MoSLD: A Extremely Parameter-Efficient Mixture-of-Shared LoRAs for Multi-Task Learning

Anonymous ACL submission

Abstract

Recently, LoRA has emerged as a crucial technique for fine-tuning large pre-trained models, yet its performance in multi-task learning scenarios often falls short. In contrast, the MoE architecture presents a natural solution to this issue. However, it introduces challenges such as mutual interference of data across multiple domains and knowledge forgetting of various tasks. Additionally, MoE significantly increases the number of parameters, posing a computational cost challenge. Therefore, in this paper, we propose MoSLD, a mixture-of-shared-LoRAs model with a dropout strategy. MoSLD addresses these challenges by sharing the upper projection matrix in LoRA among different experts, encouraging the model to learn general knowledge across tasks, while still allowing the lower projection matrix to focus on the unique features of each task. The application of dropout mitigates parameter overfitting in LoRA. Extensive experiments demonstrate that our model exhibits excellent performance in both single-task and multi-task scenarios, with robust out-of-domain generalization capabilities.

1 Introduction

The emergence of Large Language Models (LLMs) has significantly advanced Natural Language Processing (NLP) technology, serving as a robust foundation with broad applicability (Touvron et al., 2023a,b; Ouyang et al., 2022). However, as the parameter scale increases, the process of full parameter fine-tuning (FP-tuning) demands substantial computational and memory resources. To strike a balance between resource requirements and effectiveness, the research community is increasingly turning to parameter-efficient fine-tuning (PEFT) methods, with LoRA emerging as the most prevalent and effective choice. Nevertheless, training an LLM via LoRA with multi-faceted capabilities faces significant challenges due to the differences and diversity inherent in various tasks. Figure 1 illustrates that while FP-tuning demonstrates competitive performance in a multi-task mixed training data setting, plain LoRA exhibits a drop. This decline underscores the challenge posed by the heterogeneity and imbalance in training data, resulting in interference between data from different tasks and consequently degrading the performance of plain LoRA on in-domain tasks. In essence, plain LoRA proves highly sensitive to the configuration of training data.

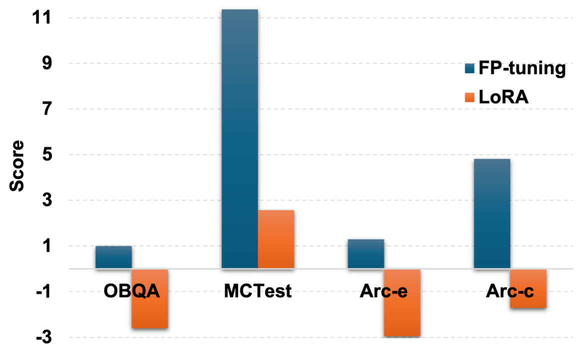


Figure 1: The increase between mixture setting and single setting for FP-tuning and LoRA on four datasets. The vertical axis is Score (mixture)-Score (single).

As we all know, MoE (Shazeer et al., 2017) has demonstrated remarkable advantages in amalgamating multiple capabilities. Particularly, the integration of MoE and LoRA (Hu et al., 2022) stands out as a promising approach to leveraging MoE in a parameter-efficient manner. This method preserves domain knowledge while significantly reducing training costs by introducing a limited number of domain-specific parameters (Dou et al., 2024; Luo et al., 2024; Liu et al., 2023). Presently, several works are devoted to applying MoE to LoRA. Some directly combine trained LoRAs linearly (Zhang et al., 2023; Huang et al., 2024), while others apply combinations of MoE and LoRA to different backbones (Chen et al., 2024; Dou et al., 2024). An-other approach involves training a LoRA module for each distinct task type and employing a routing mechanism to integrate the LoRA modules under a shared LLM (Feng et al., 2024). However, we contend that these methods inadequately address the issue of data conflicts across different domains during LoRA training. Three primary challenges emerge: (1) The MoE architecture emphasizes the unique attributes of each LoRA and overlooks the transfer of general knowledge between different LoRAs, thereby impeding cross-task generalization in LLMs; (2) The tasks (LoRAs) necessitate exhaustiveness; (3) Multiple LoRAs escalate the number of parameters and computational costs.

To solve these issues, in this paper, we propose a parameter-sharing method applied to the mixture-of-LoRAs, called MoSLD. The plain LoRA module comprises the upper projection matrix (A) and the lower projection matrix (B), which can be viewed as naturally decoupled general-feature and specific-feature matrices, respectively. Building upon the classic MoE architecture, we enable all experts at each layer to share a general-feature matrix while retaining the specific-feature matrix of each expert. This approach compels the model to capture shared general knowledge across various tasks to the fullest extent. The shared operation notably reduces the parameters of the MoE architecture, aligning with findings indicating parameter redundancy among experts (Fedus et al., 2022b; Kim et al., 2021). Despite the majority of parameters in the LoRA module being shared, differences can still be learned in each expert's specific-feature matrix due to the tight coupling between the general and specific features. We posit that this mechanism can adaptively generalize to any new task. Furthermore, recognizing that the general-feature matrix is updated more frequently than the specific-feature matrix during fine-tuning, and overfitting tends to occur in LoRA (Wang et al., 2024), we apply the dropout strategy to the general-feature matrix. This mitigates issues of parameter redundancy and unbalanced optimization.

In summary, our contributions are as follows: (1) We introduce a parameter-efficient MoSLD approach that disentangles domain knowledge and captures general knowledge by sharing a general-feature matrix, thus mitigating interference between heterogeneous datasets. (2) We implement a dropout strategy on the general-feature matrix to effectively mitigate overfitting and address the imbalance in directly optimizing MoE. (3) We conduct extensive experiments on various benchmarks to validate the effectiveness of our methods. Additionally, our approach demonstrates superior generalization to out-of-domain data.

2 Related Work

2.1 Mixture-of-Expert

The Mixture of Experts (MoE) functions as an ensemble method, conceptualized as a collection of sub-modules or experts, each tailored to process distinct types of input data. Guided by a router, each expert is selectively activated based on the input data type. This technique has garnered increasing attention and demonstrated remarkable performance across various domains, including computer vision, speech recognition, and multi-modal applications (Fedus et al., 2022a). Evolution of MoE techniques spans from early sample-level approaches (Jacobs et al., 1991) to contemporary token-level implementations (Shazeer et al., 2017; Riquelme et al., 2021), which have now become mainstream. Concurrently, some researchers (Zhou et al., 2022; Chi et al., 2022) are delving into the router selection problem within MoE. Notably, the majority of these endeavors aim to scale up model parameters while mitigating computational costs.

2.2 Mixture-of-LoRA

As LoRA gradually becomes the most common parameter-efficient fine-tuning method, researchers pay more attention to combining MoE and LoRA for more efficient and effective model tuning. Huang et al. (2024) and Feng et al. (2024) pioneer the approach of training several LoRA weights on upstream tasks and then integrating the LoRA modules into a shared LLM using a routing mechanism. However, these methods necessitate the training of numerous pre-defined LoRA modules. Chen et al. (2024) initially engage in instruction fine-tuning through sparse mixing of LoRA experts in the multi-modal domain, while Dou et al. (2024) split the LoRA experts into two groups to explicitly learn different capabilities for each group. These mixture-of-LoRA methods typically involve pre-defined hyperparameters that require careful selection, and they densely mix LoRA experts, significantly increasing computational costs. To tackle overfitting resulting from an excessive number of experts, Gao et al. (2024) allocate a varying number of experts to each layer. Wu et al. (2024) pro-

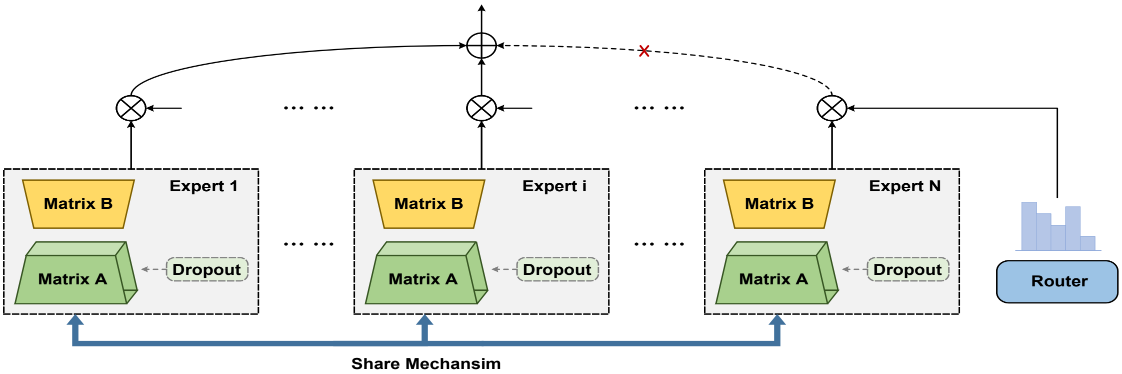


Figure 2: Overview of the share mechansim and dropout strategy in our MoSLD. Noted that the matrix A is shared among all experts in each layer.

pose MOLE, treating each layer of trained LoRAs as a distinct expert and implementing hierarchical weight control through a learnable gating function within each layer to tailor composition weights specific to a given domain's objectives. However, these approaches overlook the issue of data conflicts across different datasets during LoRA training. In this study, we conduct extensive experimental analysis for both single and mixture data settings.

3 Methodology

In this section, we describe our MoSLD from the sharing mechanism, dropout strategy and optimization details, as shown in Figure 2.

3.1 Sharing Mechanism of LoRAs

In the area of parameter-efficient fine-tuning, LoRA introduces the concept of training only two low-rank matrices as an alternative to dense layer updates. In other words, it reformulates the parameter fine-tuning process in LLMs as a low-rank decomposition. Specifically, the equation $W_0 + \Delta W = W_0 + BA$ captures this decomposition. Here, $W_0 \in \mathcal{R}^{d_{in} \times d_{out}}$ represents the parameter matrix of the pre-trained LLM, while $\Delta W \in \mathcal{R}^{d_{in} \times d_{out}}$ denotes the matrix updated during fine-tuning. The matrices $B \in \mathcal{R}^{d_{in} \times r}$ and $A \in \mathcal{R}^{r \times d_{out}}$ are low-rank and trainable.

Given a Transformer model with L layers, we allocate N_l experts for layer l and create N_l pairs of low-rank matrices $\{A_{i,l}, B_{i,l}\}_{i=1}^{N_l}$, where $A_{i,l}$ is initialized from a random Gaussian distribution and each $B_{i,l}$ is set to zero. It is worth noting that the matrix $A_{i,l}$ is shared among all experts in each layer, i.e., $A_{1,l} = A_{2,l} ... = A_{N_l,l} \quad (l \in L)$. In other words, the core idea is to share the matrix A as the general-feature matrix and keep matrix B as specific-feature matrics. In this way, we can only keep L central general-feature matrics for a L-layer MoE architecture. A router with a trainable weight matrix $W_l \in \mathcal{R}^{d_{in} \times N_l}$ is used to specify different experts for the input x. As in the original MoE, MoSLD selects the top K experts for computation, and the gate score S_l^k is calculated as follows:

$$S_l^k(x) = \frac{\mathrm{TopK}(\mathrm{softmax}(W_l x), K)_k}{\sum_{k=1}^{K} \mathrm{TopK}(\mathrm{softmax}(W_l x), K)_k} \quad (1)$$

3.2 Dropout Strategy

In order to alleviate the imbalance and over-fitting problems caused by frequent parameter matrix updates, we propose to apply the dropout strategy on the parameter matrix. That is, at each iteration, we take a certain probability p to discard the update in the parameter matrix. Specifically, we generate a binary mask matrix drawn from Bernoulli distribution with a mask probability p and the matrix is updated as follows:

$$\begin{aligned} \mathrm{Mask} &\sim \mathrm{Bernoulli}(p) \\ \mathbf{A}_l^{'} &= \mathrm{Mask} \odot \mathbf{A}_l \\ \widetilde{\mathbf{A}'}_l &= \mathbf{A}_l^{'}/(1-p) \end{aligned} \quad (2)$$

Note that the mask trick is only applied to the general-feature matrix.

3.3 The Overall Procedure

Our method is a combination of shared LoRA modules and MoE framework, as shown in Figure 3. Here, we apply our MoSLD on the matrix Q and matrix V of the self-attention layer:

$$h_l = W_0 x + \frac{\alpha}{r} \sum_{i=1}^{K} S_l^k(x) A_{i,l} B_{i,l} x \quad (3)$$

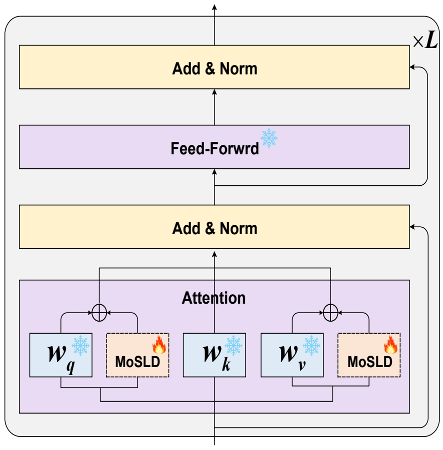


Figure 3: The overview of our proposed Mixture-of-Shared-LoRA with dropout strategy applied on W_q and W_v.

where $W_0 \in \{W_q, W_v\}$ and h_l is the output embedding. Besides,similar to previous sparse MoE works, the load balancing loss L_b is also applied on each MoE layer, which is formulated as:

$$
\begin{aligned}
L_b &= \sum_{k=1}^{K} c_k \cdot s^k \\
p_k &= \sum_{x \in X} \frac{e^{S^k(x)}}{\sum^k e^{S^k(x)}}
\end{aligned}
\tag{4}
$$

where c_k is the number of tokens assigned to the k-th expert.

4 Experimental Setup

4.1 Datasets

To evaluate the effectiveness of MoSLD, we conduct experiments on six commonsense reasoning datasets, including commonsense QA task (OBQA (Mihaylov et al., 2018), CSQA (Talmor et al., 2019)), reading comprehension task (Race (Lai et al., 2017), MCTest (Richardson et al., 2013)), and subject knowledge QA task (Arc-e (Clark et al., 2018), and Arc-c (Clark et al., 2018)). We denote the six datasets as $\{D_1, D_2, ..., D_6\}$, and we also create a mixed dataset D_{mix}, corresponding to the single setting and the mixture setting respectively. The dataset sizes are as follows for training and testing: 5457/500, 10962/1140, 10083/4934, 1330/147, 2821/2376, and 1418/1172. We allocate 10% of the training set for validation. For all datasets, we use answer accuracy as the evaluation metric.

4.2 Baselines

We compare MoSLD with three parameter-efficient fine-tuning methods: Prefix-tuning, LoRA, and MoLA. Additionally, we evaluate full-parameter fine-tuning.

Prefix-tuning (Li and Liang, 2021): This method involves incorporating soft prompts into each attention layer of the Large Language Model (LLM). These soft prompts are a series of virtual tokens pre-appended to the text. During fine-tuning, the LLM remains frozen, and only the virtual tokens are optimized.

LoRA (Hu et al., 2022): A popular parameter-efficient tuning approach widely used in LLM fine-tuning, LoRA leverages low-rank matrix decomposition of pre-trained weight matrices to significantly reduce the number of training parameters.

MoLA (Gao et al., 2024): A LoRA variant with layer-wise expert allocation, MoLA flexibly assigns a different number of LoRA experts to each Transformer layer.

4.3 Training Details

We take LLaMA2-7B (Touvron et al., 2023b) which contains 32 layers as our base model. For plain LoRA and its variants, the r is set to 8 and α is 16. Beside, the LoRA modules are used in matrix Q and matrix V in attention layers. Our MoSLD also follows the same settings. We allocate 8 experts to each layer for 1-8 layers, 6 experts to each layer for 9-16 layers, 4 experts to each layer for 17-24 layers, and 2 experts to each layer for the last 8 layers. The K of the selected experts is 2. For training details, we finetune models with 10 epochs and a peak of 3e-4 learning rate. The drop ratio applied to matrix A is set to 0.1. The batch size during model tuning is 128. The experiments are run on 16 NVIDIA A100 40GB GPUs.

4.4 Main Results

Table 1 presents the experimental outcomes of various baselines under both single and mixture settings across different datasets. Initially, we report the performance of models trained on individual datasets. LoRA notably outperforms other baselines, exhibiting improvements of 2.33% and 27.87% over FP-tuning (single) and Prefix-tuning (single), respectively. MoLA trails behind LoRA by 1.98%, indicating that simply combining LoRA and MoE does not confer an advantage in single in-domain datasets. After establishing a robust

Model		OBQA	CSQA	Race	MCTest	Arc-e	Arc-c	Avg
FP-tuning	single	75.00	75.74	80.62	39.05	72.39	60.63	67.24
	mixture	76.00	75.27	81.46	50.42	73.69	65.45	70.38
Prefix-tuning	single	47.76	42.65	53.77	25.19	45.65	35.50	41.70
	mixture	46.51	44.98	49.88	22.46	47.92	35.30	41.18
LoRA	single	75.40	76.33	76.06	53.10	73.82	62.71	69.57
	mixture	72.80	76.30	78.23	55.67	70.87	61.00	69.15
MoLA	single	74.60	77.23	75.29	44.90	72.73	60.80	67.59
	mixture	76.60	73.46	75.25	54.42	76.34	63.91	70.00
MoSL (our)	single	76.30	77.56	74.63	49.66	76.30	60.48	69.16
	mixture	**76.80** (+0.50)	75.02 (-2.54)	**74.69** (+0.06)	**58.50** (+8.84)	76.09 (-0.21)	**64.16** (+3.68)	**70.88** (+1.72)
MoSLD (our)	single	78.40	75.84	76.08	53.06	76.35	61.49	70.20
	mixture	**78.80** (+0.40)	**76.43** (+0.59)	**76.96** (+0.88)	**54.42** (+1.36)	**76.60** (+0.25)	**66.13** (+4.64)	**71.56** (+1.36)

Table 1: Results of different methods on the in-domain test sets of six commonsense reasoning datasets. We also report the increase of mixture setting compared to single setting. Results are averaged over three random runs. ($p < 0.01$ under t-test)

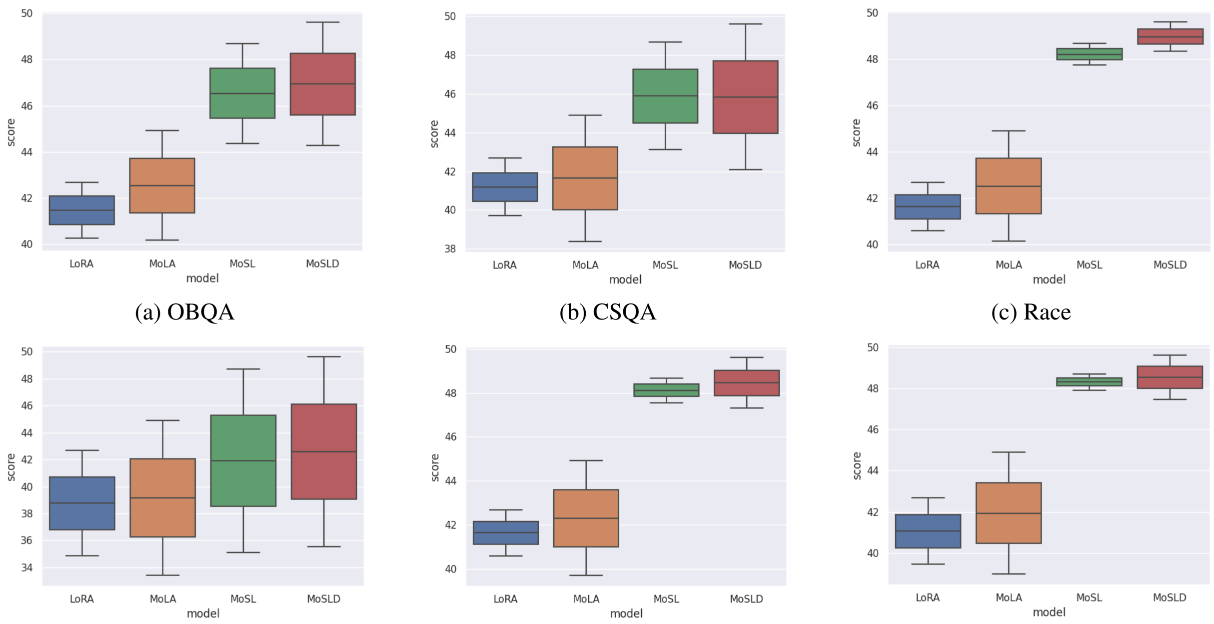
baseline in the single setting, we proceed to report results for the mixture setting. Here, we observe a decline in LoRA's performance, trailing 1.23 points behind FP-tuning (70.38%). Conversely, applying the MoE framework to LoRA, i.e., MoLA, achieves a score of 70.00%, demonstrating MoE's suitability for multi-task scenarios. Further comparison between single and mixture settings reveals that FP-tuning and MoLA improve by 3.14% and 2.41%, respectively, in the mixture setting compared to the single setting. However, LoRA's performance decreases by 0.42% in the mixture setting compared to the single setting, indicating conflicts between multi-task data and the mixture strategy's detrimental impact on performance.

Upon closer examination, our proposed MoSLD demonstrates performance enhancements of 2.61% and 1.56% over MoLA in single and mixture settings, respectively. This emphasizes the effectiveness of the sharing mechanism and dropout strategy in alleviating data conflicts and retaining shared knowledge between various tasks. Furthermore, conducting ablation experiments by removing the dropout strategy, MoSL experiences performance decreases of 1.04% and 0.68%, respectively, compared to MoSLD. This highlights the crucial role of the dropout strategy in mitigating training overfitting and optimization imbalance. Nevertheless, MoSL still achieves competitive results of 69.16% and 70.88%. We also found that our model not only achieves good results in the mixture setting, but also achieves good results in the single setting, which overcomes the disadvantage of MoLA's poor performance in the single setting. In conclusion, our approach exhibits significant advantages under both single and mixture settings, particularly in alleviating data conflicts across multiple tasks and addressing knowledge forgetting issues in multi-task learning. In addition, we also pay attention to the efficiency of training. Due to the introduction of multiple LoRAs, the trainable parameters of MoLA are higher than those of plain LoRA. However, although our MoSLD expands LoRA several times through the MoE architecture, it does not introduce a large number of additional parameters and also enables the LoRA training to have multiple capabilities. Details can be seen in Section 5.5.

5 Qualitative Analysis

5.1 Out-of-domain Test

To assess the generalization capability of our proposed model, we conducted out-of-domain experiments using the test set of MMLU. Figure 4 presents a boxplot, where the top and bottom horizontal lines represent the mixture and single settings, respectively. Our models, MoSL and MoSLD, consistently outperform others in both settings, exhibiting significant improvements, particularly on Race, Arc-e, and Arc-c datasets. This highlights the effectiveness of our models in disentangling domain knowledge and transferring general features across diverse datasets. OBQA and CSQA exhibit similar trends in the boxplot, indicating similar data distributions between the two datasets. Conversely, for MCTest, while improvements are observed in the mixture settings, the single settings remain relatively unchanged. This divergence may stem from the substantial differences between the MCTest and MMLU test sets, suggesting that introducing data from other domains or tasks could inspire general domain knowledge. In summary, our model demonstrates strong generalization capabilities, particularly in multi-task scenarios.



(a) OBQA (b) CSQA (c) Race

(d) MCTest (e) Arc-e (f) Arc-c

Figure 4: A comparision of performance for LoRA, MoLA, MoSL, and MoSLD on single and mixture settings for MMLU test set.

Model		**OBQA**	**CSQA**	**Race**	**MCTest**	**Arc-e**	**Arc-c**	**Avg**
LoRA	single	75.40	76.33	76.06	53.10	73.82	62.71	69.57
	mixture	72.80	76.30	78.23	55.67	70.87	61.00	69.15
MoLA	single	74.60	77.23	75.29	44.90	72.73	60.80	67.59
	mixture	76.60	73.46	75.25	54.42	76.34	63.91	70.00
MoSLD (matrix A)	single	78.40	75.84	76.08	53.06	76.35	61.49	**70.20**
	mixture	78.80	76.43	76.96	54.42	76.60	66.13	**71.56**
MoSLD (matrix B)	single	77.60	75.76	74.58	46.94	76.09	60.83	68.63
	mixture	76.40	74.11	75.25	56.46	77.15	65.02	70.73

Table 2: The results for applying our methods on matrix A and matrix B.

5.2 Effect of Model Parameters

In this section, we conduct parameter search experiments.

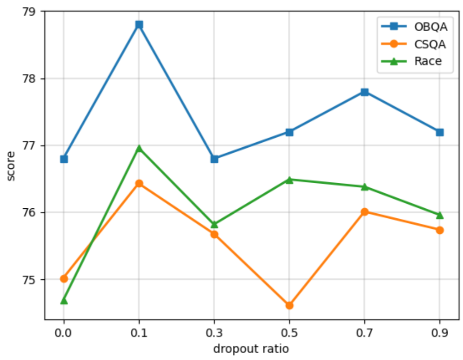
Dropout Location As shown in Table 2, we show the results of applying our methods on matrix A and matrix B. We found that in the single setting, MoSLD (matrix B) does not achieve much improvement, 0.94 points lower than the ordinary LoRA and 1.04 points higher than MoLA. The mixture setting still achieves good results. However, the results of applying our method on matrix B are lower than those of applying it on matrix A in both the single and mixture settings. This also shows that matrix A is more used to extract general features.

Dropout Ratio In Figure 5, we depict the performance of six datasets under the mixture setting with varying dropout ratios. We observe a general downward trend in most results as the dropout ratio increases. This phenomenon occurs because while dropout can mitigate overfitting to some extent, excessively high dropout rates may diminish the model's capabilities. Therefore, careful selection of the dropout ratio parameter is necessary. Interestingly, the curves for the Arc-e and Arc-c datasets remain relatively stable across different dropout ratios. We attribute this stability to the simplicity of these two datasets, where model sparsification has minimal impact on the results.

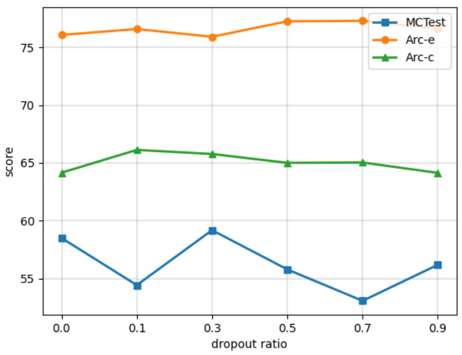
Expert Number Considering the redundancy among experts, following (Gao et al., 2024), we set different numbers of experts at different layers in Figure 6. Keeping the total number of experts constant, we choose three settings, i.e., (2,4,6,8), (5,5,5,5), (8,6,4,2). It is observed that assigning more experts at higher layers and fewer experts at

Model		OBQA	CSQA	Race	MCTest	Arc-e	Arc-c	Avg
LLaMA2-7B	single	78.40	75.84	76.08	53.06	76.35	61.49	70.20
	mixture	78.80	76.43	76.96	54.42	76.60	66.13	71.56
LLaMA2-13B	single	81.4	77.95	78.01	57.86	78.93	65.05	73.20
	mixture	82.2	78.46	79.87	58.50	79.67	70.14	74.81
LLaMA-33B	single	83.93	81.49	83.27	65.99	85.10	68.52	78.05
	mixture	84.55	83.26	84.90	66.73	85.95	74.36	79.96

Table 3: The results of six datasets in single and mixture settings based on LLaMA2-7B, LLaMA2-13B and LLaMA-33B.



(a) OBQA&CSQA&Race



(b) MCTest&Arc-e&Arc-c

Figure 5: Results of six datasets under different dropout ratios. Here, we are based on the mixture setting.

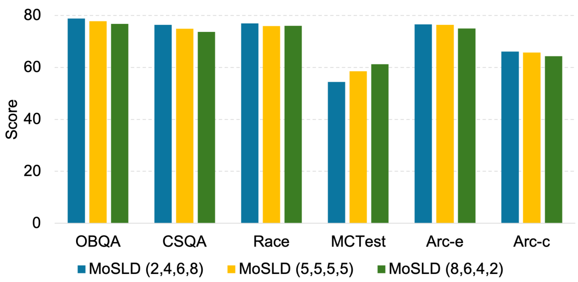


Figure 6: Different allocation strategies for the number of experts at different layers of the model. Here, we use the mixture setting.

lower layers, i.e., (2,4,6,8), works better. This is consistent with people's intuition: the lower layers of the model mainly extract general knowledge, which can be well learned by a small number of experts. While the higher layers of the model focus more on acquiring specific features of different tasks, and a larger number of experts can better capture multi-aspect capabilities.

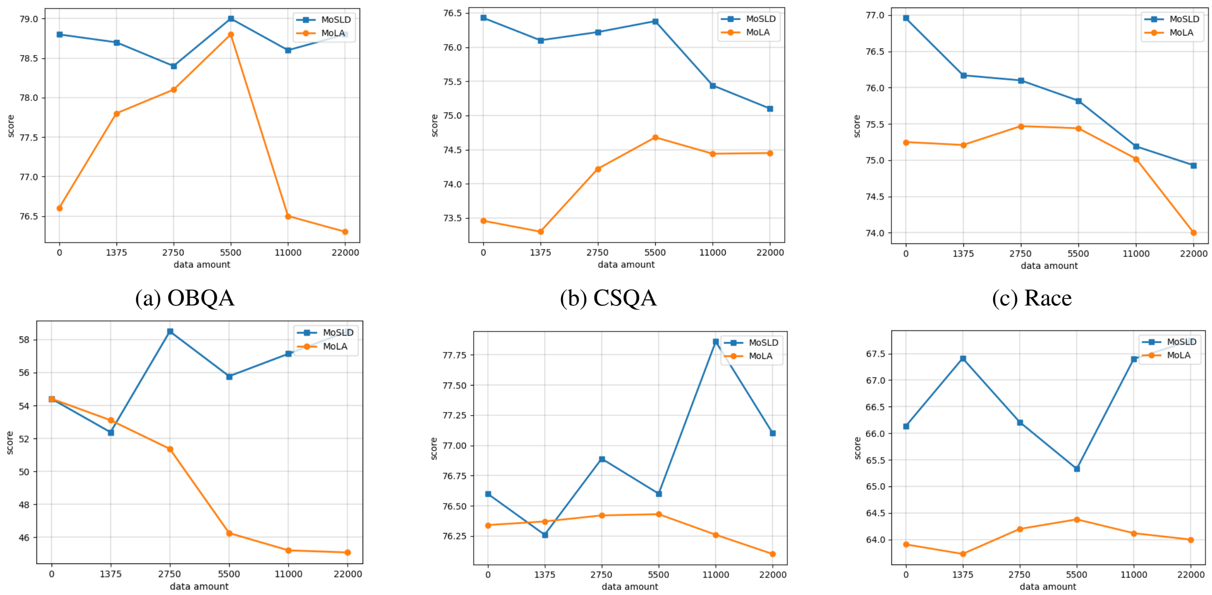
5.3 Mix with General Data

In Figure 7, we illustrate the impact of adding varying amounts of randomly filtered data from OpenOrca[1] to the mixed dataset D_{mix}. The data amount from OpenOrca ranges from 1,375 to 22,000. We observed that for MoLA, as the amount of general data increases, performance initially improves before eventually declining. This suggests that mixing a large amount of general data can lead to data conflicts and domain knowledge forgetting. In contrast, MoSLD demonstrates an upward trend in performance with the increase in data amount for OBQA, MCTest, Arc-e, and Arc-c. However, performance on CSQA and Race experiences a decline. We attribute this to significant distribution differences between these datasets and the general data. Overall, our model consistently outperforms MoLA when mixing various amounts of generic data. This underscores our model's ability to effectively leverage general knowledge across different tasks.

5.4 Scaling of Model Size

Table 3 shows the results of our model for the six datasets both in single and mixture settings as the model size scalings. We find that the performance of our model increases with the size of the model, whether in single or mixture settings, which is in line with our expectations. In addition, it is observed that the results improve by 1.36%, 1.61%, and 1.91% from single to mixture for LLaMA2-7B, LLaMA-13B, and LLaMA-33B, respectively.

[1]https://huggingface.co/datasets/Open-Orca/OpenOrca



(a) OBQA (b) CSQA (c) Race

(d) MCTest (e) Arc-e (f) Arc-c

Figure 7: Different data amount of OpenOrca between MoSLD and MoLA on six datasets. Here, we use the mixture setting.

Model	LoRA number	Forward param	Trainable param	Avg_score
FP-tuning	/	6.738B	6.738B	70.38
LoRA	(1A+1B)*32	6.743B	0.419B	69.57
MoLA	(5A+5B)*32	6.761B	2.228B	70.00
MoSLD	(1A+5B)*32	6.572B	1.389B	71.56

Table 4: The number of LoRA matrices, forward parameters, and trainable parameters for FP-tuning, LoRA, MoLA, and our MoSLD during training. Here, "A" is matrix A, "B" is matrix B, and "5" is the average number of experts per layer. We also report the average results across 6 datasets under the mixture setting.

The experimental results show that our method has achieved good performance on models of different sizes, and has a certain scaling ability.

5.5 Analysis of Computation Efficiency

In Table 4, we further show the computational efficiency of our model. We first analyze the number of new LoRA modules inserted in ordinary LoRA, MoLA, and MoSLD. Since MoLA introduces the MoE framework, the trainable parameters become 5 times that of ordinary LoRA, and its results are improved by 0.43 points from 69.57 to 70.00. We believe that despite the introduction of a large number of trainable parameters, the change in results is not very large, which is a method of sacrificing efficiency for effect. In addition, we also found that although our method reduces 128 matrix A compared to MoLA, it is still 1.56% higher than MoLA and 1.99% higher than LoRA. This shows that although our MoSLD introduces multiple LoRAs through the MoE framework, the expert sharing mechanism greatly reduces the additional parameters and achieves a balance between effect and efficiency. We also compare FP-tuning. Athough our trainable parameters are 20.6% of FP-tuning, but it still achieves a 1.18 point improvement. This also proves that our MoSLD is indeed an extremely efficient-parameter fine-tuning method.

6 Conclusion

In this paper, we propose MoSLD, which is a mixture-of-shared-LoRAs model with dropout strategy. Unlike traditional LoRA-MoE approaches, we design a sharing mechanism for matrix A, which aims to capture the general-feature among various tasks. A dropout strategy is aslo applied to the matrix A, solving the overfitting caused by parameter redundancy to a certain extent. Evaluations show that MoSLD outperforms the baseline in both single-task and multi-task scenarios. Especially in multi-task scenarios, where it can effectively alleviate knowledge conflict and forgetting problems. In general, our model is extremely parameter-efficient for fine-tuning.

Limitations

Although MoSLD achieves significant improvements over existing baselines, there are still avenues worth exploring in future research. (1) This paper focuses on applying MoSLD on the matrix Q and V of the attention layer. We hope to extend this method to the FFN layer. (2) This paper explores the multi-task setting of directly mixing multiple datasets and compares with the performance of a single task. We plan to study the impact of multi-task data ratio on MoSLD. (3) This paper emphasizes the extraction of general and unique features by the upper and lower projection matrices in LoRA, and intends to visualize this phenomenon in the future.

Ethics Statement

LoRA has emerged as a pivotal technique for refining extensive pre-trained models. Nevertheless, its efficacy tends to fail in multi-task learning. Conversely, the MoE architecture offers a promising remedy to this setback. However, it introduces hurdles such as the interference of data across diverse domains and the risk of forgetting knowledge from various tasks. Furthermore, MoE substantially inflates parameter counts, presenting computational challenges. In light of these considerations, we present MoSLD in this paper, a model that integrates the strengths of both approaches. MoSLD, a mixture-of-shared-LoRAs model with a dropout strategy, addresses these obstacles ingeniously. By sharing the upper projection matrix in LoRA among different experts, MoSLD fosters the acquisition of broad knowledge across tasks while allowing the lower projection matrix to concentrate on task-specific features. Additionally, the application of dropout mitigates parameter overfitting in LoRA. The experimental results prove the effectiveness of our model andevaluation framework. Besides, there is no hugebiased content in the datasets and the models. Ifthe knowledge base is further used, the biased con-tent will be brought into the generated responses,just like biased content posted by content creatorson the Web which is promoted by a search engine.To prevent the technology from being abused fordisinformation, we look forward to more research effort being paid to fake/biased/offensive contentdetection and encourage developers to carefullychoose the proper dataset and content to build theknowledge base.

References

Shaoxiang Chen, Zequn Jie, and Lin Ma. 2024. Llava-mole: Sparse mixture of lora experts for mitigating data conflicts in instruction finetuning mllms.

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. In *Advances in Neural Information Processing Systems*.

Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. 2018. Think you have solved question answering? try arc, the ai2 reasoning challenge. *arXiv:1803.05457v1*.

Shihan Dou, Enyu Zhou, Yan Liu, Songyang Gao, Jun Zhao, Wei Shen, Yuhao Zhou, Zhiheng Xi, Xiao Wang, Xiaoran Fan, Shiliang Pu, Jiang Zhu, Rui Zheng, Tao Gui, Qi Zhang, and Xuanjing Huang. 2024. Loramoe: Alleviate world knowledge forgetting in large language models via moe-style plugin.

William Fedus, Jeff Dean, and Barret Zoph. 2022a. A review of sparse expert models in deep learning.

William Fedus, Barret Zoph, and Noam Shazeer. 2022b. Switch transformers: scaling to trillion parameter models with simple and efficient sparsity. *J. Mach. Learn. Res.*, 23(1).

Wenfeng Feng, Chuzhan Hao, Yuewei Zhang, Yu Han, and Hao Wang. 2024. Mixture-of-loras: An efficient multitask tuning for large language models.

Chongyang Gao, Kezhen Chen, Jinmeng Rao, Baochen Sun, Ruibo Liu, Daiyi Peng, Yawen Zhang, Xiaoyuan Guo, Jie Yang, and VS Subrahmanian. 2024. Higher layers need more lora experts.

Edward J Hu, yelong shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*.

Chengsong Huang, Qian Liu, Bill Yuchen Lin, Chao Du, Tianyu Pang, and Min Lin. 2024. Lorahub: Efficient cross-task generalization via dynamic loRA composition.

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.

Young Jin Kim, Ammar Ahmad Awan, Alexandre Muzio, Andres Felipe Cruz Salinas, Liyang Lu, Amr Hendy, Samyam Rajbhandari, Yuxiong He, and Hany Hassan Awadalla. 2021. Scalable and efficient moe training for multitask multilingual models.

Guokun Lai, Qizhe Xie, Hanxiao Liu, Yiming Yang, and Eduard Hovy. 2017. RACE: Large-scale ReAding comprehension dataset from examinations. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 785–794, Copenhagen, Denmark. 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 (Volume 1: Long Papers)*, pages 4582–4597, Online. Association for Computational Linguistics.

Qidong Liu, Xian Wu, Xiangyu Zhao, Yuanshao Zhu, Derong Xu, Feng Tian, and Yefeng Zheng. 2023. Moelora: An moe-based parameter efficient fine-tuning method for multi-task medical applications.

Tongxu Luo, Jiahe Lei, Fangyu Lei, Weihao Liu, Shizhu He, Jun Zhao, and Kang Liu. 2024. Moelora: Contrastive learning guided mixture of experts on parameter-efficient fine-tuning for large language models.

Todor Mihaylov, Peter Clark, Tushar Khot, and Ashish Sabharwal. 2018. Can a suit of armor conduct electricity? a new dataset for open book question answering. In *EMNLP*.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul F Christiano, Jan Leike, and Ryan Lowe. 2022. Training language models to follow instructions with human feedback. In *Advances in Neural Information Processing Systems*, volume 35, pages 27730–27744. Curran Associates, Inc.

Matthew Richardson, Christopher J.C. Burges, and Erin Renshaw. 2013. MCTest: A challenge dataset for the open-domain machine comprehension of text. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 193–203, Seattle, Washington, USA. Association for Computational Linguistics.

Carlos Riquelme, Joan Puigcerver, Basil Mustafa, Maxim Neumann, Rodolphe Jenatton, André Susano Pinto, Daniel Keysers, and Neil Houlsby. 2021. Scaling vision with sparse mixture of experts. In *Advances in Neural Information Processing Systems*, volume 34, pages 8583–8595. Curran Associates, Inc.

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

Alon Talmor, Jonathan Herzig, Nicholas Lourie, and Jonathan Berant. 2019. CommonsenseQA: A question answering challenge targeting commonsense knowledge. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4149–4158, Minneapolis, Minnesota. Association for Computational Linguistics.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. 2023a. Llama: Open and efficient foundation language models.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. 2023b. Llama 2: Open foundation and fine-tuned chat models.

Sheng Wang, Liheng Chen, Jiyue Jiang, Boyang Xue, Lingpeng Kong, and Chuan Wu. 2024. Lora meets dropout under a unified framework.

Xun Wu, Shaohan Huang, and Furu Wei. 2024. Mixture of loRA experts. In *The Twelfth International Conference on Learning Representations*.

Jinghan Zhang, Shiqi Chen, Junteng Liu, and Junxian He. 2023. Composing parameter-efficient modules with arithmetic operation. In *Thirty-seventh Conference on Neural Information Processing Systems*.

Yanqi Zhou, Tao Lei, Hanxiao Liu, Nan Du, Yanping Huang, Vincent Zhao, Andrew M Dai, zhifeng Chen, Quoc V Le, and James Laudon. 2022. Mixture-of-experts with expert choice routing. In *Advances in Neural Information Processing Systems*, volume 35, pages 7103–7114. Curran Associates, Inc.