Group-Merger: A LoRA-based Framework for Multilingual Continual Learning

Weijian Yi[1,2], **Hongliang Li**[1,2], **Jinan Xu**[1,2] *
[1] Key Laboratory of Big Data & Artificial Intelligence in Transportation
(Beijing Jiaotong University), Ministry of Education
[2] School of Computer Science and Technology,
Beijing Jiaotong University, Beijing 100044, China
{yiweijian, hongliangli, jaxu}@bjtu.edu.cn

Abstract

Multilingual continual learning (MCL) is crucial for enabling language models to adapt across diverse linguistic environments while retaining knowledge over time. Existing parameter isolation methods allocate language-specific modules but fail to leverage cross-lingual transfer, leading to inefficient parameter growth and poor generalization. Model merging based approaches suffer from severe performance degradation as the number of language-specific tasks increases, due to interference between linguistic and task-specific knowledge. To address these challenges, we propose **Group-Merger**, a framework that employs group-wise merging to balance parameter efficiency and continual learning performance. Our framework mitigates catastrophic forgetting across languages while enabling knowledge transfer. Extensive experiments on multilingual evaluation benchmarks demonstrate superior performance compared to existing methods.

1 Introduction

Recent advances in large language models (LLMs) have achieved remarkable performance across multilingual NLP tasks by leveraging large-scale cross-lingual data (Conneau and Lample, 2019; Xue et al., 2021; Dubey et al., 2024; Yang et al., 2025). However, real-world language environments are dynamic and multilingual, requiring LLMs to continually adapt to new languages and tasks without forgetting previously acquired knowledge (Shi et al., 2024; Wu et al., 2024). Therefore, it is vital to empower LLMs with the capability of continual learning in multilingual scenarios (Wu et al., 2022).

A direct solution is joint training on mixed multilingual data (Rolnick et al., 2019), but this is impractical due to the prohibitive cost of repeatedly processing large-scale LLMs pretraining data (Hadi et al., 2023). Instead, multilingual continual learning (MCL) demands rehearsal-free methods that enable LLMs to acquire new languages or tasks while preserving prior knowledge without storing past data (Wu et al., 2022). Parameter isolation methods such as Low-Rank Adaptation (LoRA) (Hu et al., 2021) address catastrophic forgetting by freezing pre-trained weights and allocating task-specific modules (Figure 1). However, these methods suffer from two critical limitations in multilingual settings: (1) **Parameter explosion.** This approach requires assigning specific parameters to each individual task and language, which results in a rapid increase of parameters as the number of tasks and languages scale. (2) **Module independence.** Since the module for each new task is independently adapted, conventional approaches primarily focus on knowledge transfer between the original model and individual task modules, while overlooking the impact of parameter sharing across different new tasks (Wang et al., 2023b). In addition, the multilingual scenario struggles with cross-lingual interference, where knowledge ac-

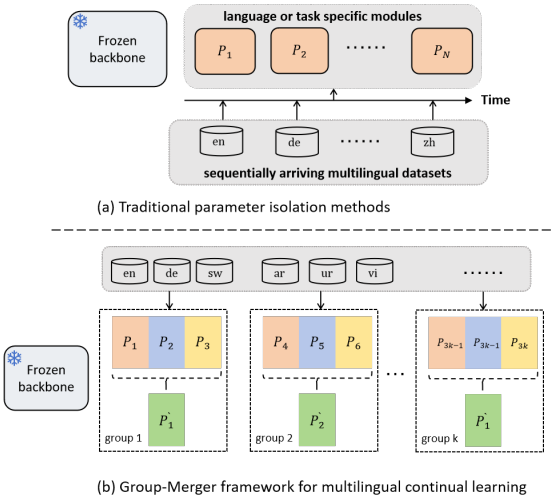

Figure 1: The difference between traditional parameter isolation methods and Group-Merger.

*Corresponding author

Proceedings of the 1st Workshop on Multilinguality in the Era of Large Language Models (MeLLM 2026), pages 308–316

quisition in one language may negatively impact performance in another.

This naturally raises the question of *"Can we achieve a better trade-off between parameter expansion and model performance?"*. Existing approaches fail to leverage previously learned modules in a rehearsal-free manner (Zhang et al., 2022). To address this issue, we propose **Group-Merger**, a merging-based MCL framework that can merge multiple language-specific modules into a single one with minimal cross-lingual interference. In particular, our framework dynamically groups and merges task-specific LoRA modules during the continual training process, enabling more efficient parameter utilization. Moreover, to alleviate the issue of task interference during merging, we introduce *group-level task-aware orthogonal training*, aiming to keep the current LoRA orthogonal with dissimilar task modules while alleviating orthogonality constraints for similar tasks, thereby implicitly facilitating knowledge transfer.

To sum up, our contributions are as follows:

- We propose a novel parameter-efficient MCL framework, **Group-Merger**, which achieves a better trade-off in parameter expansion and continual learning performance across diverse MCL benchmarks and model architectures.
- We propose a group-level task-aware orthogonal training approach, which addresses the limitation of insufficient model plasticity in vanilla LoRA as the number of learned tasks increases, enhancing knowledge transfer abilities.
- Extensive experimental results on multilingual benchmarks demonstrate that **Group-Merger** effectively mitigates catastrophic forgetting across languages while significantly improving cross-lingual knowledge transfer capabilities.

2 Related Works

2.1 Continual Learning

Continual learning focuses on overcoming the problem of catastrophic forgetting in streaming data scenarios. Existing continual learning methods can be categorized into the following three types:

Regularization-based. These approaches incorporate regularization terms into the loss function to prevent large updates to important parameters from previous tasks. Representative methods include EWC (Kirkpatrick et al., 2017) and LwF (Li and Hoiem, 2017).

Replay-based. These methods aim to combine knowledge from previous tasks with the current task to mitigate catastrophic forgetting. Old knowledge can be preserved by saving historical training samples (Chaudhry et al., 2019), generating pseudo-samples (Sun et al., 2019; Zhao et al., 2024), or using knowledge distillation (Li and Hoiem, 2017; Rebuffi et al., 2017).

Architecture-based. Since adjusting parameters from previous tasks may lead to catastrophic forgetting, these methods propose using different parameters for each task. The parameters for new tasks can either come from parts of the original network that do not overlap with the parameters of previous tasks or from newly introduced parameter modules (Wang et al., 2023b; Razdaibiedina et al., 2023; Wang et al., 2023a; Zhao et al., 2024).

2.2 Model Merging

Model merging seeks to effectively combine multiple fine-tuned models or Parameter-Efficient Fine-Tuning (PEFT) modules into a unified one that preserves the individual strengths of each (Yang et al., 2024). Therefore, merging multiple PEFT modules can be viewed as a strategy to mitigate catastrophic forgetting. Chitale et al. (2023) leverage task arithmetic (Ilharco et al., 2022) to merge all task-specific LoRA modules and fine-tune with small memory data. DAM (Cheng et al., 2024) introduces a dynamic merging method during inference for VQA domain incremental learning. O-LoRA (Wang et al., 2023a) constrains all tasks learned in distinct vector subspaces, ensuring that they remain orthogonal to each other to minimize interference when merging all task-specific modules. Existing methods either overlook the interference caused by merging or fail to account for knowledge transfer in similar tasks. In this work, we simultaneously address both of these limitations.

3 Preliminaries

3.1 Multilingual Continual Learning Formalization

Assume a sequence of T tasks arriving over time, where each task t is associated with an independent dataset $D_t = \{(x_i^t, y_i^t)\}_{i=1}^{N_t}$. The model is required

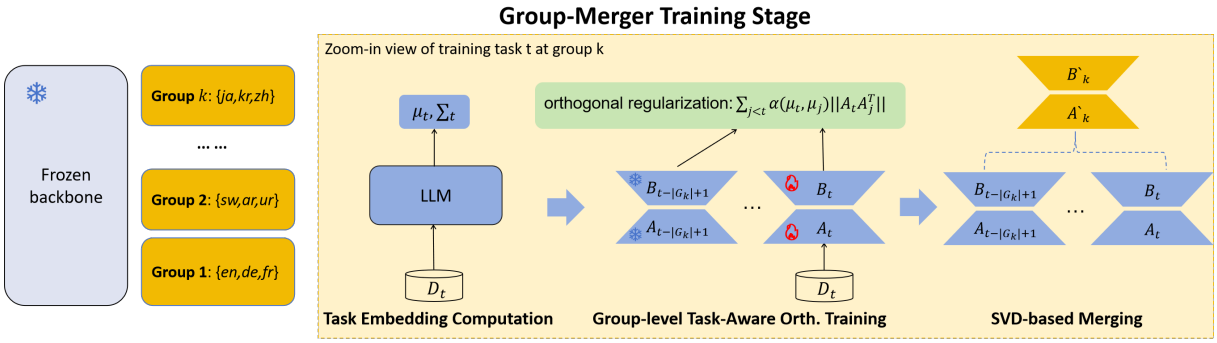

Figure 2: An overview of **Group-Merger** framework. The illustration demonstrates the process of training task 1 to task t on sequentially arriving multilingual datasets$\{D_1, D_2, ..., D_t\}$: We first partition t tasks into k groups, where each group has a single LoRA module that can handle n tasks at most. For the k-th group, we progressively train LoRA module with *group-level task-aware orthogonal training* to learn the task t in group G_k. When the number of LoRA modules reaches the group's capacity limit, we employ svd-based merging to merge all LoRA modules in group G_k into a single one, avoiding parameter expansion.

to learn these tasks sequentially, and at time step t, it can only access the current dataset D_t, with no access to data from previous tasks. During the inference phase, test samples are randomly drawn from the $D_{1:T}$ with their corresponding task identities (i.e., the task-id is available).

3.2 LoRA Module

LoRA (Hu et al., 2021) is a representative parameter-efficient fine-tuning technique that introduces two low-rank matrices into PLMs to enable efficient fine-tuning. In continual learning scenarios, different LoRA parameters can be assigned to each task, achieving continual learning through parameter isolation. When the model needs to learn a new task t, LoRA introduces two low-rank matrices $A_t \in \mathbb{R}^{d\times r}$ and $B_t \in \mathbb{R}^{r\times d}$, to the projection matrix $W \in \mathbb{R}^{d\times d}$ in the self-attention layer, transforming the original model's forward pass into $h = x_t W + x_t A_t B_t$. During training, the original parameters of the pre-trained model remain fixed, and only the introduced matrices A_t and B_t are adjusted. After training, the LoRA parameters corresponding to each task are saved and used during the inference phase to combine with the original PLM.

4 Methods

4.1 Group-Merger Framework

To address the issue of rapid parameter expansion in traditional PEFT-based continual learning approaches, we propose the **Group-Merger** framework, as illustrated in Figure 2 and Algorithm 1. Group-Merger mitigates parameter expansion through model merging while effectively preventing catastrophic forgetting caused by parameter conflicts during merging via orthogonal training. Assume that task-specific datasets arrive in sequence, denoted as $\{D_1, D_2, ..., D_T\}$. We partition T tasks into k groups $\{G_1, G_2, ..., G_k\}$, where each group G_i can handle n tasks by adding LoRA module incrementally. When the number of tasks in G_i reaches its capacity limit n, we perform a model merging operation to reduce parameter storage overhead.

4.2 Group-Level Task-Aware Orthogonal Training

Simply merging all modules would lead to parameter interference, consequently resulting in performance degradation. Existing works in model merging have demonstrated that parameter orthogonality significantly mitigate the interference (Ilharco et al., 2022). Specifically, if $\Delta W_i = A_i B_i$ and $\Delta W_j = A_j B_j$ are orthogonal, they satisfy the following condition:

$$\begin{aligned} \langle \Delta W_i, \Delta W_j \rangle &= \mathrm{Tr}[\Delta W_i^\top \Delta W_j] \\ &= \mathrm{Tr}[B_i^\top A_i^\top A_j B_j] \\ &= 0 \end{aligned} \tag{1}$$

We can apply constraints to the matrix product $A_i^\top A_j \approx \mathbf{0}$, then $\langle \Delta W_i, \Delta W_j \rangle \approx 0$ and the interference can be effectively minimized when merged. O-LoRA (Wang et al., 2023a) simply keeps the current A_t orthogonal with all previous tasks $A_i (i < t)$, which would hinder the learning of current task when the number of tasks is

Algorithm 1 Group-Merger Training Procedure

Input: Task-specific datasets $\{D_i\}_{i=1}^{T}$, model f, Pre-trained Language model f, maximum group size n, upper and lower bounds of orthogonal intensity λ_2, λ_1
Output: LoRA parameters for $\lceil \frac{T}{n} \rceil$ groups: $\{(A'_i, B'_i)\}_{i=1}^{\lceil \frac{T}{n} \rceil}$
for $t = 1$ to T **do**
  Compute group index k for current task t $k = \lceil \frac{t}{n} \rceil$
  Compute μ_t using Eq.(2)
  All groups parameter set $S = \{\}$
  for mini-batch $(x_t, y_t) \in D_t$ **do**
    Forward pass using Eq.(7)
    Compute task loss and orthogonal loss using Eq.(8)
  end for
  Add task t to G_k
  if $|G_k| == n$ **then**
    Merge all LoRA modules in G_k using Eq.(9)
    Perform SVD to obtain (A'_k, B'_k) by Eq.(10), Eq.(11)
    Add (A'_k, B'_k) into S
  end if
  Return S
end for

large. To address this issue, we propose *Group-level Orthogonal Training*. For task t in group $G_k = \{t - |G_k| + 1, t - |G_k| + 2, ..., t\}$, we only apply the orthogonal constraints in the same group, which improves the model's plasticity as the number of learning tasks increases. Additionally, orthogonality can hinder the knowledge transfer when a new task t is similar to some old tasks (Lin et al., 2022). We employ a pre-trained sentence embedding model f, to generate task-specific embeddings $\mu_t \in \mathbb{R}^d$ for each dataset D_t using Eq. (2).

$$\mu_i = \frac{1}{|D_i|} \sum_{x \in D_i} f(x) \tag{2}$$

The embedding of the current task μ_t is utilized to compute its similarity with previously learned tasks in group k, i.e. $i \in G_k \setminus \{t\}$. Suppose the G_k contains $n-1$ previous tasks, we compute the cosine similarity between the new task t and each existing task in G_k, resulting in $n-1$ similarity scores denoted as $s_{t,1}, s_{t,2}, ..., s_{t,n-1}$. These similarity scores are then normalized to the interval $[\lambda_1, \lambda_2]$ using Min-Max Normalization to obtain the final orthogonal constraint coefficients, where λ_1 and λ_2 represent predefined hyperparameters. The computational procedure can be mathematically formulated using Eqs. (3), (4) and (5).

$$min_{G_k} = \min_{(i,j) \in G_k} \left(\frac{1}{\cos(\mu_i, \mu_j)}\right) \tag{3}$$

$$max_{G_k} = \max_{(i,j) \in G_k} \left(\frac{1}{\cos(\mu_i, \mu_j)}\right) \tag{4}$$

$$\alpha(\mu_i, \mu_j, G_k) = \lambda_1 + \frac{\frac{1}{\cos(\mu_i, \mu_j)} - min_{G_k}}{max_{G_k} - min_{G_k}} (\lambda_2 - \lambda_1) \tag{5}$$

Finally, our orthogonal regularization loss function is mathematically expressed in Eq. (6).

$$L_{orth}(i, j, G_k) = \alpha(\mu_i, \mu_j, G_k) ||A_i A_j^T|| \tag{6}$$

4.3 Previous Modules Incorporation

For better knowledge transfer across groups, we incorporate the LoRA modules from previous groups $\{G_j\}_{j=1}^{k-1}$ into the forward propagation. The forward propagation can be formally expressed as follows:

$$f(x; \theta) = xW + \sum_{i \in G_k} x A_i B_i + \sum_{i \in \{G_j\}_{j=1}^{k-1}} x A'_i B'_i \tag{7}$$

The final orthogonal regularization term is described as Eq.(6) and the training objective is formulated as Eq.(8). Among all parameters θ, only the current task-specific parameters A_t, B_t are trainable.

$$\sum_{x,y \in D_t} \log p_\theta(y|x) + \sum_{i \in G_k \setminus \{t\}} \mathcal{L}_{\text{orth}}(i, t, G_k) \tag{8}$$

After learning all task, we perform a SVD-based merging to reduce parameter storage overhead

$$\Delta W^k_{\text{merge}} = \sum_{i \in G_k} A_i B_i \tag{9}$$

Specifically, we employ Singular Value Decomposition (SVD) on the $\Delta W^k_{\text{merge}}$. We first apply SVD to the $\Delta W^k_{\text{merge}}$, i.e.,$\Delta W^k_{\text{merge}} = U_k \Lambda_k V_k^\top$, where $U_k = [\boldsymbol{u}_{k,1}, \boldsymbol{u}_{k,2}, ..., \boldsymbol{u}_{k,d}]$, $V_k = [\boldsymbol{v}_{k,1}, \boldsymbol{v}_{k,2}, ..., \boldsymbol{v}_{k,d}]$ and $\Lambda_k \in \mathbb{R}^{d \times d}$ is a diagonal matrix with non-negative singular values

$\{\sigma_{k,1}, \sigma_{k,2}, ..., \sigma_{k,d}\}$. We use r-rank matrix approximation to derive the final A'_k and B'_k matrices.

$$A'_k = [\boldsymbol{u}_{k,1}, \boldsymbol{u}_{k,2}, ..., \boldsymbol{u}_{k,r}] \tag{10}$$

$$B'_k = \text{diag}(\{\sigma_{k,1}, \sigma_{k,2}, ..., \sigma_{k,r}\}) [\boldsymbol{v}_{k,1}, \boldsymbol{v}_{k,2}, ..., \boldsymbol{v}_{k,r}]^\top \tag{11}$$

5 Experiments

5.1 Datasets

We evaluate our framework on two widely-used multilingual benchmark datasets: Cross-lingual Natural Language Inference (XNLI) (Conneau et al., 2018) and Multilingual Question Answering (MLQA) (Lewis et al., 2019). XNLI is a widely used benchmark for evaluating the cross-lingual transfer capabilities of natural language understanding. MLQA is designed to evaluate the performance of question answering (QA) models across multiple languages. For the XNLI benchmark, we use 8 different language datasets, which include English, German, Greek, Arabic, Swahili, Urdu, Vietnamese and Chinese. For the MLQA benchmark, we use Arabic, German, Spanish, Hindi, Vietnamese, English. We randomly sampled 10,000 instances per language for training. The continual learning task orders are listed in Table 1.

Benchmark	Task Sequence
XNLI	en → de → el → ar → sw → ur → vi → zh
MLQA	ar → de → es → hi → vi → en

Table 1: Task sequence orders used in the continual learning experiments.

5.2 Baselines

We compare our method with various representative baselines in continual learning. The baselines are as follows:

- SeqFT: refers to the method of fine-tuning all parameters sequentially.
- SeqLoRA: finetunes a single LoRA module sequentially.
- Replay: combines real samples to mitigate catastrophic forgetting when learning a new task.
- EWC (Kirkpatrick et al., 2017): adds regularization to the loss function to prevent significant changes to important parameters of old tasks when learning new tasks.
- O-LoRA (Wang et al., 2023a): trains LoRA module for each task and keeps them orthogonal.
- TA-LoRA (Chitale et al., 2023): trains a lora module for each task and use Task Arithmetic to merge them.
- PerTask-LoRA: trains a lora for each task and use the oracle one during inference.
- MTL: indicates that training the model on all task datasets simultaneously typically serves as the performance upper bound for continual learning.

5.3 Implementation Details

All experiments in this study are implemented based on the HuggingFace Transformers library (Wolf et al., 2020). We adopt the Llama3 (3B and 8B) (Dubey et al., 2024) and Qwen3 (1.7B and 8B) (Team, 2025) as the backbone architecture across all experiments. For model optimization, we employ the AdamW optimizer($\beta_1 = 0.9, \beta_2 = 0.999$) (Loshchilov, 2017) with mixed-precision training using bfloat16 format. The training configuration includes a learning rate of 1e-3 with cosine learning rate scheduler over 5 epochs, a batch size of 64, and a maximum sequence length of 512 tokens. For the LoRA configuration, we set the rank to 32 and alpha to 64. The orthogonal regularization parameters are set to $\lambda_1 = 0.01, \lambda_2 = 0.5$ for the XNLI benchmark, while $\lambda_1 = 0.01, \lambda_2 = 0.1$ are used for the MLQA benchmark. All models are trained on NVIDIA A100 GPUs.

5.4 Metrics

Average Performance (AP). AP measures the model's overall performance after training all tasks. $AP = \sum_{i=1}^{T} A_{i,T}$, where $A_{i,j}$ denotes the model performance on task j after learning i tasks. We evaluate model performance using exact match accuracy on the XNLI benchmark and F1 score on the MLQA benchmark (Rajpurkar et al., 2016).

Forward Transfer (FT). FT quantifies the extent to which learning a previous task improves performance on a new task, reflecting the model's ability to leverage prior knowledge. $\text{FT} = \frac{1}{T} \sum_{i=1}^{T} A_{i,i} - A_{0,i}$.

Forget Rate (FR). FR measures the degree of catastrophic forgetting in continual learning

Method	XNLI					MLQA				
	AP↑	FR↓	FT↑	#TP↓	#EM↓	AP↑	FR↓	FT↑	#TP↓	#EM↓
SeqFT	37.61	42.8	0.10	3.21B	0	44.58	21.45	0.15	3.21B	0
Replay	58.73	18.65	0.12	3.21B	0	57.02	6.07	-0.22	3.21B	0
EWC	57.28	18.97	-1.18	3.21B	0	55.9	7.41	-0.14	3.21B	0
TA-LoRA	71.18	2.56	-2.21	12.85M	1	59.64	2.42	-0.64	12.85M	1
O-LoRA	72.37	0.78	-2.36	12.85M	1	62.69	0.66	-0.54	12.85M	1
PerTask-LoRA	72.48	0.0	-2.57	12.85M	8	61.90	0.0	-0.81	12.85M	6
GM-LoRA (Ours)	**74.23**	1.51	**0.27**	12.85M	2	**62.71**	0.47	**0.44**	12.85M	2
MTL(*upperbound*)	78.59	-	-	3.21B	-	64.28	-	-	3.21B	-

Table 2: The overall results on XNLI and MLQA benchmark with Llama-3B model. Performance of continual learning (AP), forget rate (FR) and forward transfer (FT) are reported after training on the last task.

scenarios. $F = \frac{1}{T-1}\sum_{t=1}^{T-1} \max_{i=1}^{T-1}(a_{i,t}) - a_{T,t}$.

Trainable Parameter (#TP). Trainable Parameter is used to measure the number of trainable parameters when learning a new task.

Extra Modules (#EM). EM is used to measure the number of task-specific modules that need to be stored at inference time. In our framework, the relationship between group size and extra modules can be mathematically expressed as $\#\mathrm{EM} = \lceil \frac{T}{n} \rceil$, where n represents the group size. For a clearer comparison of the number of additional modules, we exclude the scenario where LoRA is merged into the original parameters.

6 Experiments

6.1 Overall Results

Group-Merger framework achieves best performance across both XNLI and MLQA benchmark. Table 2 presents a comprehensive comparison between our method and existing state-of-the-art continual learning methods on XNLI and MLQA benchmark respectively. The experimental results show that Group-Merger achieves improvements of 1.75% and 0.02% in average performance on XNLI and MLQA datasets, respectively. Compared to other continual learning methods based on LoRA, our method demonstrates superior performance in terms of both AP and FT, which demonstrates that our method can effectively leverage historical information to facilitate the learning of new tasks. Moreover, our approach eliminates the need for replaying any historical data, thereby significantly enhancing the efficiency of continual learning.

Group-Merger is more efficient in terms of *trainable parameters* and *extra modules*. The LoRA-based training paradigm significantly reduces the trainable parameters while adapting to a new task. The Group-Merger further reduces the parameter storage requirements during the inference. O-LoRA does not require storing extra parameters during inference, but a critical limitation is that the merging process disables its ability for further continual learning.

6.2 Ablation Studies

Effects of Orthogonal Training. Orthogonal training plays a crucial role in our continual learning framework. It enforces the parameter subspaces learned by different tasks to remain orthogonal by applying orthogonal constraints in the loss function, thereby effectively mitigating parameter interference during model merging. Furthermore, our design takes into account the linguistic similarities across different languages, which allows for adaptive adjustment of the orthogonal coefficients to facilitate knowledge transfer between similar tasks. The upper part of Table 3 presents the results with and without orthogonal training, demonstrating its significant impact on model performance. Compared to the "ORTH" baseline, the Task-Aware ORTH (TA-ORTH) demonstrates superior performance, which not only preserves task-specific knowledge but also enables more effective cross-task knowledge transfer.

Effects of Incorporating Previous Modules. Our framework incorporates a novel forward propagation that leverages previous modules when learning a new task, thereby facilitating

ORTH	TA-ORTH	IPM	#EM	XNLI AP↑	XNLI FR↓	XNLI FT↑	MLQA AP↑	MLQA FR↓	MLQA FT↑
Effects of Orthogonal Training									
×	✓	✓	2	74.23	1.51	0.27	62.71	0.47	0.44
✓	×	✓	2	73.66	1.4	-1.40	61.40	0.61	-0.39
×	×	✓	2	70.68	3.89	-0.97	58.29	5.36	0.45
Effects of Incorporating Previous Modules									
×	✓	✓	2	74.23	1.51	0.27	62.71	0.47	0.44
×	✓	×	2	63.68	11.82	-0.34	58.21	4.86	-0.05
Effects of Different Group Size									
×	✓	✓	1	72.25	3.2	0.14	62.45	0.72	0.12
×	✓	✓	2	74.23	1.51	0.27	62.71	0.47	0.44
×	✓	✓	3	76.52	0.64	1.93	63.03	0.29	0.26

Table 3: The ablation study results on XNLI and MLQA benchmark with Llama-3B model. "ORTH", "TA-ORTH", "IPM" represent the orthogonal regularization, task-aware orthogonal regularization and incorporating previous modules, which are introduced in Group-Merger framework.

effective knowledge transfer from old to new tasks. As demonstrated in the middle part of Table 3, we compare the performance with and without incorporating previous modules. The results demonstrate that incorporating previous modules when learning a new task significantly improves the FT metric.

Effects of Different Group Size. The relationship between the number of extra modules#EM and group size n follows the equation $\#\text{EM} = \lceil \frac{T}{n} \rceil$, where T denotes the total number of tasks. This formulation provides a flexible trade-off between storage overhead and model performance across diverse continual learning scenarios. As shown in the lower part of Table 3, smaller group size yields superior performance, while larger group size trades marginal performance degradation for reduced storage overhead. This flexible design enables effective adaptation to diverse continual learning scenarios.

6.3 Comparison of Merging Methods

Merging all parameters in the same group is a crucial step for reducing extra parameters storage. We include the following merging methods:

- Weight Average (Wortsman et al., 2022): a baseline method that combines models by taking their mean.
- Task Arithmetic (Ilharco et al., 2022): uses the same weight λ to merge models.
- Ties-Merging (Yadav et al., 2023): mitigates conflicts during model merging by employing parameter pruning and directional alignment.
- Ours: computes orthogonality scores for each module based on task similarity during training.

Method↓	XNLI			MLQA		
Group Size→	$n=2$	$n=3$	$n=6$	$n=2$	$n=3$	$n=8$
Weight Average	63.65	56.83	57.72	62.94	61.09	60.27
Task Arithmetic	51.87	57.01	64.18	59.09	61.39	60.65
Ties-Merging	68.59	68.79	61.57	63.05	62.37	62.16
Ours	**77.05**	**74.70**	**73.33**	**63.28**	**62.71**	**62.54**

Table 4: The results of different model merging approaches on average performance across varying group size n.

The results in Table 4 demonstrate that optimizing orthogonality during training yields superior merging performance compared to those post-training optimization merging methods.

6.4 Generalization on Different Model

Table 5 presents the performance of models with varying sizes and architectures under the Group-Merger framework. Our framework demonstrates consistent effectiveness across models of varying sizes and architectures. Moreover, we observe that

Model	XNLI	MLQA
Qwen3-1.7B	73.66	60.31
Llama3-3B	74.23	62.71
Llama3-8B	79.35	64.28
Qwen3-8B	79.43	63.41

Table 5: The results of the XNLI and MLQA benchmarks with different architectures and parameters. The #EM is set to 2.

the average performance improves as the model parameters increase, suggesting better scalability of our framework.

7 Conclusion

In this paper, we introduce a novel framework **Group-Merger** that addresses catastrophic forgetting in incrementally learning new languages by model merging. Extensive experiments demonstrate that our method outperforms most existing methods while requiring less storage for extra language-specific modules, highlighting the effectiveness of model merging in multilingual continual learning.

Limitations

Although our framework demonstrates outstanding performance in the experiments, there are still several limitations. Firstly, Group Merger relies on the design of LoRA to achieve efficient parameter orthogonality and model merging, which is currently not applicable to other PEFT methods. Future work could explore adapting Group Merger to alternative PEFT methods. Secondly, our experiments include Natural Language Understanding and Machine Reading Comprehension, but focus only on continual learning within the same task type. Future work should explore more challenging scenarios involving continual learning across diverse tasks.

Acknowledgements

The research work described in this paper has been supported by the National Key R&D Program of China (2020AAA0108001), and the National Nature Science Foundation of China (No. 62376019, 62476023, 61976015, 61976016, 61876198 and 61370130). The authors would like to thank the anonymous reviewers for their valuable comments and suggestions to improve this paper.

References

Arslan Chaudhry, Marcus Rohrbach, Mohamed Elhoseiny, Thalaiyasingam Ajanthan, Puneet K Dokania, Philip HS Torr, and Marcaurelio Ranzato. 2019. On tiny episodic memories in continual learning. arxiv. *Learning*, 6(7).

Feng Cheng, Ziyang Wang, Yi-Lin Sung, Yan-Bo Lin, Mohit Bansal, and Gedas Bertasius. 2024. Dam: Dynamic adapter merging for continual video qa learning. *arXiv preprint arXiv:2403.08755*.

Rajas Chitale, Ankit Vaidya, Aditya Kane, and Archana Ghotkar. 2023. Task arithmetic with lora for continual learning. *arXiv preprint arXiv:2311.02428*.

Alexis Conneau and Guillaume Lample. 2019. Cross-lingual language model pretraining. *Advances in neural information processing systems*, 32.

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

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. 2024. The llama 3 herd of models. *arXiv e-prints*, pages arXiv–2407.

Muhammad Usman Hadi, Rizwan Qureshi, Abbas Shah, Muhammad Irfan, Anas Zafar, Muhammad Bilal Shaikh, Naveed Akhtar, Jia Wu, Seyedali Mirjalili, et al. 2023. A survey on large language models: Applications, challenges, limitations, and practical usage. *Authorea Preprints*.

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*.

Gabriel Ilharco, Marco Tulio Ribeiro, Mitchell Wortsman, Suchin Gururangan, Ludwig Schmidt, Hannaneh Hajishirzi, and Ali Farhadi. 2022. Editing models with task arithmetic. *arXiv preprint arXiv:2212.04089*.

James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, et al. 2017. Overcoming catastrophic forgetting in neural networks. *Proceedings of the national academy of sciences*, 114(13):3521–3526.

Patrick Lewis, Barlas Oğuz, Ruty Rinott, Sebastian Riedel, and Holger Schwenk. 2019. Mlqa: Evaluating cross-lingual extractive question answering. *arXiv preprint arXiv:1910.07475*.

Zhizhong Li and Derek Hoiem. 2017. Learning without forgetting. *IEEE transactions on pattern analysis and machine intelligence*, 40(12):2935–2947.

Sen Lin, Li Yang, Deliang Fan, and Junshan Zhang. 2022. Trgp: Trust region gradient projection for continual learning. *arXiv preprint arXiv:2202.02931*.

I Loshchilov. 2017. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. Squad: 100,000+ questions for machine comprehension of text. *arXiv preprint arXiv:1606.05250*.

Anastasia Razdaibiedina, Yuning Mao, Rui Hou, Madian Khabsa, Mike Lewis, and Amjad Almahairi. 2023. Progressive prompts: Continual learning for language models. *arXiv preprint arXiv:2301.12314*.

Sylvestre-Alvise Rebuffi, Alexander Kolesnikov, Georg Sperl, and Christoph H Lampert. 2017. icarl: Incremental classifier and representation learning. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 2001–2010.

David Rolnick, Arun Ahuja, Jonathan Schwarz, Timothy Lillicrap, and Gregory Wayne. 2019. Experience replay for continual learning. *Advances in neural information processing systems*, 32.

Haizhou Shi, Zihao Xu, Hengyi Wang, Weiyi Qin, Wenyuan Wang, Yibin Wang, Zifeng Wang, Sayna Ebrahimi, and Hao Wang. 2024. Continual learning of large language models: A comprehensive survey. *ACM Computing Surveys*.

Fan-Keng Sun, Cheng-Hao Ho, and Hung-Yi Lee. 2019. Lamol: Language modeling for lifelong language learning. *arXiv preprint arXiv:1909.03329*.

Qwen Team. 2025. Qwen3 technical report. *Preprint*, arXiv:2505.09388.

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*.

Zhicheng Wang, Yufang Liu, Tao Ji, Xiaoling Wang, Yuanbin Wu, Congcong Jiang, Ye Chao, Zhencong Han, Ling Wang, Xu Shao, et al. 2023b. Rehearsal-free continual language learning via efficient parameter isolation. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 10933–10946.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Mitchell Wortsman, Gabriel Ilharco, Samir Ya Gadre, Rebecca Roelofs, Raphael Gontijo-Lopes, Ari S Morcos, Hongseok Namkoong, Ali Farhadi, Yair Carmon, Simon Kornblith, et al. 2022. Model soups: averaging weights of multiple fine-tuned models improves accuracy without increasing inference time. In *International conference on machine learning*, pages 23965–23998. PMLR.

Tongtong Wu, Massimo Caccia, Zhuang Li, Yuan Fang Li, Guilin Qi, and Gholamreza Haffari. 2022. Pretrained language model in continual learning: A comparative study. In *International Conference on Learning Representations 2022*. OpenReview.

Tongtong Wu, Linhao Luo, Yuan-Fang Li, Shirui Pan, Thuy-Trang Vu, and Gholamreza Haffari. 2024. Continual learning for large language models: A survey. *arXiv preprint arXiv:2402.01364*.

Linting Xue, Noah Constant, Adam Roberts, Mihir Kale, Rami Al-Rfou, Aditya Siddhant, Aditya Barua, and Colin Raffel. 2021. mt5: A massively multilingual pre-trained text-to-text transformer. In *Proceedings of the 2021 conference of the North American chapter of the association for computational linguistics: Human language technologies*, pages 483–498.

Prateek Yadav, Derek Tam, Leshem Choshen, Colin A Raffel, and Mohit Bansal. 2023. Ties-merging: Resolving interference when merging models. *Advances in Neural Information Processing Systems*, 36:7093–7115.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, et al. 2025. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*.

Enneng Yang, Li Shen, Guibing Guo, Xingwei Wang, Xiaochun Cao, Jie Zhang, and Dacheng Tao. 2024. Model merging in llms, mllms, and beyond: Methods, theories, applications and opportunities. *arXiv preprint arXiv:2408.07666*.

Yanzhe Zhang, Xuezhi Wang, and Diyi Yang. 2022. Continual sequence generation with adaptive compositional modules. *arXiv preprint arXiv:2203.10652*.

Weixiang Zhao, Shilong Wang, Yulin Hu, Yanyan Zhao, Bing Qin, Xuanyu Zhang, Qing Yang, Dongliang Xu, and Wanxiang Che. 2024. Sapt: A shared attention framework for parameter-efficient continual learning of large language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 11641–11661.