# Extracting and Combining Abilities For Building Multi-lingual Ability-enhanced Large Language Models

**Zhipeng Chen**[1], **Kun Zhou**[2], **Liang Song**[3], **Wayne Xin Zhao**[1]*,
**Bingning Wang**[3]*, **Weipeng Chen**[3], **Ji-Rong Wen**[1,2]
[1]Gaoling School of Artificial Intelligence, Renmin University of China.
[2]School of Information, Renmin University of China. [3]Baichuan Inc.
zhipeng_chen@ruc.edu.cn,francis_kun_zhou@163.com
batmanfly@gmail.com,daniel@baichuan-inc.com

## Abstract

Multi-lingual ability transfer has become increasingly important for the broad application of large language models (LLMs). Existing work highly relies on training with the multi-lingual ability-related data, which may not be available for low-resource languages. To solve it, we propose a **M**ulti-lingual **A**bilities **E**xtraction and **C**ombination approach (**MAEC**), which decomposes and extracts language-agnostic ability-related weights from LLMs, and combines them across different languages by simple addition and subtraction operations without training. Specifically, our MAEC consists of the extraction and combination stages. In the extraction stage, we firstly locate *key neurons* that are highly related to specific abilities, and then employ them to extract the transferable *ability-related weights*. In the combination stage, we further select the *ability-related tensors* that mitigate the linguistic effects, and design a combining strategy based on them and the *language-specific weights*, to build the multi-lingual ability-enhanced LLM. To assess the effectiveness of our approach, we conduct extensive experiments on LLaMA-3 8B on mathematical and scientific tasks in both high-resource and low-resource lingual scenarios. Empirical results have shown that MAEC can effectively and efficiently extract and combine the advanced abilities, achieving **comparable performance with PaLM**. Resources are available at https://github.com/RUCAIBox/MAET.

## 1 Introduction

Large language models (LLMs) have shown remarkable performance on various general tasks, *e.g.,* text generation and question answering (Zhao et al., 2023; OpenAI, 2023). Despite the success, LLMs are still struggling to solve complex tasks (*e.g.,* mathematical reasoning), which require LLMs to possess specific advanced abilities (*e.g.,* deductive reasoning) and knowledge (*e.g.,* mathematical theory) (Yue et al., 2024; Lu et al., 2022). To address it and further improve LLMs, existing work either collects the related data to train LLMs (Du et al., 2024; Chen et al., 2024a), or merges the parameters of existing well-performed LLMs to transfer their abilities into one single model (Ilharco et al., 2023; Yadav et al., 2023).

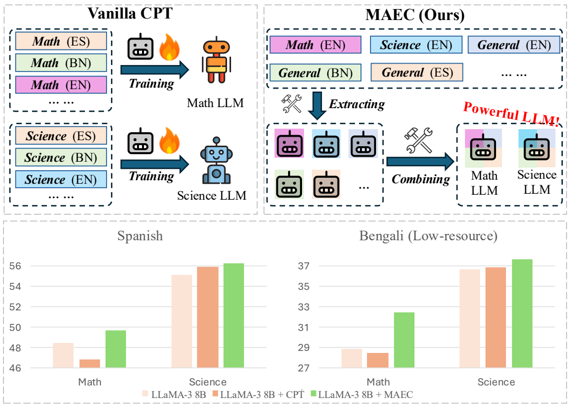


Figure 1: The comparison between CPT and MAEC. Only with the single-lingual ability-related corpus, MAEC can extract the abilities and combine them, achieving effective and efficient domain adaptation.

Despite the success, it is not easy to collect sufficient training corpus or well-trained LLMs related to specific abilities, especially in multi-lingual scenarios. Especially, some popular languages (*e.g.,* English) have dominated the linguistic expressions of the open web data, and the amount of available domain-specific data for low-resource languages (*e.g.,* Bengali or Telugu) is highly limited (Patzelt, 2024; Mirashi et al., 2024). Fortunately, existing work (Zhao et al., 2024; Schäfer et al., 2024) has revealed that the learned knowledge from one language by LLMs could be inherited and leveraged by other languages. For example, Llama-series LLMs are trained mainly on English texts, while they can also solve the tasks based on other languages. Such a finding has been widely explored in

[*] Corresponding authors.

*Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 17563–17580
November 4-9, 2025 ©2025 Association for Computational Linguistics

either improving the overall performance of multilingual LLMs (Schäfer et al., 2024) or enhancing fine-grained knowledge (Chen et al., 2024a). However, the related work mostly requires the ability-related corpus in the target language, which is not always available for low-resource languages.

To conduct a more effective ability transfer, our idea is to learn and extract the "*ability-related weights*" that preserves the knowledge about specific abilities for the LLM. If such ability-related and language-related weights could be decomposed, it is achievable to transfer the required abilities into target languages by just combining the corresponding weights, even building a multi-lingual ability-enhanced LLM like building blocks. Based on this idea, in this paper, we propose a **M**ulti-lingual **A**bilities **E**xtraction and **C**ombination approach, named as **MAEC**. Concretely, our approach consists of two major stages, *i.e.,* ability extracting and combining stage. In the extracting stage, we locate the abilities-related neurons and leverage the related corpus in a reference language to continually pre-train the LLM on the identified neurons. Then, based on the LLM trained on the general corpus, we devise a formula to extract the ability-related weights. In the combining stage, we utilize the ability-related weights to select related tensors, and design a specific merging strategy by interpolating linguistic and ability-related weights. As shown in Figure 1, MAEC only needs ability-related corpus from any rich-resource language and multi-lingual general corpus, efficiently mitigating the data scarcity issues in low-resource languages.

To assess the effectiveness of our approach, we conduct the evaluation based on two complex and comprehensive reasoning benchmarks, *i.e.,* Multi-lingual Grade School Math (MGSM) (Shi et al., 2023) and science tasks from multi-lingual MMLU (Lai et al., 2023) as the evaluation benchmarks. According to the evaluation results, with only training the specific LLM neurons on a small amount of data, the proposed approach MAEC outperforms other competitive baseline methods (*e.g.,* continual pre-training (Gururangan et al., 2020) and model merging methods with task vectors (Ilharco et al., 2023), achieving the 10% relative improvement compared to the base LLM.

## 2 Related Work

**Continual Pre-training.** LLMs still struggle in complex tasks and low-resource lingual scenarios (Hedderich et al., 2021; Shao et al., 2024). To adapt LLMs to a specific scenario, existing work (Luo et al., 2022; Taylor et al., 2022; Zhang et al., 2024a) has collected the related corpus to continually pre-train (CPT) LLMs. During the CPT process, the mixture strategy between the general and ability-related corpus should be considered to avoid hurting their general abilities (Ye et al., 2024; Xie et al., 2023; Siriwardhana et al., 2024). However, previous study (Chang et al., 2024; Lu et al., 2023) has found that it is hard to collect the task-related corpus, especially for low-resource language scenarios. Therefore, synthesizing data from powerful LLMs is utilized to expand the task-related training corpus (Chen et al., 2021b; Zhou et al., 2024a). In this work, we focus on adapting LLMs to multilingual complex reasoning scenarios with only the single-lingual ability-related corpus.

**Knowledge Editing.** According to the lottery ticket hypothesis (Frankle and Carbin, 2019), training a sub-network of the model will achieve comparable or even better performance on downstream tasks. Moreover, several study (Chen et al., 2024b; Zhang et al., 2024b) pointed out that the task-related sub-networks can be determined before the training process. Existing study (Du et al., 2024; Wang et al., 2024b; Gong et al., 2024) has leveraged the inner information of LLMs to select and train the related sub-network. Besides, the probe (*i.e.,* a newly initialized parameter) can be implemented to detect the knowledge of LLMs and process targeted repair (Wang et al., 2024a; Jiang et al., 2024).

**Model Merging.** Given the huge computation resources consumed of CPT, previous work used model merging techniques to integrate different abilities (*e.g.,* mathematical reasoning and code synthesizing) into one model (Yang et al., 2024; Xu et al., 2024b; Stoica et al., 2024). During the merging process, the parameters of different LLMs might be conflict with others, which can be mitigated by the clip (Yadav et al., 2023) or random dropout (Yu et al., 2024) mechanism. Moreover, the LLM inner parameters or external matrixes can be utilized to determine the hyper-parameters of the model merging process (Zhou et al., 2024b; Matena and Raffel, 2022). Furthermore, existing work has merged the reasoning-specialized and multi-lingual models to improve their reasoning ability in non-English scenarios (Huang et al., 2024; Yoon et al., 2024). Inspired by the above work, we try to locate the task-related sub-networks of LLMs and transfer the advanced abilities.

## 3 Preliminary

Despite that LLMs exhibit remarkable performance on general tasks, they still have limited advanced abilities, *e.g.,* mathematical and scientific reasoning abilities. A typical approach to enhance these abilities is to continually pre-train (CPT) LLMs with ability-related corpus. However, such training data might not always be available or sufficient, especially for minor domains (*e.g.,* Bengali). In this work, we focus on the task of *ability extraction and transfer* by continual pre-training and merging LLMs. Concretely, LLMs are trained on the collected corpus from a certain domain, and we aim to only transfer its learned advanced capabilities to target domains (Zhuang et al., 2021; Farahani et al., 2021) without further training. In this work, we study the cross-lingual scene where the linguistic-agnostic advanced ability and linguistic abilities should be extracted and transferred, to build a unified multi-lingual ability-enhanced LLM.

Formally, for a certain ability $A_i$ and a set of languages $L = \{L_0, L_1, \dots, L_n\}$, we assume that the general corpus of all languages can be collected, denoted as $C_{\text{general}} = \{C_{L_0}, C_{L_1}, \dots, C_{L_n}\}$, while the ability-related corpus is only available in language $L_0$ (*i.e.,* English), denoted as $C_{L_0,A_i}$. Based on the above corpora, our goal is to extract and transfer the advanced ability $A_i$ from language $L_0$ and linguistic abilities from other languages $L_1, \dots, L_n$, into a unified LLM.

## 4 Approach

In this section, we propose the **M**ulti-lingual **A**bility **E**xtraction and **C**ombination approach, named as **MAEC**, which can effectively transfer the advanced abilities from single-lingual LLMs, to build a multi-lingual ability-enhanced LLM. The key motivation of our approach is to identify and extract ability-related neurons or weights, and combine the target abilities into a LLM in an efficient way. The framework of MAEC is presented in Figure 2.

### 4.1 Ability-related Weights Extraction

In this part, we aim to locate and learn ability-related parameter weights within an LLM, to enable efficient combining of the ability into other LLMs. Concretely, it consists of two major steps, *i.e.,* key neurons locating and ability-related parameter weights learning.

**Locating the Key Neurons.** The gradient of each neuron in LLMs can be utilized to estimate its correlation degree with specific task ability (Pruthi et al., 2020; Chen et al., 2024b; Xia et al., 2024), we select those with high gradient values as key neurons. To this end, we first use the ability-related corpus $C'_{L_0,A_i}$, which denotes a subset of the original dataset $C_{L_0,A_i}$, to continually pre-train the LLM, as sampling a small amount to train the model can be also applied to reduce the computation consumption. During training, the LLM learns the language modeling task and each neuron is updated by the gradients associated by the training instances. Due to the high cost of calculating the accumulation of gradient at each training step, we calculate the value changes of the LLM neurons before and after the training process to approximate the importance. Formally, the importance function $I(A_i, \theta_j)$ of neurons can be computed as:

$$I(A_i, \theta_j) = \sum_{d_k \in C'_{L_0,A_i}} \text{Grad}\left(\theta_j, d_k\right) \approx \frac{\parallel \tilde{\theta}_j - \theta_j \parallel}{\text{LearningRate}}, \quad (1)$$

where $d_k$ denotes the $k$-th instance of training corpus $C'_{L_0,A_i}$ and $\tilde{\theta}_j$ denote the value of the $j$-th neuron of LLM after training, respectively. Based on it and inspired by previous work (Yadav et al., 2023), we rank all neurons according to their importance scores, and then select the top $k_1$% ones into the set $\mathcal{N}_{A_i}$ as the key neurons. Note that the trained model is only used for identifying the key neurons and is not used as the backbone for subsequent training processes.

**Learning Ability-related Weights.** Based on the identified key neurons in $\mathcal{N}_{A_i}$, we further learn the ability-related parameter weights. Our motivation is to decompose the parameter weights according to their changes *before* and *after* the LLM has mastered a specific ability, which is achievable owing to the modularity and composition nature of the LLM parameter matrices (Yu et al., 2024; Shazeer et al., 2017). First, we utilize the key neurons locating method mentioned above to extract the ability-related neuron set $\mathcal{N}_{A_i}$, and also obtain the language-related neuron set $\mathcal{N}_{L_0}$ via the same way. Then, we train the LLM with the mixture of ability-related corpus and general corpus on the key neuron set $\mathcal{N}_{A_i} \bigcup \mathcal{N}_{L_0}$ and $\mathcal{N}_{L_0}$ respectively, to obtain two specific models, denoted as $\text{LLM}_{A_i,L_0}$

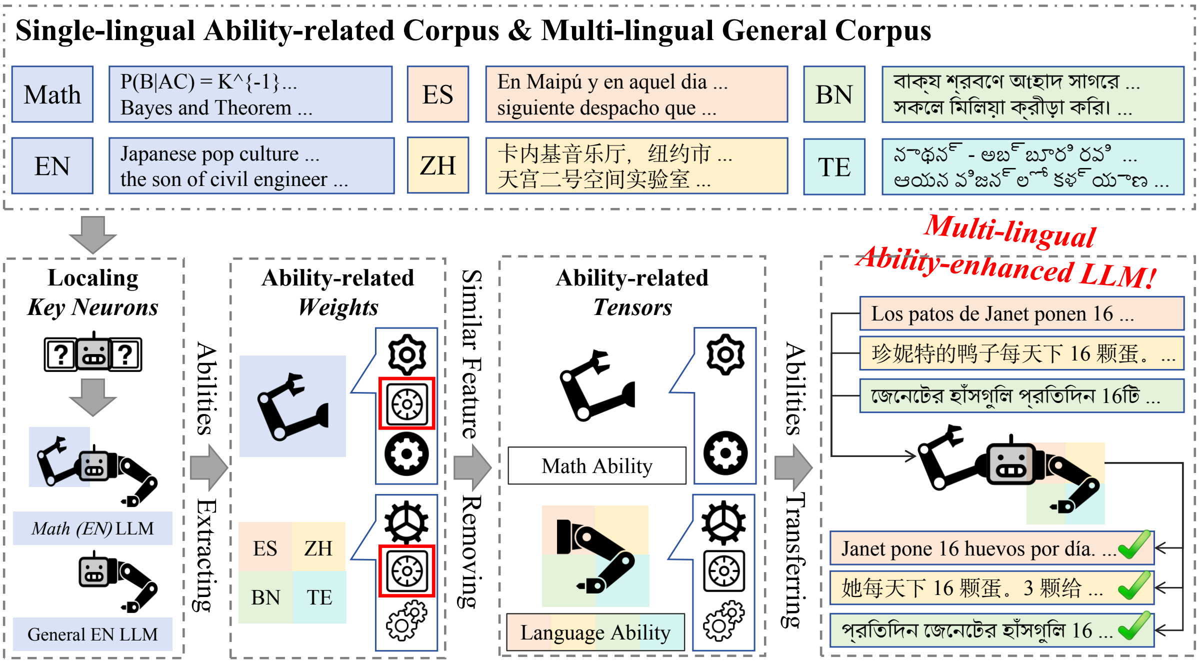

Single-lingual Ability-related Corpus & Multi-lingual General Corpus
Math
P(B|AC) = K^{-1}...
Bayes and Theorem ...
ES
En Maipú y en aquel dia ...
siguiente despacho que ...
BN
বাক্য শ্রবণে আহ্লাদ সাগরে ...
সকলে মিলিয়া ক্রীড়া করি। ...
EN
Japanese pop culture ...
the son of civil engineer ...
ZH
卡内基音乐厅，纽约市 ...
天宫二号空间实验室 ...
TE
నాథన్ - అబ్బూరి రవి ...
ఆయన వీజన్లో కళ్యాణ ...
Multi-lingual
Ability-enhanced LLM!
Locating
Key Neurons
Math (EN) LLM
General EN LLM
Abilities
Extracting
Ability-related
Weights
ES
ZH
BN
TE
Similar Feature
Removing
Ability-related
Tensors
Math Ability
Language Ability
Abilities
Transferring
Los patos de Janet ponen 16 ...
珍妮特的鸭子每天下 16 颗蛋。 ...
জেনেটের হাঁসগুলি প্রতিদিন 16টি ...
Janet pone 16 huevos por día. ... ✓
她每天下 16 颗蛋。3 颗给 ... ✓
প্রতিদিন জেনেটের হাঁসগুলি 16 ... ✓

Figure 2: The framework of MAEC. First, we locate the key neurons, and utilize the single-lingual ability-related corpus and general corpus to train the LLM on these neurons to obtain the ability-related weight. Then, we remove the tensors related to language knowledge in the ability-related weight and combine the remaining to the base LLM. Finally, we obtain a powerful LLM that can solve the related tasks in multi-lingual scenarios.

with parameters $\Theta_{A_i,L_0}$ and $\text{LLM}_{L_0}$ with parameters $\Theta_{L_0}$. Next, we measure the parameter changes between the backbone and the trained models, and obtain the ability-related weights via the parameter decomposition operation as:

$$R(A_i) = \alpha \cdot \underbrace{(\Theta_{A_i,L_0} - \Theta_o)}_{\text{Ability \& language difference}} - \beta \cdot \underbrace{(\Theta_{L_0} - \Theta_o)}_{\text{Language difference}}, \tag{2}$$

where $\alpha$ and $\beta$ are tunable coefficients to balance the two parts of weight differences, and $\Theta_o$ denote the original parameters of the LLM, which serves as the reference for parameter decomposition. As we only train the parameters within the neuron set, its weight difference should preserve the knowledge about the corresponding ability. Thus, it can be regarded as the *ability-related parameter representations*, and is promising to combine the ability into other LLMs by the addition operation.

## 4.2 Multi-lingual Ability Combination

After obtaining the ability-related weights, we combine them to transfer and integrate the abilities, building a multi-lingual ability-enhanced LLM.

**Ability-related Tensor Selection.** Although we can locate the ability-related key neurons, it is still hard to avoid the involvement of irrelevant ones. Our empirical studies in Appendix A have found that neuron-level features are easy to be affected by the noisy data. Inspired by previous work (Cheng et al., 2024a), we consider identifying ability-related tensors to further mitigate the linguistic effects, which correspond to the parameter matrices within the LLM. Specifically, we firstly leverage the ability-related weights of languages $R(L_1), \dots, R(L_n)$ to obtain the multi-lingual weight $R_{Lang}$. Given that large models have varying levels of proficiency in different languages, we use the hyper-parameters $\mu_1, \dots, \mu_n$ to tune this process as:

$$R_{Lang} = \sum_{i=1}^{n} \mu_i \cdot R(L_i), \tag{3}$$

where $R(L_i)$ preserves the linguistic ability of language $L_i$ learned based on Eq. 2. Therefore, $R_{Lang}$ can be considered as the general language ability of LLMs that spans multiple languages. As we aim to find he parameter tensors that have low linguistic effects but focus on the desired abilities (*e.g.,* mathematical reasoning), we rank all the tensors according to their similarities with $R_{Lang}$, and pick up the last $k_2$% ones. Formally, for tensor $\tau_i$, we calculate the cosine similarity of this parameter between $R(A_i)$ and $R_{Lang}$, as follows,

$$S(\tau_i) = \mathtt{sim}\left(R(A_i)[\tau_i], R_{Lang}[\tau_i]\right), \tag{4}$$

| Approaches | MLAR | TPara | AC | AT |
|---|---|---|---|---|
| CPT | Yes | Full | No | No |
| MoE | Yes | Full | No | No |
| LoRA | Yes | Low-Rank | No | No |
| MoL | Yes | Low-Rank | No | No |
| TV | Yes | Full | Yes | No |
| MAEC | **No** | **Ability-related** | **Yes** | **Yes** |

Table 1: The difference between our MAEC and the methods in previous work (*i.e.,* CPT (Hu et al., 2022), Mixture-of-Expert (MoE) (Shazeer et al., 2017), LoRA (Hu et al., 2022), Mixture-of-LoRA (MoL) (Feng et al., 2024), and Task Vector (TV) (Ilharco et al., 2023). MLAR, TPara, AC, and AT denote the abbreviation of multi-lingual ability-related corpus, parameters for training, ability composition, and ability transfer.

where we use the cosine similarity to implement the similarity function $\mathtt{sim}(\cdot)$. After obtaining the similarity of all tensors, we rank them in a descending order based on the similarity values, and then select the last $k_2\%$ parameters into the set $\mathcal{T}$ as the ability-related parameters.

**Building Multi-lingual Ability-enhanced LLM.** Based on the selected ability-related tensors $\mathcal{T}$, we design the model merging process by interpolating ability weights and multi-lingual weights, to build the multi-lingual ability-enhanced LLM. Formally, the final parameter tensors of the target LLM are computed as:

$$\tilde{\tau}_i = \tau_i^{(o)} + \begin{cases} \gamma \cdot R(A_i)[\tau_i] + \eta \cdot R_{Lang}[\tau_i], & \tau_i \in \mathcal{T} \\ R_{Lang}[\tau_i], & \tau_i \notin \mathcal{T} \end{cases}, \quad (5)$$

where $\tau_i^{(o)}$ denotes the original value of parameter tensor $\tau_i$, and $\gamma$ and $\eta$ are tunable hyper-parameters. This formula can be explained in two different cases. When a parameter tensor serves as the major role for specific abilities, we update it by adding both ability- and linguistic-related weights; otherwise, we simply enhance it with multi-lingual weights. In this way, we can derive a more powerful LLM that is equipped with the multi-lingual abilities and specific advanced abilities.

### 4.3 The Overall Procedure

To better demonstrate MAEC, we present key concepts in Table 4 for further clarifying and provide the complete procedure in Algorithm 1. The procedure of MAEC consists of two main stages, *i.e.,* ability-related weights extraction and multi-lingual ability combination. For the extraction stage, we first utilize the accumulated gradient to estimate the importance of each neuron by Eq. 1. Then, we leverage the model trained on the general corpus to remove the effect of language and obtain the ability-related weight through Eq. 2. In the combination stage, we utilize Eq. 3 and Eq. 4 to obtain the multi-lingual weight and identify the ability-related tensors in LLM. After it, we leverage Eq. 5 to fulfill the multi-lingual abilities combination, to build the multi-lingual ability-enhanced LLM.

To highlight the difference between our approach and previous work, we present the comparison of these methods in Table 1. To adapt LLMs to multi-lingual scenarios, most of the existing methods (*e.g.,* CPT and TV) require the multi-lingual ability-related corpus (*i.e.,* ability-related corpus is required for each language) for training the LLM. In comparison, our MAEC only trains and modifies the ability-related parameters, which can efficiently focus on enhancing the specific ability. A major novelty of our work is that we identify the key units and implement the sparse update in the model training and merging procedure, which can effectively decompose, extract, and combine the abilities of LLMs. In addition, compared with the LoRA-based methods (*i.e.,* LoRA and MoL) that also sparsely update the LLM parameters, our approach selectively updates the ability-related neurons, while LoRA-based methods use the low-rank matrices to approximate the original parameters.

## 5 Experiment

### 5.1 Experimental Settings

We introduce the datasets, metrics, and the baselines in our evaluation, and present the implementation details of our approach in Appendix B.

**Datasets.** We focus on transferring the advanced abilities (*i.e.,* mathematical and scientific reasoning abilities) of LLMs from English scenarios to multi-lingual scenarios, including high-resource language (*i.e., Spanish*) and low-resource languages (*i.e., Bengali* and *Telugu*). Thus, for the training corpus, we extract the corpus proposed by previous work (Yang et al., 2023; Scao et al., 2022) as the general corpus, and use *OpenWebMath* (Paster et al., 2024) and *arXiv papers* (Soldaini et al., 2024) as the ability-related corpus for mathematical and scientific tasks. For evaluation, we follow the settings in previous work (OpenAI, 2023), utilizing *Multi-lingual Grade School Math (MGSM)* (Shi et al., 2023) and science tasks from

| Methods | #Tokens | Multilingual Mathematical Tasks | | | | | Multilingual Scientific Tasks | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | | ES | BN | TE | Avg. | ICER (↓) | ES | BN | TE | Avg. | ICER (↓) |
| *Close-source Multi-lingual Large Language Models* | | | | | | | | | | | |
| GPT-3 175B | - | 54.8 | 10.8 | 4.8 | 23.5 | - | - | - | - | - | - |
| PaLM 62B | - | 46.4 | 17.6 | 12.0 | 25.3 | - | - | - | - | - | - |
| cont-PaLM 62B | - | 44.4 | 28.0 | 19.6 | 30.7 | - | - | - | - | - | - |
| Flan-cont-PaLM 62B | - | 53.6 | 34.4 | 28.8 | 38.9 | - | - | - | - | - | - |
| *Open-source Multi-lingual Large Language Models* | | | | | | | | | | | |
| Baichuan-2 7B | - | 17.2 | 4.8 | 2.4 | 8.1 | - | 42.3 | 30.2 | 26.2 | 32.9 | - |
| Mistral 7B | - | 38.8 | 9.6 | 2.8 | 17.1 | - | 52.1 | 32.9 | 28.0 | 37.7 | - |
| LLaMA-2 7B | - | 7.6 | 1.6 | 0.0 | 3.1 | - | 34.2 | 24.6 | 22.2 | 27.0 | - |
| LLaMA-3 8B | - | 48.4 | 28.8 | 20.4 | 32.5 | - | 55.1 | 36.6 | 29.3 | 40.3 | - |
| *Vanilla Continually Pre-training based Approaches* | | | | | | | | | | | |
| + F-CPT$_{\text{L\&A}}$ | 20B | 46.8 | 28.4 | 27.6 | 34.3 | 11.1 | 55.9 | 36.8 | 30.1 | 41.0 | 28.6 |
| + L-CPT$_{\text{L\&A}}$ | 20B | 44.8 | 28.8 | 23.6 | 32.4 | - | 54.8 | 36.4 | 29.9 | 40.4 | 200.0 |
| + F-CPT$_{\text{A}}$ | 4B | 47.2 | 20.0 | 13.2 | 26.8 | - | 51.9 | 33.4 | 29.4 | 38.2 | - |
| + F-CPT$_{\text{L}}$ | 8B | 38.8 | 28.0 | 23.6 | 30.1 | - | 53.6 | 35.9 | 30.6 | 40.0 | - |
| + L-CPT$_{\text{L}}$ | 8B | 46.4 | 28.4 | 22.8 | 32.5 | - | 55.0 | 36.7 | 30.4 | 40.7 | 20.0 |
| *Transfer Learning based Approaches* | | | | | | | | | | | |
| + F-CPT$_{\text{L}}$ & DA | 12B | 41.6 | 30.4 | 27.6 | 33.2 | 17.1 | 52.7 | 35.5 | 28.6 | 38.9 | - |
| + L-CPT$_{\text{L}}$ & DA | 12B | 46.8 | 28.0 | 27.2 | 34.0 | 8.0 | 55.7 | 36.5 | 29.7 | 40.6 | 40.0 |
| *Data Augmentation based Approaches* | | | | | | | | | | | |
| + F-CPT$_{\text{L\&T}}$ | 20B | 48.0 | 28.4 | 25.5 | 34.0 | 13.3 | 53.7 | 35.1 | 31.7 | 40.2 | - |
| + F-CPT$_{\text{T}}$ | 20B | 48.0 | 27.2 | 24.4 | 33.2 | 28.6 | 50.4 | 34.5 | 34.5 | 39.8 | - |
| *Model Merging based Approaches* | | | | | | | | | | | |
| + F-TV | 12B | 42.0 | 16.0 | 10.4 | 22.8 | - | 53.4 | 36.7 | 30.7 | 40.3 | - |
| + L-TV | 12B | 45.6 | 30.8 | 25.6 | 34.0 | 8.0 | 55.5 | 36.7 | 30.4 | 40.9 | 20.0 |
| + MAEC (Ours) | 12B | 49.6 | 32.4 | 25.2 | **35.7** | **3.6** | 56.2 | 37.6 | 30.4 | **41.4** | **10.9** |

Table 2: The performance of different approaches on multilingual mathematical and scientific tasks. ES, BN, and TE denote Spanish, Bengali, and Telugu, respectively. #Tokens denotes the number of training tokens.

*multi-lingual MMLU* (Lai et al., 2023) (*i.e.,* college and high school biology, chemistry, and physics) as the downstream tasks. The statistical information of the datasets is shown in Table 6.

**Evaluation Metrics** We calculate the accuracy of the predicted answers from LLMs and focus on *the average performance (Avg.)*, since our major goal is building a multi-lingual LLM. Moreover, we introduce *the incremental cost-effectiveness ratio (ICER)* (Gafni and Birch, 2006) to assess the efficiency of the approaches, *i.e.,* `ICER = #Tokens / Improvement` ×100%. Notably, we only report the ICER scores for the methods that can lead to improvements.

**Baselines.** We adopt *LLaMA-3 8B* (Dubey et al., 2024) as the backbone model and four categories of widely used methods as baselines, *i.e., continually pre-training*, *transfer learning*, *data augmentation*, and *model merging based* approaches. Concretely, a baseline can be represented as three parts, *i.e.,* training parameters, training approach, and training data. First, we conduct the full parameters training and the LoRA training (Hu et al., 2022), denoted as the *"F"* and *"L"* at the prefix, respectively. Second, for the training approach, we employ continual pre-training *(CPT)* (Gururangan et al., 2020), domain adaption *(DA)* (Taylor et al., 2022), and model merging with task vector *(TV)* (Ilharco et al., 2023). Third, for the training data, *"L", "A", and "T"* refer to the multi-lingual general corpus, English ability-related corpus, and multi-lingual ability-related corpus translated by GPT-4o (Hurst et al., 2024), respectively. Also, we present the performance of open-source LLMs (*i.e.,* Baichuan-2 7B (Yang et al., 2023), Mistral 7B (Jiang et al., 2023), and LLaMA-2 7B (Touvron et al., 2023)) and close-source LLMs (*i.e.,* GPT-3 and PaLM series model (Chung et al., 2024))

## 5.2 Main Results

The evaluation results have been shown in Table 2.

First, MAEC outperforms other baselines in the average performance of all downstream tasks by only expensing 60% computational resources, showing the best incremental cost effectiveness ratio. In our experiment, continually pre-training LLMs on a mixture of multi-lingual general corpus and single-lingual ability-related corpus (*i.e.,* F-CPT$_{\mathrm{L\&A}}$) can enhance the specific ability of LLMs, achieving the second-best performance. However, when adapting LLMs to a new domain or enhancing a new ability of LLM, CPT-based methods should retrain the LLMs on the ability-related and multi-lingual corpus, showing the lack of transferability and requirements of more computational resources. For the new domain adapting, MAEC only utilizes a small amount of single-lingual ability-related corpus (*i.e.,* English corpus in practice) to obtain the ability weight, which can be employed to combine the corresponding advanced ability, achieving both effectiveness and efficiency.

Second, although our MAEC shows similar training efficiency to transfer learning based approaches, MAEC performs better than these baselines, showing the lower ICER score (*e.g.,* 3.6 *v.s.* 8.0). For transfer learning based approaches, since the model is only trained on the single-lingual ability-related corpus during the domain adaptation process, it is difficult for LLM to handle the challenging tasks in multi-lingual scenarios. Concretely, the performance of LLM on the multi-lingual scientific tasks even decreases after domain adaptation, showing a 4% relative decrease. To alleviate this issue, MAEC leverages the calculation between the parameters of different models to extract the ability-related weights, which are language-agnostic and can be transferred to any other scenario.

Third, MAEC also achieves higher performance than data augmentation based approaches (*i.e.,* training LLM on the multi-lingual ability-related corpus translated by GPT-4o). The translation-based method consumes more computational resources and cannot achieve better performance. The reason might be that LLMs cannot perform the translation process well and the translated corpus shares similar knowledge of the specific domain, which makes LLM overfit the corresponding knowledge and cannot really understand the specific knowledge. In contrast, our approach decomposes the advanced ability and language ability, and transfers the advanced ability from one language to another, preventing overfitting, decreasing the expense, and improving performance. These results demonstrate that data-centric methods are difficult to build a multi-lingual ability-enhanced LLM.

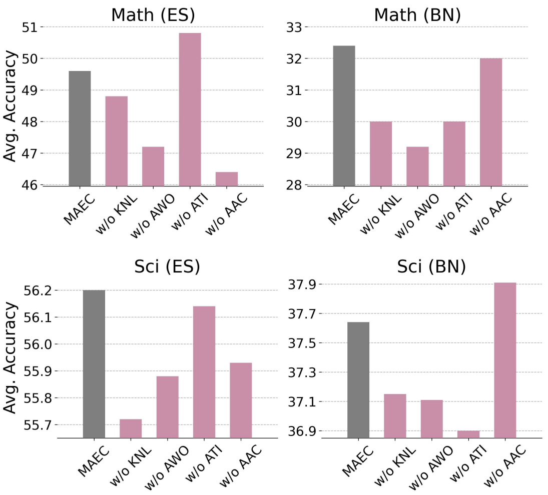


Figure 3: The ablation study. KNL, AWO, ATI, and AAC denote key neurons locating (Eq. 1), ability weights obtaining (Eq. 2), ability-related tensors identifying (Eq. 4), and advanced abilities combining (Eq. 5).

Last, compared with the model merging based approaches (*i.e.,* F-TV and L-TV), experimental results have shown that MAEC performs better than these baseline methods, since we decompose the relation between ability and the language of the training corpus. In the previous model merging approaches, they mainly added the parameters of different models to obtain the final model, without considering the relation between language and abilities. Due to the extraction mechanism of MAEC, we mitigate the effect of languages and make the weight more related to ability, which can be transferred in multi-lingual scenarios.

### 5.3 Detailed Analysis

To further analyze MAEC, we conduct an ablation study, and the analysis of the combining ratio $k_2$ and the generalization of MAEC.

**Ablation Study.** To assess the effectiveness of each component of MAEC, we conduct the ablation study and present the results in Figure 3. We implement MAEC on multi-lingual mathematical and scientific tasks without each module of MAEC, *i.e.,* key neurons locating (*i.e.,* Eq. 1), ability weight obtaining (*i.e.,* Eq. 2), ability-related parameter tensor identifying (*i.e.,* Eq. 4), and advanced abilities transferring (Eq. 5). First, in most downstream scenarios, removing any module of MAEC will affect the final performance, verifying the effectiveness of the MAEC process. Second, without ability weight obtaining, *i.e.,* directly utilizing the difference between LLM trained on

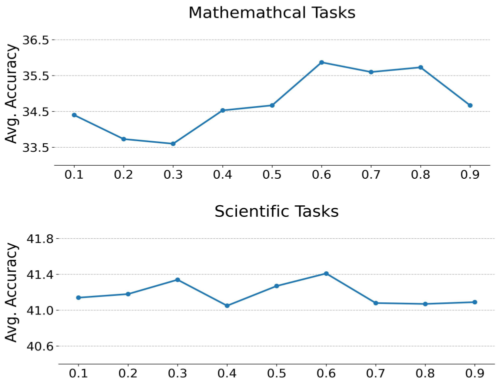


Figure 4: The performance of different proportions for the ability-related parameters identification.

| Methods | MMLU | MBPP | OpenbookQA |
|---|---|---|---|
| LLaMA-3 8B | 60.85 | 46.60 | 65.00 |
| + CPT | -2.39 | -7.00 | -3.60 |
| + MAEC | +0.22 | +0.80 | +0.00 |

Table 3: The out-of-domain performance of different methods to train LLaMA-3 8B on OpenWebMath. After the ability-enhancing process, CPT hurts the OOD abilities of LLM, while MAEC can maintain these abilities.

the ability-related corpus and the backbone LLM as the ability weight, the performance of LLMs is seriously hurt in both scenarios, indicating this process can significantly extract the advanced abilities from the single-lingual corpus and decrease the influence of the language of the training corpus. Third, comparing the results of the models whether adopting the ability transferring process, experimental results show that LLM with the multilingual ability-enhanced cannot well solve multilingual mathematical and scientific tasks, and leveraging the ability weight provided by MAEC can improve the LLM performance on advanced tasks.

**Influence of Combining Ratio $k_2$.** Identifying and updating the ability-related sub-network of LLMs is the key point of our MAEC. We analyze the influence of the combining ratio $k_2\%$ and show the results in Figure 4. Firstly, when the combining ratio $k_2$ changes within a certain range, the model's performance remains largely the same, indicating the strong robustness of our MAEC. Specifically, for the mathematical tasks, when $k_2$ increases from 0.6 to 0.8, the performance of LLM remains approximately 35.5, showing the stability of MAEC. Besides, the performance of LLM has decreased in both extremely low and high ratios of the ability-related parameters identifying process. The main reason is that the lower proportion combines incomplete knowledge to the model and makes LLM unable to possess the corresponding ability, while the higher proportion cannot extract the ability weight precisely and will combine too much language-related knowledge to the model, which conflicts with the LLM's inner knowledge.

**Out-of-Domain Performance of MAEC.** We conduct experiments about adapting mathematical ability on LLaMA-3 8B through MAEC, and assess its performance on out-of-domain (OOD) tasks (*i.e.,* MMLU (Hendrycks et al., 2021), HumanEval (Chen et al., 2021a), MBPP (Austin et al., 2021), and OpenbookQA (Mihaylov et al., 2018)). Results are presented in Table 3. We can observe that the performance of LLM on all evaluation tasks has decreased through the CPT training process, and the maximum decrease has been achieved 7.32% on the HumanEval task. One of the possible reasons is that LLaMA-3 has been trained on OpenWebMath during pre-training and the CPT process makes it overfit and forget the knowledge of other domains, hurting the performance on OOD tasks. In contrast, our proposed MAEC achieves comparable and even better performance with backbone LLM in all downstream scenarios. Since we identify and update the key neurons related to the specific ability, the ability of LLM can be precisely enhanced, and this strategy also helps the OOD tasks needed for mathematical ability, *e.g.,* mathematical tasks in MMLU and MBPP.

## 6 Conclusion

In this paper, we presented MAET, which extracted the advanced ability-related weights from the LLM and supported simple addition and subtraction operations to transfer the ability across different languages. Concretely, MAET included two main stages, *i.e.,* extraction and transfer. For the extraction stage, we located the key neurons and extracted the ability-related weights. Then, in the transfer stage, we identified the key parameter tensors and leveraged them to transfer the advanced ability into other LLMs. In this process, the multi-lingual ability-related training corpus is not required, and the experimental results have shown that our approach outperformed competitive baselines.

As future work, we will consider better methods to identify the ability-related sub-network to decompose the abilities of LLMs and utilize an automated approach to determine the hyper-parameter. Besides, we will implement MAET on larger-scale models, and scenarios with more languages and requiring more abilities to evaluate its effectiveness.

## Limitations

In this section, we discuss the limitations of our work. First, we only implement our approach MAEC on 8B LLMs (*i.e.,* LLaMA-3 8B), and do not adopt the LLMs with larger scales (*e.g.,* 13B or 70B LLMs) in the experiment, due to the limitation of computational resources. We will test the effectiveness of our approach on these LLMs in the future. Second, we only evaluate our approach on two downstream tasks (*i.e.,* mathematical and scientific reasoning tasks) in multi-lingual scenarios. Although they are challenging and widely-used testbeds, it is still meaningful to verify our methods on other tasks. Whereas, as we test the performance on diverse high-resource and low-resource languages, it can also provide comprehensive performance estimation for our approach in multi-lingual scenarios. Finally, we do not consider the potential risk and ethics issues that might hurt the alignment of LLMs when using our approach. Actually, our approach is also applicable to combining the ability to align across languages. We will investigate to it in the future.

## Acknowledgements

This work was partially supported by National Natural Science Foundation of China under Grant No. 92470205 and 62222215, Beijing Natural Science Foundation under Grant No. L233008 and Beijing Municipal Science and Technology Project under Grant No. Z231100010323009. Xin Zhao and Bingning Wang are the corresponding authors.

## References

Jacob Austin, Augustus Odena, Maxwell I. Nye, Maarten Bosma, Henryk Michalewski, David Dohan, Ellen Jiang, Carrie J. Cai, Michael Terry, Quoc V. Le, and Charles Sutton. 2021. Program synthesis with large language models. *CoRR*, abs/2108.07732.

Hsin-Yu Chang, Pei-Yu Chen, Tun-Hsiang Chou, Chang-Sheng Kao, Hsuan-Yun Yu, Yen-Ting Lin, and Yun-Nung Chen. 2024. A survey of data synthesis approaches. *CoRR*, abs/2407.03672.

Jie Chen, Zhipeng Chen, Jiapeng Wang, Kun Zhou, Yutao Zhu, Jinhao Jiang, Yingqian Min, Wayne Xin Zhao, Zhicheng Dou, Jiaxin Mao, Yankai Lin, Ruihua Song, Jun Xu, Xu Chen, Rui Yan, Zhewei Wei, Di Hu, Wenbing Huang, and Ji-Rong Wen. 2024a. Towards effective and efficient continual pre-training of large language models. *CoRR*, abs/2407.18743.

Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Pondé de Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, Alex Ray, Raul Puri, Gretchen Krueger, Michael Petrov, Heidy Khlaaf, Girish Sastry, Pamela Mishkin, Brooke Chan, Scott Gray, Nick Ryder, Mikhail Pavlov, Alethea Power, Lukasz Kaiser, Mohammad Bavarian, Clemens Winter, Philippe Tillet, Felipe Petroski Such, Dave Cummings, Matthias Plappert, Fotios Chantzis, Elizabeth Barnes, Ariel Herbert-Voss, William Hebgen Guss, Alex Nichol, Alex Paino, Nikolas Tezak, Jie Tang, Igor Babuschkin, Suchir Balaji, Shantanu Jain, William Saunders, Christopher Hesse, Andrew N. Carr, Jan Leike, Joshua Achiam, Vedant Misra, Evan Morikawa, Alec Radford, Matthew Knight, Miles Brundage, Mira Murati, Katie Mayer, Peter Welinder, Bob McGrew, Dario Amodei, Sam McCandlish, Ilya Sutskever, and Wojciech Zaremba. 2021a. Evaluating large language models trained on code. *CoRR*, abs/2107.03374.

Richard J Chen, Ming Y Lu, Tiffany Y Chen, Drew FK Williamson, and Faisal Mahmood. 2021b. Synthetic data in machine learning for medicine and healthcare. *Nature Biomedical Engineering*, 5(6):493–497.

Zhipeng Chen, Kun Zhou, Wayne Xin Zhao, Jingyuan Wang, and Ji-Rong Wen. 2024b. Low-redundant optimization for large language model alignment. *CoRR*, abs/2406.12606.

Pei Cheng, Xiayang Shi, and Yinlin Li. 2024a. Enhancing translation ability of large language models by leveraging task-related layers. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, Torino, Italia. ELRA and ICCL.

Pei Cheng, Xiayang Shi, and Yinlin Li. 2024b. Enhancing translation ability of large language models by leveraging task-related layers. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation, LREC/COLING 2024, 20-25 May, 2024, Torino, Italy*, pages 6110–6121.

Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Yunxuan Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, Albert Webson, Shixiang Shane Gu, Zhuyun Dai, Mirac Suzgun, Xinyun Chen, Aakanksha Chowdhery, Alex Castro-Ros, Marie Pellat, Kevin Robinson, Dasha Valter, Sharan Narang, Gaurav Mishra, Adams Yu, Vincent Y. Zhao, Yanping Huang, Andrew M. Dai, Hongkun Yu, Slav Petrov, Ed H. Chi, Jeff Dean, Jacob Devlin, Adam Roberts, Denny Zhou, Quoc V. Le, and Jason Wei. 2024. Scaling instruction-finetuned language models. *J. Mach. Learn. Res.*, 25:70:1–70:53.

Wenyu Du, Shuang Cheng, Tongxu Luo, Zihan Qiu, Zeyu Huang, Ka Chun Cheung, Reynold Cheng, and Jie Fu. 2024. Unlocking continual learning abilities in language models. *CoRR*, abs/2406.17245.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, Arun Rao, Aston Zhang, Aurélien Rodriguez, Austen Gregerson, Ava Spataru, Baptiste Rozière, Bethany Biron, Binh Tang, Bobbie Chern, Charlotte Caucheteux, Chaya Nayak, Chloe Bi, Chris Marra, Chris McConnell, Christian Keller, Christophe Touret, Chunyang Wu, Corinne Wong, Cristian Canton Ferrer, Cyrus Nikolaidis, Damien Allonsius, Daniel Song, Danielle Pintz, Danny Livshits, David Esiobu, Dhruv Choudhary, Dhruv Mahajan, Diego Garcia-Olano, Diego Perino, Dieuwke Hupkes, Egor Lakomkin, Ehab AlBadawy, Elina Lobanova, Emily Dinan, Eric Michael Smith, Filip Radenovic, Frank Zhang, Gabriel Synnaeve, Gabrielle Lee, Georgia Lewis Anderson, Graeme Nail, Grégoire Mialon, Guan Pang, Guillem Cucurell, Hailey Nguyen, Hannah Korevaar, Hu Xu, Hugo Touvron, Iliyan Zarov, Imanol Arrieta Ibarra, Isabel M. Kloumann, Ishan Misra, Ivan Evtimov, Jade Copet, Jaewon Lee, Jan Geffert, Jana Vranes, Jason Park, Jay Mahadeokar, Jeet Shah, Jelmer van der Linde, Jennifer Billock, Jenny Hong, Jenya Lee, Jeremy Fu, Jianfeng Chi, Jianyu Huang, Jiawen Liu, Jie Wang, Jiecao Yu, Joanna Bitton, Joe Spisak, Jongsoo Park, Joseph Rocca, Joshua Johnstun, Joshua Saxe, Junteng Jia, Kalyan Vasuden Alwala, Kartikeya Upasani, Kate Plawiak, Ke Li, Kenneth Heafield, Kevin Stone, and et al. 2024. The llama 3 herd of models. *CoRR*, abs/2407.21783.

Abolfazl Farahani, Behrouz Pourshojae, Khaled Rasheed, and Hamid R. Arabnia. 2021. A concise review of transfer learning. *CoRR*, abs/2104.02144.

Wenfeng Feng, Chuzhan Hao, Yuewei Zhang, Yu Han, and Hao Wang. 2024. Mixture-of-loras: An efficient multitask tuning method for large language models. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation, LREC/COLING 2024, 20-25 May, 2024, Torino, Italy*, pages 11371–11380.

Jonathan Frankle and Michael Carbin. 2019. The lottery ticket hypothesis: Finding sparse, trainable neural networks. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*.

Amiram Gafni and Stephen Birch. 2006. Incremental cost-effectiveness ratios (icers): The silence of the lambda. *Social Science & Medicine*, 62(9):2091–2100.

Zhuocheng Gong, Ang Lv, Jian Guan, Junxi Yan, Wei Wu, Huishuai Zhang, Minlie Huang, Dongyan Zhao, and Rui Yan. 2024. Mixture-of-modules: Reinventing transformers as dynamic assemblies of modules. *CoRR*, abs/2407.06677.

Suchin Gururangan, Ana Marasovic, Swabha Swayamdipta, Kyle Lo, Iz Beltagy, Doug Downey, and Noah A. Smith. 2020. Don't stop pretraining: Adapt language models to domains and tasks. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pages 8342–8360.

Michael A. Hedderich, Lukas Lange, Heike Adel, Jannik Strötgen, and Dietrich Klakow. 2021. A survey on recent approaches for natural language processing in low-resource scenarios. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2021, Online, June 6-11, 2021*, pages 2545–2568.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2021. Measuring massive multitask language understanding. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022. Lora: Low-rank adaptation of large language models. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net.

Zixian Huang, Wenhao Zhu, Gong Cheng, Lei Li, and Fei Yuan. 2024. Mindmerger: Efficient boosting LLM reasoning in non-english languages. *CoRR*, abs/2405.17386.

Binyuan Hui, Jian Yang, Zeyu Cui, Jiaxi Yang, Dayiheng Liu, Lei Zhang, Tianyu Liu, Jiajun Zhang, Bowen Yu, Kai Dang, An Yang, Rui Men, Fei Huang, Xingzhang Ren, Xuancheng Ren, Jingren Zhou, and Junyang Lin. 2024. Qwen2.5-coder technical report. *CoRR*, abs/2409.12186.

Aaron Hurst, Adam Lerer, Adam P Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, et al. 2024. Gpt-4o system card. *arXiv preprint arXiv:2410.21276*.

Gabriel Ilharco, Marco Túlio Ribeiro, Mitchell Wortsman, Ludwig Schmidt, Hannaneh Hajishirzi, and Ali Farhadi. 2023. Editing models with task arithmetic. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*.

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de Las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2023. Mistral 7b. *CoRR*, abs/2310.06825.

Yuxin Jiang, Yufei Wang, Chuhan Wu, Wanjun Zhong, Xingshan Zeng, Jiahui Gao, Liangyou Li, Xin Jiang, Lifeng Shang, Ruiming Tang, Qun Liu, and Wei Wang. 2024. Learning to edit: Aligning llms with knowledge editing. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2024, Bangkok, Thailand, August 11-16, 2024*, pages 4689–4705.

Viet Dac Lai, Chien Van Nguyen, Nghia Trung Ngo, Thuat Nguyen, Franck Dernoncourt, Ryan A. Rossi, and Thien Huu Nguyen. 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, EMNLP 2023 - System Demonstrations, Singapore, December 6-10, 2023*, pages 318–327.

Pan Lu, Swaroop Mishra, Tanglin Xia, Liang Qiu, Kai-Wei Chang, Song-Chun Zhu, Oyvind Tafjord, Peter Clark, and Ashwin Kalyan. 2022. Learn to explain: Multimodal reasoning via thought chains for science question answering. In *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*.

Yingzhou Lu, Huazheng Wang, and Wenqi Wei. 2023. Machine learning for synthetic data generation: a review. *CoRR*, abs/2302.04062.

Renqian Luo, Liai Sun, Yingce Xia, Tao Qin, Sheng Zhang, Hoifung Poon, and Tie-Yan Liu. 2022. Biogpt: generative pre-trained transformer for biomedical text generation and mining. *Briefings Bioinform.*, 23(6).

Michael Matena and Colin Raffel. 2022. Merging models with fisher-weighted averaging. In *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*.

Todor Mihaylov, Peter Clark, Tushar Khot, and Ashish Sabharwal. 2018. Can a suit of armor conduct electricity? A new dataset for open book question answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018*, pages 2381–2391.

Aishwarya Mirashi, Purva Lingayat, Srushti Sonavane, Tejas Padhiyar, Raviraj Joshi, and Geetanjali Kale. 2024. On importance of pruning and distillation for efficient low resource nlp. *arXiv preprint arXiv:2409.14162*.

OpenAI. 2023. GPT-4 technical report. *CoRR*, abs/2303.08774.

Keiran Paster, Marco Dos Santos, Zhangir Azerbayev, and Jimmy Ba. 2024. Openwebmath: An open dataset of high-quality mathematical web text. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*.

Tim Patzelt. 2024. Medical concept normalization in a low-resource setting. *arXiv preprint arXiv:2409.14579*.

Garima Pruthi, Frederick Liu, Satyen Kale, and Mukund Sundararajan. 2020. Estimating training data influence by tracing gradient descent. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.

Morgane Rivière, Shreya Pathak, Pier Giuseppe Sessa, Cassidy Hardin, Surya Bhupatiraju, Léonard Hussenot, Thomas Mesnard, Bobak Shahriari, Alexandre Ramé, Johan Ferret, Peter Liu, Pouya Tafti, Abe Friesen, Michelle Casbon, Sabela Ramos, Ravin Kumar, Charline Le Lan, Sammy Jerome, Anton Tsitsulin, Nino Vieillard, Piotr Stanczyk, Sertan Girgin, Nikola Momchev, Matt Hoffman, Shantanu Thakoor, Jean-Bastien Grill, Behnam Neyshabur, Olivier Bachem, Alanna Walton, Aliaksei Severyn, Alicia Parrish, Aliya Ahmad, Allen Hutchison, Alvin Abdagic, Amanda Carl, Amy Shen, Andy Brock, Andy Coenen, Anthony Laforge, Antonia Paterson, Ben Bastian, Bilal Piot, Bo Wu, Brandon Royal, Charlie Chen, Chintu Kumar, Chris Perry, Chris Welty, Christopher A. Choquette-Choo, Danila Sinopalnikov, David Weinberger, Dimple Vijaykumar, Dominika Rogozinska, Dustin Herbison, Elisa Bandy, Emma Wang, Eric Noland, Erica Moreira, Evan Senter, Evgenii Eltyshev, Francesco Visin, Gabriel Rasskin, Gary Wei, Glenn Cameron, Gus Martins, Hadi Hashemi, Hanna Klimczak-Plucinska, Harleen Batra, Harsh Dhand, Ivan Nardini, Jacinda Mein, Jack Zhou, James Svensson, Jeff Stanway, Jetha Chan, Jin Peng Zhou, Joana Carrasqueira, Joana Iljazi, Jocelyn Becker, Joe Fernandez, Joost van Amersfoort, Josh Gordon, Josh Lipschultz, Josh Newlan, Ju-yeong Ji, Kareem Mohamed, Kartikeya Badola, Kat Black, Katie Millican, Keelin McDonell, Kelvin Nguyen, Kiranbir Sodhia, Kish Greene, Lars Lowe Sjösund, Lauren Usui, Laurent Sifre, Lena Heuermann, Leticia Lago, and Lilly McNealus. 2024. Gemma 2: Improving open language models at a practical size. *CoRR*, abs/2408.00118.

Teven Le Scao, Angela Fan, Christopher Akiki, Ellie Pavlick, Suzana Ilic, Daniel Hesslow, Roman Castagné, Alexandra Sasha Luccioni, François Yvon, Matthias Gallé, Jonathan Tow, Alexander M. Rush, Stella Biderman, Albert Webson, Pawan Sasanka Ammanamanchi, Thomas Wang, Benoît Sagot, Niklas Muennighoff, Albert Villanova del Moral, Olatunji Ruwase, Rachel Bawden, Stas Bekman, Angelina

McMillan-Major, Iz Beltagy, Huu Nguyen, Lucile Saulnier, Samson Tan, Pedro Ortiz Suarez, Victor Sanh, Hugo Laurençon, Yacine Jernite, Julien Launay, Margaret Mitchell, Colin Raffel, Aaron Gokaslan, Adi Simhi, Aitor Soroa, Alham Fikri Aji, Amit Alfassy, Anna Rogers, Ariel Kreisberg Nitzav, Canwen Xu, Chenghao Mou, Chris Emezue, Christopher Klamm, Colin Leong, Daniel van Strien, David Ifeoluwa Adelani, and et al. 2022. BLOOM: A 176b-parameter open-access multilingual language model. *CoRR*, abs/2211.05100.

Anton Schäfer, Shauli Ravfogel, Thomas Hofmann, Tiago Pimentel, and Imanol Schlag. 2024. Language imbalance can boost cross-lingual generalisation. *CoRR*, abs/2404.07982.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. 2024. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *CoRR*, abs/2402.03300.

Noam Shazeer, Azalia Mirhoseini, Krzysztof Maziarz, Andy Davis, Quoc Le, Geoffrey Hinton, and Jeff Dean. 2017. Outrageously large neural networks: The sparsely-gated mixture-of-experts layer. *arXiv preprint arXiv:1701.06538*.

Freda Shi, Mirac Suzgun, Markus Freitag, Xuezhi Wang, Suraj Srivats, Soroush Vosoughi, Hyung Won Chung, Yi Tay, Sebastian Ruder, Denny Zhou, Dipanjan Das, and Jason Wei. 2023. Language models are multilingual chain-of-thought reasoners. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*.

Shamane Siriwardhana, Mark McQuade, Thomas Gauthier, Lucas Atkins, Fernando Fernandes Neto, Luke Meyers, Anneketh Vij, Tyler Odenthal, Charles Goddard, Mary MacCarthy, and Jacob Solawetz. 2024. Domain adaptation of llama3-70b-instruct through continual pre-training and model merging: A comprehensive evaluation. *CoRR*, abs/2406.14971.

Luca Soldaini, Rodney Kinney, Akshita Bhagia, Dustin Schwenk, David Atkinson, Russell Authur, Ben Bogin, Khyathi Raghavi Chandu, Jennifer Dumas, Yanai Elazar, Valentin Hofmann, Ananya Harsh Jha, Sachin Kumar, Li Lucy, Xinxi Lyu, Nathan Lambert, Ian Magnusson, Jacob Morrison, Niklas Muennighoff, Aakanksha Naik, Crystal Nam, Matthew E. Peters, Abhilasha Ravichander, Kyle Richardson, Zejiang Shen, Emma Strubell, Nishant Subramani, Oyvind Tafjord, Evan Pete Walsh, Luke Zettlemoyer, Noah A. Smith, Hannaneh Hajishirzi, Iz Beltagy, Dirk Groeneveld, Jesse Dodge, and Kyle Lo. 2024. Dolma: an open corpus of three trillion tokens for language model pretraining research. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2024, Bangkok, Thailand, August 11-16, 2024*, pages 15725–15788. Association for Computational Linguistics.

George Stoica, Daniel Bolya, Jakob Bjorner, Pratik Ramesh, Taylor Hearn, and Judy Hoffman. 2024. Zipit! merging models from different tasks without training. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*.

Tianyi Tang, Wenyang Luo, Haoyang Huang, Dongdong Zhang, Xiaolei Wang, Xin Zhao, Furu Wei, and Ji-Rong Wen. 2024. Language-specific neurons: The key to multilingual capabilities in large language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2024, Bangkok, Thailand, August 11-16, 2024*, pages 5701–5715.

Ross Taylor, Marcin Kardas, Guillem Cucurull, Thomas Scialom, Anthony Hartshorn, Elvis Saravia, Andrew Poulton, Viktor Kerkez, and Robert Stojnic. 2022. Galactica: A large language model for science. *CoRR*, abs/2211.09085.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton-Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurélien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. 2023. Llama 2: Open foundation and fine-tuned chat models. *CoRR*, abs/2307.09288.

Huanqian Wang, Yang Yue, Rui Lu, Jingxin Shi, Andrew Zhao, Shenzhi Wang, Shiji Song, and Gao Huang. 2024a. Model surgery: Modulating llm's behavior via simple parameter editing. *CoRR*, abs/2407.08770.

Mengru Wang, Ningyu Zhang, Ziwen Xu, Zekun Xi, Shumin Deng, Yunzhi Yao, Qishen Zhang, Linyi Yang, Jindong Wang, and Huajun Chen. 2024b. Detoxifying large language models via knowledge editing. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2024, Bangkok, Thailand, August 11-16, 2024*, pages 3093–3118.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi 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 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.

Mengzhou Xia, Sadhika Malladi, Suchin Gururangan, Sanjeev Arora, and Danqi Chen. 2024. LESS: selecting influential data for targeted instruction tuning. In *Forty-first International Conference on Machine Learning, ICML 2024, Vienna, Austria, July 21-27, 2024*.

Chaojun Xiao, Zhengyan Zhang, Chenyang Song, Dazhi Jiang, Feng Yao, Xu Han, Xiaozhi Wang, Shuo Wang, Yufei Huang, Guanyu Lin, et al. 2024. Configurable foundation models: Building llms from a modular perspective. *arXiv preprint arXiv:2409.02877*.

Sang Michael Xie, Hieu Pham, Xuanyi Dong, Nan Du, Hanxiao Liu, Yifeng Lu, Percy Liang, Quoc V. Le, Tengyu Ma, and Adams Wei Yu. 2023. Doremi: Optimizing data mixtures speeds up language model pretraining. In *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*.

Haoyun Xu, Runzhe Zhan, Derek F. Wong, and Lidia S. Chao. 2024a. Let's focus on neuron: Neuron-level supervised fine-tuning for large language model. *CoRR*, abs/2403.11621.

Zhengqi Xu, Ke Yuan, Huiqiong Wang, Yong Wang, Mingli Song, and Jie Song. 2024b. Training-free pretrained model merging. *CoRR*, abs/2403.01753.

Prateek Yadav, Derek Tam, Leshem Choshen, Colin A. Raffel, and Mohit Bansal. 2023. Ties-merging: Resolving interference when merging models. In *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*.

Aiyuan Yang, Bin Xiao, Bingning Wang, Borong Zhang, Ce Bian, Chao Yin, Chenxu Lv, Da Pan, Dian Wang, Dong Yan, Fan Yang, Fei Deng, Feng Wang, Feng Liu, Guangwei Ai, Guosheng Dong, Haizhou Zhao, Hang Xu, Haoze Sun, Hongda Zhang, Hui Liu, Jiaming Ji, Jian Xie, Juntao Dai, Kun Fang, Lei Su, Liang Song, Lifeng Liu, Liyun Ru, Luyao Ma, Mang Wang, Mickel Liu, MingAn Lin, Nuolan Nie, Peidong Guo, Ruiyang Sun, Tao Zhang, Tianpeng Li, Tianyu Li, Wei Cheng, Weipeng Chen, Xiangrong Zeng, Xiaochuan Wang, Xiaoxi Chen, Xin Men, Xin Yu, Xuehai Pan, Yanjun Shen, Yiding Wang, Yiyu Li, Youxin Jiang, Yuchen Gao, Yupeng Zhang, Zenan Zhou, and Zhiying Wu. 2023. Baichuan 2: Open large-scale language models. *CoRR*, abs/2309.10305.

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. *CoRR*, abs/2408.07666.

Jiasheng Ye, Peiju Liu, Tianxiang Sun, Yunhua Zhou, Jun Zhan, and Xipeng Qiu. 2024. Data mixing laws: Optimizing data mixtures by predicting language modeling performance. *CoRR*, abs/2403.16952.

Dongkeun Yoon, Joel Jang, Sungdong Kim, Seungone Kim, Sheikh Shafayat, and Minjoon Seo. 2024. Langbridge: Multilingual reasoning without multilingual supervision. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2024, Bangkok, Thailand, August 11-16, 2024*, pages 7502–7522. Association for Computational Linguistics.

Le Yu, Bowen Yu, Haiyang Yu, Fei Huang, and Yongbin Li. 2024. Language models are super mario: Absorbing abilities from homologous models as a free lunch. In *Forty-first International Conference on Machine Learning, ICML 2024, Vienna, Austria, July 21-27, 2024*.

Xiang Yue, Tuney Zheng, Ge Zhang, and Wenhu Chen. 2024. Mammoth2: Scaling instructions from the web. *CoRR*, abs/2405.03548.

Wenxuan Zhang, Hou Pong Chan, Yiran Zhao, Mahani Aljunied, Jianyu Wang, Chaoqun Liu, Yue Deng, Zhiqiang Hu, Weiwen Xu, Yew Ken Chia, Xin Li, and Lidong Bing. 2024a. Seallms 3: Open foundation and chat multilingual large language models for southeast asian languages. *CoRR*, abs/2407.19672.

Zhihao Zhang, Jun Zhao, Qi Zhang, Tao Gui, and Xuanjing Huang. 2024b. Unveiling linguistic regions in large language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2024, Bangkok, Thailand, August 11-16, 2024*, pages 6228–6247.

Jun Zhao, Zhihao Zhang, Luhui Gao, Qi Zhang, Tao Gui, and Xuanjing Huang. 2024. Llama beyond english: An empirical study on language capability transfer. *CoRR*, abs/2401.01055.

Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, Yifan Du, Chen Yang, Yushuo Chen, Zhipeng Chen, Jinhao Jiang, Ruiyang Ren, Yifan Li, Xinyu Tang, Zikang Liu, Peiyu Liu, Jian-Yun Nie, and Ji-Rong Wen. 2023. A survey of large language models. *CoRR*, abs/2303.18223.

Kun Zhou, Beichen Zhang, Jiapeng Wang, Zhipeng Chen, Wayne Xin Zhao, Jing Sha, Zhichao Sheng, Shijin Wang, and Ji-Rong Wen. 2024a. Jiuzhang3.0: Efficiently improving mathematical reasoning by training small data synthesis models. *CoRR*, abs/2405.14365.

Yuyan Zhou, Liang Song, Bingning Wang, and Weipeng Chen. 2024b. Metagpt: Merging large language models using model exclusive task arithmetic. *CoRR*, abs/2406.11385.

Fuzhen Zhuang, Zhiyuan Qi, Keyu Duan, Dongbo Xi, Yongchun Zhu, Hengshu Zhu, Hui Xiong, and Qing He. 2021. A comprehensive survey on transfer learning. *Proc. IEEE*, 109(1):43–76.

# A Empirical Study

A surge of work (Zhang et al., 2024b; Xiao et al., 2024; Tang et al., 2024) has pointed out that LLMs sparsely activate the specific sub-modules to perform corresponding tasks. Based on these findings, we conduct empirical experiments to explore whether the specific sub-module, which is related to advanced abilities, can be extracted and combined. We utilize the forum corpus (*i.e.,* Zhihu for Chinese forum corpus and Reddit for English forum corpus) to continually pre-train LLMs, and then assess the training performance (*i.e.,* the value of loss function) and similarity of LLM neurons.

The forum corpus can be considered as containing the question-answering (QA) ability, which is necessary and important for LLMs. The results from Figure 5a have shown that only training the top 5% relevant neurons of LLMs can achieve the lower training loss and fit into the training set more quickly, indicating that LLMs contain the sub-module corresponding to the QA ability. Moreover, from Figure 5b and Figure 5c, we can observe that the LLM trained on Zhihu has shown higher similarity with the LLM trained on Reddit than the LLM trained on Github (*i.e.,* lower L1 Norm and higher cosine similarity), and the cosine similarity of different layers in LLM are largely different.

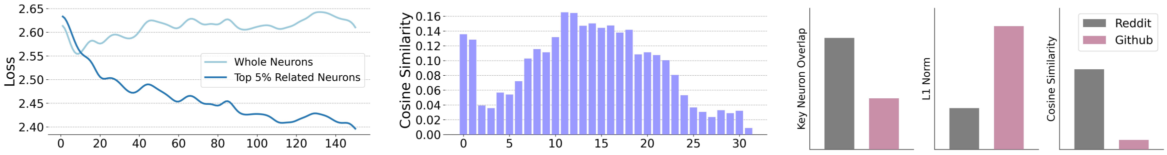
According to the above results, we have found that the different sub-networks of LLMs control the different abilities, and precisely selecting the correct sub-module of LLMs will help the extraction of advanced abilities from the single-lingual corpus and the combination of these abilities to multilingual scenarios. Concretely, although Zhihu and Reddit are in different languages, they will influence the similar sub-modules of LLM and make these sub-networks show high similarity with each other. These sub-networks can be referred to the ability-related sub-networks, which are slightly influenced by languages.

# B Implementation Details

In the experiment, we adapt LLaMA-3 8B as the backbone LLM, and employ `Transformers` (Wolf et al., 2020) and `Deepspeed` framework to perform the training process. And we also present the evaluation results of different backbone LLM (*i.e.,* Qwen2.5 0.5B (Hui et al., 2024) and Gemma2 2B (Rivière et al., 2024)) in Appendix E. For the training process, the learning rate, batch size, and training step are set as $5 \times 10^{-5}$, 1M tokens, and 2B

| Concepts | Meaning |
|---|---|
| Key Neurons | Neuron refers to one of the trainable values of the tensors in LLMs. As previous work pointed out (Xu et al., 2024a), different neurons might control the different abilities of LLMs. Following this finding, in our work, we define the neurons that control the specific ability as the ”Key Neurons”. Key neurons can be regarded as a set without duplication, and a neuron belonging to the set means that this neuron can control the specific ability (Chen et al., 2024b). During the following training process, only the neurons belonging to the key neurons will be trained and optimized. |
| Ability-related Weights | Ability-related weights refer to the value of the whole neuron in LLM, which can represent the corresponding ability of LLM (Yu et al., 2024; Ilharco et al., 2023). In MAET, we obtain the ability-related weights through equation 2. The ability-related weights contain the value of all neurons. Since only the key neurons will be trained during the training process, the value of the neurons not belonging to key neurons is zero in the ability-related weights. |
| Ability-related Tensors | Ability-related tensors can be regarded as a set of LLM tensors, which is related to the corresponding ability. Previous work has studied how the LLM layers influence the ability (Cheng et al., 2024b). Different from key neurons, ability-related tensors focus on higher-level information, integrating the sparse neurons into a coarser-grained element (Xiao et al., 2024). A tensor belonging to the ability-related tensors denotes that this tensor is highly related to the corresponding ability and can control this ability. |
| Language-specific Weights | Similar to the ability-related weights, language-specific weights also refer to the value of the whole neurons in LLMs (Zhang et al., 2024b). However, language-specific weights represent the language abilities of LLM that include multiple abilities (i.e., one language can be regarded as one ability) (Tang et al., 2024), and the method of obtaining them is also different from ability-specific weights. In MAET, we first calculate the ability-related weights of each language and then Integrating these weights together to obtain the language-specific. |

Table 4: The key concepts of our approach.



(a) Loss During Training Process

(b) Similarity of LLM Layers

(c) Similarity of LLM Parameters

Figure 5: The results of empirical experiments. We present the loss of different training methods during the training process, the cosine similarity of LLM layers after being trained on Zhihu and Reddit, and the similarity of LLMs being trained on different training corpus.

tokens, respectively. Besides, for the key neurons locating, we select the top 5% relevant neurons as the key neuron set $\mathcal{N}$ for both stages and identify the last 80% and 60% similar tensor as the key sub-network $\mathcal{T}$ for mathematical reasoning tasks and scientific reasoning tasks respectively.

**Hyper-parameters Selection.** we released all of the hyper-parameters during our experiment in Table 5, to reproduce our proposed approach better. The hyperparameters discussed in the paper can be categorized into two types: training-related parameters (*e.g.,* learning rate, batch size) and training-independent parameters (*i.e.,* $\alpha$, $\beta$, $\gamma$, $\eta$, and $\mu$). Training-related parameters do not require extensive hyperparameter tuning, as existing studies (Dubey et al., 2024; Hui et al., 2024) provide clear guidelines for setting them. On the other hand, training-independent parameters are used to construct ability-related weights, tensors, and language-specific weights. These techniques are similar to those employed in model merging (Ilharco et al., 2023; Yadav et al., 2023), and the hyperparameter setting approach outlined in the paper can be applied. A limited number of hyper-parameter sets can be defined and validated, as the process primarily involves simple additions and subtractions of model parameters, making it computationally inexpensive.

## C Details of Dataset

We present the statistical information of the datasets in Table 6. We mainly consider English,

**Algorithm 1:** The complete procedure of our proposed approach MAET.

```
Input   : Single-lingual ability-related corpus $C_{L_0,A_i}$, multi-lingual general corpus $C_{L_0}, C_{L_1}, \ldots, C_{L_n}$, and the
          parameters of the backbone model $\Theta_o$.
Output  : A well-trained multi-lingual ability-enhanced LLM.

// Ability-related Weights Extraction
$\theta' \leftarrow \mathrm{CPT}(C_{L_0,A_i}, \Theta_o)$;
for $j$-th neuron in $\Theta_o$ do
    Calculate the importance score of the corresponding neuron using Eq. 1;
Identify the key neuron set $\mathcal{N}_{A_i}$;
$\mathrm{LLM}_{A_i,L_0} \leftarrow \mathrm{CPT}(C_{L_0,A}, \Theta_o, \mathcal{N}_{A_i} \cup \mathcal{N}_{L_0})$;
$\mathrm{LLM}_{L_0} \leftarrow \mathrm{CPT}(C_{L_0}, \Theta_o, \mathcal{N}_{L_0})$;
Learning the ability-related weight $R(A_i)$ using Eq. 2;
// Multi-lingual Ability Combination
Obtaining the multi-lingual weight $R_{Lang}$ using Eq. 3;
for $j$-th parameter tensor in LLM do
    Calculate the correlation using Eq. 4;
Identify the ability-related parameters $\mathcal{T}$;
Combine the ability to multi-lingual scenarios using Eq. 5;
Obtain the well-trained multi-lingual ability-enhanced LLM.
```

| Stage | Hyper-Parameter | Mathematical Tasks | Scientific Tasks |
|---|---|---|---|
| Extraction | Learning Rate | $5 \times 10^{-5}$ | $5 \times 10^{-5}$ |
| | Batch Size | 1M Tokens | 1M Tokens |
| | Training Steps | 2B Tokens | 2B Tokens |
| | $\alpha$ in Extraction | 0.8 | 0.8 |
| | $\beta$ in Extraction | 0.2 | 0.2 |
| | Ratio of Key Neurons $k_1$ | 5% | 5% |
| Combination | Learning Rate | $5 \times 10^{-5}$ | $5 \times 10^{-5}$ |
| | Batch Size | 1M Tokens | 1M Tokens |
| | Training Steps | 2B Tokens | 2B Tokens |
| | $\gamma$ in Combining | 0.2 | 0.2 |
| | $\eta$ in Combining | 1.0 | 1.0 |
| | Ratio of Key Tensors $k_2$ | 60% | 60% |
| | $\mu$ for Spanish | 1.5 | 1.5 |
| | $\mu$ for Bengali | 1.2 | 1.2 |
| | $\mu$ for Telugu | 1.2 | 1.2 |

Table 5: The details of hyper-parameters in the training and evaluation process.

Spanish, Chinese, Bengali, and Telugu in our experiment, and utilized English as the in-domain language while others as the out-of-domain languages. For the evaluation datasets, we select MGSM and multi-lingual MMLU as the evaluation benchmarks, which contain the parallel data in different languages and are useful for multi-lingual complex tasks evaluation.

## D Prompt for Translation

```
You should translate the following text
from English to {TARGET LANGUAGE} and
should not modify the latex code or
website code. You should not add any
details that are not mentioned in the
original text.

## English
{ENGLISH TEXT}

## {TARGET LANGUAGE}
```

## E Performance of Small Scale LLMs

We conduct the different LLMs with different sizes (*i.e.*,, Qwen2.5-0.5B and Gemma2-2B) in our experiment to valid the practicality of our approach. We assess MAET and baselines on multi-lingual scientific reasoning tasks and present the evaluation results in Table 7. Comparing the performance of MAET and the baseline methods, we can observe that MAET can also enhance the performance of small scale models and outperform competitive baselines. Therefore, the evaluation results have shown the effectiveness of MAET and verified that

| **Language** | **Training Dataset (Tokens)** | | **Evaluation Dataset (Instances)** | |
|---|---|---|---|---|
| | General Corpus | Ability-related Corpus | Mathematical Tasks | Scientific Tasks |
| English | 1.81B | 1.30B (Math) / 1.82B (Sci) | 250 | 1,245 |
| Spanish | 1.81B | - | 250 | 1,232 |
| Chinese | 1.80B | - | 250 | 1,229 |
| Bengali | 1.81B | - | 250 | 1,137 |
| Telugu | 1.81B | - | 250 | 1,036 |

Table 6: The statistical information of the training and evaluation datasets.

| **Methods** | **Qwen2.5 0.5B** | | | **Gemma2 2B** | | |
|---|---|---|---|---|---|---|
| | ES | TE | Avg. | ES | TE | Avg. |
| Backbone LLM | 36.64 | 25.69 | 31.17 | 43.41 | 30.01 | 36.71 |
| + F-CPT$_{\text{L\&A}}$ | 32.90 | 22.43 | 27.67 | 38.48 | **30.39** | 34.62 |
| + F-CPT$_{\text{A}}$ | 32.62 | 25.26 | 28.94 | 37.83 | 25.39 | 31.61 |
| + MAET w/o API | 36.72 | 28.91 | 32.82 | 43.23 | 29.59 | 36.41 |
| + MAET (Ours) | **36.91** | **29.62** | **33.27** | **43.62** | 30.37 | **37.00** |

Table 7: The performance comparison of different LLMs on multilingual scientific tasks.

MAET is a general LLM enhancement technology.

## F Ability-related Sub-networks of LLM

To assess and probe the ability-related sub-networks of LLMs, we only combine the specific tensors (*i.e.,* tensors in self-attention and MLP mechanism) from the ability weight to the final models through Eq. 5, to analyze the LLM inner abilities. The experimental results are presented in Table 8. From the experiment, we can observe that although the proportion of MLP layers (41.38%) is lower than the attention layers (45.26%), only combining the MLP layers outperforms Combining the attention layers, indicating that the MLP layers are more related to the advanced abilities and stores the corresponding knowledge. In the MLP layers of LLM, the gate mechanism (*i.e.,* MLP Gate) will control the transmission of information and the down project mechanism (*i.e.,* MLP Down) will integrate the knowledge from previous layers, so that Combining the MLP layers can achieve better performance on the downstream tasks.

## G Further Explanation of Baselines

To better introduce the baseline methods, we propose a table to explain the details of each baseline in Table 9.

| **LLM Tensors** | **Proportion of $\mathcal{T}$** | **ES** | **ZH** | **BN** | **TE** | **Avg.** |
|---|---|---|---|---|---|---|
| All Tensors | 100.00% | 49.60 | 41.60 | 32.40 | 25.20 | 37.20 |
| Attention All | 45.26% | 48.80 | 41.60 | 28.80 | 26.40 | 36.40 |
| Attention Q | 12.07% | 47.60 | 40.80 | 30.80 | 26.40 | 36.40 |
| Attention K | 10.34% | 47.20 | 42.40 | 29.60 | 24.40 | 35.90 |
| Attention V | 9.48% | 47.60 | 42.40 | 28.80 | 25.20 | 36.00 |
| Attention O | 13.36% | 48.00 | 40.40 | 30.80 | 27.20 | 36.60 |
| MLP All | 41.38% | 48.80 | 39.60 | 31.60 | 27.60 | 36.90 |
| MLP Up | 13.79% | 50.00 | 40.00 | 28.80 | 25.20 | 36.00 |
| MLP Gate | 13.79% | 46.00 | 41.20 | 30.00 | 24.00 | 35.30 |
| MLP Down | 13.79% | 49.60 | 41.60 | 30.40 | 26.00 | 36.90 |

Table 8: The effect of only merging the specific LLM tensors during the Combining process (*i.e.,* Eq.5) on multilingual mathematical tasks.

| **Baselines** | **Training Parameters** | **Optimization Method** | **General Corpus** | **Ability-relared Corpus** |
|---|---|---|---|---|
| F-CPT$_{L\&A}$ | Full Parameters | CPT | Multi-lingual | Single-lingual |
| L-CPT$_{L\&A}$ | LoRA | CPT | Multi-lingual | Single-lingual |
| F-CPT$_{A}$ | Full Parameters | CPT | No | Single-lingual |
| F-CPT$_{L}$ | Full Parameters | CPT | Multi-lingual | No |
| L-CPT$_{L}$ | LoRA | CPT | Multi-lingual | No |
| F-CPT$_{L\&T}$ | Full Parameters | CPT | Multi-lingual | Translated Multi-lingual |
| F-CPT$_{T}$ | Full Parameters | CPT | No | Translated Multi-lingual |
| F-TV | Full Parameters | Model Merging | Multi-lingual | Single-lingual |
| L-TV | LoRA | Model Merging | Multi-lingual | Single-lingual |

Table 9: The detailed explanation of each baseline mentioned in Table 2.