LLM Merging Competition Technical Report: Efficient Model Merging with Strategic Model Selection, Merging, and Hyperparameter Optimization

Zixiang Di Yaoming Yang Mei Jiang Bingdong Li* Hong Qian Aimin Zhou

Abstract

The LLM Merging Competition in NeurIPS'24 aims to build LLMs efficiently through model merging, which enables the combination of multiple specialized fine-tuned models into a single model without the need for additional training. However, existing model merging techniques often suffer from performance degradation or require the model maker's apriori knowledge or intuition to set hyperparameters. To address these challenges, this technical report proposes an efficient model merging (EMM) approach with three key modules: (1) strategic model selection, (2) hybrid merging algorithm, and (3) hyperparameter optimization. Our approach aims to improve the effectiveness of merged model while maintaining efficiency throughout the merging process. Experiments on different tasks verify the effectiveness of the proposed approach.

1 Introduction

Model Selection. To improve the effectiveness of merging multiple specialized fine-tuned models, significant efforts have focused on developing model merging algorithms. [1–4] However, the crucial step of selecting the appropriate models for merging to achieve the best possible performance has been largely overlooked. Additionally, given that the LLM Merging Competition emphasizes merging models with fewer than 8B parameters and requires that the models existed when the competition was announced [5], identifying suitable models within these constraints becomes a critical challenge that cannot be ignored.

Merging Algorithm. Task Arithmetic (TA) [1], a simple delta weight merging method, often results in significant performance degradation during the merging of multiple models due to interference among different model parameters. To mitigate this issue, advanced model merging algorithms, such as TIES [2], DARE [3], and DELLA [4], have introduced parameter sparsification before merging. Among them, DELLA proposes a magnitude-based parameter pruning approach, which has demonstrated advantages over TIES and DARE. However, DELLA's pruning method requires layer-by-layer sorting of parameters in LLMs, leading to considerable additional time overhead. Given that the LLM Merging Competition requires submissions to complete merging/fine-tuning and evaluation within one hour on a single Nvidia A6000 (48 GB) or equivalent resource [5], We adopted a merging algorithm that integrates elements from current state-of-the-art methods while avoiding sparsification techniques that require parameter sorting, thereby maintaining the efficiency of the model merging process.

Hyperparameter Optimization. Existing model merging techniques require the model maker to have profound knowledge or intuition to set hyperparameters. For instance, Yu et al. [3] uses grid

*All the authors are with School of Computer Science and Technology, the Shanghai Institute of AI for Education, and the Shanghai Frontiers Science Center of Molecule Intelligent Syntheses, East China Normal University, Shanghai 200062, China. Corresponding author: bdli@cs.ecnu.edu.cn.

38th Conference on Neural Information Processing Systems (NeurIPS 2024).

search within [0.1, 0.2, ..., 0.9] to find the optimal drop rate p for DARE. However, Akiba et al.'s research [6] indicates that accurately configured hyperparameters can significantly enhance the performance of merged models. To address this, we use a multi-objective black-box optimization method to determine the optimal hyperparameter configuration before merging the models.

2 Methodology

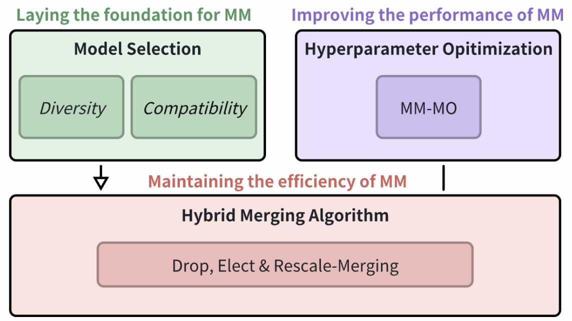
Laying the foundation for MM
Improving the performance of MM
Model Selection
Diversity
Compatibility
Hyperparameter Opitimization
MM-MO
Maintaining the efficiency of MM
Hybrid Merging Algorithm
Drop, Elect & Rescale-Merging

Figure 1: Three key modules of EMM: (1) Strategic Model Selection, (2) Hybrid Merging Algorithm, and (3) Hyperparameter Optimization.

Figure 1 shows the three key **modules** of our proposed EMM. Specifically, strategic model selection lays the foundation for model merging; the hybrid merging algorithm maintains the efficiency of model merging; and the optimized hyperparameter setting improves the performance of model merging.

2.1 Strategic Model Selection (SMS)

Considering the competition's criteria that models must be publicly available on Hugging Face, uploaded before May 31st, 2024, and have no more than 8 billion parameters, we selected Meta-Llama-3-8B-Instruct [7] as the base model. It is the latest model meeting these requirements and demonstrates strong performance across various LLM benchmarks.

For model merging, we chose the following four fine-tuned models: [Model 1] MaziyarPanahi/Llama-3-8B-Instruct-v0.8 [2], [Model 2] MaziyarPanahi/Llama-3-8B-Instruct-v0.9 [3], [Model 3] shenzhi-wang/Llama3-8B-Chinese-Chat [4] [8], and [Model 4] lightblue/suzume-llama-3-8B-multilingual [5] [9] Specifically, Models 1 and 2 are the 4th and 5th best-performing 8B models on the Open LLM Leaderboard [10] as of March 6th, 2024 [6]. Model 3 was trained on a dataset of nearly 100K preference pairs, excelling in roleplay, function calling, and math capabilities. Model 4 was fine-tuned on almost 90,000 multilingual conversations based on Llama-3 [7], enhancing its multilingual chatting abilities while retaining the intelligence of the base model.

These models were selected as source models because they share the same chat template, reducing the risk of confusing outputs. Moreover, they were fine-tuned on large, diverse datasets, ensuring that each model offers unique strengths across different tasks. This diversity aids the merging algorithm in combining the distinct capabilities of each model, resulting in good generalization ability.

2.2 Hybrid Merging Algorithm (HMA)

For model merging, we use a hybrid merging algorithm combines DROP, ELECT & RESCALE-MERGING. We tried to avoid sparsification operators with parameter sorting since they are too time-consuming.

[2] https://huggingface.co/MaziyarPanahi/Llama-3-8B-Instruct-v0.8

[3] https://huggingface.co/MaziyarPanahi/Llama-3-8B-Instruct-v0.9

[4] https://huggingface.co/shenzhi-wang/Llama3-8B-Chinese-Chat

[5] https://huggingface.co/lightblue/suzume-llama-3-8B-multilingual

[6] https://huggingface.co/datasets/open-llm-leaderboard-old/details_MaziyarPanahi__Llama-3-8B-Instruct-v0.8, https://huggingface.co/datasets/open-llm-leaderboard-old/details_MaziyarPanahi__Llama-3-8B-Instruct-v0.9

Specifically, to prevent parameter conflicts during model merging that can lead to performance degradation, HMA begins with sparsifying the parameters of each fine-tuned model. The DARE [3] algorithm has demonstrated that even discarding over 90% of delta parameters allows fine-tuned models to maintain their original performance. Therefore, we applied random sparsification to delta parameters across models with varying DROP probabilities p, effectively reducing the computational overhead of the merging process.

Moreover, for handling sign conflicts during merging, we used the ELECT SIGN method from TIES [2], which chooses the maximum absolute value of each parameter in the specified sign direction. This step mitigates interference caused by conflicting parameter signs.

Finally, for the remaining sparse parameters, we RESCALE them by $1/(1-p)$ as suggested by DARE, which reported that rescaling the remaining elements after discarding most of the task vector yields better performance than no rescaling.

2.3 Hyperparameter Optimization (HO)

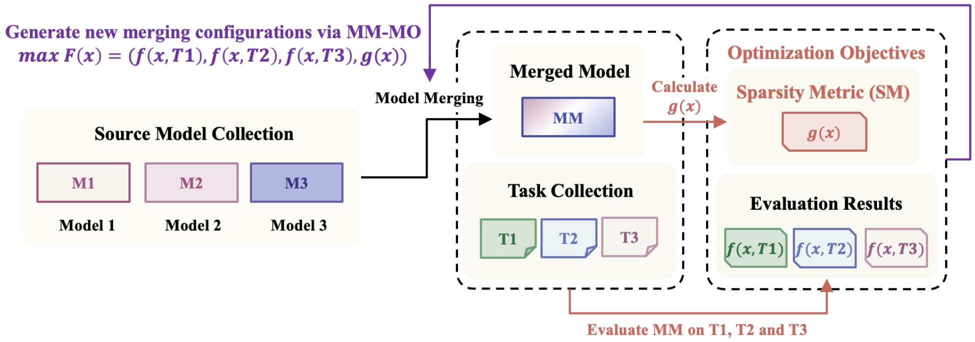

Figure 2: An illustration of hyperparameter optimization for model merging algorithms using multi-objective optimization (MM-MO).

In this section, we use a novel method for setting hyperparameters for LLM merging via black-box multi-objective optimization algorithm (MM-MO) [11]. The automated approach aims to find the optimal hyperparameters for the merging algorithm without relying on human knowledge or intuition, replacing the commonly used grid search method in previous model merging studies [3, 4].

In Figure 3, we show the overall process of hyperparameter optimization for model merging algorithms via multi-objective optimization. The method starts with a set of source models, each fine-tuned on different datasets and their hyperparameters (configurations of model merging). We merge these models to obtain a merged model, evaluate the performance of the merged model on different valid tasks as optimization objectives. MM-MO generates new model merging configurations $\boldsymbol{X} = \{\boldsymbol{x}_1, \boldsymbol{x}_2, ...\}$ based on these objective values. This process is repeated until a preset evaluation iteration limit is reached, ultimately yielding the optimal hyperparameters for the merging algorithm. We use the following steps to determine the hyperparameters of the merging algorithm with MM-MO. For more details, please refer to the MM-MO paper [11].

First, several sets of hyperparameter configurations are initialized for the hybrid merging algorithm as the initial candidates. Next, multiple objectives are defined to reflect the performance metrics to be optimized, specifically including task-specific accuracies such as those from MMLU [12] and GSM8K [13]. These metrics are crucial as they ensure the merged model performs well across different diverse tasks. Following this, as qEHVI utilizes surrogate models to approximate objective functions, a probabilistic model is created to predict the performance of unseen configurations. After that, we use an acquisition function to obtain new candidate configurations and apply a weak-to-strong adjustment method to these candidates. Subsequently, configurations are filtered based on Fisher information to select the next generation of model merging hyperparameter settings. The new generation is then evaluated, and the surrogate model is updated to improve its precision in identifying high-potential configurations. After repeating the above steps for roughly 5 iterations, the best hyperparameter settings obtained for the hybrid merging algorithm are returned.

This automated search method tries to find the optimal hyperparameters for hybrid merging algorithm, ensuring that the resulting merged model achieves the best possible performance. Unlike traditional approaches, it eliminates reliance on human knowledge or intuition and effectively replaces the time-consuming grid search process commonly used in previous model merging research [3, 4].

2.4 Improve Merged Model Capabilities with Prompts

Last but not least, as the competition rules allow modifying generation code for dynamic prompt selection, we made specific adjustments to enhance the merged model's responses. Given that the merged model integrates knowledge from multiple specialized fine-tuned models, we aimed to dynamically select prompts that would allow the model to "assume" expertise in relevant domains. For generative tasks, we classified questions into the following domains:

```
"mathematics", "programming", "medicine", "science", "logical reasoning",
"literature", "history", "geography", "economics", "sociocultural",
"technology", "commonsense"
```

We designed a system prompt template that allows the model to respond as a specialized expert. The template is structured as follows:

```
You are a highly knowledgeable, efficient {domain} AI expert.
```

In this approach, the merged model first identifies the domain of the question and then responds from the perspective of an expert in that field, improving the relevance and quality of its answers.

3 Experiments

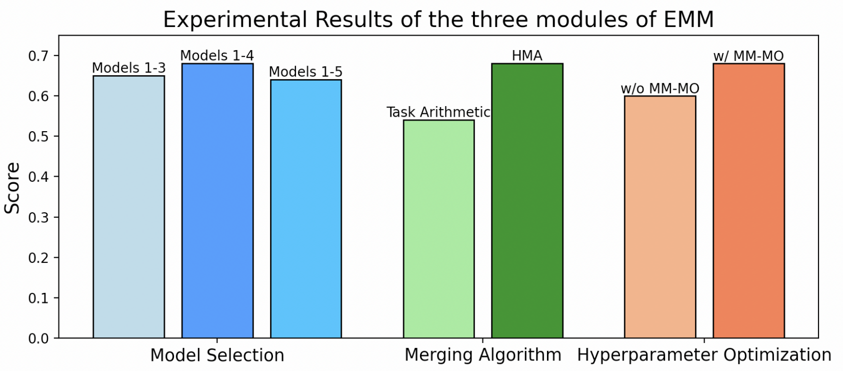

Figure 3: Ablation study of EMM modules, evaluated on the public validation set of the LLM Merging Competition. Models 1-3 refers to the merging of Models 1, 2, and 3, and so forth.

In this section, we present a series of ablation experiment results to individually verify the effectiveness of the three key modules of EMM. For model selection, we compared Models 1-4 (final version) with two variants: Models 1-3 and Models 1-5 (see Section 2.1 for detailed information on model numbering). In Models 1-3, we removed Model 4. This removal somewhat reduced the diversity of candidate models. Conversely, Models 1-5 included an additional model specifically developed for biomedical scenarios, which demonstrated limited compatibility with the other models. For merging algorithm, we utilized HMA, which effectively combines the strengths of each model to build a merged model. For hyperparameter optimization, applying the hyperparameter settings searched by MM-MO further enhances the effectiveness of HMA, resulting in a merged model with improved overall performance.

4 Conclusion

We proposed an efficient model merging (EMM) approach with three key modules: strategic model selection, hybrid merging algorithm, and hyperparameter optimization. Our approach improves the performance of the merged model while maintaining the efficiency of the merging algorithm, which infers that it would be beneficial to consider the whole process of model merging.

References

[1] 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.

[2] 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.

[3] 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.

[4] 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.

[5] Derek Tam, Margaret Li, Prateek Yadav, Rickard Brüel Gabrielsson, Jiacheng Zhu, Kristjan Greenewald, Mikhail Yurochkin, Mohit Bansal, Colin Raffel, and Leshem Choshen. Llm merging: Building llms efficiently through merging. In *NeurIPS 2024 Competition Track*, 2024.

[6] Takuya Akiba, Makoto Shing, Yujin Tang, Qi Sun, and David Ha. Evolutionary optimization of model merging recipes. *arXiv preprint arXiv:2403.13187*, 2024.

[7] AI@Meta. Llama 3 model card. 2024. URL `https://github.com/meta-llama/llama3/blob/main/MODEL_CARD.md`.

[8] Shenzhi Wang, Yaowei Zheng, Guoyin Wang, Shiji Song, and Gao Huang. Llama3-8b-chinese-chat (revision 6622a23), 2024. URL `https://huggingface.co/shenzhi-wang/Llama3-8B-Chinese-Chat`.

[9] Peter Devine. Tagengo: A multilingual chat dataset. *arXiv preprint arXiv:2405.12612*, 2024.

[10] Clémentine Fourrier, Nathan Habib, Alina Lozovskaya, Konrad Szafer, and Thomas Wolf. Open llm leaderboard v2. `https://huggingface.co/spaces/open-llm-leaderboard/open_llm_leaderboard`, 2024.

[11] Bingdong Li, Zixiang Di, Yanting Yang, Hong Qian, Peng Yang, Hao Hao, Ke Tang, and Aimin Zhou. It's morphing time: Unleashing the potential of multiple llms via multi-objective optimization. *arXiv preprint arXiv:2407.00487*, 2024.

[12] Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. Measuring massive multitask language understanding. *arXiv preprint arXiv:2009.03300*, 2020.

[13] 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. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*, 2021.