Prompt tuning is a parameter-efficient approach to adapting pre-trained language models to downstream tasks. Although prompt tuning has been shown to match the performance of full model tuning when training data is sufficient, it tends to struggle in few-shot learning settings. In this paper, we present Multi-task Pre-trained Modular Prompt (MP²) to boost prompt tuning for few-shot learning. MP² is a set of combinable prompts pre-trained on 38 Chinese tasks. On downstream tasks, the pre-trained prompts are selectively activated and combined, leading to strong compositional generalization to unseen tasks. To bridge the gap between pre-training and fine-tuning, we formulate upstream and downstream tasks into a unified (masked) language modeling ((M)LM for short) task, one can reuse the pre-trained (M)LM head instead of training a randomly initialized classification head to solve tasks with limited data. However, prompt-based learning usually fine-tunes all the parameters of the PTM for each downstream task, which can be computationally expensive and deployment-inefficient, especially for large PTMs such as GPT-3 (Brown et al., 2020).

Recently, much effort has been devoted to parameter-efficient prompt tuning (Li and Liang, 2021; Lester et al., 2021; Liu et al., 2021c; Sun et al., 2022c), which only learns a small number of soft prompt parameters while keeping the main body of the PTM untouched. In contrast to full model tuning, prompt tuning can get specialized models for specific tasks by simply attaching task-specific prompts, and therefore is highly efficient for serving different tasks. Though it has been demonstrated that prompt tuning can match the performance of full model tuning when training data is sufficient (Lester et al., 2021), the soft prompt cannot be well trained from scratch in few-shot learning settings (Gu et al., 2021) because the randomly initialized soft prompt introduces a new gap between pre-training and fine-tuning.

Figure 1: MP² achieves fast adaptation to downstream tasks through three steps: (1) Self-supervised pre-training on large-scale unlabeled data. (2) Pre-training modular prompts and the corresponding router with multi-task learning. (3) A subset of prompts is activated and tuned for adaptation to downstream tasks.
To bridge the gap between pre-training and fine-tuning for prompt tuning, we present Multi-task Pre-trained Modular Prompt (MP$^2$). As illustrated in Figure 1, we insert a second pre-training procedure before downstream fine-tuning, in which we pre-train a set of modular prompts with multi-task learning. The modular prompts are selectively activated and combined by a trainable router for specific tasks. By this, we can achieve fast adaptation to downstream tasks by learning to combine and reuse the pre-trained modular prompts. Drawing inspiration from the success of deep prompt tuning (Li and Liang, 2021; Liu et al., 2021b), we inject soft prompt into every layer of the PTM. Furthermore, considering that a variety of tasks cannot be reformulated into a (M)LM task, we instead recast upstream and downstream tasks into a unified machine reading comprehension (MRC) task, which has shown great potential to unify various NLP tasks (McCann et al., 2018; Sun et al., 2022b).

We pre-train MP$^2$ on 38 Chinese NLP tasks and evaluate on 14 downstream tasks including sentiment analysis, topic classification, natural language inference, question answering, multiple choice classification, and keyword extraction. Experimental results in few-shot learning settings demonstrate that MP$^2$ outperforms prompt tuning, full model tuning, and previous prompt pre-training methods (Gu et al., 2021; Vu et al., 2022) by a large margin. We also evaluate the compatibility of MP$^2$ with black-box tuning (BBT) (Sun et al., 2022c) and BBTv2 (Sun et al., 2022a), which are gradient-free prompt tuning methods. As a result, MP$^2$ achieves significant improvement over BBT and BBTv2. Besides, we demonstrate that MP$^2$ can achieve surprisingly fast adaptation to target tasks by merely tuning the router (only 8 parameters) while freezing the PTM and all the prompts.

### Table 1: Comparison of model size and data size for various pre-training methods.

| Method     | Params. | Data Size | Data/Param. |
|------------|---------|-----------|-------------|
| PPT        | 410K    | 10 GB     | 24.39 GB/M  |
| MP$^2$ (Ours) | 307M   | 15 GB     | 0.05 GB/M   |
| BERT       | 335M    | 16 GB     | 0.05 GB/M   |
| XLNet      | 335M    | 158 GB    | 0.47 GB/M   |
| RoBERTa    | 355M    | 160 GB    | 0.48 GB/M   |
| BART       | 406M    | 160 GB    | 0.39 GB/M   |
| T5         | 11B     | 745 GB    | 0.07 GB/M   |

Table 1: Comparison of model size and data size for various pre-training methods. In contrast to conventional PTMs, there is a mismatch between the number of learnable parameters and the volume of training data for PPT.

**Comparison with PPT.** A prior work with the similar motivation is Pre-trained Prompt Tuning (PPT) (Gu et al., 2021), which pre-trains soft prompt prepended to the input embedding on large-scale unlabeled corpora with an objective of next sentence prediction (NSP). Different from the NSP in BERT (Devlin et al., 2019), PPT recasts the NSP task into a multiple choice classification (MCC) format. For downstream tasks, PPT formulates three types of tasks, namely single-sentence, sentence-pair, and multiple choice classification, into a unified MCC format such that the gap between the pre-training task and downstream tasks can be filled. Despite their success, we argue that PPT has three possible defects: (1) **Complexity Mismatch**: The number of learnable parameters and the volume of training data are mismatched. PPT trains 410K parameters with 10 GB training data. By contrast, conventional PTMs have much smaller data-parameter ratios (see Table 1). Hence, the limited number of parameters can hardly contain the rich knowledge in the large corpora. (2) **Simple Objective**: The pre-training objective of PPT, i.e., NSP, is not difficult enough. It has been shown that the impact of the NSP objective is unreliable (Yang et al., 2019b; Liu et al., 2019). As formulated by Lan et al. (2020), NSP can be accomplished through two subtasks, **topic prediction** and **coherence prediction**. Nevertheless, topic prediction is easier to learn than coherence prediction, and therefore can dominate learning and makes NSP a rather simple task. (3) **Limited Task**: The downstream tasks handled by PPT are limited. PPT cannot address tasks that cannot be reformulated into a MCC task.

2 Related Work

This work lies in the line of parameter-efficient tuning (PET) (He et al., 2021; Ding et al., 2022), which trains a small portion of parameters to adapt PTMs to downstream tasks. The small tunable parameters can be lightweight neural adapters between PTM layers (Houlsby et al., 2019), or soft prompt attached to the input examples (Lester et al., 2021) or hidden states (Li and Liang, 2021), or bias terms in the PTM parameters (Zaken et al., 2022), or low-rank matrices to be added to attention weights (Hu et al., 2021). Especially, this work is closely related to two prior works on prompt tuning, namely PPT (Gu et al., 2021) and SPoT (Vu et al., 2022).

### Method Params. Data Size Data/Param.

| Method     | Params. | Data Size | Data/Param. |
|------------|---------|-----------|-------------|
| PPT        | 410K    | 10 GB     | 24.39 GB/M  |
| MP$^2$ (Ours) | 307M   | 15 GB     | 0.05 GB/M   |
| BERT       | 335M    | 16 GB     | 0.05 GB/M   |
| XLNet      | 335M    | 158 GB    | 0.47 GB/M   |
| RoBERTa    | 355M    | 160 GB    | 0.48 GB/M   |
| BART       | 406M    | 160 GB    | 0.39 GB/M   |
| T5         | 11B     | 745 GB    | 0.07 GB/M   |
We first introduce the MRC format used to unify where the output label words cannot be shared be-
with the NSP loss. Third, MP² increases capacity of prompt in two dimensions, i.e., depth (deep prompt) and width (modular prompt), to match the complexity of training data. Second, MP² is pre-trained on 38 real-world Chinese tasks with multi-task learning, instead of pre-training in a self-supervised fashion with the NSP loss. Third, MP² recasts upstream and downstream tasks into a unified MRC task to support a wider range of downstream tasks.

Comparison with SPoT. Another work that is similar to ours is Soft Prompt Transfer (SPoT) (Vu et al., 2022), which also explored training soft prompt with multi-task learning and then using it to initialize the prompt for a target task. By comparison, our proposed MP² has three main differences from SPoT: (1) We pre-train a set of modular prompts that are selectively combined and attached to every layer of the PTM rather than training a single prompt to be prepended merely to the input layer. (2) We formulate upstream and downstream tasks into a unified MRC task instead of unifying tasks into a text-to-text format (Raffel et al., 2020) where the output label words cannot be shared between upstream and downstream tasks.² (3) Unlike SPoT that is mainly evaluated in full data settings, MP² is dedicated to few-shot learning.

3 Methods

We first introduce the MRC format used to unify different tasks in §3.1, and then describe the deep modular prompt in §3.2, and finally we detail the procedure of multi-task pre-training and down-
stream fine-tuning in §3.3 and §3.4, respectively.

3.1 Unifying Tasks with MRC

Bridging the gap between upstream and downstream tasks is crucial for few-shot learning. Prompt-based learning (Liu et al., 2021a) formu-
lates downstream tasks into a (M)LM task, which, however, cannot cover a wide range of tasks. Be-
sides, the label words (a.k.a. verbalizer) can be different across tasks. Therefore, the soft prompt pre-trained with a certain set of label words can be less effective to be used in a target task with a different set of label words. To that end, PPT (Gu et al., 2021) recasts upstream and downstream tasks into a MCC task such that different tasks can share the same set of label words, i.e., 16 option indicators (A–P). As a result, there is still a gap between pre-training and fine-tuning when performing classification with more than 16 labels. In addition, the task types supported by MCC can still be limited.

In MP², we adopt a more general format, machine reading comprehension (MRC), to unify upstream and downstream tasks. MRC has achieved great success in unifying a variety of NLP tasks (Sun et al., 2022b). The input of MRC is comprised of a passage (also referred to as context) and a query, and the output is the answer of the query, which is a span of text in the input. Typically, the prediction of the answer is achieved by two binary classification heads on each token of the input, one for predicting the start position and one for predicting the end position (Xiong et al., 2017; Seo et al., 2017).

For classification tasks, we use the original sam-
ple as the context and construct a query consisting of all possible labels. In contrast to PPT that pre-
defines a set of option indicators, MP² directly extracts the answer from the query, and therefore can generalize across tasks with different numbers of labels. Appendix C contains some examples of converting tasks into the MRC format.

3.2 Deep Modular Prompt

To increase the capacity of the soft prompt such that it can match the complexity of training data, we extend soft prompt in two dimensions, depth and width. Figure 2 provides an overview of the deep modular prompt.

Deep Prompt. Inspired by the success of deep prompt tuning (Li and Liang, 2021; Qin and Eisner, 2021; Liu et al., 2021b), we inject soft prompt to every layer of the PTM instead of the mere input layer. The incorporation of deep prompt increases the number of learnable parameters and so as the adaptation ability to hard tasks.

Modular Prompt. For the soft prompt attached to each layer of the PTM, we extend the single static prompt to a set of modular prompts. Formally, we pre-train $K$ soft prompts $\{p_1^{(l)}, \ldots, p_K^{(l)}\}$ for each layer $l$. For a certain task, the prompt at layer

²A shared set of label words in prompt pre-training can be crucial to few-shot leaning. For example, PPT recasts tasks into the MCC format such that the label words are constrained to option words, i.e., $\{A, B, C, \ldots\}$. 
Formula for the weighted mean of soft prompts:

\[ p^{(l)} = \frac{1}{K} \sum_{k=1}^{K} w_k^{(l)} p_k \tag{1} \]

where \( w_k^{(l)} \in \{0, 1\}^K \) are layer-specific learnable parameters.

**Relaxed Bernoulli Distribution.** A challenge is that the discrete router \( w \) is not differentiable and therefore cannot be optimized by gradient descent in an end-to-end fashion. To that end, we keep \( w \in \mathbb{R}^K \) as free parameters to parameterize a relaxed Bernoulli (or binary concrete) distribution (Maddison et al., 2017), which can be considered as a continuous relaxation of the Bernoulli distribution. From the relaxed Bernoulli distribution, we sample \( \hat{w}_k \) to weight the modular prompts, i.e.,

\[ \hat{w}_k = \text{RelaxedBernoulli}(\alpha, \tau) \]

where \( \alpha \in (0, \infty) \) is the location parameter, \( \tau \in (0, \infty) \) is the temperature to control the degree of approximation. Note that \( \hat{w}_k \) can be negative during training and therefore cannot be used directly as the location parameter. To ensure that \( \alpha \in (0, \infty) \), we set \( \alpha \) as follows,

\[ \alpha = \frac{\sigma(\hat{w}_k)}{1 - \sigma(\hat{w}_k)} \tag{5} \]

During inference, we simply set \( \hat{w}_k = 1 \) if \( \hat{w}_k > 0 \), otherwise \( \hat{w}_k = 0 \).

**Intrinsic Reparameterization.** Recent studies (Sun et al., 2022c; Diao et al., 2022) have demonstrated that prompt tuning can be achieved in a much lower dimensional intrinsic subspace through gradient-free optimization. To benefit tuning in the intrinsic subspace, we perform intrinsic reparameterization, which is to decompose the original modular prompt \( p_k \in \mathbb{R}^D \) into an intrinsic prompt \( z_k \in \mathbb{R}^d \) and a projection matrix \( A \in \mathbb{R}^{D \times d} \). Note that \( A \) is shared by all prompts in the same layer. During multi-task pre-training, both \( z_k \) and \( A \) are updated. On downstream tasks, black-box tuning (BBT) (Sun et al., 2022c) can be enabled by only tuning the intrinsic prompt \( z_k \) while keeping \( A \) frozen.

---

Footnote:

3For simplicity, we omit the superscript \( (l) \) without causing confusion.
Multi-Task Pre-Training

Multi-task learning has been shown to boost the performance of prompt tuning in a variety of tasks (Vu et al., 2022). Following their success, we pre-train the deep modular prompts on a mixture of 38 Chinese NLP tasks with varying types, domains, and sizes. To handle the unbalanced data sizes, for each forward computation, we first randomly sample a task ID from 1 to 38 and then fetch a batch of training data corresponding to the sampled task, such that the number of learning steps for each task is expected to be identical.

Fast and Slow Learning. For the pre-training of the routers and the prompts, we intuitively encourage fast learning for the routers to reuse existing modular prompts to adapt to the current task, and slow learning for the task-specific prompts. In particular, we adopt a higher learning rate for the routers $z$ to change quickly, and adopt a lower learning rate for the modular prompts $p$ to change slowly and stably. Similar ideas are also explored by Madan et al. (2021); Ponti et al. (2022).

Downstream Fine-Tuning

For fast adaptation to downstream tasks, we propose the two-stage tuning. In stage I, we allocate a random router for each layer to a new target task and train the routers to selectively reuse pre-trained modular prompts to solve the target task while keeping all other parameters frozen. In stage II, we freeze the routers and only tune the selected prompts. The PTM parameters are unchanged throughout the entire fine-tuning process.

We explore fine-tuning MP$^2$ under two learning paradigms, namely gradient descent and black-box tuning. For gradient descent, we use an Adam (Kingma and Ba, 2015) optimizer to perform two-stage tuning. For black-box tuning, we adopt the Bayesian optimization (BO) (Mockus, 1974) in stage I to optimize the routers, and adopt the CMA-ES (Hansen and Ostermeier, 2001) to optimize the selected intrinsic prompts $z_k$ while freezing the projection matrices $A$. See Appendix A for detailed description of fine-tuning.

4 Experiments

4.1 Datasets and Tasks

Pre-training Tasks. We collect 38 public Chinese NLP tasks ranging from different task types, domains, and data sizes as upstream tasks for pre-training. The total size of the pre-training data is 15GB. Appendix D contains full details of the pre-training tasks.

Downstream Tasks. We divide 14 downstream tasks into two tracks: UNSEEN DATA and UNSEEN TASK. The 7 tasks in the UNSEEN DATA track are a subset of upstream tasks, for which we retain a small portion of training data from the pre-training corpora to ensure that the downstream samples are unseen to MP$^2$. The UNSEEN TASK track is comprised of 7 tasks that are completely held-out tasks. Table 2 contains statistics of the downstream tasks. The sources of the tasks are in Appendix D.

True Few-Shot Setting. For downstream tasks, we follow the same procedure as Gu et al. (2021) to form the true few-shot learning settings (Perez et al., 2021). In particular, we randomly draw 32 samples from the original training set to construct a few-shot training set $D_{train}$, and construct a development set $D_{dev}$ by randomly selecting another 32 samples from the original training set. We ensure that the number of labels is balanced for both training and development set. For tasks with more than 5 labels, we randomly select 8 samples for each label. We use the original development sets as the test sets. For datasets without development sets, we use the original test sets.

4.2 Backbones and Baselines

We choose CPT-large (Shao et al., 2021) as our backbone model, which is a competitive Chinese
we pre-trained 24 sets of modular prompts, each (Lester et al., 2021), which prepends a sequence of soft prompt tokens to the input and only tunes the soft prompt for adaptation; (5) SPoT (Vu et al., 2022), which pre-trains soft prompt with multi-task learning. For fair comparison, we reimplement PPT and SPoT with the same backbone model, i.e., CPT-large. For PPT, we pre-trained the “Unified PPT” on the same pre-training corpora as in the original paper, i.e., 10GB WuDaoCorpora (Yuan et al., 2021). For SPoT, we pre-trained a single soft prompt with the same 38 Chinese NLP tasks as used by MP2. Therefore, experiments of SPoT can be seen as an ablation study on the effect of the modular prompt. For black-box tuning, we pre-trained a 20-layered shared encoder, a 4-layered understanding decoder and a 4-layered generation decoder. In our experiment, we use the encoder and the understanding decoder to compose a 24-layered PTM. We attach soft prompt to the input layer and all intermediate layers except the last layer, which has no effect on the output. Therefore, we pre-trained 24 sets of modular prompts, each corresponding to one layer of CPT. In addition to the pre-trained Deep MP2, we also pre-trained a set of modular prompts that are merely attached to the input layer, denoted as Shallow MP2.

We evaluate MP2 under two learning paradigms: gradient descent and black-box tuning. For gradient descent, we consider (1) Model Tuning, which fine-tunes all parameters of the PTM; (2) Prompt Tuning (Lester et al., 2021), which prepends a sequence of soft prompt tokens to the input and only tunes the soft prompt for adaptation; (3) P-Tuning v2 (Liu et al., 2021b): which incorporates and tunes soft prompt at every layer of the PTM. Prompt tuning and p-tuning v2 can be seen as the baselines to Shallow MP2 and Deep MP2, respectively. Besides, we compare with two previous prompt pre-training methods: (4) PPT (Gu et al., 2021), which pre-trains soft prompt on large-scale unlabeled data with self-supervised learning; and (5) SPoT (Vu et al., 2022), which pre-trains soft prompt with multi-task learning. For fair comparison, we reimplement PPT and SPoT with the same backbone model, i.e., CPT-large. For PPT, we pre-trained the “Unified PPT” on the same pre-training corpora as in the original paper, i.e., 10GB WuDaoCorpora (Yuan et al., 2021). For SPoT, we pre-trained a single soft prompt with the same 38 Chinese NLP tasks as used by MP2. Therefore, experiments of SPoT can be seen as an ablation study on the effect of the modular prompt. For black-box tuning,
we consider two baselines: (1) BBT (Sun et al., 2022c), which adopts a gradient-free optimizer to tune a low-dimensional intrinsic prompt, and then randomly embeds it into the original prompt space to be concatenated with the input embedding; and (2) BBTv2 (Sun et al., 2022a), which extends BBT by incorporating soft prompt into every layer of the PTM and uses a divide-and-conquer algorithm to alternately optimize the soft prompt at each layer.

The prompt length is set to 50 for both shallow MP\(^2\) and deep MP\(^2\). Each set of modular prompts is consisting of \(K = 8\) soft prompts, and therefore the pre-trained routers are in the shape of \(38 \times 8\). Shallow MP\(^2\) has only one router while deep MP\(^2\) contains 24 routers corresponding to 24 layers. Hyper-parameters and more implementation details are provided in Appendix A.

### 4.3 Results

#### Main Results.

Main results on 14 downstream tasks are listed in Table 3. We report mean and standard deviation of performance over 5 runs with different random seeds. Overall, MP\(^2\) outperforms all baselines by a large margin. By further comparison, we have the following findings: (1) **Deep Prompt vs. Shallow Prompt**: Deep prompt methods (i.e., p-tuning v2, BBTv2, and deep MP\(^2\)) significantly outperform their corresponding shallow versions (i.e., prompt tuning, BBT, and shallow MP\(^2\)).

(2) **Modular Prompt vs. Single Prompt**: Shallow MP\(^2\) achieves better performance than SPoT on 13/14 tasks, demonstrating the strong compositional generalization of the modular prompts. (3) **MRC vs. MCC**: PPT lags far behind MP\(^2\) (and even p-tuning v2) on two MRC tasks, namely CMRC-2018 and DRCD, demonstrating the limitation of the MCC format. (4) **Pre-trained Prompt Tuning vs. Prompt Tuning From Scratch**: Pre-trained prompt tuning (i.e., PPT, SPoT, and MP\(^2\)) performs consistently better than tuning randomly initialized prompt with the same number of tunable parameters. (5) **Gradient Descent vs. Black-Box Tuning**: Without MP\(^2\) for initialization, BBT and BBTv2 achieve better performance than prompt tuning and p-tuning v2, respectively, on most tasks but much worse performance on a few tasks such as CCPM. By using MP\(^2\) for initialization, the gap between gradient descent and black-box tuning on these tasks are closed, and in average, BBT and BBTv2 outperform their gradient-based counterparts, showing the superiority of gradient-free optimization in few-shot learning settings.

#### Two-Stage Tuning.

As demonstrated in Table 3, by only tuning the router (only stage I), which contains merely 8 parameters for shallow MP\(^2\) or \(8 \times 24 = 192\) parameters for deep MP\(^2\), we can achieve surprisingly strong performance that can be comparable to two-stage tuning. For shallow MP\(^2\), only tuning the router even outperforms two-stage tuning in average on **Unseen Data** tasks. To take a closer look, we demonstrate the process of two-stage tuning with shallow MP\(^2\) for initialization in Figure 3. For both learning paradigms, we find that the best performance on the development set of the **Unseen Data** task (here is the BQ task) can be observed in stage I, where we only tune the router to reuse pre-trained prompts. On **Unseen Task**
(here is the LCQMC task), we observe improvement of performance during stage II. In Table 4, we compare the training time of the two stages to show the high efficiency of stage I when using black-box tuning. Results suggest that learning to combine instead of tuning the prompts is a promising way to achieve fast adaptation to downstream tasks.

**On Many-Label Classification Tasks.** In contrast to PPT that is pre-trained to perform up to 16-label classification, our proposed MP\(^2\) unifies tasks into the MRC format such that it can generalize to downstream tasks with varying numbers of labels. To simulate tasks with different numbers of labels, we extract subsets with 10/15/20/25/30 labels from the IFLYTEK dataset, which contains 119 labels in total. We follow the same procedure (§4.1) to generate train/dev/test splits from the extracted subsets. As shown in Figure 4(a), there is a sharp decline in the accuracy of PPT when the number of labels exceeds 16. By contrast, the performance of MP\(^2\) is decreasing more slowly and steadily as the number of labels increases, demonstrating the superiority of the MRC format.

**On Sample Efficiency.** We compare MP\(^2\) and PPT with different numbers of training samples on the LCQMC task. As shown in Figure 4(b), increasing training samples generally confers improved performance for both MP\(^2\) and PPT while MP\(^2\) consistently outperforms PPT under varying numbers of training samples. In addition, the gap between MP\(^2\) and PPT cannot be easily filled with enlarged training set.

**Task Partitions Induced From the Router.** We take a closer look at the learned router and find that non-trivial task partitions can be induced from it. For simplicity, we focus on the shallow MP\(^2\), which has only one router. There are totally 8 modular prompts corresponding to \(2^8 = 256\) possible combinations. We perform a hierarchical clustering on the router learned on 38 upstream tasks and visualize the task partitions in Figure 5. The 38 upstream tasks can be partitioned into 8 groups. For instance, group A is mainly comprised of topic classification tasks; group D contains all the sentiment analysis tasks; group C and E are all comprised of NLI tasks, among which group E covers all the "Zhidao" tasks, which are question-answer matching tasks.

5 Conclusion

This work aims to bridge the gap between pre-training and fine-tuning of soft prompt tuning for few-shot learning. To achieve this, we extend the soft prompt in two dimensions, depth and width. The extended prompt, named deep modular prompt, is pre-trained on a mixture of 38 public Chinese NLP tasks, which are reformulated into the MRC format. For adaptation to downstream tasks, we propose the two-stage tuning, where we first learn to combine and reuse pre-trained prompts and then tune the selected prompts with gradient descent or black-box optimization. Extensive experiments on 14 downstream tasks demonstrate that, the Multi-task Pre-trained Modular Prompt (MP\(^2\)) significantly outperforms prompt tuning, full model tuning, and previous prompt pre-training methods, namely PPT and SPoT. Surprisingly, we demonstrate that MP\(^2\) can achieve extremely fast adaptation to downstream tasks by only learning to com-
bine pre-trained prompts.

**Limitations**

In this work, we demonstrate the effectiveness of the proposed MP$^2$ with the backbone PTM of CPT-large on a set of Chinese NLP tasks. Due to the expensive pre-training cost, we did not explore MP$^2$ on other PTMs with varying sizes, pre-training objectives and architectures. Besides, it is also unknown how does the number of pre-training tasks affect the performance of MP$^2$. For resource-rich languages such as English and Chinese, it would be promising for MP$^2$ to be well-performed since one can easily collect sufficient public upstream tasks. Nevertheless, for low-resource languages or domains, the effect of MP$^2$ is still under-explored.

**Ethics Statement**

The proposed MP$^2$ is a parameter-efficient approach for few-shot learning. In addition, we demonstrate that MP$^2$ can achieve highly efficient adaptation to a target task by only tuning a few parameters. Therefore, this work helps reduce computation costs and carbon emissions, and can facilitate the adaptation of PTMs to low-resource downstream tasks. Though all the datasets used in our experiments are publicly available and have not been reported to carry social bias against any sensitive attributes, and the proposed approach would not explicitly introduce new negative societal impacts, more work is still needed to investigate the potential unfairness in these datasets.

**References**

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.*

Jing Chen, Qingcai Chen, Xin Liu, Haijun Yang, Daohe Lu, and Buzhou Tang. 2018. *The BQ corpus: A large-scale domain-specific Chinese corpus for sentence semantic equivalence identification.* In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4946–4951, Brussels, Belgium. Association for Computational Linguistics.

Alexis Conneau, Ruty Rinott, Guillaume Lample, Adina Williams, Samuel R. Bowman, Holger Schwenk, and Veselin Stoyanov. 2018. *Xnli: Evaluating cross-lingual sentence representations.* In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*. 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.

Shizhe Diao, Xuechun Li, Yong Lin, Zhichao Huang, and Tong Zhang. 2022. *Black-box prompt learning for pre-trained language models.* CoRR, abs/2201.08531.

Ning Ding, Yujia Qin, Guang Yang, Fuchao 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.* CoRR, abs/2203.06904.

Tianyu Gao, Adam Fisch, and Danqi Chen. 2021. *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, ACL/IJCNLP 2021, (Volume 1: Long Papers), Virtual Event, August 1-6, 2021*, pages 3816–3830. Association for Computational Linguistics.

Yuxian Gu, Xu Han, Zhiyuan Liu, and Minlie Huang. 2021. *PPT: pre-trained prompt tuning for few-shot learning.* CoRR, abs/2109.04332.

Nikolaus Hansen and Andreas Ostermeier. 2001. *Completely derandomized self-adaptation in evolution strategies.* Evol. Comput., 9(2):159–195.

Junxian He, Chunting Zhou, Xuezhe Ma, Taylor Berg-Kirkpatrick, and Graham Neubig. 2021. *Towards a unified view of parameter-efficient transfer learning.* CoRR, abs/2110.04366.

Neil Houlsby, Andrei Giurgiu, Stanislav Jastrzebski, Bruna Morrone, Quentin de Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly.
Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, and Weizhu Chen. 2021. Lora: Low-rank adaptation of large language models. CoRR, abs/2106.09685.

Hai Hu, Kyle Richardson, Liang Xu, Lu Li, Sandra Kübler, and Lawrence S. Moss. 2020. OCNLI: original chinese natural language inference. In Findings of the Association for Computational Linguistics: EMNLP 2020, Online Event, 16-20 November 2020, volume EMNLP 2020 of Findings of ACL, pages 3512–3526. Association for Computational Linguistics.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In 3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015. Conference Track Proceedings.

Diederik P. Kingma and Max Welling. 2014. Auto-encoding variational bayes. In 2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings.

Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. 2020. ALBERT: A lite BERT for self-supervised learning of language representations. In 8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020. OpenReview.net.

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.

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.

Yanzeng Li, Tingwen Liu, Diying Li, Quangang Li, Jingqiao Shi, and Yanqiu Wang. 2018. Character-based bilstm-crf incorporating pos and dictionaries for chinese opinion target extraction. In Asian Conference on Machine Learning, pages 518–533. PMLR.

Ziran Li, Ning Ding, Zhiyuan Liu, Haitao Zheng, and Ying Shen. 2019. Chinese relation extraction with multi-grained information and external linguistic knowledge. In Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, pages 4377–4386, Florence, Italy. 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, Zhiyin 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, Yinan Zheng, Zhengxiao Du, Ming Ding, Yujie Qian, Zhiyin Yang, and Jie Tang. 2021c. GPT understands, too. CoRR, abs/2103.10385.

Xin Liu, Qingcai Chen, Chong Deng, HuaJun Zeng, Jing Chen, Dongfang Li, and Buzhou Tang. 2018. LCQMC: A large-scale Chinese question matching corpus. In Proceedings of the 27th International Conference on Computational Linguistics, COLING 2018, Santa Fe, New Mexico, USA, August 20-26, 2018, pages 1952–1962. Association for Computational Linguistics.

Yinhua 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.

Kanika Madan, Nan Rosemary Ke, Anirudh Goyal, Bernhard Schölkopf, and Yoshua Bengio. 2021. Fast and slow learning of recurrent independent mechanisms. In 9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021. OpenReview.net.

Chris J. Maddison, Andriy Mnih, and Yee Whye Teh. 2017. The concrete distribution: A continuous relaxation of discrete random variables. In 5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings. OpenReview.net.

Bryan McCann, Nitish Shirish Keskar, Caiming Xiong, and Richard Socher. 2018. The natural language decathlon: Multitask learning as question answering. CoRR, abs/1806.08730.
Canwen Xu, Wangchunshu Zhou, Tao Ge, Ke Xu, Julian McAuley, and Furu Wei. 2021. Blow the dog whistle: A chinese dataset for cant understanding with common sense and world knowledge. arXiv preprint arXiv:2104.02704.

Jingjing Xu, Ji Wen, Xu Sun, and Qi Su. 2017. A discourse-level named entity recognition and relation extraction dataset for chinese literature text. volume abs/1711.07010.

Liang Xu, Hai Hu, Xuanwei Zhang, Lu Li, Chenjie Cao, Yudong Li, Yechen Xu, Kai Sun, Dian Yu, Cong Yu, Yin Tian, Qianqian Dong, Weitang Liu, Bo Shi, Yiming Cui, Junyi Li, Jun Zeng, Rongzhao Wang, Weijian Xie, Yanting Li, Yina Patterson, Zuoyu Tian, Yiwen Zhang, He Zhou, Shaoweiha Liu, Zhe Zhao, Qipeng Zhao, Cong Yue, Xinrui Zhang, Zhengliang Yang, Kyle Richardson, and Zhennong Lan. 2020. CLUE: A Chinese language understanding evaluation benchmark. In Proceedings of the 28th International Conference on Computational Linguistics, pages 4762–4772, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Yinfei Yang, Yuan Zhang, Chris Tar, and Jason Baldridge. 2019a. Paws-x: A cross-lingual adversarial dataset for paraphrase identification. arXiv preprint arXiv:1908.11828.

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

Sha Yuan, Hanyu Zhao, Zhengxiao Du, Ming Ding, Xiao Liu, Yukuo Cen, Xu Zou, Zhilin Yang, and Jie Tang. 2021. Wudaocorpora: A super large-scale chinese corpora for pre-training language models. AI Open, 2:65–68.

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), ACL 2022, Dublin, Ireland, May 22-27, 2022, pages 1–9. Association for Computational Linguistics.

Fan Zhang, Duyu Tang, Yong Dai, Cong Zhou, Shuangzhi Wu, and Shuming Shi. 2022a. Skillnet-nlu: A sparsely activated model for general-purpose natural language understanding.

Ningyu Zhang, Mosha Chen, Zhen Bi, Xiaozhuan Liang, Lei Li, Xin Shang, Kangping Yin, Chuanqi Tan, Jian Xu, Fei Huang, Luo Si, Yuan Ni, Guotong Xie, Zhihang Sui, Baobao Chang, Hui Zong, Zheng Yuan, Linfeng Li, Jun Yan, Hongying Zan, Kunli Zhang, Buzhou Tang, and Qingcai Chen. 2022b. CBLUE: A chinese biomedical language understanding evaluation benchmark. In Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2022, Dublin, Ireland. May 22-27, 2022, pages 7888–7915. Association for Computational Linguistics.

Hui Zong, Jinxuan Yang, Zeyu Zhang, Zuofeng Li, and Xiaoyan Zhang. 2021. Semantic categorization of chinese eligibility criteria in clinical trials using machine learning methods. BMC Medical Informatics Decis. Mak., 21(1):128.
A Implementation Details

A.1 Upstream Pre-training

MP$^2$. MP$^2$ is pre-trained on 38 upstream tasks using an Adam optimizer with batch size of 32 for 1M steps. During each forward computation, we first randomly select a task and then fetch a batch of training data corresponding to the selected task. By this, the number of learning steps on each task is expected to be identical. As demonstrated in Table 5, the fast and slow learning (FSL) can be beneficial to deep MP$^2$, and therefore we use two-speed learning rate for pre-training the routers and the prompts of deep MP$^2$. In particular, the learning rate of the routers is 5e-4, and the learning rate of the prompts is 1e-4. For shallow MP$^2$, we use a single learning rate of 1e-3 for the router and the modular prompts. The prompt length is set to 50 for both shallow MP$^2$ and deep MP$^2$. For shallow MP$^2$ and each layer of the deep MP$^2$, we allocate $K = 8$ modular prompts and one router to combine them. In addition to the routers and the prompts, we also train the randomly initialized MRC head on the top of the PTM. The original parameters of the PTM are frozen during pre-training. We run pre-training on NVIDIA A100 GPUs.

Baselines. For fair comparison, we also re-implement PPT and SPoT with the same backbone model as MP$^2$, i.e., CPT-large. For pre-training PPT, we implement the "Unified PPT" variant, which is to formulate tasks into a unified MCC format, to support a variety of downstream tasks. We follow the experimental setup in the original paper and use 10GB data sampled from the WuDaoCorpora for pre-training. We train for 400K steps using an Adam optimizer with batch size of 32 and learning rate of 3e-2. For SPoT, we pre-trained a single soft prompt on the same 38 upstream tasks as used by MP$^2$ using an Adam optimizer with batch size of 32 and learning rate of 3e-3 for 650K steps. Though the numbers of training steps for PPT and SPoT are less than MP$^2$, they are sufficient for convergence due to their limited numbers of parameters. To be consistent with MP$^2$, we set prompt length to 50 for PPT and SPoT.

A.2 Downstream Fine-tuning

We use the two-stage tuning to adapt MP$^2$ to various downstream tasks. In stage I, we only tune the router(s) while keeping all other parameters frozen. In stage II, we fix the learned router(s) and only fine-tune the modular prompts selected by the router(s). The implementation details of the two-stage tuning can be different for gradient descent and black-box tuning. We provide a graphical illustration of the two-stage tuning using gradient descent and black-box tuning in Figure 6. For gradient descent, we fine-tune MP$^2$ for 1K epochs on each task, where the first 500 epochs as stage I and the last 500 epochs as stage II. For the shallow/deep MP$^2$, we use an Adam optimizer with learning rate of 1e-2/3e-3 for tuning the router(s) (stage I) and learning rate of 3e-4/2e-5 for tuning the prompts (stage II). For black-box tuning, we fine-tune shallow/deep MP$^2$ for 8K iterations (model forward computes) on each task, where the first 200/100 iterations as stage I and the rest as stage II. In stage I, we use Bayesian optimization (BO) with the acquisition function of upper confidence bound (UCB) with $\kappa = 2$ to tune the parameters of the router(s). In stage II, we use CMA-ES to optimize the prompts. For shallow MP$^2$, we use $\mu = 0$ and $\sigma = 0.1$ for initialization of the CMA-ES. For deep MP$^2$, we follow BBTv2 and use the divide-and-conquer algorithm to alternately optimize the prompt at each layer. For optimization of the prompt at the embedding layer, we initialize CMA-ES with $\mu = 0$ and $\sigma = 5e-2$. For optimization of the prompt at intermediate layers, we adopt $\mu = 0$ and $\sigma = 1e-2$. All the hyper-parameters are tuned manually in a lightweight manner on development sets. We perform fine-tuning a single NVIDIA 3090 GPU.

B Additional Results

Ablation of Fast and Slow Learning. We conduct ablation study on fast and slow learning (FSL), which is to assign different learning rates to routers

| Methods | ChnSent | TNews | LCQMC | DRCG |
|---------|---------|-------|-------|------|
| Shallow MP$^2$ | | | | |
| w/o FSL | 90.46 ± 0.16 | 51.36 ± 0.12 | 72.50 ± 1.92 | 67.20 ± 2.96 |
| w/ FSL | 89.36 ± 0.63 | 51.36 ± 1.30 | 70.42 ± 1.27 | 58.96 ± 0.73 |
| Deep MP$^2$ | | | | |
| w/o FSL | 91.61 ± 0.18 | 55.23 ± 0.29 | 82.30 ± 1.28 | 78.69 ± 0.72 |
| w/ FSL | 92.02 ± 0.11 | 54.71 ± 0.31 | 83.45 ± 1.00 | 80.64 ± 0.87 |

Table 5: Ablation of fast and slow learning (FSL).
Stage I

Router ($\mathbb{R}^K$)

Intrinsic Prompts ($\mathbb{R}^d$)

Mean Activated Prompts ($\mathbb{R}^D$)

Fused Prompt

Pre-Trained LM (Adam)

Stage II

(a) Gradient descent (tunable parameters: $[K + D, K + KD]$ per layer)

Stage I

Router ($\mathbb{R}^K$)

Intrinsic Prompts ($\mathbb{R}^d$)

Fused Prompt ($\mathbb{R}^d$)

Pre-Trained LM (BO)

Stage II

Mean

Projected Prompt ($\mathbb{R}^D$)

Pre-Trained LM (CMA-ES)

(b) Black-box tuning (tunable parameters: $K + d$ per layer)

Figure 6: Illustration of the two-stage tuning for gradient descent and black-box tuning. For black-box tuning, which is a gradient-free optimization approach that cannot well handle high-dimensional optimization, we perform pre-fusion to obtain a low-dimensional fused prompt for optimization.

| Dataset     | Source                        |
|-------------|-------------------------------|
| ChnSent     | https://github.com/SophonPlus/ChineseNlpCorpus |
| TNews       | https://github.com/fatecbf/toutiao-text-classfication-dataset/ |
| OCNLI       | Hu et al. (2020)               |
| LCQMC       | Liu et al. (2018)              |
| DRCD        | Shao et al. (2018)             |
| C3          | Sun et al. (2020a)             |
| COTE-BD     | Li et al. (2018)               |

Table 6: Sources of the 7 downstream tasks in the UNSEEN TASK track.

and prompts. As demonstrated in Table 5, FSL exhibits positive effect on downstream tasks to deep MP$^2$ and negative effect to shallow MP$^2$. Therefore, we retain the shallow MP$^2$ pre-trained without FSL and the deep MP$^2$ pre-trained with FSL in our experiments.

C MRC Format

We unify upstream and downstream tasks into the machine reading comprehension (MRC) format, which takes as input a context and a query, and outputs the answer of the query. For topic classification and sentence-pair classification tasks, we use the original input text as the context and construct a query containing all valid labels. The context and the constructed query are concatenated and fed into the model. The model is trained to extract the answer in the query by predicting its start and end positions. For more complicated tasks such as relation extraction and poem understanding, we manually design task-specific templates to convert inputs to the desired contexts and queries. Some examples are shown in Table 7. Full templates can be found in our code.

D Additional Details of Tasks

D.1 Upstream Tasks

Table 8 contains details of the 38 upstream tasks. We only use the training sets during pre-training. For tasks that also serve as a downstream task in the UNSEEN DATA track, we remove a small portion of training samples from pre-training to avoid data leakage.

D.2 Downstream Tasks

The downstream tasks are divided into two tracks, UNSEEN DATA and UNSEEN TASK. The tasks in the UNSEEN DATA track are a subset of upstream task, for which the details have been provided in Table 8. For the 7 tasks in the UNSEEN TASK track, we provide the sources in Table 6.
Table 8: Pre-training datasets for MP

| Dataset | Task | Domain | (Train) | (Dev) | (Test) | Labelled | Reference |
|---------|------|--------|---------|-------|--------|----------|-----------|
| Amazon  | TC   | Financial | 31k   | 3k   | 4k    | 2       | Xu et al. (2020) |
| ChnSent | TC   | General   | 44k   | 5k   | 2k    | 2       | Yang et al. (2019a) |
| CSMC    | TC   | General   | 380k  | 12k  | 12k   | 3       | Xu et al. (2020) |
| CSL     | TC   | Academic   | 18k   | 2k   | 3k    | 2       | Xu et al. (2020) |
| BQ      | TC   | Financial  | 90k   | 10k  | 10k   | 2       | Chen et al. (2020) |
| CHIP-STS| TC   | Biomedical | 14k   | 2k   | 4k    | 2       | Zhang et al. (2020b) |
| KUAKE-QOR| TC   | Clinical   | 14k   | 2k   | 2k    | 3       | Zhang et al. (2020a) |
| XNLI    | NLI  | General    | 380k  | 12k  | 2k    | 3       | Conneau et al. (2018) |
| NLPCC-DIQA | NLI | General | 170k | 12k | 41k | 2 | http://tcci.ccf.org.cn/conference/2016 |
| Finance-zhidao | Financial | 64k | 12k | 38k | 2 | https://github.com/SophonPlus/ChineseNlpCorpus |
| Law-zhidao | NLI | Law | 23k | 3k | 7k | 2 | https://github.com/SophonPlus/ChineseNlpCorpus |
| Liangtong-zhidao | TC | Telecom | 150k | 12k | 20k | 2 | https://github.com/SophonPlus/ChineseNlpCorpus |
| Nonghang-zhidao | Financial | 29k | 3k | 4k | 2 | https://github.com/SophonPlus/ChineseNlpCorpus |
| Tugui-zhidao | NLI | Investment | 457k | 12k | 28k | 2 | https://github.com/SophonPlus/ChineseNlpCorpus |
| Baoxian-zhidao | NLI | Insurance | 2k | 0.6k | 2k | 2 | https://github.com/SophonPlus/ChineseNlpCorpus |
| Dianxin-zhidao | TC | Telecom | 99k | 11k | 31k | 2 | https://github.com/SophonPlus/ChineseNlpCorpus |
| Touzi-zhidao | NLI | Investment | 487k | 12k | 29k | 2 | https://github.com/SophonPlus/ChineseNlpCorpus |
| Nonghang-zhidao | NLI | Financial | 29k | 3k | 4k | 2 | https://github.com/SophonPlus/ChineseNlpCorpus |
| Liantong-zhidao | NLI | Telecom | 150k | 12k | 20k | 5 | https://github.com/SophonPlus/ChineseNlpCorpus |
| Law-zhidao | NLI | Insurance | 2k | 0.6k | 2k | 2 | https://github.com/SophonPlus/ChineseNlpCorpus |
| Dianxin-zhidao | TC | Telecom | 99k | 11k | 31k | 2 | https://github.com/SophonPlus/ChineseNlpCorpus |
| THUCNews | TC | General | 45k | 5k | 5k | 10 | https://github.com/thunlp/THUCTC |
| CHIP-CTC | TC | Biomedical | 23k | 8k | 10k | 44 | Zeng et al. (2021) |
| FII-Ro | TC | Financial | 12k | 1k | 1k | 44 | Liet al. (2019) |
| Fudan-TC | TC | General | 9k | 1k | 10k | 20 | Not found! |
| KUAKE-QIC | TC | Clinical | 6k | 0.7k | 2k | 11 | Zhang et al. (2022b) |
| NLPCC-TC | TC | General | 6k | 0.7k | 2k | 2 | http://tcci.ccf.org.cn/conference/2016 |
| Amazon | TC | Shopping review | 3.5M | 12k | 41k | 5 | https://github.com/SophonPlus/ChineseNlpCorpus |
| DianPing | TC | Shopping review | 2.6M | 12k | 30k | 5 | https://github.com/SophonPlus/ChineseNlpCorpus |
| DMSC | TC | Movie review | 1.6M | 12k | 92k | 5 | https://github.com/SophonPlus/ChineseNlpCorpus |
| Online-Shopping | TC | Shopping review | 45k | 5k | 6k | 2 | https://github.com/SophonPlus/ChineseNlpCorpus |
| Waimai | TC | Shopping review | 8k | 0.8k | 2k | 2 | https://github.com/SophonPlus/ChineseNlpCorpus |
| Weibo-sentiment | TC | General | 76k | 8k | 24k | 5 | https://github.com/SophonPlus/ChineseNlpCorpus |
| Touxiu-TC | TC | General | 321k | 12k | 11k | 14 | https://github.com/weimorostwztc/touxiu-text-classification-dataset |
| SanWen | TC | Literature | 13k | 1k | 2k | 10 | Xu et al. (2017) |
| CLUE-WSC | CR | General | 1k | 0.1k | 0.3k | 2 | Xu et al. (2020) |
| CITE-DF | KE | Shopping review | 16k | 2k | 5k | N/A | Liet al. (2018) |
| CITE-MFW | KE | Shopping review | 26k | 3k | 8k | N/A | Liet al. (2018) |
| DuReader-Checklist | MRC | General | 3k | 0.3k | 1k | N/A | https://github.com/baidu/DuReader |
| DuReader-Robust | MRC | General | 13k | 1k | 1k | N/A | Tang et al. (2021) |
| CMRC-2018 | MRC | General | 8k | 0.9k | 3k | N/A | Xu et al. (2020) |
| CCPM | MRCQA | Poem | 19k | 2k | 3k | 4 | https://github.com/SophonPlus/ChineseNlpCorpus |
| DogWhale | TC | General | 218k | 12k | 28k | 4 | Xu et al. (2021) |

Total: 10.7M 213k 499k

Table 8: Pre-training datasets for MP². NLI: natural language inference. TC: text classification. CR: coreference resolution. KE: keyword extraction. MRC: machine reading comprehension. MCQA: multiple choice question answering. † We did not find the official source of the Fudan-TC dataset.