MOP: Efficient Low-rank PHM Mixture of Experts for Prefix-based Multi-scenario Dialogue Summarizaton

Anonymous ACL submission

Abstract

As large-scale pre-training models (PLMs) expand, efficient fine-tuning becomes crucial for rapid adaptation and deployment. We propose **MOP**, a low-rank Mixture of Experts (MOE) network for Prompt reparameterization in multi-scenario summarization based on prefix-tuning. MOP assigns specific experts for summarization in each particular scenario and incorporates an efficient knowledge decoupling mechanism. Specifically, Expert weight matrices are learned as a sum of Kronecker products of shared global and specific local weights, capturing general and task-specific knowledge. We further decompose global weights into low-rank layer-share (LoRL) and expert-share (LoRE) weights, enhancing flexibility and generality. By updating only the MOP, our method outperforms strong baselines across all scenarios on the MultiSum benchmark, using just 2.93% of a pretrained model's parameters, demonstrating MOP's effectiveness in improving multi- scenarios learning performance with fewer parameters.

1 Introduction

Recently, the rapid development of ever-larger pretrained language models has been pushing the boundaries of possibility across various NLP benchmarks(Brown et al., 2020a) (Wei et al., 2021) (Sanh et al., 2021). For models with large-scale parameters, deploying a separate instance of the model for each downstream task, saving and updating separate replicas of these separate model parameters would be more time-consuming and space-consuming. Multi-task frameworks have been proposed to use the same model to handle multiple tasks(Caruana, 1998) (Wang et al., 2018). In particular, there are many scenarios in dialogue summarization and more business requirements are proposed in practical applications, such as Take-out, Taxi, etc. Therefore, it is of great significance to explore multi-task learning for multi-scenario dialogue summarization.

There exist some works for multi-task learning in dialogue summarization. They either rely on additional heavy pre-training and fine-tuning(Sun et al., 2022) (Vu et al., 2021), or employ a large number of task-specific non-shared structures and parameters, which cost grows linearly with the number of tasks (Liu et al., 2018). Some researches have demonstrated that prefix-tuning is a lightweight method (Li and Liang, 2021a) (Liu et al., 2021), which prepends tunable prefix vectors to the keys and values of multi-head attention at each layer, and fixes the original PLM parameters. However, most of them only focus on a single task and cannot outperform full-parameter fine-tuning methods when faced with more challenging tasks like summarization. Besides, reparameterizing the prefix via simple MLP structures cannot effectively alleviate the instability of the model due to the complexity of PLMs(Ding et al., 2022). When prefix-tuning is applied in multi-task learning, task interference or negative transfer often occurs(Haddow and Koehn, 2012) (Kokkinos, 2016) (Kendall et al., 2017) (Sener and Koltun, 2018), i.e. achieving good performance on one task can hinder performance on another. How to improve the performance of the model on multiple tasks while reducing the amount of model parameters and improving the efficiency of model deployment is still an open problem to be explored.

In this paper, we aim to train a unified model for multiple scenario-related dialogue summarization tasks from the perspective of parameter efficiency to reduce model deployment and maintenance. Considering cost constraints and performance requirements, we resort to MOE to expand model capacity with nearly constant computational overhead (Shazeer et al., 2017a) (Lepikhin et al., 2020). We propose **MOP**, an efficient Multi-task Prompt Reparameterization Network for multi-scenario summarization, which uses MOEs for task-aware prefix learning. Here, each expert in MOEs is considered to correspond to scenario. It is worth noting that we design an efficient knowledge decoupling mechanism, which enables the model to learn a better representation for each task.

Inspired by(Mahabadi et al., 2021), each expert weight matrix is computed as the sum of Kronecker products(Zhang et al., 2021) between shared global weights and local weights defined per MOP. This enables MOP to aggregate common knowledge across tasks into global weights and store specific information in local weights. We also introduce a low-rank sharing mechanism, decomposing global weights into low-rank layer-share (LoRL) and expert-share (LoRE) weights, enhancing flexibility in capturing general information. LoRL captures information common to all layers of the same expert, while LoRE obtains information shared by all experts at the same layer. This mechanism reduces shared parameters, improving efficiency. Consequently, **MOP** achieves paramter complexity of $\mathcal{O}(d + d_{mid})$ instead of $\mathcal{O}(dd_{mid})$ for regular prefix-tuning, where the reparameterization matrix is of size $d \times d_{mid}$.

We evaluate our approach on MultiSum, a large-scale customer-service dialogue summarization datasets. Experimental results demonstrate that our **MOP** is significantly better than all methods. In particular, it can go far beyond the performance of the strongest fine-tuning baseline. We further explore the effectiveness of the MoE network and the sharing mechanism in the low-rank decomposition. Additionally, we analyze the trainable parameter scale to verify the efficiency. To sum up, the contributions of this paper are three folds:(1) To the best of our knowledge, we are the first to propose a PHM based mixture of experts for prompt reparameterization to explore multi-scenario summarization. (2) We decompose the share weights into low-rank layer-share weights and expert-share weights, which enable flexible and fine-grained sharing by capturing layer-share information and expert-share knowledge separately. (3) A plenty of experiments and qualitative analysis are conducted to prove the effectiveness of our methods.

2 METHODOLOGY

In this section, we present MOP, a low-rank MOE network for scenario-conditioned prompt reparameterization. Instead of routing mechanisms, we assign experts to handle each scenario, which effectively avoids the problem of load imbalance. Furthermore, we adopt PHM to reduce redundant parameters and design an effective low-rank sharing mechanism to achieve the sharing of common knowledge among different experts.

2.1 Sparse Mixture of Experts

To increase model capacity without a proportional increase in computational costs, we use the SMoE to explicitly model the scenario relationships and learn features relevant to specific scenario(Shazeer et al., 2017a). The original expert network is implemented as stacked feed-forward networks(FFN):

$$f_k(\boldsymbol{h}) = \sigma(\boldsymbol{h}\boldsymbol{W_{down}})\boldsymbol{W_{up}} \tag{1}$$

where $\boldsymbol{W_{down}} \in \mathbb{R}^{d \times d_{mid}}$ is the down-project mapping and $W_{up} \in \mathbb{R}^{n \times d}$ is the up-project mapping, σ means ReLU activation function.

PHM Layer To reduce computation overhead with almost no damage to model performance, we substitute the parameterized hypercomplex multiplication (PHM) layer (Mahabadi et al., 2021) (Zhang et al., 2021) , which is on the basis of Kronecker product, for linear FFN layer in SMoE. To the best of our knowledge, we are the first to exploit PHM layers for efficient fine-tuning of SMoE networks. Assume that d and d_{mid} are both divisible by self-defined hyperparameter $p \in \mathbb{Z}_{>0}$ and $q \in \mathbb{Z}_{>0}$ respectively. W_{down} and W_{up} of PHM experts can be computed as the sum of Kronecker product as follows:

$$\begin{aligned}\boldsymbol{W_{down}} &= \sum_{i=1}^{p} \boldsymbol{A_i} \otimes \boldsymbol{B_i} \\ \boldsymbol{W_{up}} &= \sum_{i=1}^{q} \boldsymbol{C_i} \otimes \boldsymbol{D_i}\end{aligned} \tag{2}$$

where $\boldsymbol{A_i} \in \mathbb{R}^{p \times p}$, $\boldsymbol{B_i} \in \mathbb{R}^{\frac{d}{p} \times \frac{d_{mid}}{p}}$, $\boldsymbol{C_i} \in \mathbb{R}^{q \times q}$ and $\boldsymbol{D_i} \in \mathbb{R}^{\frac{d_{mid}}{q} \times \frac{d}{q}}$.

2.2 Low-Rank Sharing Mechanism

Considering that each expert needs to deal with the shared features between different tasks and the features specific to each task, we define A_i and C_i as global matrices, which aggregate shared information to reflect task commonality, and B_i and D_i are as local matrices, which serve to capture specific task information. Because low-dimensional reparameterization can significantly improve the

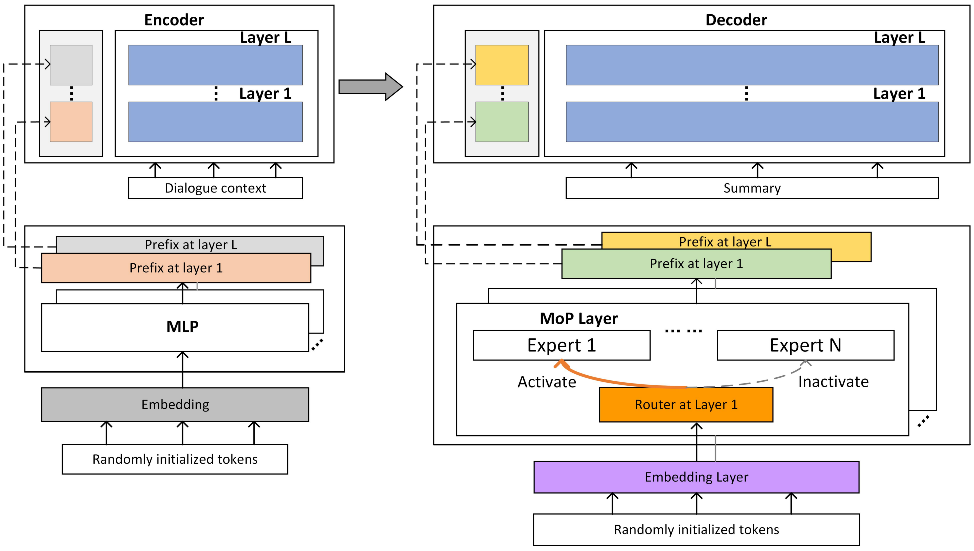


Figure 1: Overview of MOP Reparameterization Network. The reparameterized prefixes are prepended to self-attention modules of the decoder.

stability of prefix-tuning, we propose to decompose central matrices, e.g., $A_i \in \mathbb{R}^{\frac{d}{p}\times\frac{d_{mid}}{p}}$ is decomposed into two low-rank weights $\boldsymbol{l_i} \in \mathbb{R}^{\frac{d}{p}\times r}$ and $\boldsymbol{e_i} \in \mathbb{R}^{r\times\frac{d_{mid}}{p}}$, where r is the rank of the matrix. We name $\boldsymbol{l_i}$ as LoRL (Low-rank layer-share) weight, which is shared by all layers of the same experts. The $\boldsymbol{e_i}$ is named as LoRE (low-rank expert-share) weight, which is shared by all experts of the same layer. This low-rank sharing mechanism can effectively reduce the sharing of parameters between different experts, which can also greatly improve the efficiency of the model.

Based on the above formulation, we introduce MOP, which is a low-rank mixture of experts based on PHM, the weights of experts in MOP can be defined as:

$$\boldsymbol{W_{down}} = \sum_{i=1}^{p} \boldsymbol{A_i} \otimes \boldsymbol{B_i} = \sum_{i=1}^{p} (\boldsymbol{l_i e_i^T}) \otimes \boldsymbol{B_i}$$
$$\boldsymbol{W_{up}} = \sum_{i=1}^{q} \boldsymbol{C_i} \otimes \boldsymbol{D_i} = \sum_{i=1}^{q} (\boldsymbol{l_i e_i^T}) \otimes \boldsymbol{D_i} \tag{3}$$

2.3 MOE for Prompt reparameterization

Prefix-tuning prepends tunable prefix vectors to the parameters of multi-head attention (i.e. keys and values) at each Transformer layer. In the original setting, the prefix vectors P^{l_i} of the i-th attention head in the l-th layer are reparameterized by a two-layer feed-forward network:

$$P^{l_i} = MLP^{l_i}(X') = W_{up}^{l_i}\phi(W_{down}^{l_i}(X')) \tag{4}$$

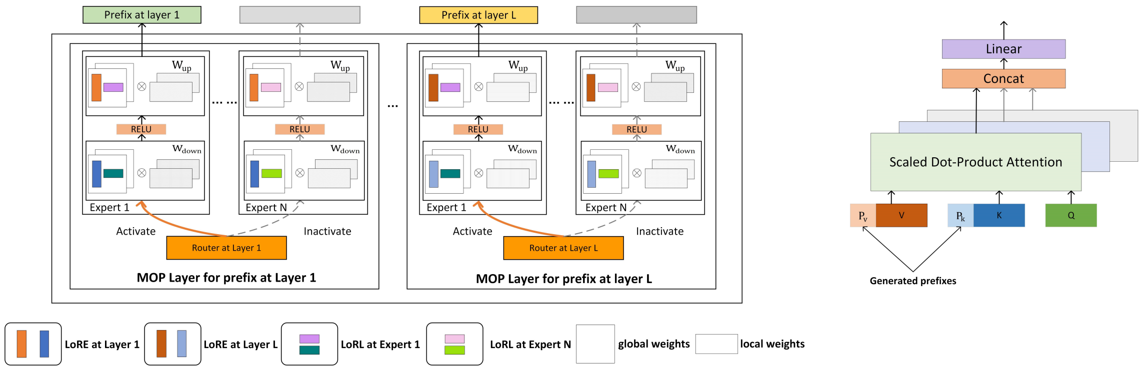
where $W_{down} \in \mathbb{R}^{d\times d_{mid}}$, $W_{up} \in \mathbb{R}^{d_{mid}\times d_h}$, and $X' \in \mathbb{R}^{n\times d}$ is the randomly initialized embedding matrix of the prefix X. The prefixes are transformed two times by Eq. 4 to get the expanded key $P_K^{l_i}$ and expanded value $P_V^{l_i}$. Then, they are concatenated with the original key and value, and the output of the attention layer is computed as:

$$A^i = Attn(Q^{l_i}, concat(P_K^{l_i}, K^{l_i}), concat(P_V^{l_i}, V^{l_i})) \tag{5}$$

where $Q^{l_i} \in \mathbb{R}^{m\times d_h}, K^{l_i} \in \mathbb{R}^{m\times d_h}, V^{l_i} \in \mathbb{R}^{m\times d_h}$ are original query, key and value, Fig 2(b) shows the details. For prefix-tuning, there are three types of attention: the self-attention of encoder, the self-attention of decoder, and the cross-attention of decoder. According to the experiments, we choose to use the MOP instead of MLP in the self-attention of decoder. While for the remaining two attentions, we still use the original MLP network. In this way, the model can perform multi-task learning in a parameter-efficient form. Figure 1 shows our overall model framework.

2.4 Training Objective

Given input dialogue context X, parameters of PLM θ, trainable prefix parameters θ_p, the summarization optimization objective is to minimize the negative log-likelihood of generating the target



(a) MOP (b) Multi-Head Attention

Figure 2: MOP frameweork:(a) in MOP, W_{down} and W_{up} are used to do down projection and up projection, respectively. They can be calculated as a sum of Kronecker products of a series of global matrices and local matrices. The global matrix can be divided into two rank-one weights LoRE(low-rank expert-share) and LoRL(low-rank layer-share), LoRE is shared by all experts at the same level, and LoRL is shared by the same experts across levels; The local matrix is unique to each expert at each level. This mechanism allows us to achieve highly flexible adjustment. (b) in each Transformer block, P_K, P_V generated via MOP are prepended to the original key K and value V for the query Q to attend to.

summary $Y = \{y_i, \cdots, y_{|Y|}\}$:

$$L_{nll}(\theta, \theta_p) = \sum_{i}^{|Y|} log\mathbb{P}(y_0|X, y_1, \cdots, y_{i-1}) \quad (6)$$

In training stage, we keep θ frozen and only optimize θ_p.

3 EXPERIMENTAL SETUP

3.1 Dataset

We collect our dialogue-summary datasets, **MultiSum**, from the logs on a large-scale customer service corpus. The dialogues are between users and customer service agents, and the summaries are written by agents. To perform multi-scenario learning, we choose 5 different business scenarios, including *Taxi*, *Ticket*, *E-Commerce*, *Take-out*, *Food*. The statistics of the data are given in Table 1. We divide the sizes of training, valid and test set to 8:1:1. To the best of our knowledge, MultiSum is the first to explore multi-task/domain summarization generation. We will release our data, code and pre-trained models after blind review.

Domains	Size	Dialog.len	Summ.len
Taxi	31,258	299.49	27.79
Ticket	10,869	204.64	22.60
E-Commerce	35,795	255.91	16.47
Take-out	28,707	189.925	42.37
Food	20,824	241.30	27.02

Table 1: Details of MultiSum. "Dialog.len" denotes the average length of dialogues, "Summ.len" denotes the average length of summaries.

3.2 Backbone and Baselines

Considering the deployment cost and model performance, we choose the Chinese generative pretraining language model T5-pegasus-base as the backbone network, which takes mT5 as the infrastructure and initial weight and pre-trains in a way similar to PEGASUS. Based on the public available pre-trained checkpoints, we conducted experiments to compare **MOP** with several general multi-task learning baselines and some novel parameter-efficient proposals:

MTL-vanilla: The standard practice of full-parameter fine-tuning T5-pegasus-base for multi-task summarization, which we refer to as MTL-vanilla(Raffel et al., 2019).

MTL: On the basis of the MTL-vanilla, we have designed templates manually, which are the natural language descriptions of conversation scenes(Brown et al., 2020b). For example, for the dialogue in the Take-out scenario, we designed the template as "The conversation comes from the Take-out business". Similar to MTL-vanilla, we perform full-parameter fine-tuning on the MultiSum dataset and we refer this kind of multi-task learning model as MTL.

prefix-tuning: We take T5-pegasus-base as the backbone network and fine-tune the model for multi-task learning under prefix-tuning paradigm(Li and Liang, 2021a), which only tunes a small number of prefix vectors while keeping the PLM frozen during training stage. The prefix vectors are initialized in random and all samples share the prefix vectors with a length of 40.

MTL-prompt: Prompt-tuning is proposed by Lester et al(Lester et al., 2021). , which prepends a sequence of soft prompt tokens to the input and only tunes the soft prompt for adaptation. We set the prompt length to 40, which is shared by all samples for multi-domain learning.

HyperFormer++: We compare our method with HyperFormer++ (Karimi Mahabadi et al., 2021), the state-of-the-art adapter-based method for multi-task learning, which use HyperNetwork to generate adapters for each task and add them after the feed-forward modules.(Houlsby et al., 2019)

HyperPrefix: HyperPrefix is a fresh approach proposed recently(Zhang et al., 2022). On the basis of prefix-tuning and hypernetwork, it uses a shared hypernetwork that takes trainable hyper-embeddings as input and outputs weights as prefix vectors. Since the position and task information have been considered in the embedding stage, this method can conduct multi-domain learning in a lightweight way.

We use the ROUGE metrics(Li and Liang, 2021b) to quantitatively evaluate the performance of models. ROUGE (Recall-Oriented Understudy for Gisting Evaluation) evaluates the n-gram overlap in the generated summary against the reference. We report F-1 scores of ROUGE-1 (R-1), ROUGE-2 (R-2) and ROUGE-L (R-L) on MultiSum.

3.3 Implementation Details

Our models are built on T5-pegasus-base (220M) and use jieba as the tokenizer to tokenize the input dialogue. During prefix reparameterization, we set $d = 768, d_{mid} = 128$ for all the experiments. For MOE network, following the recipe from (Chi et al., 2022), we set the number of experts to 5. For model training, we set maximum number of epochs as 50 and use early stopping to prevent over-fitting. For multi-task learning, we combine the training data of all tasks with temperature mixing (we set the temperature as 2). We save a checkpoint every 1000 steps and report results on a single checkpoint with the highest average validation performance across all tasks. Appendix will provide the detailed hyperparameters for **MOP** training.

3.4 Main Results

Table 2 presents the results of our experiments on MultiSum, where we treat each business scenario as a separate task and train a joint model for multi-task learning. We compare our approach with some strong full-parameter fine-tuning summarization models and some parameter-efficient baselines, including HyperFormer++ and HyperPrefix. Our results show that MTL with additional auxiliary information achieves higher ROUGE scores on most scenarios compared to MTL-vanilla, at the cost of increased parameter quantity. Prefix-tuning and MTL-prompt perform worse than full-parameter fine-tuning, due to the lack of effective strategies for adapting to complex multi-scenario summarization and the difficulty of achieving good performance with limited parameters. Recent works have attempted to conduct multi-task learning in a parameter-effective way, such as combining hypernet with prefix-tuning or adapter, which have shown promising results. Compared to the best performing hyper-based model, our model improves by 7.73%, 10.57%, 8.27% for *Ticket* domain, 5.13%, 8.28%, 5.45% for *Food* domain, 3.49%, 4.38%, 3.62% for *Taxi* domain, 2.88%, 3.87%, 3.01% for *Take-out* domain and 3.69%, 4.89%, 4.20% for *E-Commerce* domain. Relative to MTL-vanilla , our model improves by 9.28%, 13.28%, 9.89% for *Ticket* domain, 8.67%, 16.55%, 8.50% for *Food* domain, 1.82%, 3.66%, 1.85% for *Taxi* domain, 2.89%, 5.29%, 1.88% for *Take-out* domain and 6.24%, 8.12%, 5.48% for *E-Commerce* domain. In addition, our **MOP** still has higher ROUGE scores than strong baseline MTL in all scenarios . All results suggest that the performance of our model reaches new state-of-the-art.

4 QUALITATIVE ANALYSIS

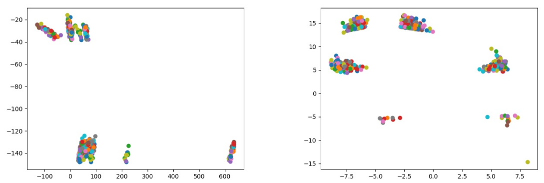
We design a series of experiments to verify the effectiveness of our proposed framework compared to existing methods.

4.1 Effect of Sparse Mixture of Experts

To shed light on how the MOP benefits multi-scenario dialogue summarization, we peek into the MOP by visualizing the generated prefix vectors. Here, the prefix vectors are mapped to the 2D projections via PCA. We use the same 5000 examples which are randomly selected from MultiSum D_{dev}. Fig 3 shows the visualization of the prefix vectors parameterized through MOP and Fig 4 is the

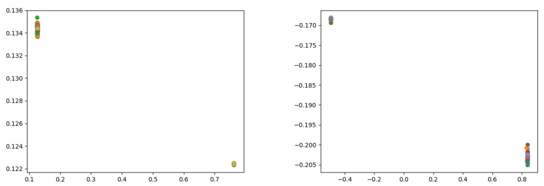
Models	Taxi			Ticket			E-Commerce			Take-out			Food			Average		
	R-1	R-2	R-L	R-1	R-2	R-L	R-1	R-2	R-L	R-1	R-2	R-L	R-1	R-2	R-L	R-1	R-2	R-L
MTL-vanilla	52.61	38.36	49.14	41.22	31.16	39.79	41.75	25.64	38.99	36.25	23.50	34.69	48.05	32.29	46.29	43.98	30.19	41.78
MTL	51.83	38.72	48.49	42.99	33.92	41.64	43.10	26.86	39.77	36.03	24.56	34.60	51.14	36.30	49.04	45.02	32.07	42.71
prompt-tuning	45.49	31.67	42.37	30.59	20.25	29.29	36.38	20.69	33.55	32.19	20.63	30.85	37.84	24.70	36.60	36.50	23.59	34.53
prefix-tuning	50.49	36.54	47.14	40.01	30.32	38.67	42.29	25.72	38.87	34.31	22.71	32.52	48.10	32.56	46.09	43.04	29.57	40.66
HyperFormer++	51.99	37.74	48.61	41.41	30.90	39.84	42.84	26.45	39.67	36.03	23.43	34.20	49.02	33.74	46.98	44.26	30.46	41.86
HyperPrefix	51.78	38.09	48.30	41.82	31.92	40.38	42.77	26.43	39.47	36.25	23.82	34.31	49.66	34.76	47.63	44.46	31.00	42.02
MOP (ours)	**53.58**	**39.76**	**50.04**	**45.05**	**35.29**	**43.72**	**44.35**	**27.72**	**41.12**	**37.30**	**24.75**	**35.34**	**52.21**	**37.63**	**50.22**	**46.50**	**33.03**	**44.09**
w/o low-rank	52.88	39.26	49.51	43.35	33.83	42.10	42.38	26.08	39.31	38.04	25.00	36.29	50.42	35.73	48.46	45.41	31.98	43.13

Table 2: ROUGE scores of all models for multi-scenario summarization on MultiSum.



(a) layer12 (b) Layer10

Figure 3: Visualization of prefix representations reparameterized by MOPs.



(a) layer12 (b) Layer10

Figure 4: Visualization of prefix representations reparameterized by MLP.

visualization of MLP. We can see that the prefix vectors generated by MOP present a more sparse distribution in space, and we can also observe the clustering. While the M LP-reparameterized prefix vectors still reside in a narrow subset of the entire space. Previous work (Su et al., 2022) has proved that congestion in the representation space (anisotropic distrbution) will lead to the degeneration of neural language models, which is because that the model devotes most attention to a small part local features while ignoring other global auxiliary information. Conversely, MOPs disperse the prefix vectors in a relatively sparse space, which encourages the model to obtain global features, identify specific information, which is beneficial to multi-task learning. Specifically, the dispersed prefix vectors enable the model to capture a wider range of information and avoid overfitting to specific tasks. Overall, the design of MOPs promotes the model's ability to achieve feature differentiation, and improve its performance in multi-task learning.

Model	R-1	R-2	R-L
MOP(ours)	46.50	33.03	44.09
w/o LoRL	45.48	31.94	43.12
w/o LoRE	45.29	31.81	42.94
w/o LoRL & LoRE	44.75	31.45	42.41

Table 3: Average F1 scores on 5 domains of MultiSum dataset."LoRL" denotes low-rank layer-share weights and "LoRE" means low-rank expert-share weights, "w/o LoRL and LoRE" means the removal of low-rank sharing mechanism.

4.2 Effect of the Sharing Mechanism

Table 3 shows the effect of two sharing mechanisms, i.e., LoRL and LoRE. We remove LoRL and LoRE one-by-one from our model. As we can see, the removal of the LoRL makes the R-1, R-2, and R-L drop by 3.03, 1.09, 0.97 points, which suggests that low-rank layer-share features can effectively accumulate the "layer inherent knowledge" by allowing all layers of the same expert to modify according to optimization objectives. Besides, after we get rid of the LoRE, R-1, R-2, and R-3 drop by 1.21, 1.23, 1.15 points respectively, which demonstrates that low-rank expert-share features can effectively obtain the common features among experts, so as to realize the communication across experts. After removing the LoRE, the connection between experts would be interrupted, our experts will work in complete isolation, making it unable to perform multi-scenario sharing well. Finally, we remove the LoRL and LoRE at the same time, which leads to 44.75%, 31.45%, and 42.41% for R-1, R-2, and R-L.

4.3 Robustness Analysis

Prefix-tuning is sensitive to the initialization of the prefix, particularly random initialization. Fig 5 shows the robustness of **MOP**, **MOP** w/o low-rank, and prefix-tuning. We conduct experiments

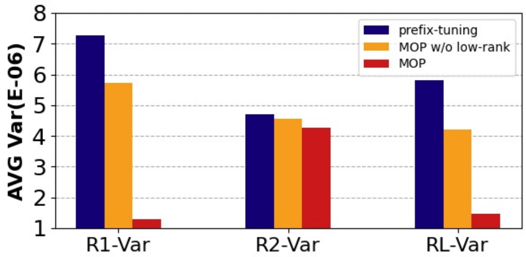


Figure 5: Average variance of F1 scores on MultiSum. "w/o low-rank" means the removal of low-rank decomposition.

Model	#Total params	Trained params	R-L
MTL-vanilla	1.000	100%	41.78
prefix-tuning	1.027	2.734%	40.66
prompt-tuning	1.001	0.112%	31.54
HyperFormer++	1.023	2.320%	41.86
MOP (ours)	1.025	2.514%	44.09

Table 4: Proportion of different models' trainable parameter quantity to MTL-vanilla and their average F1 scores on MultiSum.

on three models with three different random seeds in the same setting. The low-rank decomposition significantly enhances the robustness and stability of MOP for initialization, as evidenced by the much lower average variance of F1 scores compared to prefix-tuning and MOP without low-rank. In addition, we find the average variance of MOP w/o low-rank is also slightly lower than that in prefix-tuning. We contribute this reduction of sensitivity to initialization to the strong learning ability of SMOE structure. The experiment proves that our **MOP** has high robustness.

4.4 Parameter Scale of Models

In this section, we compare the number of parameters of **MOP** with other baseline multi-domain joint models. Taking the parameter quantity of T5-pegasus-base (275M) as a reference, we show the proportions of total parameter quantity and trainable parameter quantity of each method and their average ROUGE scores on the 5 domains of MultiSum (*Food, Ticket, Taxi, Take-out, E-Commerce*) in Table 4. Among the parameter efficient methods, prompt-tuning only tunes the continues vectors prepended before input embeddings and require the least trainable parameters, only 0.112%, but its performance is more than poor. prefix-tuning, HyperFormer++ and our **MOP** greatly reduce the storage space of the model with frozen PLM and a small number of trainable parameters, which are applied to each layer of the model and contribute to a trade-off between performance and parameter quantity. Additionally, our method, achieves better results with fewer parameters compared with prefix-tuning and also greatly outperforms full-parameter fine-tuning model. Specifically, our **MOP** performs 5.55% better on R-L than MTL-vanilla, using only 2.51% of its parameters. In addition, we compare **MOP** with the state-of-the-art lightweight multi-task learning model HyperFormer++. Please note that our model takes into account the total number of all experts' parameters when calculating the trainable parameters. Even so, the number of parameters of our **MOP** is only slightly higher than that of HyperFormer++, and the performance of our method is superior. All these points show our **MOP** has achieved a better trade-off between parameter efficiency and performance.

4.5 Impact of Prefix Length

We set different lengths of continuous prefix vectors to test the performance of **MOP** and prefix-tuning on MultiSum dataset and report their average F1 scores. As shown in Fig 6, among these setting candidates, we find 40 is the best length to make the F1 scores of two models reach the peak. Before reaching the optimal length, we observe that the performance of the model shows a positive correlation with the length of prefix. We attribute this phenomenon to the insufficient trainable parameters. Moreover, we find the increase of the prefix length has a greater impact on improving the performance of our **MOP** model, which indicates that **MOP** has stronger learning ability. When exceeding the optimal length, the trainable parameters of the model reach saturation, at this point the increase of length will increase burden of the model, and eventually cause the decline of the model performance, while this degradation is less obvious in **MOP** model, which proves the robustness of our model.

4.6 Case Study

Fig1 in appendix A shows two examples from MultiSum, For example one from *Take-out* domain, summary generated by MTL omits the final solution, i.e. merchant refund. Prefix-tuning generates incorrect solution, distorts the fact that customers do not accept red envelopes. These factual errors significantly affect the quality of the sum-

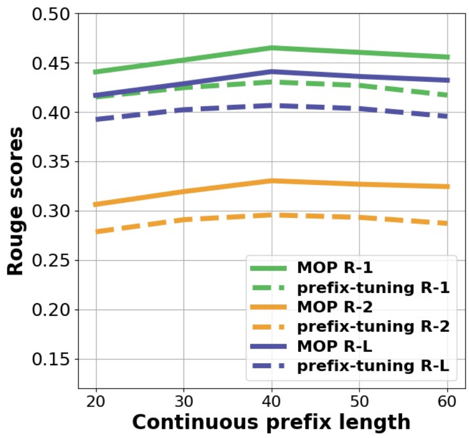


Figure 6: Average F1 scores of **MOP** and prefix-tuning with different lengths of the continuous prefix on Multi-Sum.

mary. For example two, both summaries generated by MTL and prefix-tuning include the error message of "punishing the driver", which shows that summaries generated by these two methods fail to comprehensively cover all important information.

Compared to the above two models, our method generates summaries with similar events and faithful descriptions compared with the gold summary. In example one, **MOP** accurately shows the "merchant refund" scheme, and in example 2, our method reflects the relevant content of "don't punish the driver". This indicates that summaries generated by **MOP** are more reliable, thanks to the effectiveness of MoE and efficient low-rank sharing mechanism.

5 RELATED WORK

Multi-scenario Dialogue Summarization Multi-task learning jointly optimizes models on several tasks(Vandenhende et al., 2020). By sharing representations between these tasks, we enable model to generalize better on each task. Particularly, multi-scenario summarization is a kind of multi-task learning(Wang et al., 2021), which trains model on mixed multi-scenario data to achieve good performance in each scenario. Despite efforts on designing models for improved joint learning(Kokkinos, 2016) (Misra et al., 2016) (Rosenbaum et al., 2017), the scope of this study is rather limited. For instance, in the LLM area, fine-tuning large-scale language models with full parameters is still the mainstream paradigm.(Raffel et al., 2019) (Zhang et al., 2019). Our work explores a lightweight approach tomulti-scenario summarization, effectively addressing the issue of over-parameterization and filling a gap in relevant research.

Prompt learning for Text Generation The idea of prompt learning is first proposed in GPT3(Brown et al., 2020a), where it guides a large language model to different tasks by prepending task-related natural languague description. Prefix-tuning(Li and Liang, 2021a) extends this idea to continuous tokens. It prepends trainable continuous tokens (prefix) to the input and hidden stats of each Transformer layer. Each prefix is drawn from a newly initialized trainable matrix **P**, while other parameters of the PLM remain unchanged during training. To further simplify prompt-tuning, Lester et al(Lester et al., 2021). proposes a strategy that only adds soft prompts to the input layer. While prompt-based methods show promise for adapting PLMs, challenges remain. Prefix-tuning is sensitive to initialization and unstable during training. To address these issues, we conduct multi-scenario summarization using prefix-tuning, stabilize the training process through inherent bias representation in multi-task learning, and introduce low-rank decomposition to enhance robustness.

Prompt learning with MoE Numerous studies have shown that models with more parameters typically yield better performance. To increase model capacity without added computational overhead, exploring scaling properties with MoE, introduced by (Jacobs et al., 1991), is a promising direction. There have been many existing works that combine MoE and PLMs for research(Shazeer et al., 2017b) (Fedus et al., 2021) (Lepikhin et al., 2021) (Lewis et al., 2021). However, few of them focus on parameter-efficient MoE. Also, there are few works that attemp to combine the MoE with prompt learning.

6 CONCLUSION

In this paper, We propose a lightweight low-rank MOE network for Prompt reparameterization in multi-scenario summarization, which integrates MoE into the prefix reparameterization process and achieves expert integration. Our proposed low-rank sharing weights (LoRL and LoRE) enable cross-layer and cross-expert knowledge sharing, effectively reducing the number of parameters while improving performance. Experimental results demonstrate that our model outperforms all strong baselines and achieves significant progress in multi-scenario summarization.

7 Limitations

Our work still has certain limitations.

First, although we designed an effective MOP mechanism, the performance of the joint model trained on multiple scenarios still has a gap compared to models fine-tuned for each specific scenario. This suggests that interference still exists due to the differences in data distribution across scenarios.

Second, reparameterizing prompts using a mixture of experts network reduces the number of trainable parameters, but it inevitably increases the deployment cost of the mixture of experts' parameters.

Finally, our mixture of experts reparameterization network can be applied to various parameter-efficient fine-tuning methods. We only explored reparameterizing prompts using a mixture of experts network, and further experiments are needed to verify the role of the mixture of experts network in other parameter-efficient fine-tuning methods.

References

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D 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 Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020a. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc.

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. 2020b. Language models are few-shot learners.

Rich Caruana. 1998. *Multitask learning*. Springer.

Zewen Chi, Li Dong, Shaohan Huang, Damai Dai, Shuming Ma, Barun Patra, Saksham Singhal, Payal Bajaj, Xia Song, Xian-Ling Mao, Heyan Huang, and Furu Wei. 2022. On the representation collapse of sparse mixture of experts.

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.

William Fedus, Barret Zoph, and Noam Shazeer. 2021. Switch transformers: Scaling to trillion parameter models with simple and efficient sparsity.

Barry Haddow and Philipp Koehn. 2012. Analysing the effect of out-of-domain data on SMT systems. In *Proceedings of the Seventh Workshop on Statistical Machine Translation*, pages 422–432, Montréal, Canada. Association for Computational Linguistics.

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.

Robert A. Jacobs, Michael I. Jordan, Steven J. Nowlan, and Geoffrey E. Hinton. 1991. Adaptive mixtures of local experts. *Neural Computation*, 3(1):79–87.

Rabeeh Karimi Mahabadi, Sebastian Ruder, Mostafa Dehghani, and James Henderson. 2021. Parameter-efficient multi-task fine-tuning for transformers via shared hypernetworks. 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 565–576, Online. Association for Computational Linguistics.

Alex Kendall, Yarin Gal, and Roberto Cipolla. 2017. Multi-task learning using uncertainty to weigh losses for scene geometry and semantics.

Iasonas Kokkinos. 2016. Ubernet: Training a 'universal' convolutional neural network for low-, mid-, and high-level vision using diverse datasets and limited memory.

Dmitry Lepikhin, HyoukJoong Lee, Yuanzhong Xu, Dehao Chen, Orhan Firat, Yanping Huang, Maxim Krikun, Noam Shazeer, and Zhifeng Chen. 2020. Gshard: Scaling giant models with conditional computation and automatic sharding.

Dmitry Lepikhin, HyoukJoong Lee, Yuanzhong Xu, Dehao Chen, Orhan Firat, Yanping Huang, Maxim Krikun, Noam Shazeer, and Zhifeng Chen. 2021. {GS}hard: Scaling giant models with conditional computation and automatic sharding. In *International Conference on Learning Representations*.

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

Mike Lewis, Shruti Bhosale, Tim Dettmers, Naman Goyal, and Luke Zettlemoyer. 2021. Base layers: Simplifying training of large, sparse models.

Xiang Lisa Li and Percy Liang. 2021a. Prefix-tuning: Optimizing continuous prompts for generation.

Xiang Lisa Li and Percy Liang. 2021b. 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, Online. Association for Computational Linguistics.

Shikun Liu, Edward Johns, and Andrew J. Davison. 2018. End-to-end multi-task learning with attention.

Xiao Liu, Kaixuan Ji, Yicheng Fu, Weng Lam Tam, Zhengxiao Du, Zhilin Yang, and Jie Tang. 2021. P-tuning v2: Prompt tuning can be comparable to fine-tuning universally across scales and tasks.

Rabeeh Karimi Mahabadi, James Henderson, and Sebastian Ruder. 2021. Compacter: Efficient low-rank hypercomplex adapter layers.

Ishan Misra, Abhinav Shrivastava, Abhinav Gupta, and Martial Hebert. 2016. Cross-stitch networks for multi-task learning.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2019. Exploring the limits of transfer learning with a unified text-to-text transformer.

Clemens Rosenbaum, Tim Klinger, and Matthew Riemer. 2017. Routing networks: Adaptive selection of non-linear functions for multi-task learning.

Victor Sanh, Albert Webson, Colin Raffel, Stephen H. Bach, Lintang Sutawika, Zaid Alyafeai, Antoine Chaffin, Arnaud Stiegler, Teven Le Scao, Arun Raja, Manan Dey, M Saiful Bari, Canwen Xu, Urmish Thakker, Shanya Sharma Sharma, Eliza Szczechla, Taewoon Kim, Gunjan Chhablani, Nihal Nayak, Debajyoti Datta, Jonathan Chang, Mike Tian-Jian Jiang, Han Wang, Matteo Manica, Sheng Shen, Zheng Xin Yong, Harshit Pandey, Rachel Bawden, Thomas Wang, Trishala Neeraj, Jos Rozen, Abheesht Sharma, Andrea Santilli, Thibault Fevry, Jason Alan Fries, Ryan Teehan, Tali Bers, Stella Biderman, Leo Gao, Thomas Wolf, and Alexander M. Rush. 2021. Multitask prompted training enables zero-shot task generalization.

Ozan Sener and Vladlen Koltun. 2018. Multi-task learning as multi-objective optimization. In *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc.

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

Noam Shazeer, *Azalia Mirhoseini, *Krzysztof Maziarz, Andy Davis, Quoc Le, Geoffrey Hinton, and Jeff Dean. 2017b. Outrageously large neural networks: The sparsely-gated mixture-of-experts layer. In *International Conference on Learning Representations*.

Yixuan Su, Tian Lan, Yan Wang, Dani Yogatama, Lingpeng Kong, and Nigel Collier. 2022. A contrastive framework for neural text generation.

Tianxiang Sun, Zhengfu He, Qin Zhu, Xipeng Qiu, and Xuanjing Huang. 2022. Multi-task pre-training of modular prompt for few-shot learning.

Simon Vandenhende, Stamatios Georgoulis, Marc Proesmans, Dengxin Dai, and Luc Van Gool. 2020. Revisiting multi-task learning in the deep learning era. *CoRR*, abs/2004.13379.

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

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. 2018. Glue: A multi-task benchmark and analysis platform for natural language understanding.

Jindong Wang, Cuiling Lan, Chang Liu, Yidong Ouyang, Tao Qin, Wang Lu, Yiqiang Chen, Wenjun Zeng, and Philip S. Yu. 2021. Generalizing to unseen domains: A survey on domain generalization.

Jason Wei, Maarten Bosma, Vincent Y. 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.

Aston Zhang, Yi Tay, Shuai Zhang, Alvin Chan, Anh Tuan Luu, Siu Cheung Hui, and Jie Fu. 2021. Beyond fully-connected layers with quaternions: Parameterization of hypercomplex multiplications with $1/n$ parameters.

Jingqing Zhang, Yao Zhao, Mohammad Saleh, and Peter J. Liu. 2019. Pegasus: Pre-training with extracted gap-sentences for abstractive summarization.

Zhengkun Zhang, Wenya Guo, Xiaojun Meng, Yasheng Wang, Yadao Wang, Xin Jiang, Qun Liu, and Zhenglu Yang. 2022. Hyperpelt: Unified parameter-efficient language model tuning for both language and vision-and-language tasks.

A Appendix

Example one: dialogue in the Take-out scenario
U:这花怎么这么脏, 亏我还提前定了，我花了七八十块钱买给我闺蜜，这怎么拿得出手。(Why is this flower so dirty? I ordered it in advance. I spent 70 or 80 yuan to buy it for my girlfriend. How can I take it.) A：明白了亲亲。为您处理。亲亲您看给您赔偿红包可以吗？小美刚才看了一下确实是不好呢。(Understand . For you. May I show you the red envelope for compensation? Xiaomei just looked at it. It's really bad.) U：不可能。你叫商家拿回去吧。(Impossible. Tell the merchant to take it back.) A：小美为您申请一下。亲亲小美这边为您申请了赔偿您看可以接受吗？(Xiaomei applies for it for you. Do you think it is acceptable to apply for compensation for you here?) U：**不要。退钱**。(No. **Refund**.) A：好的，小美为您申请一下。(Ok, Xiaomei will apply for it for you.)
Generate Summary
Ground Truth： 客户表示花脏的很，安抚解释红包不接受，充值卡不接受，**商家自动退款，**安抚协商认可(The customer said that the flowers were very dirty, comforted and explained that the red envelope was not accepted, the recharge card was not accepted, **the merchant automatically refunded**, comforted and agreed.)
T5-pegasus-base：安抚致歉，补偿红包，不认可，充值卡不认可。(Appease and apologize, compensate for the red envelope, do not approve, do not approve the recharge card.)
Prefix-tuning:安抚致歉，补偿红包，不认可，充值卡认可，结案。(Appease and apologize, compensate for red packets, do not approve, recharge card approval, and close the case.)
LAD：客户投诉，安抚致歉，补偿红包，不认可，充值卡不认可，**商家退款**，认可。(LAD: customer complaints, appeasement and apology, compensation for red envelopes, non-recognition, non-recognition of recharging cards, **refund of merchants**, recognition.)
Example two: dialogue in dache scenario
U：这个订单实际支付怎么比预估费用超出这么多。(How can the actual payment of this order exceed the estimated cost so much.) A：小美这边需要与您核实几个问题，您在中途是否与司机口头更改过目的地或者给司机指定路线行驶呢。（Xiaomei needs to check with you a few questions. Have you changed the destination orally with the driver or assigned the route to the driver in the midway.） U：没有啊，没有跟司机说过话。(No, I haven't talked to the driver.) A：亲核实轨迹是因为司机绕路了导致费用超出的哦。亲您看这样，小美为您操作部分退款到您的原支付账户里1-7个工作日内到账在帮您申请补偿打车红包可以吗。(Verify the track in person because the driver bypassed the road, which caused the cost to exceed. Look at this, Xiaomei will refund part of your operation to your original payment account within 1-7 working days. Can I help you apply for compensation for the red packet of taxi.) U：可以。(Sure.) A：亲您看需要对司机进行投诉处罚吗。(Do you think it is necessary to complain and punish the driver.) U：**算了，退钱就行**，人家也不容易。(**Forget it, just refund the money**. It's not easy for others.) A：好的亲非常感谢您的理解与支持呢。(Well, thank you very much for your understanding and support.)
Generate Summary
Ground Truth：核实轨迹司机绕路，部分退款补偿红包处罚司机，**乘客表示算了**。(Verify that the track driver detours, and punish the driver with some refunds, compensation and red packets. **The passenger said that it was okay**.)
T5-pegasus-base：核实司机绕路，处罚司机，补偿红包，认可。(Verify the driver's detour, punish the driver, compensate for the red packet, and approve.)
Prefix-tuning：核实司机绕路，改为预估价，处罚司机，补偿红包，认可。(Verify the driver's detour, change to the estimated price, punish the driver, compensate for the red packet, and approve.)
LAD：核实司机绕路，部分退款， 补偿红包，**乘客表示不用处罚司机**。(Verify that the driver detours, refunds part of the money and compensates for the red packet. **The passenger said that the driver should not be punished**.)

Figure 1: Case study for two examples from MultiSum dataset. We present the dialogue context, ground truth, MTL prediction, prefix-tuning prediction and our **MOP** prediction.