LoRAMoE: Alleviating World Knowledge Forgetting in Large Language Models via MoE-Style Plugin

Shihan Dou[1*], **Enyu Zhou**[1*], **Yan Liu**[1], **Songyang Gao**[1], **Wei Shen**[1], **Limao Xiong**[1], **Yuhao Zhou**[1], **Xiao Wang**[1], **Zhiheng Xi**[1], **Xiaoran Fan**[1], **Shiliang Pu**[5], **Jiang Zhu**[5], **Rui Zheng**[1], **Tao Gui**[2†], **Qi Zhang**[1,3†], **Xuanjing Huang**[1,4†]

[1] School of Computer Science, Fudan University

[2] Institute of Modern Languages and Linguistics, Fudan University

[3] Shanghai Collaborative Innovation Center of Intelligent Visual Computing

[4] International Human Phenome Institutes, Shanghai, China

[5] Hikvision Inc

{shdou21, eyzhou23}@m.fudan.edu.cn

Abstract

Supervised fine-tuning (SFT) is a crucial step for large language models (LLMs), enabling them to align with human instructions and enhance their capabilities in downstream tasks. Substantially increasing instruction data is a direct solution to align the model with a broader range of downstream tasks or notably improve its performance on a specific task. However, we find that large-scale increases in instruction data can damage the world knowledge previously stored in LLMs. To address this challenge, we propose LoRAMoE, a novel framework that introduces several low-rank adapters (LoRA) and integrates them by using a router network, like a plugin version of Mixture of Experts (MoE). It freezes the backbone model and forces a portion of LoRAs to focus on leveraging world knowledge to solve downstream tasks, to alleviate world knowledge forgetting. Experimental results show that, as the instruction data increases, LoRAMoE can significantly improve the ability to process downstream tasks, while maintaining the world knowledge stored in the LLM. Our code is available at https://github.com/Ablustrund/LoRAMoE.

1 Introduction

Supervised fine-tuning (SFT) provides a pivotal technique to make large language models (LLMs) follow human instructions and improve their performance of downstream tasks (Chung et al., 2022; Ouyang et al., 2022). Although some studies (Zhou et al., 2023; Cao et al., 2023) indicate that LLMs trained on a little data can follow instructions well, increasing the amount of data is a straightforward way to enhance their ability to multiple downstream tasks or improve their performance on a specific task, as shown in the left of Figure 1.

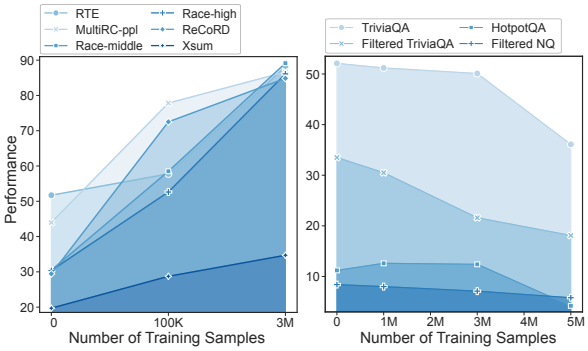

Figure 1: **(Left)** With the number of fine-tuning data increases from 10K to 3M, the performance of many downstream tasks is significantly improved. **(Right)** With the amount of instruction data increasing, fine-tuning the language models results in a decline in performance on the benchmarks that measure their world knowledge, such as TriviaQA (Han et al., 2019), Natural Questions (Kwiatkowski et al., 2019). The details of training implementation can be seen in Section 2.1.

However, the large-scale increase in instruction data can destroy the world knowledge stored in LLMs, as illustrated in the right of Figure 1. Specifically, as the amount of instruction data increases, we observe a notable decline in performance on Closed-Book Question Answering (CBQA) datasets, which are used to measure world knowledge in LLMs (Touvron et al., 2023; Neeman et al., 2022). In the paradigm of supervised fine-tuning, the conflict between maintaining world knowledge inside LLMs and improving their performance on downstream tasks by scaling up instruction data has not been thoroughly examined.

In this paper, we propose LoRAMoE, a novel framework for SFT, to enhance the models' capability of solving downstream tasks, while alleviat-

[*] Equal contribution.

[†] Corresponding author.

Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 1932–1945

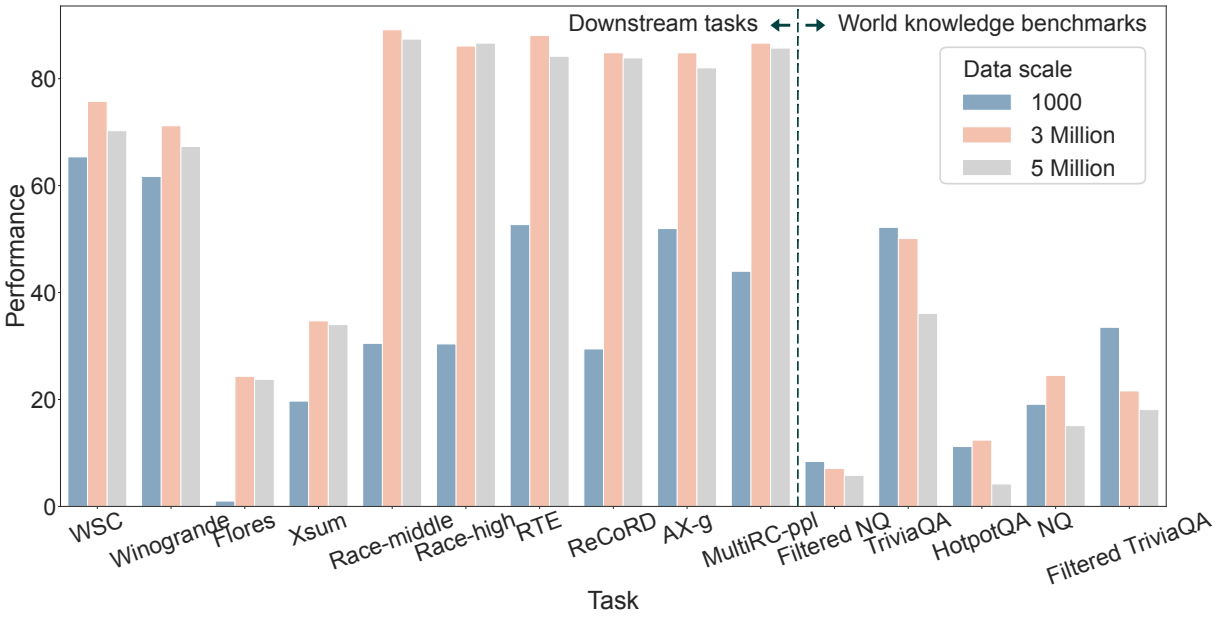

Figure 2: Performance on the various tasks after expanding the amount of fine-tuning data. For most of the downstream tasks (e.g., NLI and summarization), with the expansion of training data, performance on these tasks remains stable after the improvement. Whereas, for the world knowledge benchmark, a significant **decline** can be witnessed after a large amount of instruction data.

ing world knowledge forgetting during the training phase. LoRAMoE is a Mixture-of-Experts-style (MoE-style) plugin, which introduces several low-rank adapters (LoRA) (Hu et al., 2021) as experts and integrates them by using a router network. The router network automatically assigns weights to experts, which can improve the LLM's performance on multiple downstream tasks.

To demonstrate the efficacy of our proposed method, we conduct extensive experiments across a range of downstream tasks. Experiment results show that LoRAMoE can significantly improve LLM's ability to address the various downstream tasks by fine-tuning the model on a large amount of instruction data, while maintaining the world knowledge stored in the model. In addition, we further evaluate our method by visualizing the expert weight for tasks. The result indicates that LoRAMoE adequately alleviates world knowledge forgetting and achieves an improvement of models by fostering collaboration among experts. The main contributions of our paper are as follows:

1. We find that significantly increasing the amount of instruct data during the SFT phase can damage the world knowledge inside the LLMs. The need for improvement in downstream tasks by scaling up instruction data conflicts with maintaining the world knowledge inside the model.
2. We introduce LoRAMoE, a novel framework for SFT, which introduces LoRAs as experts and integrates them by the router. LoRAMoE can enhance the model's ability to address downstream tasks, while alleviating the world knowledge forgetting.
3. Extensive Experiments demonstrate the efficacy of our proposed approach in multi-tasks and mitigating the forgetting of world knowledge inside the model. The visualization experiment shows that LoRAMoE can achieve an improvement by fostering collaboration among experts.

2 Motivation

In this section, we verify that a large-scale SFT can cause irreversible damage to world knowledge within the LLMs while improving the LLMs' performance in various downstream tasks.

2.1 A Diverging Trend

We constructed a dataset containing seven categories of tasks with a total of five million training samples, and used it to conduct SFT on a Llama-2-7B model. The implementation details are described in Appendix A. During the expansion of fine-tuning data, we observed a diverging trend in the performance across two types of tasks, as shown in Figure 2:

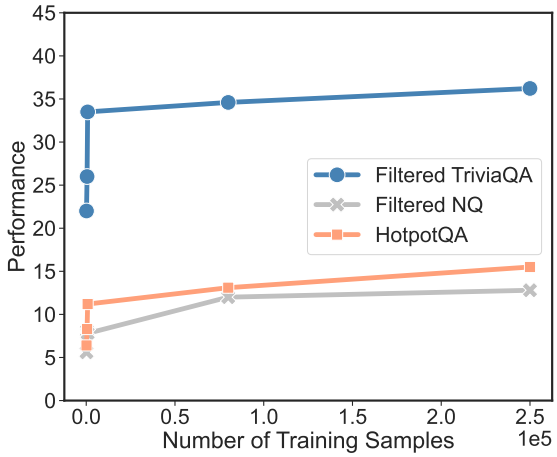

Figure 3: Performance on world knowledge benchmarks after training on CBQA solely. Its performance rises greatly after training with very few samples and remains relatively stable thereafter.

Across downstream tasks such as summarization, Natural Language Inference (NLI), machine translation, and others, the performance of the fine-tuned model initially showed a magnificent increase and eventually stabilized at a promising level. However, when it comes to closed-book QA (CBQA) tasks that are used as world knowledge benchmark (Touvron et al., 2023; Neeman et al., 2022), the model's performance catastrophically declines under the baseline Notably, with the training data expanding, a contiguous decline can be witnessed. Moreover, this decline will occur earlier if the test set is filtered.[1] Appendix B case with a larger dataset including more tasks shows an even steeper drop on world knowledge benchmarks, although performance remains competitive on others.

2.2 Irreversible Knowledge Forgetting

In this section, we dissect the reason behind the decline on these world knowledge benchmarks during the expansion of fine-tuning data. We find this results from the occurrence of irreversible knowledge forgetting inside the LLM.

The performance on world knowledge benchmarks highly relies on the knowledge and skills learned during pre-training phase. To investigate the relationship between the performance on world knowledge benchmarks and the knowledge embedded in pre-trained models (Petroni et al., 2019; Roberts et al., 2020; AlKhamissi et al., 2022), we conduct fine-tuning solely on the CBQA dataset with 250k samples and run evaluation on the test sets without training-testing overlap (e.g. Filtered NQ and Filtered TriviaQA). Results in Figure 3 show initial training boosts performance significantly, especially the first 1% (approximately 1k samples), with limited gains thereafter. This is because early fine-tuning aligns existing knowledge with new instructions, improving CBQA results. However, due to minimal training-testing data overlap, adding more samples doesn't further enhance performance. Thus, a model's benchmark success relies on world knowledge acquired from the pre-training.

Task Name	Baseline	SFT solely on CBQA	Two-stage Fine-tuning
TriviaQA	33.5	36.22	13.7
NQ	7.8	12.8	3.6
HotpotQA	11.2	16.1	7.1

Table 1: Performance from left to right: LlaMA-2-7B, model tuned on CBQA, and model tuned on 3M instructions then on CBQA. Despite further tuning on CBQA, the large-scale SFT model's knowledge-answering doesn't improve, staying below the baseline.

Given this, it is naturally assumed that **the diminished performance on knowledge benchmark stems from the damage of knowledge stored in the LLM due to large-scale instruction tuning**. To verify the hypothesis, we sequentially fine-tuned a model using two datasets, first excluding CBQA data, then with CBQA data. Results presented in Table 1 show a great decline in knowledge capabilities versus the original LLM. This indicates that the world knowledge within the model was compromised during the first stage of large-scale fine-tuning, resulting in the model's inability to forge the alignment between human instructions and the already destroyed knowledge in the subsequent stage of fine-tuning solely with CBQA.

To sum up, the pursuit of enhancing performance on downstream tasks through the expansion of training data conflicts with the preservation of world knowledge within the model in vanilla SFT.

3 LoRAMoE

In this section, we elaborate on the methodological details of LoRAMoE, which is an MoE-style plugin and introduced Localized Balancing Constraint during the training phase to alleviate the

[1]Considering previous work that has noted train-test overlap in CBQA datasets (Lewis et al., 2020), we elaborately select parts of the CBQA dataset without train-test overlap for our testing set, namely *Filtered NQ* and *Filtered TriviaQA*.

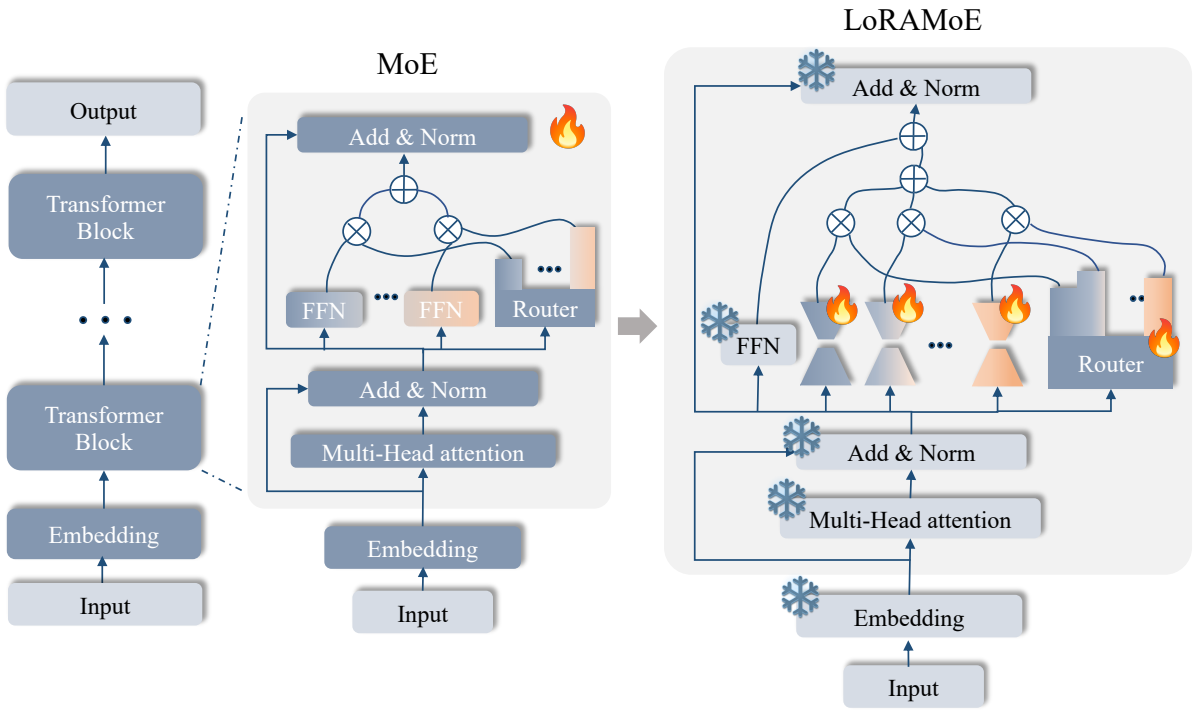

Figure 4: The architecture of LoRAMoE, compared with classic MoE. LoRAMoE utilizes multiple LoRAs as adaptable experts and a router to gate them in the FFN layer of every transformer block. During the training process, only the experts and the router are optimized.

world knowledge, as shown in Figure 4.

3.1 Architecture

The left of Figure 4 illustrates the forward process of the standard MoE architecture (Shazeer et al., 2016; Fedus et al., 2021; Lepikhin et al., 2020). In the MoE, the router assigns weights of experts according to the data, allowing them to divide their labor to complete the forward process (Jacobs et al., 1991). The key sight of LoRAMoE is that we freeze the backbone model to maintain world knowledge and introduce experts to leverage this knowledge to address tasks, while improving the performance on multiple downstream tasks. Additionally, we utilize the LoRA (Hu et al., 2021) as the architecture of the expert to improve training and inference efficiency.

Formally, for the traditional transformers architecture, the forward propagation process of the feed-forward neural (FFN) network block can be simplified as follows:

$$f(x) = x + f_{\text{FNN}}(x). \tag{1}$$

The matrix operation of the linear layer in this forward propagation can be expressed as:

$$o = Wx = W_0x + \Delta Wx \tag{2}$$

where $W_0 \in \mathbb{R}^{d_{\text{in}} \times d_{\text{out}}}$ represents the parameter matrix of the backbone model and $\Delta W \in \mathbb{R}^{d_{\text{in}} \times d_{\text{out}}}$ denotes the updated parameter during the training phase. For LoRAMoE, we replace the linear layer in the FFN block with the MoE-style plugin, which makes experts collaborate to address tasks. During the training phase, we freeze the backbone to maintain the world knowledge and only update ΔW. Consider the LoRAMoE layer containing N experts, which is denoted as $\{E_i\}_{i=1}^{N}$, the forward process of the layer can be mathematically expressed as follows:

$$o = W_0x + \Delta Wx = W_0x + \sum_{i=1}^{N} G(x)_i E_i(x) \tag{3}$$

where $E_i(\cdot)$ and $G(\cdot) = \text{Softmax}(xW_g)$ represent the i-th expert and the router in the LoRAMoE layer, respectively. The W_g is the trainable parameter matrix of the route network. By this, the experts and the outer work in tandem, enabling the experts to develop varied capabilities and efficiently handle diverse types of tasks.

In addition, LoRA has been proven to be both effective and efficient for the SFT phase of LLMs (Wang et al., 2023a; Liu et al., 2022; Pan et al., 2022). To enhance the efficiency and resource conservation of the fine-tuning process, we replace the

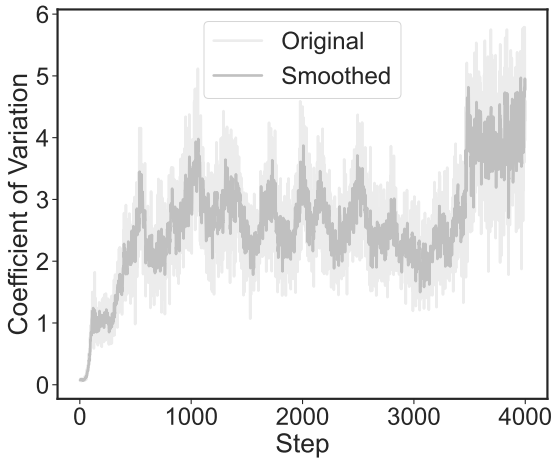

Figure 5: The coefficient of variation for the experts of the unconstrained LoRAMoE progressively escalates and sustains at a high value, i.e., approximately three, similar to the phenomenon observed at Shazeer et al. (2016). This indicates that the router assigns large weights to the same few experts.

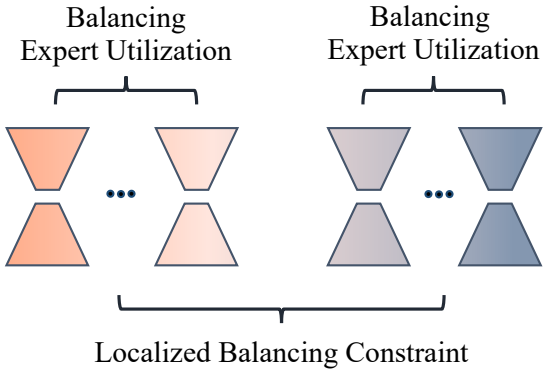

Figure 6: Localized balancing constraint. We softly force experts to focus on two types, one for leveraging world knowledge by learning on its related tasks, and another for concentrating on other downstream tasks. Meanwhile, the experts in solving the same aspect are balancing.

parameter matrix of the experts with a low-rank format. Specifically, the matrix $\Delta W_E \in \mathbb{R}^{d_{\text{in}} \times d_{\text{out}}}$ of the expert $E(\cdot)$ in the LoRAMoE layer can be written as follows:

$$\Delta W_E = BA \tag{4}$$

where $A \in \mathbb{R}^{d_{\text{in}} \times r}$, $B \in \mathbb{R}^{r \times d_{\text{out}}}$, and the rank $r \ll \min(d_{\text{in}}, d_{\text{out}})$. LoRA contributes to a significant reduction in the trainable parameters, thereby enhancing efficiency and saving costs during the fine-tuning process.

Overall, the forward process of the LoRAMoE layer replaced the traditional FFN layer can be represented as:

$$o = W_0 x + \frac{\alpha}{r} \sum_{i=1}^{N} \omega_i \cdot B_i A_i x \tag{5}$$

where ω_i denotes the weight of i-th expert and α is the constant hyper-parameter, approximately equivalent to the learning rate.

3.2 Localized Balancing Constraint

The imbalance of the experts' utilization is a typical problem in MoE (Shazeer et al., 2016; Fedus et al., 2021), which is also observed in our proposed method, as shown in Figure 5. The conventional solution is balancing expert utilization (Shazeer et al., 2016), which involves making the coefficient of variation of the experts' importance as the loss function. However, this method assumes all the training samples are under the same distribution, which ignores the fact that samples may be from different distributions such as the question-answering task and other downstream tasks, more detailed analysis and conceptual proof in Appendix C.

Considering the mixed characteristics of data distributions are important, during the training phase, we introduce localized balancing constraint, a novel balancing expert utilization method to make a portion of experts focus more on leveraging world knowledge to solve tasks. As shown in Figure 6, during the fine-tuning phase, we softly constrain experts to concentrate on two aspects, one of which focuses on leveraging world knowledge by learning on its related datasets, while another focuses on other downstream tasks. In addition, all experts within the same aspects are balanced such as balancing expert utilization.

Formally, we define the importance matrix $\mathbf{Q}$ of the LoRAMoE layer and $\mathbf{Q}_{n,m}$ denotes the sum of router values of the n-th expert for the m-th training sample in a batch, which can be represented as follows:

$$\mathbf{Q}_{n,m} = \sum_{j=1}^{T_m} G(x_j)_i = \frac{\exp(\omega_i^j/\tau)}{\sum_{k=1}^{N} \exp(\omega_i^j/\tau)} \tag{6}$$

where N and T_m denote the number of experts and the number of tokens of m-th training sample, respectively. x_j is the hidden input of the j-th token. We then define the coefficient matrix $\mathbf{I}$ with the same size of $\mathbf{Q}$, corresponding to the importance matrix $\mathbf{Q}$. $\mathbf{I}_{n,m}$ denotes the importance coefficient of $\mathbf{Q}_{n,m}$, which can be written as follows:

$$\mathbf{I}_{n,m} = \begin{cases} 1+\delta, & \text{Type}_e(n) = \text{Type}_s(m) \\ 1-\delta, & \text{Type}_e(n) \neq \text{Type}_s(m) \end{cases} \tag{7}$$

where $\delta \in [0, 1]$ controls the degree of imbalance between experts types. $\text{Type}_e(n)$ and $\text{Type}_s(m)$ are pre-defined target type of n-th expert and the task type of m-th training sample in a batch, respectively.

We categorize the instruction data into two distinct types: world knowledge-related tasks such as TriviaQA, and other downstream tasks such as Flores. Then, we enable a portion of experts to learn on world knowledge-related tasks to align human instructions with world knowledge, while making other experts focus more on enhancing the performance of downstream tasks. Formally, suppose that $\mathbf{I}_{i,k}$ and $\mathbf{I}_{j,k}$ denote the importance coefficient of the i-th and j-th expert for the k-th sample, respectively. If experts are in the same group, their values at corresponding positions in the coefficient matrix are identical, i.e., $\mathbf{I}_{i,k} = \mathbf{I}_{j,k}$. This indicates that these experts have the same importance because they are assigned to focus on learning the same type of tasks. On the contrary, the values of experts from distinct groups at their coefficient matrix are different, i.e., $\mathbf{I}_{i,k} \neq \mathbf{I}_{j,k}$.

The localized balancing constraint loss $\mathcal{L}_{lbc}$ is defined to measure the dispersion of the weighted importance matrix $\mathbf{Z} = \mathbf{I} \circ \mathbf{Q}$, which can be mathematically represented as:

$$\mathcal{L}_{lbc} = \frac{\sigma^2(\mathbf{Z})}{\mu(\mathbf{Z})} \tag{8}$$

where $\sigma^2(\mathbf{Z})$ and $\mu(\mathbf{Z})$ represent the variance and mean of $\mathbf{Z}$, respectively. Specifically, if a specific sample is from the world knowledge-related dataset, experts focusing on solving this type will have larger values in the coefficient matrix $\mathbf{I}$. Optimizing the loss $\mathcal{L}_{lbc}$ reducing can make corresponding experts learn more from this sample and be assigned a larger weight by the router. Meanwhile, experts solving the same type of task are balanced such as Shazeer et al. (2016). In addition, the constraint is soft to encourage cooperation among experts to preserve the capacity for generalization.

Overall, localized balancing constraint $\mathcal{L}_{lbc}$ achieves a localized balance between two types of experts: one specializes in leveraging world knowledge by training more on world knowledge-related datasets, while the other concentrates on various downstream tasks. The loss of LoRAMoE can be represented as follows:

$$\mathcal{L}_{\text{total}} = \mathcal{L} + \beta \mathcal{L}_{lbc} \tag{9}$$

where $\mathcal{L}$ is the next-token prediction loss of LLMs and β controls the strength of localized balancing constraint. In the training phase, we freeze the backbone model and the trainable parameters are only those of the experts and routers within the LoRAMoE layers. In the inference process, the router automatically assigns weights to all experts, which avoids the need for pre-specified data types.

4 Experiments

4.1 Experiment Setup

In this section, we introduce the training implementation for LoRAMoE. We only replace the linear layer in the feed-forward neural network of LLM with the LoRAMoE layer, initializing each layer with six experts, of which three experts are dedicated to addressing downstream tasks, and the other three are responsible for leveraging world knowledge in the base model by learning on its related tasks. The hyperparameters for control constraint strength β and degree of imbalance δ are both set to 0.1. For LoRA settings, the α, and r are set to 32 and four for the main result, respectively. The dropout is 0.05, and the learning rate is $2e-4$. The training dataset is the 3 million set the same as the one described in Appendix A, so as the evaluation settings. We freeze the parameters of the base model, rendering only the experts and router in LoRAMoE trainable. The batch size per node is set to 16.

4.2 Main Results

Table 2 displays the performance of LoRAMoE and compares this result with the outcomes of directly applying SFT to the model or utilizing LoRA tuning. The results show that the language model with LoRAMoE gets good performance on both world knowledge benchmarks and others, indicating its effectiveness in avoiding knowledge forgetting while improving multi-tasking abilities.

For world knowledge benchmarks, contrary to the catastrophic collapse seen in Section 2, LoRAMoE not only avoids this issue but also surpasses the model fine-tuned solely with the CBQA dataset. LoRAMoE shows a significant performance boost on world knowledge benchmarks over vanilla SFT, with up to a 63.9% improvement and an average increase of 35.3%.

For other downstream tasks, LoRAMoE is capable of achieving performance close to or even surpassing that of direct SFT. For instance, in all read-

Task Name	Baseline	SFT solely on CBQA	SFT	LoRA	LoRAMoE	LoRAMoE (with $\mathcal{L}_{lbc}$)
WSC	65.4	-	**76.0**	65.4	71.2	70.2
winogrande	61.7	-	**71.2**	64.3	66.3	69.6
Flores	0.1	-	24.3	**26.6**	26.4	25.9
Xsum	19.7	-	34.7	34.5	**34.8**	33.2
Race-middle	30.5	-	89.1	78.8	84.5	**90.0**
Race-high	30.4	-	86.1	75.3	80.6	**86.5**
RTE	52.7	-	**88.1**	77.3	80.9	87.4
ReCoRD	29.4	-	84.8	83.2	84.3	**85.9**
AX-g	52.0	-	84.8	76.1	81.7	**87.1**
multiRC	44.0	-	86.7	81.4	87.3	**87.9**
TriviaQA	52.2	57.8	51.1	47.8	55.3	**58.1**
NQ	18.5	28.6	24.5	16.2	23.8	**28.0**
Filtered TriviaQA	33.5	36.2	21.6	33.4	**38.5**	35.4
Filtered NQ	7.8	12.8	7.3	11.6	**13.4**	12.0
HotpotQA	11.2	16.1	13.4	10.7	14.4	**16.1**

Table 2: Results of LoRAMoE. Contrary to direct full fine-tuning and the use of LoRA-tuning that exhibits reduced performance on world knowledge benchmarks after training, our approach ensures simultaneous growth of both world knowledge benchmarks and other downstream tasks.

ing comprehension tasks (i.e., Race, ReCoRD, multiRC), LoRAMoE achieved superior performance.

We also compare our method against PEFT by single LoRA. The knowledge forgetting also occurred during the single LoRA-tuning, as it is essentially the same as vanilla SFT (Hu et al., 2021). Compared with a single LoRA, multiple collaborative LoRAs in LoRAMoE enhance both world knowledge retention and multitasking performance. They offer an average boost of 30.9% in world knowledge benchmarks and 8.4% in other downstream tasks.

Besides, $\mathcal{L}_{lbc}$ improves outcomes for LoRAMoE in the vast majority of tasks, both world knowledge benchmarks and others. Notably, for reading comprehension, NLI, and the original CBQA dataset, the benefits of this method were quite substantial, up to 17.6%. This indicates capability partitioning in the expert group benefits the performance in multi-task learning.

4.3 Sensitivity Analysis

In this section, we analyze the parameter sensitivity of LoRAMoE. Keeping other settings constant, we vary the number of experts and the rank of LoRA. The average performance with varied parameter settings on all test sets including the world knowledge benchmark and all other downstream tasks is shown in Table 3. In Appendix D there are detailed results.

As the number of trainable parameters increases, performance is generally stable. the number of 6 experts is the most beneficial choice, as more experts do not lead to higher performance. While the increase in LoRA rank improves the model's capabilities somewhat, it brings about an exponential rise in trainable parameters.

# Experts	# LoRA Rank	# Trainable Param.	Avg. Results
6	4	0.57%	58.21
4	4	0.38%	55.84
8	4	0.76%	56.58
6	8	1.07%	58.11
6	16	2.08%	58.86

Table 3: Performance of LoRAMoE varies with the number of experts and LoRA rank across all test sets. This includes the average results on both the world knowledge benchmark and all other downstream tasks. LoRAMoE shows stability to parameter changes.

4.4 Visualizing the Experts Utilization

To confirm the effectiveness of LoRAMoE in specializing the experts with two types, we visualize their weight assigned by the router when encountered with data from downstream tasks and knowledge benchmarks respectively, as illustrated in Figure 7.

There is a distinct contrast in the utilization of the two types of experts when dealing with world knowledge benchmarks and other down-

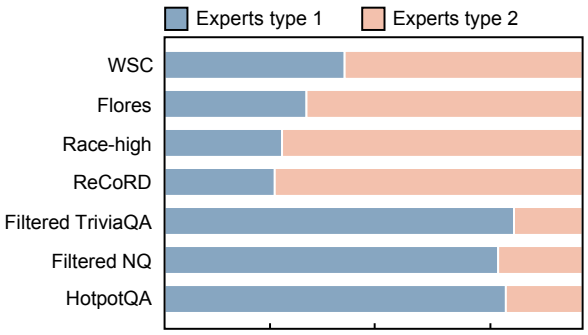

Figure 7: Visualization of routers' weight on different types of data, where type 1 refers to the experts dedicated to aligning the world knowledge in the base model with the human instruction and type 2 refers to the experts that focus on downstream tasks. The utilization rate of the type of experts diverged significantly across tasks.

stream tasks. This suggests that the routers can automatically allocate specific tasks to experts with corresponding abilities during the inference phase. Specifically, the experts requested to leverage world knowledge are greatly employed in world knowledge benchmarks (e.g., TriviaQA, Natural Questions, and HotpotQA), underscoring their vital role in preventing world knowledge forgetting. This corresponds to the fact we state in Section 2 that supervised fine-tuning boosts the model's capabilities in these tasks by associating pre-stored world knowledge in the model with human instructions. On the other hand, experts assigned to focus on enhancing performance in downstream tasks are given increased prominence when encountering these tasks. Through this visualized result, we find that some downstream tasks still require experts of another type. It is reasonable. For example, in reading comprehension tasks, the knowledge learned by the model during pre-training can better assist in making factual judgments. This phenomenon is even more pronounced in language-based tasks. In the WSC task (Levesque et al., 2012), the router allocates an average of about 45% of its attention to the experts responsible for world knowledge.

5 Related Work

Parameter-Efficient Fine-tuning. With the size of language models growing larger, parameter-efficient fine-tuning (PEFT) (He et al., 2021) has become crucial for resource savings. Researchers have proposed several approaches such as LoRA (Hu et al., 2021), adapters (Houlsby et al., 2019), and prompt learning (Lester et al., 2021), to enhance fine-tuning efficiency. PEFT based on low-rank adapters (Hu et al., 2021) is popular and widely used, which introduces two trainable low-rank matrices in each fully connected layer, to achieve significant savings in training resources without adding additional inference computation cost. We apply low-rank techniques to the structure of experts to save resource consumption.

Mixture-of-Experts. The mixture of Experts (MoE) replaces the feed-forward neural network layer with sparsely activated experts, which significantly enlarges the model without remarkably increasing the computational cost (Jacobs et al., 1991). Currently, the token-level MoE architectures are widely used in pre-trained language models and vision models (Shazeer et al., 2016; Lepikhin et al., 2020; Du et al., 2022; Riquelme et al., 2021). In addition, researchers (Zhou et al., 2022; Chi et al., 2022) aim to investigate the router selection problem in MoE. Unlike these efforts to expand the model size and address the selection problem, we propose an MoE-style framework for multi-task learning and maintaining the world knowledge stored in LLMs.

Multi-LoRA Architecture. Researchers also have utilized multiple LoRAs for enhanced model performance. Huang et al. (2023) propose LoraHub to choose different LoRA combinations for task generalization. MOELoRA (Liu et al., 2023) leverage LoRA and MoE for task-specific tuning and multitasking, especially in healthcare. However, these methods need the data type as the input during the inference phase, which limits the application of the model to other tasks. Chen et al. (2023a) first introduces multiple LoRA serving systems and Sheng et al. (2023) proposes S-LoRA, a system that can serve thousands of LoRA adapters from a single machine. Chen et al. (2023b) introduces several experts to enhance the model's ability for multimodal learning. Unlike these approaches, LoRAMoE introduces an MoE-style plugin and Localize Balancing Constraint to tackle world knowledge forgetting in LLMs, while enhancing the model's ability to multi-task learning.

6 Conclusion

In this paper, we first delve into the conflict between improving LLM's performance on downstream tasks by scaling up data during the SFT phase and discouraging world knowledge forgetting. To address this conflict, we then introduce Lo-

RAMoE, a novel framework for SFT, which introduces LoRAs as experts and integrates them by the router. Extensive experimental results demonstrate that LoRAMoE can foster collaboration among experts to enhance the model's performance of downstream tasks, while preserving the world knowledge inside it.

7 Limitations

In this section, we discuss the potential limitations of our proposed method LoRAMoE. Firstly, although we have demonstrated the effectiveness of LoRAMoE in alleviating world knowledge forgetting while enhancing the downstream ability of the LLMs with SFT, we limit the model size to 7B due to resource and time constraints. Further work will be conducted on the larger LLMs, to understand the influence of large-scale SFT on these LLMs and to boost their multitasking abilities. Secondly, the localized balancing constraint can softly constrain the type of experts and balance the experts utilization. However, we haven't studied the case where there are more experts types for a more fine-grained task category. Future work will be conducted on a more fine-grained understanding of the influence of SFT and the utilization of LoRAMoE.

8 Acknowledgements

The authors wish to thank the anonymous reviewers for their helpful comments. This work was partially funded by National Natural Science Foundation of China (No.62206057,62076069), Shanghai Rising-Star Program (23QA1400200), Natural Science Foundation of Shanghai (23ZR1403500), Program of Shanghai Academic Research Leader under grant 22XD1401100.

References

Badr AlKhamissi, Millicent Li, Asli Celikyilmaz, Mona Diab, and Marjan Ghazvininejad. 2022. A review on language models as knowledge bases. *arXiv preprint arXiv:2204.06031*.

Yihan Cao, Yanbin Kang, and Lichao Sun. 2023. Instruction mining: High-quality instruction data selection for large language models. *arXiv preprint arXiv:2307.06290*.

Lequn Chen, Zihao Ye, Yongji Wu, Danyang Zhuo, Luis Ceze, and Arvind Krishnamurthy. 2023a. Punica: Multi-tenant lora serving. *arXiv preprint arXiv:2310.18547*.

Zeren Chen, Ziqin Wang, Zhen Wang, Huayang Liu, Zhenfei Yin, Si Liu, Lu Sheng, Wanli Ouyang, Yu Qiao, and Jing Shao. 2023b. Octavius: Mitigating task interference in mllms via moe. *arXiv preprint arXiv:2311.02684*.

Zewen Chi, Li Dong, Shaohan Huang, Damai Dai, Shuming Ma, Barun Patra, Saksham Singhal, Payal Bajaj, Xia Song, Xian-Ling Mao, et al. 2022. On the representation collapse of sparse mixture of experts. *Advances in Neural Information Processing Systems*, 35:34600–34613.

Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Yunxuan Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, et al. 2022. Scaling instruction-finetuned language models. *arXiv preprint arXiv:2210.11416*.

Nan Du, Yanping Huang, Andrew M Dai, Simon Tong, Dmitry Lepikhin, Yuanzhong Xu, Maxim Krikun, Yanqi Zhou, Adams Wei Yu, Orhan Firat, et al. 2022. Glam: Efficient scaling of language models with mixture-of-experts. In *International Conference on Machine Learning*, pages 5547–5569. PMLR.

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

Francisco Guzmán, Peng-Jen Chen, Myle Ott, Juan Pino, Guillaume Lample, Philipp Koehn, Vishrav Chaudhary, and Marc'Aurelio Ranzato. 2019. The flores evaluation datasets for low-resource machine translation: Nepali-english and sinhala-english. *arXiv preprint arXiv:1902.01382*.

Moonsu Han, Minki Kang, Hyunwoo Jung, and Sung Ju Hwang. 2019. Episodic memory reader: Learning what to remember for question answering from streaming data. *arXiv preprint arXiv:1903.06164*.

Junxian He, Chunting Zhou, Xuezhe Ma, Taylor Berg-Kirkpatrick, and Graham Neubig. 2021. Towards a unified view of parameter-efficient transfer learning. *Cornell University - arXiv,Cornell University - arXiv*.

Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin De Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. 2019. Parameter-efficient transfer learning for nlp. In *International Conference on Machine Learning*, pages 2790–2799. PMLR.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2021. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*.

Chengsong Huang, Qian Liu, Bill Yuchen Lin, Tianyu Pang, Chao Du, and Min Lin. 2023. Lorahub: Efficient cross-task generalization via dynamic lora composition. *arXiv preprint arXiv:2307.13269*.

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.

Daniel Khashabi, Snigdha Chaturvedi, Michael Roth, Shyam Upadhyay, and Dan Roth. 2018. Looking beyond the surface: A challenge set for reading comprehension over multiple sentences. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 252–262.

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, et al. 2019. Natural questions: a benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, 7:453–466.

Guokun Lai, Qizhe Xie, Hanxiao Liu, Yiming Yang, and Eduard Hovy. 2017. Race: Large-scale reading comprehension dataset from examinations. *arXiv preprint arXiv:1704.04683*.

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. In *International Conference on Learning Representations*.

Brian Lester, Rami Al-Rfou, and Noah Constant. 2021. The power of scale for parameter-efficient prompt tuning. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 3045–3059.

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

Patrick Lewis, Pontus Stenetorp, and Sebastian Riedel. 2020. Question and answer test-train overlap in open-domain question answering datasets. *arXiv preprint arXiv:2008.02637*.

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

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. *arXiv preprint arXiv:2310.18339*.

Xiaodong Liu, Yu Wang, Jianshu Ji, Hao Cheng, Xueyun Zhu, Emmanuel Awa, Pengcheng He, Weizhu Chen, Hoifung Poon, Guihong Cao, et al. 2020. The microsoft toolkit of multi-task deep neural networks for natural language understanding. *arXiv preprint arXiv:2002.07972*.

Shashi Narayan, Shay B Cohen, and Mirella Lapata. 2018. Don't give me the details, just the summary! topic-aware convolutional neural networks for extreme summarization. *arXiv preprint arXiv:1808.08745*.

Ella Neeman, Roee Aharoni, Or Honovich, Leshem Choshen, Idan Szpektor, and Omri Abend. 2022. Disentqa: Disentangling parametric and contextual knowledge with counterfactual question answering. *arXiv preprint arXiv:2211.05655*.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. 2022. Training language models to follow instructions with human feedback. *Advances in Neural Information Processing Systems*, 35:27730–27744.

Junting Pan, Ziyi Lin, Xiatian Zhu, Jing Shao, and Hongsheng Li. 2022. St-adapter: Parameter-efficient image-to-video transfer learning. *Advances in Neural Information Processing Systems*, 35:26462–26477.

Fabio Petroni, Tim Rocktäschel, Patrick Lewis, Anton Bakhtin, Yuxiang Wu, Alexander H Miller, and Sebastian Riedel. 2019. Language models as knowledge bases? *arXiv preprint arXiv:1909.01066*.

Peng Qi, Xiaowen Lin, Leo Mehr, Zijian Wang, and Christopher D Manning. 2019. Answering complex open-domain questions through iterative query generation. *arXiv preprint arXiv:1910.07000*.

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. *Advances in Neural Information Processing Systems*, 34:8583–8595.

Adam Roberts, Colin Raffel, and Noam Shazeer. 2020. How much knowledge can you pack into the parameters of a language model? *arXiv preprint arXiv:2002.08910*.

Amrita Saha, Rahul Aralikatte, Mitesh M Khapra, and Karthik Sankaranarayanan. 2018. Duorc: Towards complex language understanding with paraphrased reading comprehension. *arXiv preprint arXiv:1804.07927*.

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

Noam Shazeer, Azalia Mirhoseini, Krzysztof Maziarz, Andy Davis, Quoc Le, Geoffrey Hinton, and Jeff Dean. 2016. Outrageously large neural networks: The sparsely-gated mixture-of-experts layer. In *International Conference on Learning Representations*.

Ying Sheng, Shiyi Cao, Dacheng Li, Coleman Hooper, Nicholas Lee, Shuo Yang, Christopher Chou, Banghua Zhu, Lianmin Zheng, Kurt Keutzer, et al. 2023. S-lora: Serving thousands of concurrent lora adapters. *arXiv preprint arXiv:2311.03285*.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

Xiao Wang, Tianze Chen, Qiming Ge, Han Xia, Rong Bao, Rui Zheng, Qi Zhang, Tao Gui, and Xuanjing Huang. 2023a. Orthogonal subspace learning for language model continual learning. *arXiv preprint arXiv:2310.14152*.

Xiao Wang, Weikang Zhou, Can Zu, Han Xia, Tianze Chen, Yuansen Zhang, Rui Zheng, Junjie Ye, Qi Zhang, Tao Gui, et al. 2023b. Instructuie: Multi-task instruction tuning for unified information extraction. *arXiv preprint arXiv:2304.08085*.

Yizhong Wang, Swaroop Mishra, Pegah Alipoormolabashi, Yeganeh Kordi, Amirreza Mirzaei, Anjana Arunkumar, Arjun Ashok, Arut Selvan Dhanasekaran, Atharva Naik, David Stap, et al. 2022. Super-naturalinstructions: Generalization via declarative instructions on 1600+ nlp tasks. *arXiv preprint arXiv:2204.07705*.

Zhiguo Wang, Wael Hamza, and Radu Florian. 2017. Bilateral multi-perspective matching for natural language sentences. *arXiv preprint arXiv:1702.03814*.

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

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

Chunting Zhou, Pengfei Liu, Puxin Xu, Srini Iyer, Jiao Sun, Yuning Mao, Xuezhe Ma, Avia Efrat, Ping Yu, Lili Yu, et al. 2023. Lima: Less is more for alignment. *arXiv preprint arXiv:2305.11206*.

Yanqi Zhou, Tao Lei, Hanxiao Liu, Nan Du, Yanping Huang, Vincent Zhao, Andrew M Dai, Quoc V Le, James Laudon, et al. 2022. Mixture-of-experts with expert choice routing. *Advances in Neural Information Processing Systems*, 35:7103–7114.

A Details about Experiment Implementation

Datasets. The seven tasks are closed-book question answering (CBQA), coreference resolution, natural language inference (NLI), abstract summarization, multi-lingual translation, reading comprehension, and text classification. Table 4 shows the composition of the 3-million-sample dataset. The five million fine-tuning data we use includes three million versions and their variants from data augmentation strategies. The 1-million-sample version is the subset of the original 3-million-sample dataset.

Evaluation. We utilize the opencompass[6] framework to run the evaluation process on the aforementioned tasks. Notably, considering previous work that has noted train-test overlap in CBQA datasets (Lewis et al., 2020), we elaborately select parts of the CBQA dataset without train-test overlap for our testing set, namely *Filtered NQ* and *Filtered TriviaQA*, to analyze the world knowledge of models better.

B The World Knowledge of LLM Further Declines after Being Trained with More Data

With the task types increasing, there is an inevitable trend to increase the amount of SFT training data. To further verify that a large-scale SFT training process can lead to knowledge forgetting of LLM as stated in Section 2, we construct a much larger dataset containing ten million training samples. In addition to the dataset from the previous section, we also added the following tasks:

- Named Entity Recognition: sampled from Wang et al. (2023b). Contains 17 different NER tasks.
- Program Execution: sampled from Wang et al. (2022). Contains 90 different tasks requiring the LLM to understand the instructions about a program and execute it.
- Question Generation: sampled from a existing huggingface dataset [7]. Given a context, the LLM needs to generate an appropriate question based on the answer.
- Text2sql: sampled from two existing huggingface datasets[8]. Given a description in natural language, the LLM needs to generate an appropriate sequence of SQL.

[6] https://opencompass.org.cn/

[7] https://huggingface.co/datasets/qa_zre

[8] https://huggingface.co/datasets/Clinton/Text-to-sql-v1, https://huggingface.co/datasets/cfq

Task Name	# Train	# Test	Task Type
TriviaQA (Han et al., 2019)	78785	254	closed-book QA
NQ (Kwiatkowski et al., 2019)	104071	357	closed-book QA
HotpotQA (Qi et al., 2019)	72798	5622	closed-book QA
WSC (Levesque et al., 2012)	554	146	coreference resolution
WinoGrande (Sakaguchi et al., 2021)	40398	1767	coreference resolution
Flores (Guzmán et al., 2019)	0	1600	machine translation
WMT [2]	500000	-	machine translation
RTE [3]	2490	3000	NLI
ReCoRD (Zhang et al., 2018)	100730	10000	reading comprehension
AX-g [4]	0	356	NLI
multiRC (Khashabi et al., 2018)	27243	9693	reading comprehension
anli r1/r2/r3 (Liu et al., 2020)	162874	-	NLI
qqp (Wang et al., 2017)	363846	-	NLI
Xsum (Narayan et al., 2018)	204045	11334	single-document summarization
Race (Lai et al., 2017)	87866	4934	reading comprehension
duorc-selfRC (Saha et al., 2018)	60721	-	reading comprehension
AG-news (Zhang et al., 2015)	120000	-	topic classification
yelp review (Zhang et al., 2015)	650000	-	sentiment classification
openai tldr [5]	232188	-	summarization

Table 4: Details about the tasks in our fine-tuning dataset. "-" means we do not use the test set of this dataset for evaluation.

- Toxic Classification: sampled from a existing huggingface datasets[9].

After training the LLaMa-2-7b on this 10-million-sample dataset with the same experiment setup with Appendix A, we find the LLM exhibit a greater knowledge-forgetting but a promising performance in other tasks apart from knowledge benchmarks.

C Mixed Distribution Dilemmas for Expert Balancing

When fine-tuning MoE without any constraints, the router mechanism often converges to a state in which a small number of experts receive a disproportionately large share of preferences by the router, as depicted in Figure 5. This imbalance among experts presents a challenge to correct, as experts that receive greater routing weights in the early stages of training undergo more rapid optimization, thereby garnering increased preferences from the router. A similar phenomenon has been documented in the work presented in Shazeer et al. (2016) and Fedus et al. (2021).

A conventional solution for balancing experts utilization involves employing the coefficient of

Task Name	Baseline	Result
NER	42.1	82.2
Program Execution	18.7	78.5
Toxic Classification	96	97.4
Question Generation	46.2	61.1
Text2sql	56	96.2
WSC	65.4	70.2
winogrande	61.7	66.1
Flores	0.1	26.0
Xsum	19.7	33.2
Race-middle	30.5	87.0
Race-high	30.4	83.3
RTE	52.7	87.4
ReCoRD	29.5	56.6
AX-g	52.0	87.9
multiRC	44.0	86.0
TriviaQA	52.2	30.9
NQ	18.5	14.2
Filtered TriviaQA	33.5	15.7
Filtered NQ	7.8	5.0
HotpotQA	11.2	7.6

Table 5: Performance of Llama-2-7B after vanilla SFT with a 10-million-sample datasets. There is a much more severe decrease in the performance on the CBQA tasks, while a great enhancement in other tasks compared with the baseline.

[9] https://huggingface.co/datasets/google/civil_comments

variation of the experts' importance as the loss function, aimed at equalizing the significance of each expert (Shazeer et al., 2016). This solution assumes that the distribution of training samples for optimising MoE is a single distribution, which inherently eliminates the necessity of considering the diverse origins of data distribution. Specifically, this traditional approach simplifies the modeling process by assuming homogeneity in data sources that often do not align with fine-tuning data containing both factual knowledge QA and other downstream tasks. Therefore, such simplification can lead to significant biases, particularly when encountering datasets with varied distributional characteristics.

Traditional balancing constraints, which aim to allocate a uniform distribution of training samples across all experts, can lead to inaccurate parameter estimation. This is because such constraints do not account for the intrinsic differences in data representation and importance across various categories. Recognizing the disparate nature of data distributions, LoRAMoE strategically assigns data to experts, not uniformly, but based on the observed imbalances. This allocation is governed by a set of weights that are calibrated to reflect the varying significance and representation of different data categories within the overall dataset.

Such a specialized allocation method is pivotal in addressing the challenges posed by uneven data distributions. By tailoring the distribution of training samples to each expert based on the inherent disparities in the data, LoRAMoE facilitates a more accurate and representative parameter estimation. This nuanced approach to data distribution allows for a more effective fitting of the model to diverse data subsets, significantly enhancing the model's predictive accuracy and generalization capability. This strategy is particularly effective in scenarios where data imbalance could otherwise lead to skewed learning and generalization errors, ensuring that each data category is appropriately represented and modeled within the overall system.

To illustrate the concept with a simplified model, let's assume our training data is sampled from a mixture of two Gaussian distributions. The means (μ_1, μ_2) and variances (σ_1^2, σ_2^2) of these distributions are implicit. The proportion of training data from each distribution is denoted as $p_1$andP_2 where $p_1 + p_2 = 1$, without loss of generality, we assume that $p_1 \leq p_2$. When a MoE model fits the proposed distribution with balanced weights m, the likelihood of the model given the data can be expressed as:

$$\begin{aligned} L(\mathbf{X}) = \prod_{x \in \mathbf{X}_1} & \left(m\mathcal{N}\left(x; \mu_1', \sigma_1'^2\right)\right. \\ & \left.+(1-m)\mathcal{N}\left(x; \mu_2', \sigma_2'^2\right)\right) \\ & \times \prod_{x \in \mathbf{X}_2} \left(m\mathcal{N}\left(x; \mu_1', \sigma_1'^2\right)\right. \\ & \left.+(1-m)\mathcal{N}\left(x; \mu_2', \sigma_2'^2\right)\right), \end{aligned} \tag{10}$$

where $Card(\mathbf{X}_1) : Card(\mathbf{X}_2) = p_1 : p_2$. Using $\mathcal{N}_1(x)$ and $\mathcal{N}_2(x)$ for $\mathcal{N}\left(x; \mu_1', \sigma_1'^2\right)$ and $\mathcal{N}\left(x; \mu_2', \sigma_2'^2\right)$,

The optimal mean value for μ_1' satisfies the following conditions, whose value is 0 when the fitted distribution is in the same family of mixed distributions $\mathbb{N}(\theta, p_1)$ as the sampling distribution:

$$\begin{aligned} \frac{\partial \log L(\mathbf{X})}{\partial \mu_1'} = & \sum_{x \in \mathbf{X}_1 \cup \mathbf{X}_2} \frac{\partial}{\partial \mu_1'} \log\left(m\mathcal{N}_1(x)\right. \\ & \left.+(1-m)\mathcal{N}_2(x)\right) \\ = & \sum_{x \in \mathbf{X}_1 \cup \mathbf{X}_2} \left(\frac{x - \mu_1'}{\sigma_1'^2}\right) \\ & \times \frac{m\mathcal{N}_1(x)}{m\mathcal{N}_1(x) + (1-m)\mathcal{N}_2(x)}, \end{aligned} \tag{11}$$

In equation 10, we can replace part of the summation with the empirical estimate of the mean of the input x. For an ideal routing network, there must exist a distribution N_i such that the data allocated to this distribution is independently and identically distributed with one of the peaks in the sampling distribution. Let's assume this distribution to be N_2. In this case, if $m \geq p_1$, then the fitting result for distribution $\mu_{1'}$ will be $\mu_1' = (p_1\mu_1 + (m - p_1)\mu_2)/m$. Based on the chain rule of differential derivation, we end up with:

$$\begin{aligned} \frac{d \log L}{dm} = & \frac{\partial \log L}{\partial \mu_1'} \frac{d\mu_1'}{dm} \\ = & \left(\sum_{x \in \mathbf{X}_1 \cup \mathbf{X}_2} \left(\frac{x - \mu_1'}{\sigma_1'^2}\right) \right. \\ & \left. \times \frac{m\mathcal{N}_1(x)}{m\mathcal{N}_1(x) + (1-m)\mathcal{N}_2(x)} \right) \\ & \times \frac{p_1(\mu_2 - \mu_1)}{m^2} \\ \leq & 0, \end{aligned} \tag{12}$$

The inverse result can be derived similarly. Therefore, the best training error is achieved only when the mixing coefficient m of the prior distribution is consistent with the actual sampling distribution weight p_1.

Task Name	# Expert=8 # rank=4	# Expert=4 # rank=4	# Expert=6 # rank=8	# Expert=6 # rank=16
WSC	71.2	76.0	70.2	76.9
winogrande	69.8	56.0	69.5	70.9
Flores	25.0	25.8	26.1	26.3
Xsum	32.8	33.3	33.7	34.0
Race-middle	90.3	84.2	90.3	90.5
Race-high	87.1	80.7	87.3	87.2
RTE	84.5	80.1	88.1	85.2
ReCoRD	85.6	85.5	86.0	86.1
AX-g	88.8	77.5	88.2	85.7
multiRC	77.2	87.6	81.1	87.3
TriviaQA	54.4	57.8	58.2	58.9
NQ	25.6	27.9	27.8	28.2
Filtered TriviaQA	30.7	35.8	36.7	34.3
Filtered NQ	11.5	13.4	12.0	15.4
HotpotQA	14.5	16.0	16.4	16.5

Table 6: Detailed result on sensitivity study.

D Detalied Results of Sensitivity Study

Table 6 shows the detailed results presented in Section 4.3.