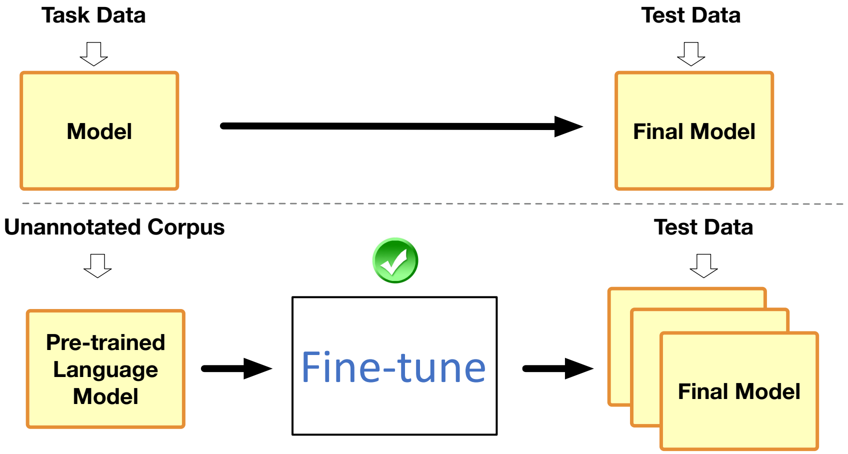
On Transferability of Prompt Tuning for Natural Language Processing

Yusheng Su*, **Xiaozhi Wang***, Yujia Qin, Chi-Min Chan, Yankai Lin, Huadong Wang,
Kaiyue Wen, Zhiyuan Liu, Peng Li, Juanzi Li, Lei Hou, Maosong Sun, Jie Zhou

Department of Computer Science and Technology, Tsinghua University, Beijing, 100084, China

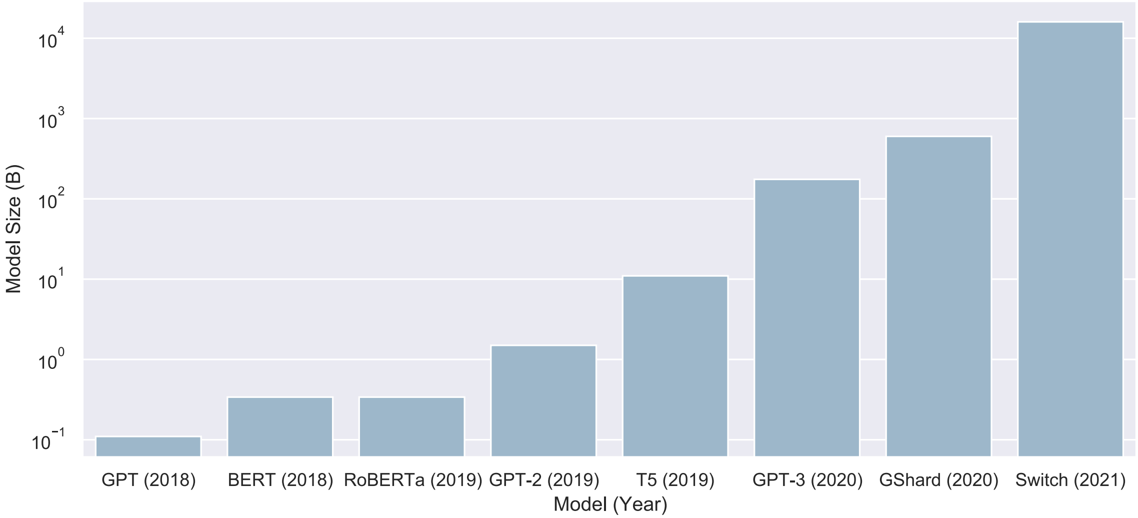
Background

Development of PLM



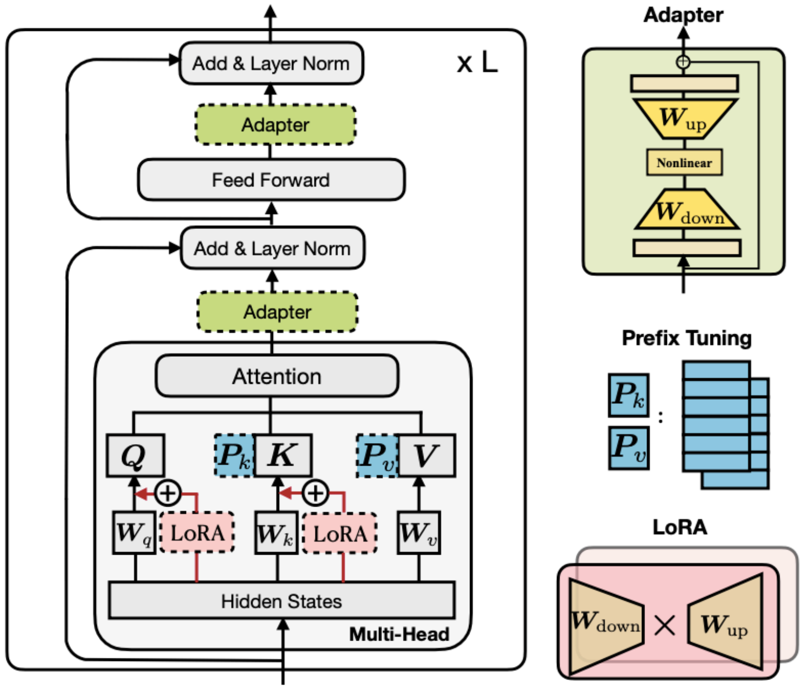
Fine-tuning paradigm becomes de-facto standard

Challenge in Fine-tuning



Background

- **Parameter-Efficient Tuning (PET) Methods**
  - PET methods only optimize a small part of parameters for downstream tasks while freezing the rest of the parameters of the PLM. [6]
  - Methods: Adapter [1], Prefix [2], LoRA [3], BitFit [4], Prompt [5], etc.

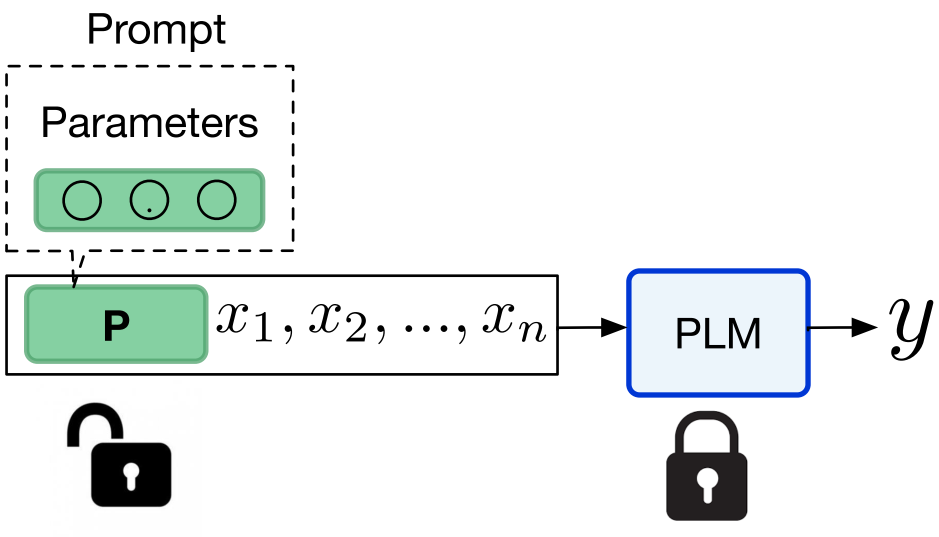


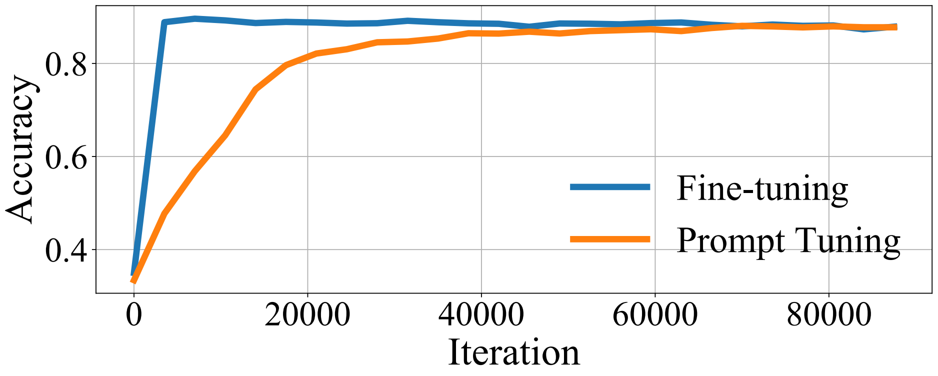
[6]

[1] Neil, et al., Parameter-Efficient Transfer Learning for NLP, ICML, 2019.
[2] Li, et al., Prefix-Tuning: Optimizing Continuous Prompts for Generation, ACL, 2021.
[3] Hu, et al., LoRA: Low-Rank Adaptation of Large Language Models, ICLR, 2022.
[4] Zaken, et al., BitFit: Simple Parameter-efficient Fine-tuning for Transformer-based Masked Language-models, ACL, 2022.
[5] Lester et al., The Power of Scale for Parameter-Efficient Prompt Tuning, EMNLP, 2021.
[6] He et al., Towards a Unified View of Parameter-Efficient Transfer Learning, ICLR, 2022.

Background

- **Prompt Tuning (PT)**
  - Advantage: Lowest computation costs
  - Challenge: Slow Convergence



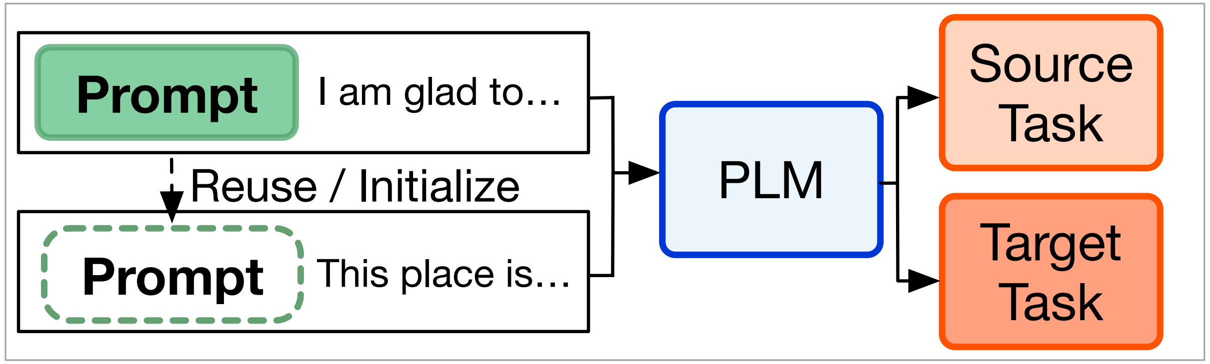


$$L = p(y|\boldsymbol{P}, x_1, \ldots, x_n)$$

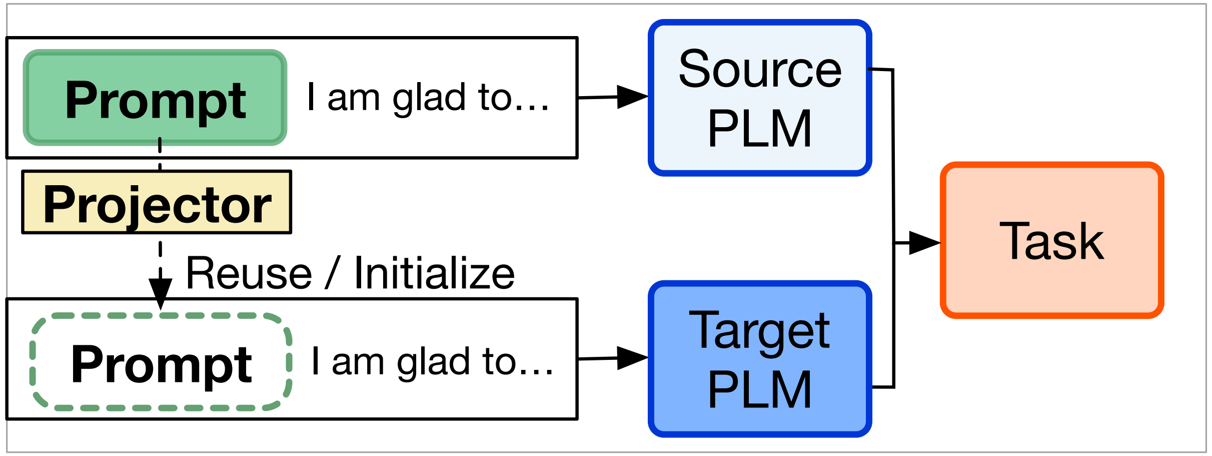
On Transferability of Prompt Tuning for NLP

- **Prompt Tuning (PT)**
  - Solution: Transferring the trained prompts
    - Cross-Task Transfer
    - Cross-Model Transfer

Cross-Task Transfer

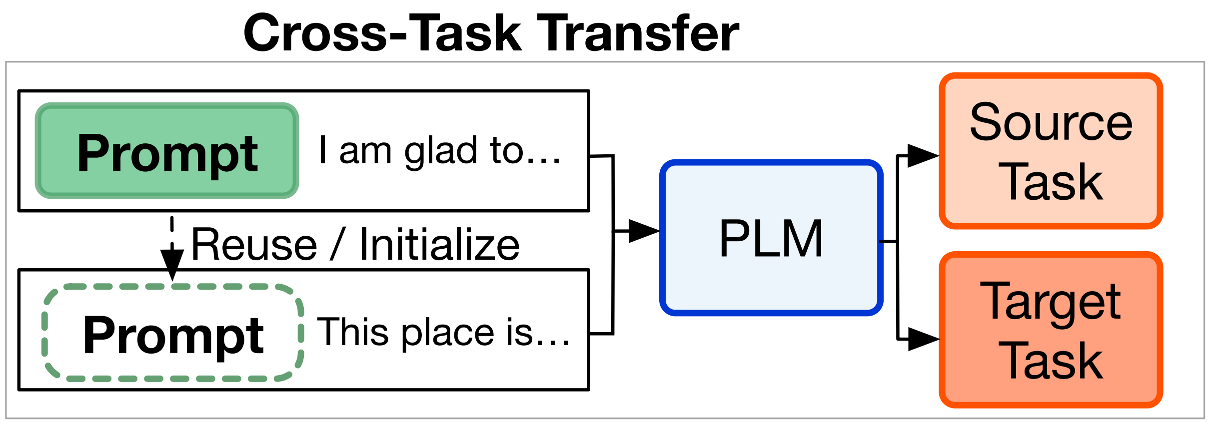


Cross-Model Transfer

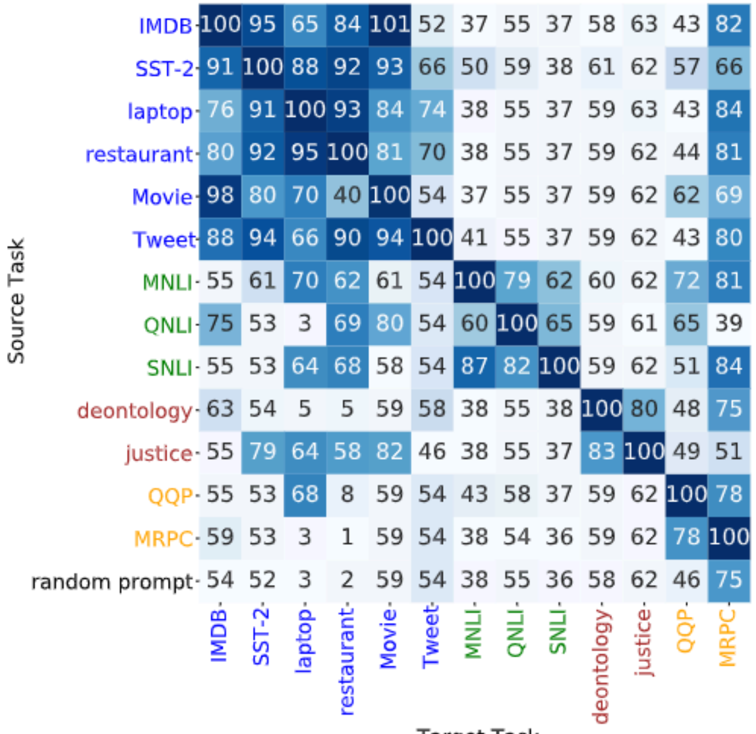


On Transferability of Prompt Tuning for NLP

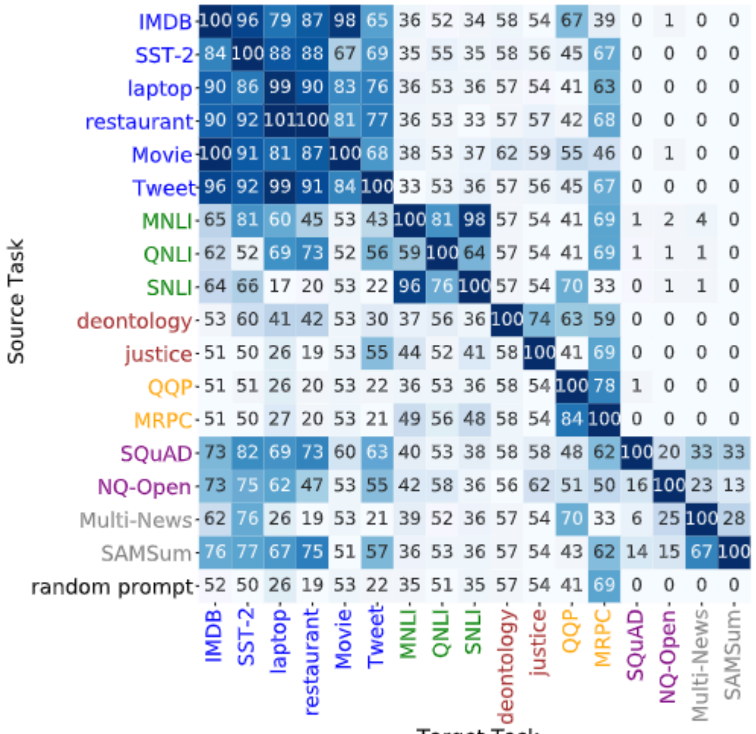
- **Prompt Tuning (PT)**
  - Solution: Transferring the trained prompts
    - **Cross-Task Transfer**
      - Motivation: Similar tasks may require similar skills



- **Cross-Task Transfer**
  - Zero-shot Transferability
    - For the tasks within the same type, transferring prompts between them can generally perform well



(a) RoBERTa$_{\text{LARGE}}$



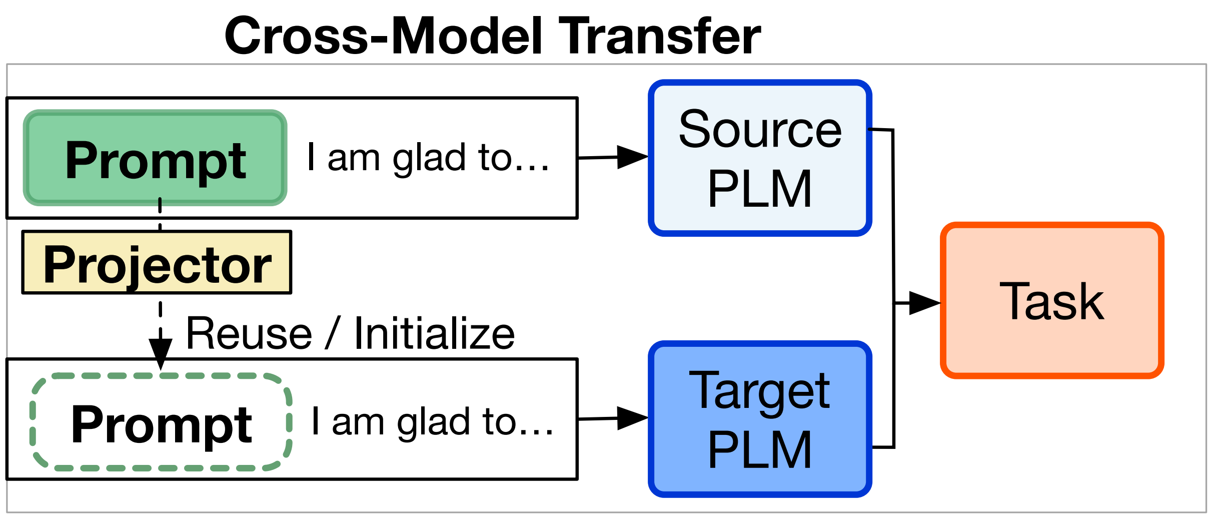
(b) T5$_{\text{XXL}}$

(Relative Performance)

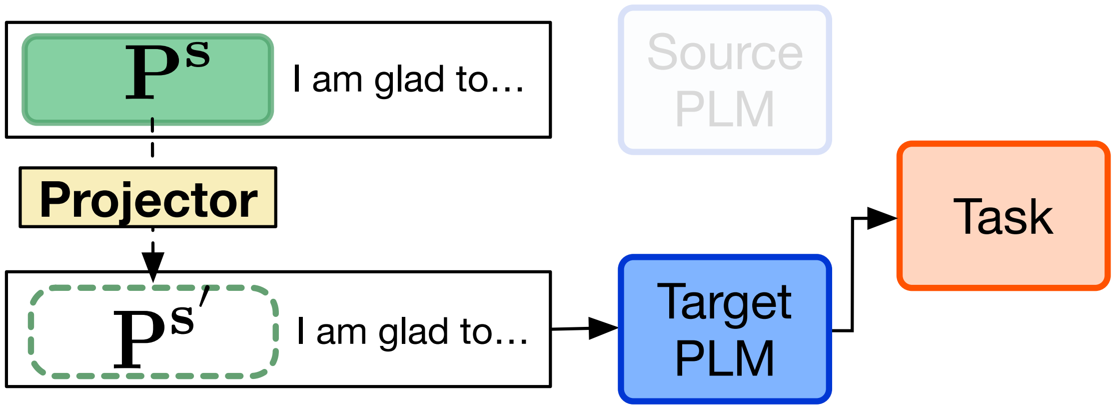
- **Cross-Task Transfer**
  - Transfer with Initialization (TPT_{TASK})
    - Initializing soft prompts with well-trained prompts of the most similar task and then starts PT can speed up training and achieve better performance

Task Type	SA						NLI			EJ		PI		QA		SUM	
Task	IMDB	SST-2	laptop	restaurant	Movie	Tweet	MNLI	QNLI	SNLI	deontology	justice	QQP	MRPC	SQuAD	NQ-Open	Multi-News	SAMSum
Metric	Acc.	Acc.	Acc.	Acc.	Acc.	Acc.	Acc.	Acc.	Acc.	Acc.	Acc.	Acc.	Acc.	F1	F1	ROUGE-L	ROUGE-L
RoBERTa$_{LARGE}$																	
Performance (PT) (%)	92.2	96.1	76.4	83.7	84.9	**76.1**	87.3	92.4	**91.9**	**85.6**	**81.0**	**88.9**	**81.2**	N/A	N/A	N/A	N/A
Performance (TPT_{TASK}) (%)	**92.4**	**96.3**	**79.1**	**85.8**	**85.1**	**76.1**	**87.9**	**93.1**	**91.9**	**85.6**	78.2	86.1	79.2	N/A	N/A	N/A	N/A
Convergence Speedup	1.7	1.1	1.0	1.9	1.2	0.9	1.2	1.2	1.3	0.9	0.7	0.8	0.9	N/A	N/A	N/A	N/A
Comparable-result Speedup	2.5	2.4	1.0	3.8	1.5	1.3	1.1	2.3	1.0	0.9	N/A	N/A	N/A	N/A	N/A	N/A	N/A
T5$_{XXL}$																	
Performance (PT) (%)	96.5	97.4	76.6	**90.1**	**97.9**	76.2	90.5	95.2	93.4	87.0	**92.5**	90.0	86.3	**86.3**	20.8	29.2	45.8
Performance (TPT_{TASK}) (%)	**96.6**	**97.8**	**84.2**	88.6	97.5	**77.0**	**92.0**	**96.2**	**94.0**	**95.3**	90.7	**90.9**	**89.0**	85.9	**21.3**	**29.3**	**46.8**
Convergence Speedup	1.2	49.7	2.2	1.1	3.9	1.4	12.5	24.9	49.9	29.8	1.5	1.0	3.3	1.1	1.0	2.0	2.0
Comparable-result Speedup	1.2	48.9	219.8	N/A	N/A	1.5	12.5	29.9	49.9	29.9	N/A	1.0	5.0	N/A	1.0	2.0	2.5

- **Prompt Tuning (PT)**
  - Solution: Transferring the trained prompts
    - Cross-Task Transfer
    - **Cross-Model Transfer**
      - Motivation: Train prompts on a small and computationally efficient PLM and use them on a massive and computationally expensive PLM



- **Cross-Model Transfer**
  - Cross-Model Prompt Projection
    - Distance Minimizing
    - Task Tuning



Source prompt: $\mathbf{P}^{\mathrm{s}}$; Target prompt: $\mathbf{P}^{\mathrm{t}}$;

$\mathbf{P}^{\mathrm{s}'} = Proj(\mathbf{P}^{\mathrm{s}})$;

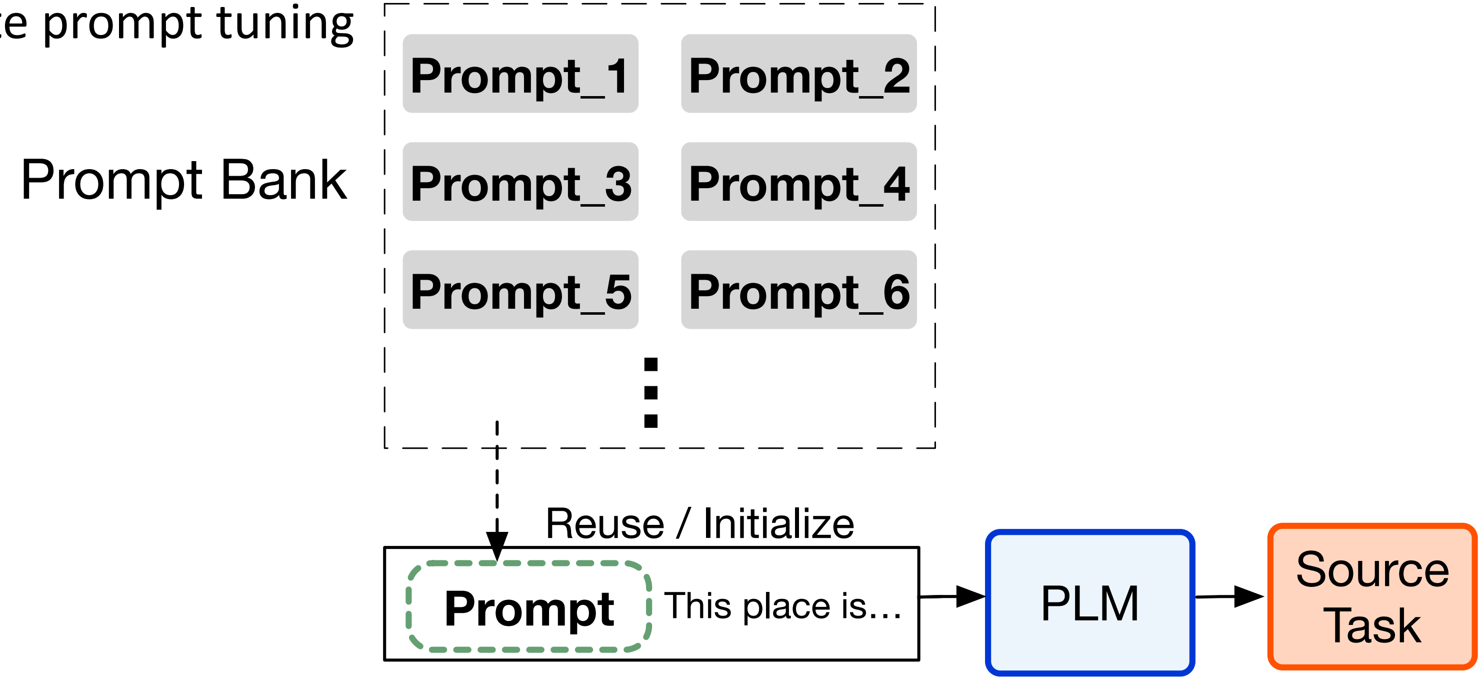
Distance Minimizing: $L_D = \min||Proj(\mathbf{P}^{\mathrm{s}}) - \mathbf{P}^{\mathrm{t}}||_2$

Task Tuning: $L_T = p(y|\mathbf{P}^{\mathrm{s}'}, x_1, \ldots, x_n)$

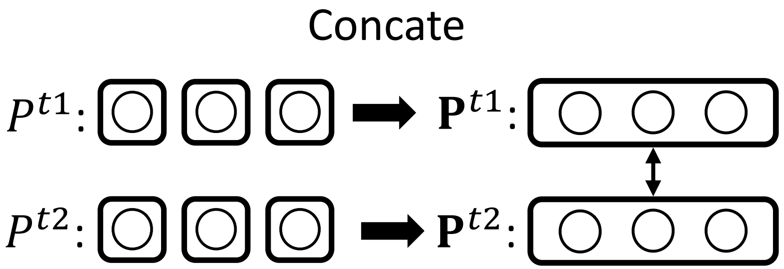
- **Cross-Model Transfer**
  - Zero-shot Transfer Performance
    - **Task Tuning** (projector) generalizes to same-type unseen tasks of the training tasks
  - Transfer with trained prompt Initialization (TPT_{TASK})
    - Accelerate convergence, improve performance

Method		SA						NLI			EJ		PI	
		IMDB	SST-2	laptop	restaurant	Movie	Tweet	MNLI	QNLI	SNLI	deontology	justice	QQP	MRPC
PT on $T5_{XXL}$		96.5	97.4	76.6	88.1	97.9	72.5	90.5	95.2	93.4	87.0	92.5	90.0	86.3
Random Prompt		49.7	49.0	19.8	17.0	51.6	15.5	31.8	49.3	31.9	51.3	50.0	36.4	67.0
(a) Zero-shot Transfer Performance (%)														
laptop	Prompt Mapping	49.6	49.0	76.6	17.5	51.5	14.4	31.8	48.1	32.8	53.3	49.9	36.8	66.6
	Task Tuning	**82.9**	**89.3**	**80.3**	**85.7**	**78.6**	**58.4**	32.4	50.7	33.6	54.9	51.6	33.9	63.7
MNLI	Prompt Mapping	49.6	50.1	19.8	18.3	51.2	15.0	**90.5**	49.0	32.9	50.3	49.0	36.8	65.6
	Task Tuning	49.7	48.8	19.8	17.0	51.6	16.0	89.8	**82.7**	**88.2**	49.7	50.0	36.8	67.7
(b) Transfer with Initialization (TPT_{MODEL})														
laptop	Performance (%)	96.5	97.4	82.9	90.3	97.4	74.4	91.0	95.4	93.4	92.5	92.5	90.0	87.9
	Convergence Speedup	1.1	1.7	1.9	1.3	0.6	1.3	0.9	0.9	1.0	1.0	0.7	1.1	1.1
	Comparable-result Speedup	1.0	19.0	16.0	6.0	N/A	2.2	3.6	1.1	6.0	6.0	0.9	1.8	3.4
MNLI	Performance (%)	96.5	97.4	82.7	88.5	95.8	74.7	91.2	95.9	93.5	94.6	92.5	90.0	87.7
	Convergence Speedup	1.0	1.6	1.8	0.9	0.4	1.3	1.0	1.1	1.4	2.0	1.7	0.9	0.9
	Comparable-result Speedup	1.0	18.0	15.0	1.6	N/A	1.5	18.0	20.0	30.0	7.5	5.0	1.5	1.9

- **Exploring Transferability Indicator**
  - Motivation
    - Explore why the soft prompts can transfer across tasks and what decides the transferability between them
    - Find suitable prompts for performing transfer to achieve better performance or accelerate prompt tuning



- **Exploring Transferability Indicator**
  - Embedding Similarity
    - Euclidean similarity
    - Cosine similarity
  - Model Stimulation Similarity

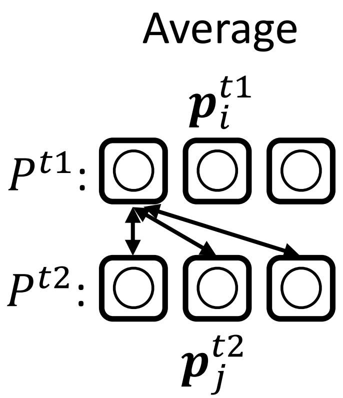


$$\mathrm{E}_{\mathrm{concat}}(P^{t_1}, P^{t_2}) = \frac{1}{1 + \|\mathbf{P}^{t_1} - \mathbf{P}^{t_2}\|}$$

$$\mathrm{C}_{\mathrm{concat}}(P^{t_1}, P^{t_2}) = \frac{\mathbf{P}^{t_1} \cdot \mathbf{P}^{t_2}}{\|\mathbf{P}^{t_1}\|\|\mathbf{P}^{t_2}\|}$$

[1] Geva, et al., Transformer Feed-Forward Layers Are Key-Value Memories, EMNLP, 2021.

- **Exploring Transferability Indicator**
  - Embedding Similarity
    - Euclidean similarity
    - Cosine similarity
  - Model Stimulation Similarity

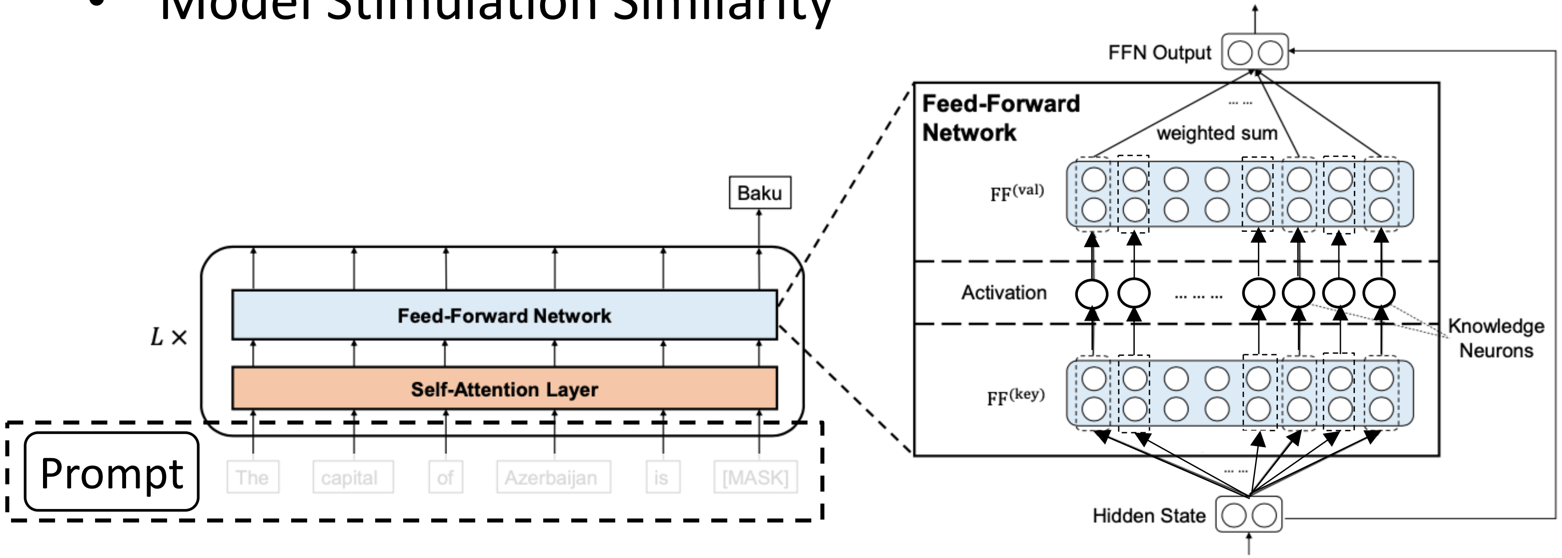


$$\mathrm{E}_{\mathrm{average}}(P^{t_1}, P^{t_2}) = \frac{1}{1 + \frac{1}{l^2}\sum_{i=1}^{l}\sum_{j=1}^{l}\|\mathbf{p}_i^{t_1} - \mathbf{p}_j^{t_2}\|}$$

$$\mathrm{C}_{\mathrm{average}}(P^{t_1}, P^{t_2}) = \frac{1}{l^2}\sum_{i=1}^{l}\sum_{j=1}^{l}\frac{\mathbf{p}_i^{t_1} \cdot \mathbf{p}_j^{t_2}}{\|\mathbf{p}_i^{t_1}\|\|\mathbf{p}_j^{t_2}\|}$$

[1] Geva, et al., Transformer Feed-Forward Layers Are Key-Value Memories, EMNLP, 2021.

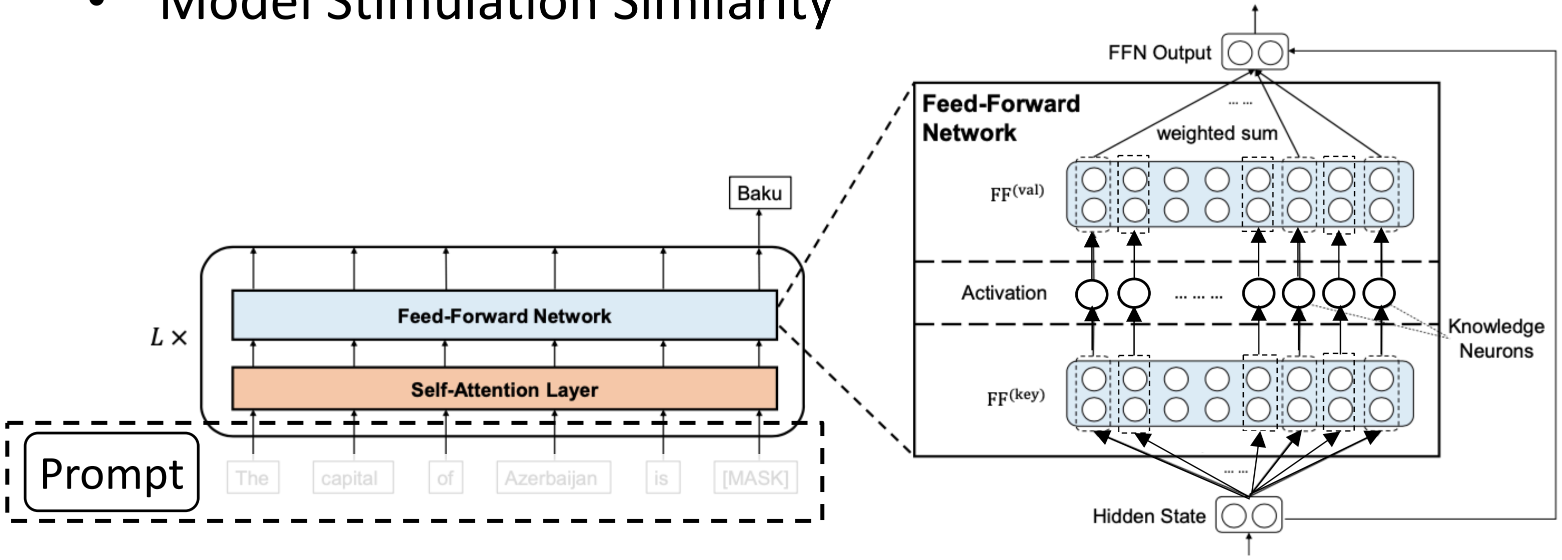
- **Exploring Transferability Indicator**
  - Embedding Similarity
    - Euclidean similarity
    - Cosine similarity
  - Model Stimulation Similarity



$$\mathrm{FFN}(\mathbf{x}) = \boxed{\max(\mathbf{x}W_1^\top + \mathbf{b_1}, \mathbf{0})}W_2 + \mathbf{b_2},$$

[1] Geva, et al., Transformer Feed-Forward Layers Are Key-Value Memories, EMNLP, 2021.

- **Exploring Transferability Indicator**
  - Embedding Similarity
    - Euclidean similarity
    - Cosine similarity
  - Model Stimulation Similarity

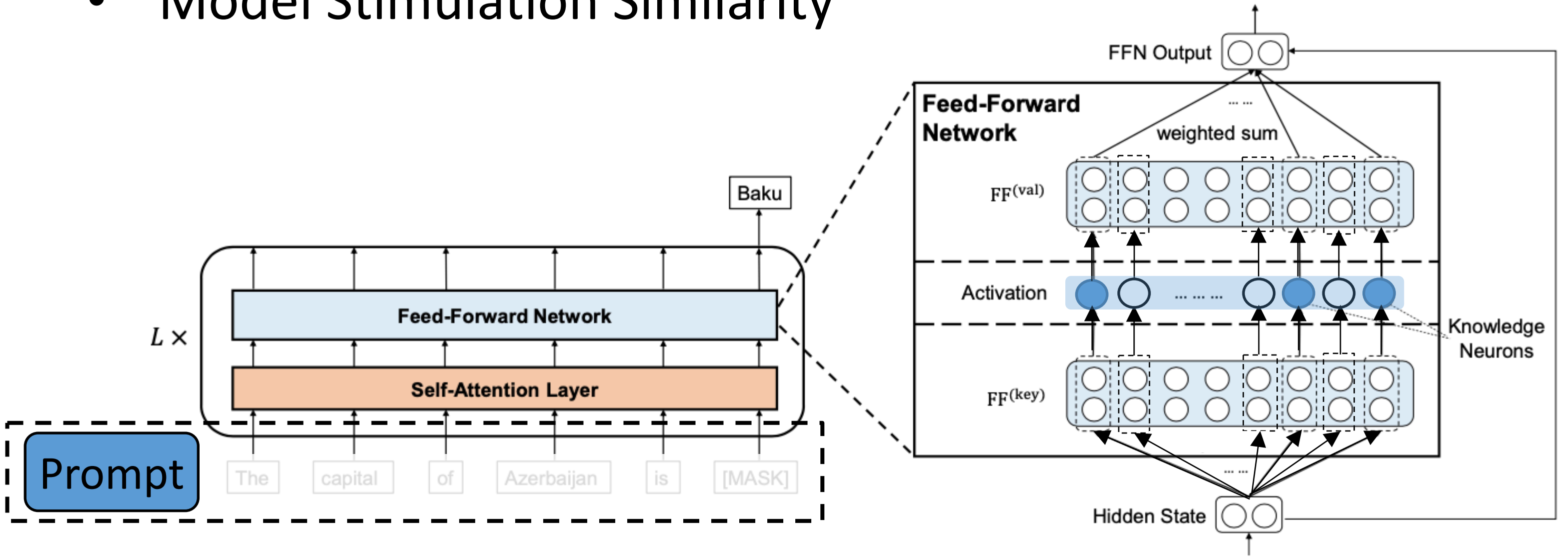


$$\mathrm{FFN}(\mathbf{x}) = \max(\mathbf{x}W_1^\top + \mathbf{b_1}, \mathbf{0})W_2 + \mathbf{b_2},$$

$$\mathrm{AS}(P) = [\mathbf{s}_1; \mathbf{s}_2; ...; \mathbf{s}_L]$$

[1] Geva, et al., Transformer Feed-Forward Layers Are Key-Value Memories, EMNLP, 2021.

- **Exploring Transferability Indicator**
  - Embedding Similarity
    - Euclidean similarity
    - Cosine similarity
  - Model Stimulation Similarity

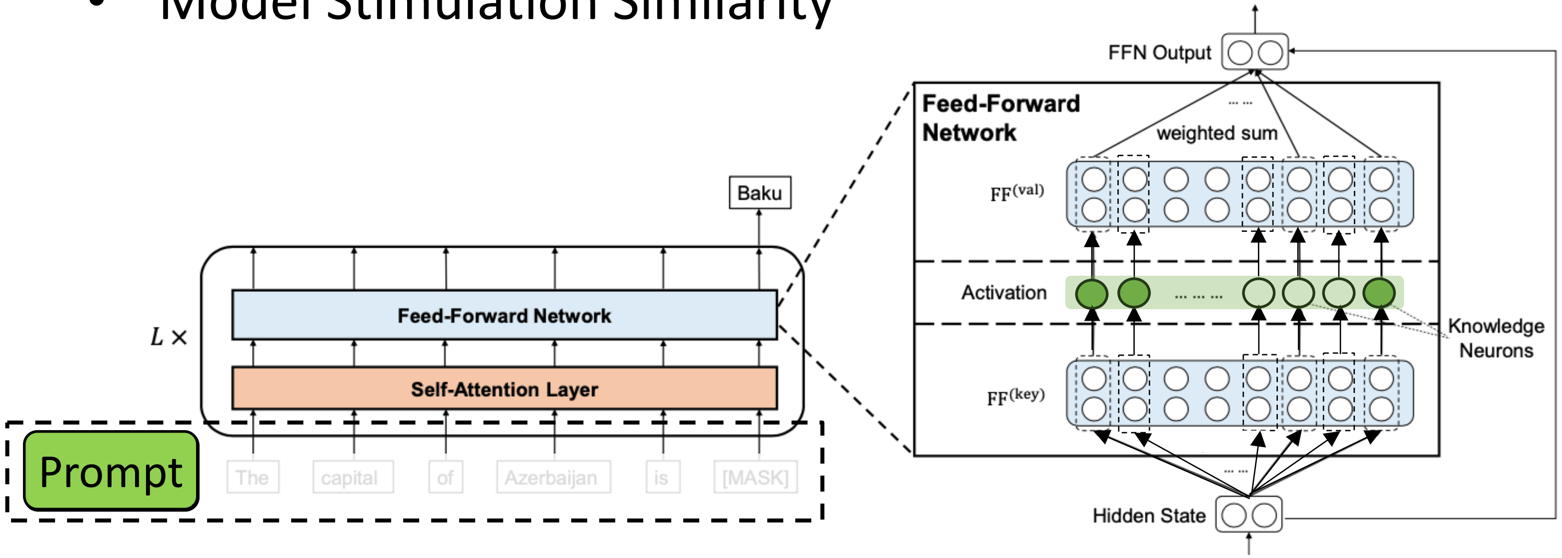


$$\mathrm{FFN}(\mathbf{x}) = \boxed{\max(\mathbf{x}W_1^\top + \mathbf{b_1}, \mathbf{0})}W_2 + \mathbf{b_2},$$

$$\mathrm{AS}(P) = [\mathbf{s}_1; \mathbf{s}_2; \ldots; \mathbf{s}_L]$$

[1] Geva, et al., Transformer Feed-Forward Layers Are Key-Value Memories, EMNLP, 2021.

- **Exploring Transferability Indicator**
  - Embedding Similarity
    - Euclidean similarity
    - Cosine similarity
  - Model Stimulation Similarity

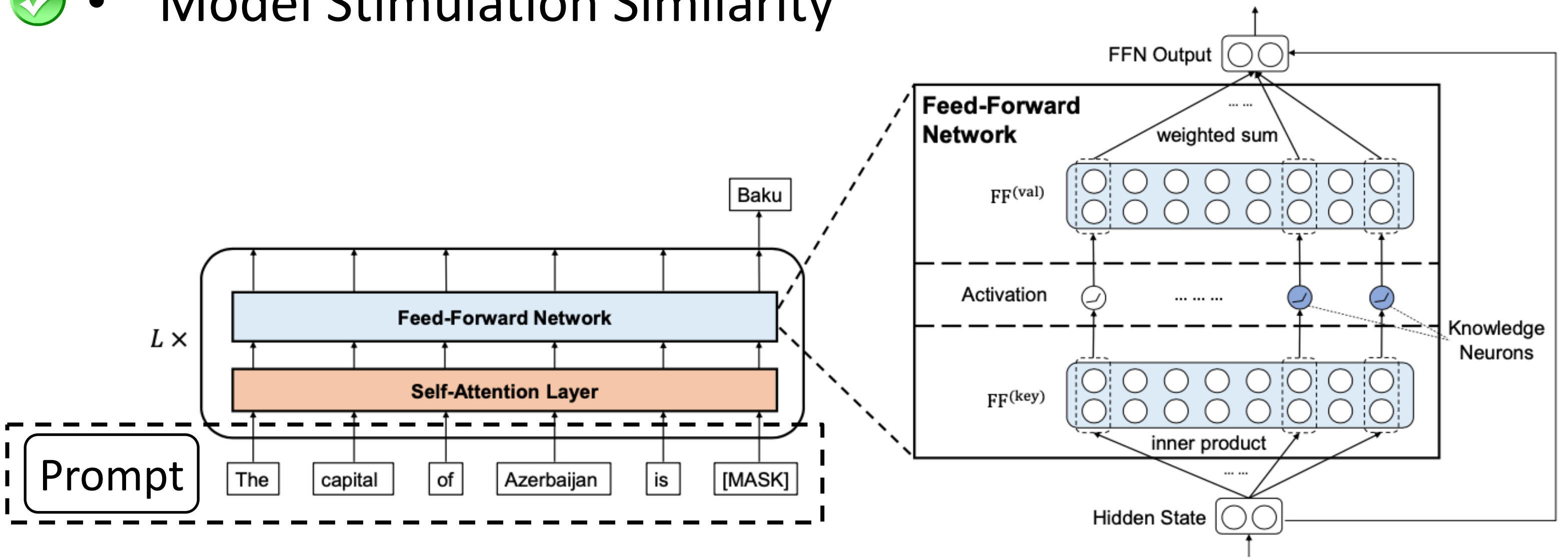


$$\mathrm{FFN}(\mathbf{x}) = \boxed{\max(\mathbf{x}W_1^\top + \mathbf{b_1}, \mathbf{0})} W_2 + \mathbf{b_2},$$

$$\mathrm{AS}(P) = [\mathbf{s}_1; \mathbf{s}_2; \ldots; \mathbf{s}_L]$$

[1] Geva, et al., Transformer Feed-Forward Layers Are Key-Value Memories, EMNLP, 2021.

- **Exploring Transferability Indicator**
  - Embedding Similarity
    - Euclidean similarity:
    - Cosine similarity:
  - Model Stimulation Similarity



$$\mathrm{FFN}(\mathbf{x}) = \boxed{\max(\mathbf{x}W_1^\top + \mathbf{b_1}, \mathbf{0})} W_2 + \mathbf{b_2},$$

$$\mathrm{AS}(P) = [\mathbf{s}_1; \mathbf{s}_2; \ldots; \mathbf{s}_L]$$

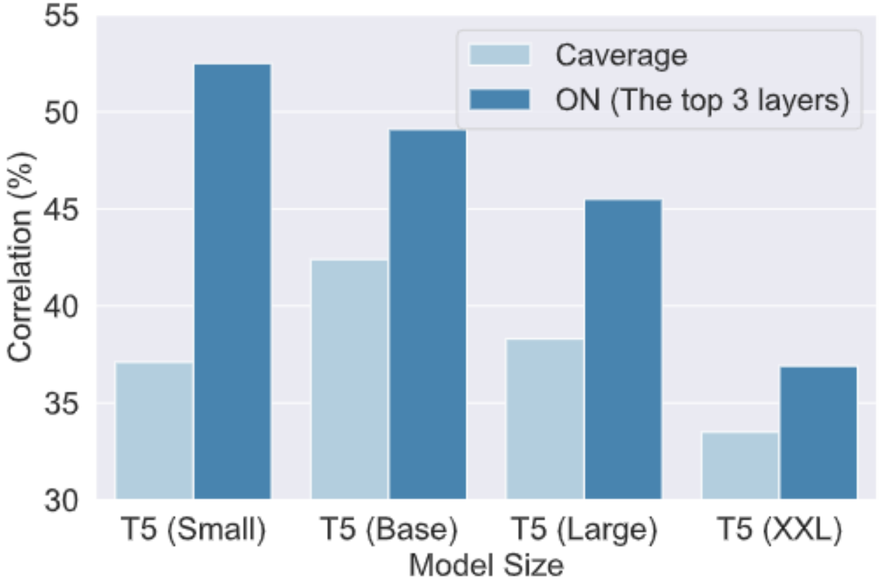
$$\mathrm{ON}(P^{t_1}, P^{t_2}) = \frac{\mathrm{AS}(P^{t_1}) \cdot \mathrm{AS}(P^{t_2})}{\|\mathrm{AS}(P^{t_1})\| \|\mathrm{AS}(P^{t_2})\|}$$

[1] Geva, et al., Transformer Feed-Forward Layers Are Key-Value Memories, EMNLP, 2021.

- **Exploring Transferability Indicator**
  - Model Stimulation Similarity (ON)
    - ON has the higher Spearman's correlation with the transferability
    - ON works worse on the larger PLMs because of the higher redundancy [1]

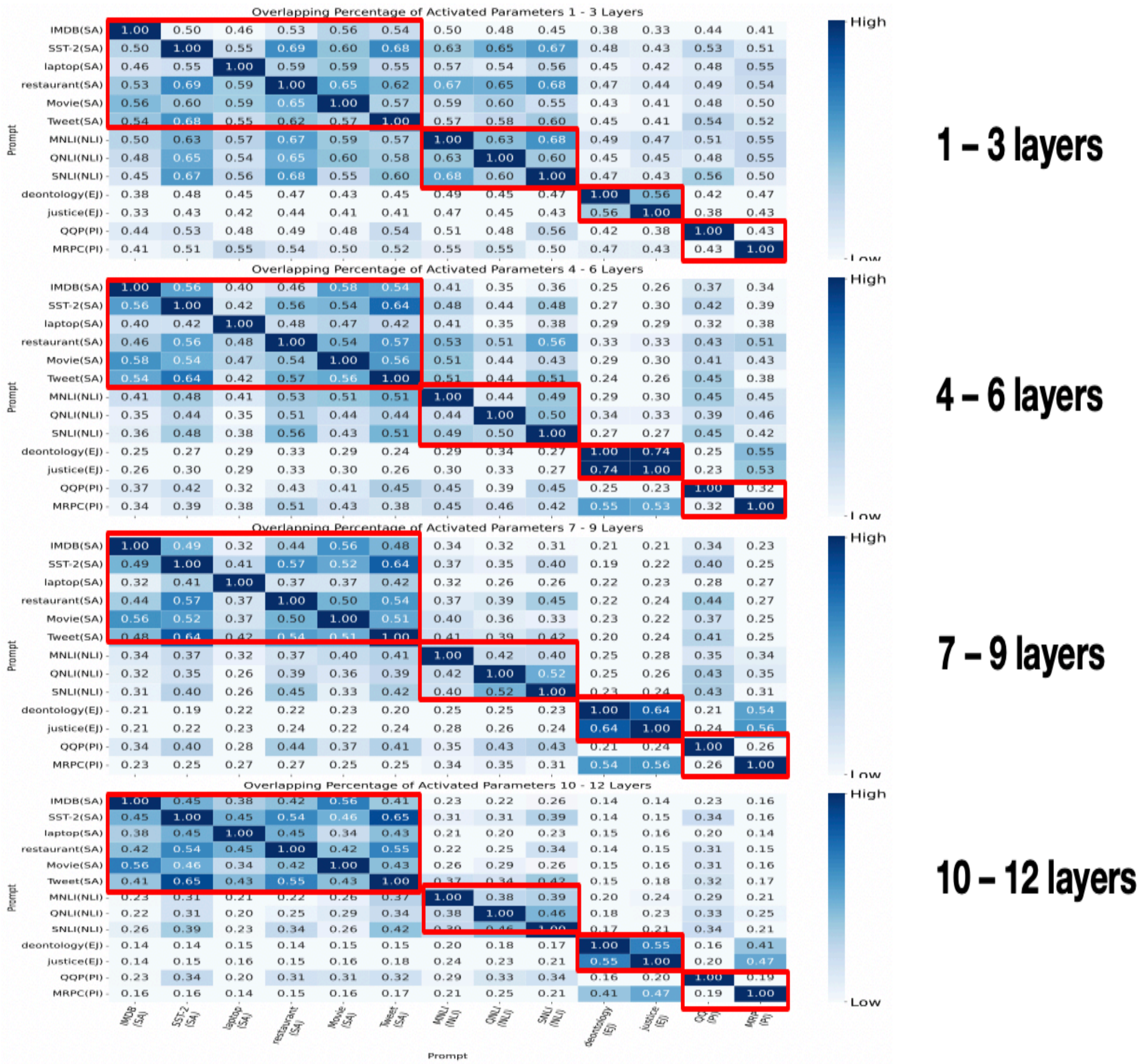
Metric	Model	
	$\text{RoBERTa}_{\text{LARGE}}$	T5_{XXL}
E_{concat}	22.6	12.9
$E_{average}$	2.8	-2.5
C_{concat}	24.8	31.6
$C_{average}$	44.7	33.5
ON	**49.7**	**36.9**

Spearman's correlation



[1] Aghajanyan, et al., Intrinsic Dimensionality Explains the Effectiveness of Language Model Fine-Tuning, ACL, 2021.

- **Exploring Transferability Indicator**
  - Distribution of Activated Neuron
    - The activated neurons are comm-on in the bottom layers but more task-specific in top layers



Conclusion

- **Transferability of Prompts**
  - In the cross-task setting
    - Soft prompts can transfer to similar tasks to accelerate prompt tuning and achieve better performance
  - In the cross-model setting
    - Soft prompts can transfer to different PLMs with a projector trained on similar tasks
  - Transferability Indicator
    - We can utilize activated neurons of soft prompts to well indicate transferability between tasks

Q&A

THUNLP

[paper]

[code]