MoE-LPR: Multilingual Extension of Large Language Models through Mixture-of-Experts with Language Priors Routing

Hao Zhou[1*], Zhijun Wang[1*], Shujian Huang[1†],
Xin Huang[2], Xue Han[2], Junlan Feng[2], Chao Deng[2], Weihua Luo[3], Jiajun Chen[1]

[1]National Key Laboratory for Novel Software Technology, Nanjing University, China
[2]China Mobile Research Beijing, China;
[3]Alibaba International Digital Commerce,China
{zhouh,wangzj}@smail.nju.edu.cn,{huangsj,chenjj}@nju.edu.cn,
{huangxinyjy,hanxueai,fengjunlan,dengchao}@chinamobile.com, weihua.luowh@alibaba-inc.com

Abstract

Large Language Models (LLMs) are often English-centric due to the disproportionate distribution of languages in their pre-training data. Enhancing non-English language capabilities through post-pretraining often results in catastrophic forgetting of the ability of original languages. Previous methods either achieve good expansion with severe forgetting or slight forgetting with poor expansion, indicating the challenge of balancing language expansion while preventing forgetting. In this paper, we propose a method called **MoE-LPR** (**M**ixture-**o**f-**E**xperts with **L**anguage **P**riors **R**outing) to alleviate this problem. MoE-LPR employs a two-stage training approach to enhance the multilingual capability. First, the model is post-pretrained into a Mixture-of-Experts (MoE) architecture by upcycling, where all the original parameters are frozen and new experts are added. In this stage, we focus improving the ability on expanded languages, without using any original language data. Then, the model reviews the knowledge of the original languages with replay data amounting to less than 1% of post-pretraining, where we incorporate language priors routing to better recover the abilities of the original languages. Evaluations on multiple benchmarks show that MoE-LPR outperforms other post-pretraining methods. Freezing original parameters preserves original language knowledge while adding new experts preserves the learning ability. Reviewing with LPR enables effective utilization of multilingual knowledge within the parameters. Additionally, the MoE architecture maintains the same inference overhead while increasing total model parameters. Extensive experiments demonstrate MoE-LPR's effectiveness in improving expanded languages and preserving original language proficiency with superior scalability.

Introduction

Large Language Models (LLMs) such as ChatGPT (OpenAI 2023), GPT-4 (Achiam et al. 2023), Llama2 (Touvron et al. 2023), Llama3 (Dubey et al. 2024), and Qwen (Bai et al. 2023) have demonstrated remarkable performance across different tasks, including multiple-choice question-answering (Robinson and Wingate 2023), summarization (Pu, Gao, and Wan 2023), and reasoning (Yu et al. 2023). However, many studies have highlighted a significant discrepancy between performances on English and non-English tasks (Gao et al. 2024; Wang et al. 2024).

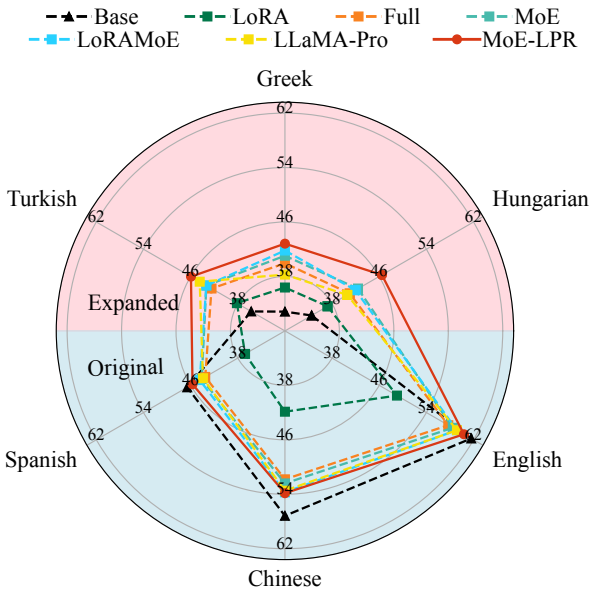

Figure 1: MoE-LPR performs the best in both expanded languages and original languages. We define expanded languages as languages that the model is not very good at and we are going to enhance, and original languages as languages that the model is relatively strong in and prone to catastrophic forgetting.

Pre-training a LLM with data from multiple languages may achieve better multilingual capabilities, but highly resource-intensive and often impractical given limited computational budgets. Consequently, current research predominantly focus on post-pretraining (also known as continue training) techniques (Csaki et al. 2024; Kuulmets et al. 2024), which carry out further multilingual pre-training on

*These authors contributed equally.
†Corresponding author.

a pre-trained LLM, aiming to inject extensive language knowledge for certain language(s). Despite its efficiency, this method significantly increases the risk of catastrophic forgetting, where the performance of LLMs in the languages they are initially good at (such as English or Chinese) may dramatically decline. As a result, improving the performance of expanded languages while maintaining the performance of existing ones becomes a critical challenge in the field.

To prevent forgetting, existing work (Dou et al. 2024; Wu et al. 2024) usually retain the original parameters of the model as much as possible, and train new parameters to fit knowledge for new languages. However, less attention is paid on effectively incorporating these new and old parameters for tasks in different languages. In this paper, we propose a novel two-stage training method called Mixture-of-Experts with Language Priors Routing (**MoE-LPR**) that improves multilingual capability with the retention of original language proficiency. MoE-LPR contains two stages: post-pretraining with MoE and review with LPR.

In the post-pretraining stage, we upcycle the LLM into a MoE architecture and post-pretrain the newly added parameters with a substantial amount of high-quality monolingual data, while keeping the original parameters frozen. This ensures that the original capabilities of the model are preserved while expanding its proficiency in additional languages. We also incorporate load balancing loss to unleash the model's learning potential and maintain training stability. In the review stage, we further train the router to better utilize the experts for different languages. We design LPR training to recover the model's capabilities in its original languages using replay data that amounts to less than 1% of the post-pretraining corpus.

As shown in Figure 1, experiment results demonstrate that our method not only significantly improves proficiency in newly expanded languages (languages in the top half) but also substantially retains the model's capabilities in its original languages (languages in the bottom half). Moreover, our approach allows for easy upscaling for the number of model parameters while maintaining a fixed inference overhead. Our approach represents a step forward in developing LLMs that are both powerful and versatile across a wide range of languages, addressing the critical need for more inclusive and effective NLP technologies in a multilingual world. The contributions of our proposed method are as follows:

- Two-Stage Training Strategy: MoE-LPR employs a two-stage training strategy, with a special focus on balancing the capability of newly expanded languages and the original languages.
- Language Priors Routing: MoE-LPR introduces the LPR mechanism to mitigate catastrophic forgetting of original languages with replay data amounting to less than 1% of the post-pretraining corpus. LPR also exhibits excellent generalization to languages it has not been trained on.
- Scalability: MoE-LPR allows for easy upscaling of model parameters without increasing inference overhead or risking catastrophic forgetting, making it a cost-effective and stable solution for multilingual NLP tasks.

Methodology

Figure 2 describes the overall framework of our MoE-LPR. In the post-pretraining with MoE stage, we train the new experts on a large amount of monolingual data in the expanded languages for injecting language knowledge. In the review with LPR stage, we train the router on a small amount of monolingual data in both the expanded and original languages for better utilizing the experts.

Post-pretraining with MoE

As shown in Figure 2, inspired by Mixtral (Jiang et al. 2024) and upcycling (Komatsuzaki et al. 2022), we upcycle the dense model to a MoE model by copying the FFN parameters and incorporating a router matrix $W_r \in \mathbb{R}^{h \times N}$ in each layer, where h represents the token dimension and N denotes the number of experts within the model.

The router in MoE allows the model to dynamically select the most suitable experts. Formally, let $x \in \mathbb{R}^h$ be a token representation, the router score is expressed as:

$$G(x) = \text{Softmax}(x \cdot W_r) \tag{1}$$

where $G(x) \in \mathbb{R}^N$. After obtaining router scores, We select the index set $\mathcal{T}$ of the top-K experts and combine their outputs using normalized weights from the router scores to obtain the final representation as:

$$\mathcal{T} = \{i | G_i(x) \in \text{Topk}(G(x), K)\} \tag{2}$$

$$y = \sum_{i \in \mathcal{T}} \frac{G_i(x)}{\sum_{j \in \mathcal{T}} G_j(x)} E_i(x) + x \tag{3}$$

where $G_i(x)$ and $E_i(x)$ represent the router score and the output of the i-th expert respectively, and K denotes the number of activated experts.

To enhance the multilingual capability of the MoE model while preserving its performance in the original languages, we freeze the parameters of the original dense model. During post-pretraining on the expanded language corpus, we only update the parameters of the newly added experts and the router, which ensures that the core knowledge embedded in the initial model remains intact.

The model is trained with a combination of a next token prediction loss and a load balancing loss as follows.

Next Token Prediction Loss. Given an expanded language corpus D, a batch $\mathcal{B}$ with T tokens, and N experts indexed by i from 0 to $N-1$, where index 0 is used to denote the original dense FFN, the post-pretraining next token prediction loss is:

$$L_{\text{NTP}}(\theta_{\text{new}}, W_r) = -\sum_{i=1}^{|\mathcal{B}|} \sum_{j=1}^{|d^i|} \log p_{\mathcal{M}}\left(d_j^i \mid d_{<j}^i\right) \tag{4}$$

where $\mathcal{M}$ denotes the whole MoE model, θ_{new} indicates the parameters of the newly added experts and W_r is the parameter of the router.

Load Balancing Loss. We also use an expert-level load balance loss (Fedus, Zoph, and Shazeer 2022) to mitigate the risk of routing collapse:

$$L_{\text{balance}}(\theta_{\text{new}}, W_r) = \sum_{i=1}^{N} f_i P_i \tag{5}$$

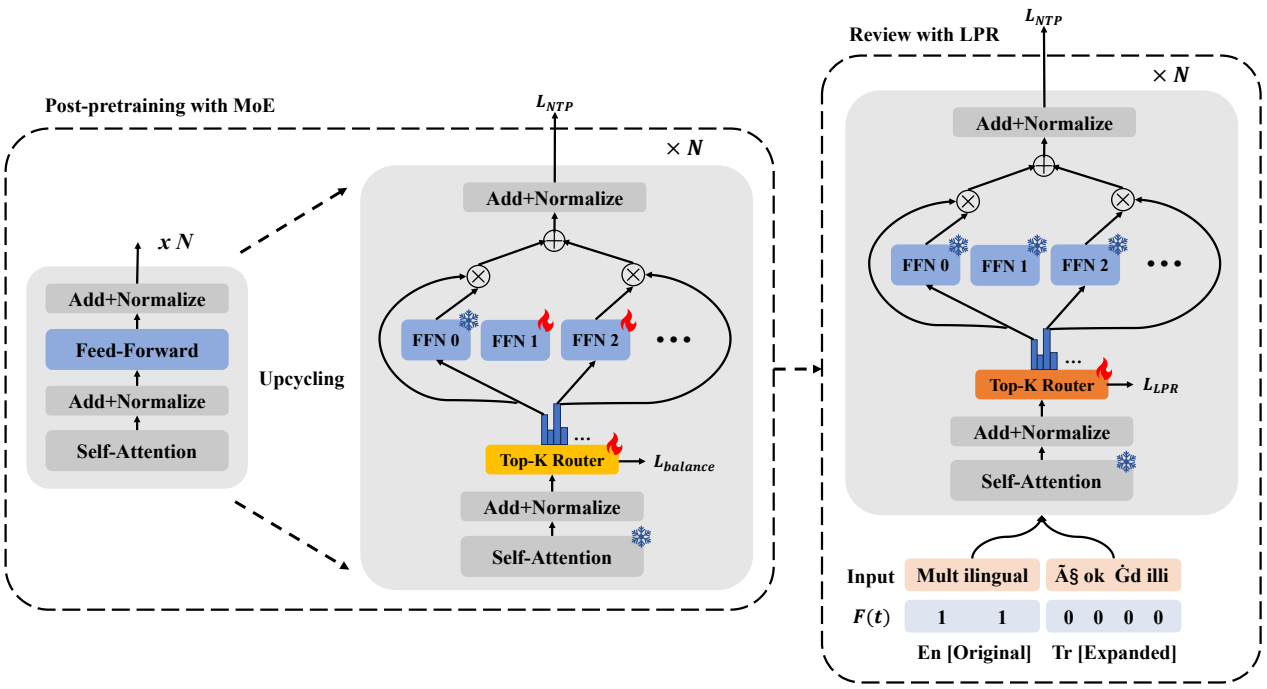

Figure 2: Overall framework of our MoE-LPR. Two-stage strategy is performed to enhance the multilingual capability.

$$f_i = \frac{N}{KT} \sum_{t \in \mathcal{B}} \mathbb{1}\{\text{Token t selects expert i}\} \tag{6}$$

$$P_i = \frac{1}{T} \sum_{t \in \mathcal{B}} G_i(t) \tag{7}$$

where $\mathbb{1}$ denotes the indicator function. We opt for a top-2 strategy by setting $K = 2$ to select the two most suitable experts with normalization, intending to achieve a trade-off between inference overhead and learning capabilities.

The final optimization objective during post-pretraining is:

$$\underset{\theta_{\text{new}}, W_r}{\operatorname{argmin}} L_{\text{NTP}} + \alpha L_{\text{balance}} \tag{8}$$

where α is a hyper-parameter that controls the weight of the load balancing loss.

Review with LPR

After post-pretraining on the expanded language corpus, the router, which has only been trained on the expanded languages but not on the original languages, may incorrectly assign experts for the original languages. This misallocation is also an important factor for catastrophic forgetting in the MoE model. Therefore, we design this review stage to train the model to deal with both original and expanded languages.

As the router is the main source of the problem, we only update the parameters of the router and freeze the other parts of the model. Because the number of router parameters accounts for a negligible proportion, this stage could be efficient and requires very little computational resource and training data.

In fact, the amount of original language data used in our review stage, is less than 1% of the post-pretraining corpus. In comparison, traditional replay strategy (Ibrahim et al. 2024) incorporates data from original languages into the post-pretraining stage, which usually requires a much larger amount (25%).

LPR Loss. Intuitively, the routing could be led by language priors: all the original language tokens should be routed to the originally frozen expert (i.e. expert 0 in this case), making the model work exactly the same as before the expansion. Therefore, we design a LPR loss to be a Cross-Entropy loss for the tokens from the original languages, forcing the top-1 selection of these tokens to be expert 0, where the top-1 selection refers to the expert selection with the highest routing score.

Formally, considering original language tokens set D_{original} and the indicator function $\mathbf{F}(t)$:

$$\mathbf{F}(t) = \begin{cases} 1 & \text{if } t \in D_{\text{original}}, \\ 0 & \text{if } t \notin D_{\text{original}}. \end{cases} \tag{9}$$

The LPR loss is defined as:

$$L_{\text{LPR}}(W_r) = -\sum_{t \in \mathcal{B}} \mathbf{F}(t) \log G_0(t) \tag{10}$$

where index 0 denotes the originally frozen expert.

In practice, when training with LPR loss, we remove the load balancing loss in Eq. (8). The final optimization objective for the review stage is:

$$\underset{W_r}{\operatorname{argmin}} L_{\text{NTP}} + \gamma L_{\text{LPR}} \tag{11}$$

where γ is a hyper-parameter that controls the weight of the LPR loss.

Experiments

Experiment Setup

Given the focus on multilingual capability enhancement, we introduce the language selection first. Then follow the training details, several baselines, and the evaluation details.

Model and Languages We choose Qwen-1.5 [1] as our base model. The 1.8B version of the Qwen-1.5 series is selected for its lower computation overhead and ease of upcycling. For our study, we choose three low-resource languages as the expanded languages where Qwen-1.5-1.8B performs poorly as shown in Figure 1: Greek (El), Hungarian (Hu), and Turkish (Tr). Additionally, we select three high-resource languages as the original languages to observe the catastrophic forgetting phenomenon: English (En), Chinese (Zh), and Spanish (Es).

Details of Post-pretraining We construct a dataset focusing on the three expanded languages by sampling 8 billion tokens from the monolingual data of each language in CulturalX (Nguyen et al. 2024), a substantial multilingual dataset with 6.3 trillion tokens in 167 languages. Our base model, Qwen-1.5-1.8B, is upcycled into the MoE structure with 5 newly added FFN (6 experts in total). We post-pretrain this model with the 24 billion tokens, marking only the new experts and the router as trainable. The training setup includes a batch size of 512, a sequence length of 1024, a learning rate of 5e-5, and a cosine learning rate scheduler. We incorporate the load balancing loss with a weight of 0.01 and utilize bf16 mixed precision and flash attention (Dao et al. 2022) to speed up the training process.

Our experiments are conducted on 8 A800 GPUs, involving 45856 steps, totaling approximately 848 A800 GPU hours.

Details of Review We randomly sample 50K documents for each original language and 100K documents for each expanded language. The English data are sampled from Slimpajama (Soboleva et al. 2023), the Chinese data from SkyPile-150B (Wei et al. 2023), and the Spanish data from CulturalX (Nguyen et al. 2024). The number of tokens in original languages is 0.138B, accounting for less than 1% of the post-pretraining data (24B). As for the three expanded languages, we sample from the post-pretraining dataset. We concatenate these data for the review stage training. We employ a batch size of 512, a sequence length of 512, a learning rate of 5e-5, and a cosine learning rate scheduler. The load balancing loss is removed and the LPR loss is added as introduced in Eq. (10) with a weight of 0.1. Only the router parameters are trainable. Bf16 mixed precision and flash attention (Dao et al. 2022) mechanism is used for training.

Baselines We conducted experiments on several existing baseline methods trained on the same data, including the small amount of replay data, to ensure that our approach is competitive and effective.

- **Full Fine-tuning**: Fine-tune all parameters directly on the dense model.
- **LoRA** (Hu et al. 2021): The LoRA targets include all linear modules. We set the LoRA rank to 8.
- **MoE**: The same settings as MoE-LPR except for training all the parameters only in one post-pretraining stage.
- **LoRAMoE** (Dou et al. 2024): A novel framework combines multiple LoRAs with a router network to effectively learn new knowledge while avoiding catastrophic forgetting. The router selects all LoRAs for each token. We set the number of LoRAs as 8 and a LoRA rank of 180 to match the same inference overhead.
- **LLaMA-Pro** (Wu et al. 2024): A method is considered where a dense LLM periodically duplicates and inserts new transformer blocks at fixed layer intervals. During post-pretraining, only these newly added transformer blocks are trained to acquire new knowledge while preserving the original knowledge. We add 12 new layers because this is the best setting in our experiments.

Evaluation Details We evaluate our method on several benchmarks including multiple-choice tasks and generation tasks. Examining the model's multilingual capabilities from multiple perspectives.

- **ARC-Challenge (25-shot)** (Clark et al. 2018): A benchmark for evaluating comprehension and reasoning across diverse academic fields.
- **MMLU (5-shot)** (Hendrycks et al. 2020): A multiple-choice dataset testing general knowledge and problem-solving across various subjects.
- **HellaSwag (10-shot)** (Zellers et al. 2019): A dataset with 70k questions for studying grounded commonsense inference.
- **Belebele (5-shot)** (Bandarkar et al. 2023): A machine reading comprehension dataset covering 122 language variants.
- **FLORES-101 (8-shot)** (Goyal et al. 2022): A parallel corpus for evaluating multilingual translation capabilities. We report the performance evaluated by COMET (Rei et al. 2022) [2]

We mainly follow Okapi (Lai et al. 2023) to evaluate the multilingual versions of ARC-Challenge, MMLU and HellaSwag, which are translated from the original English version using GPT-3.5-turbo or DeepL.

Experiment Results

Table 1 presents the performance of various methods across different benchmarks for both expanded and original languages. We report here the performance of the best setting of all baselines. With the additional small amount of replay data, full fine-tuning outperforms LoRA in preventing catastrophic forgetting but still drops about 4 points in original languages. Full fine-tuning can recover to 92.1% performance in original languages with replay data amounting to less than 1% of the post-pretraining data. Ibrahim

[1] Qwen-1.5 has a powerful multilingual tokenizer that produces shorter sequences in expanded languages, which means we don't have to worry about vocabulary expansion.

[2] We use the wmt22-comet-da version.

Model	n_{params}	$n_{\text{act-params}}$	ARC	MMLU	HellaSwag	Belebele	Flores	Avg.
Expanded Languages								
Qwen1.5-1.8B	1.8B	1.8B	23.13	30.97	29.15	33.15	55.40	34.36
LoRA (Hu et al. 2021)	1.8B	1.8B	23.89	29.30	29.78	26.93	55.19	33.02
Full Fine-tuning	1.8B	1.8B	25.98	33.18	35.28	33.70	77.48	41.12
LLaMA-Pro (Wu et al. 2024)	2.4B	2.4B	24.35	34.02	33.85	31.52	81.76	41.10
MoE	5.8B	2.6B	26.43	**35.07**	37.01	32.74	80.01	42.25
LoRAMoE (Dou et al. 2024)	2.6B	2.6B	26.63	34.17	37.17	32.81	81.09	42.37
MoE-LPR	5.8B	2.6B	**28.43**	34.10	**41.06**	**39.93**	**81.83**	**45.07**
Original Languages								
Qwen1.5-1.8B	1.8B	1.8B	33.48	47.55	49.82	56.52	82.50	53.97
LoRA (Hu et al. 2021)	1.8B	1.8B	28.33	37.42	41.48	39.45	75.49	44.43
Full Fine-tuning	1.8B	1.8B	31.72	43.51	47.38	45.26	80.77	49.73
LLaMA-Pro (Wu et al. 2024)	2.4B	2.4B	31.77	44.06	48.36	48.78	81.97	50.99
MoE	5.8B	2.6B	32.51	44.16	48.54	45.37	81.63	50.44
LoRAMoE (Dou et al. 2024)	2.6B	2.6B	32.43	**45.41**	48.61	47.74	82.03	51.24
MoE-LPR	5.8B	2.6B	**32.71**	44.62	**49.12**	**51.81**	**82.36**	**52.12**

Table 1: Evaluation results in expanded and original languages. n_{params} is the total number of model parameters, $n_{\text{act-params}}$ is the number of activated model parameters per token. The best and second-best results are marked in **bold** and underlined fonts.

et al. (2024) demonstrates that training new languages suffers from dramatic distribution shifts. Only when using more than 25% replay data can the model recover to more than 95.7% performance, indicating that significant language shifts in post-pretraining data require more replay data and computational overhead. However, our MoE-LPR can recover to 96.6% performance (52.12/53.97) with less than 1% replay data.

LoRA performs poorly in expanded languages due to the excessive data in the post-pretraining stage. We also experiment with LoRA at rank=64 to achieve comparable effects in expanded languages, but this results in worse catastrophic forgetting.

LLaMA-Pro demonstrates a strong ability to retain knowledge, but its performance in expanded languages is only comparable to full fine-tuning, with the drawback of higher inference overhead. LoRAMoE performs better than other baselines in both expanded and original languages. Our proposed method, MoE-LPR, surpasses LoRAMoE by 2.7 points in expanded languages and by 0.88 points in original languages on average. While adding more new parameters, the inference overhead of LLaMA-Pro and LoRAMoE increases accordingly, while that of MoE-LPR does not. More details about scaling will be discussed in the following sections.

The results also demonstrate that MoE underperforms our MoE-LPR both in expanded and original languages, which implies that freezing all the original parameters will not limit the model's learning ability. In contrast, the frozen parameters contribute a robust basic capabilities of the model during post-pretraining, resulting in significant performance improvement.

Ablation & Analysis

Review with LPR

Model	Expanded	Original	Avg.
Qwen1.5-1.8B	34.36	53.97	44.17
LoRAMoE	42.37	51.24	46.81
MoE-LPR w/o EC	38.37	49.28	43.83
MoE-LPR w/o Review	45.04	47.14	46.09
MoE-LPR w/o LPR	**45.13**	51.32	48.23
MoE-LPR	45.07	**52.12**	**48.60**

Table 2: Evaluation average results with different settings. "w/o EC" means without expert-copy, corresponding to randomly initialize the new experts when upcycling.

Performance Gain from Review & EC The review with LPR stage is proposed to recover the capabilities of the original languages. As shown in Table 2, without the review stage, MoE-LPR exhibits severe catastrophic forgetting. However, after review training, the performance in original languages improves substantially, by about 5 points on average, while not harming the performance in expanded languages. Furthermore, the performance in original languages drops without the LPR loss, indicating that the LPR mechanism pushes this ability closer to its upper bound. These results show that the review stage allows the model to learn how to handle both new and old languages.

We also conduct experiment without the Expert-Copy, which means that the parameters of new experts are randomly initialized but not copied from the original FFN. As shown in Table 2, performance in original languages does not suffer a serious decrease, but performance in expanded languages shows a significant decrease. Results imply that

copying the original FFN to construct new experts is important to the learning of expanded language knowledge.

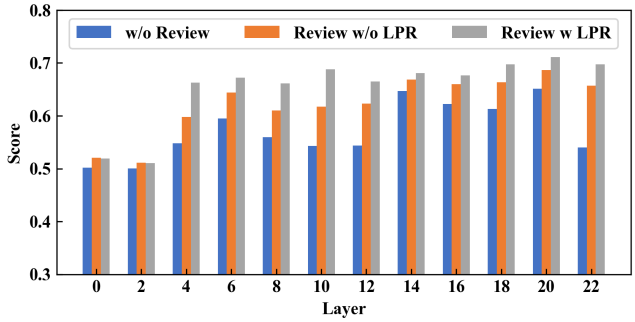

Figure 3: Router scores of the frozen expert for English (original language) tokens in the Belebele benchmark.

Routing Scheme for Different Languages In this section, we examine whether the review stage works properly. As shown in Figure 3, the router scores of the frozen expert on original language tokens show obvious improvement with the review stage. In addition, without the LPR loss, the router scores demonstrate a significant drop. The router scores of the frozen expert on expanded language tokens almost remain unchanged. In the review stage, we optimize the model with only the next token prediction loss for expanded languages. The results show that the next token prediction loss effectively prevents expanded languages from being influenced by the language priors of original languages. These observations indicate that the review stage is functioning correctly, biasing the routing scheme of original language tokens toward the frozen expert.

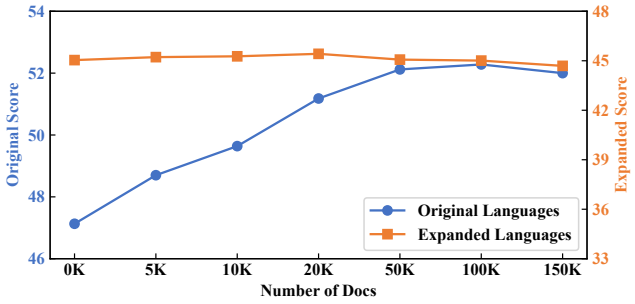

Figure 4: Average scores in expanded and original languages with varying numbers of documents for review.

How much Data is Enough for Review In this section, we experiment with varying numbers of original language documents in the review stage, ranging from 0K to 150K, while maintaining the 1:2 mix of original and expanded languages. As shown in Figure 4, the original language performance continues to improve significantly while the expanded language performance continues to decrease slightly. After 50k, the original language performance improvement starts to become slow. Therefore, considering both training cost and effects, we choose 50K as the best data size in this experiment, which amounts to less than 1% of the post-pretraining corpus. Using 50K results in a 4.98 points performance boost in the original languages while almost maintaining the performance in the expanded languages. These results indicate that a small amount of replay data is sufficient for the model to review its original languages.

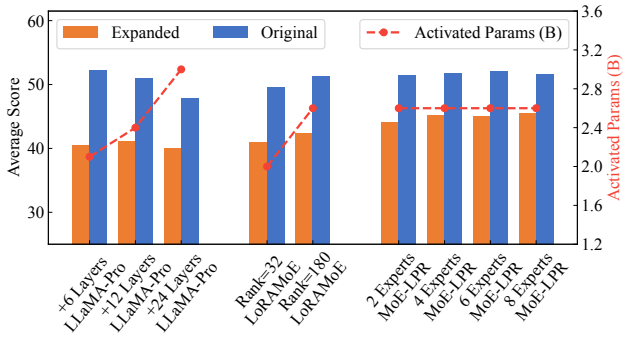

Figure 5: Average scores in expanded and original languages with different model settings. "34.36" and "53.97" refer to the expanded and original language performance of the base model respectively.

Scaling Law

We compare the performance of LLaMA-Pro with different numbers of extending layers, LoRAMoE with different ranks and MoE-LPR with different numbers of experts. All the models are trained on the 24 billion tokens dataset in the three expanded languages.

Figure 5 demonstrates the superior scalability of MoE-LPR. For expanded languages, adding 12 layers to LLaMA-Pro improves performance more than adding 6 layers, but adding 24 layers, matching the base model's layer count, results in a performance drop. Increasing the rank of LoRAMoE from 32 to 180 shows significant improvements. MoE-LPR consistently outperforms these configurations as more experts are added, even with just 2 experts, maintaining a significant advantage over LLaMA-Pro and LoRAMoE. For original languages, LLaMA-Pro suffers from catastrophic forgetting, worsening with more layers. Adding 24 layers even performs worse than full fine-tuning. Although LoRAMoE's catastrophic forgetting does not worsen with increased parameters, it still underperforms MoE-LPR. Even with 8 experts and a 7B parameter size, MoE-LPR can still greatly mitigate catastrophic forgetting.

Unlike LLaMA-Pro and LoRAMoE, whose activated parameters per token increase linearly with more parameters, adding experts to MoE-LPR does not increase the inference overhead. This improves performance in expanded languages while maintaining stable levels of catastrophic forgetting. MoE-LPR demonstrates superior scalability.

Language Generalization

In the review stage, we only use documents of three of the original languages. We conduct evaluations on two addi-

Model	Exp.	Ori. ID	Ori. OOD
Qwen1.5-1.8B	34.36	53.97	46.35
Full Fine-tuning	41.12	49.73	42.46
LLaMA-Pro	41.10	50.99	42.93
LoRAMoE	42.37	51.24	43.41
MoE-LPR w/o LPR	45.13	51.32	44.22
MoE-LPR One-Stage	**45.38**	51.90	43.71
MoE-LPR	45.07	**52.12**	**45.25**

Table 3: Evaluation results in French and Portuguese.

tional high-resource languages that the base model is good at relatively: French and Portuguese to examine the generalization of MoE-LPR when preventing catastrophic forgetting. We name them out-of-domain original languages because the review stage training does not contain tokens in these two languages. Table 3 demonstrates that MoE-LPR successfully generalizes its catastrophic forgetting prevention effect to these languages. Despite the router not being trained on French and Portuguese tokens, our LPR mechanism minimizes the performance gap from the base model for these languages, outperforming other post-pretraining methods. This demonstrates MoE-LPR's excellent language generalization in preventing catastrophic forgetting.

We also try to move the LPR loss and the small amount of replay data to the post-pretraining stage. As shown in Table 3, MoE-LPR One-Stage shows comparable performance to the two-stage strategy. However, it demonstrates worse language generalization, which showcases a 1.54 points performance drop in the out-of-domain original languages. Therefore, we choose the two-stage strategy as a better proposal.

Related Work

Mixture of Experts

Recent studies (Kaplan et al. 2020; Hoffmann et al. 2022) have shown a strong correlation between the number of parameters in a model and its capabilities. When the number of parameters is large, the model demonstrates emergent abilities (Zoph et al. 2022b). Traditional dense models require the activation of all parameters for a given input, significantly increasing computational overhead. Distinct from conventional dense models, Mixture of Experts (MoE) achieves computational feasibility and expanded model capacity by utilizing a router that selectively activates a limited number of experts for each input. There are several works, such as Switch-transformer (Fedus, Zoph, and Shazeer 2022), ST-MoE (Zoph et al. 2022a), Glam (Du et al. 2022) , attempts to train an MoE model from scratch. These works have demonstrated that MoE models can achieve significantly lower loss and performance gains compared to dense models with the same activated parameters and require less energy consumption compared to dense models with the same total parameters. However, considering the huge computational budget, Komatsuzaki et al. (2022) indicates that a sparse MoE model could be initialized from dense models. In the era of LLMs, numerous MoE works have been developed. For instance, Mixtral (Mixtral 2024) adds experts to each layer, increasing the total parameter count to 141B. DeepSeek (DeepSeek-AI 2024) utilizes shared experts, enabling the model to select experts more effectively. Snowflake Arctic (Research 2024)incorporates many fine-grained experts, enhancing the diversity of expert selection. Chen et al. (2023b); Dou et al. (2024); Zadouri et al. (2023) combines MoE with LoRA, resulting in more effective training and alleviating data conflict issues.

The most relevant work to us is Lifelong-MoE (Chen et al. 2023a), which effectively expands the number of experts during lifelong learning and introduces a regularization to avoid catastrophic forgetting. However, we employ a different freezing method and a two-stage training framework, significantly alleviating catastrophic forgetting and gaining a promising performance in expanded languages.

LLM for Multilingual

Post-pretraining on a massive multilingual corpus is an effective way to improve the multilingual abilities of LLMs. Alves et al. (2024) and Xu et al. (2024) highlight monolingual data's importance in post-pretraining. Notably, Xu et al. (2024) demonstrates that with fixed computational resources, allocating more to monolingual data rather than translation data better improves a model's translation performance, allowing large models to achieve translation abilities comparable to traditional supervised models NLLB (Costa-jussà et al. 2022). Blevins et al. (2024) have explored using the Branch Then Merge (BTM;Gururangan et al. (2023)), where separate models are trained independently for different languages and then merged, partially overcoming the challenges of the multilingual curse (Wu and Dredze 2020). Geng et al. (2024) employs the LoRA (Hu et al. 2021) architecture to help migrate a chat LLM to the target language while preserving its chat capabilities.

Conclusion

In this paper, we propose MoE-LPR, a scalable post-pretraining method that effectively expands languages and prevents catastrophic forgetting using the Mixture-of-Experts architecture. Expanding new languages often encounters severe catastrophic forgetting due to significant distribution changes, and the challenge lies in balancing old and new languages. Through two-stage training, MoE-LPR addresses this with efficient parameter assignment and balanced routing. The post-pretraining stage enables the model to have a strong enough learning ability and steadily enhances the capabilities of the expanded languages. The review stage brings a performance boost to the original languages without harming the performance in expanded languages. Our two-stage training achieves both expansion and prevention of forgetting effects well. Additionally, MoE-LPR shows better scalability and generalization than SOTA methods. Overall, MoE-LPR is an effective and scalable approach for expanding new languages during the post-pretraining stage.

Acknowledgements

We thank the anonymous reviewers for their insightful comments. Shujian Huang is the corresponding author. This work is supported by National Science Foundation of China (No.62376116, 62176120) and Nanjing University China Mobile Communications Group Co.,Ltd.Joint Institute.

References

Achiam, J.; Adler, S.; Agarwal, S.; Ahmad, L.; Akkaya, I.; Aleman, F. L.; Almeida, D.; Altenschmidt, J.; Altman, S.; Anadkat, S.; et al. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Alves, D. M.; Pombal, J.; Guerreiro, N. M.; Martins, P. H.; Alves, J.; Farajian, A.; Peters, B.; Rei, R.; Fernandes, P.; Agrawal, S.; et al. 2024. Tower: An open multilingual large language model for translation-related tasks. *arXiv preprint arXiv:2402.17733*.

Bai, J.; Bai, S.; Chu, Y.; Cui, Z.; Dang, K.; Deng, X.; Fan, Y.; Ge, W.; Han, Y.; Huang, F.; et al. 2023. Qwen technical report. *arXiv preprint arXiv:2309.16609*.

Bandarkar, L.; Liang, D.; Muller, B.; Artetxe, M.; Shukla, S. N.; Husa, D.; Goyal, N.; Krishnan, A.; Zettlemoyer, L.; and Khabsa, M. 2023. The Belebele Benchmark: a Parallel Reading Comprehension Dataset in 122 Language Variants. *arXiv preprint arXiv:2308.16884*.

Blevins, T.; Limisiewicz, T.; Gururangan, S.; Li, M.; Gonen, H.; Smith, N. A.; and Zettlemoyer, L. 2024. Breaking the Curse of Multilinguality with Cross-lingual Expert Language Models. *arXiv preprint arXiv:2401.10440*.

Chen, W.; Zhou, Y.; Du, N.; Huang, Y.; Laudon, J.; Chen, Z.; and Cui, C. 2023a. Lifelong language pretraining with distribution-specialized experts. In *International Conference on Machine Learning*, 5383–5395. PMLR.

Chen, Z.; Wang, Z.; Wang, Z.; Liu, H.; Yin, Z.; Liu, S.; Sheng, L.; Ouyang, W.; and Shao, J. 2023b. Octavius: Mitigating Task Interference in MLLMs via MoE. In *The Twelfth International Conference on Learning Representations*.

Clark, P.; Cowhey, I.; Etzioni, O.; Khot, T.; Sabharwal, A.; Schoenick, C.; and Tafjord, O. 2018. Think you have solved question answering? try arc, the ai2 reasoning challenge. *arXiv preprint arXiv:1803.05457*.

Costa-jussà, M. R.; Cross, J.; Çelebi, O.; Elbayad, M.; Heafield, K.; Heffernan, K.; Kalbassi, E.; Lam, J.; Licht, D.; Maillard, J.; et al. 2022. No language left behind: Scaling human-centered machine translation. *arXiv preprint arXiv:2207.04672*.

Csaki, Z.; Li, B.; Li, J.; Xu, Q.; Pawakapan, P.; Zhang, L.; Du, Y.; Zhao, H.; Hu, C.; and Thakker, U. 2024. Sambalingo: Teaching large language models new languages. *arXiv preprint arXiv:2404.05829*.

Dao, T.; Fu, D.; Ermon, S.; Rudra, A.; and Ré, C. 2022. Flashattention: Fast and memory-efficient exact attention with io-awareness. *Advances in Neural Information Processing Systems*, 35: 16344–16359.

DeepSeek-AI. 2024. DeepSeek-V2: A Strong, Economical, and Efficient Mixture-of-Experts Language Model. arXiv:2405.04434.

Dou, S.; Zhou, E.; Liu, Y.; Gao, S.; Shen, W.; Xiong, L.; Zhou, Y.; Wang, X.; Xi, Z.; Fan, X.; Pu, S.; Zhu, J.; Zheng, R.; Gui, T.; Zhang, Q.; and Huang, X. 2024. LoRAMoE: Alleviating World Knowledge Forgetting in Large Language Models via MoE-Style Plugin. In Ku, L.-W.; Martins, A.; and Srikumar, V., eds., *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 1932–1945. Bangkok, Thailand: Association for Computational Linguistics.

Du, N.; Huang, Y.; Dai, A. M.; Tong, S.; Lepikhin, D.; Xu, Y.; Krikun, M.; Zhou, Y.; Yu, A. W.; Firat, O.; et al. 2022. Glam: Efficient scaling of language models with mixture-of-experts. In *International Conference on Machine Learning*, 5547–5569. PMLR.

Dubey, A.; Jauhri, A.; Pandey, A.; Kadian, A.; Al-Dahle, A.; Letman, A.; Mathur, A.; Schelten, A.; Yang, A.; Fan, A.; et al. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

Fedus, W.; Zoph, B.; and Shazeer, N. 2022. Switch transformers: Scaling to trillion parameter models with simple and efficient sparsity. *Journal of Machine Learning Research*, 23(120): 1–39.

Gao, C.; Hu, H.; Hu, P.; Chen, J.; Li, J.; and Huang, S. 2024. Multilingual Pretraining and Instruction Tuning Improve Cross-Lingual Knowledge Alignment, But Only Shallowly. In Duh, K.; Gomez, H.; and Bethard, S., eds., *NAACL 2024*, 6101–6117.

Geng, X.; Zhu, M.; Li, J.; Lai, Z.; Zou, W.; She, S.; Guo, J.; Zhao, X.; Li, Y.; Li, Y.; et al. 2024. Why Not Transform Chat Large Language Models to Non-English? *arXiv preprint arXiv:2405.13923*.

Goyal, N.; Gao, C.; Chaudhary, V.; Chen, P.-J.; Wenzek, G.; Ju, D.; Krishnan, S.; Ranzato, M.; Guzmán, F.; and Fan, A. 2022. The flores-101 evaluation benchmark for low-resource and multilingual machine translation. *Transactions of the Association for Computational Linguistics*, 10: 522–538.

Gururangan, S.; Li, M.; Lewis, M.; Shi, W.; Althoff, T.; Smith, N. A.; and Zettlemoyer, L. 2023. Scaling expert language models with unsupervised domain discovery. *arXiv preprint arXiv:2303.14177*.

Hendrycks, D.; Burns, C.; Basart, S.; Zou, A.; Mazeika, M.; Song, D.; and Steinhardt, J. 2020. Measuring Massive Multitask Language Understanding. In *International Conference on Learning Representations*.

Hoffmann, J.; Borgeaud, S.; Mensch, A.; Buchatskaya, E.; Cai, T.; Rutherford, E.; de Las Casas, D.; Hendricks, L. A.; Welbl, J.; Clark, A.; et al. 2022. An empirical analysis of compute-optimal large language model training. *Advances in Neural Information Processing Systems*, 35: 30016–30030.

Hu, E. J.; Wallis, P.; Allen-Zhu, Z.; Li, Y.; Wang, S.; Wang, L.; Chen, W.; et al. 2021. LoRA: Low-Rank Adaptation of Large Language Models. In *International Conference on Learning Representations*.

Ibrahim, A.; Thérien, B.; Gupta, K.; Richter, M. L.; Anthony, Q. G.; Belilovsky, E.; Lesort, T.; and Rish, I. 2024.

Simple and Scalable Strategies to Continually Pre-train Large Language Models. *Transactions on Machine Learning Research*.

Jiang, A. Q.; Sablayrolles, A.; Roux, A.; Mensch, A.; Savary, B.; Bamford, C.; Chaplot, D. S.; de las Casas, D.; Hanna, E. B.; Bressand, F.; Lengyel, G.; Bour, G.; Lample, G.; Lavaud, L. R.; Saulnier, L.; Lachaux, M.-A.; Stock, P.; Subramanian, S.; Yang, S.; Antoniak, S.; Scao, T. L.; Gervet, T.; Lavril, T.; Wang, T.; Lacroix, T.; and Sayed, W. E. 2024. Mixtral of Experts. arXiv:2401.04088.

Kaplan, J.; McCandlish, S.; Henighan, T.; Brown, T. B.; Chess, B.; Child, R.; Gray, S.; Radford, A.; Wu, J.; and Amodei, D. 2020. Scaling laws for neural language models. *arXiv preprint arXiv:2001.08361*.

Komatsuzaki, A.; Puigcerver, J.; Lee-Thorp, J.; Ruiz, C. R.; Mustafa, B.; Ainslie, J.; Tay, Y.; Dehghani, M.; and Houlsby, N. 2022. Sparse Upcycling: Training Mixture-of-Experts from Dense Checkpoints. In *The Eleventh International Conference on Learning Representations*.

Kuulmets, H.-A.; Purason, T.; Luhtaru, A.; and Fishel, M. 2024. Teaching Llama a New Language Through Cross-Lingual Knowledge Transfer. In Duh, K.; Gomez, H.; and Bethard, S., eds., *Findings of NAACL 2024*.

Lai, V.; Nguyen, C.; Ngo, N.; Nguyen, T.; Dernoncourt, F.; Rossi, R.; and Nguyen, T. 2023. Okapi: Instruction-tuned Large Language Models in Multiple Languages with Reinforcement Learning from Human Feedback. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, 318–327.

Mixtral. 2024. mixtral 8*22b.

Nguyen, T.; Nguyen, C. V.; Lai, V. D.; Man, H.; Ngo, N. T.; Dernoncourt, F.; Rossi, R. A.; and Nguyen, T. H. 2024. CulturaX: A Cleaned, Enormous, and Multilingual Dataset for Large Language Models in 167 Languages. In Calzolari, N.; Kan, M.-Y.; Hoste, V.; Lenci, A.; Sakti, S.; and Xue, N., eds., *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, 4226–4237. Torino, Italia: ELRA and ICCL.

OpenAI. 2023. ChatGPT (Mar 23 version) [Large language model].

Pu, X.; Gao, M.; and Wan, X. 2023. Summarization is (almost) dead. *arXiv preprint arXiv:2309.09558*.

Rei, R.; De Souza, J. G.; Alves, D.; Zerva, C.; Farinha, A. C.; Glushkova, T.; Lavie, A.; Coheur, L.; and Martins, A. F. 2022. COMET-22: Unbabel-IST 2022 submission for the metrics shared task. In *Proceedings of the Seventh Conference on Machine Translation (WMT)*, 578–585.

Research, S. A. 2024. Snowflake Arctic: The Best LLM for Enterprise AI — Efficiently Intelligent, Truly Open.

Robinson, J.; and Wingate, D. 2023. Leveraging Large Language Models for Multiple Choice Question Answering. In *The Eleventh International Conference on Learning Representations*.

Soboleva, D.; Al-Khateeb, F.; Myers, R.; Steeves, J. R.; Hestness, J.; and Dey, N. 2023. SlimPajama: A 627B token cleaned and deduplicated version of RedPajama. https://www.cerebras.net/blog/slimpajama-a-627b-token-cleaned-and-deduplicated-version-of-redpajama.

Touvron, H.; Martin, L.; Stone, K.; Albert, P.; Almahairi, A.; Babaei, Y.; Bashlykov, N.; Batra, S.; Bhargava, P.; Bhosale, S.; et al. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

Wang, B.; Liu, Z.; Huang, X.; Jiao, F.; Ding, Y.; Aw, A. T.; and Chen, N. F. 2024. SeaEval for Multilingual Foundation Models: From Cross-Lingual Alignment to Cultural Reasoning. *NAACL*.

Wei, T.; Zhao, L.; Zhang, L.; Zhu, B.; Wang, L.; Yang, H.; Li, B.; Cheng, C.; Lü, W.; Hu, R.; Li, C.; Yang, L.; Luo, X.; Wu, X.; Liu, L.; Cheng, W.; Cheng, P.; Zhang, J.; Zhang, X.; Lin, L.; Wang, X.; Ma, Y.; Dong, C.; Sun, Y.; Chen, Y.; Peng, Y.; Liang, X.; Yan, S.; Fang, H.; and Zhou, Y. 2023. Skywork: A More Open Bilingual Foundation Model. arXiv:2310.19341.

Wu, C.; Gan, Y.; Ge, Y.; Lu, Z.; Wang, J.; Feng, Y.; Shan, Y.; and Luo, P. 2024. LLaMA Pro: Progressive LLaMA with Block Expansion. In Ku, L.-W.; Martins, A.; and Srikumar, V., eds., *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 6518–6537. Bangkok, Thailand: Association for Computational Linguistics.

Wu, S.; and Dredze, M. 2020. Are All Languages Created Equal in Multilingual BERT? In *Proceedings of the 5th Workshop on Representation Learning for NLP*, 120–130.

Xu, H.; Kim, Y. J.; Sharaf, A.; and Awadalla, H. H. 2024. A Paradigm Shift in Machine Translation: Boosting Translation Performance of Large Language Models. In *The Twelfth International Conference on Learning Representations*.

Yu, L.; Jiang, W.; Shi, H.; Yu, J.; Liu, Z.; Zhang, Y.; Kwok, J. T.; Li, Z.; Weller, A.; and Liu, W. 2023. Metamath: Bootstrap your own mathematical questions for large language models. *arXiv preprint arXiv:2309.12284*.

Zadouri, T.; Üstün, A.; Ahmadian, A.; Ermis, B.; Locatelli, A.; and Hooker, S. 2023. Pushing Mixture of Experts to the Limit: Extremely Parameter Efficient MoE for Instruction Tuning. In *The Twelfth International Conference on Learning Representations*.

Zellers, R.; Holtzman, A.; Bisk, Y.; Farhadi, A.; and Choi, Y. 2019. HellaSwag: Can a Machine Really Finish Your Sentence? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*.

Zoph, B.; Bello, I.; Kumar, S.; Du, N.; Huang, Y.; Dean, J.; Shazeer, N.; and Fedus, W. 2022a. St-moe: Designing stable and transferable sparse expert models. *arXiv preprint arXiv:2202.08906*.

Zoph, B.; Raffel, C.; Schuurmans, D.; Yogatama, D.; Zhou, D.; Metzler, D.; Chi, E. H.; Wei, J.; Dean, J.; Fedus, L. B.; Bosma, M. P.; Vinyals, O.; Liang, P.; Borgeaud, S.; Hashimoto, T. B.; and Tay, Y. 2022b. Emergent abilities of large language models. *TMLR*.