Surrogate Benchmarks for Model Merging Optimization

Rio Akizuki[1] **Yuya Kudo**[1] **Nozomu Yoshinari**[1] **Yoichi Hirose**[1] **Toshiyuki Nishimoto**[1] **Kento Uchida**[1] **Shinichi Shirakawa**[1]

[1]Yokohama National University

Abstract Model merging techniques aim to integrate the abilities of multiple models into a single model. Most model merging techniques have hyperparameters, and their setting affects the performance of the merged model. Because several existing works show that tuning hyperparameters in model merging can enhance the merging outcome, developing hyperparameter optimization algorithms for model merging is a promising direction. However, its optimization process is computationally expensive, particularly in merging LLMs. In this work, we develop surrogate benchmarks for optimization of the merging hyperparameters to realize algorithm development and performance comparison at low cost. We define two search spaces and collect data samples to construct surrogate models to predict the performance of a merged model from a hyperparameter. We demonstrate that our benchmarks can predict the performance of merged models well and simulate optimization algorithm behaviors.

1 Introduction

Model merging (Yang et al., 2024a) is a promising approach to build a powerful single model from multiple separate models without accessing large datasets and requiring expensive computations. Model merging techniques have succeeded in enhancing the ability of large language models (LLMs) by merging multiple models fine-tuned by different downstream tasks. Most model merging techniques have hyperparameters to set before model merging. Because these hyperparameters affect the performance of merged models, tuning such hyperparameters can maximize the model merging capabilities. Akiba et al. (2025) proposed the evolutionary model merging that optimizes hyperparameters in model merging using an evolutionary algorithm. They used the covariance matrix adaptation evolution strategy (CMA-ES) (Hansen, 2006, 2023) and succeeded in finding the superior merging hyperparameters and building high-performance models.

We call the approach for optimizing hyperparameters in model merging techniques *model merging optimization*. Model merging optimization is a kind of automated machine learning (AutoML) task. The literature on evolutionary model merging (Akiba et al., 2025) shows the effectiveness and potential of the model merging optimization approach using the standard CMA-ES. Therefore, the development of sophisticated model merging optimization algorithms is a promising direction in the AutoML community. However, the computational cost of model merging optimization is relatively high, as with hyperparameter optimization and neural architecture search (NAS).[1] This large computational load will burden the development of new model merging optimization algorithms and increase the cost of algorithm comparison.

The benchmarks for hyperparameter optimization and NAS (Eggensperger et al., 2015; Ying et al., 2019; Dong and Yang, 2020; Hirose et al., 2021; Zela et al., 2022) greatly contribute to algorithm

[1]For example, the evaluation of a merging configuration for DARE-TIES (Akiba et al., 2025) took about five minutes on an NVIDIA A100 (40GB) in our implementation. Although there is work (Mencattini et al., 2025) to reduce the computational cost of evolutionary model merging by reducing the evaluation dataset and using a performance estimator, it still requires about 3.5 minutes on a single NVIDIA 4090 with 24GB of VRAM. Therefore, when yielding 1,000 merge evaluations, one run of the optimization algorithm requires more than 58 hours.

development and evaluation. There are two types of benchmarks: tabular and surrogate benchmarks, where tabular benchmarks provide table lookup for hyperparameter settings and their evaluations through prior exhaustive search, and surrogate benchmarks construct a regression model that returns the performance values from a hyperparameter setting using a sampled actual evaluation dataset. While tabular benchmarks can provide exact hyperparameter evaluations, creating them in continuous or large search spaces is intractable. Surrogate benchmarks can be constructed even in a continuous and large search space, while the provided hyperparameter evaluation values are predicted by a certain model. Referring to the success of benchmarks for hyperparameter optimization and NAS, the benchmark for model merging optimization will also be essential for further algorithm development, which will enable us to compare model merging optimization methods at low cost and realize a fair and reproducible comparison.

He et al. (2025) have proposed a benchmark suite for evaluating model merging techniques, which provides standardized fine-tuning models and evaluation protocols. However, evaluating the performance of merged models requires model merging computation and LLM inferences. Therefore, it cannot reduce the cost of model merging optimization.

In this work, we construct a surrogate benchmark to significantly reduce the evaluation cost of model merging optimization. To our knowledge, this is the first surrogate benchmark for model merging optimization. We collect the paired data of the hyperparameter for model merging and its evaluation values, and construct the surrogate model that predicts the evaluation values from a given hyperparameter. We evaluate the proposed surrogate benchmark for model merging optimization, termed `SMM-Bench`, and demonstrate the use of our surrogate benchmark. The code of `SMM-Bench` will be made available at `https://github.com/shiralab/SMM-Bench`.

2 Surrogate Model Merging Benchmark (SMM-Bench)

Akiba et al. (2025) optimized model merging configurations in two model merging settings: parameter space (PS) and data flow space (DFS) merging. For PS merging, where the parameters of the multiple source models are aggregated, the continuous hyperparameters in DARE-TIES (Yadav et al., 2024; Yu et al., 2024) were optimized. For DFS merging, the merged model is constructed by stacking source models' layers, and binary variables for the choice of layers and continuous variables for input scaling were optimized. Because the hyperparameters in model merging contain continuous variables and can be high-dimension, a surrogate benchmark is a reasonable choice. We introduce surrogate benchmarks for PS and DFS merging as `SMM-Bench-PS` and `SMM-Bench-DFS`.

We use Japanese mathematics as a task to evaluate merged models as in Akiba et al. (2025). The datasets, `gsm8k-ja-test_250_1319` (denoting `gsm8k-ja` in short) (Cobbe et al., 2021; Akiba et al., 2025) and the Japanese test set of `MGSM` (Shi et al., 2023), are used to calculate objective values in optimization and final test score, respectively. To evaluate the ability to solve mathematical tasks and provide answers in Japanese, we calculate accuracy, defined as the ratio of correct answers and reasoning texts in Japanese. This evaluation protocol is the same as in Akiba et al. (2025).

2.1 SMM-Bench-PS

Search Space Design. Tuning the hyperparameters for each layer, called layer-wise merging, has the potential for performance improvement (Yang et al., 2024b), while it increases the number of hyperparameters to be tuned. We focus on layer-wise merging using two source models and the simple merging method of task arithmetic (Ilharco et al., 2023). We use `shisa-gamma-7b-v1`[2] and `WizardMath-7B-V1.1` (Luo et al., 2025) as source LLMs for PS merging. These LLMs are fine-tuned from `Mistral-7B-v0.1` (Jiang et al., 2023) and consist of 32 layers. Task arithmetic has a hyperparameter `weight`, the weighting factor of a task vector. Considering layer-wise merging,

[2] `https://huggingface.co/augmxnt/shisa-gamma-7b-v1`

different hyperparameters can be specified for each source model's layer, resulting in 64 design variables. We restrict the search space to $[0, 1]^{64}$.

Data Collection. We created a dataset of merging hyperparameters and their evaluation values on `gsm8k-ja` and `MGSM` based on three strategies: random sampling, CMA-ES, and the tree-structured Parzen estimator (TPE) (Bergstra et al., 2011). Random sampling collected 64,000 data uniformly at random. We ran CMA-ES 13 times for 188 generations with the default hyperparameter setting and TPE implemented in Optuna (Akiba et al., 2019) 12 times for 300 iterations with a batch size of 8. In addition, we uniformly randomly sampled 1,500 data in the model-wise merging setting, i.e., sampling from a two-dimensional subspace. We collected 133,404 data points in total.

Surrogate Model Construction. We split the dataset into training and test sets with a 9:1 ratio. Separate surrogate models predicting the `gsm8k-ja` and `MGSM` scores from the merging hyperparameters are trained using the training set. We used LightGBM (Ke et al., 2017) as the surrogate model because it exhibited good performance on the surrogate NAS benchmark (Zela et al., 2022). The hyperparameters in LightGBM are optimized using Optuna with five-fold cross-validation. The best-performing cross-validated models are used as our surrogate model by averaging the five models' outputs. Table 1 shows the R^2 score and the Kendall's Tau coefficient (KT) of the predictions made by the surrogate models for the test set. We observe that our surrogate models achieved good prediction performance for both the `gsm8k-ja` and `MGSM` scores.

Table 1: The predictive performance of surrogate models.

	Dataset	R^2	KT
PS	`gsm8k-ja`	0.950	0.883
	`MGSM`	0.921	0.791
DFS	`gsm8k-ja`	0.962	0.863
	`MGSM`	0.957	0.839

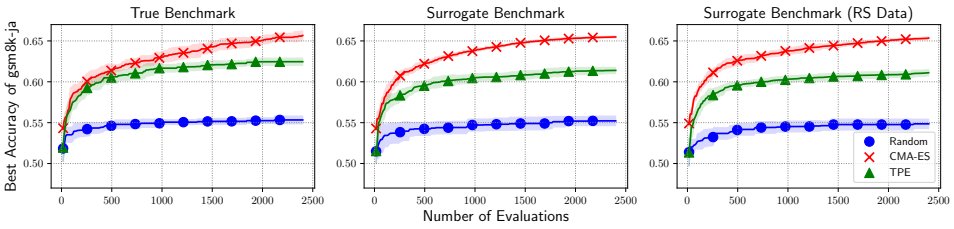
2.2 SMM-Bench-DFS

Search Space Design. Referring to Akiba et al. (2025), we define a mixed category-continuous search space for DFS merging. We use `EvoLLM-JP-v1-7B`,[3] called `model A`, and `shisa-gamma-7b-v1`, called `model B`, as the source models. These source models consist of 32 layers. We construct a merged model by inserting up to 32 layers selected from `model A` and `model B` between the 31st and 32nd layers of `model A`. The i-th layer of the 32 potential inserted layers is selected from three options: the i-th layer of `model A`, the i-th layer of `model B`, and without insertion. This layer insertion is determined by a 32-dimensional categorical variable with three categories. In addition, we introduce layer input scaling factors as hyperparameters to mitigate the input distribution shift (Akiba et al., 2025). We fix the first layer's scaling of the merged model to 1.0 and treat those for the other 63 layers, including potentially inserted layers, as hyperparameters. The range of scaling factors is $[0.4, 1.5]$. As a result, the merging hyperparameters consist of 32 categorical variables and 63 continuous variables.

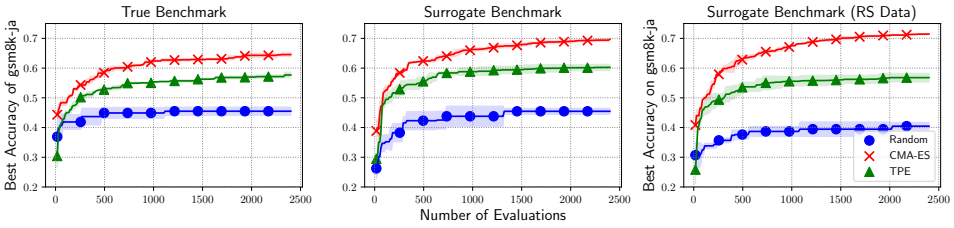
Data Collection. We totally collected 40,913 data points, where 22,286 data were sampled uniformly at random from the search space. We ran CatCMA (Hamano et al., 2024; Nomura and Shibata, 2024) three times for 177 generations with the default setting and TPE four times for 300 iterations with a batch size of 8.

Surrogate Model Construction. The surrogate models are trained in the same procedure as `SMM-Bench-PS`. The R^2 score and Kendall's Tau coefficient (KT) for the test set are also displayed in Table 1. Our surrogate models achieved high predictive performance, while the dataset size was smaller than that of `SMM-Bench-PS`.

[3] `https://huggingface.co/SakanaAI/EvoLLM-JP-v1-7B`



(a) Search space for PS merging (`SMM-Bench-PS`)



(b) Search space for DFS merging (`SMM-Bench-DFS`)

Figure 1: Transitions of best accuracy of `gsm8k-ja` on true benchmark (left), surrogate benchmark (middle), and surrogate benchmark when using only a random sampled dataset (right). Random search, CMA-ES, and TPE are compared. The mean and standard deviation over ten and three runs are plotted in the search spaces for PS and DFS merging, respectively.

3 Evaluation of SMM-Bench

We simulate optimization trajectories using `SMM-Bench`. Figures 1a and 1b show the transitions of the best accuracy for three algorithms on the search spaces for PS and DFS merging, respectively. In these figures, we display the transitions on true benchmark (i.e., using actual evaluations of merged models) and surrogate benchmarks constructed using all dataset and only random sampling data. We observe that our surrogate benchmarks can capture the behavior of algorithms on true benchmarks, although the performance prediction on the search space for DFS merging tends to overestimate. In addition, our surrogate benchmarks work well even when using only random sampled data.

4 Benchmark Demonstration Using SMM-Bench-PS

We ran two algorithms, separable CMA-ES (Sep-CMA) (Ros and Hansen, 2008; Nomura and Shibata, 2024) with the default setting and differential evolution (DE) (Storn and Price, 1997) implemented in `SciPy` with a population size of 64, on `SMM-Bench-PS`. Figure 2 shows the transitions of accuracy for `gsm8k-ja` and `MGSM` datasets. We plotted the accuracy of the best solutions for `gsm8k-ja` that is the objective value in optimization. The accuracy of `MGSM` is the test performance of the best solutions. We observe that Sep-CMA outperforms DE on both `gsm8k-ja` and `MGSM` datasets, although the test performance difference is not so significant. This experimental comparison can be conducted in several minutes on a laptop owing to the surrogate benchmark. However, it requires many GPU days if it uses the actual evaluation of merged LLMs. Our surrogate benchmarks will be useful for hyperparameter tuning and comprehensive evaluation for optimizers.

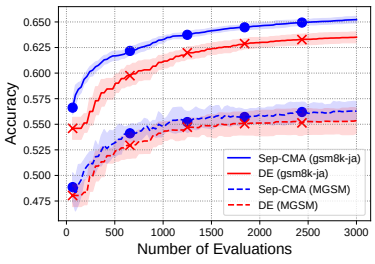


Figure 2: Performance of Sep-CMA and DE on `SMM-Bench-PS`.

5 Conclusion

We have proposed and evaluated surrogate benchmarks for model merging optimization. We also demonstrated a performance comparison of optimization algorithms not used for data collection on our benchmark. We believe that our surrogate benchmarks will contribute to algorithm development for model merging optimization and reproducible algorithm comparison.

Acknowledgements. We used ABCI 3.0 provided by AIST and AIST Solutions with support from "ABCI 3.0 Development Acceleration Use."

References

Akiba, T., Sano, S., Yanase, T., Ohta, T., and Koyama, M. (2019). Optuna: A next-generation hyperparameter optimization framework. In *25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining (KDD)*.

Akiba, T., Shing, M., Tang, Y., Sun, Q., and Ha, D. (2025). Evolutionary optimization of model merging recipes. *Nature Machine Intelligence*, 7:195–204.

Bergstra, J., Bardenet, R., Bengio, Y., and Kégl, B. (2011). Algorithms for hyper-parameter optimization. In *Advances in Neural Information Processing Systems*.

Cobbe, K., Kosaraju, V., Bavarian, M., Chen, M., Jun, H., Kaiser, L., Plappert, M., Tworek, J., Hilton, J., Nakano, R., Hesse, C., and Schulman, J. (2021). Training verifiers to solve math word problems. *arXiv:2110.14168*.

Dong, X. and Yang, Y. (2020). NAS-Bench-201: Extending the scope of reproducible neural architecture search. In *International Conference on Learning Representations (ICLR)*.

Eggensperger, K., Hutter, F., Hoos, H., and Leyton-Brown, K. (2015). Efficient benchmarking of hyperparameter optimizers via surrogates. In *AAAI Conference on Artificial Intelligence*.

Hamano, R., Saito, S., Nomura, M., Uchida, K., and Shirakawa, S. (2024). CatCMA: Stochastic optimization for mixed-category problems. In *Genetic and Evolutionary Computation Conference (GECCO)*.

Hansen, N. (2006). The CMA evolution strategy: A comparing review. In *Towards a New Evolutionary Computation: Advances in the Estimation of Distribution Algorithms*, pages 75–102. Springer Berlin Heidelberg.

Hansen, N. (2023). The CMA evolution strategy: A tutorial. *arXiv:1604.00772*.

He, Y., Zeng, S., Hu, Y., Yang, R., Zhang, T., and Zhao, H. (2025). MergeBench: A benchmark for merging domain-specialized LLMs. *arXiv:2505.10833*.

Hirose, Y., Yoshinari, N., and Shirakawa, S. (2021). NAS-HPO-Bench-II: A benchmark dataset on joint optimization of convolutional neural network architecture and training hyperparameters. In *Asian Conference on Machine Learning (ACML)*.

Ilharco, G., Ribeiro, M. T., Wortsman, M., Schmidt, L., Hajishirzi, H., and Farhadi, A. (2023). Editing models with task arithmetic. In *International Conference on Learning Representations (ICLR)*.

Jiang, A. Q., Sablayrolles, A., Mensch, A., Bamford, C., Chaplot, D. S., de las Casas, D., Bressand, F., Lengyel, G., Lample, G., Saulnier, L., Lavaud, L. R., Lachaux, M.-A., Stock, P., Scao, T. L., Lavril, T., Wang, T., Lacroix, T., and Sayed, W. E. (2023). Mistral 7B. *arXiv:2310.06825*.

Ke, G., Meng, Q., Finley, T., Wang, T., Chen, W., Ma, W., Ye, Q., and Liu, T. (2017). LightGBM: a highly efficient gradient boosting decision tree. In *Advances in Neural Information Processing Systems*.

Luo, H., Sun, Q., Xu, C., Zhao, P., Lou, J.-G., Tao, C., Geng, X., Lin, Q., Chen, S., Tang, Y., and Zhang, D. (2025). WizardMath: Empowering mathematical reasoning for large language models via reinforced evol-instruct. In *International Conference on Learning Representations (ICLR)*.

Mencattini, T., Minut, A. R., Crisostomi, D., Santilli, A., and Rodolà, E. (2025). MERGE^3: Efficient evolutionary merging on consumer-grade GPUs. In *International Conference on Machine Learning (ICML)*.

Nomura, M. and Shibata, M. (2024). cmaes: A simple yet practical python library for CMA-ES. *arXiv preprint arXiv:2402.01373*.

Ros, R. and Hansen, N. (2008). A simple modification in CMA-ES achieving linear time and space complexity. In *Parallel Problem Solving from Nature (PPSN X)*.

Shi, F., Suzgun, M., Freitag, M., Wang, X., Srivats, S., Vosoughi, S., Chung, H. W., Tay, Y., Ruder, S., Zhou, D., Das, D., and Wei, J. (2023). Language models are multilingual chain-of-thought reasoners. In *International Conference on Learning Representations (ICLR)*.

Storn, R. and Price, K. (1997). Differential evolution – a simple and efficient heuristic for global optimization over continuous spaces. *Journal of Global Optimization*, 11(4):341–359.

Yadav, P., Tam, D., Choshen, L., Raffel, C. A., and Bansal, M. (2024). TIES-Merging: Resolving interference when merging models. In *Advances in Neural Information Processing Systems (NeurIPS)*.

Yang, E., Shen, L., Guo, G., Wang, X., Cao, X., Zhang, J., and Tao, D. (2024a). Model merging in LLMs, MLLMs, and beyond: Methods, theories, applications and opportunities. *arXiv:2408.07666*.

Yang, E., Wang, Z., Shen, L., Liu, S., Guo, G., Wang, X., and Tao, D. (2024b). AdaMerging: Adaptive model merging for multi-task learning. In *International Conference on Learning Representations (ICLR)*.

Ying, C., Klein, A., Christiansen, E., Real, E., Murphy, K., and Hutter, F. (2019). NAS-Bench-101: Towards reproducible neural architecture search. In *International Conference on Machine Learning (ICML)*.

Yu, L., Yu, B., Yu, H., Huang, F., and Li, Y. (2024). Language models are Super Mario: Absorbing abilities from homologous models as a free lunch. In *International Conference on Machine Learning (ICML)*.

Zela, A., Siems, J., Zimmer, L., Lukasik, J., Keuper, M., and Hutter, F. (2022). Surrogate NAS benchmarks: Going beyond the limited search spaces of tabular NAS benchmarks. In *International Conference on Learning Representations (ICLR)*.