arXiv:2411.16815v3 [cs.CV] 28 Jul 2025

# FREE-Merging: Fourier Transform for Efficient Model Merging

Shenghe Zheng
Harbin Institute of Technology
shenghez.zheng@gmail.com

Hongzhi Wang*
Harbin Institute of Technology
wangzh@hit.edu.cn

## Abstract

*With the rapid growth of deep learning, there is an increasing availability of open-source models for various tasks. However, single fine-tuned models often fall short of meeting the diverse needs of users. Model merging has thus emerged as an efficient method to integrate the capabilities of existing models into a unified model. Nevertheless, existing model merging methods face challenging trade-offs between performance and deployment costs, primarily due to task interference. For the first time, we reveal that task interference is evident in the frequency domain of model parameters, yet current efforts only focus on spatial domain solutions, which are largely ineffective in addressing frequency domain interference. To mitigate the impact of frequency domain interference, we propose* ***FR-Merging****, an innovative method that effectively filters harmful frequency domain interference on the backbone with minimal computational overhead. Since performance loss is inevitable with cost-free methods, we propose a lightweight task-specific expert module that dynamically compensates for information loss during merging. This proposed framework,* ***FREE-Merging*** *(FR-Merging with experts), strikes a balanced trade-off between training cost, inference latency, storage requirements, and performance. We demonstrate the effectiveness of both FR-Merging and FREE-Merging on multiple tasks across CV, NLP, and Multi-Modal domains and show that they can be flexibly adapted to specific needs. Our code is available at: https://github.com/Zhengsh123/FREE-Merging.*

## 1. Introduction

In the current era of deep learning, the pretrain-finetune paradigm is a standard procedure [9, 37, 40]. However, edge applications often lack powerful computing resources and data, favoring ready-to-use models for specific scenarios. The recent growth of open-source platforms like HuggingFace [43] has made this feasible. Yet, practical demands are often not fully covered by a single existing model and are typically a union of several models [23, 52]. However, running separate models for each subtask increases storage and inference costs, especially with large models. Model merging attempts to combine existing models to build a model capable of addressing all target tasks [21, 33, 48]. This method reduces training costs, data privacy issues in multi-task learning (MTL) [20, 56] and deployment costs, garnering widespread attention.

While model merging aims to reduce training costs, it often suffers from performance loss due to task interference [20]. To mitigate this issue, existing methods develop along three directions. The first direction computes merging coefficients based on weights or data [23, 33]. The second uses the parameter differences before and after fine-tuning (*i.e.* task vectors) for merging [21, 48, 52, 58]. The third method introduces independent knowledge for each task [20, 29]. While the third way balances performance and costs, it still faces two critical limitations. First, it neglects backbone optimization during merging, resulting in subpar performance. Second, it requires high computation and storage, making it impractical for edge deployment.

We argue that an effective model merging method requires addressing both of the above issues. First, since the backbone constitutes the majority of parameters and thus determines performance, an efficient backbone merging method is essential to reduce performance loss. Then, given the *no free lunch* theorem, cost-free merging cannot avoid task interference. Therefore, introducing task-specific experts is necessary. However, they must remain lightweight to avoid storage and inference overhead. Thus, we propose FREE-Merging, a two-stage method that ensures an efficient backbone and lightweight experts as shown in Fig 1.

For backbone merging, the key lies in eliminating task interference [51]. We identify a compelling phenomenon in which task interference is evident in the frequency domain of model parameters. And we present in Sec. 4 that frequency-domain interference is strongly correlated with performance. Notably, existing methods, which focus only on spatial-domain operations, fail to address frequency-domain interference, leading to suboptimal outcomes. Ex-

*Corresponding Author

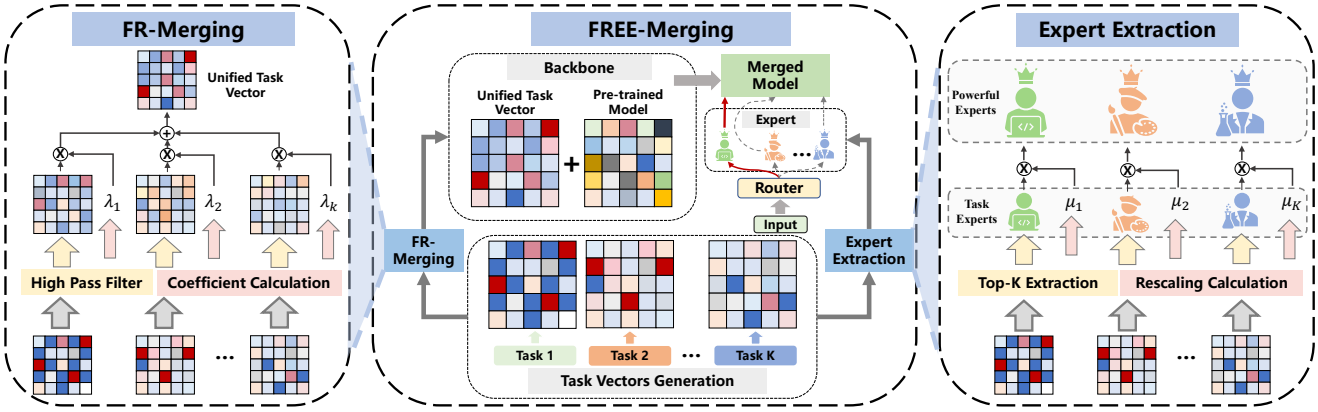



Figure 1. The workflow of FREE-Merging involves two main steps. First, FR-Merging utilizes high-pass filtering to remove harmful specialized information from each model, thereby constructing a high-quality merged backbone. Second, it employs lightweight extraction of task experts, which are dynamically added during inference to mitigate the impact of task interference.

perimental evidence reveals that task interference is severe in the low-frequency region. This occurs because, high-frequency signals represent fine-grained variations, while low-frequency parts capture the global structure [18], which is more likely to contain task-specific information that leads to task interference. Leveraging the inherent redundancy in fine-tuned parameters [27, 47, 53], we propose to directly filter out low-frequency parts with severe task interference. As detailed in Sec. 5.2, this high-pass filtering substantially enhances generalization with minimal performance loss, preventing task interference during merging. Our lightweight yet highly effective approach, FR-Merging, is the first to apply Fourier filtering to neural network parameters, showing its potential for broader applications.

After obtaining a high-performance backbone through FR-Merging, we use a lightweight task-specific expert to recover the inevitable information loss with minimal storage. Unlike existing methods that store extensive knowledge, we rescale low-frequency signals, which store task-specific information, retaining task expertise with only 1% of the parameters. Inspired by MoE [39, 57], we employ a router for dynamic expert routing, enhancing model flexibility.

In summary, we introduce FREE-Merging, which utilizes FR-Merging to reduce task interference and incorporates lightweight experts to eliminate information loss of model merging. Our main contributions are: 1). We identify frequency-domain task interference as a critical factor that greatly impacts merging performance but remains challenging to address through spatial-domain methods. 2). To tackle frequency-domain interference, we propose FR-Merging, which reduces interference while preserving performance, constructing the merged backbone. 3). We theoretically validate the necessity of introducing new information during merging and propose an effective expert construction method. 4). Extensive experiments show that FR-Merging and FREE-Merging effectively enhance performance across vision, language, and multi-modal tasks.

## 2. Related Work

**Model Merging.** Model merging integrates existing models to handle multiple tasks without training [21, 23, 49], but faces task interference challenges [36, 48]. Simple averaging [44] causes great performance degradation. Methods like Fisher-Merging [33] and RegMean [23] use matrices to determine merging coefficients but are computationally expensive or data-intensive, limiting edge deployment. Task Arithmetic [21] merges task vectors instead of weights, while Ties-Merging [48] resolves parameter conflicts, and AdaMerging [52] automates coefficient selection. However, these approaches have data distribution requirements and limited applicability. Techniques such as DARE [53] and PCB-Merging [12] mitigate performance drops but still face task conflicts. EMR-Merging [20] and Twin-Merging [29] store task-specific knowledge but require large storage and neglect backbone optimization. Our method uses high-pass filtering and lightweight experts to balance merging costs and effectiveness across various tasks. For further details, please refer to Appendix C.2.

**Model Ensemble.** Model ensemble combines outputs from multiple models [11, 15, 22], but with large models, it faces storage and inference challenges. In contrast, Model merging enables a single model to solve multiple tasks, significantly reducing storage and inference costs.

**Multi-task Learning.** Multi-task learning (MTL) aims to solve multiple tasks using a single model [4, 28, 41], but in the era of large models, it faces challenges like high training costs, data privacy issues, and expertise requirements [52]. In contrast, model merging uses existing open-source models to create a multi-task model with little or no training, significantly reducing deployment costs.

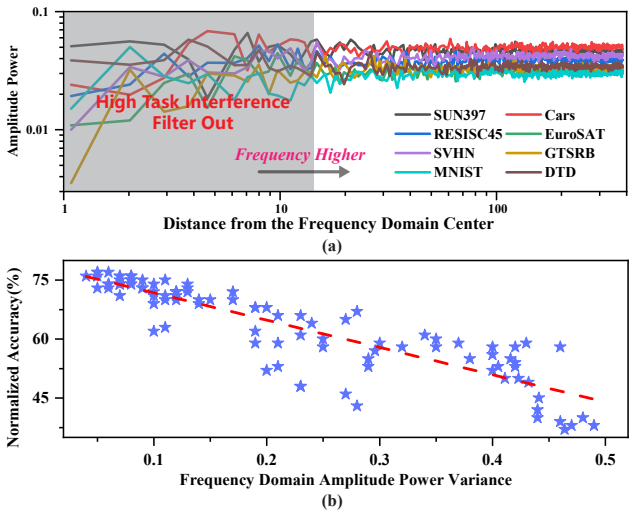



Figure 2. a) The frequency domain amplitude power distribution of an attention head across different fine-tuned ViT-B-32 models. b) The correlation between the merging performance of ViT-B-32 and the variance of frequency domain amplitude power. Normalized accuracy is defined as the ratio of merged model performance to fine-tuned model performance on the same task.

## 3. Preliminaries

**Notation.** Assume that $f_\theta(x) \to y$ represents a neural network parameterized by $\theta$, where $x \in R^m$ is the input and $y \in R^n$ is the output. Recent studies have focused on model merging via task vectors. For task $k$, the task vector is defined as $v_k = \theta_k - \theta_{pre}$, where $\theta_k$ is the fine-tuned model for task $k$, and $\theta_{pre}$ is the corresponding pre-trained model. Additionally, $e_k$ represents the task expert that retains the ability to solve the task $k$ with few parameters.

**Problem Definition.** The objective of model merging is to combine models $\{\theta_k\}_{k=1}^K$ for tasks $T = \{t_i\}_{i=1}^K$ into a single model $\theta_m$ that can handle all tasks in $T$. We focus on a practical scenario where a shared pre-trained model $\theta_{pre}$ serves as the base for all fine-tuned models. It is formulated as $\theta_m = \lambda \sum_{k=1}^K \hat{G}(\theta_k)$, and with task vector, it becomes $\theta_m = \theta_{pre} + \lambda \Sigma_{k=1}^K G(v_k)$, where $\lambda$ is the coefficient, and $\hat{G}/G$ is transformation on the parameters. Assuming a shared pre-trained model aligns with trends toward foundation models, like LLaMa [40] for language and CLIP [37] for vision, which are then fine-tuned for specific tasks.

## 4. Task Interference in Frequency Domain

Task interference is a key factor that leads to the decrease in merging performance. While current methods focus on addressing interference in the spatial domain [48, 50, 53], we observe that task vectors from different fine-tuned models also exhibit great misalignment in the frequency domain.

As shown in Fig. 2(a), we visualize the amplitude power distribution in the frequency domain of task vectors from a ViT-B-32 attention head fine-tuned on various tasks [21]. Here, greater distance from the center represents a higher frequency. Since amplitude indicates signal strength, merging two signals with vastly different strengths inevitably results in the weaker signal being masked. Therefore, task interference in the low-frequency part are severe.

Furthermore, in Fig.2(b), we train multiple groups of models on eight image tasks [21] and find a clear negative correlation between the variance of amplitude power among models and the normalized merging performance (details can be found in Appendix D.1), confirming the great impact of frequency-domain interference on merging performance.

Additionally, as shown in Table 1, we show that existing works addressing task interference in the spatial domain fail to effectively mitigate frequency-domain task interference. Therefore, we propose our approach to tackle this issue.

Table 1. The impact of current methods on the mean variance of frequency-domain amplitude across 8 image classification tasks.

| Method | ViT-B/32 | ViT-L/14 |
|---|---|---|
| Task Arithmetic [21] | 0.059 | 0.110 |
| DARE [53] | 0.057(↓3%) | 0.108(↓2%) |
| Ties-Merging [48] | 0.058(↓2%) | 0.109(↓1%) |
| Breadcrumbs [7] | 0.058(↓2%) | 0.107(↓3%) |
| PCB-Merging [12] | 0.056(↓5%) | 0.107(↓3%) |
| FR-Merging(ours) | **0.045(↓24%)** | **0.088(↓20%)** |

## 5. Methodology

This section presents FREE-Merging: Sec. 5.1 explains the need for a two-part design, Sec. 5.2 and Sec. 5.3 cover the FR-Merging for the backbone and the lightweight expert module, respectively, followed by the FREE-Merging in Sec. 5.4 and a complexity analysis in Sec. 5.5.

### 5.1. Motivation

We first analyze why both merging optimization for the backbone and the introduction of experts are necessary.

First, backbone optimization is crucial but overlooked by current expert-based merging ways. As the backbone holds most parameters, its performance is key to the overall success. For instance, upgrading from LLaMa2 [40] to LLaMa3 [13] greatly improves results even with the identical fine-tuning. Thus, we propose a training-free merging method to reduce task interference in the backbone.

Next, we present a theorem showing the necessity of task experts. Based on Theorem 5.1, introducing additional information is essential. For proof, see Appendix B, and the effectiveness of task experts is proved in Appendix D.4.

**Theorem 5.1** ***Model Merging has no free lunch.*** *Merged model $\theta_m$ cannot simultaneously retain the capabilities of $\{\theta_k\}_{k=1}^K$ without introducing additional information.*

### 5.2. FR-Merging

We propose an effective backbone merging method by addressing task interference in the low-frequency fine-tuned parameters as discussed in Sec. 4, which is hard to resolve in the spatial domain. Given the high redundancy of fine-tuned parameters [53], we apply high-pass filtering to directly remove the interfering regions. Let $v(x, y)$ be the task vector and $G(x, y)$ the transformed result. Then we have:

$$G(x, y) = \mathcal{F}^{-1}\{H(\eta, \gamma) \cdot \mathcal{F}\{v(x, y)\}\}, \tag{1}$$

where

$$H(\eta, \gamma) = \begin{cases} 1, & \sqrt{\eta^2 + \gamma^2} \geq D_0 \\ 0, & \sqrt{\eta^2 + \gamma^2} < D_0 \end{cases}, \tag{2}$$

where $\mathcal{F}$ and $\mathcal{F}^{-1}$ represent the Fourier and inverse Fourier transform, respectively. $D_0$ is the adjustable cutoff frequency. And $\eta$ and $\gamma$ are the distances from the frequency domain center along the $x$- and $y$-axes.

Next, we analyze why high-pass filtering is effective from both intuitive and experimental perspectives. Fine-tuned weights occupy distinct positions in the loss landscape, and linear interpolation often leads to high-loss regions, causing task interference [1, 50]. To address this, we minimize model differences while preserving performance, bringing them closer in the loss landscape to increase the likelihood of the merged model falling into a loss basin [45].

Results in Sec. 4 show great differences in low-frequency regions. This is because, if we consider neural network parameters as signals carrying training information, inspired by image processing [3, 35], high-frequency signals represent sharp variations, while low-frequency parts define the overall structure [5]. Since most information resides in the structure, filtering out low-frequency signals removes key task-specific information. Then, the likelihood that the interpolation falls within the loss basin increases [1].

Experimentally, we validate that high-pass filtering enhances model generalization with minimal performance loss as shown in Fig 3. We fine-tune ViT-B/16 [37] on 30 visual tasks following [20]. We find that removing low-frequency information results in a smaller performance drop (diagonal) compared to the improvement in generalization (off-diagonal). This indicates that low-frequency parts carry task-specific information that boosts task performance but reduces generalization. Thus, applying filtering like FR-Merging helps alleviate task interference during merging.

To maintain consistent output after merging, we define the merging coefficient $\lambda_i$ for the task vector $v_i$ as:

$$\lambda_i = \mathbb{E}(v_i)(\sum_{j=1}^{K} \mathbb{E}(v_j))^{-1}, \tag{3}$$

where $\mathbb{E}$ represents the average. Overall, we construct a merged backbone in this part that minimizes task conflicts.

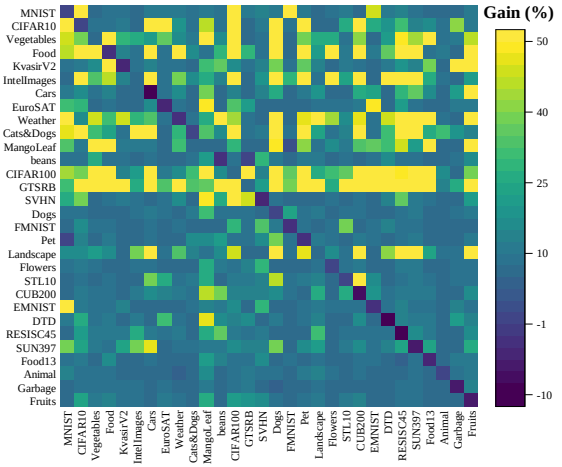



Figure 3. High-pass filtering improves performance over the unfiltered version. Rows represent the added task vectors, while columns are the test datasets. The results show a significant boost in generalization with only a slight decline in task performance.

### 5.3. Lightweight Expert Extraction

To compensate for information lost due to discarding low-frequency signals, we utilize lightweight experts added during inference. As analyzed in Sec. 5.2, low-frequency signals contain task-specific information, but preserving them requires an inefficient inverse Fourier transform during each inference. Thus, we propose a more efficient way.

Since fine-tuning leads to task specialization, the most changed parameters can be approximated as having the greatest impact [32]. After reasonable rescaling, these selected parameters serve as task experts, minimizing storage (typically around 1% of parameters) without extra inference computation. The validity is detailed in Appendix D.4.

Although magnitude-based extraction [48] exists, limited research on rescaling hinders compression. As rescaling is crucial to expert extraction, we propose a rescaling method. Let $M(v_i, d)$ represent the top-$d$ ($d$ is a percentage) parameters for task vector $v_i$, $\mu_i$ be the rescale factor, and $e(v_i)$ be the expert for task $t_i$. Then we have:

$$e(v_i) = \mu_i M(v_i, d), \ \mu_i = -\frac{\mathbb{E}(M(v_i, d)) \cdot \log(d)}{\lambda_i \cdot \mathbb{E}(v_i)}. \tag{4}$$

The goal is to maintain a consistent output after extraction to preserve performance. Details are in Appendix D.5.

In summary, we efficiently extract task-specific experts to integrate into the backbone, mitigating task conflicts without excessive storage or inference overhead.

### 5.4. FREE-Merging

This section presents FREE-Merging, combining both parts to make it suitable for multiple tasks without affecting inference speed. Inspired by MoE [39], a router dynamically assigns tasks to activated experts. The router can take

Table 2. Comparison of model merging performance using ViT-B/32 (B) and ViT-L/14 (L) on eight visual tasks evaluating by accuracy.

| Method | Base Model | SUN397 | | Cars | | RESISC45 | | EuroSAT | | SVHN | | GTSRB | | MNIST | | DTD | | Avg. | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | B | L | B | L | B | L | B | L | B | L | B | L | B | L | B | L | B | L |
| Individual | | 75.3 | 82.3 | 77.7 | 92.4 | 96.1 | 97.4 | 99.7 | 100 | 97.5 | 98.1 | 98.7 | 99.2 | 99.7 | 99.7 | 79.4 | 84.1 | 90.5 | 94.2 |
| Traditional MTL | | 73.9 | 80.8 | 74.4 | 90.6 | 93.9 | 96.3 | 98.2 | 96.3 | 95.8 | 97.6 | 98.9 | 99.1 | 99.5 | 99.6 | 77.9 | 84.4 | 88.9 | 93.5 |
| Weight Averaging | | 65.3 | 72.1 | 63.4 | 81.6 | 71.4 | 82.6 | 71.7 | 91.9 | 64.2 | 78.2 | 52.8 | 70.7 | 87.5 | 97.1 | 50.1 | 62.8 | 65.8 | 79.6 |
| Fisher Merging [33] | | **68.6** | 69.2 | **69.2** | **88.6** | 70.7 | 87.5 | 66.4 | 93.5 | 72.9 | 80.6 | 51.1 | 74.8 | 87.9 | 93.3 | 59.9 | 70.0 | 68.3 | 82.2 |
| Task Arithmetic [21] | | 63.8 | 74.1 | 62.1 | 82.1 | 72.0 | 86.7 | 77.6 | 93.8 | 74.4 | 87.9 | 65.1 | 86.8 | 94.0 | 98.9 | 52.2 | 65.6 | 70.1 | 84.5 |
| Ties-Merging [48] | | 64.8 | 76.5 | 62.9 | 85.0 | 74.3 | 89.3 | 78.9 | 95.7 | 83.1 | 90.3 | 71.4 | 83.3 | 97.6 | 99.0 | 56.2 | 68.8 | 73.6 | 86.0 |
| Breadcrumbs [7] | | 63.0 | 74.9 | 61.0 | 84.9 | 75.7 | 88.6 | 84.7 | 95.4 | 83.4 | 89.8 | 76.4 | 90.1 | 98.1 | 99.2 | 57.9 | 68.2 | 75.0 | 86.4 |
| PCB-Merging [12] | | 65.5 | 75.8 | 64.1 | 86.0 | 78.1 | 88.6 | 80.2 | 96.0 | 84.7 | 88.0 | 77.1 | 90.9 | 98.0 | 99.1 | 58.4 | 70.0 | 75.8 | 86.9 |
| FR-Merging(ours) | | 66.2 | **76.4** | 64.5 | 87.0 | **77.2** | **90.2** | **90.1** | **96.8** | **85.4** | **92.0** | **82.3** | **92.8** | **98.5** | **99.3** | **60.0** | **71.5** | **78.1** | **88.3** |
| AdaMerging++ [52] | | 66.6 | 79.4 | 68.3 | 90.3 | 82.2 | 91.6 | 94.2 | 97.4 | 89.6 | 93.4 | 89.0 | 97.5 | 98.3 | 99.0 | 60.6 | 79.2 | 81.1 | 91.0 |
| Twin-Merging [29] | | 73.6 | 82.6 | 71.7 | 90.3 | 92.1 | 94.9 | 99.3 | 99.4 | 95.3 | 96.7 | 97.2 | 98.0 | 99.1 | 99.4 | 74.0 | 80.1 | 87.8 | 92.7 |
| EMR-Merging [20] | | 74.1 | 82.3 | 72.7 | 90.8 | 91.9 | 95.3 | 99.4 | 99.5 | 95.8 | 96.9 | 96.9 | 98.1 | 99.1 | 99.5 | 72.1 | 80.1 | 87.7 | 92.8 |
| FREE-Merging(ours) | | **77.1** | **83.5** | **78.2** | **92.4** | **93.4** | **96.2** | **99.5** | **99.6** | **96.3** | **98.1** | **98.2** | **98.7** | **99.5** | **99.7** | **75.4** | **81.7** | **89.7** | **93.7** |

**Algorithm 1** Workflow of FREE-Merging

**Input:** Task vectors $\{v_i\}_{i=1}^K$, pre-trained model $\theta_{pre}$, router $R$, and input data $X$.

1: **FR-Merging:** ▷ Only excute once.
2: Get backbone $\theta_m = \theta_{pre} + \sum_{i=1}^K \lambda_i G(v_i)$, where $G$ and $\lambda$ are defined by Eq. 1 and Eq. 3, respectively.
3: **Expert Extraction:** ▷ Only excute once.
4: Get task experts $\{e_i\}_{i=1}^K$ as Eq. 4.
5: **Inference:**
6: **for** $x \in X$ **do**
7: $[w_1, ...w_K] \leftarrow argmax(R(x))$
8: $\theta_* = \theta_m + \sum_{i=1}^K w_i e_i$
9: $Y \leftarrow Y \cup f(x; \theta_*)$
10: **end for**

**Output:** Output $Y$ for input $X$.

various forms [24], allowing for more flexibility in handling different tasks. Detailed discussions about router can be found in Appendix D.6. For detailed algorithms, see Alg 1.

First, we obtain the merged backbone (line 2) and the task experts (line 4). Then, during the inference, we dynamically route the experts based on inputs (lines 7-9).

### 5.5. Time Complexity and Storage Space Analysis

This section analyzes the merging time complexity and storage requirements. We aim to minimize both for practical use. Assume there are $n$ models to merge, each with $m$ parameters and requiring storage space $s$.

The time complexity includes the merging and inference part. In the merging part, FR-Merging has a complexity of $O(nmlogm)$ [14] while expert extraction is $O(nm)$, making the overall merging complexity $O(nmlogm)$, which is manageable. For instance, merging three 10B models requires only $O(10^{12})$ addition operations, a minor load for modern devices. Moreover, with only a lightweight router added, the extra inference cost is negligible.

Next, we analyze storage cost. We typically save only about 1% additional parameters per model, requiring $(1 + 0.01n)s$ storage, which is far less than the requirement of $ns$ for the model ensemble. For large models, reducing storage requirements is crucial for alleviating storage pressure on edge users. For detailed analysis, please refer to Sec 6.7.

## 6. Experiments

In this section, we evaluate the proposed methods across various tasks. We compare FR-Merging with other cost-free methods (which do not require additional training or data), including Weight Averaging, Fisher Merging [33], Task Arithmetic [21], Ties-Merging [48], Breadcrumbs [7], and PCB-Merging [12]. We also compare our FREE-Merging with popular methods, including RegMean [23], AdaMerging [52], EMR-Merging [20], and Twin-Merging [29]. The original EMR-Merging uses a perfect router; however, we use an imperfect router for realism. Some experiments include traditional MTL with data from all tasks as a baseline. The evaluation covers CV, NLP, and multi-modal tasks.

### 6.1. Merging Vision Models

**Merging 8 ViTs.** We select the two visual encoders of CLIP, ViT-B/32 and ViT-L/14 [37], as the pre-trained models and choose eight image classification tasks as target tasks: SUN397, Cars, RESISC45, EuroSAT, SVHN, GTSRB, MNIST, and DTD [52]. Additionally, an MLP is used as the lightweight router. Detailed information on the datasets and baselines can be found in the Appendix C.1.

Table 2 presents the comparative results. First, FR-Merging outperforms other cost-free methods, improving ViT-B/32 by 2.5% and ViT-L/14 by 1.5% over PCB-Merging [12], demonstrating its effectiveness in reducing performance loss caused by task interference. Through FR-Merging, we can construct a merged backbone that can be used independently at a low cost. Here, we focus on the basic FR-Merging. In Sec. 6.6, we discuss further im-

Table 3. Comparison of model merging results using ViT-B/16 on 30 visual tasks. *Additional Cost* indicates whether additional training or additional data is required during model merging.

| Method | Additional Cost | Avg. Acc |
|---|---|---|
| Individual | ✗ | 93.02 |
| Weight Averaging | ✗ | 42.51 |
| Task Arithmetic [21] | ✗ | 48.88 |
| Ties-Merging [48] | ✗ | 37.53 |
| Breadcrumbs [7] | ✗ | 43.57 |
| FR-Merging(ours) | ✗ | **53.90** |
| RegMean [23] | ✓ | 68.13 |
| AdaMerging [52] | ✓ | 60.24 |
| Twin-Merging [29] | ✓ | 75.52 |
| EMR-Merging [20] | ✓ | 76.39 |
| FREE-Merging(ours) | ✓ | **79.67** |

Table 4. Language model merging results: RoBERTa fine-tuned on 8 discriminative tasks, T0-3B on 11 discriminative tasks with $IA^3$, and Qwen-14B on 3 generative tasks with LoRA.

| Method | Base Model | **Full** | **PEFT** | | **Avg.** |
|---|---|---|---|---|---|
| | | RoBERTa | T0-3B | Qwen-14B | |
| Individual | | 85.55 | 71.35 | 72.07 | 76.32 |
| Weight Averaging | | 51.34 | 58.01 | 66.80 | 58.71 |
| Task Arithmetic[21] | | 66.65 | 63.91 | 66.40 | 65.65 |
| Ties-Merging [48] | | 64.04 | 66.41 | 67.46 | 65.97 |
| Breadcrumbs [7] | | 68.19 | 54.64 | 66.77 | 63.20 |
| PCB-Merging [12] | | 67.86 | 66.10 | 66.69 | 66.39 |
| FR-Merging(ours) | | **70.02** | **66.88** | **68.00** | **68.30** |
| RegMean [23] | | 70.01 | 58.01 | 67.52 | 65.18 |
| EMR-Merging [20] | | 74.20 | 67.11 | 70.98 | 70.76 |
| Twin-Merging [29] | | 78.29 | 66.70 | 71.68 | 72.22 |
| FREE-Merging(ours) | | **80.16** | **68.68** | **72.78** | **73.87** |

provements achieved by combining it with existing methods. Then, compared to methods requiring extra information, FREE-Merging also improves performance by 2%, leveraging lightweight computation and efficient expert extraction, requiring only 1% additional storage per model, outperforming 3% of EMR-Merging [20] and 2% of Twin-Merging [29]. Thus, FREE-Merging provides an excellent balance between storage, computation, and performance.

**Merging 30 ViTs.** We select ViT-B/16 as the pre-trained model and evaluate the model merging performance across 30 image classification tasks following [20].

Table 3 presents the results, with detailed analysis provided in Appendix E.1. It is evident that our proposed FR-Merging and FREE-Merging exhibit remarkable performance. Although merging a large number of models is uncommon in practice, this experiment underscores the effectiveness of Fourier filtering and lightweight task experts in mitigating performance loss caused by task interference. These results highlight the robustness and broad applicability of our methods across diverse scenarios.

Table 5. Comparison of performance of different model merging methods after full fine-tuning LLaMa2-13B on three tasks: Instruction Following (LM), Math, and Code Generation.

| Method | AlpacaEval | GSM8K | MBPP | Avg. |
|---|---|---|---|---|
| LM Fine-tuned [46] | 88.9 | 45.9 | 31.4 | 55.4 |
| Math Fine-tuned [30] | 22.3 | 63.5 | 23.0 | 36.5 |
| Code Fine-tuned [31] | 16.2 | 18.8 | 27.0 | 20.7 |
| Task Arithmetic [21] | 72.1 | 48.3 | 0 | 40.1 |
| TIES-Merging [48] | 77.5 | 66.9 | 27.2 | 57.2 |
| DARE [53] | 77.5 | 62.7 | 29.4 | 56.5 |
| DELLA-Merging [8] | 80.4 | 61.8 | 31.4 | 57.9 |
| PCB-Merging [12] | 81.2 | 60.6 | 30.7 | 57.5 |
| FR-Merging(ours) | **87.5** | **64.3** | **32.8** | **61.5** |
| Twin-Merging [29] | 83.5 | 61.4 | 30.4 | 58.4 |
| EMR-Merging [20] | 82.6 | 62.3 | 31.2 | 58.7 |
| FREE-Merging(ours) | **88.1** | **65.7** | **31.8** | **61.8** |

## 6.2. Merging Language Models

**Merging Medium-sized Language Models.** We investigate the generalization ability of our method on medium-scale language models, using RoBERTa [59] as the pre-trained model with full fine-tuning on eight discriminative tasks [48]. Details can be found in Appendix C.1.

The average merging results of the eight tasks are presented in Table 4, with complete results in Appendix E.2. It can be observed that, FR-Merging outperforms other cost-free methods by nearly 2%. Additionally, FREE-Merging also exhibits remarkable effectiveness, achieving optimal results on almost all datasets and surpassing current methods in average performance. This highlights the great potential of our proposed methods for language models.

**Merging PEFT models.** PEFT is a popular method that reduces fine-tuning costs [10], but storing PEFT parameters for multiple tasks remains space-inefficient [55]. The effectiveness of our approach on PEFT demonstrates the generalization capability and offers an efficient deployment way.

Firstly, we use T0-3B [38] as the pre-trained model and $(IA)^3$ [26] as the PEFT method, fine-tuning on 11 discriminative tasks [20]. Our proposed FR-Merging and FREE-Merging improve performance by nearly 1%, as shown in Table 4. Next, we employ Qwen-14B [2] with LoRA [19] as the PEFT way, fine-tuning on three generative tasks [29]. The results show that our methods lead to a 2% advantage in both cost-free and full frameworks. This confirms the feasibility with PEFT and strong generalization of our methods. Full results are available in the Appendix E.2.

**Merging Large Language models.** We explore large model merging using LLaMa2-13B [40] as the pre-trained model, with WizardLM [46] for instruction following, WizardMath [30] for math, and WizardCoder-Python [31] for coding. Experiments are conducted on AlpacaEval (instruction following), GSM8K (math), and MBPP (code) [53].

Table 5 shows that our proposed FR-Merging even surpasses existing methods that rely on additional information.

Table 6. Results of merging multi-modal BEiT3 models on five vision-language tasks.

| Methods | Task / Metric | COCO-Retrieval Accuracy(↑) | COCO-Captioning BLEU4(↑) | COCO-Captioning CIDEr(↑) | COCO-Captioning METEOR(↑) | COCO-Captioning ROUGE-L(↑) | ImageNet-1k Classification Accuracy(↑) | NLVR2 Accuracy(↑) | VQAv2 Accuracy(↑) |
|---|---|---|---|---|---|---|---|---|---|
| Individual | | 0.8456 | 0.394 | 1.337 | 0.311 | 0.601 | 0.8537 | 0.7765 | 0.8439 |
| Weight Averaging | | 0.1893 | 0.031 | 0.001 | 0.115 | 0.159 | 0.6771 | 0.2800 | 0.6285 |
| Task Arithmetic [21] | | 0.3177 | **0.033** | 0.000 | **0.118** | **0.176** | 0.6732 | 0.3809 | 0.6933 |
| Ties-Merging [48] | | 0.3929 | 0.029 | 0.001 | 0.108 | 0.167 | 0.6978 | 0.3206 | 0.6717 |
| Breadcrumbs [7] | | 0.1893 | 0.011 | 0.012 | 0.065 | 0.114 | 0.6323 | 0.6073 | 0.6823 |
| FR-Merging(ours) | | **0.7010** | 0.016 | **0.019** | 0.096 | 0.141 | **0.6992** | **0.6407** | **0.6965** |
| EMR-Merging [20] | | 0.7946 | 0.289 | 1.060 | 0.272 | 0.534 | 0.7742 | 0.7475 | 0.7211 |
| Twin-Merging [29] | | 0.7937 | 0.325 | 1.078 | 0.278 | 0.558 | 0.7730 | 0.7762 | 0.7232 |
| FREE-Merging(ours) | | **0.8093** | **0.347** | **1.195** | **0.289** | **0.572** | **0.7850** | **0.8168** | **0.7297** |

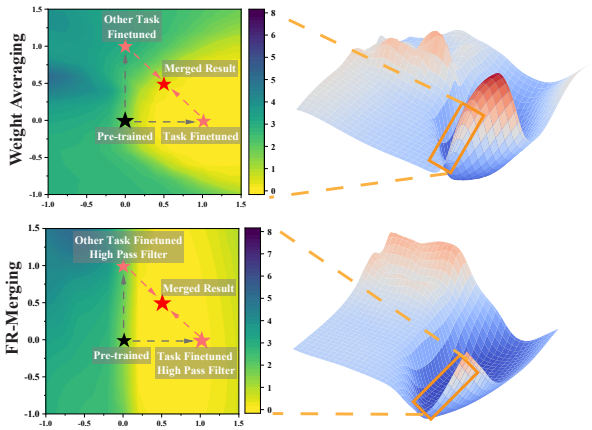



Figure 4. Visualization of the loss landscape on EuroSAT before and after high-pass filtering (top vs. bottom) in the model merging of the fine-tuned ViT-B/32 on EuroSAT and DTD tasks [21].

This indicates that our findings in FR-Merging regarding low-frequency signals and task interference also hold when extended to large models. This is crucial for practical usage of large models. Additionally, our proposed FREE-Merging also demonstrates strong advantages, enabling efficient cross-domain deployment of large models.

### 6.3. Merging Multi-Modal Models

We merge various fine-tuned BEiT3-base [42] models across five tasks: ImageNet-1k (Classification), VQAv2 (Visual Question Answering), NLVR2 (Reasoning), COCO Captioning (Image Captioning), and COCO Retrieval (Image-Text Retrieval). COCO Captioning is evaluated with BLEU4, CIDEr, METEOR, and ROUGE-L, while the others use accuracy [20]. Details are in Appendix C.1.

Table 6 shows that our methods gain great improvements in multi-modal models. Unlike vision or language model merging, which often involve similar tasks, such as classification, the task similarity here is relatively low. The strong performance highlights the generalization ability of our methods, making them widely applicable.

Table 7. Ablation of FREE-Merging components on 8 visual tasks.

| FR-Merging | Equa.3 | Top-K Expert | Scaling | ViT-B/32 | ViT-B/16 | ViT-L/14 |
|---|---|---|---|---|---|---|
| ✗ | ✗ | ✗ | ✗ | 70.10 | 75.24 | 84.50 |
| ✔ | ✗ | ✗ | ✗ | 76.96 | 81.34 | 88.21 |
| ✗ | ✔ | ✗ | ✗ | 71.43 | 77.41 | 84.20 |
| ✔ | ✔ | ✗ | ✗ | 78.10 | 82.54 | 88.30 |
| ✗ | ✗ | ✔ | ✗ | 65.24 | 68.98 | 70.13 |
| ✔ | ✔ | ✔ | ✗ | 79.38 | 84.31 | 89.53 |
| ✗ | ✗ | ✔ | ✔ | 82.32 | 86.87 | 90.47 |
| ✔ | ✔ | ✔ | ✔ | **89.68** | **91.09** | **93.70** |

### 6.4. Fourier Transform Analysis

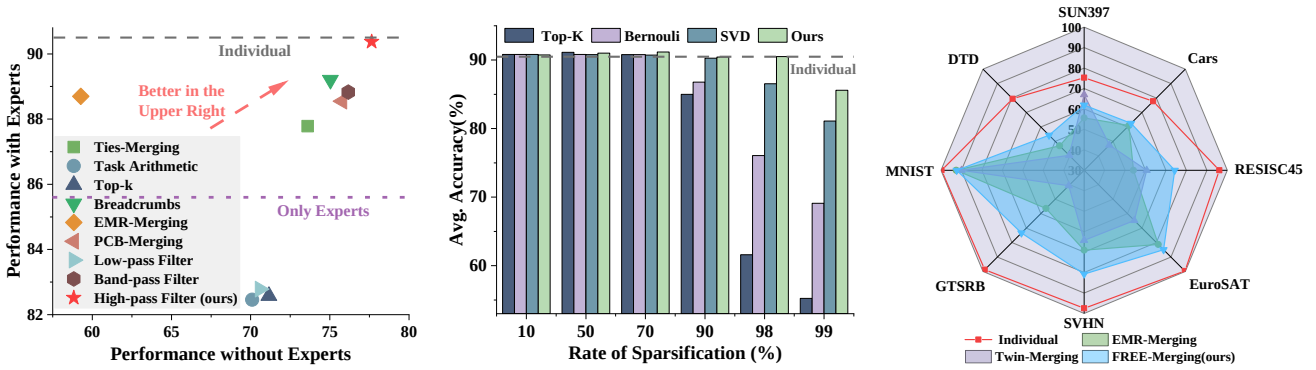
In this section, we explain why Fourier high-pass filtering improves merging performance from an optimization perspective. Previous work [25] shows that landscape geometry impacts model generalization. Fig. 4 demonstrates that high-pass filtering widens the valley in the loss landscape (bottom plot). Since fine-tuned models from the same pre-trained model are closer in the landscape [34], a wider valley increases the probability of merged results falling within the loss basin as shown in the bottom plot of Fig. 4, leading to better performance. Without filtering, the landscape is flat at a high level, which has little effect on model merging and only highlights the advantage of pre-training [17].

### 6.5. Ablation Study

In this section, we conduct ablation experiments on the components of our method. All experiments are performed on ViT-B/32 with eight classification tasks as target tasks.

**Ablation on FR-Merging.** We analyze the impact of FR-Merging on the backbone and overall performance. Table 7 shows that the performance of various models greatly improves after employing FR-Merging. We then compare high-pass filtering with other cost-free methods, including Ties [48], Top-K, Task Arithmetic [21], outlier dropout [7], EMR-Merging [20], and PCB-Merging [12], as well as low-pass and band-pass filtering, with results in Fig. 5a. For fairness, all experts use 1% additional parameters.

Our proposed FR-Merging outperforms other cost-free methods on both the backbone and with task experts, minimizing task interference. Examining the different filtering methods, we find that low-pass filtering leads to great per-




(a) Ablation on FR-Merging. (b) Ablation on Task Expert. (c) Router Analysis.

Figure 5. a) The ablation experiments on FR-Merging show that our proposed methods achieve optimal performance. b) Performance of experts at different sparsity rates. c) In experiments with random routers, our approach demonstrates optimal performance.

Table 8. Transferability analysis of FR-Merging on 8 visual tasks.

| Method | FR-Merging | ViT-B/32 | ViT-L/14 |
|---|---|---|---|
| Ties-Merging [48] | ✗ | 73.6 | 86.0 |
| Ties-Merging [48] | ✔ | **78.3** | **88.4** |
| Breadcrumbs [7] | ✗ | 75.0 | 86.4 |
| Breadcrumbs [7] | ✔ | **78.8** | **88.9** |
| TSVM [16] | ✗ | 83.9 | 91.9 |
| TSVM [16] | ✔ | **84.8** | **92.5** |

formance degradation, confirming that task interference in low-frequency regions is severe. Band-pass filtering also performs worse than FR-Merging, and this is because some high-frequency signals preserve abrupt changes in the signal and their removal harms the performance of the model.

**Ablation on Merging Coefficients.** We conduct an ablation study on our merging coefficients as in Eq. 3, with results in Table 7. Using them instead of simple averaging yields slight improvements by considering output variations, benefiting tasks with varying fine-tuning magnitudes.

**Ablation on Task Expert.** We conduct an ablation study on the expert extraction way. Table 7 shows that incorporating our experts and rescaling method results in significant improvements, confirming their effectiveness. To validate the superiority of our method, we evaluate expert-only performance with varying parameter proportions, comparing Top-K selection, Bernoulli selection [53], SVD extraction [29], and our Top-K selection with rescaling. Fig. 5b shows that our method excels, particularly under high sparsity.

**Router Analysis.** We analyze the proposed router method, aiming for strong merging performance despite sub-optimal routing. Since router inaccuracies may arise from limited training resources or edge deployment issues, our methods maintain performance in real-world scenarios. Using a random router to simulate inaccuracy, we compare our method with existing router-based approaches. Our method outperforms Twin-Merging by 13.5% and EMR-Merging by 8.5% on average, as shown in Fig. 5c, indicating that our methods construct a reasonable backbone and generalizable experts.

Table 9. Trade-off of performance, training, inference, and storage costs using ViT-L/14 across 8 image classification tasks.

| Method | Training Tokens | Training Cost | Storage Param. | Inference Cost (/1000 items) | Acc. |
|---|---|---|---|---|---|
| Individual | 0 | 0 | 2.4B | 31.0s | 94.2 |
| MTL | 304M | 12h | 304M | 31.0s | 93.5 |
| FR-Merging | 0 | 0 | 304M | 31.0s | 88.2 |
| Twin-Merging | 0.9M | 147s | 352M | 32.2s | 92.7 |
| FREE-Merging | 0.9M | 147s | 328M | 32.2s | 93.7 |

### 6.6. Transferability Analysis

Our proposed FR-Merging can integrate with other methods, as it addresses frequency-domain task interference that current methods overlook. As shown in Table 8, combining various methods with FR-Merging leads to performance improvements. Therefore, FR-Merging can be integrated with more approaches to achieve greater benefits.

### 6.7. Speed and Storage Space Analysis

This part analyzes training costs, inference speed, and storage as shown in Table 9. FR-Merging achieves strong performance with significantly reducing storage costs without extra training or inference costs, while FREE-Merging slightly increases overhead for better performance. Both of these two methods enhance deep learning deployment.

## 7. Conclusion

In this paper, we propose FR-Merging and FREE-Merging to efficiently build a multi-task model by merging existing ones. FR-Merging constructs a backbone without training by high-pass filtering to remove information that dominates task interference. Meanwhile, FREE-Merging employs lightweight experts to offset information loss during merging. Our method balances training, storage, inference costs, and performance, showing high practical value.

## Acknowledgements

This work is supported by National Natural Science Foundation of China (NSFC) (62232005, 62202126); the National Key Research and Development Program of China (2021YFB3300502).

## References

[1] Samuel Ainsworth, Jonathan Hayase, and Siddhartha Srinivasa. Git re-basin: Merging models modulo permutation symmetries. In *The Eleventh International Conference on Learning Representations*, 2023. 4

[2] Jinze Bai, Shuai Bai, Yunfei Chu, Zeyu Cui, Kai Dang, Xiaodong Deng, Yang Fan, Wenbin Ge, Yu Han, Fei Huang, et al. Qwen technical report. *arXiv preprint arXiv:2309.16609*, 2023. 6

[3] Tim-Oliver Buchholz and Florian Jug. Fourier image transformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1846–1854, 2022. 4

[4] Shijie Chen, Yu Zhang, and Qiang Yang. Multi-task learning in natural language processing: An overview. *ACM Computing Surveys*, 56(12):1–32, 2024. 2

[5] Tianyi Chu, Jiafu Chen, Jiakai Sun, Shuobin Lian, Zhizhong Wang, Zhiwen Zuo, Lei Zhao, Wei Xing, and Dongming Lu. Rethinking fast fourier convolution in image inpainting. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 23195–23205, 2023. 4

[6] Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Yunxuan Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, et al. Scaling instruction-finetuned language models. *Journal of Machine Learning Research*, 25(70):1–53, 2024. 3, 4

[7] MohammadReza Davari and Eugene Belilovsky. Model breadcrumbs: Scaling multi-task model merging with sparse masks. *arXiv preprint arXiv:2312.06795*, 2023. 3, 5, 6, 7, 8, 9, 10

[8] Pala Tej Deep, Rishabh Bhardwaj, and Soujanya Poria. Della-merging: Reducing interference in model merging through magnitude-based sampling. *arXiv preprint arXiv:2406.11617*, 2024. 6

[9] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 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, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota, 2019. Association for Computational Linguistics. 1

[10] Ning Ding, Yujia Qin, Guang Yang, Fuchao Wei, Zonghan Yang, Yusheng Su, Shengding Hu, Yulin Chen, Chi-Min Chan, Weize Chen, et al. Parameter-efficient fine-tuning of large-scale pre-trained language models. *Nature Machine Intelligence*, 5(3):220–235, 2023. 6

[11] Xibin Dong, Zhiwen Yu, Wenming Cao, Yifan Shi, and Qianli Ma. A survey on ensemble learning. *Frontiers of Computer Science*, 14:241–258, 2020. 2

[12] Guodong Du, Junlin Lee, Jing Li, Runhua Jiang, Yifei Guo, Shuyang Yu, Hanting Liu, Sim Kuan Goh, Ho-Kin Tang, Daojing He, et al. Parameter competition balancing for model merging. *arXiv preprint arXiv:2410.02396*, 2024. 2, 3, 5, 6, 7, 8, 9, 10

[13] Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024. 3

[14] Pierre Duhamel and Martin Vetterli. Fast fourier transforms: a tutorial review and a state of the art. *Signal processing*, 19 (4):259–299, 1990. 5

[15] Mudasir A Ganaie, Minghui Hu, Ashwani Kumar Malik, Muhammad Tanveer, and Ponnuthurai N Suganthan. Ensemble deep learning: A review. *Engineering Applications of Artificial Intelligence*, 115:105151, 2022. 2

[16] Antonio Andrea Gargiulo, Donato Crisostomi, Maria Sofia Bucarelli, Simone Scardapane, Fabrizio Silvestri, and Emanuele Rodolà. Task singular vectors: Reducing task interference in model merging. *arXiv preprint arXiv:2412.00081*, 2024. 8

[17] Timur Garipov, Pavel Izmailov, Dmitrii Podoprikhin, Dmitry P Vetrov, and Andrew G Wilson. Loss surfaces, mode connectivity, and fast ensembling of dnns. *Advances in neural information processing systems*, 31, 2018. 7

[18] Rafael C Gonzales and Paul Wintz. *Digital image processing*. Addison-Wesley Longman Publishing Co., Inc., 1987. 2

[19] Edward J Hu, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, Weizhu Chen, et al. Lora: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2022. 6

[20] Chenyu Huang, Peng Ye, Tao Chen, Tong He, Xiangyu Yue, and Wanli Ouyang. Emr-merging: Tuning-free high-performance model merging. *arXiv preprint arXiv:2405.17461*, 2024. 1, 2, 4, 5, 6, 7, 3, 8, 9, 10

[21] Gabriel Ilharco, Marco Tulio Ribeiro, Mitchell Wortsman, Suchin Gururangan, Ludwig Schmidt, Hannaneh Hajishirzi, and Ali Farhadi. Editing models with task arithmetic. *arXiv preprint arXiv:2212.04089*, 2022. 1, 2, 3, 5, 6, 7, 4, 8, 9, 10

[22] Dongfu Jiang, Xiang Ren, and Bill Yuchen Lin. Llm-blender: Ensembling large language models with pairwise ranking and generative fusion. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 14165–14178, 2023. 2

[23] Xisen Jin, Xiang Ren, Daniel Preotiuc-Pietro, and Pengxiang Cheng. Dataless knowledge fusion by merging weights of language models. *arXiv preprint arXiv:2212.09849*, 2022. 1, 2, 5, 6, 3, 8, 9, 10

[24] Dmitry Lepikhin, HyoukJoong Lee, Yuanzhong Xu, Dehao Chen, Orhan Firat, Yanping Huang, Maxim Krikun, Noam Shazeer, and Zhifeng Chen. Gshard: Scaling giant models with conditional computation and automatic sharding. In *International Conference on Learning Representations*, 2021. 5

[25] Hao Li, Zheng Xu, Gavin Taylor, Christoph Studer, and Tom Goldstein. Visualizing the loss landscape of neural nets. *Advances in neural information processing systems*, 31, 2018. 7

[26] Haokun Liu, Derek Tam, Mohammed Muqeeth, Jay Mohta, Tenghao Huang, Mohit Bansal, and Colin A Raffel. Few-shot parameter-efficient fine-tuning is better and cheaper than in-context learning. *Advances in Neural Information Processing Systems*, 35:1950–1965, 2022. 6

[27] James Liu, Guangxuan Xiao, Kai Li, Jason D Lee, Song Han, Tri Dao, and Tianle Cai. Bitdelta: Your fine-tune may only be worth one bit. *Advances in Neural Information Processing Systems*, 37:13579–13600, 2024. 2

[28] Yajing Liu, Yuning Lu, Hao Liu, Yaozu An, Zhuoran Xu, Zhuokun Yao, Baofeng Zhang, Zhiwei Xiong, and Chenguang Gui. Hierarchical prompt learning for multi-task learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10888–10898, 2023. 2

[29] Zhenyi Lu, Chenghao Fan, Wei Wei, Xiaoye Qu, Dangyang Chen, and Yu Cheng. Twin-merging: Dynamic integration of modular expertise in model merging. *arXiv preprint arXiv:2406.15479*, 2024. 1, 2, 5, 6, 7, 8, 3, 9, 10

[30] Haipeng Luo, Qingfeng Sun, Can Xu, Pu Zhao, Jianguang Lou, Chongyang Tao, Xiubo Geng, Qingwei Lin, Shifeng Chen, and Dongmei Zhang. Wizardmath: Empowering mathematical reasoning for large language models via reinforced evol-instruct. *arXiv preprint arXiv:2308.09583*, 2023. 6

[31] Ziyang Luo, Can Xu, Pu Zhao, Qingfeng Sun, Xiubo Geng, Wenxiang Hu, Chongyang Tao, Jing Ma, Qingwei Lin, and Daxin Jiang. Wizardcoder: Empowering code large language models with evol-instruct. In *The Twelfth International Conference on Learning Representations*, 2024. 6

[32] Xinyin Ma, Gongfan Fang, and Xinchao Wang. Llm-pruner: On the structural pruning of large language models. *Advances in neural information processing systems*, 36:21702–21720, 2023. 4

[33] Michael S Matena and Colin A Raffel. Merging models with fisher-weighted averaging. *Advances in Neural Information Processing Systems*, 35:17703–17716, 2022. 1, 2, 5, 9

[34] Behnam Neyshabur, Hanie Sedghi, and Chiyuan Zhang. What is being transferred in transfer learning? *Advances in neural information processing systems*, 33:512–523, 2020. 7

[35] Tan Nguyen, Minh Pham, Tam Nguyen, Khai Nguyen, Stanley Osher, and Nhat Ho. Fourierformer: Transformer meets generalized fourier integral theorem. *Advances in Neural Information Processing Systems*, 35:29319–29335, 2022. 4

[36] Biqing Qi, Fangyuan Li, Zhen Wang, Junqi Gao, Dong Li, Peng Ye, and Bowen Zhou. Less is more: Efficient model merging with binary task switch. *arXiv preprint arXiv:2412.00054*, 2024. 2

[37] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR, 2021. 1, 3, 4, 5

[38] Victor Sanh, Albert Webson, Colin Raffel, Stephen Bach, Lintang Sutawika, Zaid Alyafeai, Antoine Chaffin, Arnaud Stiegler, Arun Raja, Manan Dey, et al. Multitask prompted training enables zero-shot task generalization. In *International Conference on Learning Representations*, 2022. 6

[39] Noam Shazeer, Azalia Mirhoseini, Krzysztof Maziarz, Andy Davis, Quoc Le, Geoffrey Hinton, and Jeff Dean. Outrageously large neural networks: The sparsely-gated mixture-of-experts layer. In *International Conference on Learning Representations*, 2016. 2, 4

[40] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023. 1, 3, 6

[41] Simon Vandenhende, Stamatios Georgoulis, Wouter Van Gansbeke, Marc Proesmans, Dengxin Dai, and Luc Van Gool. Multi-task learning for dense prediction tasks: A survey. *IEEE transactions on pattern analysis and machine intelligence*, 44(7):3614–3633, 2021. 2

[42] Wenhui Wang, Hangbo Bao, Li Dong, Johan Bjorck, Zhiliang Peng, Qiang Liu, Kriti Aggarwal, Owais Khan Mohammed, Saksham Singhal, Subhojit Som, et al. Image as a foreign language: Beit pretraining for vision and vision-language tasks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 19175–19186, 2023. 7

[43] T Wolf. Huggingface's transformers: State-of-the-art natural language processing. *arXiv preprint arXiv:1910.03771*, 2019. 1

[44] Mitchell Wortsman, Gabriel Ilharco, Samir Ya Gadre, Rebecca Roelofs, Raphael Gontijo-Lopes, Ari S Morcos, Hongseok Namkoong, Ali Farhadi, Yair Carmon, Simon Kornblith, et al. Model soups: averaging weights of multiple fine-tuned models improves accuracy without increasing inference time. In *International conference on machine learning*, pages 23965–23998. PMLR, 2022. 2

[45] Chengyue Wu, Teng Wang, Yixiao Ge, Zeyu Lu, Ruisong Zhou, Ying Shan, and Ping Luo. \$\pi$-tuning: Transferring multimodal foundation models with optimal multi-task interpolation. In *International Conference on Machine Learning*, pages 37713–37727. PMLR, 2023. 4

[46] Can Xu, Qingfeng Sun, Kai Zheng, Xiubo Geng, Pu Zhao, Jiazhan Feng, Chongyang Tao, Qingwei Lin, and Daxin Jiang. Wizardlm: Empowering large pre-trained language models to follow complex instructions. In *The Twelfth International Conference on Learning Representations*, 2024. 6

[47] Jing Xu and Jingzhao Zhang. Random masking finds winning tickets for parameter efficient fine-tuning. In *Forty-first International Conference on Machine Learning*. 2

[48] Prateek Yadav, Derek Tam, Leshem Choshen, Colin A Raffel, and Mohit Bansal. Ties-merging: Resolving interference when merging models. *Advances in Neural Information Processing Systems*, 36, 2024. 1, 2, 3, 4, 5, 6, 7, 8, 9, 10

[49] Enneng Yang, Li Shen, Guibing Guo, Xingwei Wang, Xiaochun Cao, Jie Zhang, and Dacheng Tao. Model merging

in llms, mllms, and beyond: Methods, theories, applications and opportunities. *arXiv preprint arXiv:2408.07666*, 2024. 2

[50] Enneng Yang, Li Shen, Zhenyi Wang, Guibing Guo, Xiaojun Chen, Xingwei Wang, and Dacheng Tao. Representation surgery for multi-task model merging. *arXiv preprint arXiv:2402.02705*, 2024. 3, 4

[51] Enneng Yang, Li Shen, Zhenyi Wang, Guibing Guo, Xiaojun Chen, Xingwei Wang, and Dacheng Tao. Representation surgery for multi-task model merging. In *Forty-first International Conference on Machine Learning*, 2024. 1

[52] Enneng Yang, Zhenyi Wang, Li Shen, Shiwei Liu, Guibing Guo, Xingwei Wang, and Dacheng Tao. Adamerging: Adaptive model merging for multi-task learning. In *The Twelfth International Conference on Learning Representations*, 2024. 1, 2, 5, 6, 3, 9

[53] Le Yu, Bowen Yu, Haiyang Yu, Fei Huang, and Yongbin Li. Language models are super mario: Absorbing abilities from homologous models as a free lunch. In *Forty-first International Conference on Machine Learning*, 2024. 2, 3, 4, 6, 8, 10

[54] Kerem Zaman, Leshem Choshen, and Shashank Srivastava. Fuse to forget: Bias reduction and selective memorization through model fusion. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 18763–18783, Miami, Florida, USA, 2024. Association for Computational Linguistics. 8

[55] Jinghan Zhang, Junteng Liu, Junxian He, et al. Composing parameter-efficient modules with arithmetic operation. *Advances in Neural Information Processing Systems*, 36: 12589–12610, 2023. 6

[56] Yu Zhang and Qiang Yang. A survey on multi-task learning. *IEEE transactions on knowledge and data engineering*, 34 (12):5586–5609, 2021. 1

[57] Yanqi Zhou, Tao Lei, Hanxiao Liu, Nan Du, Yanping Huang, Vincent Zhao, Andrew M Dai, Quoc V Le, James Laudon, et al. Mixture-of-experts with expert choice routing. *Advances in Neural Information Processing Systems*, 35:7103–7114, 2022. 2

[58] Yuyan Zhou, Liang Song, Bingning Wang, and Weipeng Chen. Metagpt: Merging large language models using model exclusive task arithmetic. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 1711–1724, 2024. 1

[59] Liu Zhuang, Lin Wayne, Shi Ya, and Zhao Jun. A robustly optimized BERT pre-training approach with post-training. In *Proceedings of the 20th Chinese National Conference on Computational Linguistics*, pages 1218–1227, Huhhot, China, 2021. Chinese Information Processing Society of China. 6

# FREE-Merging: Fourier Transform for Efficient Model Merging

## Supplementary Material

## A. Notations

We first list the notations for key concepts in our paper.

Table 10. Notations.

| Notations | Descriptions |
|---|---|
| $f_\theta$ | A neural network with parameters $\theta$. |
| $\theta_{pre}$ | Pre-trained model parameters. |
| $\theta_k$ | Fine-tuning parameters for task $k$. |
| $\theta_m$ | Model merging backbone network. |
| $v_k$ | Task vector for task $k$, $v_k = \theta_k - \theta_{pre}$. |
| $M(v_k, d)$ | The top 100d% parameters with the most significant changes corresponding to $v_k$. |
| $\mu_k$ | The rescaling coefficient of the expert of task $k$ |
| $e_k$ | Lightweight task expert for task $k$, $e_k = \mu_k M(v_k, d)$. |
| $\theta_*$ | The backbone with the corresponding expert. |
| $G(\cdot)$ | Transformation operations on parameters. |
| $\lambda_k$ | Merging coefficient of the model parameters for task $k$. |
| $x \in R^m$ | The input of the neural network. |
| $Y \in R^n$ | The output of the neural network. |

## B. Proof

In this section, we provide a proof for Theorem 5.1. We begin by introducing the notation used in this proof.

### B.1. Notations and preliminaries

To illustrate the proof, let's consider two fine-tuned models with parameters $\theta_A$ and $\theta_B$, For clarity, we focus on a single-layer MLP (without activation functions), though we will explain at the end how our proof generalizes to all cases. The merged result is given by: $\theta_m = \lambda_1 G(\theta_A) + \lambda_2 G(\theta_B)$, where $G(\cdot)$ represents a transformation operation applied to the parameters. Assume that $\theta_A$ corresponds to task input $X_A$, and $\theta_B$ corresponds to task input $X_B$.

The goal is to prove that, without additional storage or extra training for $\theta_A$ and $\theta_B$, simply storing $\theta_m$ cannot perfectly retain the capabilities of both $\theta_A$ and $\theta_B$.

### B.2. Details of proof

Next, we will analyze the cases separately.

**Without transformations.** Assume that we do not apply transformations to the output of the merged network. the output of the model $f_\theta$ can be expressed as:

$$output_f = \theta x \tag{5}$$

Suppose $\theta_m$ can perfectly retain the capabilities of both $\theta_A$ and $\theta_B$ without introducing new knowledge. That is, for $\forall x_A \in X_A, x_B \in X_B$, we have:

$$\theta_m x_A = \theta_A x_A,\ \theta_m x_B = \theta_B x_B. \tag{6}$$

Then we have:

$$[\lambda_1 G(\theta_A) + \lambda_2 G(\theta_B)]x_A = \theta_A x_A, \tag{7}$$
$$[\lambda_1 G(\theta_A) + \lambda_2 G(\theta_B)]x_B = \theta_B x_B. \tag{8}$$

Assume that $\exists\ x_m = \mu_1 x_A + \mu_2 x_B \in X_A$, where $\mu_1, \mu_2 \in \mathbb{N}$. Substituting, we can obtain:

$$\theta_m x_m = \theta_A x_m \tag{9}$$
$$\Rightarrow \theta_m(x_A - x_m) = \theta_A(x_A - x_m) \tag{10}$$
$$\Rightarrow \theta_m[(1-\mu_1)x_A - \mu_2 x_B] \tag{11}$$
$$= \theta_A[(1-\mu_1)x_A - \mu_2 x_B] \tag{12}$$
$$\Rightarrow \mu_2\theta_m x_B = \mu_2\theta_A x_B \tag{13}$$
$$\Rightarrow \mu_2\theta_B x_B = \mu_2\theta_A x_B \tag{14}$$

Due to:

$$\mu_2\theta_m x_B = \mu_2\theta_B x_B \tag{15}$$

Thus, if the Eq. 14 and Eq. 15 holds for $\forall x_A \in X_A, x_B \in X_B$, then $\mu_2 = 0$. Therefore, we have:

$$\forall \mu_2 \neq 0, \nexists\ x_m = \mu_1 x_A + \mu_2 x_B \in X_A \tag{16}$$

Eq. 16 indicates that $X_A$ is not continuous. However, we usually consider the input space of the model to be continuous. Even if it is discontinuous, we expect the model to be robust to slight perturbations in the input.

Thus, the assumption that $\theta_m$ can perfectly retain $\theta_A$ and $\theta_B$ without extra data or training does not hold in this case.

**With transformations.** Assuming we apply certain transformations to the output of the merged network, the output of the model $f_\theta$ can be expressed as:

$$output_f = h(\theta x), \tag{17}$$

where $h(\cdot)$represents a transformation applied to the output.

By the universal approximation theorem, $h(\cdot)$ can be approximated by a two-layer MLP with sigmoid activation. Then, assuming $\theta_m$ can perfectly retain the capabilities of both $\theta_A$ and $\theta_B$. Then for $\forall x_A \in X_A, x_B \in X_B$, we have:

$$W_2[sigmoid(W_1\theta_m x_A)] = \theta_A x_A \tag{18}$$
$$W_2[sigmoid(W_1\theta_m x_B)] = \theta_B x_B. \tag{19}$$

Substituting, we obtain:

$$\frac{W_2}{1 + e^{-W_1\theta_m x_A}} = \theta_A x_A, \tag{20}$$
$$\frac{W_2}{1 + e^{-W_1\theta_m x_B}} = \theta_B x_B. \tag{21}$$

Then we have:

$$W_2 = \theta_A x_A (1 + e^{-W_1 \theta_m x_A}) \tag{22}$$
$$= \theta_B x_B (1 + e^{-W_1 \theta_m x_B}). \tag{23}$$

Thus, we can obtain the following equation:

$$\theta_A x_A (e^{W_1 \theta_m x_A} + 1) e^{W_1 \theta_m x_B} \tag{24}$$
$$= \theta_B x_B (e^{W_1 \theta_m x_B} + 1) e^{W 1 \theta_m x_A}. \tag{25}$$

Therefore,

$$(\theta_A x_A - \theta_A x_B)(e^{W_1 \theta_m x_B} e^{W_1 \theta_m x_A} + e^{W_1 \theta_m x_B}) \tag{26}$$
$$= \theta_B x_B (e^{W_1 \theta_m x_B} + e^{W_1 \theta_m x_A}) \tag{27}$$

Therefore, based on Equa. 22 and Equa. 26, since they hold for $\forall x_A \in X_A, x_B \in X_B$, at least one of $W_1$ or $W_2$ can be expressed as $\Phi(\theta_A, \theta_B)$, where $\Phi$ represents a function.

In other words, only by additionally storing part of the information from $\theta_A$ or $\theta_B$ and processing $\theta_m$ at the output can $\theta_m$ perfectly retain the capabilities of both $\theta_A$ and $\theta_B$. Therefore, the assumption that $\theta_m$ can perfectly retain both $\theta_A$ and $\theta_B$ does not hold in this case.

Next, we briefly outline the rationale for extending this result to all layers of a neural network. Our assumptions involve only a single layer of the neural network, which already cannot perfectly retain capabilities without introducing new data. Therefore, it is even less feasible for the entire network to achieve this. Thus, this extension is reasonable.

In summary, the proposition is proven.

# C. Reproducibility

## C.1. Datasets

**Merging 8 ViTs.** We use ViT-B/32 and ViT-L/14 as pre-trained models, and fine-tune them on 8 image classification datasets (SUN397, Cars, RESISC45, EuroSAT, SHVN, GTSRB, MNIST, and DTD), then merge the models and test their performance. Configuration details follow [20].

**Merging 30 ViTs.** We use ViT-B/16 as the pre-trained model and test its merging performance on 30 image datasets. The datasets include MNIST, CIFAR-10, Vegetables, Food-101, Kvasir-v2, Cars, Intel Images, EuroSAT, Weather, Cats and Dogs, MangoLeafBD, Beans, CIFAR-100, GTSRB, SHVN, Dogs, Fashion MNIST, Oxford-IIIT-Pet, Landscape Recognition, Flowers Recognition, STL-10, CUB-200-2011, EMNIST, DTD, RESISC45, SUN397, KenyanFood13, Animal-10N, Garbage Classification, and Fruits-360. These datasets cover a wide range of major image categories. Configuration details follow [20].

**Merging Medium-sized Language Models.** We use RoBERTa as the pre-trained model, fine-tune it on 8 classification task datasets from GLUE benchmark for model merging, including CoLA, SST-2, MRPC, STS-B, QQP, MNLI, QNLI, and RTE. CoLA is evaluated with the Matthews correlation coefficient, STS-B with the average of the Pearson and Spearman correlation coefficients, and the others by accuracy. Details follow [20].

**Merging PEFT Models.** In this section, we conduct two experiments. The first uses T0-3B as the pre-trained model and $IA^3$ as the PEFT method, fine-tuned and merged on RTE, CB, Winogrande, WiC, WSC, COPA, H-SWAG, Story Cloze, and ANLI (R1 to R3). Details can be found in [20]. The second part uses Qwen-14B as the pre-trained model and LoRA as the PEFT method. We fine-tune and merging on three generative tasks: MMLU, TruthfulQA, and BBQ. Configuration details can be found in [29].

**Merging Large Language Models.** We use LLaMa2-13B as the pre-trained model and apply fine-tuned results from WizardLM, WizardMath, and WizardCoder-Python for instruction following, math, and code tasks, respectively. We perform merging and test the model on three datasets: AlpacaEval (instruction following task), GSM8K (math task), and MBPP (code generation task). Details follow [53].

**Merging Multi-Modal Models.** We use BEiT3 as the pre-trained model, fine-tune it on five multi-modal datasets, and test model merging performance. The datasets are ImageNet-1k (Image Classification), VQAv2 (Visual Question Answering), NLVR2 (Visual Reasoning), COCO Captioning (Image Captioning), and COCO Retrieval (Image-Text Retrieval). COCO Captioning is evaluated using BLEU4, CIDEr, METEOR, and ROUGE-L, while the other tasks are measured by accuracy. Details follow [20].

## C.2. Baselines

**Individual.** Solve each task with its fine-tuned model, but this requires storing separate models for each task, leading to significant storage overhead.

**Traditional MTL.** Train a model on data from all tasks, which incurs high training costs and data privacy issues.

**Weight Averaging.** Average the parameters for merging. The method is simple, but it faces great performance loss.

**Fisher Merging** [33]. Use Fisher information matrices to assess parameter importance and determine merging coefficients. However, this requires calculating the matrix for each model, demanding high computational resources, making it unsuitable for edge deployment.

**Task Arithmetic** [21]. Define task vectors as the merging target. For task $k$, the task vector is defined as $v_k = \theta_k - \theta_{pre}$, where $\theta_{pre}$ is the pre-trained model parameters, and $\theta_k$ is the fine-tuned parameters for task $k$. The merging process can be represented as $\theta_m = \theta_{pre} + \lambda \sum_{i=1}^{K} v_i$, where $\lambda$ is the merging coefficient. This method suffers significant performance degradation due to unaddressed task conflicts.

**Ties-Merging** [48]. Attempts to resolve parameter conflicts during model merging by eliminating redundancy and sign conflicts. However, resolving parameter conflicts is insuffi-

cient to address task conflicts, resulting in performance loss.

**Breadcrumbs** [7]. Discards parameters with the largest and smallest absolute values as redundant, negatively impacting model merging. This approach is simple, but it shows a significant performance decline on certain tasks.

**PCB-Merging** [12]. Uses internal balancing to measure parameter importance within tasks and mutual balancing to assess parameter similarity across tasks, discarding redundant parameters and adjusting merging coefficients. This approach requires considerable computational resources, whereas our FR-Merging approach requires less computational power while yielding better results.

**RegMean** [23]. A weighted merging model based on a closed-form solution for the merging problem, aiming to maximize the similarity between the merged model and each pre-merged model on the same input. Requires inner product data from training. Since this method relies on data information, which is typically not provided by the model provider, its applicability is limited to specific scenarios.

**AdaMerging** [52]. Uses an unsupervised approach to learn the merging coefficient for each task vector or layer. AdaMerging++ additionally applies Ties-Merging before calculating the merging coefficient. This method is limited to classification tasks and requires certain training resources, making it unsuitable for edge deployment.

**DARE** [53]. DARE randomly discards a large portion of task vector parameters before merging, potentially reducing parameter interference among models. This method is simple, but due to the lack of further optimization in the merging, it suffers from significant performance degradation.

**EMR-Merging** [20]. Proposes retaining a mask matrix and rescale parameters for each model. During inference, the mask matrix and rescaling parameters are selected to recover performance. This method cannot further improve performance because it does not optimize the merged backbone. Additionally, since a mask must be stored for each task, it requires considerable storage space. In contrast, our method optimizes these aspects simultaneously.

**Twin-Merging** [29]. Proposes saving an expert module extracted by SVD from the original parameters for each task, which is dynamically added during inference. This method does not optimize the merged backbone, resulting in suboptimal performance. It requires storing a large number of parameters when constructing task experts. In contrast, our proposed method optimizes both aspects, making it suitable for resource-constrained edge deployment scenarios.

### C.3. Details

In this section, we discuss the details of the experiments. All experiments, except for the speed tests, are conducted on eight NVIDIA RTX-3090 GPUs. The inference speed tests are performed using a single RTX-3090 GPU.

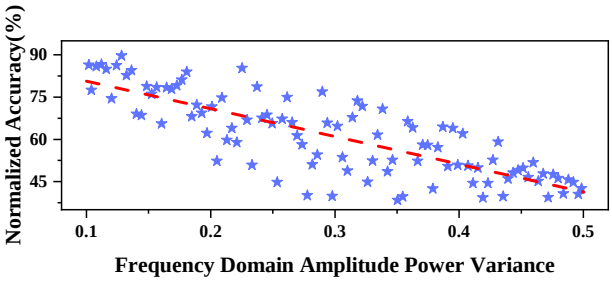



Figure 6. The relationship between the merging performance of ViT-B-32 and the variance of frequency domain amplitude power. Normalized accuracy is the ratio of merged model performance to fine-tuned model performance on the same task.

## D. More Discussions and Future Directions

In this section, we first provide a more detailed discussion of task interference in the frequency domain in D.1. Next, we present the experimental results of FR-Merging on language models in D.2 and D.3. Additionally, we discuss the effectiveness and construction of the expert module in D.4 and D.5, as well as the role of the router in D.6. We also explore the selection of the cutoff frequency in D.7 and explored the implications of forgetting model-specific capabilities in D.8. Finally, we conclude with a discussion of future work in D.9.

### D.1. Task Interference in Frequency Domain

In this section, we further discuss the task interference presented in Sec. 4 and provide visualization results.

The first point to address is why we consider the difference in low-frequency amplitude in the frequency domain as the cause of task interference. The amplitude in the frequency domain represents the strength of a signal. During the merging process, if one signal is strong while the other is weak, the weaker signal is likely to be overshadowed by the stronger one, regardless of their directional relationship. This manifests in the merged model as the near-complete loss of performance from one of the fine-tuned models. Therefore, a significant difference in the low-frequency amplitude will indeed lead to task interference during merging.

In Sec. 4, we quantify interference using the mean variance, observing a strong negative correlation with merged model performance on ViT-B/32. Here, normalized performance refers to the ratio of the performance of merged model on a task to that of the fine-tuned model on the same task. This negative correlation serves as strong evidence of task interference in the frequency domain.

Next, we verify whether this negative correlation also holds for language models. We choose Flan-T5 [6] as the base model and conduct experiments on eight GLUE datasets. As shown in Fig. 6, the negative correlation is observed in language models, demonstrating that task inter-

ference in the frequency domain is a general phenomenon.

Finally, we examine whether low-frequency task interference is widespread across different models. In Fig. 7 and Fig. 8, we visualize the frequency distribution of fine-tuned parameters for ViT-B/32 and Flan-T5 [6] on their respective tasks. The results show that task interference in the low-frequency region is consistently present across various models. Therefore, our hypothesis is well-supported. Meanwhile, we observe an interesting phenomenon: in some layers, frequency-domain amplitude clustering occurs, meaning that certain models have very similar amplitude distributions. This may be related to the training path during the finetuning process, and we will further investigate this phenomenon in future research.

### D.2. Expanded Loss Landscape Analysis

In this section, we attempt to analyze why our proposed FR-Merging outperforms existing approaches from the perspective of the loss landscape. A loss landscape comparison with two popular sparse methods, Top-K retention and DARE, is shown in Fig. 9 (a-c). Our proposed FR-Merging method brings the two fine-tuned models closer in the loss landscape for the same task, making their linear combination fall closer to the loss basin of the target task, thereby achieving better performance. This aligns with our analysis from the perspective of task interference and is also consistent with the performance improvements in experiments. Therefore, our proposed FR-Merging method offers significant advantages over current approaches.

### D.3. Fourier Transform on Language Models

In Sec. 5.2, we analyze the impact of high-pass filtering on the generalization ability of fine-tuned models and their performance on corresponding tasks using ViT-B/32. In this part, we extend the analysis to language models, further demonstrating the effectiveness of high-pass filtering. Fig. 9(d) shows the generalization test of RoBERTa on the GLUE datasets, where filtering out low-frequency information also enhances the generalization ability of fine-tuned language models. This improvement is also reflected in the merging performance improvement for language models. However, compared to the effect on language models, high-pass filtering has a more pronounced impact on ViT. This may be attributed to the nature of image data, which inherently possesses certain high-frequency and low-frequency properties. Nevertheless, in summary, FR-Merging, based on high-pass filtering, is a generalizable, efficient, and high-performing model merging approach.

### D.4. Role of Task Experts

In this section, we demonstrate that introducing additional information can potentially mitigate task conflicts.

We use ViT-B/32 as the pre-trained model and fine-tune it on 8 image classification tasks, following the setup in [21]. To prevent the impact of parameter erasure, we select only 5% of non-overlapping parameters for each task. As a result, after this selection, only task conflicts affect the model merging process. Merging resulted in a significant performance drop, indicating the presence of task conflicts. However, when we double task-specific vectors for each task during inference, performance improves substantially. This suggests that introducing additional information for each task has the potential to resolve task conflicts.

Table 11. Using ViT-B/32 as the pre-trained model, we fine-tune on eight classification tasks, selecting 5% non-overlapping parameters per task. Performance improves significantly by adding task-specific parameters during inference.

| | Avg acc(%) | Norm acc(%) |
|---|---|---|
| Fine-tuned | 90.5 | 100.0 |
| No Overlap 5% params. | 89.91 | 99.34 |
| No Overlap Merging | 67.31 | 74.37 |
| + Task Knowledge | 84.55 | 93.42 |

### D.5. Task Experts Construction Analysis

In this section, we analyze the construction of experts from two aspects: the parameter selection and rescaling.

**Top-K Selection.** We discuss the rationale behind using the parameters with the largest changes during the fine-tuning to approximate task-specific information introduced by fine-tuning, thereby constructing task experts.By comparing L1 distances of model parameters in the frequency domain, with smaller distances indicating better approximation, we assess the effectiveness of different methods.

Based on the analysis in Sec 5.3, low-frequency information contains more task-specific information but is discarded during model merging in our proposed FR-Merging. Therefore, task experts should compensate for this missing information. First, we briefly explain why approximation methods are preferred over directly saving low-frequency information. If low-frequency data is directly saved, ensuring lightweight storage requires saving frequency-domain data instead of inverse-transformed results. This would necessitate performing inverse transformations during expert invocation, significantly increasing the inference time. Considering the constraints of time and storage in edge-side deployment, a more efficient approximation method is needed.

Next, we compare the extent to which different approximation methods approximate low-frequency information to assess their suitability for constructing task experts. In Fig 10, we present the effects of approximating using the top 5% largest fine-tuned parameters and random sampling. It can be observed that using the parameters with the largest changes during the fine-tuning closely approximates the frequency-domain characteristics of low-frequency in-

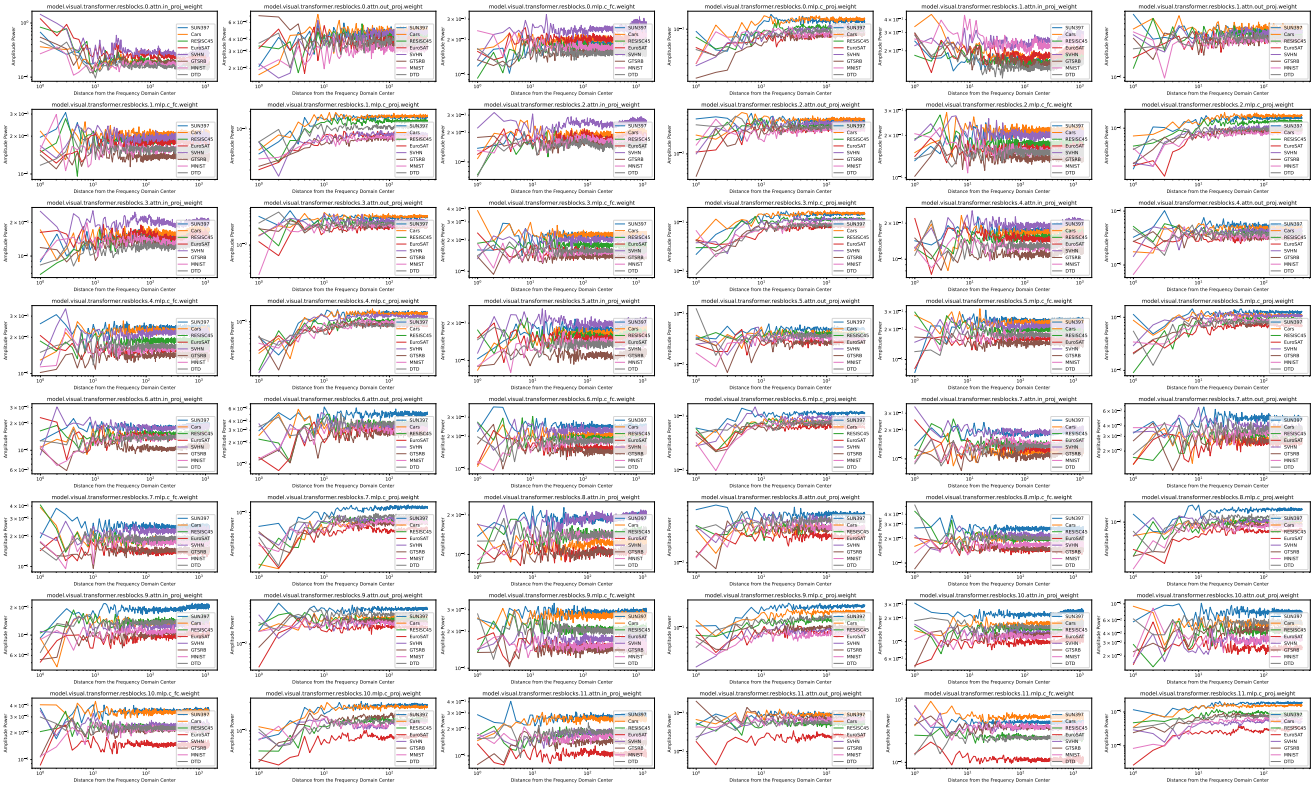


Figure 7. The parameter frequency domain distribution of ViT-B-32 fine-tuning results on eight image tasks.

Table 12. Ablation of scaling for experts on 8 visual tasks.

| | ViT-B/32 | ViT-B/16 | ViT-L/14 |
|---|---|---|---|
| w/o scaling | 0.5314 | 0.5323 | 0.6090 |
| w/ scaling | **0.8484** | **0.8486** | **0.8895** |

formation, strongly supporting the rationale for preserving task-specific information by saving the parameters with the largest changes during the fine-tuning. Additionally, the high similarity between low-frequency information and the original information confirms our hypothesis: low-frequency information contains more task-specific information introduced during fine-tuning.

In summary, using the parameters with the largest changes during fine-tuning as task experts to offset task-specific information lost during merging is reasonable. Furthermore, as validated in Sec 6.5, employing our proposed rescaling method yields even better results. Thus, the expert construction method proposed in this paper is justified.

**Rescaling.** We then discuss the necessity of rescaling. Since we retain only a small portion of the parameters during expert extraction, it alters the mean and variance of parameters, similar to dropout. Our scaling method considers both the mean and the backbone to preserve the similarity between the expert and original model outputs. Table 12 shows the CKA similarity between expert outputs and original outputs before and after scaling, with higher values indicating better expert performance. Scaling can be found to greatly improve performance compared to not applying it.

Table 13. Ablation study of router training.

| Training Data Size | 0% | 0.1% | 0.5% | 1% | 2% | 5% |
|---|---|---|---|---|---|---|
| ViT-B/32 | 78.1 | 87.0 | 88.2 | 89.7 | 90.0 | 90.1 |
| ViT-L/14 | 88.3 | 92.1 | 93.0 | 93.7 | 93.9 | 94.1 |

## D.6. Role of Router

In this section, we discuss why a router is necessary for dynamically assigning input data to the appropriate expert. First, let's review how the router works: when a new data arrives, the router determines which existing expert is best suited to handle it. For example, if we have experts for mathematics and physics, and a math problem appears, the router will automatically route it to the mathematics expert. The mathematics expert will then collaborate with the merged backbone to solve the problem. Manually assigning each input sample to the correct task is impractical, and in real-world applications, we cannot expect users to know in advance which expert is best suited for each piece of data. Therefore, having a router to automate this process is both crucial and aligned with real-world usage scenarios.



Figure 8. The parameter frequency domain distribution of Flan-T5 fine-tuning results on eight GLUE tasks.

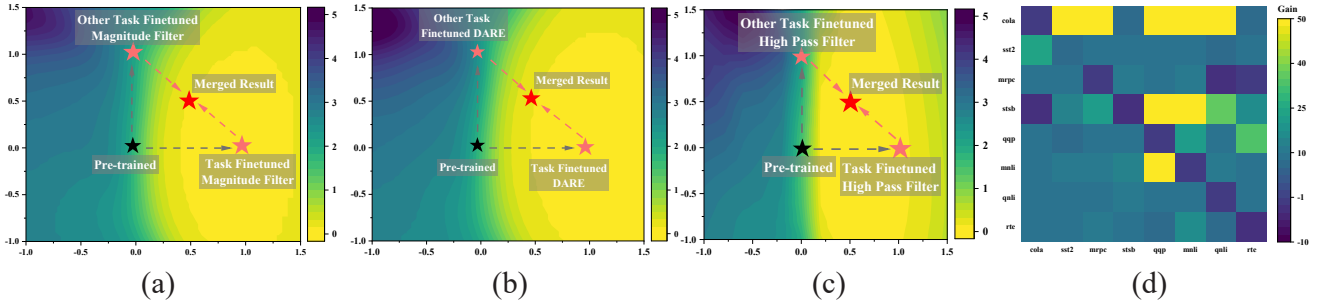



Figure 9. a) Merge by preserving the top 20% of parameters; b) Merge using the DARE method; c) FR-Merging. **a-c** complete on ViT/B-32 and test on 8 visual tasks. d) High-pass filtering improves performance over the unfiltered version on RoBERTa and test on GLUE.

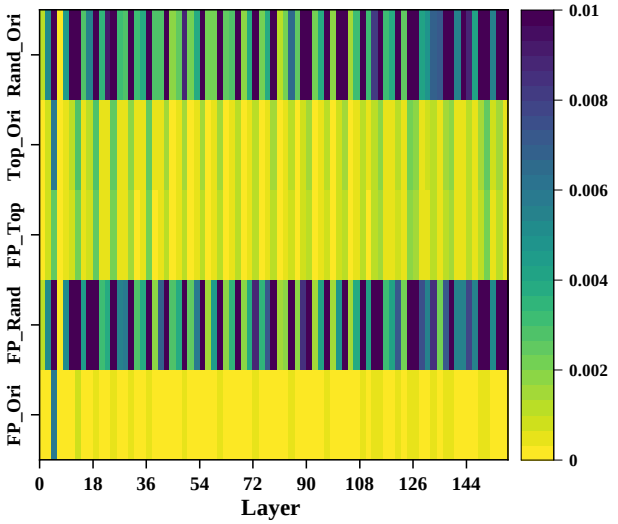



Figure 10. The analysis evaluates the reasonableness of approximating task-specific information using the most changed parameters during fine-tuning, by calculating the L1 distance of each layer's parameters in CLIP-B/32 fine-tuned on 8 visual tasks with different approximation methods. Here, *Ori* refers to the original parameters, *FP* to low-pass filtering, *Top* to maximum value approximation, and *Rand* to random sampling approximation.

Meanwhile, in this section, we discuss the amount of data required for training the router. In Tab. 13, we present the average performance across eight tasks using FREE-Merging with different sizes of training data in the fusion experiments on ViT-B/32 and ViT-B/14. It can be observed that the performance improvement becomes marginal when the training data exceeds 1%; therefore, in practical applications, we use only 1% of the data for training. If resources are even more limited, using only 0.1% of the data can still yield acceptable results.

## D.7. Filtering Frequency Hyperparameter Analysis

In this section, we analyze the cutoff frequency of the Fourier high-pass filter. We perform merging using the results after fine-tuning the ViT-B/32 on eight visual classification tasks. We explore the impact of filtering different proportions of low-frequency signals on model merging.

The results are shown in the Table 14. It can be observed that filtering a certain proportion of low-frequency information significantly improves the performance of the model merging. For example, filtering 10% of the low-frequency information, despite its small proportion, leads to an impressive 12% improvement in average performance. This is similar to the analysis of low-frequency information in image processing, where low-frequency components contain most of the features in the overall information. In the context of model merging, low-frequency information represents the specialized knowledge brought by fine-tuning. While this specialized information can improve model performance on specific tasks, it severely affects the generalization ability of the model, leading to task conflicts during merging. Therefore, filtering a small proportion of low-frequency information helps minimize model specialization without significantly impacting performance, thus avoiding performance loss caused by task conflicts.

From the experimental results in Table 14, it is evident that our method is robust to the filtering ratio. Even after filtering 40% of the information, the method still demonstrates strong performance improvements, with a noticeable performance drop only after filtering 60% of the information. This strongly indicates that the method proposed in this paper is highly robust to the cutoff frequency as a hyper-parameter. The robustness to hyper-parameters makes our method simple to deploy, providing a significant advantage.

## D.8. Fuse to Forget

In this section, we discuss the forgetting of task-specific components caused by the merging process. In the setting explored in this paper, we aim to preserve the capabilities of multiple individual models simultaneously, so that the merged model can address multiple tasks. Therefore, in our setting, it is crucial to minimize the loss of task-specific capabilities during the fusion process. To address this issue,

Table 14. The performance of Fourier high-pass filtering at different filtering percentages.

| Filtering Percentage | SUN397 | Cars | RESISC45 | EuroSAT | SVHN | GTSRB | MNIST | DTD | Avg. |
|---|---|---|---|---|---|---|---|---|---|
| Individual | 75.3 | 77.7 | 96.1 | 99.7 | 97.5 | 98.7 | 99.7 | 79.4 | 90.5 |
| 0% | 65.3 | 63.4 | 71.4 | 71.7 | 64.2 | 52.8 | 87.5 | 50.1 | 65.8 |
| 10% | 66.1 | 65.3 | 77.2 | 90.0 | 84.3 | 81.0 | 98.2 | 58.4 | 77.6 |
| 20% | 65.6 | 64.5 | 76.9 | 90.3 | 84.6 | 82.6 | 98.5 | 59.1 | 77.7 |
| 30% | 66.2 | 64.5 | 77.2 | 90.1 | 85.4 | 82.3 | 98.5 | 60.0 | 78.1 |
| 40% | 65.2 | 64.4 | 76.1 | 90.4 | 84.5 | 82.2 | 98.5 | 59.0 | 77.5 |
| 50% | 64.9 | 64.4 | 76.3 | 88.9 | 82.3 | 81.2 | 98.4 | 58.9 | 76.9 |
| 60% | 65.4 | 65.0 | 76.9 | 86.7 | 77.3 | 76.6 | 96.9 | 57.7 | 75.3 |

Table 15. Results of merging RoBERTa pre-trained models on eight datasets from GLUE benchmark.

| Method | CoLA | SST2 | MRPC | STSB | QQP | MNLI | QNLI | RTE | Avg |
|---|---|---|---|---|---|---|---|---|---|
| Individual | 60.18 | 94.04 | 89.22 | 90.63 | 91.41 | 87.20 | 92.71 | 79.06 | 85.55 |
| Weight Averaging | 13.96 | 64.11 | 69.36 | 31.84 | 75.36 | 42.19 | 58.70 | 55.23 | 51.34 |
| Task Arithmetic [21] | 18.78 | 85.89 | 79.90 | 74.03 | 83.78 | 59.08 | 69.67 | 62.09 | 66.65 |
| Ties-Merging [48] | 20.48 | 84.40 | 81.13 | 58.19 | 85.70 | 64.65 | 74.81 | 42.96 | 64.04 |
| Breadcrumbs [7] | **25.23** | 85.21 | 78.19 | 68.45 | 81.07 | 64.40 | **79.83** | 63.18 | 68.19 |
| PCB-Merging [12] | 23.46 | 84.59 | 79.36 | 68.89 | 81.98 | 63.57 | 76.56 | 64.47 | 67.86 |
| FR-Merging(ours) | 15.25 | **86.81** | **82.39** | **80.81** | **86.21** | **65.69** | 78.26 | **64.73** | **70.02** |
| RegMean [23] | 36.67 | 90.60 | 75.74 | 62.68 | 83.55 | 70.02 | 82.35 | 58.48 | 70.01 |
| EMR-Merging [20] | 37.82 | 87.91 | 82.31 | **72.90** | 82.61 | 78.91 | 83.24 | 67.92 | 74.20 |
| Twin-Merging [29] | 52.30 | 93.23 | **88.65** | 68.64 | 81.99 | 78.80 | 88.40 | 74.37 | 78.29 |
| FREE-Merging(ours) | **54.50** | **93.69** | 88.46 | 67.04 | **88.03** | **80.60** | **89.90** | **79.06** | **80.16** |

we propose FR-Merging and FREE-Merging.

On the other hand, [54] suggests that such forgetting is not necessarily harmful, for example, in reducing model bias and other undesirable effects. In their setup, the goal is to actively forget certain undesirable or sensitive information through merging. Thus, the desired behavior of model fusion differs between the two settings.

However, what remains consistent is that the merging process can lead to a loss of each model's specific capabilities. As discussed earlier, this is primarily due to potential conflicts with the knowledge from other models.

### D.9. Future Works

In this section, we present the directions for our future work. First, we plan to extend our current merging methods to fine-tuned models with different architectures, thereby enhancing the generalizability and applicability of our approach across a wider range of model types. In addition, we intend to conduct a deeper investigation into the differences between parameter-efficient fine-tuning (PEFT) and full fine-tuning, aiming to understand their respective characteristics and behaviors more thoroughly. Based on these insights, we will develop customized merging strategies specifically tailored to PEFT, enabling more effective integration and deployment of existing PEFT-based models.

## E. Additional Experimental Results

### E.1. Merging Vision Models

In this section, we comprehensively present the detailed results of model merging using ViT-L/14 as the pre-trained backbone model across all 30 diverse vision datasets introduced and described in Sec. 6.1, as shown in Table 16.

From the results, we can see that, consistent with the conclusions presented in the main text, our proposed FR-Merging method significantly outperforms all existing cost-free model merging techniques. It is able to efficiently construct a high-performance merged backbone, demonstrating strong generalization and robustness across various tasks without incurring any additional training or computational overhead. This makes FR-Merging a highly effective solution for model merging in resource-constrained settings. In addition, when we consider the dynamic addition of expert models through FREE-Merging, the method is able to maintain satisfactory performance levels while introducing only a minimal amount of extra computational cost. This flexibility allows for adaptive model expansion depending on the complexity of the task, making FREE-Merging a practical approach for real-world applications where both performance and efficiency are important considerations.

Table 16. Task-specific and average performance when merging ViT-B/16 models on **30** tasks.

| **Task-specific Acc** | MNIST | Cifar-10 | Vegetables | Food-101 | Kvasir-v2 | Intel-Images | Cars | EuroSAT | Weather | Cats and Dogs |
|---|---|---|---|---|---|---|---|---|---|---|
| Individual | 99.22 | 97.88 | 100.00 | 87.93 | 94.31 | 94.63 | 85.96 | 99.04 | 98.22 | 99.05 |
| Weight Averaging | 27.63 | 42.91 | 83.20 | 68.02 | 25.27 | 82.40 | 7.74 | 24.37 | 61.06 | 91.28 |
| Task Arithmetic [21] | 30.81 | 59.86 | 91.97 | 73.06 | 31.05 | 89.03 | 9.34 | 31.25 | 74.56 | 93.61 |
| Ties-Merging [48] | 23.21 | 42.82 | 92.31 | 73.22 | 21.09 | 89.39 | 5.30 | 10.98 | 72.86 | 91.88 |
| Breadcrumbs [7] | 33.95 | 54.29 | 91.83 | 66.24 | 32.12 | 14.30 | 0.19 | 32.33 | 69.04 | 90.92 |
| FR-Merging(ours) | **42.78** | **63.35** | **93.16** | **76.25** | **38.13** | **90.73** | **12.00** | **36.37** | **75.60** | **96.69** |
| RegMean [23] | 90.71 | 89.65 | **99.10** | 76.14 | 71.00 | 93.60 | 16.28 | 74.13 | 86.62 | 98.54 |
| AdaMerging [52] | 81.22 | 87.54 | 97.97 | 75.23 | 22.76 | 91.02 | 0.42 | 44.60 | 89.13 | 96.91 |
| Twin-Merging [29] | 92.10 | 89.34 | 98.54 | 81.43 | 78.45 | 94.14 | 58.12 | 69.43 | 92.23 | **98.85** |
| EMR-Merging [20] | 93.40 | 90.23 | 98.34 | 83.54 | 76.54 | **95.12** | 54.34 | 76.64 | 94.54 | 98.65 |
| FREE-Merging(ours) | **97.30** | **95.60** | 99.00 | **84.65** | **85.25** | 94.45 | **58.52** | **79.12** | **96.38** | 98.62 |

| | Dogs | Fashion | Pet | LandScape | Flowers | STL-10 | CUB-200-2011 | EMNIST | DTD | RESISC45 |
|---|---|---|---|---|---|---|---|---|---|---|
| Individual | 85.16 | 93.26 | 92.23 | 86.83 | 98.19 | 99.07 | 84.79 | 94.67 | 71.76 | 98.90 |
| Weight Averaging | 47.80 | 20.46 | 31.26 | 73.14 | 68.97 | 37.74 | 37.66 | 7.73 | 14.63 | 13.56 |
| Task Arithmetic [21] | 47.65 | 37.11 | 33.24 | 79.59 | 80.68 | 39.66 | **41.86** | 11.05 | 14.73 | 15.50 |
| Ties-Merging [48] | 26.03 | 27.05 | 12.84 | 78.27 | 34.33 | 6.17 | 31.28 | 5.61 | 3.71 | 6.79 |
| Breadcrumbs [7] | 40.91 | 35.29 | 39.16 | 83.80 | 81.94 | 54.73 | 27.58 | 3.25 | 17.81 | 15.20 |
| FR-Merging(ours) | **49.86** | **42.73** | **41.26** | **85.80** | **84.49** | **59.53** | 41.79 | **23.80** | **22.23** | **19.49** |
| RegMean [23] | 42.89 | 83.42 | 34.62 | 83.64 | 95.26 | 78.94 | 49.78 | 48.67 | 30.53 | 34.66 |
| AdaMerging [52] | 53.09 | 76.76 | 48.34 | 81.98 | 95.69 | 68.91 | 48.19 | 18.02 | 16.68 | 24.83 |
| Twin-Merging [29] | 72.65 | 80.64 | 68.78 | 85.49 | 96.75 | 74.72 | 60.47 | 57.47 | 41.25 | 50.57 |
| EMR-Merging [20] | 72.34 | 82.35 | 67.35 | 85.73 | 96.46 | 76.65 | 63.11 | 60.03 | 40.48 | 49.57 |
| FREE-Merging(ours) | **75.86** | **86.43** | **72.57** | **86.42** | **97.54** | **79.01** | **65.65** | **63.64** | **45.71** | **54.25** |

| | MangoLeafBD | Beans | Cifar-100 | GTSRB | SVHN | SUN397 | KenyanFood13 | Animal-10N | Garbage | Fruits-360 |
|---|---|---|---|---|---|---|---|---|---|---|
| Individual | 100.00 | 97.73 | 89.85 | 95.74 | 96.22 | 78.98 | 85.53 | 92.52 | 93.36 | 99.63 |
| Weight Averaging | 68.58 | 70.98 | 77.98 | 15.00 | 10.88 | 57.42 | 33.55 | 46.00 | 22.89 | 5.38 |
| Task Arithmetic [21] | **87.02** | 84.62 | 80.20 | 37.01 | 17.41 | 55.88 | 36.32 | 51.14 | 25.23 | 6.15 |
| Ties-Merging [48] | 76.58 | 67.22 | 78.61 | 40.74 | 10.54 | 52.69 | 19.90 | 19.13 | 3.91 | 1.50 |
| Breadcrumbs [7] | 79.25 | 85.15 | 75.65 | 38.51 | 22.89 | 47.96 | 26.07 | 17.303 | 28.06 | 1.40 |
| FR-Merging(ours) | 86.50 | **87.38** | **82.03** | **44.19** | **30.08** | **56.98** | **38.90** | **49.10** | **32.01** | **12.00** |
| RegMean [23] | 98.10 | 92.58 | 82.59 | 56.96 | 66.13 | 58.58 | 57.11 | 68.74 | 65.31 | 19.79 |
| AdaMerging [52] | 99.13 | 93.38 | 84.19 | 59.90 | 25.70 | 64.09 | 48.66 | 66.55 | 38.54 | 7.94 |
| Twin-Merging [29] | 100.0 | 94.24 | 85.57 | 65.87 | 75.49 | 71.25 | 63.81 | 71.67 | 70.82 | 25.56 |
| EMR-Merging [20] | 100.0 | 95.45 | 86.43 | 65.94 | 76.46 | 72.45 | 65.43 | 73.84 | 72.24 | 28.24 |
| FREE-Merging(ours) | **100.0** | **96.40** | **88.65** | **68.63** | **82.23** | **76.78** | **69.45** | **78.40** | **75.84** | **37.97** |

| **Average Acc** | Individual | Weight Averaging | Task Arithmetic | Ties-Merging | Breadcrumbs | **FR-Merging(ours)** |
|---|---|---|---|---|---|---|
| Acc | 93.02 | 42.52 | 48.89 | 37.53 | 43.57 | **53.91** |

| **Average Acc** | Individual | RegMean | AdaMerging | Twin-Merging | EMR-Merging | **FREE-Merging(ours)** |
|---|---|---|---|---|---|---|
| Acc | 93.02 | 68.14 | 60.25 | 75.52 | 76.39 | **79.67** |

Table 17. Results of merging $(IA)^3$ models with T0-3B as pre-trained model on eleven NLP tasks.

| Method | RTE | CB | Winogrande | WiC | WSC | COPA | H-SWAG | Story Cloze | ANLI-R1 | ANLI-R2 | ANLI-R3 | Avg. Acc |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Individual | 82.7 | 95.8 | 75.1 | 71.7 | 65.3 | 85.3 | 44.4 | 94.9 | 70.2 | 46.5 | 53 | 71.4 |
| Traditional MTL | 88.6 | 95.8 | 75.5 | 61.1 | 80.6 | 94.1 | 42.3 | 97.6 | 70.5 | 49.8 | 47.7 | 73.1 |
| Fisher Merging [33] | 83.3 | 83.3 | 56.7 | 54.2 | 58.3 | 83.1 | 42.2 | 94.1 | 45.9 | 41.0 | 42.2 | 62.2 |
| Task Arithmetic [21] | 74.1 | 83.3 | 62.8 | 49.1 | 49.3 | 87.5 | 41.5 | 95.3 | 60.8 | 49.4 | 50.0 | 63.9 |
| Weight Averaging | 81.2 | 58.3 | 53.8 | 55.2 | 53.5 | 80.9 | 40.1 | 92.5 | 43.3 | 39.2 | 40.2 | 58.0 |
| Ties-Merging [48] | 78.0 | 83.3 | 67.9 | 57.6 | 59.7 | 81.7 | 42.8 | 90.3 | **66.9** | **51.3** | **51.1** | 66.4 |
| Breadcrumbs [7] | 71.9 | 59.4 | 53.1 | 31.2 | 64.1 | 78.1 | 46.9 | 90.2 | 50.0 | 31.2 | 25.0 | 54.6 |
| PCB-Merging [12] | **85.9** | 83.3 | 61.9 | 57.1 | 63.9 | 82.4 | 42.7 | 91.2 | 64.2 | 47.8 | 45.9 | 66.1 |
| FR-Merging(ours) | 81.3 | **83.8** | **68.3** | **57.9** | **65.2** | **88.4** | **48.6** | **93.5** | 56.2 | 46.2 | 46.3 | **66.9** |
| RegMean [23] | 81.2 | 58.3 | 53.8 | 55.2 | 53.5 | 80.9 | 40.1 | 92.5 | 43.3 | 39.2 | 40.2 | 58.0 |
| EMR-Merging [20] | 81.8 | 87.5 | 66.6 | 56.1 | 65.3 | 82.4 | 44.7 | 93.6 | 65.7 | 43.8 | 50.8 | 67.1 |
| Twin-Merging [29] | 81.2 | 85.6 | 65.4 | 57.2 | **66.3** | 81.6 | 44.4 | 92.9 | 66.5 | 42.4 | 51.2 | 66.8 |
| FREE-Merging(ours) | **83.2** | **89.3** | **69.9** | **58.4** | 65.5 | **84.3** | **45.2** | **94.2** | **67.9** | **45.2** | **52.4** | **68.7** |

Table 18. Results of merging LoRA models with QWEN-14B as pre-trained model on three generative tasks.

| Method | MMLU | TruthfulQA | BBQ | Avg. |
|---|---|---|---|---|
| Pre-trained | 69.30 | 51.27 | 80.69 | 67.09 |
| Fine-tuned | 68.35 | 54.34 | 93.53 | 72.07 |
| Weight Averaging | 68.10 | 50.01 | 82.31 | 66.80 |
| Task Arithmetic [21] | 67.62 | 53.38 | 78.24 | 66.41 |
| Task Arithmetic (w/ DARE) [53] | 67.82 | 52.66 | 82.83 | 67.77 |
| Ties-Merging [48] | 68.27 | 50.01 | **84.10** | 67.46 |
| Ties-Merging (w/ DARE) [53] | **69.32** | 53.07 | 81.19 | 67.86 |
| Breadcrumbs [7] | 68.24 | 52.88 | 79.20 | 66.77 |
| PCB-Merging [12] | 67.23 | 52.48 | 80.37 | 66.69 |
| FR-Merging(ours) | 68.16 | **53.39** | 82.44 | **68.00** |
| RegMean [23] | 67.89 | 52.45 | 82.24 | 67.52 |
| EMR-Merging [20] | 67.82 | 55.01 | 90.13 | 70.98 |
| Twin-Merging [29] | 68.32 | 55.76 | 90.98 | 71.68 |
| FREE-Merging(ours) | **68.83** | **57.39** | **92.14** | **72.78** |

## E.2. Merging Language Models

In this section, we present the complete and detailed experimental results of merging a diverse set of language models, each fine-tuned under different settings, across multiple widely-used benchmark datasets that cover both classification and generative tasks.

First, we present the merging results using RoBERTa as the pre-trained model, which was fine-tuned individually on the eight tasks included in the GLUE benchmark. These results are summarized in Table 15. Our proposed method achieves the best performance on nearly all of the datasets, highlighting its strong generalization capability when applied to language models. In addition, Table 17 reports the merging results across 11 language classification tasks, where T0-3B serves as the pre-trained model and $IA^3$ is used for fine-tuning. Our method also demonstrates a clear advantage in this setting. Finally, Table 18 provides the results for merging with QWEN-14B as the pre-trained model and LoRA as the fine-tuning technique, evaluated on three generative tasks. In all these experiments, both FR-Merging and FREE-Merging deliver consistently superior performance compared to other existing model merging methods, confirming their effectiveness across various fine-tuning strategies and model types.

Overall, our proposed methods demonstrate strong generalization capabilities when applied to language models, consistently delivering robust performance across a wide range of tasks, architectures, and fine-tuning strategies.