# LoRE-Merging: Exploring Low-Rank Estimation For Large Language Model Merging

Anonymous ACL submission


## Abstract

While most current approaches rely on further training techniques, such as fine-tuning or reinforcement learning, to enhance model capacities, model merging stands out for its ability of improving models without requiring any additional training. In this paper, we propose a unified framework for model merging based on low-rank estimation of task vectors without the need for access to the base model, named LoRE-Merging. Our approach is motivated by the observation that task vectors from fine-tuned models frequently exhibit a limited number of dominant singular values, making low-rank estimations less prone to interference. We implement the method by formulating the merging problem as an optimization problem. Extensive empirical experiments demonstrate the effectiveness of our framework in mitigating interference and preserving task-specific information, thereby advancing the state-of-the-art performance in model merging techniques.


## 1 Introduction

Large Language Models (LLMs) have become ubiquitous in numerous real-world applications (Bommasani et al., 2021; Zhuang et al., 2020). The utilization of LLMs typically involves fine-tuning them for specific tasks, a process that often yields superior performance compared to general-purpose LLMs. A rapidly emerging technique in this domain is model merging (Garipov et al., 2018; Wortsman et al., 2022; Yu et al., 2024b), which aims to create a single multi-task model by combining the weights of multiple task-specific models. This approach facilitates the construction of multi-task models by integrating knowledge from fine-tuned (FT) models without requiring additional training.

Building on recent studies (Ilharco et al., 2022; Yadav et al., 2024; Yu et al., 2024b), task vector-based merging approaches have demonstrated significant effectiveness, where task vectors are defined as the parameter differences between fine-tuned models and the base LLM. Achieving optimal results in model merging often requires minimizing interference between task vectors associated with different tasks. To address this, existing approaches utilize modified task vectors instead of the original ones. For instance, Yu et al. (2024b) applied random dropping with probability $p$ to obtain a sparse representation of task vectors, while Yadav et al. (2024) retained only the top-$k$ elements of each task vector based on magnitude, setting the remaining elements to zero. These strategies aim to produce sparse estimations of task vectors, a common technique for mitigating interference.

Nevertheless, task vector-based model merging approaches remain constrained by two fundamental limitations. First, the computation of task vectors necessitates access to the base model parameters and demonstrates heightened sensitivity to parametric variations (Yu et al., 2024b). As fine-tuning progress goes deeper, substantial parametric divergence emerges between the original base model and its fine-tuned counterpart, thereby greatly hindering them merging effectiveness (Yu et al., 2024a). Second, empirical evidence from Yadav et al. (2024) reveals that conflicting task vectors interactions could appear even when employing sparse estimation techniques. On the other hand, the sparsification process risks inadvertently eliminating essential task-specific features, thereby compromising the efficacy of the resultant merged model. These inherent constraints of sparse approximation methodologies underscore the necessity for developing alternative frameworks to estimate higher-fidelity low-rank task vector representations.

To this end, we first empirically validate that task vectors exhibit a small number of dominant singular values, with the remaining singular values being significantly smaller in magnitude, as shown in Figure 1. Additionally, the dimension of the intersection of the images of two matrices is bounded

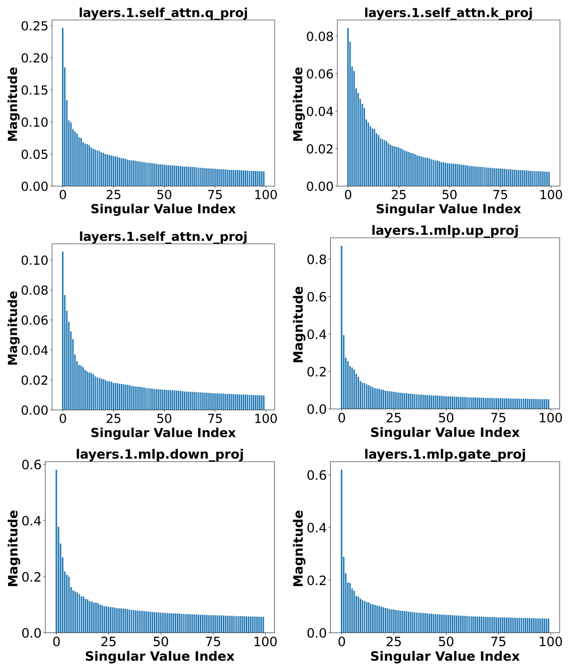


Figure 1: Singular value distributions for the task vector in layer 1. We show the top-100 singular values, out of 4096 within the full rank.

by the minimum of their ranks. Therefore, we propose LORE-MERGING, a unified framework for model merging based on **Lo**w-**R**ank **E**stimation of task vectors, which eliminates the need for access to the base model. Specifically, given a set of FT models, we formulate the merging problem as an optimization problem whose goal is to simultaneously identify an approximate base model integrated with a set of low-rank task vectors that collectively approximate the behavior of the FT models. By leveraging low-rank estimations, task vectors become inherently less susceptible to interference, effectively addressing a fundamental challenge in model merging. We conduct extensive experiments on optimization modeling problems and math word problems to confirm the effectiveness of our method.

## 2 Related Works

Merging fine-tuned models has been shown to offer several benefits, such as improving performance on a single target task (Gupta et al., 2020; Choshen et al., 2022; Wortsman et al., 2022), enhancing out-of-domain generalization (Cha et al., 2021; Arpit et al., 2022; Ilharco et al., 2022; Ramé et al., 2023), creating multi-task models from different tasks (Jin et al., 2022; Li et al., 2022; Yadav et al., 2024), supporting continual learning (Yadav and Bansal, 2022; Yadav et al., 2023), and addressing other challenges (Don-Yehiya et al., 2022; Li et al., 2022). Among these methods, task-vector-based merging approaches play an important role. Task Arithmetic (Ilharco et al., 2022) first introduced the concept of task vectors and shows that simple arithmetic operations can be performed to obtain the merged models. Building on this idea, methods like DARE (Yu et al., 2024b) and Ties (Yadav et al., 2024) adopt pruning-then-scaling techniques to merge task vectors, based on the assumption that not all parameters equally contribute to the final performance. However, these methods based on sparsity estimation consistently suffer from the interference among task vectors and require access to the base model, thus limiting their overall effectiveness.

## 3 Methodology

### 3.1 Problem Setting

We denotes $\mathcal{M}_i$ as the candidate models to be merged, where each $\mathcal{M}_i$ is parameterized by $\boldsymbol{\theta}_i$. In this work, we focus on the homogeneous model merging (Wortsman et al., 2022; Ilharco et al., 2022; Yadav et al., 2024), suggesting that the base models share the same model architecture. Specifically, these models can be obtained from the training process, such as checkpoints, or fine-tuned from the same pre-trained model, referred to as task-specific models. The primary objective of model merging is to construct a new model, $\mathcal{M}^*$, having better performance on the target single or multiple tasks.

### 3.2 Implicit Low-Rank Estimation for Model Merging

As demonstrated in Yu et al. (2024b,a), model pairs would exhibit limited mergeability, especially when comprehensive fine-tuning or extended pre-training procedures are employed and result in substantial parameter shifts. Under such conditions, existing task-vector based merging methods struggle to work due to the significant representational divergence between the base model and its fine-tuned derivative. To address this challenge, we propose an implicit low-rank estimation model merging method, named LORE-MERGING, which not only employs the robustness of low-rank estimation against the interference but also eliminates the need for access to the base model.

The core idea of LORE-MERGING is straightforward: instead of using the original base model, we first construct an approximate base model and subsequently integrate the task-specific vectors via a low-rank approximation technique. Formally, denote the approximate base model as $\boldsymbol{\theta}_0$ and the low-rank task vectors $\{\boldsymbol{\delta}_i\}_{i=1}^n$ where $n$ is the number of FT models, our objective is to minimize the discrepancy between each FT model and its corresponding integrated version derived from the constructed base model, expressed as $\boldsymbol{\theta}_0 + \boldsymbol{\delta}_i \approx \boldsymbol{\theta}_i$.

To ensure the low-rank structure of $\boldsymbol{\delta}$, we apply a nuclear norm penalty, as suggested in Cai et al. (2008). Then, we formulate the merging problem as the following optimization problem:

$$\min_{\boldsymbol{\theta}_0, \boldsymbol{\delta}_1, \ldots, \boldsymbol{\delta}_n} f := \sum_{i=1}^{n} \left( \|\boldsymbol{\theta}_0 + \boldsymbol{\delta}_i - \boldsymbol{\theta}_i\|_F^2 + \mu \|\boldsymbol{\delta}_i\|_*^2 \right), \tag{1}$$

where $\|\cdot\|_*$ represents the nuclear norm, and $\mu > 0$ is a hyperparameter. In Equation (1), the first term minimizes the difference between $\boldsymbol{\theta}_0 + \boldsymbol{\delta}_i$ and $\boldsymbol{\theta}_i$, ensuring reconstruction accuracy. The second term acts as a penalty that encourages the task vectors $\boldsymbol{\delta}_i$ to exhibit low-rank properties.

This problem is a standard multi-variable convex optimization problem. To solve it efficiently, we employ the coordinate descent method (Wright, 2015). Starting from an initial point $\{\boldsymbol{\theta}_0^0, \boldsymbol{\delta}_1^0, \ldots, \boldsymbol{\delta}_n^0\}$, each iteration (round $k+1$) updates the variables by iteratively solving the following single-variable minimization problem:

$$\begin{cases} \boldsymbol{\theta}_0^{k+1} = \arg\min_{\boldsymbol{\theta}} f(\boldsymbol{\theta}, \boldsymbol{\delta}_1^k, \cdots, \boldsymbol{\delta}_n^k) \\ \boldsymbol{\delta}_i^{k+1} = \arg\min_{\boldsymbol{\delta}} f(\cdots, \boldsymbol{\delta}_{i-1}^k, \boldsymbol{\delta}, \boldsymbol{\delta}_{i+1}^k, \cdots), \ \forall i \end{cases} \tag{2}$$

The update for $\boldsymbol{\theta}_0^*$ is trivial, while the update for $\boldsymbol{\delta}$ is less straightforward due to the presence of the nuclear norm. Fortunately, as shown in Cai et al. (2010), closed-form solutions for the coordinate descent method iterations in Problem (1) can be obtained using the Singular Value Thresholding (SVT) technique. Recall that for a given matrix $\boldsymbol{\delta}$ with the Singular Value Decomposition (SVD) $\boldsymbol{\delta} = U\Sigma V^\top$, and a hyperparameter $\mu$, the SVT operator is defined as follows. Let $\Sigma^+(\mu) := \mathrm{diag}((\sigma_i - \mu)^+)$, where $(\cdot)^+$ denotes the positive part function. The $\mathrm{SVT}(\boldsymbol{\delta}; \mu)$ operator with hyperparameter $\mu$ is then defined as $\mathrm{SVT}(\boldsymbol{\delta}; \mu) := U\Sigma^+(\mu)V^\top$. Using the SVT operator, the update for $\boldsymbol{\delta}_i$ can be expressed as: $\boldsymbol{\delta}_i^{k+1} = \mathrm{SVT}(\boldsymbol{\theta}_i - \boldsymbol{\theta}_0^{k+1}; \mu)$.

**Algorithm 1** Implicit low-rank merging method

Input: fine-tuned models $\{\boldsymbol{\theta}_i\}_{i=1}^n$, parameter dimension $d$, and hyperparameter $\lambda, \mu$.
Output: merged model $\boldsymbol{\theta}^*$.

$\triangleright$ Step 1: Coordinate descent method to solve problem (1).
Set $\boldsymbol{\delta}_i = 0$ for $i = 1, 2, \ldots, n$.
**while** iteration NOT converges **do**
  $\boldsymbol{\theta}_0 = \frac{1}{n}\sum_{i=1}^n (\boldsymbol{\theta}_i - \boldsymbol{\delta}_i)$
  **for** $i = 1, \ldots, n$ **do**
    $\boldsymbol{\delta}_i = \mathrm{SVT}(\boldsymbol{\theta}_i - \boldsymbol{\theta}_0; \mu)$;
  **end for**
**end while**

$\triangleright$ Step 2 (Optional 1): Direct sum.
$\boldsymbol{\tau} = \sum_{i=1}^n \boldsymbol{\delta}_i$.

$\triangleright$ Step 2 (Optional 2): TIES selection (Yadav et al., 2024).
$\gamma = sgn(\sum_{i=1}^n \boldsymbol{\delta}_i)$.
**for** $p = 1, 2, \ldots, d$ **do**
  $\mathcal{A}^p = \{i : \gamma_i^p = \gamma^p\}$
  $\boldsymbol{\tau}^p = \frac{1}{|\mathcal{A}^p|}\sum_{i \in \mathcal{A}^p} \boldsymbol{\tau}^p$
**end for**

$\triangleright$ Step 3: Obtain merged checkpoint.
$\boldsymbol{\theta}^* = \boldsymbol{\theta}_0 + \lambda\boldsymbol{\tau}$.
**return** $\boldsymbol{\theta}^*$

Once the optimization problem is solved, we can obtain the approximate base model and a set of low-rank task vectors. Then, existing task-vector based approaches, such as Average Merging and Ties-Merging, can be applied to combine the task vectors and the base model. In this work, we directly adopt Average Merging as our post-calculation merging methods for simplicity, as as it demonstrated comparable performance to Ties-Merging in our preliminary experiments. The overall process is outlined in Algorithm 1.

## 4 Experiments

**Baselines & Settings** We compare LoRE-Merging with following popular merging methods. **Average Merging** (Choshen et al., 2022): This method computes the element-wise mean of all the individual models. **DARE** (Yu et al., 2024b): This approach randomly drops task-specific vectors and rescales the remaining vectors back to the base model. We set the hyperparameter for the random probability to 0.5. **Ties-Merging** (Yadav et al.,

| Method | Deepseek & NuminaMath | | WizardLM & WizardMath | | Checkpoints | | | Avg. |
|---|---|---|---|---|---|---|---|---|
| | GSM8K | MATH | GSM8K | MATH | EasyLP | ComplexLP | NL4OPT | |
| Baseline | 76.3 | **55.8** | 54.8 | 12.4 | 81.9 | 39.3 | 94.0 | 59.21 |
| Average | 75.0 | 45.8 | 58.8 | 12.6 | 75.9 | 40.3 | 91.6 | 57.14 |
| DARE | 81.0 | 54.2 | 14.9 | 3.7 | 80.7 | 35.1 | **95.1** | 52.10 |
| Ties-Merging | 80.8 | 51.6 | 58.5 | 11.8 | 82.4 | 42.7 | 94.8 | 60.37 |
| LoRE-Merging | **81.0** | 52.7 | **60.3** | **13.0** | **83.4** | **47.4** | 94.8 | **61.80** |

Table 1: The evaluation results on math word problems and optimization modeling problems. We use the best performance of base models as the baseline.

2024): In this method, task-specific vectors are randomly dropped, and only the parameters aligned with the final agreed-upon sign are merged. For Ties-merging, we set the top-$k$ value to 20%, and the hyperparameter $\lambda$ is fixed at 1. For LoRE-Merging, the rank $r$ is determined dynamically. For a given task vector $\boldsymbol{\delta} \in \mathbf{R}^{m \times n}$, we set the rank $r = 0.2 \times \min\{m, n\}$ to get a low-rank estimation.

**Evaluation** We first evaluate LoRE-Merging on math word problems using the popular benchmarks GSM8K (Cobbe et al., 2021) and MATH (Hendrycks et al.). For comprehensive evaluation, we test both DeepSeek-series models (NuminaMath-7B (Beeching et al., 2024) and DeepSeek-Math-7B-Base (Shao et al., 2024)) and LLaMA-series models (WizardLM-13B (Xu et al., 2023) and WizardMath-13B (Luo et al., 2023)). Additionally, we also evaluate the effectiveness of LoRE-Merging on another advanced task, i.e. mathematical optimization modeling problems (Ramamonjison et al., 2023; Huang et al., 2024, 2025). This task aims to generate solvable mathematical models given an optimization problem in natural language. As the lack of public models on this task, we first fine-tune Qwen-2.5-Coder-7B-Instruct model (Hui et al., 2024) with the datasets provided by Huang et al. (2025) and merge the checkpoints in the training process. The evaluations are conducted on MAMO dataset (Huang et al., 2024) which includes two subsets EasyLP and ComplexLP, and NL4OPT dataset (Ramamonjison et al., 2023).

**Main Results** As shown in Table 1, LoRE-Merging achieves superior performance across most metrics, as well as the highest overall score. For the math word problem evaluation, our method demonstrates consistently superior performance compared to the baselines, except for the evaluation on the MATH dataset when merging DeepSeek-Math with NuminaMath. due to the significant performance gap between the base models, where DeepSeek-Math achieves only a score of 36.2 on the MATH dataset, while NuminaMath reaches 55.8. As indicated in Yao et al. (2024), a large performance gap can significantly impact the effectiveness of model merging. Another worthy-noting observation is that DARE demonstrates significantly poorer performance when merging WizardLM and WizardMath. This can likely be attributed to the substantial parameter divergence between these models, which results in the failure of calculating the task vector derived from the base model. In contrast, our LoRE-Merging with the approximate base model and low-rank task vectors demonstrates superior robustness and effectiveness in solving math word problems. For the evaluations on optimization modeling with checkpoints merging, we can see existing task-vector based merging methods consistently improve the performance because of the marginal gap between the checkpoints. Therefore, we believe that checkpoint merging can serve as a highly effective technique complementary to training methods, particularly our LoRE-Merging method. We also conduct a detailed analysis how our method enhance the modeling capacity on ComplexLP dataset. We found that the earlier checkpoint is more good at identifying the variables and parameters in the questions while the later one focuses on more complex components, such as formulating variables and the constraints. With the merging of task vectors, the merged model exhibits superior overall performance on the task.

## 5 Conclusion

In this paper, we propose a unified framework for merging homogeneous models based on low-rank estimation, named LoRE-Merging. The main motivation of our work is to estimate task vectors using low-rank approximation without the need of access to the base model. We achieve it by formulating the merging problem as an optimization problem. Extensive experiments demonstrate the efficacy and efficiency of our proposed methods.

## Limitations

Although we have demonstrated the effectiveness of our method on merging homogeneous models, we have not yet evaluated it on merging heterogeneous models which is a much more challenging task. Compared to existing task-vector based model merging methods, our method is the most suitable one that can be adapted to heterogeneous model merging, as we disentangle the base model and task vectors. We think how to expand LORE-MERGING to heterogeneous model merging should be a promising future direction.

## References


Devansh Arpit, Huan Wang, Yingbo Zhou, and Caiming Xiong. 2022. Ensemble of averages: Improving model selection and boosting performance in domain generalization. *Advances in Neural Information Processing Systems*, 35:8265–8277.

Edward Beeching, Shengyi Costa Huang, Albert Jiang, Jia Li, Benjamin Lipkin, Zihan Qina, Kashif Rasul, Ziju Shen, Roman Soletskyi, and Lewis Tunstall. 2024. Numinamath 7b tir. `https://huggingface.co/AI-MO/NuminaMath-7B-TIR`.

Rishi Bommasani, Drew A Hudson, Ehsan Adeli, Russ Altman, Simran Arora, Sydney von Arx, Michael S Bernstein, Jeannette Bohg, Antoine Bosselut, Emma Brunskill, et al. 2021. On the opportunities and risks of foundation models. *arXiv preprint arXiv:2108.07258*.

Jian-Feng Cai, Emmanuel J. Candes, and Zuowei Shen. 2008. A singular value thresholding algorithm for matrix completion. *Preprint*, arXiv:0810.3286.

Jian-Feng Cai, Emmanuel J Candès, and Zuowei Shen. 2010. A singular value thresholding algorithm for matrix completion. *SIAM Journal on optimization*, 20(4):1956–1982.

Junbum Cha, Sanghyuk Chun, Kyungjae Lee, Han-Cheol Cho, Seunghyun Park, Yunsung Lee, and Sungrae Park. 2021. Swad: Domain generalization by seeking flat minima. *Advances in Neural Information Processing Systems*, 34:22405–22418.

Leshem Choshen, Elad Venezian, Noam Slonim, and Yoav Katz. 2022. Fusing finetuned models for better pretraining. *arXiv preprint arXiv:2204.03044*.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. 2021. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*.

Shachar Don-Yehiya, Elad Venezian, Colin Raffel, Noam Slonim, Yoav Katz, and Leshem Choshen. 2022. Cold fusion: Collaborative descent for distributed multitask finetuning. *arXiv preprint arXiv:2212.01378*.

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

Vipul Gupta, Santiago Akle Serrano, and Dennis DeCoste. 2020. Stochastic weight averaging in parallel: Large-batch training that generalizes well. *arXiv preprint arXiv:2001.02312*.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. Measuring mathematical problem solving with the math dataset. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2)*.

Chenyu Huang, Zhengyang Tang, Shixi Hu, Ruoqing Jiang, Xin Zheng, Dongdong Ge, Benyou Wang, and Zizhuo Wang. 2025. Orlm: A customizable framework in training large models for automated optimization modeling. *Preprint*, arXiv:2405.17743.

Xuhan Huang, Qingning Shen, Yan Hu, Anningzhe Gao, and Benyou Wang. 2024. Mamo: a mathematical modeling benchmark with solvers. *Preprint*, arXiv:2405.13144.

Binyuan Hui, Jian Yang, Zeyu Cui, Jiaxi Yang, Dayiheng Liu, Lei Zhang, Tianyu Liu, Jiajun Zhang, Bowen Yu, Kai Dang, et al. 2024. Qwen2. 5-coder technical report. *arXiv preprint arXiv:2409.12186*.

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

Xisen Jin, Xiang Ren, Daniel Preotiuc-Pietro, and Pengxiang Cheng. 2022. Dataless knowledge fusion by merging weights of language models. *arXiv preprint arXiv:2212.09849*.

Margaret Li, Suchin Gururangan, Tim Dettmers, Mike Lewis, Tim Althoff, Noah A Smith, and Luke Zettlemoyer. 2022. Branch-train-merge: Embarrassingly parallel training of expert language models. *arXiv preprint arXiv:2208.03306*.

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

Rindranirina Ramamonjison, Timothy T. Yu, Raymond Li, Haley Li, Giuseppe Carenini, Bissan Ghaddar, Shiqi He, Mahdi Mostajabdaveh, Amin Banitalebi-Dehkordi, Zirui Zhou, and Yong Zhang. 2023. Nl4opt competition: Formulating optimization problems based on their natural language descriptions. *Preprint*, arXiv:2303.08233.

Alexandre Ramé, Kartik Ahuja, Jianyu Zhang, Matthieu Cord, Léon Bottou, and David Lopez-Paz. 2023. Model ratatouille: Recycling diverse models for out-of-distribution generalization. In *International Conference on Machine Learning*, pages 28656–28679. PMLR.

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

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

Stephen J. Wright. 2015. Coordinate descent algorithms. *Preprint*, arXiv:1502.04759.

Can Xu, Qingfeng Sun, Kai Zheng, Xiubo Geng, Pu Zhao, Jiazhan Feng, Chongyang Tao, and Daxin Jiang. 2023. Wizardlm: Empowering large language models to follow complex instructions. *arXiv preprint arXiv:2304.12244*.

Prateek Yadav and Mohit Bansal. 2022. Exclusive supermask subnetwork training for continual learning. *arXiv preprint arXiv:2210.10209*.

Prateek Yadav, Qing Sun, Hantian Ding, Xiaopeng Li, Dejiao Zhang, Ming Tan, Xiaofei Ma, Parminder Bhatia, Ramesh Nallapati, Murali Krishna Ramanathan, et al. 2023. Exploring continual learning for code generation models. *arXiv preprint arXiv:2307.02435*.

Prateek Yadav, Derek Tam, Leshem Choshen, Colin A Raffel, and Mohit Bansal. 2024. Ties-merging: Resolving interference when merging models. *Advances in Neural Information Processing Systems*, 36.

Yuxuan Yao, Han Wu, Mingyang Liu, Sichun Luo, Xiongwei Han, Jie Liu, Zhijiang Guo, and Linqi Song. 2024. Determine-then-ensemble: Necessity of top-k union for large language model ensembling. *arXiv preprint arXiv:2410.03777*.

Le Yu, Bowen Yu, Haiyang Yu, Fei Huang, and Yongbin Li. 2024a. Extend model merging from fine-tuned to pre-trained large language models via weight disentanglement. *arXiv preprint arXiv:2408.03092*.

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

Fuzhen Zhuang, Zhiyuan Qi, Keyu Duan, Dongbo Xi, Yongchun Zhu, Hengshu Zhu, Hui Xiong, and Qing He. 2020. A comprehensive survey on transfer learning. *Proceedings of the IEEE*, 109(1):43–76.