PEMT: Multi-Task Correlation Guided Mixture-of-Experts Enables Parameter-Efficient Transfer Learning

Zhisheng Lin[1], **Han Fu**[1], **Chenghao Liu**[2], **Zhuo Li**[3], **Jianling Sun**[1]
[1]Zhejiang University
{linzhisheng, 11821003, sunjl}@zju.edu.cn
[2]Salesforce Research Asia
chenghao.liu@salesforce.com
[3]State Street Technology (Zhejiang) Ltd.
lizhuo@zju.edu.cn

Abstract

Parameter-efficient fine-tuning (PEFT) has emerged as an effective method for adapting pre-trained language models to various tasks efficiently. Recently, there has been a growing interest in transferring knowledge from one or multiple tasks to the downstream target task to achieve performance improvements. However, current approaches typically either train adapters on individual tasks or distill shared knowledge from source tasks, failing to fully exploit task-specific knowledge and the correlation between source and target tasks. To overcome these limitations, we propose PEMT, a novel parameter-efficient fine-tuning framework based on multi-task transfer learning. PEMT extends the mixture-of-experts (MoE) framework to capture the transferable knowledge as a weighted combination of adapters trained on source tasks. These weights are determined by a gated unit, measuring the correlation between the target and each source task using task description prompt vectors. To fully exploit the task-specific knowledge, we also propose the Task Sparsity Loss to improve the sparsity of the gated unit. We conduct experiments on a broad range of tasks over 17 datasets. The experimental results demonstrate our PEMT yields stable improvements over full fine-tuning, and state-of-the-art PEFT and knowledge transferring methods on various tasks. The results highlight the effectiveness of our method which is capable of sufficiently exploiting the knowledge and correlation features across multiple tasks.

1 Introduction

Fine-tuning pre-trained models (PLMs) has become an effective way to migrate model capabilities to downstream tasks (Devlin et al., 2019; Liu et al., 2019; Raffel et al., 2020). However, training and storing a full copy of the model parameters for each task becomes expensive as the scale of PLM increases. To mitigate this problem, parameter-efficient fine-tuning methods (Houlsby et al., 2019; Schick and Schütze, 2021b; Pfeiffer et al., 2020; Lester et al., 2021; Liu et al., 2023) have been proposed to reduce the number of trainable parameters. Despite their efficiency gains, these methods often sacrifice performance compared to full fine-tuning (Gao et al., 2021a; Hu et al., 2021; Li and Liang, 2021).

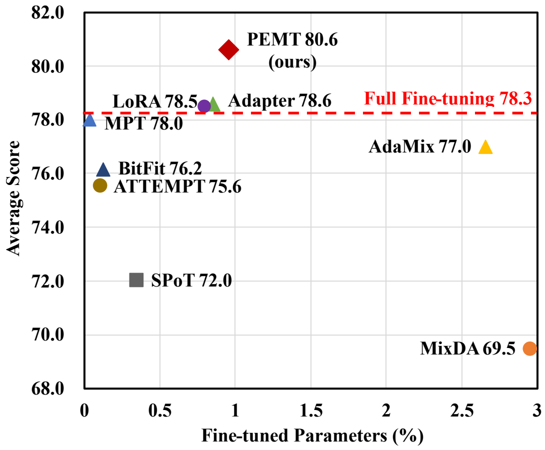

Figure 1: Performance of different parameter-efficient fine-tuning approaches. All results are based on T5-base for a fair comparison. The proposed PEMT achieves significant improvements over all compared methods while fine-tuning only a small number of parameters.

Recent work has proposed to distill knowledge from one or multiple source tasks and adapt it to various downstream target tasks to achieve further improvements (Vu et al., 2022; Asai et al., 2022; Wang et al., 2022c). Despite significant success, there remains a substantial performance gap between these methods and full fine-tuning. The limitations of existing methods can be categorized as follows: (1) Most existing methods primarily focus on utilizing shared knowledge across all source tasks, neglecting task-specific knowledge during adaptation to downstream tasks. (2) Task-specific representations of source and target tasks are typically trained independently, leading to insufficient exploitation of the correlation between them. As a result, the performance of multi-task transfer often lags behind that of distilling knowledge from a single source task. (3) The formulation of source and target tasks may be inconsistent, hindering cross-task adaptation. (4) The knowledge from source tasks is typically used as an initialization, but during fine-tuning, this knowledge may become intertwined with downstream tasks and gradually forgotten.

To mitigate these challenges, we propose PEMT, a parameter efficient fine-tuning framework based on multi-task transfer learning. PEMT comprises two training stages for source task learning and target task adaptation, respectively. (1) In Stage 1, we follow adapter-based tuning (*e.g.*, Adapter (Houlsby et al., 2019) and LoRA (Hu et al., 2021)) to train task-specific adapters on multiple source tasks. We incorporate a sequence of task-specific prompt vectors to distinguish different source tasks and utilize task descriptions to initialize each task prompt effectively. (2) In Stage 2, we train the adapter for the downstream task while incorporating knowledge from source tasks into the model. To enable multi-task transfer learning and prevent knowledge forgetting, we freeze the source adapters and integrate them using a mixture-of-experts architecture (MoE). Instead of relying on a single source task, the knowledge of source tasks is incorporated as a soft combination of all adapters trained during Stage 1. We employ an MoE gated unit to measure the correlation between the target task and each source task, leveraging the task-specific prompt vectors. To ensure the effective utilization of the specific knowledge from source tasks, we introduce the Task Sparsity Loss, encouraging the MoE gate to prioritize the most relevant source expert.

We conduct experiments on 17 NLP datasets involving multiple tasks and domains to evaluate the effectiveness of our approach. On all benchmarks, PEMT achieves an overall improvement of more than 2 points over full fine-tuning and all the compared PEFT methods as shown in Figure 1. Under the few-shot setting, PEMT also proves a significant improvement of 10 points over the compared transfer learning models. Further analysis on the weights of different task experts demonstrates that the model tends to incorporate knowledge from the most relevant source task expert, which explains the efficiency and adaptability of our method.

Overall, this work makes the following contributions:

- We propose PEMT, a two-stage parameter-efficient fine-tuning method facilitating multi-task transfer learning. PEMT captures the transferable knowledge through a combination of adapters trained on source tasks, effectively leveraging task-specific knowledge.

- We propose a task-correlation-based gated unit to determine the weight of each source adapter by measuring the correlation between source and target downstream tasks. To capture interdependency across tasks, we introduce a sequence of task-specific prompt vectors to describe each task.

- Experimental results indicate PEMT consistently outperforms full fine-tuning and state-of-the-art PEFT methods across a broad range of tasks, which demonstrates the robustness and adaptability of our method. PEMT is proven to be also effective for few-shot learning using 4-32 labels.

- We also conduct extensive experiments to analyze how the performance changes under various settings, which provides a clear interpretation for the effectiveness of the proposed method.

2 Related Work

Parameter-Efficient Fine-tuning. Parameter-efficient fine-tuning freezes the original PLM and introduces a small number of additional parameters for fine-tuning. Existing works can be categorized into two classes, adapter-based tuning and prompt-based tuning. Adapter-based methods (Houlsby et al., 2019; Pfeiffer et al., 2020) incorporate a trainable bottleneck module to each transformer layer. Prompt-based tuning (Lester et al., 2021; Schick and Schütze, 2021a; Gao et al., 2021b) prepends continuous or discrete prompt vectors to the input. Recently, some methods are proposed (Pfeiffer et al., 2021; Vu et al., 2022; Wang et al., 2022a; Gururangan et al., 2022; Diao et al., 2023; He et al., 2022; Asai et al., 2022; Wang et al., 2022c; Zhao et al., 2023) to transfer knowledge of trained adapters to downstream tasks.

Multi-Task Transfer Learning. Transferring knowledge from tasks has been proven to be an effective approach (Vu et al., 2020; Aghajanyan et al., 2021; Zhong et al., 2021; Clark et al., 2019b; Singh et al., 2022). Many studies (Sanh et al., 2022; Wei et al., 2021; Wang et al., 2022b; Liu et al., 2022) show the zero-shot or few-shot transferring capabilities of language models through massive multi-task training over a broad range of tasks. However, the corresponding overhead could be enormous. To overcome the issue, some more recent works (Vu et al., 2022; Asai et al., 2022; Wang et al., 2022c; Diao et al., 2023) propose to transfer the knowledge shared by various tasks using parameter-efficient fine-tuning.

Among the related works, AdaMix (Wang et al., 2022a) and MPT (Wang et al., 2022c) are the most relevant methods. Compared to our PEMT, AdaMix trains the representation of source and target tasks independently and fails to sufficiently leverage the interdependency across tasks. MPT learns a single prompt by distilling the shared knowledge while ignoring the rich task-specific information.

3 Approach

Task. Given a set of $\mathcal{K}$ source tasks $\boldsymbol{\mathcal{S}} = \{\mathcal{S}_1, \mathcal{S}_2, \cdots, \mathcal{S}_{\mathcal{K}}\}$ and a set of $\mathcal{M}$ target tasks $\boldsymbol{\mathcal{T}} = \{\mathcal{T}_1, \mathcal{T}_2, \cdots, \mathcal{T}_{\mathcal{M}}\}$, our goal is to capture the knowledge of $\boldsymbol{\mathcal{S}}$ and adapt it to any target task $\mathcal{T}_m \in \boldsymbol{\mathcal{T}}$.

Overview. To sufficiently exploit the task-specific knowledge of each source task, we divide the training process of PEMT into two stages, which are illustrated in Figure 2 and Figure 3 respectively. In the first stage, we follow the vanilla adapter-based (*e.g.*, LoRA (Hu et al., 2021) or Adapter (Houlsby et al., 2019)) to train the source task adapters. For each source task, we freeze the original PLM parameters and inject a task-specific adapter to the feed-forward layer (FFN) for each Transformer layer. Besides, to learn a better representation of each task, we incorporate a task-specific description prompt which is used to measure the correlation between tasks. In Stage 2, we distill the knowledge of source tasks as a weighted combination of the source adapters. The Mixture-of-experts architecture (MoE) is exploited to integrate the frozen source adapters and the MoE gate measures the weight of each source expert using the task prompts learned in Stage 1. The task prompt for the target task is a correlation-based combination of the trainable prompt vectors and the frozen prompts of the source tasks. To adapt to a downstream task, another task adapter is injected *after* the MoE module as shown in Figure 3.

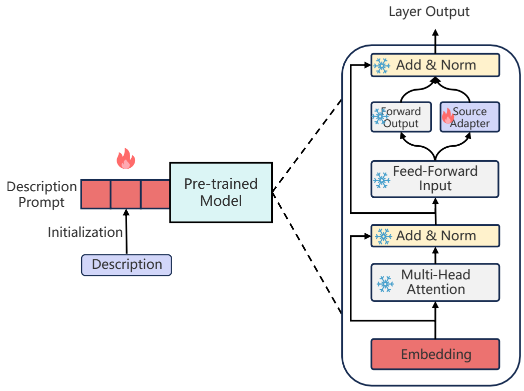

Figure 2: The training process of Stage 1. The task-specific adapters and task representation prompts are trained on multiple source tasks.

3.1 Source Training

The goal of Stage 1 is to capture the task-specific knowledge of each source task. To this end, we fine-tune the PLM on multiple source tasks using adapter-based PEFT methods.

Source Task Adapter. As shown in Figure 2, the task adapter is injected in each transformer layer, which works parallel to the FFN layer to learn the task-specific knowledge. This design is inspired by recent studies (Geva et al., 2021; De Cao et al., 2021; Meng et al., 2022) that FFN captures the major knowledge of the training data. To be specific, a transformer FFN consists of two stacked layers, an up projection layer and a down projection layer. We integrate an adapter module to the FFN using either a parallel Adapter (He et al., 2021) or a LoRA, which works parallel to the up projection layer. The task adapter is implemented as two stacked low-rank matrices for reducing overheads.

Task Description Prompts. We introduce a task description prompt for each source task. The prompt describes the task formulation and is utilized to measure the correlation between tasks. Existing methods (Vu et al., 2022; Asai et al., 2022; Wang et al., 2022c) train the representations of various tasks from scratch independently, which brings a gap between tasks. To address this issue, we propose a simple but effective method to use handcrafted task descriptions as the initialization for the prompt vectors. Concretely, given a source task $\mathcal{S}_k \in \boldsymbol{\mathcal{S}}$, we prepend a trainable prompt matrix $\mathbf{P}_k \in \mathbb{R}^{N_k \times d}$ to the input tokens of PLM, where d is the embedding dimension and N_k denotes the length of the task description (*i.e.* prompt length). The task description is a sentence consisting of a

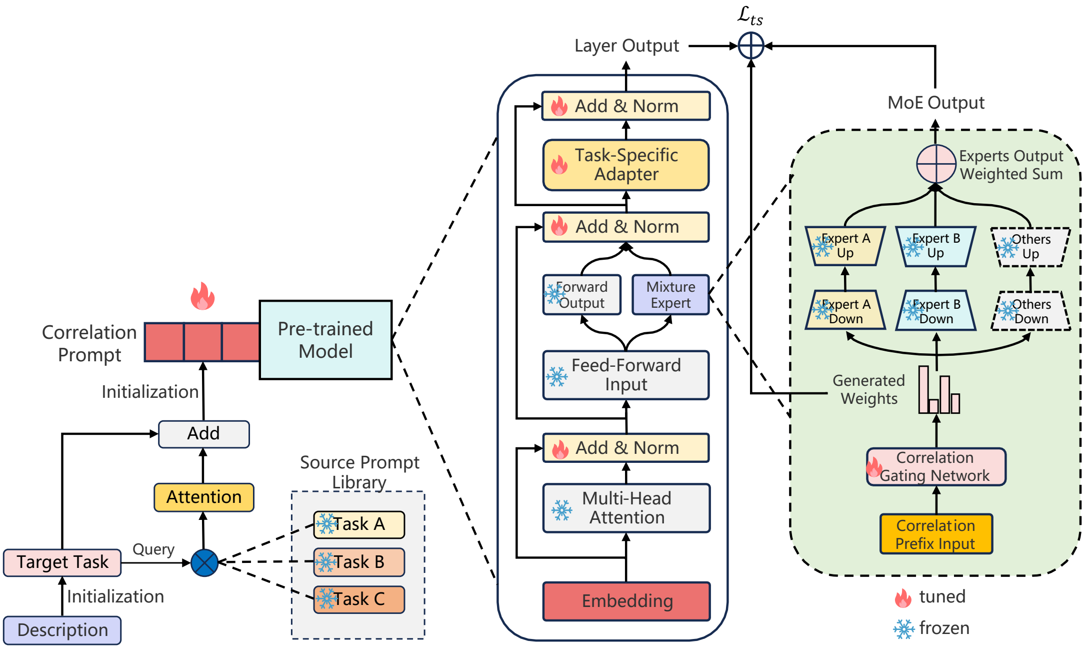

Figure 3: The training process of Stage 2. A MoE module is employed to distill knowledge from source tasks. The source task adapters are used as the experts and combined with a MoE gate which measures the correlation between the target task and each source task. The specific adapter for the target task is injected *after* the MoE module. The task sparsity loss $\mathcal{L}_{ts}$ is incorporated to improve the sparsity of the MoE gate. The task prompt for the target task is a task-correlation-based combination of the trainable prompt vectors and the frozen prompts of the source tasks.

task definition and input-output format based on the distinctive features of various tasks. It should be noted that the description length for different tasks could be different. The details for the template of the task descriptions are provided in Appendix A.

Training on Source Tasks. Both the task adapters and task description prompts are trained following the typical PEFT procedure. There is no particular requirement for the source tasks. To bridge the gap between different tasks, we uniformly formulate all source tasks as text-to-text generation. We follow the format as proposed by (Raffel et al., 2020).

3.2 Target Adaptation

In the second stage, PEMT is guided by the correlation between tasks to utilize the distilled knowledge of all source tasks for adaptation to the downstream target task.

Mixture of Source Task Adapters. We employ a Mixture-of-Experts (MoE) module to combine the source adapters as the transferable knowledge. Instead of only focusing on the shared knowledge, we maintain the task-specific information of each source task during adapting to the downstream task. As illustrated in Figure 3, the task adapters trained in Stage 1 are exploited as the experts in the MoE module. Instead of fine-tuning the source task adapters, we freeze the parameters of the experts to avoid the catastrophic forgetting problem. Formally, the output of the MoE module in the l-th layer is calculated as:

$$\mathbf{H}_e^l = \sum_{k=1}^{\mathcal{K}} w_k^l \cdot \mathbf{E}_k^l, \tag{1}$$

where $\mathbf{E}_k^l$ is the task adapter in the l-th Transformer layer trained on the k-th source task $\mathcal{S}_k$, and $\mathcal{K}$ is the total number of source tasks. w_k^l denotes the weight of $\mathbf{E}_k^l$, which is obtained by the MoE gate, calculated as:

$$w_k^l = \mathrm{softmax}\left(\mathbf{W}_g^l \cdot \mathrm{avg}\left(\mathbf{H}\right)\right)_k, \tag{2}$$

where $\mathbf{W}_g^l \in \mathbb{R}^{d\times\mathcal{K}}$ is a trainable matrix, and avg is an average pooling layer. $\mathbf{H}$ is the prompt matrix for the current target task, which captures the correlation between tasks.

Correlation-Guided Task Prompt. As aforementioned, existing methods train source and target representations independently, which leaves an obstacle to acquire knowledge interdependency across tasks. To exploit the correlation between tasks sufficiently, we propose to incorporate the prompts trained on source tasks into the target adaptation process based on attention mechanism. Following (Vaswani et al., 2017), the attention function $\mathrm{attn}(Q, K, V)$ takes three inputs, query, key, and value respectively. Here, we utilize the target prompt as a query and the source prompts as key and value. Formally, let $\mathbf{Q} = (\mathbf{q}_1, \cdots, \mathbf{q}_T)$ denotes the trainable prompt matrix of the target task. $\mathbf{Q} \in \mathbb{R}^{T \times d}$ is initialized with a task description of T tokens following the same way as in Stage 1 and $\mathbf{q}_t$ denotes the t-th prompt vector. Given the prompt of the k-th source task $\mathbf{P}_k$, the correlation feature between source task $\mathcal{S}_k$ and the target task is obtained as:

$$\mathbf{C}_k = \mathrm{attn}(\mathbf{Q}, \mathbf{P}_k, \mathbf{P}_k) \in \mathbb{R}^{T \times d}. \tag{3}$$

Once the correlation feature for each source task is obtained, we simply add all the correlation information to the original prompt of the target task. Concretely, the final prompt matrix $\mathbf{H} \in \mathbb{R}^{T \times d}$ for the target task is calculated by:

$$\mathbf{H} = \mathbf{Q} + \sum_{k=1}^{\mathcal{K}} \mathbf{C}_k. \tag{4}$$

This design is inspired by the additive compositionality of word embedding (Mikolov et al., 2013), which is proven to be simple but effective according to the experimental results. The prompt for the target task captures the representation information and interdependency across source and target tasks, and is exploited to measure the weight of each source task adapter (Eq 2). It should be noted that all Transformer layers of PEMT share the same $\mathbf{H}$ for the sake of efficiency.

Target Task Adapter. To adapt to the downstream task, we incorporate another task-specific adapter into each Transformer layer. The target task adapter, which is inserted *after* the MoE module, is exploited to mine the knowledge which is not covered by the experts trained on source tasks. The combination of source and target adapters facilitates the model to take advantage of both the rich knowledge learned from each source task and the task-specific knowledge of the target task.

Fine-Tuning on the Target Tasks To sufficiently utilize the knowledge of the source tasks, we propose the Task Sparsity Loss (TSL) to improve the sparsity of the MoE module. The intuition is to ensure the MoE gate assigns a higher priority to the top-1 source task expert by measuring the similarity between specific expert output and the final layer output. Formally, the TSL is defined as:

$$\mathcal{L}_{ts} = -\frac{1}{L\mathcal{K}} \sum_{l=1}^{L} \sum_{k=1}^{\mathcal{K}} w_k^l \cdot \mathrm{sim}(\mathbf{H}_o^l, \mathbf{E}_k^l), \tag{5}$$

where $\mathbf{H}_o^l$ denotes the final hidden state of the l-th Transformer layer, L is the total number of layers, and sim is a similarity score function and we choose cosine similarity in this paper.

Similar to the training process on source tasks, we formulate the target tasks as a text-to-text generation problem. The training objective is to minimize the negative log-likelihood of output $\mathbf{y}$ conditioned on the input text $\mathbf{x}$ and the task prompt $\mathbf{H}$. Finally, the fine-tuning loss on the target task is defined as:

$$\mathcal{L} = -\sum_{j} P(y_j | \mathbf{y}_{<j}; \mathbf{x}, \mathbf{H}) + \alpha \mathcal{L}_{ts}, \tag{6}$$

where α is a hyperparameter to balance the losses.

4 Experiment

We conduct experiments on a comprehensive range of NLP datasets to demonstrate the effectiveness of PEMT. The performance of different methods is compared under both full-dataset and few-shot settings.

4.1 Datasets and Tasks

As in (Wang et al., 2022c), 6 high-resource datasets are used as the as source tasks: MNLI (Williams et al., 2018), QNLI (Demszky et al., 2018), QQP (Wang et al., 2018), SST-2 (Socher et al., 2013), SQuAD (Rajpurkar et al., 2016) and ReCoRD (Zhang et al., 2018). We use other datasets from four benchmarks as target tasks: MultiRC (Khashabi et al., 2018), BoolQ (Clark et al., 2019a), WiC (Pilehvar and Camacho-Collados, 2019), WSC (Levesque et al., 2012) and CB (De Marneffe et al., 2019) from SuperGLUE (Wang et al., 2019); RTE (Giampiccolo et al., 2007), CoLA (Warstadt et al., 2019), STS-B (Cer et al., 2017), MRPC (Dolan and Brockett, 2005) from GLUE (Wang et al., 2018); Natural Questions (NQ) (Kwiatkowski et al., 2019), HotpotQA (HP) (Yang et al., 2018), NewsQA (News) (Trischler et al., 2017), and SearchQA (SQA) (Dunn et al., 2017) from MRQA (Fisch et al., 2019); WinoGrande (Sakaguchi et al., 2021), Yelp-2 (Zhang

Method	GLUE & SuperGLUE									
	STS-B	MRPC	RTE	CoLA	Multi	BoolQ	WiC	WSC	CB	Avg.
FT	89.7	**89.1**	71.9	61.8	72.8	81.1	**70.2**	59.6	85.7	75.8
PT	89.5	68.1	54.7	10.6	58.7	61.7	48.9	51.9	67.9	56.9
BitFit	90.9	86.8	67.6	58.2	74.5	79.6	70.0	59.6	78.6	74.0
Adapter	90.7	85.3	71.9	64.0	75.9	82.5	67.1	**67.3**	85.7	76.7
LoRA	**91.1**	86.8	74.1	61.5	75.2	81.8	69.2	65.4	85.7	76.7
SPoT	90.0	79.7	69.8	57.1	74.0	77.2	48.9	51.9	67.9	68.5
ATTEMPT	89.7	85.7	73.4	57.4	74.4	77.1	66.8	53.8	78.6	73.0
MPT	90.4	**89.1**	79.4	62.4	74.8	79.6	69.0	**67.3**	79.8	76.9
MixDA	90.8	88.2	66.9	60.8	59.2	61.7	48.9	50.0	78.6	67.2
Adamix	91.0	88.2	70.5	58.7	72.9	80.2	63.6	51.9	85.7	73.6
PEMT	$\mathbf{91.1}_{0.22}$	$88.7_{0.40}$	$\mathbf{83.0}_{1.36}$	$\mathbf{67.0}_{2.12}$	$\mathbf{75.5}_{0.36}$	$\mathbf{82.6}_{0.38}$	$68.7_{0.89}$	$\mathbf{67.3}_{0.0}$	$\mathbf{94.1}_{1.68}$	$\mathbf{79.8}_{0.17}$

Table 1: Results on GLUE and SuperGLUE. The metrics are Pearson correlation for STS-B, F1 for MultiRC (Multi), and accuracy for other tasks as evaluation metrics. Our results are averaged over three runs, and subscripts denote standard deviation.

et al., 2015), SciTail (Khot et al., 2018), and PAWS-Wiki (Zhang et al., 2019) from the *Others* benchmark as in (Asai et al., 2022).

Compared Methods We compare PEMT with the state-of-the-art fine-tuning methods: (1) Full fine-tuning (FT), which fine-tunes all parameters of the pre-trained model. (2) Prompt-based tuning, including vanilla prompt tuning (PT) (Lester et al., 2021), SPoT (Vu et al., 2022), ATTEMPT (Asai et al., 2022) and MPT (Wang et al., 2022c). (3) Adapter-based tuning, including vanilla adapter (Houlsby et al., 2019), AdaMix (Wang et al., 2022a) and MixDA (Diao et al., 2023). (4) Other parameter-efficient tuning methods, including LoRA (Hu et al., 2021) and BitFit (Zaken et al., 2022).

4.2 Implementation

Following existing works, we use the publicly available pre-trained T5-Base model (Raffel et al., 2020) with 220M parameters from HuggingFace [1] as the backbone.

Following (Karimi Mahabadi et al., 2021), if a dataset does not have a publicly available test split with annotations, we use the full set of a subset of the developing partition or a subset of the for testing. PEMT is trained on 4 x NVIDIA A800 GPUs. The implementation details and hyper-parameters are listed in Appendix B.

We run all the experiments three times with different random seeds, and report the mean values and standard deviations. Under the few-shot setting, for each number of shots $k \in \{4, 16, 32\}$, we randomly collect k samples from the downstream task data. The random seed is shared by all compared methods for a fair comparison.

4.3 Results

Full Data. Experimental results in Table 1 and 2 show that PEMT significantly outperforms full fine-tuning and all other parameter-efficient tuning methods. As observed from Table 1, PEMT establishes the new state-of-the-art results for parameter-efficient fine-tuning on GLUE and SuperGLUE. According to the results, Adapter and MPT are the most competitive methods, while our method yields an improvement of 2.75% and 2.91%. Especially, On CB task, the improvement comes to 13.06% and 7.16%. On RTE task, PEMT outperforms all other methods with over 10 points, which illustrates the capability of knowledge transferring of our method.

Table 2 shows the performance of different methods on MRQA and *Others* benchmark. Compared with GLUE and SuperGLUE, the data sizes of these two datasets are larger, and the contexts of the samples are longer. Due to these complexities, the performance of previous PEFT methods is significantly inferior to full fine-tuning. From the results, PEMT successfully outperforms full fine-tuning on these datasets, suggesting the stability and robustness of PEMT across different data sizes and context lengths.

Few-shot. Following prior works, we conduct few-shot experiments on GLUE and SuperGLUE benchmark to measure the generalization of PEMT to new tasks with only a few training examples available ($k \in \{4, 16, 32\}$). Table 3 shows the

[1] https://huggingface.co/

Method	**MRQA**					**Others**				
	NQ	**HP**	**SQA**	**News**	**Avg.**	**WG**	**Yelp**	**SciTail**	**PAWS**	**Avg.**
FT	**75.1**	77.5	81.1	65.2	74.7	61.9	96.7	95.8	94.1	87.1
PT	67.9	72.9	75.7	61.1	69.4	49.6	95.1	87.9	55.8	72.1
BitFit	70.7	75.5	77.7	64.1	72.0	57.2	94.7	94.7	92.0	84.7
Adapter	74.2	77.6	81.4	65.6	74.7	59.2	96.9	94.5	94.3	86.2
LoRA	73.9	77.1	80.1	64.9	74.0	60.2	96.4	94.5	94.2	86.3
SPoT	68.2	74.8	75.3	58.2	69.1	50.4	95.4	91.2	91.1	82.0
ATTEMPT	70.4	75.2	77.3	62.8	71.4	57.6	96.7	93.1	92.1	84.9
MPT	70.4	75.2	77.3	62.8	72.8	56.5	96.4	95.5	93.5	85.5
MixDA	71.2	76.1	78.3	63.9	72.4	55.2	95.7	50.8	82.7	71.1
Adamix	73.2	77.5	80.4	65.2	74.1	59.8	96.6	96.0	94.0	86.6
PEMT	**75.1**$_{0.04}$	**78.3**$_{0.10}$	**81.8**$_{0.09}$	**65.9**$_{0.12}$	**75.3**$_{0.02}$	**62.3**$_{0.08}$	**97.0**$_{0.06}$	**96.9**$_{0.69}$	**94.3**$_{0.08}$	**87.6**$_{0.18}$

Table 2: Results on MRQA and the *Others* benchmark. Our results are averaged over three runs and subscripts indicate standard deviation.

k**-shot**	**Method**	**GLUE & SuperGLUE**									
		STS-B	**MRPC**	**RTE**	**CoLA**	**Multi**	**BoolQ**	**WiC**	**WSC**	**CB**	**Avg.**
4	PT	88.8	68.1	56.3	27.4	61.8	61.6	51.2	60.4	53.5	58.8
	MPT	89.1	68.1	62.6	34.8	62.2	62.2	52.9	**67.3**	73.6	63.6
	PEMT	**89.2**	**78.4**	**64.0**	**44.7**	**72.0**	**71.0**	**62.1**	44.2	**78.6**	**67.1**
16	PT	87.8	68.1	54.7	28.5	60.3	61.9	48.9	44.2	63.5	57.5
	MPT	89.1	70.1	64.8	32.1	64.5	63.3	49.8	**67.3**	78.6	64.4
	PEMT	**89.8**	**86.8**	**69.8**	**43.4**	**72.4**	**74.0**	**66.5**	44.2	**82.1**	**69.9**
32	PT	87.5	68.1	54.7	23.2	59.2	61.7	52.6	**67.3**	67.8	60.2
	MPT	89.7	74.5	59.7	30.8	63.3	68.9	53.9	**67.3**	82.1	65.6
	PEMT	**89.8**	**86.3**	**71.9**	**45.5**	**72.2**	**74.4**	**61.8**	51.9	**85.7**	**71.1**

Table 3: Few-shot learning results on GLUE with 4, 16, and 32 training examples.

results. With limited data resources, our method still yields a significant improvement, especially on some tasks such as WiC, MultiRC, CoLA, and MRPC. Another interesting observation is that the improvement of PEMT over baselines becomes more pronounced as the number of training samples increases. This further underscores that the task-shared knowledge of MPT gradually fades during the training process of downstream tasks when more training data is provided. In contrast, PEMT freezes the source task adapters, which not only preserves shared knowledge to the greatest extent possible but also sufficiently exploits the associations and distinctions across various tasks.

5 Analysis

We conduct further analysis to investigate the effectiveness of different components of PEMT.

Weights of Source Adapters. In order to explore how the weights of source experts change on various target tasks, we collect the outputs of the MoE gate and visualize them through histograms as shown in Figure 4. As observed, there are obvious tendencies and priorities in the weight distribution. For GLUE and SuperGLUE benchmarks, the knowledge of the MNLI plays a dominant role, with a weighting of more than 50% of all tasks. The contributions of some individual tasks are close to 0 under the constraint of Task Sparsity Loss. Contrastively, the distribution of weights on MRQA is totally different, where the two tasks SQuAD and ReCorD account for about 80% of the weights. The reason is that all the three datasets MRQA, SQuAD and ReCorD belong to the Q&A category, which also indicates the correlation guided MoE module and the task sparsity loss effectively work as expected.

Task Description Prompts. As introduced in Section 3.1, We initialize the task prompt with a sentence of task description. To measure the effectiveness of this method in maintaining consistency in task representation, we replace it with a randomly initialized prompt and keep the prompt length the same. As shown in Table 5 (Row 2), the averaged score on the two benchmarks decreases by 0.8% without initialization with task descriptions.

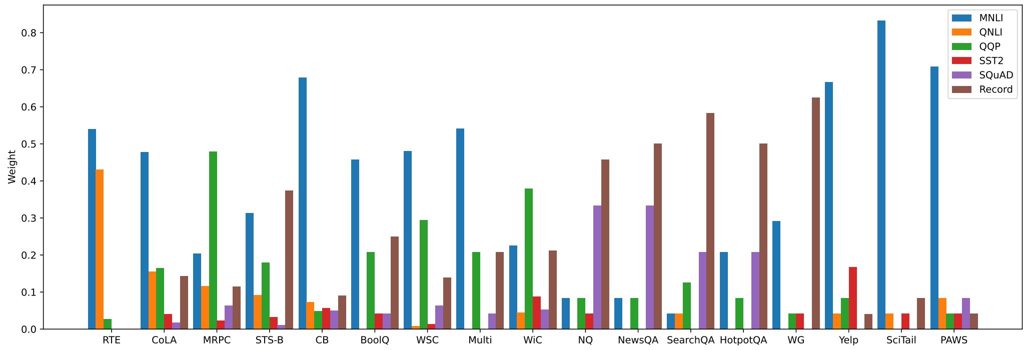

Figure 4: The source expert weight distribution in GLUE, SuperGLUE, MRQA and *Others* benchmarks.

Number of Source Task	STS-B	MRPC	RTE	CoLA	Multi	BoolQ	WiC	WSC	CB	Avg.
1	91.3	86.8	80.6	66.3	76.1	81.5	65.4	67.3	92.9	78.7
2	91.0	87.8	83.5	64.3	75.4	82.1	68.3	67.3	92.9	79.2
4	91.3	89.2	82.0	64.2	74.9	82.6	68.7	67.3	92.9	79.2
6	91.1	89.7	83.0	67.0	75.5	82.6	68.7	67.3	94.1	79.8

Table 4: Average scores on GLUE and SuperGLUE benchmark with different number of source tasks.

No.	Ablation	Avg. Score
1	PEMT with LoRA	**79.8**
2	w/o description	79.0
3	w/o correlation	78.3
4	w/o correlation and MoE	76.6
5	PEMT with Adapter	79.0

Table 5: Results of ablation studies on GLUE and SuperGLUE benchmark.

Correlation Guided Task Prompt. We conduct experiments to evaluate the effectiveness of task correlation features in facilitating the model to select the optimal source expert. We remove the entire prompt module in both source task training and target adaptation while maintaining the MoE module. For the MoE gate, we use an average pooling on the hidden states of the previous FFN layer as input. The ablation study in Table 5 (Row 3) shows that task correlation features produce a 1.5% average performance improvement.

Mixture-of-Source-Adapters. We further investigate the effectiveness of the source adapters on target adaptation. To this end, we remove both the target prompt and the source adapters, while only maintaining the task-specific adapter *after* each FFN layer. This change degenerates the model to the simple variant of Adapter which inserts an adapter module into each multi-head attention and FFN layer. We evaluate the performance on target adaptation without training on source tasks. The results in Table 5 (Row 4) show that, without the task prompt and source adapters, the performance drops sharply by 3.2% on average.

5.1 The Number of Source Task

To substantiate the scalability of PEMT, we also investigate how the performance changes when different numbers of source tasks are used in Stage 1. As shown in Table 4, compared to MixDA, PEMT exhibits a gradual improvement as the number of source tasks increases, which is different from the results of existing methods (Diao et al., 2023). This observation suggests the capability of PEMT to sufficiently capture the commonalities and differences among various tasks, which demonstrates a certain degree of continual learning proficiency.

6 Conclusion

In this paper, we propose PEMT, a new parameter-efficient fine-tuning framework that is capable of adapting the knowledge from multiple tasks to the downstream target tasks. PEMT is facilitated with the correlation features between tasks and sufficiently leverages the task-specific knowledge of source tasks with prompt tuning and the mixture-of-experts architecture. We also introduce novel methods to improve prompt initialization and model sparsity. Experiments are conducted on a comprehensive range of datasets involving multiple tasks and domains and the results demonstrate PEMT significantly outperforms existing SOTA methods.

Limitations

The model's inference latency rises proportionally with the number of experts, prompting the necessity to identify a stable reparameterization for merging the weights of multiple experts or to explore a reliable pruning method. Additionally, the entire framework involves a two-stage supervised learning process. Though maintaining efficient at inference, the two-stage architecture incurs both training overhead and data costs, and also introduces potential risks of data leakage or model attacks.

References

Armen Aghajanyan, Anchit Gupta, Akshat Shrivastava, Xilun Chen, Luke Zettlemoyer, and Sonal Gupta. 2021. Muppet: Massive multi-task representations with pre-finetuning. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 5799–5811.

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.

Daniel Cer, Mona Diab, Eneko E Agirre, Iñigo Lopez-Gazpio, and Lucia Specia. 2017. Semeval-2017 task 1: Semantic textual similarity multilingual and cross-lingual focused evaluation. In *The 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 1–14.

Christopher Clark, Kenton Lee, Ming-Wei Chang, Tom Kwiatkowski, Michael Collins, and Kristina Toutanova. 2019a. 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.

Kevin Clark, Minh-Thang Luong, Urvashi Khandelwal, Christopher D Manning, and Quoc Le. 2019b. Bam! born-again multi-task networks for natural language understanding. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5931–5937.

Nicola De Cao, Wilker Aziz, and Ivan Titov. 2021. Editing factual knowledge in language models. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 6491–6506.

Marie-Catherine De Marneffe, Mandy Simons, and Judith Tonhauser. 2019. The commitmentbank: Investigating projection in naturally occurring discourse. In *proceedings of Sinn und Bedeutung*, volume 23, pages 107–124.

Dorottya Demszky, Kelvin Guu, and Percy Liang. 2018. Transforming question answering datasets into natural language inference datasets. *arXiv preprint arXiv:1809.02922*.

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.

Shizhe Diao, Tianyang Xu, Ruijia Xu, Jiawei Wang, and Tong Zhang. 2023. Mixture-of-domain-adapters: Decoupling and injecting domain knowledge to pre-trained language models' memories. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2023, Toronto, Canada, July 9-14, 2023*, pages 5113–5129. Association for Computational Linguistics.

William B. Dolan and Chris Brockett. 2005. Automatically constructing a corpus of sentential paraphrases. In *Proceedings of the Third International Workshop on Paraphrasing (IWP2005)*.

Matthew Dunn, Levent Sagun, Mike Higgins, V. Ugur Güney, Volkan Cirik, and Kyunghyun Cho. 2017. Searchqa: A new q&a dataset augmented with context from a search engine. *CoRR*, abs/1704.05179.

Adam Fisch, Alon Talmor, Robin Jia, Minjoon Seo, Eunsol Choi, and Danqi Chen. 2019. MRQA 2019 shared task: Evaluating generalization in reading comprehension. In *Proceedings of the 2nd Workshop on Machine Reading for Question Answering, MRQA@EMNLP 2019, Hong Kong, China, November 4, 2019*, pages 1–13. Association for Computational Linguistics.

Tianyu Gao, Adam Fisch, and Danqi Chen. 2021a. Making pre-trained language models better few-shot learners. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 3816–3830, Online. Association for Computational Linguistics.

Tianyu Gao, Adam Fisch, and Danqi Chen. 2021b. Making pre-trained language models better few-shot learners. In *Joint Conference of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference* *on Natural Language Processing, ACL-IJCNLP 2021*, pages 3816–3830. Association for Computational Linguistics (ACL).

Mor Geva, Roei Schuster, Jonathan Berant, and Omer Levy. 2021. Transformer feed-forward layers are key-value memories. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 5484–5495.

Danilo Giampiccolo, Bernardo Magnini, Ido Dagan, and William B Dolan. 2007. The third pascal recognizing textual entailment challenge. In *Proceedings of the ACL-PASCAL workshop on textual entailment and paraphrasing*, pages 1–9.

Suchin Gururangan, Mike Lewis, Ari Holtzman, Noah A Smith, and Luke Zettlemoyer. 2022. Demix layers: Disentangling domains for modular language modeling. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5557–5576.

Junxian He, Chunting Zhou, Xuezhe Ma, Taylor Berg-Kirkpatrick, and Graham Neubig. 2021. Towards a unified view of parameter-efficient transfer learning. In *International Conference on Learning Representations*.

Yun He, Steven Zheng, Yi Tay, Jai Gupta, Yu Du, Vamsi Aribandi, Zhe Zhao, YaGuang Li, Zhao Chen, Donald Metzler, et al. 2022. Hyperprompt: Prompt-based task-conditioning of transformers. In *International Conference on Machine Learning*, pages 8678–8690. PMLR.

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 *International Conference on Machine Learning*, pages 2790–2799. PMLR.

Edward J Hu, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, Weizhu Chen, et al. 2021. Lora: Low-rank adaptation of large language models. In *International Conference on Learning Representations*.

Rabeeh Karimi Mahabadi, James Henderson, and Sebastian Ruder. 2021. Compacter: Efficient low-rank hypercomplex adapter layers. *Advances in Neural Information Processing Systems*, 34:1022–1035.

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, Volume 1 (Long Papers)*, pages 252–262, New Orleans, Louisiana. Association for Computational Linguistics.

Tushar Khot, Ashish Sabharwal, and Peter Clark. 2018. Scitail: A textual entailment dataset from science question answering. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32.

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, Kristina Toutanova, Llion Jones, Matthew Kelcey, Ming-Wei Chang, Andrew M. Dai, Jakob Uszkoreit, Quoc Le, and Slav Petrov. 2019. Natural questions: A benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, 7:452–466.

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*, pages 3045–3059.

Hector Levesque, Ernest Davis, and Leora Morgenstern. 2012. The winograd schema challenge. In *Thirteenth international conference on the principles of knowledge representation and reasoning*.

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 (Volume 1: Long Papers)*, pages 4582–4597.

Haokun Liu, Derek Tam, Mohammed Muqeeth, Jay Mohta, Tenghao Huang, Mohit Bansal, and Colin A Raffel. 2022. Few-shot parameter-efficient fine-tuning is better and cheaper than in-context learning. *Advances in Neural Information Processing Systems*, 35:1950–1965.

Xiao Liu, Yanan Zheng, Zhengxiao Du, Ming Ding, Yujie Qian, Zhilin Yang, and Jie Tang. 2023. Gpt understands, too. *AI Open*.

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. *arXiv preprint arXiv:1907.11692*.

Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. 2022. Locating and editing factual associations in gpt. *Advances in Neural Information Processing Systems*, 35:17359–17372.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

Jonas Pfeiffer, Aishwarya Kamath, Andreas Rücklé, Kyunghyun Cho, and Iryna Gurevych. 2021. Adapterfusion: Non-destructive task composition for transfer learning. In *16th Conference of the European Chapter of the Associationfor Computational*

Linguistics, EACL 2021, pages 487–503. Association for Computational Linguistics (ACL).

Jonas Pfeiffer, Ivan Vulić, Iryna Gurevych, and Sebastian Ruder. 2020. Mad-x: An adapter-based framework for multi-task cross-lingual transfer. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7654–7673.

Mohammad Taher Pilehvar and Jose Camacho-Collados. 2019. Wic: the word-in-context dataset for evaluating context-sensitive meaning representations. In *Proceedings of NAACL-HLT*, pages 1267–1273.

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. *The Journal of Machine Learning Research*, 21(1):5485–5551.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. SQuAD: 100,000+ questions for machine comprehension of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2383–2392, Austin, Texas. Association for Computational Linguistics.

Keisuke Sakaguchi, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. 2021. Winogrande: An adversarial winograd schema challenge at scale. *Communications of the ACM*, 64(9):99–106.

Victor Sanh, Albert Webson, Colin Raffel, Stephen H Bach, Lintang Sutawika, Zaid Alyafeai, Antoine Chaffin, Arnaud Stiegler, Teven Le Scao, Arun Raja, et al. 2022. Multitask prompted training enables zero-shot task generalization. In *ICLR 2022-Tenth International Conference on Learning Representations*.

Timo Schick and Hinrich Schütze. 2021a. Exploiting cloze-questions for few-shot text classification and natural language inference. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 255–269.

Timo Schick and Hinrich Schütze. 2021b. 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*, pages 2339–2352.

Janvijay Singh, Fan Bai, and Zhen Wang. 2022. Frustratingly simple entity tracking with effective use of multi-task learning models. *arXiv preprint arXiv:2210.06444*.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1631–1642, Seattle, Washington, USA. Association for Computational Linguistics.

Adam Trischler, Tong Wang, Xingdi Yuan, Justin Harris, Alessandro Sordoni, Philip Bachman, and Kaheer Suleman. 2017. NewsQA: A machine comprehension dataset. In *Proceedings of the 2nd Workshop on Representation Learning for NLP*, pages 191–200, Vancouver, Canada. Association for Computational Linguistics.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

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.

Tu Vu, Tong Wang, Tsendsuren Munkhdalai, Alessandro Sordoni, Adam Trischler, Andrew Mattarella-Micke, Subhransu Maji, and Mohit Iyyer. 2020. Exploring and predicting transferability across nlp tasks. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7882–7926.

Alex Wang, Yada Pruksachatkun, Nikita Nangia, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. 2019. Superglue: A stickier benchmark for general-purpose language understanding systems. *Advances in neural information processing systems*, 32.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. 2018. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 353–355, Brussels, Belgium. Association for Computational Linguistics.

Yaqing Wang, Sahaj Agarwal, Subhabrata Mukherjee, Xiaodong Liu, Jing Gao, Ahmed Hassan Awadallah, and Jianfeng Gao. 2022a. AdaMix: Mixture-of-adaptations for parameter-efficient model tuning. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 5744–5760, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Yizhong Wang, Swaroop Mishra, Pegah Alipoormolabashi, Yeganeh Kordi, Amirreza Mirzaei, Atharva Naik, Arjun Ashok, Arut Selvan Dhanasekaran, Anjana Arunkumar, David Stap, Eshaan Pathak, Giannis Karamanolakis, Haizhi Lai, Ishan Purohit, Ishani Mondal, Jacob Anderson, Kirby Kuznia, Krima Doshi, Kuntal Kumar Pal, Maitreya Patel, Mehrad Moradshahi, Mihir Parmar, Mirali Purohit, Neeraj Varshney, Phani Rohitha Kaza, Pulkit Verma, Ravsehaj Singh Puri, Rushang Karia, Savan Doshi, Shailaja Keyur Sampat, Siddhartha Mishra, Sujan Reddy A, Sumanta Patro, Tanay Dixit, and Xudong Shen. 2022b. Super-NaturalInstructions: Generalization via declarative instructions on 1600+ NLP tasks. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 5085–5109, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Zhen Wang, Rameswar Panda, Leonid Karlinsky, Rogerio Feris, Huan Sun, and Yoon Kim. 2022c. Multitask prompt tuning enables parameter-efficient transfer learning. In *The Eleventh International Conference on Learning Representations*.

Alex Warstadt, Amanpreet Singh, and Samuel R. Bowman. 2019. Neural network acceptability judgments. *Transactions of the Association for Computational Linguistics*, 7:625–641.

Jason Wei, Maarten Bosma, Vincent Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M Dai, and Quoc V Le. 2021. Finetuned language models are zero-shot learners. In *International Conference on Learning Representations*.

Adina Williams, Nikita Nangia, and Samuel Bowman. 2018. A broad-coverage challenge corpus for sentence understanding through inference. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1112–1122, New Orleans, Louisiana. Association for Computational Linguistics.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. 2018. HotpotQA: A dataset for diverse, explainable multi-hop question answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2369–2380, Brussels, Belgium. Association for Computational Linguistics.

Elad Ben Zaken, Yoav Goldberg, and Shauli Ravfogel. 2022. Bitfit: Simple parameter-efficient fine-tuning for transformer-based masked language-models. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 1–9.

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. *arXiv preprint arXiv:1810.12885*.

Xiang Zhang, Junbo Zhao, and Yann LeCun. 2015. Character-level convolutional networks for text classification. *Advances in neural information processing systems*, 28.

Yuan Zhang, Jason Baldridge, and Luheng He. 2019. PAWS: Paraphrase adversaries from word scrambling. 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 1298–1308, Minneapolis, Minnesota. Association for Computational Linguistics.

Hao Zhao, Jie Fu, and Zhaofeng He. 2023. Prototype-based hyperadapter for sample-efficient multi-task tuning. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 4603–4615.

Ruiqi Zhong, Kristy Lee, Zheng Zhang, and Dan Klein. 2021. Adapting language models for zero-shot learning by meta-tuning on dataset and prompt collections. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 2856–2878.

A Task Description Details

We designed task descriptions based on the distinctive features of various tasks. Take MNLI task as an example, we use a description "Given a premise sentence and a hypothesis sentence, predict whether the premise entails the hypothesis, contradicts the hypothesis, or neither" to initialize the continuous prompt vectors prepended to the input. The descriptions for all tasks are shown as Table 8.

Param	Value
Optimizer	AdamW
Learning rate	5e-4
Batch size	128
Warmup steps	500
Expert dimension	64
Training epochs	5
Learning rate schedule	linear decay

Table 6: Stage 1 training: experimental setup.

Param	Value
Optimizer	AdamW
Learning rate	{6e-4, 1e-3}
Batch size	{64, 128}
Expert dimension	64
Training epochs	20
Seed	{42, 1024, 4096}
MoE loss factor	0.1
Learning rate schedule	linear decay

Table 7: Stage 2 training: experimental setup.

B Implementation Details

We use down projection dimension $r = 64$ in both source training and target adaptation. For source training, we train PEMT on each source task for 5 epochs. For target adaptation, we train all of the baselines for 20 epochs on small datasets with less than 10k examples, 10 epochs on medium size data with more than 10k examples, and 5 epochs on MRQA datasets. We limit the maximum training data number of Yelp-2 to be 100k samples. We run inferences on the test data using the model with the best development performance. We set the maximum token length to be 512 for MRQA datasets, 348 for MultiRC and 256 for all of other datasets. We set the maximum length of the input to be 256, 256, 512, 256 for GLUE, SuperGLUE, MRQA 2019, and *Others* task set, respectively. We set the maximum length of input to be 348 for MultiRC. The details for training parameters are shown in Table 6 and Table 7

Task	Description
QNLI	Given a question and a context sentence, determine whether the context sentence contains the answer to the question.
MNLI	Given a premise sentence and a hypothesis sentence, predict whether the premise entails the hypothesis (entailment), contradicts the hypothesis (contradiction), or neither (neutral).
QQP	Given a pair of sentences, determine if the two sentences are semantically equivalent or not.
SST-2	Given a sentence, predict whether a given sentence expresses a positive or negative sentiment.
ReCoRD	Given a passage and a cloze-style question about the article in which one entity is masked out, predict the masked out entity from a list of possible entities in the provided passage.
SQuAD	Given an article and a corresponding question about the article, answer the question accurately based on the provided context in the articles.
CoLA	Given a sentence, judge the grammatical acceptability of the sentence.
RTE	Given a premise sentence and a hypothesis sentence, determine whether the hypothesis can be inferred from the premise.
MRPC	Given a pair of sentences, determine whether the two sentences are semantically equivalent or not.
STS-B	Given a pair of sentences, measure the degree of semantic similarity or relatedness between pairs of sentences.
CB	Given a premise and a hypothesis, determine the type and strength of the commitment being expressed.
WiC	Given a target word and a pair of sentences, determine if a given target word in a sentence has the same meaning in two different contexts.
WSC	Given a set of sentences that contain an ambiguous pronoun, determine the referent of the ambiguous pronoun based on the context provided.
BoolQ	Given a question and a paragraph, determine if a given question can be answered with a simple "true" or "false" based on a given passage of text.
Multi	Given a passage of text and a set of related multiple-choice questions, where each question is accompanied by several answer choices, select the correct answer choice for each question based on the information provided in the passage.
MRQA	Given an article and a corresponding question about the article, answer the question accurately based on the provided context in the articles.
SciTail	Given a premise and a hypothesis, classify the relationship between the premise and the hypothesis as entail or neutral.
Yelp	Given a Yelp sentence, predict the sentiment polarity (positive or negative) of customer reviews from the Yelp dataset.
WG	Given a sentence and two options, choose the right option for a given sentence which requires commonsense reasoning.
PAWS	Given a pair of sentence, where one sentence is a paraphrase of the other. Determine if the given sentence pair is a paraphrase or not.

Table 8: Tasks descriptions for prompt Initialization