XPROMPT: Exploring the Extreme of Prompt Tuning

Fang Ma[1], **Chen Zhang**[1], **Lei Ren**[2], **Jingang Wang**[2]✉, **Qifan Wang**[3], **Wei Wu**[2], **Xiaojun Quan**[4], **Dawei Song**[1]✉

[1] Beijing Institute of Technology {mfang,czhang,dwsong}@bit.edu.cn

[2] Meituan NLP {wangjingang02,wuwei30}@meituan.com, renlei_work@163.com

[3] Meta AI wqfcr@fb.com

[4] Sun Yat-Sen University quanxj3@mail.sysu.edu.cn

Abstract

Prompt tuning learns soft prompts to condition frozen Pre-trained Language Models (PLMs) for performing downstream tasks in a parameter-efficient manner. While prompt tuning has gradually reached the performance level of fine-tuning as the model scale increases, there is still a large performance gap between prompt tuning and fine-tuning for models of moderate and small scales (typically less than 11B parameters). In this paper, we empirically show that the trained prompt tokens can have a negative impact on a downstream task and thus degrade its performance. To bridge the gap, we propose a novel PROMPT tuning model with an eXtremely small scale (XPROMPT) under the regime of lottery tickets hypothesis. Specifically, XPROMPT eliminates the negative prompt tokens at different granularity levels through a hierarchical structured pruning, yielding a more parameter-efficient prompt yet with a competitive performance. Comprehensive experiments are carried out on SuperGLUE tasks, and the extensive results indicate that XPROMPT is able to close the performance gap at smaller model scales.

1 Introduction

Pre-trained Language Models (PLMs) have been widely applied and achieved a remarkable success in various NLP tasks (Devlin et al., 2019; Raffel et al., 2020; Zhou et al., 2020) under the *pre-train-then-fine-tune* paradigm (Liu et al., 2019). Despite of its compelling performance, fine-tuning is parameter-inefficient for large scale PLMs due to the fact that the memory footprint is proportional to the number of trainable parameters whose gradients and optimizer states need to be stored (Guo et al., 2021).

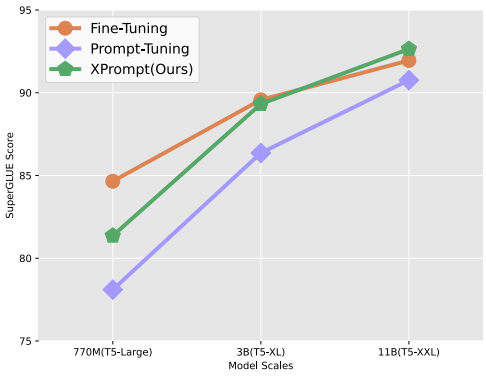

Figure 1: XPROMPT outperforms the vanilla Prompt-Tuning (Lester et al., 2021) and can significantly improve over Prompt-Tuning across tasks and model scales. It is worth noting that there is a small performance gap between prompt tuning and fine-tuning on T5-XXL (11B) due to different hyperparameter settings and initialization. Similar observations have been found in Figure3-a and Figure3-b of Lester et al. (2021).

Recently, Prompt-Tuning (Lester et al., 2021; Liu et al., 2021b) has been proposed to address this issue by prepending a *soft prompt* to the input and only updating the parameters of prompt tokens during tuning. Prompt-Tuning provides a parameter-efficient alternative to fine-tuning, since the scale of the soft prompt is tens of thousand smaller. It is also conceptually simpler and more flexible than other parameter-efficient tuning methods such as Adapters that require intrusive modifications to transformer layers (Houlsby et al., 2019; Guo et al., 2021). Using fewer tunable parameters, prompt tuning achieves competitive performance to fine-tuning with the increase of the model scale. However, there is still a large performance gap between prompt tuning and fine-tuning for models of smaller scales (as shown in Figure 1).

This paper aims to fill the gap, from the perspective of the lottery tickets hypothesis (LTH) (Frankle and Carbin, 2019). We are motivated by an observation that, on a specific task, not all prompt tokens contribute equally to the task performance, while

✉ Dawei Song and Jingang Wang are the corresponding authors.

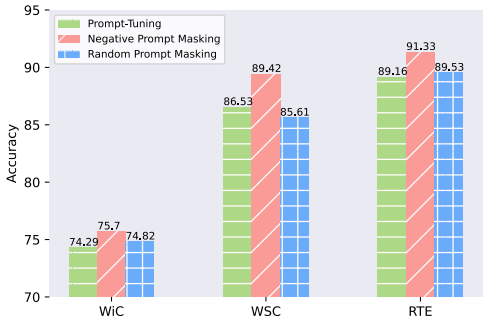

Figure 2: The performance comparison of Prompt-Tuning, Negative Prompt Masking and Random Prompt Masking with T5-XL(3B) on three SuperGLUE tasks. Prompt-Turning uses all prompt tokens. Negative Prompt Masking masks selected (negative) prompt tokens with low importance scores. Random Prompt Masking randomly masks the same number of tokens as in Negative Prompt Masking.

certain prompt tokens may even bring a negative impact. Figure 2 provides a preliminary result of this observation. These *negative prompt tokens* can be circumvented under the regime of LTH. Essentially, LTH states that an over-parameterized network contains a sub-network that, when initialized and trained in isolation, can match or exceed the test accuracy of the original network after training for at most the same number of iterations. The sub-network is called lottery ticket, and the collection of the tickets is referred to as winning tickets in PLMs (Liang et al., 2021). In the problem of prompt-tuning, the winning tickets are the collection of positive prompt tokens that can achieve the same performance as using the entire collection of prompts, while the losing tickets are the collection of negative prompt tokens.

Therefore, the key is to identify the winning tickets and eliminate the losing ones, in the collection of trained prompt tokens. In particular, we propose to eliminate the losing tickets through a hierarchical structured pruning, which first removes negative tokens at the token-level and then prunes the remaining ones at a finer graularity level, i.e., the piece-level, for a better trade-off between effectiveness and efficiency. In line with LTH, weight rewinding (Renda et al., 2020) is adopted to retrain the identified positive soft prompts. With the elimination of negative prompt tokens, a more parameter-efficient PROMPT of an eXtremely small scale (XPROMPT) is obtained.

To verify the effectiveness of XPROMPT, we conduct an extensive set of experiments on SuperGLUE (Wang et al., 2019) in both high-resource and low-resource scenarios. As shown in Figure 1 and Table 1, the results demonstrate that XPROMPT significantly improves the prompt-tuning methods across tasks and model scales. For models of moderate scales, XPROMPT closes the gap and achieves a performance comparable to fine-tuning. For models of large scales, XPROMPT also leads to large performance gains over Prompt-Tuning, and even exceeds fine-tuning for most tasks.

2 Related Work

2.1 Pre-trained Language Models

Pre-trained Language Models (PLMs) have achieved remarkable success in various NLP tasks (Zhou et al., 2020; Raffel et al., 2020; Brown et al., 2020). BERT (Devlin et al., 2019) and RoBERTa (Liu et al., 2019) are two pioneers that learn contextual representations with masked language model (MLM) and next sentence prediction pre-training tasks. Recently, a series of large scale PLMs have emerged with different pre-training designs, such as GPT-2 (Radford et al., 2019), GPT-3 (Brown et al., 2020), ELECTRA (Clark et al., 2020), XLNet (Yang et al., 2019), BART (Lewis et al., 2020) and T5 (Raffel et al., 2020). However, with the exploding number of parameters, fine-tuning models become parameter-inefficient and computationally expensive due to the maintenance of all parameters in the PLMs. Moreover, one has to fine-tune different models for different tasks and store them separately, which is resource-intensive.

2.2 Prompt Learning in NLP

With the development of GPT-3 (Brown et al., 2020), prompt learning has drawn much attention in the NLP community (Liu et al., 2021a; Ding et al., 2022), which enables efficient learning by adding a number of *prompt* tokens to the input. Prompt learning has been proven to be effective in various downstream tasks (Davison et al., 2019; Gong and Eldardiry, 2021; Radford et al., 2019; Wang et al., 2021; Khashabi et al., 2020). Recently, prompt has been extended from discrete tokens (tokens in the vocabularies) to continuous tokens (trainable embeddings), i.e., soft prompt (Li and Liang, 2021; Zhong et al., 2021; Qin and Eisner, 2021). For example, (Lester et al., 2021) proposes a parameter-efficient prompt tuning approach by only tuning soft prompts and fixing the entire parameters in PLM. Prompt tuning achieves great success and shows that it can reach the performance of fine-tuning with large PLM. However, there is

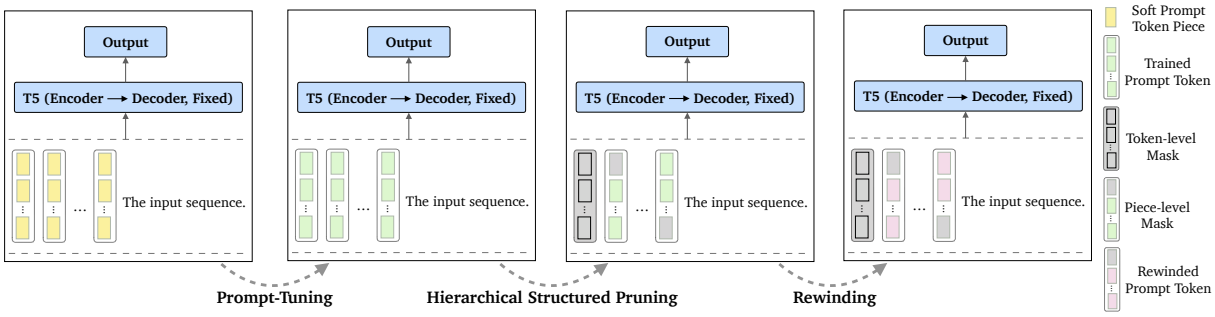

Figure 3: The illustration of our proposed XPROMPT approach. XPROMPT consists of three stages, namely *Prompt-Tuning*, *Hierarchical Structured Pruning* and *Rewinding*. Among all the stages, the parameters of T5 are frozen - only the parameters of the prompts are tuned. The prompts trained in the previous stage are fed into the next stage as the initialization prompts. The change of color represents the process that the parameters of the prompts are tuned or pruned.

still a large performance gap between prompt tuning and fine-tuning for models of moderate scales. More recently, (Vu et al., 2021) proposes a prompt-based transfer learning approach, SPOT, to improve the performance of prompt tuning, which learns a prompt on source tasks and then applied to initialize the target task's prompt. Most recently, (He et al., 2022) proposes HyperPrompt which uses the hypernetworks to generate hyper-prompts and obtains superior performance. However, it needs to tune all parameters and shows that only tuning task-conditioned parameters is not enough to achieve competitive results as full model fine-tuning for multi-task learning.

2.3 Lottery Ticket Hypothesis

The lottery ticket hypothesis (Frankle and Carbin, 2019) finds that an over-parameterized network contains a subnetwork that is initialized such that - when trained in isolation - it can match the test accuracy of the original network after training for at most the same number of iterations. The subnetwork is called lottery ticket. In NLP, the collection of lottery tickets is referred to as winning tickets in highly over-parametrized models, e.g., PLMs (Liang et al., 2021). Such winning tickets have demonstrated their abilities to transfer across tasks and datasets (Morcos et al., 2019; Yu et al., 2020; Desai et al., 2019). Recently, Chen et al. (2021) has shown the existence of the winning tickets in PLMs. Liang et al. (2021) observes that the generalization performance of the winning tickets can even exceed that of the full model.

3 Preliminary

Built upon the text-to-text approach of T5 (Raffel et al., 2020), prompt tuning formulates all tasks as text generation by prepending additional l tunable soft prompt tokens to the input and only updating the parameters of the inserted soft prompt tokens. Specifically, given a series of n input tokens $X = \{x_1, x_2, ..., x_n\}$, T5 first generates the token embeddings $X_e \in \mathbb{R}^{n \times e}$, where e is the dimension of the embedding space. It also generates soft prompt embeddings $P_e = \{p_1, p_2, ..., p_m\} \in \mathbb{R}^{m \times e}$, where m is the length of the soft prompt. Then the soft prompts are prepended to the input sequence as $[P_e; X_e] \in \mathbb{R}^{(m+n) \times e}$. The goal of prompt tuning is to maximize the likelihood of the labels Y by only optimizing over P_e:

$$\arg\max_{P_e} \log p(Y|[P_e; X_e]) \quad (1)$$

Prompt tuning becomes more effective as the model scale increases. However, there is still a significant performance gap between prompt tuning and fine-tuning especially for models of small and moderate scales. Our hypothesis is that not all soft prompt tokens contribute equally to the performance after training on the target task. There exist certain soft prompt tokens that may have negative impacts on the task. Therefore, combining the idea of the lottery ticket hypothesis, we propose XPROMPT with hierarchical structured pruning to identify the optimal soft prompts and bridge the performance gap.

4 XPROMPT

The overall process of XPROMPT is illustrated in Figure 3, which consists of three main stages: *Prompt-Tuning*, *Hierarchical Structured Pruning* and *Rewinding*. Specifically, the prompt tuning learns an initial set of values for all soft prompt tokens on the target task. During the hierarchical structured pruning, token-level and piece-level

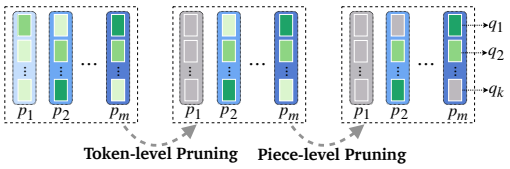

Figure 4: The illustration of *Hierarchical Structured Pruning*. Among them, the shade of the color indicates the level of the importance score, and the darker the color, the higher the importance score of the corresponding structure (token or piece).

pruning processes are repeatedly conducted to identify the optimal soft tokens and pieces at different compression ratios. Finally, a weight rewinding technique is applied to re-train the soft prompts.

4.1 Prompt Tuning

Prompt tuning approaches prepend a number of soft prompt tokens to the input, and only tune soft prompts by fixing the entire parameters in PLM. Prompt tuning has been proven to be effective in various downstream tasks. In our prompt tuning stage, following previous work (Liang et al., 2021), we conduct a complete tuning on the target task to obtain the embeddings for all the soft prompt tokens. These trained soft prompts are used as initialization in the hierarchical structured pruning.

4.2 Hierarchical Structured Pruning

Hierarchical structured pruning is designed to separate negative prompt tokens from the trained prompt tokens, and identify an optimal set of soft prompts. The approach is illustrated in Figure 4. The token-level pruning is first used to identify negative prompt tokens, however, the rest prompt tokens may still contain negative pieces. Thus, the piece-level pruning is then applied to identify more fine-grained negative prompt pieces within each prompt token. Token-level and piece-level pruning together play a better trade-off between effectiveness and efficiency.

4.2.1 Token-level Pruning

To identify negative prompt tokens in the trained prompt tokens, we associate mask variable γ_i to each soft prompt token vector p_i:

$$\hat{P}_e = \gamma \cdot P_e \tag{2}$$

where $\gamma = \{\gamma_1, \gamma_2, ..., \gamma_m\}, \gamma_i \in \{0, 1\}$, and a 0 value indicates that the corresponding soft prompt token is pruned.

We then calculate the importance score (Michel et al., 2019) of each token to distinguish the negative prompt tokens from the other ones. The importance score is defined as the expected sensitivity of the model outputs to the mask variables. Formally, the importance score I_{p_i} of each soft prompt token p_i is calculated as:

$$I_{p_i} = \mathbb{E}_{x \sim \mathcal{D}_x} \mid \frac{\partial \mathcal{L}(x)}{\partial \gamma_i} \mid \tag{3}$$

where $\mathcal{L}$ is the loss function and $\mathcal{D}_x$ is the training data distribution.

Essentially, the importance score of each soft prompt token indicates its individual contribution to the model performance. A low importance score means that the corresponding soft prompt token has a small or even negative contribution to the model. In other words, such a soft prompt token contains negligible prompt information for generating the outputs. On the contrary, a large importance score implies a major contribution with more meaningful prompt information. Therefore, the prompt tokens with low importance scores are most likely negative prompt tokens, which are pruned during the token-level pruning stage.

4.2.2 Piece-level Pruning

Token-level pruning finds the most important soft prompt tokens. However, it may not be sufficient as there are still fine-grained negative prompt pieces remaining in the embedding of each soft prompt token. Different pieces of the embedding may lead to different effects on downstream tasks. Therefore, we further conduct piece-level pruning to eliminate the negative prompt pieces within each token. In particular, we divide the embedding vector of each soft prompt token p_{i_e} into k pieces with equal scale, $q_e = \{q_{1_e}, q_{2_e}, ..., q_{k_e}\}$, and treat each piece as an independent unit that can be optimized with gradient updates. Mask variable ζ_i is associated with each piece in the soft prompt token to identify the negative prompt pieces:

$$\hat{q_e} = \zeta \cdot q_e \tag{4}$$

where $\zeta = \{\zeta_1, \zeta_2, ..., \zeta_k\}, \zeta_i \in \{0, 1\}$, and 0 value indicates that the corresponding piece is pruned.

We then calculate the importance score I_{q_i} of each piece for every prompt token embedding to prune the low-importance pieces:

$$I_{q_i} = \mathbb{E}_{x \sim \mathcal{D}_x} \mid \frac{\partial \mathcal{L}(x)}{\partial \zeta_i} \mid \tag{5}$$

Similar to the token-level importance score, a low piece-level importance score indicates that the piece has a small or even negative contribution towards the model performance. Such low-importance pieces contain limited information for generating the outputs. We repeatedly conduct both token-level and piece-level pruning to obtain the sub-prompt tokens and pieces at different compression ratios.

4.3 Rewinding

The lottery ticket hypothesis (LTH) (Frankle and Carbin, 2019) states that sparse subnetworks (the unpruned prompts) can be trained in isolation to the same accuracy as the original network (all prompts), and proposes training to pruning and then rewinding the unpruned weights. Following the idea in LTH, we adopt the weight rewinding technique (Renda et al., 2020) to re-train the soft prompts after the two-level hierarchical structured pruning. Specifically, we reset the parameters of the selected optimal soft prompts using their values after the prompt tuning stage. The other soft prompts are pruned by setting the corresponding mask variables to 0. Finally, we re-train the soft prompts using the original learning strategies in prompt tuning.

5 Experiments

5.1 Datasets

To cover broad and diverse NLP tasks in our experiments, we evaluate our method on various datasets of SuperGLUE benchmark (Wang et al., 2019) in both high-resource and low-resource scenarios. Due to restricted test access for SuperGLUE, following previous works (Lester et al., 2021; Ding et al., 2021), we tune the prompt model on the training set for a fixed number of steps and report results on the validation set using the best checkpoint. The detailed description, statistics and metrics of SuperGLUE tasks are provided in Table 9 of Appendix E. The soft prompt templates and generation verbalizers are provided in Table 10 of Appendix E.

5.2 Baselines

Fine-Tuning We compare with the standard fine-tuning approach (Raffel et al., 2020; Aribandi et al., 2021) of T5, where all the pre-trained parameters are fine-tuned on each target task separately.

Prompt-Tuning The vanilla prompt tuning approach of (Lester et al., 2021) showed that prompt tuning is a competitive technique for adapting frozen PLMs to downstream tasks.

P-Tuning (Liu et al., 2021c) is a prompt-based method that uses the masked PLM to convert the target task into a cloze problem. It employs soft-prompting techniques to optimize prompts in the continuous space. We also compare with its second version P-TuningV2 (Liu et al., 2021b).

Prefix-Tuning (Li and Liang, 2021) is a lightweight alternative to fine-tuning for natural language generation tasks, which only optimizes a small continuous task-specific vector (called prefix). Prefix-Tuning prepends the prefix to inputs of every transformer layer independently.

5.3 Implementation

Our method is implemented with the OpenPrompt library (Ding et al., 2021), which is a unified and extensible toolkit for prompt learning. We translate each SuperGLUE dataset into a text-to-text format following (Raffel et al., 2020), except that we omit the task names prepend to inputs indicating which SuperGLUE task an example belongs to.

Our XPROMPT is built on top of the pre-trained T5 checkpoints of three scales: `Large`, `XL`, `XXL` with 770M, 3B and 11B parameters, respectively. Following previous studies (Lester et al., 2021; Ding et al., 2021), we train our prompts for 100 epochs with a constant learning rate of 0.3 and a batch size of 16. (Lester et al., 2021) shows that an increase beyond 20 tokens only yields marginal gains, so throughout our experiments, we set the default number of prompt tokens to 20 to control the number of trainable parameters and use sampled vocabulary to initialize the prompt parameters. The number of pieces in each token is set to 16. The pruning frequencies are linearly searched from {10%, 20%, 30%, 40%, 50%, 60%, 70%, 80%, 90%}. The weight rewinding is applied only once to re-train the pruned soft prompts. The best checkpoints are selected via early stopping on the development set. The models are trained using the Adafactor (Shazeer and Stern, 2018) optimizer with weight decay $1e$-5.

6 Results

6.1 Results on High-resource Scenarios

XPROMPT significantly improves the performance of prompt tuning and helps close the gap with fine-tuning across all model scales.

Model		WiC Acc	WSC Acc	CB Acc	COPA Acc	RTE Acc	Boolq Acc	MultiRC $F1_a$	Average Score
T5-Large 770M	Fine-Tuning*	73.50	88.50	94.30	72.0	90.60	88.30	85.40	84.65
	P-Tuning	70.37	64.42	92.85	76.0	79.78	83.02	79.96	78.06
	Prefix-Tuning	62.50	64.46	78.78	-	55.70	65.17	60.19	64.46
	Prompt-Tuning	72.25	68.26	82.14	76.0	85.19	83.02	79.86	78.10
	XPROMPT	**73.51**↑1.26	**70.39**↑2.13	**91.07**↑8.93	**82.0**↑6.0	**87.72**↑2.53	**83.82**↑0.8	**81.02**↑1.16	**81.36**↑3.26
T5-XL 3B	Fine-Tuning*	74.30	95.20	92.00	96.0	91.70	89.60	88.20	89.57
	P-Tuning	72.54	81.73	91.07	73.0	89.53	84.54	85.45	82.55
	Prompt-Tuning	74.29	86.53	91.07	91.0	89.16	87.58	84.89	86.36
	XPROMPT	**76.95**↑2.66	**91.34**↑4.84	**92.85**↑1.78	**95.0**↑4.0	**92.79**↑3.63	**89.00**↑1.42	**87.34**↑2.45	**89.32**↑2.96
T5-XXL 11B	Fine-Tuning*	78.50	95.20	100.00	99.0	92.10	90.40	88.60	91.97
	P-Tuning	76.80	94.23	92.85	93.0	89.80	86.98	87.56	88.75
	Prompt-Tuning	76.10	96.15	96.42	98.0	91.69	89.08	87.90	90.76
	XPROMPT	**77.69**↑1.59	**97.11**↑0.96	**100.00**↑3.58	**99.0**↑1.0	**94.94**↑3.25	**90.87**↑1.79	**88.90**↑1.0	**92.64**↑1.88

Table 1: Main experimental results (%) on seven SuperGLUE tasks. Our method and better results are in bold (the larger, the better). The small number next to each score indicates performance improvement (↑) compared with the vanilla Prompt-Tuning. Methods with '*' indicate the results reported in Aribandi et al. (2021). We only present the results of Prefix-Tuning on T5-Large, since it can diverge with larger models (Ding et al., 2022). The '-' results in Prefix-Tuning indicate diverged results in the corresponding task.

Table 1 and Table 8 (in the appendix) present the main results on SuperGLUE. We compare XPROMPT with strong prompt learning baselines, including Prompt-Tuning, Prefix-Tuning, P-Tuning and P-TuningV2 for different PLMs and model scales. It can be seen that XPROMPT outperforms vanilla Prompt-Tuning by a large margin across all tasks and model scales. For instance, XPROMPT yields an improvement of 3.26 %, 2.96 %, and 1.88 % in terms of average score on T5-Large, T5-XL, and T5-XXL, respectively. We also observe that the performance of Prompt-Tuning and P-Tuning are comparable at the same model scale. Moreover, P-TuningV2 outperforms Prompt-Tuning and P-Tuning on CB, RTE, and Boolq. However, XPROMPT achieves more predominant performances than P-TuningV2 at similar model scales, demonstrating its effectiveness. It is worth noting that Prefix-Tuning is less performable on most NLU tasks, since it is designed for natural language generation (NLG) tasks.

It is clear from Table 1 that XPROMPT enables prompt tuning to match the fine-tuning performance on all tasks with T5-XL, and even exceeds fine-tuning performance on most tasks at the T5-XXL scale. For example, XPROMPT achieves the best average score of 89.32% with T5-XL, leaving only 0.25% gap to fine-tuning. It is worth mentioning that XPROMPT significantly outperforms fine-tuning on WiC, CB and RTE with T5-XL, as well as COPA and WiC with T5-Large. Especially for T5-XXL, XPROMPT achieves the best score of 97.11%, 100.00%, 94.94%, 90.87% and 88.90% on WSC, CB, RTE, Boolq, MultiRC respectively, leading to +1.91%, +0.0%, +2.84%, +0.47%, +0.30% improvements over fine-tuning. We also observe that there are certain gaps between prompt tuning and fine-tuning, especially for small and moderate scale models (see Figure 1). However, our XPROMPT narrows down the gap significantly across all model scales, demonstrating that it learns efficient and informative soft prompts which empower downstream tasks effectively.

Model	Boolq	WiC	RTE
P-Tuning	64.99	54.23	57.40
GPT-3 XL1.3B‡	64.10	53.00	50.90
GPT-3 2.7B‡	**70.30**	51.60	56.30
PromptTuning	69.81	60.81	66.08
XPROMPT	70.23	**62.85**	**67.87**

Table 2: The few-shot (32 samples) results (Acc, %) on three SuperGLUE tasks for the T5-XL model with 20 soft prompt tokens. Methods with '‡' indicate results reported in Schick and Schütze (2021). XPROMPT is better than vanilla Prompt-Tuning and P-Tuning in low resource scenarios.

6.2 Results on Low-resource Scenarios

XPROMPT performs much better in low resource scenarios. Since prompt learning is surprisingly effective in low-resource regime (Schick and Schütze, 2021), we also explore the effect of XPROMPT in low-resource scenarios. Following the setting used in (Schick and Schütze, 2021), we randomly select 32 examples as the new training set for each task using a fixed random seed. We tune the prompt model on the 32-shot training set and directly report the full dev set results using the

best checkpoint.

As demonstrated in Table 2, our XPROMPT further improves the performance of prompt tuning and outperforms the baseline models at the same scale on Boolq, WiC, and RTE. For example, XPROMPT achieves the best score of 62.85% on WiC, +2.04% improvement over Prompt-Tuning. These few-shot results suggest that although overfitting is severe especially when training with limited data, XPROMPT consistently lifts the performance of prompt tuning.

7 Analysis and Discussion

To better understand the effectiveness of the XPROMPT and explore the impact of various factors in XPROMPT, we further conduct a series of ablation studies and analysis.

7.1 Do Positive Prompts and Negative Prompts Exist?

We identify both positive and negative prompts through hierarchical structured pruning. For positive prompts, the first evidence is the large performance improvement of XPROMPT over vanilla prompt tuning across all tasks and model scales, which shows the effectiveness of these positive prompts. Another evidence is the high sparsities of pruning. Figure 9 and Figure 10 in Appendix D show the original and pruned gradient saliency maps (Simonyan et al., 2014) of the importance scores on WSC task, i.e., the gray elements in Figure 10 indicate that the prompt tokens or pieces are pruned due to low importance scores, and the remaining parts are the winning tickets. The performance of XPROMPT with 15% positive sub-prompts is 4.8% higher than the full prompt tuning.

The negative prompts perform worse than Prompt Tuning and XPROMPT. To further investigate the existence and effect of negative prompts, we conduct another experiment to compare prompt tuning performances with different configurations. Specifically, in addition to the vanilla Prompt-Tuning (using all prompts) and our XPROMPT, we introduce three variations - Reversed XPROMPT, Random Prompt and Length Prompt. The Reversed XPROMPT reverses the masked sub-prompt structures in XPROMPT, which essentially uses all the low score prompt tokens and pieces. For Random Prompt, we mask tokens and pieces randomly at the rewind stage. The Length Prompt retrains prompt tuning with the same prompt length of the resulting XPROMPT. The comparison results are shown in Figure 5. It can be seen that our XPROMPT achieves the best performance among them. We also observe that the Reversed XPROMPT performs significantly worse than all other prompt tuning variants, including Random Prompt and Length Prompt. This observation is consistent with our expectation and further validates the existence of the negative prompts. It is worth noting that the Length Prompt performs worse than Random Prompt and Prompt Tuning on average, indicating the effectiveness of our hierarchical structured pruning. The distribution of the importance scores of the prompt tokens is shown in Figure 6 in the appendix.

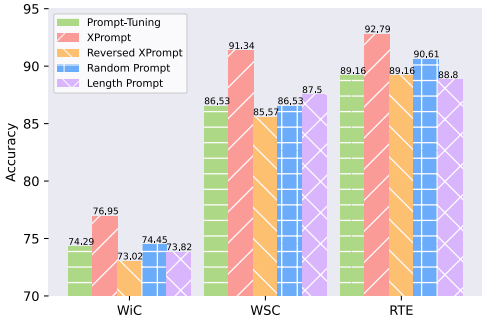

Figure 5: The performance of Prompt-Tuning, XPROMPT, Reversed XPROMPT, Random Prompts and Length Prompt comparison with T5-XL model on three tasks. Among them, Reversed XPROMPT denotes the masked sub-prompt, Random Prompt denotes the randomly masked sub-prompt, and Length Prompt denotes the reserved prompt whose prompt length is the same as XPROMPT.

Model	WiC	WSC	CB	COPA	RTE	Boolq	MultiRC
Fine-Tuning	3×10^9	3×10^9	3×10^9	3×10^9	3×10^9	3×10^9	3×10^9
Prompt-Tuning	40960	40960	40960	40960	40960	40960	40960
XPROMPT	2560	6144	2560	15232	512	29184	27648
Percentage	6.25%	15%	6.25%	37.18%	1.25%	71.25%	67.5%

Table 3: The number of tunable parameters comparison for T5-XL model with 20 prompt tokens. The percentage means the number of tunable parameters in XPROMPT compared to Prompt-Tuning.

7.2 Parameter Efficiency

XPROMPT is more parameter-efficient than Prompt-Tuning. The number of tunable parameters comparison is shown in Table 3. Clearly, Prompt-Tuning is already parameter-efficient, which only needs to tune 0.0014% parameters compared to full model fine-tuning. However, XPROMPT further reduces the tunable parameters in Prompt-Tuning through hierarchical structured pruning. For instance, XPROMPT only tunes 15% and 37.18% parameters compared to Prompt-

Tuning.

Model		WSC	CB	COPA	RTE
T5-Large	Prompt-Tuning	68.26	82.14	76.0	85.19
	Token-level	70.19	91.07	80.0	86.28
	Piece-level	69.23	89.28	79.0	86.64
	XPROMPT	**70.39**	**91.07**	**82.0**	**87.72**
T5-XL	Prompt-Tuning	86.53	91.07	91.0	89.16
	Token-level	89.42	92.85	93.0	92.41
	Piece-level	90.38	92.85	93.0	91.33
	XPROMPT	**91.34**	**92.85**	**95.0**	**92.79**

Table 4: The results of different pruning levels on four SuperGLUE tasks using T5-Large and T5-XL models.

7.3 Granularity of Pruning

Token-level pruning and fine-grained piece-level pruning are both important. To further investigate the effects of the two-level pruning, we conduct extensive ablation experiments on four SuperGLUE tasks, whose results are included in Table 4. In general, both two levels of structured pruning outperform vanilla Prompt-Tuning, demonstrating the effectiveness of both token-level and piece-level pruning. The results also show the existence of sub-prompt structures in trained prompts that can be further optimized. Obviously, XPROMPT outperforms individual one level pruning, which suggests the combination of the two levels of structured pruning further benefits the training of the soft prompts for downstream tasks.

Length	Model	WSC	CB	COPA	RTE
10	Prompt-Tuning	82.69	87.50	87.0	88.44
	XPROMPT	87.50	91.07	93.0	90.61
20	Prompt-Tuning	86.53	91.07	91.0	89.16
	XPROMPT	91.34	92.85	95.0	92.79
100	Prompt-Tuning	89.42	91.07	90.0	89.16
	XPROMPT	91.94	92.85	94.0	92.79

Table 5: The results of different prompt lengths on four SuperGLUE tasks using the T5-XL model.

7.4 Prompt Length

Increasing prompt length (beyond 20) only yields marginal gains for XPROMPT. To explore the effect of prompt length on XPROMPT, we train XPROMPT for the T5-XL model with different prompt lengths in {10, 20, 100}. The results are reported in Table 5. From these results we can see that although prompt length plays an importance role for XPROMPT and Prompt-Tuning, the improvements are limited when increasing the prompt length to beyond 20 tokens. This observation is consistent with the findings in (Lester et al., 2021), and that is why we set the number of prompt tokens to 20 in all our experiments.

Initialization Methods		WSC	COPA
Prompt-Tuning(SampledVocab)		86.53	91.0
XPrompt Initialization	RandomUniform	88.61	93.0
	SampledVocab	91.34	95.0

Table 6: The results of different prompt initialization methods for XPROMPT on two SuperGLUE tasks using T5-XL model.

TransferMethod	WSC	⇔	COPA
TaskTransfer	86.53		92.0
XPromptTransfer$_o$	86.93		95.0
XPromptTransfer	91.40		98.0

Table 7: The results of XPROMPT Transfer on two SuperGLUE tasks using T5-XL model. XPROMPTTransfer$_o$ only uses the resulting prompts of the source task through XPROMPT to initialize the prompts of the target task, without the rewinding phase.

7.5 Prompt Initialization and Transfer

Motivated by the soft prompts transfer approach (SPOT) (Vu et al., 2021), to explore the effect of task transfer and different prompt initialization methods, we introduce a XPROMPT based transfer learning approach - XPROMPT Transfer. It first trains the prompts through XPROMPT on the source task and then uses the learned prompts to initialize the prompts on the target task. More details are provided in Appendix C.

Prompt initialization plays an important role in XPROMPT, and XPROMPT Transfer can lead to performance gains. We compare two sample initialization methods for XPROMPT, including random uniform and sampled vocabulary, the results are shown in Table 6. We observe that sampled vocabulary performs best, and our XPROMPT can also lead to performance gains for the random uniform initialization. Furthermore, we compare our XPROMPT Transfer with the TaskTransfer, which only uses the resulting prompts of the source task to initialize the prompts of the target task, the results are shown in Table 7. We can see that XPROMPT Transfer without rewinding stage outperforms the TaskTransfer, resulting in large performance gains through the pruning and rewinding. These results further validate our hypothesis and the effect of our XPROMPT Transfer.

8 Conclusions

This paper aims to close the large performance gap between prompt tuning and fine-tuning, especially for models of small and moderate scales. By

exploring the lottery ticket hypothesis in the context of prompt tuning, we have proposed a novel hierarchical structured pruning approach, namely XPROMPT, to separate the positive prompts from the negative ones at both token-level and piece-level. Extensive experimental results have demonstrated that XPROMPT yields a more parameter-efficient prompt at an extremely small scale, yet with a competitive performance in effectiveness. Taken as a whole, our work sheds light on the development of more efficient and effective prompt-based learning approaches.

Limitations

Eliminating negative prompt tokens at different granularity levels through hierarchical structured pruning requires rewinding the pruned model at different compression ratios. Therefore, a key question is left under-explored: how to find the optimal compression ratio without trial training, which can automate the training process and improve the efficiency. Moreover, there are other scenarios in prompt tuning that we plan to further investigate, including the multi-task learning scenario (He et al., 2022), out-of-domain (domain shift) scenario (Lester et al., 2021), and prompt ensembling scenario (Lester et al., 2021). We leave these for future research.

Acknowledgments

This research was supported in part by Natural Science Foundation of Beijing (grant number: 4222036) and Huawei Technologies (grant number: TC20201228005).

References

Vamsi Aribandi, Yi Tay, Tal Schuster, Jinfeng Rao, Huaixiu Steven Zheng, Sanket Vaibhav Mehta, Honglei Zhuang, Vinh Q. Tran, Dara Bahri, Jianmo Ni, Jai Prakash Gupta, Kai Hui, Sebastian Ruder, and Donald Metzler. 2021. Ext5: Towards extreme multi-task scaling for transfer learning. *CoRR*, abs/2111.10952.

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, and Dario Amodei. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.

Tianlong Chen, Jonathan Frankle, Shiyu Chang, Sijia Liu, Yang Zhang, Michael Carbin, and Zhangyang Wang. 2021. The lottery tickets hypothesis for supervised and self-supervised pre-training in computer vision models. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2021, virtual, June 19-25, 2021*, pages 16306–16316. Computer Vision Foundation / IEEE.

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, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 2924–2936. Association for Computational Linguistics.

Kevin Clark, Minh-Thang Luong, Quoc V. Le, and Christopher D. Manning. 2020. ELECTRA: pre-training text encoders as discriminators rather than generators. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

Ido Dagan, Oren Glickman, and Bernardo Magnini. 2005. The PASCAL recognising textual entailment challenge. In *Machine Learning Challenges, Evaluating Predictive Uncertainty, Visual Object Classification and Recognizing Textual Entailment, First PASCAL Machine Learning Challenges Workshop, MLCW 2005, Southampton, UK, April 11-13, 2005, Revised Selected Papers*, volume 3944 of *Lecture Notes in Computer Science*, pages 177–190. Springer.

Joe Davison, Joshua Feldman, and Alexander M. Rush. 2019. Commonsense knowledge mining from pretrained models. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, Hong Kong, China, November 3-7, 2019*, pages 1173–1178. Association for Computational Linguistics.

Shrey Desai, Hongyuan Zhan, and Ahmed Aly. 2019. Evaluating lottery tickets under distributional shifts. In *Proceedings of the 2nd Workshop on Deep Learning Approaches for Low-Resource NLP, DeepLo@EMNLP-IJCNLP 2019, Hong Kong, China, November 3, 2019*, pages 153–162. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: pre-training of

deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 4171–4186. Association for Computational Linguistics.

Ning Ding, Shengding Hu, Weilin Zhao, Yulin Chen, Zhiyuan Liu, Hai-Tao Zheng, and Maosong Sun. 2021. Openprompt: An open-source framework for prompt-learning. *CoRR*, abs/2111.01998.

Ning Ding, Yujia Qin, Guang Yang, Fu Wei, Zonghan Yang, Yusheng Su, Shengding Hu, Yulin Chen, Chi-Min Chan, Weize Chen, Jing Yi, Weilin Zhao, Xiaozhi Wang, Zhiyuan Liu, Hai-Tao Zheng, Jianfei Chen, Yang Liu, Jie Tang, Juanzi Li, and Maosong Sun. 2022. Delta tuning: A comprehensive study of parameter efficient methods for pre-trained language models.

Jonathan Frankle and Michael Carbin. 2019. The lottery ticket hypothesis: Finding sparse, trainable neural networks. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net.

Jiaying Gong and Hoda Eldardiry. 2021. Prompt-based zero-shot relation classification with semantic knowledge augmentation. *CoRR*, abs/2112.04539.

Demi Guo, Alexander M. Rush, and Yoon Kim. 2021. Parameter-efficient transfer learning with diff pruning. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, ACL/IJCNLP 2021, (Volume 1: Long Papers), Virtual Event, August 1-6, 2021*, pages 4884–4896. Association for Computational Linguistics.

Yun He, Huaixiu Steven Zheng, Yi Tay, Jai Prakash Gupta, Yu Du, Vamsi Aribandi, Zhe Zhao, YaGuang Li, Zhao Chen, Donald Metzler, Heng-Tze Cheng, and Ed H. Chi. 2022. Hyperprompt: Prompt-based task-conditioning of transformers. *CoRR*, abs/2203.00759.

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. In *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*, volume 97 of *Proceedings of Machine Learning Research*, pages 2790–2799. PMLR.

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 the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2018, New Orleans, Louisiana, USA, June 1-6, 2018, Volume 1 (Long Papers)*, pages 252–262. Association for Computational Linguistics.

Daniel Khashabi, Sewon Min, Tushar Khot, Ashish Sabharwal, Oyvind Tafjord, Peter Clark, and Hannaneh Hajishirzi. 2020. Unifiedqa: Crossing format boundaries with a single QA system. In *Findings of the Association for Computational Linguistics: EMNLP 2020, Online Event, 16-20 November 2020*, volume EMNLP 2020 of *Findings of ACL*, pages 1896–1907. Association for Computational Linguistics.

Brian Lester, Rami Al-Rfou, and Noah Constant. 2021. The power of scale for parameter-efficient prompt tuning. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 7-11 November, 2021*, pages 3045–3059. Association for Computational Linguistics.

Hector J. Levesque, Ernest Davis, and Leora Morgenstern. 2012. The winograd schema challenge. In *Principles of Knowledge Representation and Reasoning: Proceedings of the Thirteenth International Conference, KR 2012, Rome, Italy, June 10-14, 2012*. AAAI Press.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. BART: denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pages 7871–7880. Association for Computational Linguistics.

Xiang Lisa Li and Percy Liang. 2021. Prefix-tuning: Optimizing continuous prompts for generation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, ACL/IJCNLP 2021, (Volume 1: Long Papers), Virtual Event, August 1-6, 2021*, pages 4582–4597. Association for Computational Linguistics.

Chen Liang, Simiao Zuo, Minshuo Chen, Haoming Jiang, Xiaodong Liu, Pengcheng He, Tuo Zhao, and Weizhu Chen. 2021. Super tickets in pre-trained language models: From model compression to improving generalization. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, ACL/IJCNLP 2021, (Volume 1: Long Papers), Virtual Event, August 1-6, 2021*, pages 6524–6538. Association for Computational Linguistics.

Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. 2021a. Pre-train, prompt, and predict: A systematic survey of

prompting methods in natural language processing. *CoRR*, abs/2107.13586.

Xiao Liu, Kaixuan Ji, Yicheng Fu, Zhengxiao Du, Zhilin Yang, and Jie Tang. 2021b. P-tuning v2: Prompt tuning can be comparable to fine-tuning universally across scales and tasks. *CoRR*, abs/2110.07602.

Xiao Liu, Yanan Zheng, Zhengxiao Du, Ming Ding, Yujie Qian, Zhilin Yang, and Jie Tang. 2021c. GPT understands, too. *CoRR*, abs/2103.10385.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized BERT pretraining approach. *CoRR*, abs/1907.11692.

Paul Michel, Omer Levy, and Graham Neubig. 2019. Are sixteen heads really better than one? In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 14014–14024.

Ari S. Morcos, Haonan Yu, Michela Paganini, and Yuandong Tian. 2019. One ticket to win them all: generalizing lottery ticket initializations across datasets and optimizers. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 4933–4943.

Mohammad Taher Pilehvar and José Camacho-Collados. 2019. Wic: the word-in-context dataset for evaluating context-sensitive meaning representations. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 1267–1273. Association for Computational Linguistics.

Guanghui Qin and Jason Eisner. 2021. Learning how to ask: Querying lms with mixtures of soft prompts. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2021, Online, June 6-11, 2021*, pages 5203–5212. Association for Computational Linguistics.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

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. *J. Mach. Learn. Res.*, 21:140:1–140:67.

Alex Renda, Jonathan Frankle, and Michael Carbin. 2020. Comparing rewinding and fine-tuning in neural network pruning. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

Melissa Roemmele, Cosmin Adrian Bejan, and Andrew S. Gordon. 2011. Choice of plausible alternatives: An evaluation of commonsense causal reasoning. In *Logical Formalizations of Commonsense Reasoning, Papers from the 2011 AAAI Spring Symposium, Technical Report SS-11-06, Stanford, California, USA, March 21-23, 2011*. AAAI.

Timo Schick and Hinrich Schütze. 2021. It's not just size that matters: Small language models are also few-shot learners. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2021, Online, June 6-11, 2021*, pages 2339–2352. Association for Computational Linguistics.

Noam Shazeer and Mitchell Stern. 2018. Adafactor: Adaptive learning rates with sublinear memory cost. In *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018*, volume 80 of *Proceedings of Machine Learning Research*, pages 4603–4611. PMLR.

Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. 2014. Deep inside convolutional networks: Visualising image classification models and saliency maps. In *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Workshop Track Proceedings*.

Tu Vu, Brian Lester, Noah Constant, Rami Al-Rfou, and Daniel Cer. 2021. Spot: Better frozen model adaptation through soft prompt transfer. *CoRR*, abs/2110.07904.

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. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 3261–3275.

Chengyu Wang, Jianing Wang, Minghui Qiu, Jun Huang, and Ming Gao. 2021. Transprompt: Towards an automatic transferable prompting framework for few-shot text classification. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 7-11 November, 2021*, pages 2792–2802. Association for Computational Linguistics.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime G. Carbonell, Ruslan Salakhutdinov, and Quoc V. Le. 2019.

Xlnet: Generalized autoregressive pretraining for language understanding. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 5754–5764.

Haonan Yu, Sergey Edunov, Yuandong Tian, and Ari S. Morcos. 2020. Playing the lottery with rewards and multiple languages: lottery tickets in RL and NLP. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

Sheng Zhang, Xiaodong Liu, Jingjing Liu, Jianfeng Gao, Kevin Duh, and Benjamin Van Durme. 2018. Record: Bridging the gap between human and machine commonsense reading comprehension. *CoRR*, abs/1810.12885.

Zexuan Zhong, Dan Friedman, and Danqi Chen. 2021. Factual probing is [MASK]: learning vs. learning to recall. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2021, Online, June 6-11, 2021*, pages 5017–5033. Association for Computational Linguistics.

Kun Zhou, Yuanhang Zhou, Wayne Xin Zhao, Xiaoke Wang, and Ji-Rong Wen. 2020. Towards topic-guided conversational recommender system. In *Proceedings of the 28th International Conference on Computational Linguistics, COLING 2020, Barcelona, Spain (Online), December 8-13, 2020*, pages 4128–4139. International Committee on Computational Linguistics.

A More Results of P-TuningV2

We observe that the performance of Prompt-Tuning and P-Tuning are comparable at the same model scale. Moreover, P-TuningV2 outperforms Prompt-Tuning and P-Tuning on CB, RTE, and Boolq. However, XPROMPT achieves more predominant performances than P-TuningV2 at similar model scales, demonstrating its effectiveness.

Model		CB	RTE	Boolq
GLM-XL 2B	Fine-Tuning†	96.40	90.30	88.30
	P-Tuning†	76.40	85.60	79.70
	P-TuningV2†	**96.40**	90.30	87.00
T5-XL 3B	**XPROMPT**	92.85	**92.79**	**89.00**
GLM-XXL 10B	Fine-Tuning†	98.70	93.10	88.70
	P-Tuning†	98.20	89.90	88.80
	P-TuningV2†	96.40	93.10	88.80
T5-XXL 11B	**XPROMPT**	**100.00**	**94.94**	**90.87**

Table 8: The results on three SuperGLUE tasks for different models and similar model scales. The better results are in bold. Methods with '†' indicate results reported in Liu et al. (2021b). XPROMPT surpasses P-tuningV2 on models with similar scales.

B Token and Piece Importance Score Distribution

Figure 6 and Figure 7 show the distribution of prompt tokens' and prompt token pieces' importance scores on the WSC task. It is clear that most prompt tokens have a low importance score, and only a few prompt tokens have a large importance score. These results further demonstrate our hypothesis that the existence of negative prompts, and their stability.

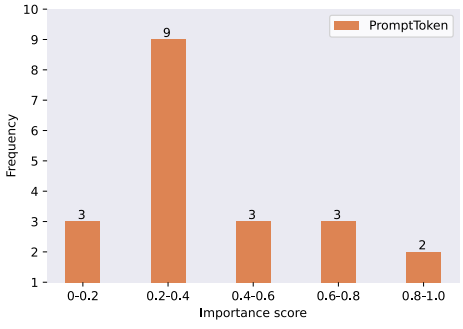

Figure 6: The distribution of prompt tokens' importance scores on WSC task.

C XPROMPT Transfer

As shown in Figure 8, given a source task and a target task, XPROMPT Transfer first trains the prompts through our XPROMPT on the source task

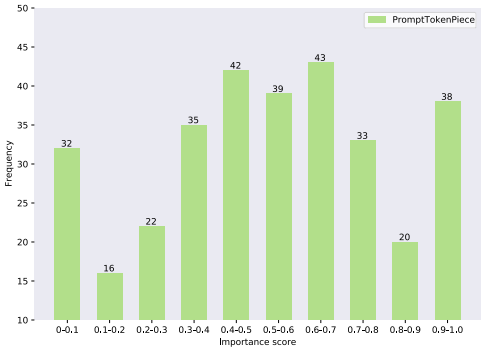

Figure 7: The distribution of prompt token pieces' importance scores on WSC task.

and then uses the resulting prompts to initialize the prompts of the target task, followed by the XPrompt training on the target task. Different from SPOT, we do not use the trained prompts to initialize the prompts for the target task, and our approach can provide more cross tasks information to the prompts. The results of different prompt initialization methods are shown in Table 6, and the results of XPROMPT Transfer are shown in Table 7.

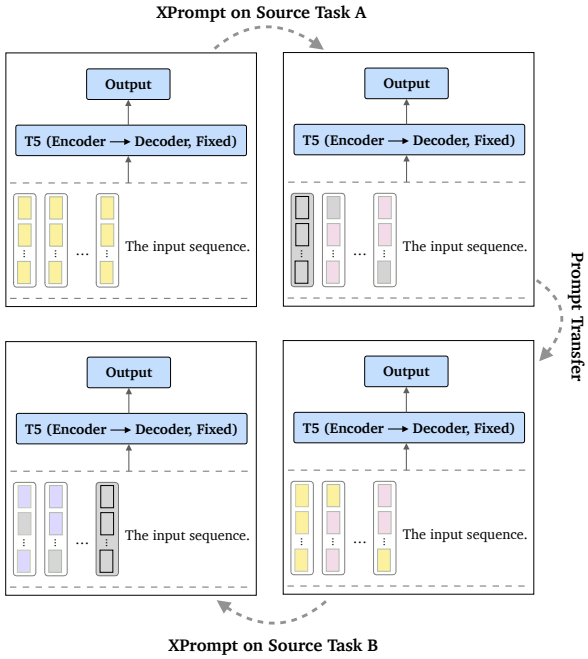

Figure 8: The illustration of XPROMPT Transfer approach. XPROMPT Transfer first trains the prompts through XPROMPT on the source task A and then uses the resulting prompts to initialize the prompts of the target task B, followed by the XPrompt training on the target task B.

D Importance Scores Visualization

Figure 9 and Figure 10 show the original and pruned gradient saliency maps of the importance scores on WSC task. The gray cells in Figure 10 indicate that the prompt tokens and pieces are pruned due to low importance scores, and the remaining ones are the winning tickets.

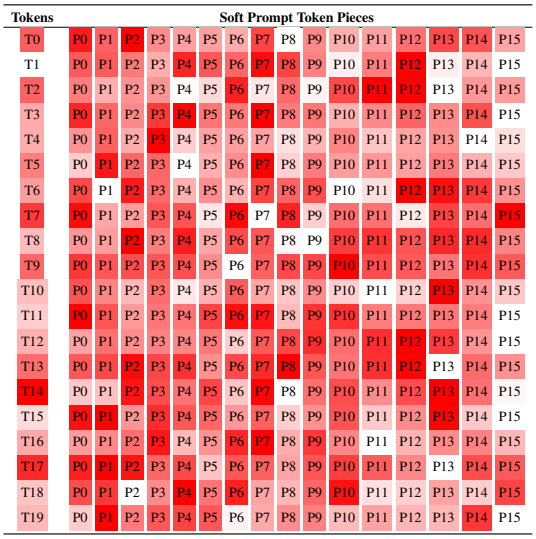

Figure 9: Importance scores visualization on WSC task. Among them, the shade of the red color indicates the level of the importance score, and the darker the color, the higher the importance score of the corresponding structure (token or piece). T_i in first column denotes the i-th prompt token. P_i in each row denotes the i-th prompt token piece.

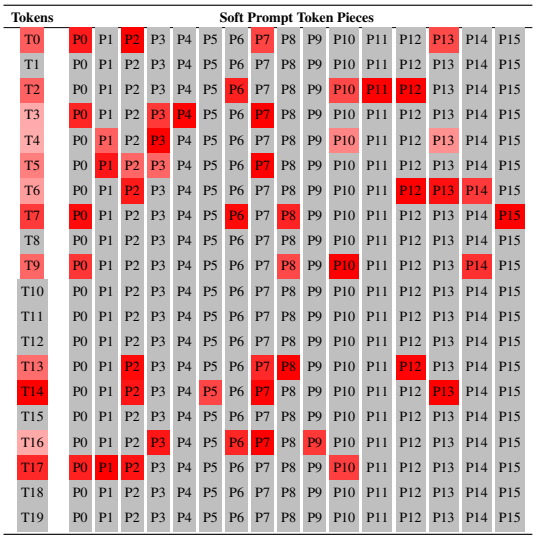

Figure 10: Importance scores visualization after XPROMPT on WSC task. The gray elements indicate that the prompt tokens or pieces are pruned due to low importance scores, and the remaining parts are the positive tokens or positive pieces.

E SuperGLUE Statistics, Metrics and Soft Prompt Templates

SuperGLUE benchmark is a collection of eight challenging language understanding tasks designed to be summarized into a single metric, including question answering (BoolQ (Clark et al., 2019), MultiRC (Khashabi et al., 2018), ReCoRD (Zhang et al., 2018)), textual entailment (RTE (Dagan et al., 2005), CB (Clark et al., 2019)), coreference resolution (WSC (Levesque et al., 2012)), word sense disambiguation (WiC (Pilehvar and Camacho-Collados, 2019)), and causal reasoning (COPA (Roemmele et al., 2011)). Following previous works (Schick and Schütze, 2021; Liu et al., 2021c), we focus on 7 of them, excepting ReCoRD task, since the ReCoRD is also QA tasks. The detailed statistics and metrics are provided in Table 9, and the soft prompt templates and generation verbalizers are provided in Table 10.

Dataset	Train	Dev	Test	Task	Metrics	Text Sources
BoolQ	9427	3270	3245	QA	Acc	Google queries, Wikipedia
CB	250	57	250	NLI	Acc	Various
COPA	400	100	500	QA	Acc	Blogs, Photography encyclopedia
MultiRC	5100	953	1800	QA	$F1_a$	Various
RTE	2500	278	300	NLI	Acc	News, Wikipedia
WiC	6000	638	1400	WSD	Acc	WordNet, VerbNet, Wiktionary
WSC	554	104	146	Coreference	Acc	Fiction books

Table 9: The data statistics and metrics of seven SuperGLUE tasks. WSD stands for word sense disambiguation, NLI is natural language inference, Coreference is coreference resolution, and QA is question answering. Acc is accuracy, and $F1_a$ is F1-score over all answer-options.

Dataset	Task	Soft Template	Generation Verbalizers
BoolQ	QA	{Soft Prompt Tokens} hypothesis: {"placeholder":"text_b", "shortenable":False, "post_processing": lambda x:x+"."} premise: {"placeholder":"text_a"} {"mask"}	"yes" / "no"
CB	NLI	{Soft Prompt Tokens} hypothesis: {"placeholder":"text_b","post_processing": lambda x:x+"."} premise: {"placeholder":"text_a"} {"mask"}	"entailment" / "contradiction" / "neutral"
COPA	QA	{Soft Prompt Tokens} choice1: {"meta":"choice1"} choice2: {"meta":"choice2"} premise: {"placeholder":"text_a"} question: {"meta":"question"} {"mask"}	"choice1" / "choice2"
MultiRC	QA	{Soft Prompt Tokens} question: {"placeholder":"text_b", "shortenable":False} answer: {"meta":"answer", "shortenable":False, "post_processing": lambda x:x+"."} paragraph: {"placeholder":"text_a"} {"mask"}	"yes" / "no"
RTE	NLI	{Soft Prompt Token} sentence1: {"placeholder":"text_a"} sentence2: {"placeholder":"text_b"} {"mask"}	"entailment" / "contradiction"
WiC	WSD	{Soft Prompt Tokens} sentence1: {"placeholder":"text_a"} sentence2: {"placeholder":"text_b"} word: {"meta":"word", "shortenable": False} {"mask"}	"yes" / "no"
WSC	Coreference	{Soft Prompt Tokens} {"placeholder":"text_a"} "{"meta":"span2_text"}" refers to "{"meta":"span1_text"}" or another word ? {"mask"}	"another word" / "span1_text"

Table 10: The soft prompt templates and generation verbalizers for the seven SuperGLUE tasks used in our experiments.