Training-free LLM Merging for Multi-task Learning

Zichuan Fu[1*], Xian Wu[2*], Yejing Wang[1], Wanyu Wang[1], Shanshan Ye[3], Hongzhi Yin[4], Yi Chang[5], Yefeng Zheng[2,6], Xiangyu Zhao[1†]

[1] City University of Hong Kong [2] Tencent Jarvis Lab
[3] University of Technology Sydney [4] University of Queensland
[5] Jilin University [6] Westlake University

zc.fu@my.cityu.edu.hk, kevinxwu@tencent.com, xianzhao@cityu.edu.hk

Abstract

Large Language Models (LLMs) have demonstrated exceptional capabilities across diverse natural language processing (NLP) tasks. The release of open-source LLMs like LLaMA and Qwen has triggered the development of numerous fine-tuned models tailored for various tasks and languages. In this paper, we explore an important question: is it possible to combine these specialized models to create a unified model with multi-task capabilities. We introduces **H**ierarchical **I**terative **Merging** (Hi-Merging), a training-free method for unifying different specialized LLMs into a single model. Specifically, Hi-Merging employs model-wise and layer-wise pruning and scaling, guided by contribution analysis, to mitigate parameter conflicts. Extensive experiments on multiple-choice and question-answering tasks in both Chinese and English validate Hi-Merging's ability for multi-task learning. The results demonstrate that Hi-Merging consistently outperforms existing merging techniques and surpasses the performance of models fine-tuned on combined datasets in most scenarios. Code is available at Applied-Machine-Learning-Lab/Hi-Merging.

1 Introduction

Large Language Models (LLMs) have revolutionized Natural Language Processing (NLP) by demonstrating unprecedented capabilities in capturing and utilizing world knowledge (Zhao et al., 2024). Recent advances in architecture design and training methodologies have enabled models like GPT-4 (OpenAI, 2023) to engage in human-like dialogue and solve real-world problems, enabling breakthroughs in healthcare, recommender system, and scientific research (Liu et al., 2024a; Fu et al., 2024; Xu et al., 2024b).

With the advent of open-source large language models (LLMs) like LLaMA-3 (Dubey et al., 2024) and Qwen (Yang et al., 2024a), significant research efforts have been dedicated to fine-tuning these models for specific tasks, domains, and languages (Xu et al., 2024c). As a result, Hugging Face[1] now hosts over one million specialized LLMs across various tasks, and this number continues to grow rapidly. These models represent a vast repository of task-specific and language-specific expertise, ranging from medical applications (Chen et al., 2023; Liu et al., 2024b) to financial question and answering (Cheng et al., 2024). A natural question arises: is it possible to combine these task-specific fine-tuned LLMs into a single unified model with broad capabilities, including multilingual and multi-task functionalities? If achievable, the deployment of such a unified model could perform multiple tasks that currently require multiple LLMs, thereby significantly enhancing the application of LLMs. One potential solution is to gather all fine-tuning data and retrain the LLMs from scratch. However, this approach has three significant disadvantages: 1) the availability of fine-tuning data, as the models are often public but the data is not (He et al., 2024); 2) retraining large LLMs requires substantial computational resources; and 3) balancing the training data from different tasks to maintain strong performance across all tasks without compromising any individual task (avoiding the "seesaw effect" (Tang et al., 2020) where improving one task's performance leads to degradation in others) is a non-trivial challenge.

Based on the above considerations, model merging (Yang et al., 2024b) emerges as a promising solution for unifying multiple specialized models while preserving their individual capabilities. However, current model merging methods face two fundamental challenges. First, interference between

[*]Co-first authors with equal contributions.
[†]Corresponding author.

[1]https://huggingface.co/

Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 33111–33124
July 27 - August 1, 2025 ©2025 Association for Computational Linguistics

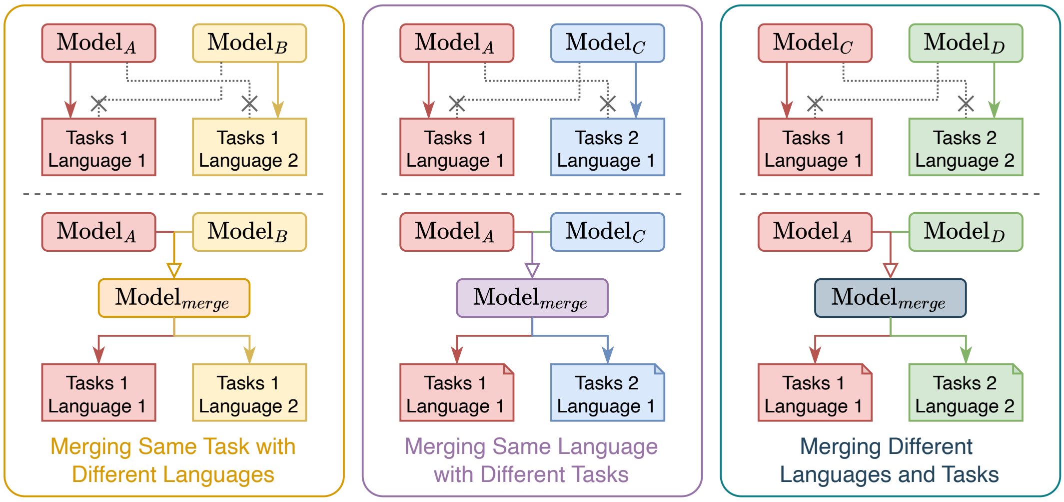


Figure 1: Illustration of three paradigms for our LLM merging: merging models that specialize in different languages (left), merging models that excel at different tasks (middle), and merging models that exhibit expertise in both different languages and different tasks (left). Through such merging, a single model can inherit the combined capabilities of both original models, enabling broader applicability and enhanced performance.

merged models can arise from noise introduced by data bias (Tsuchiya, 2018) or the training process, such as overfitting, impairs the merged model's generalization. Second, models trained independently follow distinct optimization trajectories, leading to different knowledge alignments in their parameter spaces (Ilharco et al., 2023). These misaligned parameters become incompatible for direct combination without additional training.

To address these challenges, we propose Hi-Merging, a **H**ierarchical **I**terative **Merging** method. It first applies model-wise pruning and scaling to the delta vectors (parameter differences between fine-tuned models and the foundation model) to eliminate noisy parameters introduced during fine-tuning. Then, we apply layer-wise pruning and scaling iteratively for the knowledge misalignment, starting from the most conflicted layers. To identify the severity of layer-wise conflicts, we develop contribution analysis - a method that quantifies each layer's contribution by measuring how adding or removing specific layers affects model capabilities. By analyzing how our contribution metrics change before and after a pre-merging process, we can identify potential conflicts, thereby guiding our iterative optimization process to resolve parameter incompatibilities without additional training.

Our contributions can be summarized as follows:

- We investigate the challenges and potential of training-free model merging for integrating LLMs specialized in diverse tasks (e.g., MCQA, QA) and languages (e.g., English, Chinese), addressing a complex multi-task scenario.
- We propose Hi-Merging, a hierarchical iterative approach that effectively reduces the interference of noise and knowledge alignment conflicts during model merging.
- Extensive experiments on four datasets demonstrate the effectiveness of Hi-Merging in multi-task merging across different tasks and languages, consistently achieving superior performance.

2 Preliminary for LLM Merging

In this section, we detail notations and introduce existing LLM merging solutions as the preliminary.

Model merging aims to combine multiple models with distinct capabilities as a single model, which has all the strengths of these models. In this paper, we user two-model merging for illustration: Given models $\mathcal{M}_A$ and $\mathcal{M}_B$ with parameters $\boldsymbol{\theta}_A$ and $\boldsymbol{\theta}_B$, both fine-tuned from a foundation model $\mathcal{M}_F$ with parameters $\boldsymbol{\theta}_F$ for tasks t_A and t_B respectively, model merging aims to combine them into a single model $\mathcal{M}_{\text{merge}}$ with parameters $\boldsymbol{\theta}_{\text{merge}}$ that preserves capabilities for both tasks.

Typical model merging strategies include weighted averaging and delta vector-based merging. The former combines model parameters through a weighted sum (Wortsman et al., 2022):

$$\boldsymbol{\theta}_{\text{merge}} = \sum_{m \in \{A,B\}} \omega_m \boldsymbol{\theta}_m, \qquad (1)$$

where ω_m is the weight to balance different capabilities constrained to $\sum_{m \in \{A,B\}} \omega_m = 1, \omega_m > 0$.

And $m \in \{A, B\}$ is the model identifier.

The second strategy merges models based on delta vectors, the parameter differences between fine-tuned models and their foundation model, which can be mathematically defined as:

$$\boldsymbol{\delta}_m = \boldsymbol{\theta}_m - \boldsymbol{\theta}_F. \tag{2}$$

Delta vectors $\boldsymbol{\delta}_m$ defined in Equation (2) reveal model-specific updates from the foundation model, enabling a delta-weighted merging strategy (Ilharco et al., 2023):

$$\boldsymbol{\theta}_{\text{merge}} = \boldsymbol{\theta}_F + \sum_{m \in \{A,B\}} \omega_m \boldsymbol{\delta}_m. \tag{3}$$

where $\omega_m > 0$. Note that both strategies, illustrated in Equation (1) and Equation (3), can be easily extended to multiple model merging scenarios by expanding the model list $\{A, B\}$.

3 Method

In this section, we introduce the proposed method, which consists of two major components: (1) model-wise pruning and scaling that removes noisy and redundant parameters and moderate excessive ones and (2) layer-wise pruning and scaling iterating on conflicted layers to address knowledge misalignment issues.

3.1 Model-wise Pruning and Scaling

This section introduces two operations to process delta vectors: pruning and scaling.

During the fine-tuning, models can accumulate noisy parameters and learn sharp parameters for the specific fine-tuning task. We introduce the pruning and scaling operations to tackle these two problems, respectively, which are controlled by the following hyperparameters:

- **Pruning Threshold** (p): This parameter specifies the proportion of the delta vector that should be preserved. By retaining the largest p percentage of the vector's components and rendering the remaining $(1 - p)$ to zero, the pruning operation can eliminates trivial parameter updates (data-specific noise) while preserving meaningful task-specific knowledge.
- **Scaling Factor** (s): This factor controls the magnitude of the delta vector. With this parameter, the scaling operation contributes to addressing over-aggressive parameters by scaling down sharp updates, which may result from the overfitting during fine-tuning. The pruning does not apply to large parameter changes as they likely encode essential knowledge. The scaling provides a way to moderate their excessive influence.

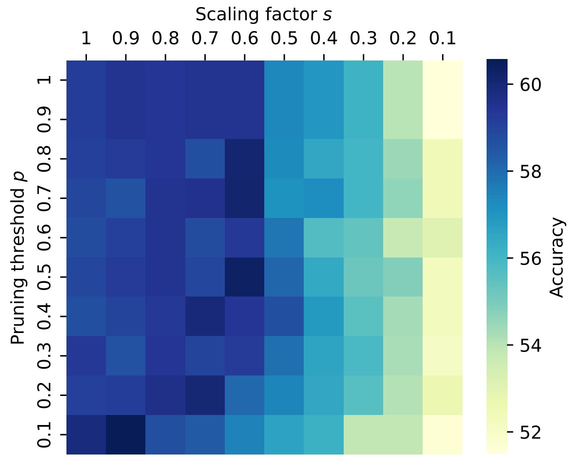


Figure 2: The accuracy of the fine-tuned Qwen2-7B-Instruct on the MedQA dataset after the model-wise pruning and scaling process with different combinations of the pruning threshold p and the scaling factor s.

With these hyperparameters, the pruning and scaling cooperatively process the delta vectors in a complementary manner: pruning eliminates negligible parameter changes while scaling moderates the significant ones. Note that both p and s constrained to $[0, 1]$.

We empirically validate the effectiveness of the pruning and scaling operations by iterating p and s from $[0.1, 1]$. The result is visualized in Figure 2. We can find that the individual model can maintain or even improve performance with appropriate pruning and scaling. For example, $p = 0.1, s = 0.9$ (preserving 10% of parameters and scaling all delta values with 0.9) can defeat the original model ($p = 1, s = 1$). This finding supports our idea of conducting model-wise pruning and scaling to overcome noisy and radical parameter updates.

Next, we introduce the model-wise pruning and scaling details. Specifically, the delta vector (defined in Equation (2)) for a given LLM $\mathcal{M}_m$ can be defined as $\boldsymbol{\delta}_m = [\delta_{m,1}, \delta_{m,2}, \ldots, \delta_{m,N}]$, where $m \in \{A, B\}$ is the model identifier and N indicates the size of trainable parameters.

The **pruning** operation Top_p retains the $\lceil p \cdot N \rceil$ elements of $\boldsymbol{\delta}_m$ with the largest absolute value and zeros out the rest, resulting in $\tilde{\boldsymbol{\delta}}_m$:

$$\tilde{\boldsymbol{\delta}}_m = \mathrm{Top}_p(\boldsymbol{\delta}_m). \tag{4}$$

In detail, the n-th component of $\tilde{\boldsymbol{\delta}}_m$ is:

$$\tilde{\delta}_{m,n} = \begin{cases} \delta_{m,n}, & \text{if } n \in \{\pi(1), \pi(2), \ldots, \pi(\lceil p \cdot N \rceil)\} \\ 0, & \text{otherwise} \end{cases}, \tag{5}$$

where $\pi(n)$ represents the index of the n-th largest component of $\boldsymbol{\delta}_m$ in absolute value, such that:

$$\left|\delta_{m,\pi(1)}\right| \geq \left|\delta_{m,\pi(2)}\right| \geq \cdots \geq \left|\delta_{m,\pi(N)}\right|. \tag{6}$$

The **scaling** operation adjusts the magnitude of the pruned delta vector $\tilde{\boldsymbol{\delta}}_m$ by multiplying it with the scaling factor $s \in [0, 1]$ as $s\tilde{\boldsymbol{\delta}}_m$.

Regarding the different setting of p and s for each model, the model-wise pruning and scaling can be compactly expressed as:

$$\hat{\boldsymbol{\delta}}_m = s_m \cdot \mathrm{Top}_{p_m}(\boldsymbol{\delta}_m) = s_m \tilde{\boldsymbol{\delta}}_m, \tag{7}$$

where $\hat{\boldsymbol{\delta}}_m$ represents the delta vector after the model-wise pruning and scaling.

Through model-wise process with pruning and scaling, we effectively identify noisy and excessive parameter updates from the fine-tuning, maintaining and moderating the key knowledge about the fine-tuning task for the subsequent merging.

3.2 Layer-wise Pruning and Scaling

In this section, we conduct the layer-wise model merging with pruning and scaling operations with a novel contribution analysis method to measure the parameter conflict.

3.2.1 Contribution Analysis

Directly merging the model-wise processed delta vectors $\{\hat{\boldsymbol{\delta}}_m\}_{m \in \{A,B\}}$ as in Equation 1 or Equation 3 will encounter the weight misalignment problem, which is overlooked by existing methods.

To investigate potential conflicts when merging a specific layer, we measure its contribution by calculating the performance difference before and after the merge. Precisely, we assess the merging contribution from two directions:

- **Deletion Impact** (α): To estimate this impact, we first construct a merged model $\mathcal{M}_G$ that merges all layers using the merging process mentioned in Equation (1) or Equation (3). Then, we calculate the performance degradation caused by removing the delta vector for a specific layer.

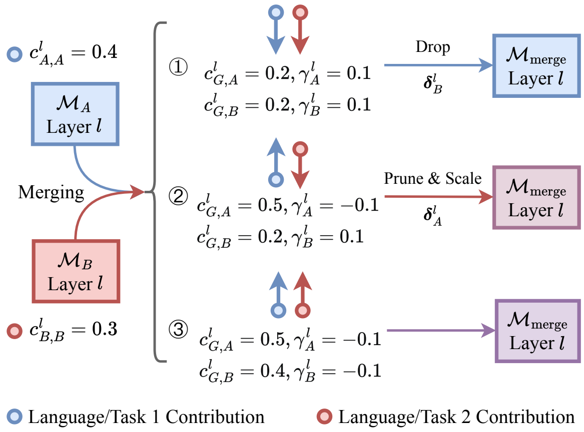


Figure 3: The demonstration of different conflict elimination strategies for three pre-merging conditions.

- **Addition Impact** (β): This impact is measured by the performance improvement of adding the delta vector for a specific layer to the pre-trained foundation model $\mathcal{M}_F$.

These impacts can be mathematically represented as:

$$\alpha^l_{m1,m2} = \mathrm{P}_{t_{m1}}(\boldsymbol{\theta}_{m2} - \hat{\boldsymbol{\delta}}^l_{m2}) - \mathrm{P}_{t_{m1}}(\boldsymbol{\theta}_{m2}), \tag{8}$$

$$\beta^l_{m1,m2} = \mathrm{P}_{t_{m1}}(\hat{\boldsymbol{\theta}}_F + \boldsymbol{\delta}^l_{m2}) - \mathrm{P}_{t_{m1}}(\boldsymbol{\theta}_F), \tag{9}$$

where $m1 \in \{A, B\}$ is the task capability identifier and $m_2 \in \{A, B, G\}$ is the model identifier. We investigate the layer-wise contribution so that l is the layer index. $\hat{\boldsymbol{\delta}}^l_{m2}$ is the delta vector for $\mathcal{M}_{m2}$ at layer l. $\mathrm{P}_{t_{m1}}(\cdot)$ represents the performance metric on the task t_{m1}. For example, BLEU-4 (Papineni et al., 2002) score for the QA task.

We sum up two impacts as the overall contribution:

$$c^l_{m1,m2} = \alpha^l_{m1,m2} + \beta^l_{m1,m2}. \tag{10}$$

3.2.2 Iterative Conflict Elimination

The contribution analysis method defined in Equation (8)-(10) provides a solution to measure the importance of merging specific layers. We can then define the conflict resulted by model $\mathcal{M}_m (m \in \{A, B\})$ within the layer l of the merged model as:

$$\gamma^l_m = c^l_{m,m} - c^l_{m,G}. \tag{11}$$

In this formula, we set the capability identifier $m1 = m$ as we expect the merged model can maintain the performance of $\mathcal{M}_m$ on t_m. We can then identify the most severe conflicting layers that impair the fine-tuned performance by sorting $\Gamma^l = \sum_{m \in \{A,B\}} \gamma^l_m$.

To mitigate the parameter misalignment, we iteratively merge the most conflicting layers (with the largest Γ^l). Specifically, to process a specific layer, there are three types of conflict as illustrated in Figure 3:

1. **Severe Conflict:** $\gamma_A^l > 0$ and $\gamma_B^l > 0$, indicating both capabilities are impaired by the merging. In such cases, only the delta vector with a larger contribution is retained, e.g., dropping $\hat{\boldsymbol{\delta}}_B^l$ in the figure. Namely, $\hat{\boldsymbol{\delta}}_B^l$ is set to zero.
2. **Partial Conflict:** $\gamma_A^l * \gamma_B^l < 0$, i.e., one of the delta vectors leads to the parameter misalignment. The solution for this case is to prune and scale the conflict delta vector again, as we defined in Section 3.1. For example, in Figure 3, the overfitting on t_A ($\gamma_A^l < 0$ and $\gamma_B^l > 0$) leads to the degradation of the ability for t_B. As a result, we prune and scale $\hat{\boldsymbol{\delta}}_A^l$ again as[2]:

$$\hat{\boldsymbol{\delta}}_A^l = s_A \cdot \mathrm{Top}_{p_A}(\hat{\boldsymbol{\delta}}_A^l). \tag{12}$$

3. **Mutual Enhancement:** If $\gamma_A^l \leq 0$ and $\gamma_B^l \leq 0$, the merging process improves for both capabilities. In this case, no further adjustment is necessary for this layer.

After resolving the conflicts of all layers, the parameters of the final merged model $\mathcal{M}_{merge}$ is:

$$\boldsymbol{\theta}_{\text{merge}} = \boldsymbol{\theta}_{\text{F}} + \hat{\boldsymbol{\delta}}_A + \hat{\boldsymbol{\delta}}_B. \tag{13}$$

4 Experiments

In this section, we conduct comprehensive experiments to evaluate the effectiveness of Hi-Merging by answering following research questions (RQ):

- **RQ1:** How does Hi-Merging perform when merging LLMs that excel at the same task but in different languages?
- **RQ2:** How does Hi-Merging perform when merging LLMs excel at different tasks with the same languages?
- **RQ3:** Is Hi-Merging applicable for merging LLMs across languages and tasks?
- **RQ4:** Can Hi-Merging effectively merge different open-source LLMs?
- **RQ5:** How is the merging conflict under our method's settings?
- **RQ6:** What is the impact of different components of Hi-Merging on its overall performance?

[2]We use the same notation $\hat{\boldsymbol{\delta}}_A^l$ for clarity.

Table 1: The brief description and statistics of the four datasets (MedQA (Jin et al., 2020), CMExam (Liu et al., 2023), HealthCareMagic (Li et al., 2023), and cMedQA2) (Zhang et al., 2018) used for fine-tuning.

Name	Task	Language	Train	Validation	Test
MedQA	MCQA	English	10,000	400	400
CMExam	MCQA	Chinese	50,000	4,000	4,000
HealthCareMagic	QA	English	30,000	1,000	1,000
cMedQA2	QA	Chinese	30,000	1,000	1,000

4.1 Experimental Settings

4.1.1 Datasets

We select four datasets listed in Table 1 that cover multilingual multi-task capabilities, including English and Chinese languages, with multiple-choice question answering (MCQA) and open-domain question answering (QA) tasks.

4.1.2 Baselines

In our experiments, we use the multilingual and multi-task models fine-tuned on combined datasets as strong baselines. For model merging approaches, we consider a range of general model merging methods, including weighted averaging (Model Soups (Wortsman et al., 2022)) and delta vector-based approaches (Arithmetic (Ilharco et al., 2023), TIES-Merging (Yadav et al., 2023), DARE (Yu et al., 2024), DELLA (Deep et al., 2024), and Model Breadcrumbs (Davari and Belilovsky, 2024)). We further compare with two knowledge transfer approaches: OT-Fusion (Singh and Jaggi, 2020) and Layer Swapping (Bandarkar et al., 2025). Details are in Appendix A.1.1.

4.1.3 Implementation Details

We use Qwen2-7B-Instruct as foundation modelswith results for other foundation models presented in Appendix A.2. For fine-tuning, we employ LLaMA-Factory [3] with LoRA (rank=8, alpha=16, dropout=0.01) and a batch size of 64. The learning rate is 1.0^{-4} with cosine decay and warm-up. LLM merging is performed using mergekit [4]. Both p and s in model-wise process range from 0.1 to 1.0 with a step of 0.1. In layer-wise process, the pruning threshold p and scaling factor s are successively set to half of their model-wise values.

4.1.4 Evaluation Metrics

For the MCQA task, accuracy is employed to measure the proportion of correct answers (Devlin et al.,

[3]https://github.com/hiyouga/LLaMA-Factory
[4]https://github.com/arcee-ai/mergekit

Table 2: Performance comparison of merging methods for bilingual MCQA task. Model A is fine-tuned on MedQA. Model B is fine-tuned on CMExam. Multi-task model is fine-tuned on both. The overall best result is in bold and the best merging result is underlined.

Types	Methods	L1 (MedQA)	L2 (CMExam)	Avg Impr.	Avg Rank.
Pre-trained	Qwen2-7B-Instruct	51.4062	74.6217	-	17.0
	Yi-1.5-9B	46.8185	58.6499	-16.31%	18.0
	Baichuan2-7B	6.4415	7.1439	-89.22%	19.0
Fine-tuned	Model A (L1)	59.1406	83.7771	+13.40%	10.0
	Model B (L2)	54.4531	88.6171	+13.52%	11.5
	Multi-task	60.0781	88.2246	+17.67%	3.5
Merged	Model Soups	59.6094	88.6926	+17.67%	5.0
	Task Arithmetic	59.5312	88.7681	+17.67%	4.0
	TIES	59.0625	88.7832	+17.31%	4.5
	DARE	58.6719	88.6926	+16.93%	7.5
	DARE + TIES	58.9063	88.6021	+17.04%	8.0
	Model Breadcrumbs	58.8281	88.6322	+17.00%	8.5
	DELLA	58.9844	88.7681	+17.24%	5.5
	DELLA + TIES	58.2812	88.7530	+16.67%	9.5
	OT-Fusion	59.8271	88.6543	+17.60%	3.0
	Layer Swapping	55.6406	87.0859	+16.24%	16.0
	Hi-Merging (Ours)	**60.1562**	**89.0700**	**+18.41%**	**1.0**

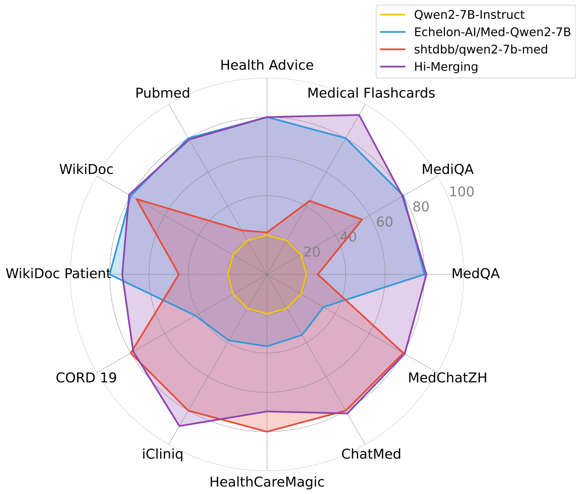


Figure 4: Performance of Hi-Merging on two open-source medical models, Echelon-AI/Med-Qwen2-7B and shtdbb/qwen2-7b-med, which are fine-tuned from the foundation model Qwen/Qwen2-7B-Instruct.

2019).For the QA task, we use BLEU-4 (Papineni et al., 2002) to evaluate the precision of the generation, and ROUGE-1,2,L (Lin, 2004) to assess the overlap and coherence with the ground truth. Additionally, we report both the average relative performance improvement (Avg Impr.) and the mean ranking (Avg Rank.) across all methods.

4.2 Bilingual Task Merging (RQ1)

We first verify the effectiveness of Hi-Merging on bilingual task merging. Here, we merge models trained on the MCQA task in English and Chinese, as shown in Table 2. Additional experiments on the QA task and a different LLM are provided in Appendix A.2 due to space constraints.

Baseline methods like Model Soups and Task Arithmetic that combine models without considering noises and conflicts achieve stable but lower performance. Methods that reduce conflicts, such as TIES and DARE, occasionally achieve the best results on individual metrics. However, without a clear guidance, their performance highly randomised. In contrast, our Hi-Merging method, with hierarchical pruning and scaling approach, not only achieves the best average performance but attains optimal results in about half of the individual metrics. We also investigate the impact of different training sample sizes in Appendix A.3.

4.3 Monolingual Multi-task Merging (RQ2)

For monolingual multi-task merging, we combine models trained on different tasks with the same language (e.g., English MCQA with English QA), as shown in Table 3. The results show that merged models consistently outperform their individual fine-tuned counterparts, with many even surpassing multi-task fine-tuned models. Notably, our Hi-Merging approach achieves a 1.84% relative improvement over the multi-task fine-tuned model. We attribute this success to three factors. 1) During multi-task fine-tuning with limited data (compared to pre-training), tasks can interfere with each other due to the "seesaw effect". In contrast, model merging allows parameters to be optimized independently before integration, avoiding such interference. 2) Since both models are fine-tuned from the same foundation model, their parameter updates tend to follow similar optimization trajectories, making successful merging more likely. 3) The inherent sparsity of LLMs provides sufficient parameter space to accommodate multi-task knowledge from both models.

4.4 Bilingual Multi-task Merging (RQ3)

For bilingual multi-task merging, we combine models trained on completely different tasks and languages. Specifically, we merge a model trained for MCQA in one language (Model A: MedQA in English or CMExam in Chinese) with another model trained for QA in the opposite language (Model B: cMedQA2 in Chinese or HealthCareMagic in English), as illustrated in Table 4.

Our experiments reveal an interesting pattern: bilingual multi-task fine-tuning mainly affects QA performance, while MCQA performance remain. This can be explained by two factors: (1) QA

Table 3: Performance comparison of merging methods for tasks with different question formats. Model A is fine-tuned on MCQA tasks (T1), while model B is fine-tuned on QA tasks (T2). The overall best result is marked in bold and the best merging result is underlined.

Types	Methods	L1 (English)					L2 (Chinese)					Avg Impr.	Avg Rank.
		T1 (MedQA)	T2 (HealthCareMagic)				T1 (CMExam)	T2 (cMedQA2)					
		Accuracy	BLEU-4	ROUGE-1	ROUGE-2	ROUGE-l	Accuracy	BLEU-4	ROUGE-1	ROUGE-2	ROUGE-l		
Pre-trained	Qwen2-7B-Instruct	51.4062	30.1209	26.3524	5.3280	15.7451	74.6217	1.7090	14.1527	1.7822	9.0934	-	-
Fine-tuned	Model A (T1)	59.1406	34.6533	28.7482	6.9168	17.9525	88.6171	2.8064	16.8617	2.5603	12.0561	+17.36%	11.2
	Model B (T2)	53.0469	35.5717	30.2512	8.9044	20.3625	81.5670	4.4159	21.2210	4.0680	17.4600	+20.21%	7.2
	Multi-task	59.2188	35.6009	30.2101	9.1375	20.4645	88.6926	3.7790	20.5919	3.8096	16.9265	+25.23%	8.3
Merged	Model Soups	58.5156	36.4411	30.5654	9.1754	20.4259	88.8285	4.3912	21.0216	4.0040	17.2843	+26.19%	5.6
	Task Arithmetic	58.5938	36.3290	**30.6624**	**9.1945**	**20.5406**	88.7983	4.3018	20.6467	3.7496	16.9995	+26.13%	6.1
	TIES	60.4688	35.7851	30.3243	9.0310	20.3723	88.6171	4.5434	**21.5629**	4.1910	17.4909	+26.78%	4.2
	DARE	58.4375	**36.5802**	30.5488	9.0818	20.3945	88.7681	4.5487	21.3255	3.8403	17.4471	+26.29%	4.4
	DARE+TIES	59.3750	35.7062	30.1950	8.7840	20.0878	88.8285	4.1587	21.1291	3.8124	17.2868	+25.63%	7.5
	Model Breadcrumbs	57.8906	36.4620	30.2173	8.7845	20.0169	88.8889	4.4472	21.2492	3.8931	17.2846	+25.53%	6.4
	DELLA	58.5938	36.3494	30.1715	8.8125	20.1879	88.8134	4.3718	21.0226	3.9300	17.3403	+25.83%	7.1
	DELLA+TIES	59.5312	36.0774	30.4743	9.1151	20.4599	88.6021	4.3202	21.2269	4.0164	17.3779	+25.96%	5.7
	Hi-Merging (Ours)	**60.5469**	36.4926	30.5467	9.1231	20.3523	**88.9795**	**4.6781**	21.5367	**4.2165**	**17.5038**	**+27.07%**	**2.1**

Table 4: Performance comparison of merging methods for bilingual multi-task learning. Model A is fine-tuned on MCQA datasets (T1: MedQA or CMExam). Model B is fine-tuned on QA datasets (T2: cMedQA2 or HealthCareMagic). The overall best result is marked in bold and the best merging result is underlined.

Types	Methods	T1, L1 (MedQA)	T2, L2 (cMedQA2)				T1, L2 (CMExam)	T2, L1 (HealthCareMagic)				Avg Impr.	Avg Rank.
		Accuracy	BLEU-4	ROUGE-1	ROUGE-2	ROUGE-l	Accuracy	BLEU-4	ROUGE-1	ROUGE-2	ROUGE-l		
Pre-trained	Qwen2-7B-Instruct	51.4062	1.7090	14.1527	1.7822	9.0934	74.6217	30.1209	26.3524	5.3280	15.7451	-	-
Fine-tuned	Model A (T1)	59.1406	2.8064	16.8617	2.5603	12.0561	88.6171	34.6713	28.4279	6.6122	18.1117	+17.17%	10.7
	Model B (T2)	54.4922	4.4159	21.2210	4.0680	17.4600	79.6875	35.5717	30.2512	8.9044	20.3625	+20.03%	7.2
	Multi-task	**60.7812**	3.8473	20.8741	4.0434	16.9525	**88.9795**	35.7429	30.1735	8.9153	20.3902	+26.22%	6.8
Merged	Model Soups	58.3584	4.6592	21.2316	4.0559	17.3805	88.6322	36.1765	**30.7169**	**9.2702**	**20.5227**	+26.35%	4.5
	Task Arithmetic	58.0469	4.6682	21.2618	4.0984	17.4231	88.7379	36.1222	30.2256	8.7570	20.1357	+25.69%	5.7
	TIES	59.6094	4.3764	21.0083	3.9002	17.4194	88.7228	35.7708	30.5143	8.8994	20.3487	+26.16%	6.4
	DARE	57.8906	4.5671	21.1856	3.9549	17.2328	88.6322	35.8639	30.1489	8.8150	20.1025	+25.22%	8.1
	DARE+TIES	58.75	4.4929	21.3194	4.0824	17.4826	88.5568	34.8223	29.7597	8.3004	19.7624	+24.76%	7.8
	Model Breadcrumbs	57.1094	4.7217	21.4192	4.1477	17.4182	88.6021	36.4961	30.3911	9.0696	20.4108	+25.82%	4.3
	DELLA	58.0469	**4.8065**	**21.5135**	4.1356	17.4962	88.6167	36.0159	30.3747	9.0414	20.3929	+26.11%	**3.9**
	DELLA+TIES	59.0625	4.4854	20.9954	4.0491	**17.5630**	88.6624	35.0176	29.9666	8.6580	20.1406	+25.31%	7.5
	Hi-Merging (Ours)	60.2344	4.7743	21.1954	**4.1749**	17.3991	88.7983	**36.5223**	30.3932	8.7882	20.1619	**+27.02%**	4.1

Table 5: Performance of Hi-Merging on two open-source medical models: Echelon-AI/Med-Qwen2-7B and shtdbb/qwen2-7b-med.

Types	Models	Medical		Math
		MedQA	Pubmed	GSM8K
Single	Pre-trained	37.3868	5.7994	**65.96**
	Echelon-AI/Med-Qwen2-7B	64.2862	**92.9898**	15.92
	shtdbb/qwen2-7b-med	40.1598	11.5017	57.77
Merged	Hi-Merging (Medical)	**64.9011**	92.1692	56.63

Table 6: Performance of Hi-Merging after merging the medical model with the math-specialized model Qwen2-Math-7B-Instruct.

Types	Models	Medical		Math
		MedQA	Pubmed	GSM8K
Single	Pre-trained	37.3868	5.7994	65.96
Merged	Merged (Medical)	**64.9011**	**92.1692**	56.63
	Qwen2-Math-7B-Instruct	36.7716	15.3618	**79.00**
Merged	Hi-Merging (Medical + Math)	63.6742	91.7320	77.45

tasks require complex free-form generation, making them more vulnerable to joint fine-tuning; (2) MCQA tasks involve clear classification boundaries and simpler choice selection, making them more robust to merging process.

4.5 Open-source LLM Merging (RQ4)

To validate the generality of our merging approach, we conduct experiments using two open-source medical models from Hugging Face: Echelon-AI/Med-Qwen2-7B [5], fine-tuned on English datasets for tasks such as medical QA and information retrieval (IR), and shtdbb/qwen2-7b-med [6], fine-tuned on Chinese datasets for dialogue generation. Both models are derived from Qwen2-7B-Instruct. Figure 4 illustrates the performance comparison across 12 medical datasets, with metrics normalized for better visualization.

Our approach demonstrates robust performance across the task spectrum. In 7 out of 12 datasets, Hi-Merging achieves the best performance among all models, with only two datasets showing appar-

[5]https://huggingface.co/Echelon-AI/Med-Qwen2-7B

[6]https://huggingface.co/shtdbb/qwen2-7b-med

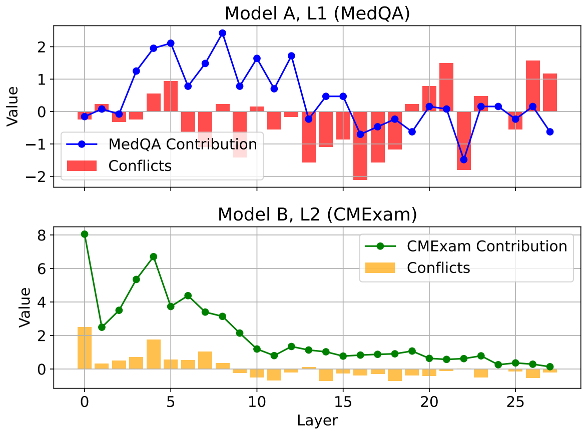


Figure 5: The visualization of layer contributions and merging conflicts when merging model fine-tuned on MedQA and CMExam.

ent degradation compared to the better-performing individual model. These results demonstrate Hi-Merging's ability to effectively fuse medical knowledge while maintaining or enhancing performance across diverse languages and tasks. Detailed implementation setup and unprocessed numerical results can be found in Appendix A.1.2 and A.4.

To further demonstrate the adaptability of our approach in other domains, we merge the combined medical model with an additional mathematical model Qwen/Qwen2-Math-7B-Instruct [7]. As shown in Table 5 and Table 6, merging two models from the same domain (medical) leads to mutually beneficial integration. However, when further merging the medical model with an LLM from a different domain (math), we observe slight performance drops on both domains, suggesting that larger divergence in fine-tuning data increases the difficulty of effective knowledge integration.

4.6 Case Study (RQ5)

Figure 5 visualizes layer-wise contributions and merging conflicts when combining MedQA and CMExam models, revealing that conflicts are not uniformly distributed. Model A (MedQA) shows significant conflicts in later layers, while Model B (CMExam) exhibits conflicts in earlier layers. This non-uniformity highlights the need for Hi-Merging's hierarchical pruning and scaling strategy, leading to the improved performance demonstrated in previous experiments in Table 8. Such layer-specific conflict patterns suggest that different layers may specialize in different tasks, making a uniform merging strategy suboptimal.

[7] https://huggingface.co/Qwen/Qwen2-Math-7B-Instruct

4.7 Ablation Study (RQ6)

Table 7: Ablation study of different processes in Hi-Merging for bilingual MCQA task merging. Model A is fine-tuned on MedQA. Model B is fine-tuned on CMExam. Multi-task model is fine-tuned on both. The overall best result is in bold.

Types	Methods	L1 (MedQA)	L2 (CMExam)	Avg Impr.
Pre-trained	Qwen2-7B-Instruct	51.4062	74.6217	-
Fine-tuned	Model A (L1)	59.1406	83.7771	+13.40%
	Model B (L2)	54.4531	88.6171	+13.52%
	Multi-task	60.0781	88.2246	+17.67%
Merged	w/o All	59.5312	88.5291	+17.48%
	w/o Model-wise Process	59.8437	88.6501	+17.83%
	w/o Model-wise Pruning	60.0781	88.9342	+18.24%
	w/o Model-wise Scaling	59.9219	88.7863	+18.00%
	w/o Layer-wise Process	59.6094	88.5417	+17.55%
	w/o Layer-wise Pruning	**61.0156**	88.6473	**+18.75%**
	w/o Layer-wise Scaling	59.7656	88.6926	+17.80%
	Hi-Merging	60.1562	**89.0700**	+18.41%

The ablation study in Table 7 reveals several key insights. Layer-wise process has a more significant impact than model-wise process, and removing scaling operations leads to larger performance drops than removing pruning. While removing layer-wise pruning achieves the highest average improvement, it shows less consistent performance across tasks compared to the full Hi-Merging approach, indicating that pruning helps stabilize the merging process despite potentially limiting peak performance on specific tasks.

5 Related Works

5.1 Multilingual Task-Oriented LLMs

Multi-task learning (MTL) has proven valuable across various domains, from recommendation systems (Wang et al., 2023a; Zhang et al., 2024) to knowledge graphs (Xu et al., 2024a) and healthcare (Liu et al., 2024c), by enabling models to share knowledge between related tasks. This paradigm becomes especially relevant for multilingual NLP, where different languages face similar challenges in tasks like machine translation (Wang et al., 2022), text summarization (Gambhir and Gupta, 2017), and sentiment analysis (Dashtipour et al., 2016).

Recently, LLMs have greatly contributed to advancing multilingual tasks by leveraging massive amounts of multilingual data (Brown et al., 2020; Devlin et al., 2019; Xue et al., 2021). Despite their success, LLMs exhibit a clear performance gap across languages: they excel at widely-spoken languages with abundant training data but struggle with less-represented languages that have limited online presence (Wang et al., 2023b).

To enhance multilingual capabilities, LLMs employ continual training on specific languages, as seen in models like Chinese-LLaMA (Cui et al., 2023) and EuroLLM (Martins et al., 2024). Additionally, supervised fine-tuning techniques, such as LoRA in Chinese-Alpaca (Cui et al., 2023), further improve multilingual understanding. However, LLMs are usually enhanced for one language at a time, resulting in multiple isolated models.

5.2 Model Merging

Model merging aims to integrate knowledge from multiple fine-tuned models into a single one. These methods are categorized into two types: weighted-based merging and interference mitigation.

Weighted-based merging focuses on combining model parameters effectively. This includes simple techniques like parameter averaging, such as Model Soups (Wortsman et al., 2022), Fisher-weighted merging (Matena and Raffel, 2022) and RegMean (Jin et al., 2023). While computationally efficient, these methods often miss conflicting parameter updates, leading to performance degradation. Therefore, Task Arithmetic (Ilharco et al., 2023) proposes manipulating delta vectors. AdaMerging (Yang et al., 2024c) and evolutionary algorithms (Akiba et al., 2024) optimize merging coefficients and blend diverse models, respectively.

Interference mitigation techniques aim to reduce parameter conflicts based on the over-parameterization and sparsity of LLM. SparseGPT (Frantar and Alistarh, 2023) show high LLM performance despite significant parameter pruning. DELLA (Deep et al., 2024) introduces MAGPRUNE for selective pruning and parameter rescaling. However, these techniques focus mainly on individual parameter-level operations without considering the structural relationships and knowledge dependencies across model layers.

6 Conclusion

In this paper, we proposed Hi-Merging, a novel approach for merging LLMs for multilingual multi-task learning. Hi-Merging leverages model-wise and layer-wise pruning and scaling strategy to minimize the conflict between fine-tuned models' delta vectors. The model-wise process eliminates the fine-tuning noise and overfitting parameters of the original models. Then, the layer-wise process analyzes the contribution of each layer's delta vector to the fine-tuning performance, reducing the interference of conflicts in several key layers. Extensive experiments on the MCQA and QA datasets demonstrated that Hi-Merging outperforms traditional merging techniques and even surpasses models trained on multiple datasets. Future work will explore finer-grained conflict analysis strategies.

7 Limitations

While our proposed Hi-Merging method demonstrates promising results, several limitations should be acknowledged. First, our current method only supports merging two models at a time. Extending the approach to simultaneously merge multiple models presents additional challenges in terms of conflict resolution and computational complexity, which requires further investigation.

Second, our evaluation is currently limited to two task types (MCQA and QA) and two languages (English and Chinese). The effectiveness of Hi-Merging on a broader range of NLP tasks and language families remains to be investigated. This includes exploring its applicability to tasks such as text generation, summarization, and semantic parsing across diverse language groups.

Third, our method focuses on merging models fine-tuned from the same foundation model. The applicability and performance of Hi-Merging when merging models from different architectural families or pre-training approaches is yet to be explored. This limitation becomes particularly relevant as the field continues to see diverse model architectures and training paradigms.

Finally, our current implementation assumes relatively balanced task importance. The method might need adaptation for scenarios where certain tasks or languages should be prioritized over others, potentially requiring a more flexible weighting mechanism in the merging process. Future work could explore dynamic weighting strategies that adapt to specific application requirements and performance objectives.

Acknowledgments

This research was partially supported by Research Impact Fund (No.R1015-23), Collaborative Research Fund (No.C1043-24GF) and Tencent (CCF-Tencent Open Fund, Tencent Rhino-Bird Focused Research Program).

References

Takuya Akiba, Makoto Shing, Yujin Tang, Qi Sun, and David Ha. 2024. Evolutionary optimization of model merging recipes. *CoRR*, abs/2403.13187.

Lucas Bandarkar, Benjamin Muller, Pritish Yuvraj, Rui Hou, et al. 2025. Layer swapping for zero-shot cross-lingual transfer in large language models. In *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*. OpenReview.net.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, et al. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.

Junying Chen, Xidong Wang, Anningzhe Gao, Feng Jiang, Shunian Chen, Hongbo Zhang, Dingjie Song, Wenya Xie, Chuyi Kong, Jianquan Li, Xiang Wan, Haizhou Li, and Benyou Wang. 2023. Huatuogpt-ii, one-stage training for medical adaption of llms. *Preprint*, arXiv:2311.09774.

Daixuan Cheng, Shaohan Huang, and Furu Wei. 2024. Adapting large language models via reading comprehension. In *The Twelfth International Conference on Learning Representations*.

Yiming Cui, Ziqing Yang, and Xin Yao. 2023. Efficient and effective text encoding for chinese llama and alpaca. *arXiv preprint arXiv:2304.08177*.

Kia Dashtipour, Soujanya Poria, Amir Hussain, Erik Cambria, Ahmad Y. A. Hawalah, Alexander F. Gelbukh, and Qiang Zhou. 2016. Multilingual sentiment analysis: State of the art and independent comparison of techniques. *Cogn. Comput.*, 8(4):757–771.

MohammadReza Davari and Eugene Belilovsky. 2024. Model breadcrumbs: Scaling multi-task model merging with sparse masks. In *Computer Vision - ECCV 2024 - 18th European Conference, Milan, Italy, September 29-October 4, 2024, Proceedings, Part LXXV*, volume 15133 of *Lecture Notes in Computer Science*, pages 270–287. Springer.

Pala Tej Deep, Rishabh Bhardwaj, and Soujanya Poria. 2024. Della-merging: Reducing interference in model merging through magnitude-based sampling. *CoRR*, abs/2406.11617.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 4171–4186. 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 preprint arXiv:2407.21783*.

Elias Frantar and Dan Alistarh. 2023. Sparsegpt: Massive language models can be accurately pruned in one-shot. In *International Conference on Machine Learning, ICML 2023, 23-29 July 2023, Honolulu, Hawaii, USA*, volume 202 of *Proceedings of Machine Learning Research*, pages 10323–10337. PMLR.

Zichuan Fu, Xiangyang Li, Chuhan Wu, Yichao Wang, Kuicai Dong, Xiangyu Zhao, Mengchen Zhao, Huifeng Guo, and Ruiming Tang. 2024. A unified framework for multi-domain ctr prediction via large language models. *ACM Trans. Inf. Syst.* Just Accepted.

Mahak Gambhir and Vishal Gupta. 2017. Recent automatic text summarization techniques: a survey. *Artif. Intell. Rev.*, 47(1):1–66.

Jinmin He, Kai Li, Yifan Zang, Haobo Fu, QIANG FU, Junliang Xing, and Jian Cheng. 2024. Efficient multi-task reinforcement learning with cross-task policy guidance. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*.

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*. OpenReview.net.

Di Jin, Eileen Pan, Nassim Oufattole, Wei-Hung Weng, Hanyi Fang, and Peter Szolovits. 2020. What disease does this patient have? A large-scale open domain question answering dataset from medical exams. *CoRR*, abs/2009.13081.

Xisen Jin, Xiang Ren, Daniel Preotiuc-Pietro, and Pengxiang Cheng. 2023. Dataless knowledge fusion by merging weights of language models. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net.

Yunxiang Li, Zihan Li, Kai Zhang, Ruilong Dan, Steve Jiang, and You Zhang. 2023. Chatdoctor: A medical chat model fine-tuned on a large language model meta-ai (llama) using medical domain knowledge. *Cureus*, 15(6).

Chin-Yew Lin. 2004. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain. Association for Computational Linguistics.

Junling Liu, Peilin Zhou, Yining Hua, Dading Chong, Zhongyu Tian, Andrew Liu, Helin Wang, Chenyu You, Zhenhua Guo, Lei Zhu, et al. 2023. Benchmarking large language models on cmexam–a comprehensive chinese medical exam dataset. *arXiv preprint arXiv:2306.03030*.

Qidong Liu, Xian Wu, Yejing Wang, Zijian Zhang, Feng Tian, Yefeng Zheng, and Xiangyu Zhao. 2024a. LLM-ESR: large language models enhancement for long-tailed sequential recommendation. In *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024*.

Qidong Liu, Xian Wu, Xiangyu Zhao, Yuanshao Zhu, Derong Xu, Feng Tian, and Yefeng Zheng. 2024b. When MOE meets llms: Parameter efficient fine-tuning for multi-task medical applications. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2024, Washington DC, USA, July 14-18, 2024*, pages 1104–1114. ACM.

Qidong Liu, Xian Wu, Xiangyu Zhao, Yuanshao Zhu, Zijian Zhang, Feng Tian, and Yefeng Zheng. 2024c. Large language model distilling medication recommendation model. *CoRR*, abs/2402.02803.

Pedro Henrique Martins, Patrick Fernandes, João Alves, Nuno M. Guerreiro, Ricardo Rei, Duarte M. Alves, José Pombal, Amin Farajian, Manuel Faysse, Mateusz Klimaszewski, Pierre Colombo, Barry Haddow, José G. C. de Souza, Alexandra Birch, and André F. T. Martins. 2024. Eurollm: Multilingual language models for europe. *Preprint*, arXiv:2409.16235.

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*.

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

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics, July 6-12, 2002, Philadelphia, PA, USA*, pages 311–318. ACL.

Sidak Pal Singh and Martin Jaggi. 2020. Model fusion via optimal transport. In *Advances in Neural Information Processing Systems*, volume 33, pages 22045–22055. Curran Associates, Inc.

Hongyan Tang, Junning Liu, Ming Zhao, and Xudong Gong. 2020. Progressive layered extraction (ple): A novel multi-task learning (mtl) model for personalized recommendations. In *Proceedings of the 14th ACM Conference on Recommender Systems*, RecSys '20, page 269–278, New York, NY, USA. Association for Computing Machinery.

Masatoshi Tsuchiya. 2018. Performance impact caused by hidden bias of training data for recognizing textual entailment. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan. European Language Resources Association (ELRA).

Haifeng Wang, Hua Wu, Zhongjun He, Liang Huang, and Kenneth Ward Church. 2022. Progress in machine translation. *Engineering*, 18:143–153.

Yejing Wang, Zhaocheng Du, Xiangyu Zhao, Bo Chen, Huifeng Guo, Ruiming Tang, and Zhenhua Dong. 2023a. Single-shot feature selection for multi-task recommendations. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2023, Taipei, Taiwan, July 23-27, 2023*, pages 341–351. ACM.

Yuhao Wang, Ha Tsz Lam, Yi Wong, Ziru Liu, Xiangyu Zhao, Yichao Wang, Bo Chen, Huifeng Guo, and Ruiming Tang. 2023b. Multi-task deep recommender systems: A survey. *Preprint*, arXiv:2302.03525.

Mitchell Wortsman, Gabriel Ilharco, Samir Yitzhak Gadre, Rebecca Roelofs, et al. 2022. Model soups: averaging weights of multiple fine-tuned models improves accuracy without increasing inference time. In *International Conference on Machine Learning, ICML 2022, 17-23 July 2022, Baltimore, Maryland, USA*, volume 162 of *Proceedings of Machine Learning Research*, pages 23965–23998. PMLR.

Derong Xu, Ziheng Zhang, Zhenxi Lin, Xian Wu, Zhihong Zhu, Tong Xu, Xiangyu Zhao, Yefeng Zheng, and Enhong Chen. 2024a. Multi-perspective improvement of knowledge graph completion with 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 11956–11968. ELRA and ICCL.

Derong Xu, Ziheng Zhang, Zhihong Zhu, Zhenxi Lin, Qidong Liu, Xian Wu, Tong Xu, Xiangyu Zhao, Yefeng Zheng, and Enhong Chen. 2024b. Mitigating hallucinations of large language models in medical information extraction via contrastive decoding. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 7744–7757, Miami, Florida, USA. Association for Computational Linguistics.

Derong Xu, Ziheng Zhang, Zhihong Zhu, Zhenxi Lin, Qidong Liu, Xian Wu, Tong Xu, et al. 2024c. Editing factual knowledge and explanatory ability of medical large language models. In *Proceedings of the 33rd ACM International Conference on Information and Knowledge Management, CIKM 2024, Boise, ID, USA, October 21-25, 2024*, pages 2660–2670. ACM.

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, NAACL-HLT 2021, Online, June 6-11, 2021*, pages 483–498. Association for Computational Linguistics.

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*.

An Yang, Baosong Yang, Binyuan Hui, et al. 2024a. Qwen2 technical report. *CoRR*, abs/2407.10671.

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

Enneng Yang, Zhenyi Wang, Li Shen, Shiwei Liu, Guibing Guo, Xingwei Wang, and Dacheng Tao. 2024c. Adamerging: Adaptive model merging for multi-task learning. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net.

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*. OpenReview.net.

Sheng Zhang, Xin Zhang, Hui Wang, Lixiang Guo, and Shanshan Liu. 2018. Multi-scale attentive interaction networks for chinese medical question answer selection. *IEEE Access*, 6:74061–74071.

Zijian Zhang, Shuchang Liu, Jiaao Yu, Qingpeng Cai, Xiangyu Zhao, Chunxu Zhang, Ziru Liu, Qidong Liu, Hongwei Zhao, Lantao Hu, Peng Jiang, and Kun Gai. 2024. M^3oe: Multi-domain multi-task mixture-of experts recommendation framework. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2024, Washington DC, USA, July 14-18, 2024*, pages 893–902. ACM.

Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, et al. 2024. A survey of large language models. *Preprint*, arXiv:2303.18223.

A Appendix

A.1 Experimental Settings

A.1.1 Baselines

In our experiments, we compare it against a comprehensive set of baseline methods, including traditional weighted averaging techniques and state-of-the-art approaches specifically developed for fine-tuned models.

- **Multilingual Multi-task Training** This approach trains a single model on the combined datasets of multiple languages simultaneously, without distinguishing between tasks.
- **Model Soups** (Wortsman et al., 2022) Uniform Soup is a simple merging method where the parameters of the fine-tuned models are averaged based on their importance.
- **Task Arithmetic** (Ilharco et al., 2023) This method performs arithmetic operations on the parameter differences between the pre-trained and fine-tuned models.
- **TIES** (Yadav et al., 2023) The Task Interference Elimination Strategy (TIES) minimize negative transfer and task interference by pruning redundant parameters and using a chosen sign to determine parameter update directions.
- **DARE** (Yu et al., 2024) Delta Alignment for Robust Ensemble (DARE) reduces the interference across tasks by randomly drop the delta vectors.
- **Model Breadcrumbs** (Davari and Belilovsky, 2024) This approach tracks and prunes maxima and minima in delta vectors to retain critical task-specific features.
- **DELLA** (Deep et al., 2024) DELLA follows DARE and assign drop rates to delta vectors according to their absolute values, improving performance stability.
- **OT-Fusion** (Singh and Jaggi, 2020) This method aligns and averages model weights via optimal transport, enabling one-shot parameter merging across heterogeneous models without retraining.
- **Layer Swapping** (Bandarkar et al., 2025) This approach composes task and language experts by directly replacing top and bottom transformer layers, facilitating cross-lingual transfer.

A.1.2 Implementation Details

For model adaptation, we applied LoRA to all linear networks in the model. The learning rate schedule was carefully designed with a 100-step warmup phase followed by cosine decay, which helped achieve stable convergence while maintaining optimal model performance. This configuration proved effective in balancing training efficiency and model quality across both multilingual and multi-task scenarios.

In addition to Qwen2-7B-Instruct, we also experimented with other foundation models including Llama-3-8B-Instruct (results shown in A.2). However, Qwen2-7B-Instruct demonstrated more consistent performance, particularly in handling both English and Chinese tasks, making it the preferred choice for our main experiments.

For visualization in Figure 4, we normalized the performance metrics to facilitate clear comparisons.

Table 8: Performance comparison of merging methods for bilingual QA tasks. Model A is fine-tuned on HealthCareMagic, Model B is fine-tuned on cMedQA2, Multi-task model is fine-tuned on both datasets. The overall best result is marked in bold and the best merging result is underlined.

Types	Methods	L1 (HealthCareMagic)				L2 (cMedQA2)				Avg Impr.	Avg Rank.
		BLEU-4	ROUGE-1	ROUGE-2	ROUGE-l	BLEU-4	ROUGE-1	ROUGE-2	ROUGE-l		
Pre-trained	Qwen2-7B-Instruct	30.1209	26.3524	5.3280	15.7451	1.7090	14.1527	1.7822	9.0934	-	-
Fine-tuned	Model A (L1)	35.5717	**30.2512**	**8.9044**	20.3625	3.7609	19.1370	3.1364	15.1441	+30.66%	6.875
	Model B (L2)	24.8587	24.9841	4.1492	15.1967	4.4159	21.2210	4.0680	17.4600	+11.57%	8.375
	Multi-task	35.7637	29.9781	8.6687	20.1184	3.7660	20.9869	3.7784	16.8850	+34.19%	6.125
Merged	Model Soups	33.2627	28.8258	7.5487	18.9459	4.6801	**21.5564**	4.0502	17.5380	+30.80%	6.125
	Task Arithmetic	33.0398	28.7169	7.5726	18.9600	4.7181	21.4108	4.0503	17.6772	+30.55%	5.625
	TIES	33.6571	29.0496	7.7769	19.1503	4.3751	20.8551	3.7518	17.1978	+30.23%	7.375
	DARE	33.3031	28.9575	7.8222	19.1702	**4.7578**	21.0865	3.8996	17.2488	+30.64%	5.625
	DARE+TIES	26.8091	26.0330	5.2307	16.5201	4.2456	20.6276	3.7531	17.1445	+15.41%	10.375
	Model Breadcrumbs	34.3247	29.4403	8.1518	19.6443	4.4092	20.9365	3.8138	17.1378	+32.19%	6.750
	DELLA	33.4207	28.9234	7.6728	18.9674	4.6827	21.1596	4.0709	17.4775	+30.77%	5.500
	DELLA+TIES	27.2331	26.1723	5.4339	16.6009	4.7130	21.2275	**4.2944**	**17.7694**	+18.37%	6.000
	Hi-Merging (Ours)	**35.9500**	29.9826	8.8738	**20.3844**	4.7009	21.1752	3.9704	17.2361	**+36.42%**	**3.250**

Table 9: Performance comparison of merging methods for multilingual QA using Llama-3-8B-Instruct. Model A is fine-tuned on HealthCareMagic (L1: English). Model B is fine-tuned on cMedQA2 (L2: Chinese). Multi-task model is fine-tuned on both datasets. The overall best result is marked in bold and the best merging result is underlined.

Types	Methods	L1 (HealthCareMagic)				L2 (cMedQA2)				Avg.	Impr.
		BLEU-4	ROUGE-1	ROUGE-2	ROUGE-L	BLEU-4	ROUGE-1	ROUGE-2	ROUGE-L		
Pre-trained	Llama-3-8B-Instruct	16.3118	21.6011	3.1389	10.8666	0.0225	0.4710	0.0211	0.2343	6.5834	-
Fine-tuned	Model A (L1)	**36.0325**	30.4111	**9.2743**	**20.7236**	0.0185	0.1288	0.0025	0.0841	12.0844	+83.5%
	Model B (L2)	3.9950	7.7673	0.9267	4.6358	3.0638	20.4016	3.4178	16.3067	7.5643	+14.9%
	Multi-task	35.6154	**30.5447**	9.2156	20.4271	3.0250	20.3136	**3.4964**	16.0911	**17.3411**	**+163.5%**
Merged	Model Soups	32.1199	28.2278	6.3715	18.2456	3.3256	19.5499	2.8670	15.4688	15.7720	+139.5%
	Task Arithmetic	31.6679	27.7646	6.0354	18.0448	3.3805	19.6475	2.9507	15.4806	15.6215	+137.2%
	TIES	32.1494	28.0527	6.7440	18.2913	3.2238	19.5369	2.8854	15.2112	15.7618	+139.4%
	DARE	25.9679	25.6716	4.5173	16.3803	3.5337	20.8586	3.1736	16.6716	14.5968	+121.7%
	DARE+TIES	26.6707	25.9106	5.2031	16.5525	3.2236	19.8564	2.9963	15.5967	14.5012	+120.2%
	Model Breadcrumbs	26.9844	26.1004	4.7037	16.3247	3.3307	20.7442	3.3069	16.2874	14.7228	+123.6%
	DELLA	25.6313	25.6792	4.5522	16.1313	**3.6612**	**20.9176**	3.3286	**16.7355**	14.5796	+121.4%
	DELLA+TIES	27.1246	26.0186	5.3163	16.6170	3.3433	19.9122	3.0848	15.9942	14.6764	+122.9%
	Hi-Merging (Ours)	33.5960	28.4141	7.2167	18.8804	3.1967	19.8207	2.9509	15.7833	16.2324	+146.5%

The performance values of the models on each dataset represent the average of the QA task metrics (BLEU-4, ROUGE-1, ROUGE-2, and ROUGE-L). We scaled the pre-trained Qwen2-7B-Instruct's performance to 20 and the better-performing fine-tuned model's performance to 80 for each task. The performance values of the other fine-tuned model and our merged model were then proportionally adjusted within this range to maintain their relative differences.

A.2 Bilingual Task Merging

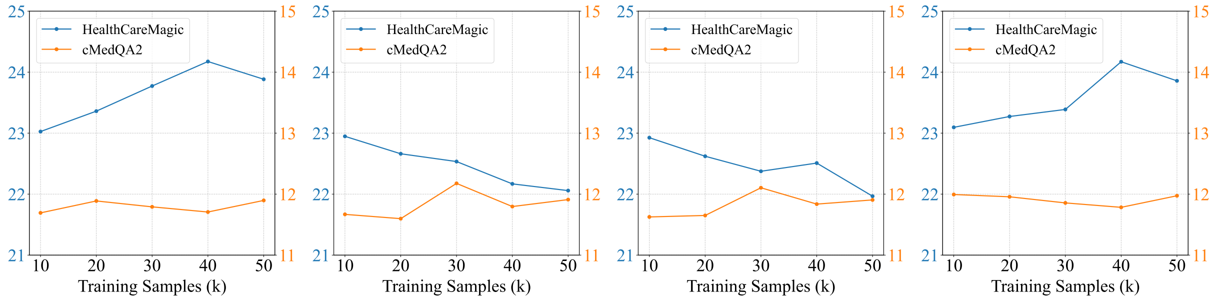
For merging LLMs that specialise in different languages on the same task, we further conduct experiments on the QA task (Table 8) and extend the foundation LLM to Llama-3-8B-Instruct, as presented in Table 11 and 9.

The results in Table 8 demonstrate the effectiveness of our approach in merging bilingual QA models. Hi-Merging achieves the best performance on English QA metrics (BLEU-4: 35.95, ROUGE-1: 29.98, ROUGE-2: 8.87, ROUGE-L: 20.38) while maintaining competitive performance on Chinese QA metrics. This balanced performance leads to the highest average improvement (+36.42%) and best average ranking (3.25) among all merging methods. Notably, while some baseline methods like DELLA+TIES achieve better performance on specific Chinese metrics, they significantly compromise English performance, highlighting our method's advantage in maintaining cross-lingual capabilities.

The results in Table 9 show that the performance of merged models based on Llama-3-8B-Instruct is generally inferior to that of the pre-merged fine-tuned models. This indicates that the effectiveness of the merging process is strongly influenced by the quality of the foundation models. The observed degradation in performance can be attributed to several factors. First, weaker foundation models, such as Llama-3-8B-Instruct, tend to produce delta vectors with more dispersed and less coherent param-

Table 10: Numerical performance of Hi-Merging on two open-source models, Echelon-AI/Med-Qwen2-7B and shtdbb/qwen2-7b-med.

Models	MedQA	MediQA	Medical Flashcards	Health Advice	Pubmed	WikiDoc	WikiDoc Patient	CORD 19	iCliniq	HealthCareMagic	ChatMed	MedChatZH
Qwen2-7B-Instruct	37.3868	17.3595	22.7668	2.8205	5.7994	17.6217	18.785	39.1748	19.3292	28.7051	9.9138	8.0654
Echelon-AI/Med-Qwen2-7B	64.2862	**32.052**	41.1081	97.7523	**92.9898**	20.7237	**26.9203**	40.7167	26.5593	30.3212	15.1218	9.2714
shtdbb/qwen2-7b-med	40.1598	27.1442	29.85	4.096	11.5017	20.5808	21.1528	**41.2026**	27.332	**33.3678**	19.4513	11.2665
Hi-Merging (Ours)	**64.9011**	31.9421	**45.1714**	**97.755**	92.1692	**21.0211**	26.3293	40.9803	**28.7816**	31.6779	**19.8074**	**11.2958**



(a) Fine-tuned models (b) Model Soups (c) Task Arithmetic (d) Our method

Figure 6: Impact of training sample size on model merging conflicts. Blue and orange lines represent the average performance metrics for HealthCareMagic and cMedQA2, respectively.

Table 11: Performance comparison of merging methods for bilingual MCQA using Llama-3-8B-Instruct. Model A is fine-tuned on MedQA (L1: English). Model B is fine-tuned on CMExam (L2: Chinese), Multi-task model is fine-tuned on both datasets. Overall best result is in bold and the best merging result is underlined.

Types	Methods	L1 (MedQA)	L2 (CMExam)	Avg.	Impr.
Pre-trained	Llama-3-8B-Instruct	57.9733	17.2821	37.6277	+0.00%
	GLM-4-9B	54.7656	69.5194	62.1425	**+65.15%**
	Gemma-2-9B	14.2583	2.7698	8.5141	-77.37%
Fine-tuned	Model A (L1)	60.4688	52.2706	56.3697	+49.81%
	Model B (L2)	60.3906	60.5525	61.0575	+62.27%
	Multi-task	**62.8906**	61.0356	**61.9631**	+64.70%
Merged	Model Soups	61.2500	61.0507	61.1504	+62.54%
	Task Arithmetic	61.2500	61.8750	61.5625	+63.65%
	TIES	61.7188	61.3225	61.5207	+63.56%
	DARE	61.5625	61.3678	61.4652	+63.42%
	DARE + TIES	60.9375	59.4656	60.2016	+60.05%
	Model Breadcrumbs	61.0156	60.4318	60.7237	+61.43%
	DELLA	60.8594	60.7186	60.7890	+61.58%
	DELLA + TIES	61.9531	61.3527	61.6529	+63.91%
	Hi-Merging (Ours)	62.2656	61.0757	61.6707	+63.96%

eter distributions during fine-tuning. These delta vectors often carry noisy or conflicting information, which makes the merging process prone to parameter conflicts. Second, the weaker representational capacity of these models limits their ability to encode robust and semantically aligned knowledge, further exacerbating the challenges of merging.

A.3 Number of training samples

We examine the impact of varying the number of training samples on the conflict during model merging, as shown in Figure 6. In the experiment, we use two QA datasets, HealthCareMagic (English) and cMedQA2 (Chinese), sampling 10k, 20k, 30k, 40k, and 50k training examples from each to produce a series of fine-tuned models, five per dataset. This setup evaluates how the number of training samples influences both individual model performance and compatibility during merging. The x-axis of Figure 6 represents the number of training samples, while the y-axis denotes the average performance metrics, including BLEU-4, ROUGE-1, ROUGE-2, and ROUGE-L.

However, Figures 6b and 6c show that merged models through either Model Soups or Task Arithmetic suffer from performance drops driven by the increasing size of training sample as further training leads to conflicting highly specialized models. Figure 6d shows the opposite: our method retains performance trends in line with fine-tuned models and addresses conflicts to retain improving performance through larger training sets.

These results highlight the robustness of our method in resolving merging conflicts, ensuring that the merged models retain the strengths of individual models while achieving stable and superior performance across training sample sizes.

A.4 Open Source LLM Merging

Table 10 presents the detailed numerical results for all models across the 12 medical datasets. The datasets cover a wide range of medical tasks and languages, allowing us to comprehensively evaluate the models' capabilities and the effectiveness of our merging approach.