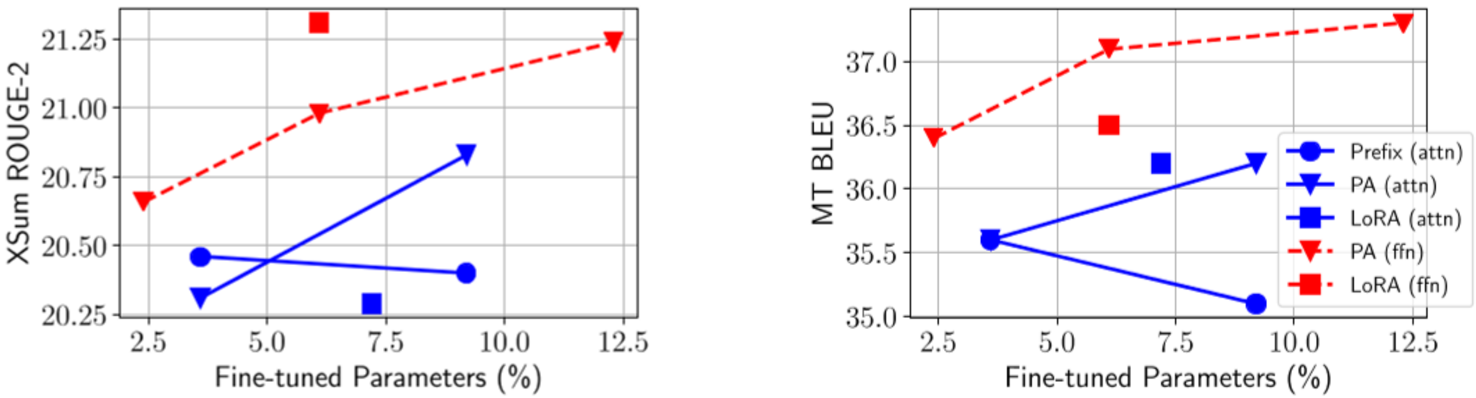
- Model backbone: mBART-LARGE (and RoBERTa-BASE)
- Datasets:
  - Encoder-Decoder (mBART-LARGE):
    - XSum: English singledocument summarization
    - WMT: en-ro translation
  - Encoder only (RoBERTa-BASE):
    - MNLI: English natural language inference
    - SST2: English sentiment classification

Ablation: Sequential or Parallel ?

Method	# params	XSum (R-1/2/L)	MT (BLEU)
Prefix, l=200	3.6%	43.40/20.46/35.51	35.6
SA (attn), r=200	3.6%	42.01/19.30/34.40	35.3
SA (ffn), r=200	2.4%	43.21/19.98/35.08	35.6
PA (attn), r=200	3.6%	43.58/20.31/35.34	35.6
PA (ffn), r=200	2.4%	**43.93/20.66/35.63**	**36.4**

Parallel!

Ablation: Attention or FFN ?



Method	# params	MT (BLEU)
PA (attn), r=200	3.6%	35.6
Prefix, l=200	3.6%	35.6
MH PA (attn), r=200	3.6%	35.8
Prefix, l=30	0.1%	35.2
-gating, l=30	0.1%	34.9
PA (ffn), r=30	0.1%	33.0
PA (attn), r=30	0.1%	33.7
MH PA (attn), r=30	0.1%	**35.3**

Figure 5: Results on XSum (left) and en-ro (right). PA represents parallel adapter. Blue and red markers modifications at attention and FFN sub-layers respectively (best viewed in color).

- High parameter budget: FFN
- Low parameter budget: Attention
- Observation: overfit of prefix also exists

"We hypothesize that this is because the FFN learns task-specific textual patterns, while attention learns pairwise positional interactions which do not require large capacity for adapting to new tasks."

Ablation: Composition Function

Method (# params)	XSum (R-1/2/LSum)
LoRA (6.1%), s=4	44.59/21.31/36.25
LoRA (6.1%), s=1	44.17/20.83/35.74
PA (6.1%)	44.35/20.98/35.98
Scaled PA (6.1%), s=4	**44.85/21.54/36.58**
Scaled PA (6.1%), trainable s	44.56/21.31/36.29

e one while being easily applicable.

- LoRA ($s = 4$) performs better
- A learned scalar does not give better results.

Mix-And-Match Adapter (MAM Adapter)

An effective integration:

- use Prefix-Tuning ($l = 30$, small bottleneck dimension) at the attention sub-layers
- use Scaled Parallel Adapter ($r = 512$) to modify FFN representation

Method	# params	XSum (R-1/2/L)	MT (BLEU)
Full fine-tuning[†]	100%	45.14/22.27/37.25	37.7
Full fine-tuning (our run)	100%	44.81/21.94/36.83	37.3
Bitfit (Ben Zaken et al., 2021)	0.1%	40.64/17.32/32.19	26.4
Prompt tuning (Lester et al., 2021)	0.1%	38.91/15.98/30.83	21.0
Prefix tuning (Li & Liang, 2021), l=200	3.6%	43.40/20.46/35.51	35.6
Pfeiffer adapter (Pfeiffer et al., 2021), r=600	7.2%	44.03/20.89/35.89$_{\pm.13/.10/.08}$	36.9$_{\pm.1}$
LoRA (ffn), r=102	7.2%	44.53/21.29/36.28$_{\pm.14/.07/.10}$	36.8$_{\pm.3}$
Parallel adapter (PA, ffn), r=1024	12.3%	44.71/21.41/36.41$_{\pm.16/.17/.16}$	37.2$_{\pm.1}$
PA (attn, r=30) + PA (ffn, r=512)	6.7%	44.29/21.06/36.12$_{\pm.31/.19/.18}$	37.2$_{\pm.1}$
Prefix tuning (attn, l=30) + LoRA (ffn, r=102)	6.7%	44.84/21.71/36.77$_{\pm.07/.05/.03}$	37.0$_{\pm.1}$
MAM Adapter (our variant, l=30, r=512)	6.7%	**45.06/21.90/36.87**$_{\pm.08/.01/.04}$	**37.5**$_{\pm.1}$

Number of tunable parameters

Method	number of parameters
Prompt Tuning	$l \times d$
Prefix Tuning (attn)	$2ld \times 3 \times 12$
Adapter variants (attn)	$2rd \times 3 \times 12$
Adapter variants (ffn)	$2rd \times 2 \times 12$
LoRA (attn)	$4rd \times 3 \times 12$
LoRA (ffn)	$10rd \times 2 \times 12$
MAM Adapter (our proposed model)	$2ld \times 3 \times 12 + 2rd \times 2 \times 12$