Multi-mask Prefix Tuning: Applying Multiple Adaptive Masks on Deep Prompt Tuning

Qui Tu, Trung Nguyen, Long Nguyen, Dien Dinh

Proceedings of the 38th Pacific Asia Conference on
Language, Information and Computation (PACLIC 38)

Nathaniel Oco, Shirley N. Dita, Ariane Macalinga Borlongan, Jong-Bok Kim (eds.)

2024

Multi-mask Prefix Tuning: Applying Multiple Adaptive Masks on Deep Prompt Tuning

Qui Tu[1,2*], **Trung Nguyen**[1,2*], **Long Nguyen**[1,2†], **Dien Dinh**[1,2]
[1]Faculty of Information Technology, University of Science, Ho Chi Minh City, Vietnam
[2]Vietnam National University, Ho Chi Minh City, Vietnam
{vanqui.tu,audreytrungnguyen}@gmail.com, {nhblong,ddien}@fit.hcmus.edu.vn

Abstract

Prompt tuning is a notable Parameter-efficient Fine-tuning approach that allows users to fine-tune a pre-trained language model for a specific task with significantly lower computational resources compared to traditional full fine-tuning. However, it still faces challenges related to convergence and stability, particularly concerning the sensitivity to the length of the prompts used. In this work, we propose a novel prompt tuning method **Multi-mask Prefix Tuning**[1] that can derive multiple versions of prompt adapted to each instance of the data. To do this, we utilize a routing mechanism and multiple tunable adaptive masks which then are applied on a trainable task-specific soft prompt. Our method practically shows improvements in training time and performance across Natural Language Understanding (NLU) tasks compared to other prompt tuning baselines, narrows down the gap to LoRA and full fine-tuning while not requiring any modifications to model structure and pre-trained weights.

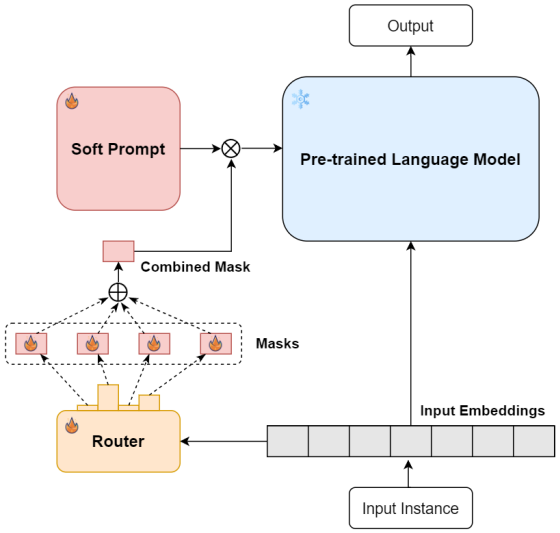

Figure 1: An illustration of the proposed approach Multi-mask Prefix Tuning. A gating mechanism is utilized to route each input instance to a specific combination of masks, which then is applied on the shared trainable soft prompt.

1 Introduction

In recent years, pre-trained language models have achieved significant performance in the field of natural language processing. Since the pre-trained language models can be fine-tuned to quickly adapt to downstream tasks, this *pretrain-then-finetune* paradigm has been a common approach for researchers in the field. However, the rapidly increasing size of pre-trained language models also places great pressure on the computational infrastructure required to fully fine-tune and store them. A particularly interesting research direction in the current context is the development of Parameter-Efficient Fine-Tuning (PEFT) methods (He et al., 2022), which require tuning only a significantly smaller set of parameters.

Among PEFT lines of research, prompt tuning is a worth-noticing one. At first, prompt tuning methods solely tune the soft prompt (Lester et al., 2021; Liu et al., 2023), which is trainable token embeddings, prepended to the model input. Subsequent studies referred to as Deep Prompt Tuning (Li and Liang, 2021; Liu et al., 2022b) continually improve the design of soft prompts by adding length-equivalent soft prompt tokens to each layer of the models, achieving performance comparable to other PEFT methods and even full fine-tuning with only 3% of the parameters tuned. In practice, there have been challenges that prompt tuning methods still face regarding the convergence rate, stability as well as sensitivity to hyper-parameters such as *prompt length* (Han et al., 2024).

*Both authors contributed equally to this research.
†Corresponding author.
[1]Code availability: https://github.com/vanqui-tu/Multi-mask-Prefix-Tuning

There has been active research in the field aimed at improving the effectiveness and efficiency of prompt tuning. On the one hand, improvement in the performance of prompt tuning can be achieved by modifying the soft prompt design such as incorporating input-specific soft prompts (Jiang et al., 2022; Wu et al., 2022; Liu et al., 2022a), controlling each prompt token importance (Zhang et al., 2023) or extending the influence of the prompt to model weights (Wang et al., 2023a). On the other hand, other works have been proposed to enhance efficiency by decomposing the soft prompt (Shi and Lipani, 2024; Xiao et al., 2023) or reducing the actual length of used prompt by leveraging a sparse activation mechanism (Choi et al., 2023). After all, there is still a need to address the existing limitations of prompt tuning.

Carefully inspected, we found that Deep Prompt Tuning is the base architecture with better performance and also better practical efficiency, in comparison with typical Prompt Tuning architecture. Besides, we adopt the initiatives of adaptive soft prompts from Adaptive Prefix Tuning (APT) (Zhang et al., 2023) and the idea of short prompts fit with subsets of training datasets from Sparse Mixture-of-Prompts (SMoP) (Choi et al., 2023). As far as we are concerned, some limitations are coming along with the design of SMoP regarding the overfitting and unbalanced activation of prompts. By adopting the advantages and addressing the existing disadvantages of those previous works, we aim to develop a novel prompt tuning method with practically improved performance and efficiency.

To this end, we propose **Multi-mask Prefix Tuning**, a novel prompt tuning method utilizing multiple trainable adaptive masks controlling the influence of each prompt token and a sparse activation mechanism to guide each input instance to a different combination of masks, which then is used to extract an instance-specific version of soft prompt from the common tunable part. Our method provides a flexible prompt tuning design allowing effective training and instance-specific prompts while maintaining a common soft prompt to share useful task-specific knowledge between versions of each extracted soft prompt.

As in previous works, our experiments are conducted on six Natural Language Understanding tasks from the SuperGLUE benchmark (Wang et al., 2019) to evaluate the method's performance in practice. Experimental results depict that our proposed method shows an improvement in average accuracy on the six SuperGLUE tasks with T5-base (Raffel et al., 2023) although requires under one-half of training time and one-third of training memory in comparison with other prompt tuning baselines.

Our contributions are as follows:

- We propose a novel prompt tuning method named **Multi-mask Prefix Tuning** that utilizes a set of adaptive masks and a sparse activation mechanism.
- Our method shows a flexible design that can provide prompts that fit each instance whilst sharing valuable task-specific knowledge.
- Experimental results demonstrate that our proposed method, with significantly lower training costs, surpasses the baseline methods on T5-base.

2 Related Works

Since fully fine-tuning pre-trained language models is more and more expensive due to their increases in size, Paramter-Efficient Fine-Tuning (PEFT) methods became a lightweight alternative that requires tuning only a small portion of task-specific parameters while keeping most pre-trained parameters frozen. Adapter tuning (Houlsby et al., 2019) is a popular approach of PEFT, which involves inserting small neural modules named adapters into each pre-trained Transformer layer and then optimizing only those adapters at fine-tuning time. In another approach, LoRA (Hu et al., 2021) injects trainable low-rank matrices into Transformer layers to approximate the weight updates, becoming the most widely recognized PEFT technique.

Prompt Tuning is another simple yet effective PEFT approach, that even requires minimal modification to be applied on pre-trained language models. Concurrent works P-tuning (Liu et al., 2023) and Prompt Tuning (Lester et al., 2021) started the line of research by applying learnable soft prompt tokens at the initial word embedding layer. Later works introduced Deep Prompt Tuning design through Prefix Tuning (Li and Liang, 2021) and P-tuning v2 (Liu et al., 2022b), which is claimed to achieve comparable performance to full fine-tuning in some particular tasks, with only 0.1%-3% tuned parameters. Later advancements aimed to enhance prompt tuning performance and efficiency by modifying soft prompt design (Wang et al., 2023a; Zhu and Tan, 2023), leveraging instance-specific

prompts (Jiang et al., 2022; Wu et al., 2022; Liu et al., 2022a), adopting transfer learning (Vu et al., 2022; Asai et al., 2022; Wang et al., 2023b) or reparameterizing the soft prompt part (Shi and Lipani, 2024; Xiao et al., 2023).

Among advanced prompt tuning studies, we notice XPrompt (Ma et al., 2022) proved the existence of trained prompt tokens posing negative impacts on the performance of the model on a downstream task. This finding raised a need for controlling the importance of each soft prompt token, which then was implemented in the research Adaptive Prompt Tuning (Zhang et al., 2023). Another prompt tuning research SMoP (Choi et al., 2023) adopted the idea of instance-aware prompts and proposed a novel method that utilizes a routing mechanism and multiple short soft prompts. The idea was inspired by the Mixture-of-Experts architecture (Shazeer et al., 2017) and can be found in another PEFT method AdaMix (Wang et al., 2022).

3 Method

3.1 Preliminaries

Deep Prompt Tuning As a variation of Prompt Tuning, Deep Prompt Tuning (Li and Liang, 2021; Liu et al., 2022b) is also applied on Transformer-based pre-trained models. A typical Transformer block (Vaswani et al., 2023) consists of multi-head attention, which is multiple parallel self-attention functions, and a fully connected feed-forward network. The calculations within a Transformer block can be simplified as follows:

$$\text{Attn}(\boldsymbol{Q}, \boldsymbol{K}, \boldsymbol{V}) = \text{softmax}(\frac{\boldsymbol{Q}\boldsymbol{K}^T}{\sqrt{d}}\boldsymbol{V}) \quad (1)$$

$$\text{FFN}(\boldsymbol{x}) = \text{ReLU}(\boldsymbol{x}\boldsymbol{W}_1 + \boldsymbol{b}_1)\boldsymbol{W}_2 + \boldsymbol{b}_2 \quad (2)$$

Deep Prompt Tuning inserts soft prompt tokens of length l into each layer of the pre-trained language model. This was done by representing the soft prompt as 2 separate key-value parts and concatenating them to the corresponding key-value matrix at each layer. In particular, let $\boldsymbol{P}_k$ and $\boldsymbol{P}_v$ represent the keys and values of the soft prompt, respectively, where $\boldsymbol{P}_k, \boldsymbol{P}_v \in \mathbb{R}^{l \times d}$. Here, l indicates the length of the prefix, and d refers to the dimension. Consequently, the self-attention function can be restructured as follows:

$$\text{Attn}(\boldsymbol{Q}, \boldsymbol{K'}, \boldsymbol{V'}) = \text{softmax}(\frac{\boldsymbol{Q}(\boldsymbol{K'})^T}{\sqrt{d}}\boldsymbol{V'}) \quad (3)$$

$$\text{where } \boldsymbol{K'} = [\boldsymbol{P}_k; \boldsymbol{K}], \boldsymbol{V'} = [\boldsymbol{P}_v; \boldsymbol{V}]$$

Since this method was initially introduced through Prefix Tuning (Li and Liang, 2021), it can also be referred to as Prefix Tuning in the context of prompt tuning.

3.2 Multi-mask Prefix Tuning

The objective of our proposed method, **Multi-mask Prefix Tuning**, is to extract versions of prompt that are suitable for each input instance, from the common trainable soft prompt. To achieve this, we leverage multiple tunable adaptive masks that manage the significance of each soft prompt token, along with a gating mechanism to direct each input instance to a specific combination of masks. For more details, an overview of our proposed method is presented in Figure 1.

Our method involves three main trainable components: a soft prompt, a routing component (router) and a set of masks of the same size. To correctly route an input sequence to the appropriate combination of masks, the routing component needs to understand the semantics of each input. Consider an input sequence of length l: $X = \{x_1, x_2, ..., x_p\}$, where $x_i \in \mathbb{R}^d$ are the token embeddings. We take the average of all token embeddings as the semantic representation for each input sequence: $\bar{X} = \text{mean}(x_1, x_2, ..., x_p)$.

Assuming we use k different masks, the parameterization of the set of masks is $\{\theta_j\}_{j=1}^k$, where $\theta_j \in \mathbb{R}^{l \times m}$, with l and m being the length of the prompt and the number of layers of the model, respectively. The routing component is a Linear layer with parameters μ, denoted as L_μ. The probability that an input instance $\bar{X}$ is routed to the j-th mask is as follows:

$$p_j(X) = [\,\text{softmax}(\,L_\mu(\,\bar{X}\,)\,)\,]_j \quad (4)$$

The distribution obtained through the above operation serves as the basis for determining the appropriate mask. While the authors of SMoP (Choi et al., 2023) select the mask with the highest probability for the next step, we take a different approach by summing the weighted masks according to the distribution from the previous step, resulting in a final combined mask $\bar{\theta}$:

$$\bar{\theta} = \sum_{j=1}^{k} p_j(\bar{X}) \cdot \theta_j \quad (5)$$

The use of a combination of the tuned masks facilitates gradient computation and optimization of

all the components. The instance-specific prompt $\tilde{P}$ is generated by applying the mask $\bar{\theta}$ through the sigmoid function on the shared soft prompt $P \in \mathbb{R}^{l\times(m\times d)}$. By preserving a shared common soft prompt, valuable task-relevant knowledge is able to be shared across different versions of extracted prompts, enhancing generalization compared to SMoP. To ensure that $\bar{\theta}$ having dimensions (l, m) can be applied to the prompt P with dimensions $(l \times (m \times d))$, we denote $\bar{\theta}^{ext} \in \mathbb{R}^{l\times(m\times d)}$ as the extension of $\bar{\theta}$, where each element of $\bar{\theta}$ corresponds to d elements of P:

$$\tilde{P} = \text{sigmoid}(\ \bar{\theta}^{ext}\ ) \odot P \tag{6}$$

In this context, $\odot$ denotes the Hadamard product, or element-wise multiplication. Next, the objective function of the method can be expressed as follows:

$$\underset{\mu,\theta,P}{\text{argmax}} \log p(\ Y \mid \tilde{P}, X\ ) \tag{7}$$

4 Experiments

4.1 Experimental Settings

Dataset We perform evaluations across a range of tasks from the SuperGLUE benchmark (Wang et al., 2019). Our analysis included six tasks: BoolQ (Clark et al., 2019), COPA (Roemmele et al., 2011), CB (De Marneff et al., 2019), (Roemmele et al., 2011), MultiRC (Khashabi et al., 2018), RTE (Bentivogli et al., 2009) and WiC (Pilehvar and os'e Camacho-Collados, 2018). Similar to the SMoP's experiment setting (Choi et al., 2023), due to the absence of official test datasets for these benchmarks, we adopt the approach recommended by (Chen et al., 2022), using the validation sets as stand-ins for the test sets. Additionally, we reorganize the original training dataset, splitting it into new training and validation subsets with a division ratio of 90%/10%. Detailed information about the datasets, including their sizes, metrics, and tasks, is provided in Appendix A.2.

Baselines To assess the efficacy of our approach, we conduct comparative analyses between our Multi-mask Prefix Tuning method and notable several existing methods, including Prompt Tuning, P-tuning, SMoP, Prefix Tuning, LoRA and Fine-tuning. These experiments utilized the pre-trained T5-base model (Raffel et al., 2020) .

Hyperameters In our Multi-mask Prefix Tuning approach, we explore settings incorporating [5, 10, 20] prompt tokens paired with [1, 2, 4] masks. We employ distinct learning rates for different components of our model. Specifically, the mask and router are optimized with learning rates of [0.05, 0.001] and [0.001, 0.0005], respectively. We train our model for 15 epochs on tasks with more than 8000 samples such as BoolQ and MultiRC, and 50 epochs for other tasks. In line with SMoP's practices, we employ the Adafactor optimizer (Shazeer and Stern, 2018), setting a weight decay of 1e-5 and implementing a linear learning rate decay with a warm-up ratio of 0.06. In addition, we also apply a drop-out rate of 0.2 on the routing component during the training process and add a small L1 regularization term to promote the sparsity of the masks.

4.2 Results

4.2.1 Main result

Table 1 represents the performance from best setting of Multi-mask Prefix Tuning and other methods. Our method achieves the highest accuracy score among listed methods on six SuperGLUE tasks. It improves by 1.23% on average score compared to vanilla Prefix Tuning method, and by 0.58% compared to the second-best method, SMoP. Our method demonstrates significant improvements across various tasks, particularly in small datasets, compared to the vanilla Prefix Tuning approach. Specifically, in the COPA dataset (5.3%), our method outperforms SMoP, while maintaining comparable results in relatively large dataset to SMoP. Besides, the corresponding standard deviation for our method is the lowest at 0.7. Compared to others, this indicates that our method has the greatest stability and reliability.

4.2.2 Prompt length and number of masks

We train Multi-mask Prefix Tuning for the T5-base model with different prompt lengths in [5, 10, 20] and number of masks in [1, 2, 4]. The results are reported in Table 2.

Each task has a different optimal setting, and it's challenging to predict these settings due to the unique characteristics and difficulty levels. We observe performance degradation when using multiple masks with a prompt length of 20. This observation is consistent with SMoP. We believe that the performance degradation may be due to the limited labeled data available for training in several SuperGLUE tasks, leading to insufficient training of each mask.

Method	Trainable Params(%)	BoolQ	CB	COPA	Multi	RTE	WiC	Average
Fine-tuning*	100	$81.9_{0.1}$	$96.4_{1.8}$	$64.3_{1.5}$	$80.2_{0.2}$	$79.2_{0.2}$	$67.0_{2.3}$	$78.2_{1.3}$
LoRA* (r=8)	0.3954	79.0_{0}	$90.5_{1.0}$	$60.0_{0.6}$	$\mathbf{80.0}_{0.0}$	$\mathbf{77.9}_{2.9}$	$\mathbf{66.9}_{0.8}$	$75.7_{1.3}$
P-tuning* (l=20)	0.103	$78.7_{0.2}$	$91.7_{2.7}$	$58.3_{3.8}$	$79.3_{0.2}$	$77.1_{1.8}$	$65.9_{0.7}$	$75.2_{1.1}$
Prompt Tuning* (l=100)	0.0344	$79.1_{0.1}$	$86.9_{3.7}$	$56.7_{2.1}$	$78.3_{0.2}$	$73.2_{1.7}$	$65.6_{1.2}$	$73.3_{1.9}$
SMoP* (l=5, k=4)	0.0083	$\mathbf{79.4}_{0.3}$	$94.6_{1.8}$	$58.3_{2.9}$	$79.6_{0.1}$	$77.5_{3.2}$	$65.2_{0.5}$	$75.8_{1.9}$
Prefix Tuning (l=20)	0.1651	$78.8_{0.1}$	$91.5_{0.8}$	$62.0_{3.2}$	79.2_{0}	$76.8_{0.5}$	$64.2_{0.3}$	$75.42_{0.8}$
Multi-mask (l=10, m=4)	0.0843	$78.9_{0.1}$ (+0.13%)	$\mathbf{94.62}_{0.9}$ (+3.41%)	$\mathbf{63.6}_{1.4}$ (+2.58%)	79.4_{0} (+0.25%)	$77.26_{0.8}$ (+0.60%)	$64.48_{0.5}$ (+0.44%)	$\mathbf{76.38}_{0.7}$ (+1.23%)

Table 1: Main experimental results (%) on six SuperGLUE tasks. l indicates prompt length, r for LoRa indicates the rank of matrices, k for SMoP indicates number of prompts, m for Multi-mask indicates number of masks. Best results are in bold (the larger, the better). The number next to each score indicates the performance improvement (+) compared with vanilla Prefix-Tuning. The subscript of scores indicates the corresponding standard deviation. Methods with '*' indicate the results reported in (Choi et al., 2023).

We notice that using 10 soft prompts and 4 masks yields the highest scores across tasks. It is important to note that while Multi-mask Prefix Tuning generally improves upon Prefix Tuning, the optimal prompt length and number of soft prompts may vary depending on the specific task or dataset.

4.2.3 Training costs

Table 3, 4 present peak memory (GB) and training time (s/100 steps) of our method compared to other methods. Given that Multi-mask Prefix Tuning builds on the foundation of Prefix Tuning, its training cost aligns with that of Prefix Tuning while significantly reducing these costs compared to other approaches. Our method maintains nearly the same training time and results in only a slight increase in peak memory usage (from 2.96 to 3.79 GB). Specifically, our approach achieves a 1.97 times reduction in training time and a 3.14 times decrease in memory usage compared to SMoP.

5 Conclusion

In this paper, we introduce a novel prompt tuning approach that leverages both task-specific and instance-specific learning strategies. By employing a soft prompt for task-specific adjustments and a routing mechanism to tailor masks for individual instances, Multi-mask Prefix Tuning outperforms other prompt tuning methods in accuracy while significantly reducing training costs. Overall, our work contributes an innovative idea to improve the prompt tuning method and aims to inspire future research in this area.

Limitations

Although our method can be adapted for use with both encoder-only and decoder-only models, our experiments are conducted exclusively on the encoder-decoder model, specifically using the T5-base. Extensive experiments across a broader range of models and datasets would be beneficial. The architecture of the router component, which customizes masks to fit each instance, requires further exploration to enhance efficiency while still avoiding overfitting. Additionally, determining the optimal prompt length and number of masks necessitates extensive trials for each task. We leave these considerations for future research, aiming to develop a method that performs consistently well across all variations of prompt length and number of masks, thereby increasing stability and reliability.

Acknowledgments

This work is supported by research funding from Faculty of Information Technology, University of Science, Vietnam National University - Ho Chi Minh City.

References

Akari Asai, Mohammadreza Salehi, Matthew Peters, and Hannaneh Hajishirzi. 2022. ATTEMPT: Parameter-efficient multi-task tuning via attentional mixtures of soft prompts. In *Proceedings of the*

2022 Conference on Empirical Methods in Natural Language Processing, pages 6655–6672, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Luisa Bentivogli, Peter Clark, Ido Dagan, and Danilo Giampiccolo. 2009. The fifth pascal recognizing textual entailment challenge. In *TAC*.

Guanzheng Chen, Fangyu Liu, Zaiqiao Meng, and Shangsong Liang. 2022. Revisiting parameter-efficient tuning: Are we really there yet? In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 2612–2626, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Joon-Young Choi, Junho Kim, Jun-Hyung Park, Wing-Lam Mok, and SangKeun Lee. 2023. SMoP: Towards efficient and effective prompt tuning with sparse mixture-of-prompts. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 14306–14316, Singapore. Association for Computational Linguistics.

Christopher Clark, Kenton Lee, Ming-Wei Chang, Tom Kwiatkowski, Michael Collins, and Kristina Toutanova. 2019. BoolQ: Exploring the surprising difficulty of natural yes/no questions. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 2924–2936, Minneapolis, Minnesota. Association for Computational Linguistics.

Marie-Catherine De Marneff, Mandy Simons, and Judith Tonhauser. 2019. The commitmentbank: Investigating projection in naturally occurring discourse. *proceedings of Sinn und Bedeutung 23*.

Zeyu Han, Chao Gao, Jinyang Liu, Jeff Zhang, and Sai Qian Zhang. 2024. Parameter-efficient fine-tuning for large models: A comprehensive survey. *Preprint*, arXiv:2403.14608.

Junxian He, Chunting Zhou, Xuezhe Ma, Taylor Berg-Kirkpatrick, and Graham Neubig. 2022. Towards a unified view of parameter-efficient transfer learning. *Preprint*, arXiv:2110.04366.

Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin de Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. 2019. Parameter-efficient transfer learning for nlp. *Preprint*, arXiv:1902.00751.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2021. Lora: Low-rank adaptation of large language models. *Preprint*, arXiv:2106.09685.

Yuezihan Jiang, Hao Yang, Junyang Lin, Hanyu Zhao, An Yang, Chang Zhou, Hongxia Yang, Zhi Yang, and Bin Cui. 2022. Instance-wise prompt tuning for pretrained language models. *Preprint*, arXiv:2206.01958.

Daniel Khashabi, Snigdha Chaturvedi, Michael Roth, Shyam Upadhyay, and Dan Roth. 2018. Looking beyond the surface:a challenge set for reading comprehension over multiple sentences. In *Proceedings of North American Chapter of the Association for Computational Linguistics (NAACL)*.

Brian Lester, Rami Al-Rfou, and Noah Constant. 2021. The power of scale for parameter-efficient prompt tuning. *Preprint*, arXiv:2104.08691.

Xiang Lisa Li and Percy Liang. 2021. Prefix-tuning: Optimizing continuous prompts for generation. *Preprint*, arXiv:2101.00190.

Xiangyang Liu, Tianxiang Sun, Xuanjing Huang, and Xipeng Qiu. 2022a. Late prompt tuning: A late prompt could be better than many prompts. *Preprint*, arXiv:2210.11292.

Xiao Liu, Kaixuan Ji, Yicheng Fu, Weng Lam Tam, Zhengxiao Du, Zhilin Yang, and Jie Tang. 2022b. P-tuning v2: Prompt tuning can be comparable to fine-tuning universally across scales and tasks. *Preprint*, arXiv:2110.07602.

Xiao Liu, Yanan Zheng, Zhengxiao Du, Ming Ding, Yujie Qian, Zhilin Yang, and Jie Tang. 2023. Gpt understands, too. *Preprint*, arXiv:2103.10385.

Fang Ma, Chen Zhang, Lei Ren, Jingang Wang, Qifan Wang, Wei Wu, Xiaojun Quan, and Dawei Song. 2022. XPrompt: Exploring the extreme of prompt tuning. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 11033–11047, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Mohammad Taher Pilehvar and os'e Camacho-Collados. 2018. Wic: 10, 000 example pairs for evaluating context-sensitive representations. *CoRR*, abs/1808.09121.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2023. Exploring the limits of transfer learning with a unified text-to-text transformer. *Preprint*, arXiv:1910.10683.

Melissa Roemmele, Cosmin Adrian Bejan, and Andrew S Gordon. 2011. Choice of plausible alternatives: An evaluation of commonsense causal reasoning. In *2011 AAAI Spring Symposium Series*.

Noam Shazeer, Azalia Mirhoseini, Krzysztof Maziarz, Andy Davis, Quoc Le, Geoffrey Hinton, and Jeff Dean. 2017. Outrageously large neural networks: The sparsely-gated mixture-of-experts layer. *Preprint*, arXiv:1701.06538.

Noam Shazeer and Mitchell Stern. 2018. Adafactor: Adaptive learning rates with sublinear memory cost. *Preprint*, arXiv:1804.04235.

Zhengxiang Shi and Aldo Lipani. 2024. Dept: Decomposed prompt tuning for parameter-efficient fine-tuning. *Preprint*, arXiv:2309.05173.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2023. Attention is all you need. *Preprint*, arXiv:1706.03762.

Tu Vu, Brian Lester, Noah Constant, Rami Al-Rfou', and Daniel Cer. 2022. SPoT: Better frozen model adaptation through soft prompt transfer. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5039–5059, Dublin, Ireland. Association for Computational Linguistics.

Alex Wang, Yada Pruksachatkun, Nikita Nangia, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R Bowman. 2019. Superglue: A stickier benchmark for general-purpose language understanding systems. *arXiv preprint arXiv:1905.00537*.

Qifan Wang, Yuning Mao, Jingang Wang, Hanchao Yu, Shaoliang Nie, Sinong Wang, Fuli Feng, Lifu Huang, Xiaojun Quan, Zenglin Xu, and Dongfang Liu. 2023a. APrompt: Attention prompt tuning for efficient adaptation of pre-trained language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 9147–9160, Singapore. Association for Computational Linguistics.

Yaqing Wang, Sahaj Agarwal, Subhabrata Mukherjee, Xiaodong Liu, Jing Gao, Ahmed Hassan Awadallah, and Jianfeng Gao. 2022. Adamix: Mixture-of-adaptations for parameter-efficient model tuning. *Preprint*, arXiv:2205.12410.

Zhen Wang, Rameswar Panda, Leonid Karlinsky, Rogerio Feris, Huan Sun, and Yoon Kim. 2023b. Multitask prompt tuning enables parameter-efficient transfer learning. *Preprint*, arXiv:2303.02861.

Zhuofeng Wu, Sinong Wang, Jiatao Gu, Rui Hou, Yuxiao Dong, V.G.Vinod Vydiswaran, and Hao Ma. 2022. IDPG: An instance-dependent prompt generation method. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5507–5521, Seattle, United States. Association for Computational Linguistics.

Yao Xiao, Lu Xu, Jiaxi Li, Wei Lu, and Xiaoli Li. 2023. Decomposed prompt tuning via low-rank reparameterization. *Preprint*, arXiv:2310.10094.

Zhen-Ru Zhang, Chuanqi Tan, Haiyang Xu, Chengyu Wang, Jun Huang, and Songfang Huang. 2023. Towards adaptive prefix tuning for parameter-efficient language model fine-tuning. *Preprint*, arXiv:2305.15212.

Wei Zhu and Ming Tan. 2023. SPT: Learning to selectively insert prompts for better prompt tuning. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 11862–11878, Singapore. Association for Computational Linguistics.

A Appendix

A.1 Experimental Results

A.1.1 Detailed Experiment Tables

Table 2 presents the experimental results on six SuperGLUE tasks on T5-base.

A.1.2 Memory Usage Analysis

Table 3 presents the peak memory used during training (GB).

A.1.3 Time Performance Analysis

Table 4 presents the training time (s/100 steps).

A.2 Dataset Details

Table 5 provides detailed information about six SuperGLUE datasets, including their sizes, metrics, and tasks.

A.3 Mask Visualization

The observation from Figure 2 shows that most values in each mask are high, indicating that most soft tokens after being trained are important, and only a few negative tokens need to be masked. Additionally, each mask captures different features, suggesting that the masks are effectively trained to capture distinct information.

Model	Method	Total Prompt Length	Utilized Prompt Length	Number of masks	BoolQ	CB	COPA	MultiRC	RTE	WIC	Average Score (%)
T5-base	Full Fine-tuning	-	-	-	$81.90_{0.1}$	$96.41_{0.8}$	$64.31_{0.5}$	$80.20_{0.2}$	$79.20_{0.2}$	$67.02_{0.3}$	$78.21_{0.3}$
	P-tuning	5	5	-	$79.0_{0.1}$	$89.3_{3.7}$	$59.0_{1.0}$	$79.2_{0.1}$	$73.8_{1.4}$	$65.4_{1.3}$	$74.4_{1.8}$
		20	20	-	$78.7_{0.2}$	$91.7_{2.7}$	$58.3_{3.8}$	$79.3_{0.2}$	$77.1_{3.3}$	$65.9_{0.7}$	$75.2_{2.1}$
	Prompt Tuning	5	5	-	$78.5_{0.0}$	$89.3_{1.8}$	$54.0_{3.6}$	$79.1_{0.1}$	$69.9_{0.8}$	$64.4_{0.0}$	$72.5_{1.7}$
		20	20	-	$78.6_{0.0}$	$86.9_{2.1}$	$55.0_{3.5}$	$79.2_{0.2}$	$70.6_{1.8}$	$64.3_{0.2}$	$72.4_{1.8}$
	SMoP	10	5	-	$78.5_{0.0}$	$92.9_{0.0}$	$58.0_{0.6}$	$79.4_{0.0}$	$76.4_{1.3}$	$64.9_{0.8}$	$75.0_{2.0}$
		20	5	-	$\mathbf{79.4}_{0.3}$	$94.6_{1.8}$	$58.3_{2.9}$	$\mathbf{79.6}_{0.1}$	$77.5_{3.2}$	$\mathbf{65.2}_{0.5}$	$75.8_{1.9}$
		50	5	-	$79.3_{0.1}$	$92.3_{1.0}$	$58.7_{4.2}$	$79.3_{0.0}$	$77.1_{0.3}$	$65.2_{0.4}$	$75.3_{1.8}$
		100	5	-	$79.0_{0.3}$	$93.4_{2.0}$	$55.3_{3.1}$	$79.3_{0.2}$	$76.9_{2.0}$	$64.3_{0.2}$	$74.7_{1.7}$
		20	10	-	$78.7_{0.1}$	$93.5_{1.0}$	$59.3_{3.5}$	$79.2_{0.3}$	$76.0_{1.8}$	$64.2_{0.1}$	$75.1_{1.7}$
		40	10	-	$78.6_{0.1}$	$92.3_{3.7}$	$56.0_{1.7}$	$78.9_{0.1}$	$76.9_{0.8}$	$66.4_{0.8}$	$74.8_{1.7}$
		100	10	-	$78.5_{0.1}$	$95.8_{1.0}$	$57.7_{2.5}$	$79.2_{0.1}$	$75.1_{1.0}$	$64.8_{1.7}$	$75.2_{1.4}$
		200	10	-	$79.0_{0.4}$	$91.1_{1.8}$	$56.0_{3.5}$	$79.4_{0.1}$	$74.2_{2.8}$	$64.9_{0.7}$	$74.1_{2.0}$
	Prefix Tuning	5	5	-	$78.75_{0.1}$	$91.46_{0.80}$	$58.6_{3.4}$	79.3_{0}	$75.62_{0.81}$	$63.9_{0.36}$	$74.6_{0.9}$
		10	10	-	$78.83_{0.1}$	$91.46_{0.8}$	$58.8_{1.3}$	79.5	$74.63_{0.4}$	$63.76_{0.4}$	$74.5_{0.6}$
		20	20	-	$78.85_{0.1}$	$91.5_{0.8}$	$62_{3.2}$	79.2_{0}	$76.8_{0.5}$	$64.2_{0.3}$	$75.42_{0.1}$
	Multi-mask Prefix Tuning	5	5	1	$78.75_{0.1}$	$93.94_{2.4}$	$60.2_{3.3}$	79.4_{0}	$\mathbf{77.56}_{0.6}$	$64.76_{0.3}$	$75.76_{1.1}$
		5	5	2	$78.75_{0.2}$	$92.86_{1.6}$	$62.8_{1.9}$	79.4_{0}	$76.92_{1.6}$	$64.52_{0.2}$	$75.86_{0.9}$
		5	5	4	$78.8_{0.3}$	$93.92_{0.9}$	$62_{1.6}$	79.5_{0}	$77_{0.5}$	64.5	$75.69_{0.7}$
		10	10	1	$78.75_{0.1}$	$93.92_{2.9}$	$62_{3.3}$	79.42_{1}	77.42_{1}	$64.3_{0.5}$	$75.28_{1.3}$
		10	10	2	$78.75_{0.2}$	$94.28_{1.5}$	$60.75_{2.4}$	79.2_{0}	$76.62_{0.5}$	$64.44_{0.7}$	$75.67_{0.9}$
		10	10	4	$78.9_{0.1}$	$94.62_{0.9}$	$\mathbf{63.6}_{1.4}$	79.4_{0}	$77.26_{0.8}$	$64.48_{0.5}$	$\underline{76.38}_{0.7}$
		20	20	1	79_{0}	$\mathbf{96.04}_{0.8}$	$63.4_{2.7}$	79.4_{0}	$77.26_{0.6}$	$64.34_{0.6}$	$\mathbf{76.57}_{0.8}$
		20	20	2	$78.85_{0.21}$	$95.08_{1.7}$	$63.2_{2.8}$	79.4_{0}	$\mathbf{77.56}_{1.3}$	$64.6_{0.3}$	$75.86_{1.1}$
		20	20	4	$79.0_{0.14}$	$93.75_{1.0}$	$61.8_{2.4}$	79.3_{0}	$76.54_{0.7}$	$64.68_{0.5}$	$75.85_{0.8}$

Table 2: Experimental results on baseline methods and SMoP on six SuperGLUE tasks with T5-base. Subscripts of each score represent the standard deviation over multiple runs.

Model	Method	Total Prompt Length	Utilized Prompt Length	BoolQ	CB	COPA	MultiRC	RTE	WiC	Average
T5-base	Fine-tuning	-	-	27.0	14.3	3.1	27.0	13.9	4.1	14.9
	Prompt Tuning	100	100	21.8	16.0	5.0	21.8	15.6	6.2	14.4
	P-Tuning	20	20	21.8	12.0	2.7	21.8	11.7	3.5	12.3
	SMoP	5	5	21.8	11.3	2.3	21.8	11.0	3.1	11.9
	Prefix Tuning	5	5	4.37	2.97	1.33	4.52	3.04	1.54	**2.96**
	Multi-mask	5	5	6.64	3.09	1.46	6.64	3.30	1.62	<u>3.79</u>

Table 3: Peak memory (GB) during training on SuperGLUE tasks

Model	Method	Total Prompt Length	Utilized Prompt Length	BoolQ	CB	COPA	MultiRC	RTE	WiC	Average
T5-base	Fine-tuning	-	-	105.8	92.6	45.8	131.6	76.5	36.0	81.4
	Prompt Tuning	100	100	93.1	90.3	37.2	103.7	71.4	28.4	70.7
	P-Tuning	20	20	84.8	85.9	30.5	108.2	59.0	21.1	64.9
	SMoP	5	5	82.5	74.1	30.8	104.6	54.2	19.8	61.0
	Prefix Tuning	5	5	44.06	37.78	14.27	66.72	24.99	7.81	32.61
	Multi-mask	5	5	47.95	35.68	8.31	57.47	26.83	9.13	**30.90**

Table 4: Training time (s/100 steps) on SuperGLUE tasks.

Dataset	Train	Valid	Test	Task	Metrics
BoolQ	9427	3270	3245	Question Answering	Accuracy
CB	250	57	250	Natural Language Inference	Accuracy
COPA	400	100	500	Question Answering	Accuracy
MultiRC	5100	953	1800	Question Answering	F1-score
RTE	2500	278	300	Natural Language Inference	Accuracy
WiC	6000	638	1400	Word Sense Disambiguation	Accuracy

Table 5: The data statistics and metrics of six SuperGLUE tasks. Train, Valid and Test indicate the number of samples in the official training, validation and test sets, respectively.

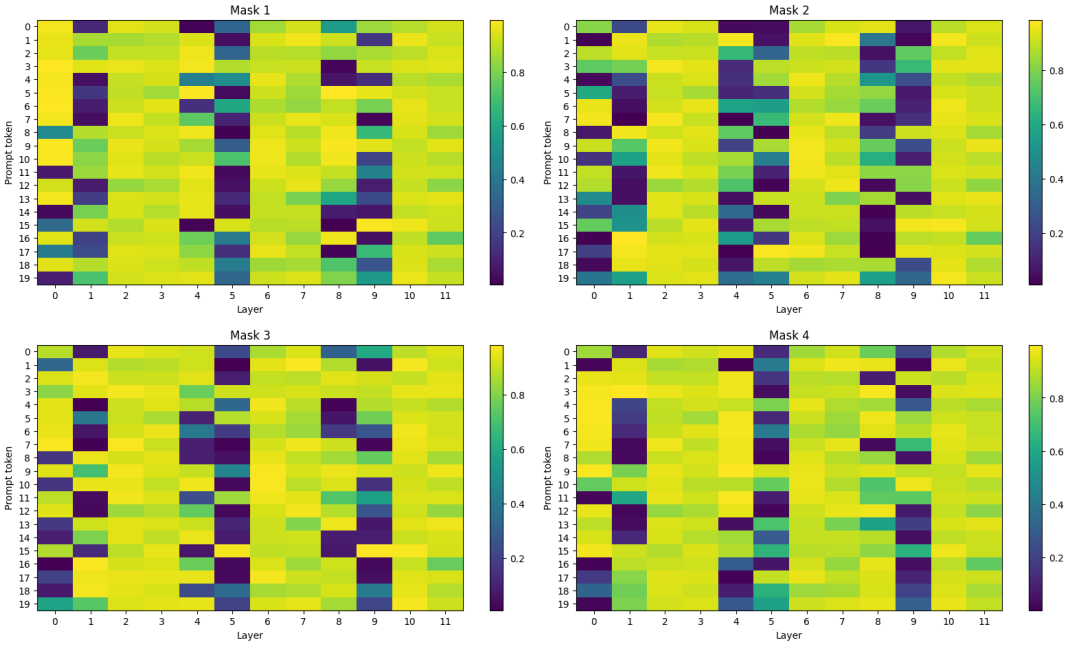

Figure 2: Masks trained on the RTE task with a prompt length of 20