Merging in a Bottle: Differentiable Adaptive Merging (DAM) and the Path from Averaging to Automation

Anonymous NACCL submission

Abstract

Model merging allows AI systems to harness the unique strengths of individual language models, creating a balanced set of capabilities without the need for extensive retraining. However, this integration is often complex due to variations in training techniques and fine-tuning approaches, which demand specialized expertise and iterative refinement to achieve optimal results. Our paper explores model merging techniques across a spectrum of complexity, examining where automated methods like evolutionary strategies stand compared to hyperparameter-driven approaches such as DARE, TIES-Merging and simpler methods like Model Soups. In addition, we introduce Differentiable Adaptive Merging (DAM)[1], an efficient, adaptive merging approach as an alternative to evolutionary merging that optimizes model integration through scaling coefficients, minimizing computational demands. Our findings reveal that even simple averaging methods, like Model Soups, perform competitively when model similarity is high, underscoring each technique's unique strengths and limitations.

1 Introduction

As demand for versatile AI systems grows, merging Large Language Models (LLMs) with specialized capabilities, such as multilingual or domain-specific knowledge, has become essential. Effective model merging allows systems to utilize individual models' strengths without extensive retraining and helps mitigate catastrophic forgetting, preserving learned knowledge from each model (Sukhbaatar et al., 2024; Siriwardhana et al., 2024; Labrak et al., 2024). However, due to varying training and fine-tuning processes, model merging remains complex, often requiring significant expertise and iterative tuning to achieve balanced integration.

Model merging techniques fall into two main categories: manual and automated. Manual methods like Trim, Elect, Sign, & Merge (TIES-Merging) (Yadav et al., 2024), and Spherical Linear intERPolation (SLERP)[2] focus on merging model parameters directly without any reliance on data, making them computationally efficient but requiring manual tuning, which can limit scalability. Automated methods, such as Adaptive Model Merging (AdaMerging) (Yang et al., 2023) and Evolutionary Model Merging (Akiba et al., 2024), use representative data to optimize parameter adjustments, enabling fine-grained control but at a higher computational cost. Our analysis spans basic averaging to advanced automated approaches, detailing their unique strengths and trade-offs.

Building on these insights, we introduce Differentiable Adaptive Merging (DAM), a new approach developed as a more efficient alternative to compute-heavy evolutionary strategies. In this paper, we compare DAM with established techniques, including DRops And REscales with TIES sign election (DARE-TIES) (Yu et al., 2024), evolutionary merging (Akiba et al., 2024), and Model Soups (Wortsman et al., 2022), aiming to benchmark these methods against each other. Our objective is to analyze how these different merging families perform across specific scenarios, showcasing where each technique excels or faces limitations. By positioning DAM as a cost-effective yet powerful and scalable automatic merging alternative.

Our contributions are as follows:

- We conduct an extensive comparative analysis of model merging techniques, ranging from simple averaging methods to advanced automated approaches.
- We introduce DAM as a novel and efficient

[1] We made the implementation code for DAM available on Google Drive: https://drive.google.com/file/d/1C3UMEY6ehDeb-qzkjK-dcazdZtzqlRum/view?usp=sharing

[2] https://github.com/Digitous/LLM-SLERP-Merge

alternative to evolutionary merging, highlighting its effectiveness in reducing computational overhead while achieving competitive performance.

- Our findings underscore that simple methods, such as model averaging, can sometimes outperform more complex techniques, challenging established assumptions in the field and offering practical insights for researchers and practitioners.

2 Related Work

This section provides an overview of the related work.

2.1 Data-Free Merging

Data-Free Merging methods aim to merge model parameters through computationally efficient linear operations, avoiding data-driven inference. Model Soups (Wortsman et al., 2022) is among the first to show the feasibility of weight averaging for pre-trained LLMs, though it can suffer from parameter conflicts. SLERP[3] improves on this by interpolating between two models on a spherical path, but it is limited to merging only two models at a time.

More advanced techniques like TIES-Merging (Yadav et al., 2024) and Drop and rEscaLe via sampLing with mAgnitude (DELLA-Merging) (Deep et al., 2024) add steps like parameter trimming, sign conflict resolution, and pruning to enhance Model Soups. DARE (Yu et al., 2024) further refines merging by sparsifying delta parameters, reducing redundancy for improved efficiency. However, these methods typically require manual parameter tuning, which can hinder scalability in complex tasks.

In practice, the open-source community often uses methods such as DARE-TIES (Yu et al., 2024), TIES (Yadav et al., 2024), and SLERP with recommended hyperparameter guidelines. However, balancing weights across layers remains challenging and typically involves extensive trial and error to fine-tune layer weightings and parameter settings. The lack of full automation and transparency in these approaches underscores the need for more efficient, scalable, and interpretable model merging techniques.

2.2 Data-Informed Merging

Recognizing the limitations of manual parameter tuning, researchers have explored automated model merging techniques. These techniques leverage representative data during inference to better balance the trade-offs between the input models being merged. By utilizing the additional insights gained during inference, they facilitate the automatic selection of hyperparameters that would otherwise need to be approximated in data-free methods. Given these methods are fully automated, they can optimize more granular hyperparameters than the data-free methods, such as per-layer or even per-feature weighting parameters.

AdaMerging (Yang et al., 2023) is an advanced data-informed technique that automates model merging by focusing on minimizing the entropy associated with both the input models and their respective datasets. The process begins by calculating task vectors, which represent the difference in weights between the input models and the base model. Coefficients are then assigned and optimized for each task vector, determining the impact each input model has on the final merged model. AdaMerging adjusts these coefficients in an unsupervised manner, minimizing entropy on multi-task, unlabeled test data, which serves as a surrogate objective for minimizing loss on the full dataset. This optimization not only balances task vectors but also ensures the merged model's predictions are more deterministic and robust.

ZipLoRA (Shah et al., 2023) is a technique developed to merge independently trained Low-Rank Adaptations (LoRAs) (Hu et al., 2021) by optimizing coefficients to reduce interference between content and style representations. They demonstrate that orthogonality between the columns of the LoRAs is strongly correlated with successful merging that minimizes interference. Primarily applied in the image generation domain with diffusion models, ZipLoRA preserves subject fidelity and style accuracy without requiring extensive manual adjustments.

Evolutionary Model Merging (Akiba et al., 2024) leverages evolutionary algorithms to optimize merging parameters across both the parameter space and the data flow (layer) space. Unlike AdaMerging, which assigns coefficients to task vectors, evolutionary techniques define merging coefficients on a per-layer basis. The process involves initializing a population of potential solutions, each representing a different set of merging coefficients, and iteratively refining these through selection, mutation, and recombination based on task-specific

[3] https://github.com/Digitous/LLM-SLERP-Merge

performance metrics. Although evolutionary merging can produce highly effective merged models, it is computationally expensive. The need to evaluate numerous combinations of merging coefficients across large models with many layers demands substantial computational resources and time, making it less practical in scenarios with limited resources or time constraints.

2.3 Model Alignment

Another area of research is model alignment, which addresses the challenge of merging models with different initializations, particularly mapping equivalent features to consistent positions in weight matrices to avoid interference. Although approaches like ZipIt (Stoica et al., 2023), Git Re-Basin (Ainsworth et al., 2022), and Optimal Transport (OT) Fusion (Singh and Jaggi, 2020) have shown effectiveness, alignment for LLMs remains in its early stages. Git Re-Basin and OT Fusion aim to align tasks by parameter permutation and soft matching, respectively, with ZipIt excelling in multi-task scenarios. However, merging LLMs fine-tuned from the same initialization is more practical and commonly assumed, which simplifies alignment challenges in our scenario.

3 Differentiable Adaptive Merging (DAM)

In this section, we introduce DAM, a novel approach designed to merge multiple LLMs efficiently and effectively. DAM leverages a data-informed methodology to learn cost-efficient scaling coefficients for each model prior to merging, optimizing the integration process. This method can be applied to all linear layers, embedding layers, and layer normalization layers within the models.

3.1 Mathematical Formulation

The core idea of DAM is to find the optimal scaling coefficients for multi-task merging. By scaling the columns of the source models' weight matrices, we can effectively adjust the input features, ensuring that the merged model leverages the strengths of each individual model. For a given layer l in the model, let W_i^l represent the weight matrix of the i-th model. The goal is to find optimal scaling coefficients c_{ij}^l for each column j in the weight matrix of each model such that the merged weight matrix W^l is given by:

$$W^l = \sum_{i=1}^{N} W_i^l \cdot C_i^l$$

where N is the number of models being merged, and C_i^l is a diagonal matrix with the scaling coefficients c_{ij}^l on the diagonal. This ensures that each column of the weight matrix is scaled individually.

3.2 Column-wise Scaling

To provide a more intuitive understanding, consider a weight matrix W_i^l of layer l in model i with dimensions $M \times N$. Each column j in this weight matrix has a corresponding scaling factor c_{ij}^l. The merged weight matrix W^l can then be expressed as:

$$W^l = \sum_{i=1}^{N} W_i^l \cdot \text{diag}(c_{i1}^l, c_{i2}^l, \ldots, c_{iN}^l)$$

where $\text{diag}(c_{i1}^l, c_{i2}^l, \ldots, c_{iN}^l)$ is a diagonal matrix with the scaling factors on the diagonal. These factors are learned to optimally scale the input features, effectively adjusting the contribution of each model's weight matrix to the merged model. See Figure 1.

For example, consider the MLP (Multi-Layer Perceptron) layers in three models. Each model has its own weight matrix for a given layer. By learning the optimal scaling factors for each column of these weight matrices, we can merge the models in a way that scales the input features appropriately, ensuring that the merged model performs optimally across tasks.

The DAM method is applied uniformly across different types of layers in the models: for each linear layer, the weight matrices W_i^l are scaled and merged using the learned coefficients c_{ij}^l; similarly, the embedding matrices in the embedding layers are scaled and merged; and in the layer normalization layers, the normalization parameters are scaled and merged to ensure consistent normalization across the merged model. In our experiments, we found that merging only the linear layers, while retaining the embeddings and normalization weights from the base model, performs best.

3.3 Objective Function for DAM

The objective function for DAM is designed to optimize the scaling coefficients for each model in a way that ensures the best possible merge. The core

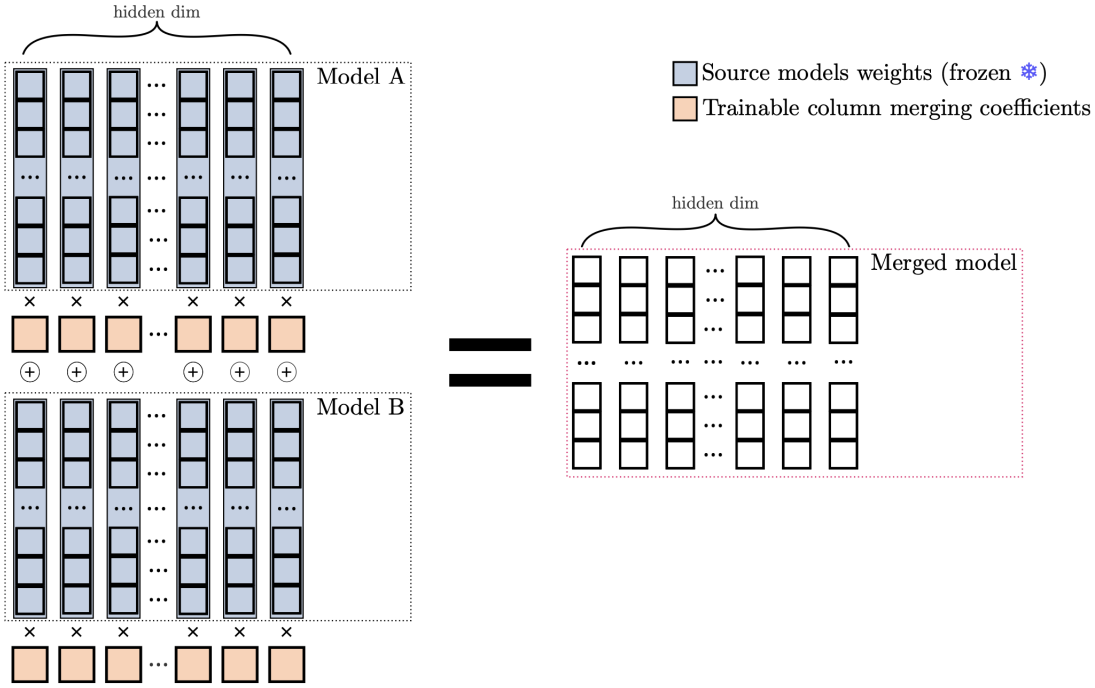

Figure 1: Graphical illustration of activation adjusting via the trainable coefficients in the proposed DAM method.

idea is to balance task performance, regularization, and similarity constraints between coefficients to achieve a robust and efficient merging process. The objective function consists of the following components:

3.3.1 Kullback-Leibler (KL) Divergence Loss

Given three models and their corresponding datasets $D_1, D_2, D_3, .., D_n$, the idea is to minimize the KL divergence between the logits of the merged model and the logits of each individual model on their respective datasets. Let $\mathrm{KL}(P\|Q)$ denote the KL divergence between distributions P and Q. The KL divergence loss for the merged model is given by:

$$\mathcal{L}_{\text{KL}} = \sum_{i=1}^{N} \mathrm{KL}(\text{logits}_{\text{merged}}(D_i) \| \text{logits}_i(D_i))$$

where $\text{logits}_{\text{merged}}(D_i)$ are the logits of the merged model on dataset D_i, and $\text{logits}_i(D_i)$ are the logits of the i-th model on its corresponding dataset. This ensures that the merged model's predictions are aligned with those of the individual models on their respective datasets.

3.3.2 Cosine Similarity Loss

Inspired by ZipLoRA, we add a constraint to reduce the cosine similarity between the scaling coefficients of different models for each layer. This encourages the models to scale the feature space in unique ways, promoting diversity in the merged model. The cosine similarity loss is given by:

$$\mathcal{L}_{\text{cosine}} = \lambda_{\text{cosine}} \sum_{l} \sum_{i<j} \cos(c_{ij}^{l}, c_{ik}^{l})$$

where λ_{cosine} is the regularization coefficient for the cosine similarity loss. This term helps to ensure that the scaling coefficients for different models are not too similar, which can improve the robustness of the merged model.

3.3.3 L1 and L2 Regularization

To ensure stable training and add sparsity to the coefficients, we apply L1 and L2 regularization to the scaling coefficients:

$$\mathcal{L}_{\text{reg}} = \lambda_1 \sum_{i=1}^{N} \sum_{j=1}^{d} \|c_{ij}^{l}\|_1 + \lambda_2 \sum_{i=1}^{N} \sum_{j=1}^{d} \|c_{ij}^{l}\|_2^2$$

where λ_1 and λ_2 are regularization coefficients. L1 regularization encourages sparsity in the scaling coefficients, while L2 regularization ensures that the coefficients remain small and stable during training.

3.3.4 Overall Objective Function

The overall objective function for training the scaling coefficients in DAM combines the KL divergence loss, cosine similarity loss, and regularization terms:

$$\mathcal{L} = \mathcal{L}_{\text{KL}} + \mathcal{L}_{\text{cosine}} + \mathcal{L}_{\text{reg}}$$

This comprehensive objective function ensures that the scaling coefficients are optimized for task performance, regularization, and similarity, leading to a more robust and efficient merging process.

4 Experiment Design

4.1 Benchmarking Different Merging Techniques against DAM

To thoroughly benchmark model merging approaches, we designed experiments focusing on two primary model families, Mistral (Jiang et al., 2023a) and Llama 3 (Dubey et al., 2024). In these experiments, we compare the effectiveness of various merging methods across models with diverse specialized capabilities, including multilingual processing, coding proficiency, and mathematical reasoning. This approach allows us to evaluate how well each merging technique retains and integrates these distinct functionalities within the merged models. Given the proprietary nature of the original datasets used to train each of the candidate models, we applied closely aligned, representative datasets.

4.1.1 Case Study 1: Japanese Language Processing and Mathematical Reasoning

In this case study, we used Mistral-based models specializing in Japanese language and mathematical reasoning to evaluate merging techniques for integrating these capabilities. We focused on DAM's adaptability and compositional generalization in Japanese mathematical reasoning tasks, benchmarking it against DARE-TIES, Model Soups, and evolutionary merging methods.

Selected Models: For this case study, we selected the `shisa-gamma-7b` model (augmxnt, 2023), a Japanese language model trained specifically on Japanese language tasks, demonstrating proficiency in linguistic comprehension and expression in Japanese. Additionally, we utilized the `WizardMath-7B-V1.1` model (Luo et al., 2023) and `Abel-7B-002` model (Chern et al., 2023), both of which are trained on datasets oriented towards mathematical reasoning, equipping them with capabilities in numerical and logical problem-solving. Each of these models is derived from the `Mistral-7B-v0.1` base model (Jiang et al., 2023b), and our merged models retain the embedding and RMSNorm weights from `Mistral-7B-v0.1` to maintain consistency.

Representative Datasets: For DAM training, we used the `Ichikara` Japanese instruction tuning dataset (Sekine et al., 2024), covering the same kind of broad general-purpose Japanese language chat conversations used to train `shisa-gamma-7b`. For mathematical reasoning, we used the `MetaMathQA` (Yu et al., 2023) and `Orca-Math` (Mitra et al., 2024) datasets, covering basic arithmetic, algebraic operations, and logical reasoning to simulate the reasoning tasks likely included in `WizardMath` and `Abel`. The final training dataset comprises 1,729 samples, formatted with the Alpaca instruction template (Taori et al., 2023) to ensure consistency. DAM training was conducted with a learning rate of $2e^{-3}$ and a batch size of 1 to balance linguistic and mathematical capabilities across the merged models.

Evaluation Metrics: To assess the performance of the merged models in Japanese and math, we used the JP Language Model Evaluation Harness[4]. We evaluated Japanese language proficiency of the merged models through a range of metrics for linguistic accuracy and comprehension in Japanese, as shown in Table 1, and we tested mathematical reasoning abilities on the Japanese subset of the Multilingual Grade School Math (MGSM) Benchmark (Shi et al., 2022).

4.1.2 Case Study 2: SQL Coding, German, and Korean Language Processing

In this case study, we expanded our experiments to encompass multilingual and domain-specific models in German, Korean, and SQL, aiming to further evaluate DAM's effectiveness in managing diverse languages and tasks. In this context, we conducted comparative evaluations of DARE-TIES, Model Soups, against DAM, assessing their respective performances across these varied domains.

Selected Models: For this case study, we used the following Llama-3-based models: `Llama-3-SauerkrautLM-8b-Instruct`[5], a Ger-

[4] https://github.com/Stability-AI/lm-evaluation-harness/tree/jp-stable

[5] https://huggingface.co/VAGOsolutions/Llama-3-SauerkrautLM-8b-Instruct

man language model fine-tuned for various linguistic tasks in German; `Llama-3-Open-Ko-8B`[6], specialized in Korean language processing; and `llama-3-sqlcoder-8b`[7], tailored for SQL coding tasks. Each of these models is based on the `Meta-Llama-3-8B`[8] model, with the German and SQL models being initialized from the Instruct variant.

Representative Datasets: For German, we sourced diverse linguistic samples from corpora, using `Fischerboot/another-german-alpaca-dataset`[9], covering comprehension and dialogue tasks. For Korean, we included samples in both formal and informal registers from `lcw99/wikipedia-korean-20240501`[10] for broader contextual understanding. SQL query examples were used from the `wikiSQL` (Zhong et al., 2017), emphasizing syntactical accuracy and command structure. All tasks were formatted using the Alpaca instruction template to maintain consistency (Taori et al., 2023). Training parameters mirrored our prior study, with a learning rate of $(2e^{-3})$ and a batch size of 1, achieving balanced performance across multilingual and domain-specific models.

Evaluation Metrics: To evaluate the language capabilities of the merged models in German, we used the German subset of the Okapi benchmark (Lai et al., 2023), which provides a comprehensive set of tasks for assessing model's language comprehension. For Korean, we employed the Korean Balanced Evaluation of Significant Tasks (KoBEST) benchmark (Jang et al., 2022), which consists of five distinct Korean-language tasks designed to gauge various aspects of language comprehension in Korean, including factual reasoning, causality, word sense disambiguation, contextual inference, and sentiment analysis. For SQL generation, we measured model performance using the SQL-Eval[11] framework, focusing on the syntactical correctness and structural accuracy of the generated queries.

[6]https://huggingface.co/beomi/Llama-3-Open-Ko-8B
[7]https://huggingface.co/defog/llama-3-sqlcoder-8b
[8]https://huggingface.co/meta-llama/Meta-Llama-3-8B
[9]https://huggingface.co/datasets/Fischerboot/another-german-alpaca-dataset
[10]https://huggingface.co/datasets/lcw99/wikipedia-korean-20240501
[11]https://github.com/defog-ai/sql-eval

5 Ablation Studies on the DAM Method

The overall objective function for DAM, as shown in Equation 3.3.4, comprises multiple components, each designed with a specific purpose, as outlined in Section 3.3. Here, we analyzed the impact of each component and examined alternative approaches, based on the results in Table 3.

5.1 Comparison of Different Loss Functions

In addition to KL divergence (Equation 3.3.1), we experimented with mean square error (MSE) and entropy loss as alternatives for output distribution loss in the merging process.

MSE Loss minimizes the difference between the output distribution of the merged model and the input models, calculated as:

$$L_{\text{MSE}} = \frac{1}{N}\sum_{i=1}^{N}\left(\text{logits}_{\text{merged}}(D_i) - \text{logits}_i(D_i)\right)^2$$

where N is the total number of datasets, $\text{logits}_{\text{merged}}(D_i)$ are the logits of the merged model on dataset D_i, and $\text{logits}_i(D_i)$ are the logits of the input model i on dataset D_i.

Entropy Loss encourages more confident predictions from the merged model, following AdaMerging (Yang et al., 2023). This loss is computed as:

$$\mathcal{L}_{\text{entropy}} = \sum_{i=1}^{N} H(\text{logits}_{\text{merged}}(D_i))$$

where $H(\cdot)$ denotes the entropy function. Unlike KL divergence or MSE, entropy loss minimizes the entropy of the merged model's output distribution without directly referencing the input models, aiming to enhance the decisiveness of predictions across datasets.

5.2 Effectiveness of Weight Regularization

In this experiment, we examined the effect of adding cosine similarity loss to the objective function by comparing models trained with KL divergence or entropy loss, with and without this regularization term. Additionally, we tested L1 and L2 regularization within both KL-only and KL + Cosine Similarity loss settings to analyze their impact on model accuracy and consistency. Using the same input models as in Case Study 1 and the JP Language Evaluation Harness framework, we measured the merged model's average accuracy across benchmarks, running each configuration with three random seeds to ensure robustness and consistency.

Table 1: **Case study 1: Performance comparison of different merging methods in japanese Language processing and mathematical reasoning**. This table presents the performance of the merging methods across a range of Japanese language tasks. The average (Avg) column serves as a comprehensive indicator of each method's overall effectiveness in handling these tasks.

	JP Language Model Evaluation Harness									
Method	JAQKET	JComQA	JCoLA	XWino	JNLI	MGSM	JSQuAD	MARC	XLSum	Avg
DAM	0.64	0.81	0.59	0.80	0.47	0.40	0.67	0.94	**0.22**	**0.62**
Evolutionary Merging	**0.71**	0.80	0.57	**0.81**	0.34	**0.44**	**0.76**	**0.96**	0.21	**0.62**
Model Soups	0.64	0.81	0.60	0.79	0.42	0.42	0.66	**0.96**	**0.22**	0.61
DARE-TIES	0.39	**0.82**	**0.61**	0.79	**0.62**	0.38	0.19	**0.96**	0.20	0.55

Table 2: **Case study 2: Performance comparison of different merging methods in SQL coding, Korean, and German language processing**. This table presents the effectiveness of the merging techniques evaluated on the Okapi (German), KoBEST (Korean), and SQL-Eval benchmarks, along with an overall average (Avg) score across these three benchmarks.

Merging method	German	Korean	SQL	Avg
DAM	**0.4434**	0.5933	**0.6125**	**0.5497**
Model Soups	0.4337	0.6121	0.5969	0.5476
DARE-TIES	0.4293	**0.6307**	0.5563	0.5388

Table 3: **Ablation studies on the DAM method.** This table presents model evaluation results for various configurations of the DAM objective function, assessed using the JP Language Model Evaluation Harness. Results include the average (Avg) and standard deviation (Std) across three runs with different random seeds.

	JP Language Model Evaluation Harness	
DAM Objective Function	Avg	Std
KL	**0.62**	0.034
KL + Reg	0.60	-
KL + Cosine	**0.62**	**0.007**
KL + Cosine + Reg	**0.62**	**0.007**
MSE + Cosine	0.53	-
Entropy	0.60	0.019
Entropy + Cosine	0.60	0.027

6 Results and Discussion

This section presents a detailed analysis of our results, highlighting the comparative performance of DAM, Evolutionary Merging, DARE-TIES, and Model Soups. We discuss insights from the case studies on Japanese and Mathematics domains (Table 1) and the German, SQL, and Korean domains (Table 2), as well as our findings from ablation studies (Table 3) on DAM's objective function components.

6.1 Performance Analysis of DAM Compared to Different Merging Methods

6.1.1 Case study 1: Japanese Language Processing and Mathematical Reasoning

Our experiments in the Japanese and Mathematics domains reveal clear distinctions among merging methods in handling language and reasoning capabilities.

As shown in Table 1, DAM outperforms other methods on average, effectively balancing performance across tasks involving both Japanese comprehension (e.g., `JAQKET`, `JComQA`), and mathematical reasoning (`MGSM`). Evolutionary Merging, while competitive with a 0.62 average score, requires considerable computational resources, highlighting DAM's advantage in balancing performance with efficiency.

DARE-TIES and Model Soups both show moderate effectiveness, with DARE-TIES achieving higher scores in specific tasks such as `JComQA` (0.82) and `MGSM` (0.62). However, their average scores (0.55 and 0.61, respectively) remain below that of DAM, underscoring DAM's adaptability across diverse tasks without intensive hyperparameter tuning. Model Soups, which is a simple linear averaging approach, performs relatively well, suggesting that linear merging can sometimes rival more complex methods when the merged models are sufficiently similar. Compared to DARE-TIES, Model Soups yields more balanced results across metrics, while DARE-TIES exhibits considerably reduced performance on reading comprehension (`JSQuAD`) and open-domain question answering (`JAQKET`) tasks.

6.1.2 Case Study 2: SQL Coding, German, and Korean Language Processing

In the case study on German, SQL, and Korean domains, DAM's effectiveness extends across multilingual and structured data processing tasks, as

shown in Table 2.

DAM again leads in performance, particularly in German and SQL tasks, with average scores of 0.4434 and 0.6125, respectively. The DAM model performs well across German linguistic tasks with different complexity levels, showing a slight edge over DARE-TIES and Model Soups. In SQL generation tasks, DAM's performance also surpasses Model Soups, which scores 0.5969. DARE-TIES achieves higher effectiveness in Korean tasks (0.6307), likely attributed to differences in model initialization; unlike the other two models, the Korean model was initialized from the Llama 3 base model rather than the Llama 3 Instruct variant. DARE-TIES's sparsification strategy appears beneficial for minimizing interference in cases where task vectors diverge significantly.

6.2 Additional Key Insights

This section highlights key insights gained from both the primary experiments and the ablation studies.

6.2.1 Effectiveness of DAM Compared to Evolutionary Merging

DAM demonstrates clear advantages over Evolutionary Merging in terms of efficiency and practicality. Evolutionary Merging methods, which involve iterative searches through parameter configurations, require substantial computational resources, whereas DAM's gradient-based optimization streamlines this process. As evidenced by Table 3, DAM achieves comparable or superior results with far fewer resources, particularly evident in its performance with Japanese and Korean tasks. DAM's KL-divergence-based configuration, for instance, not only matches but sometimes surpasses Evolutionary Merging in compositional generalization across these languages. This computational efficiency and high performance make DAM a valuable tool for multilingual settings with limited resources, where retaining specific language attributes is essential.

6.2.2 Insights on Model Soups and Simplicity in Model Averaging

An intriguing outcome of the experiments is the effectiveness of simple averaging methods like Model Soups in multilingual settings. While Model Soups lacks the fine-grained control of DAM or DARE-TIES, it nonetheless achieves competitive results in both Japanese and Korean tasks. For example, Model Soups shows an average score of 0.61 in Japanese and 0.5969 in SQL, as seen in Table 1 and Table 2. This performance aligns with findings on linear interpolation, indicating that in cases where models share significant linguistic similarities, simple averaging methods like Model Soups can offer satisfactory results with minimal computational cost. Model Soup's efficiency and ease of implementation make it a viable option when prioritizing resource savings over detailed control.

6.3 Impact of Objective Function Components on DAM's Performance

The ablation studies, shown in Table 3, provide insights into the influence of each component within DAM's objective function on model performance across languages. KL divergence serves as the primary driver, with configurations incorporating KL divergence yielding the highest scores, particularly in Japanese and Korean. The cosine similarity constraint, initially proposed by ZipLoRA to diversify scaling coefficients, showed limited effectiveness in this multilingual setting, especially in tasks requiring retention of unique language characteristics. Meanwhile, entropy minimization, inspired by AdaMerging, demonstrates potential as a secondary objective in language modeling tasks. While KL divergence remains the dominant component, entropy minimization offers a lower-computational alternative when language-specific data requirements are less stringent.

7 Conclusion

This paper provides a comprehensive analysis of model merging techniques, spanning from simple averaging methods to automated, data-informed approaches such as evolutionary merging. We introduce DAM as an efficient alternative to evolutionary merging, significantly reducing computational overhead while achieving competitive performance. Our findings challenge the traditional assumption that more complex methods are inherently superior, showing that straightforward techniques like linear averaging can perform just as well, especially when merged models share similar characteristics. Future work could focus on assessing the scalability of different merging techniques in real-world applications and evaluating their suitability for resource-constrained environments, thereby enhancing their practical impact across the AI landscape.

8 Limitations

While our study presents a comprehensive comparison across multiple model merging techniques, it is important to acknowledge certain limitations in our methodology. One notable constraint is our reliance on open-sourced merged models when evaluating techniques like evolutionary merging, which are not fully available as open-source. This limitation restricts our experimental scope, as we could only test the outputs provided by these proprietary methods without full access to their detailed merging process or specific hyperparameters.

Furthermore, the datasets used in our evaluation were selected to approximate the original training domains of the models; however, we did not have access to the exact datasets that were utilized for domain adaptation in those models. Having such access would enable more controlled experimentation, particularly beneficial when assessing the impact of merging techniques like DAM in domain-specific scenarios.

Another valuable direction for future work would be to establish consistent benchmarks with specific metrics, datasets, and a diverse set of models. This would create a more standardized testing environment, allowing for a deeper analysis of model merging effects in precise, replicable settings. Such structured benchmarks would also facilitate a clearer assessment of long-term model stability and adaptability across different domains.

Overall, while our findings underscore the strengths and adaptability of DAM, further research with these enhancements would provide additional insights into the robustness and versatility of model merging approaches.

References

Samuel K Ainsworth, Jonathan Hayase, and Siddhartha Srinivasa. 2022. Git re-basin: Merging models modulo permutation symmetries. *arXiv preprint arXiv:2209.04836*.

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

augmxnt. 2023. shisa-gamma-7b. https://hf.co/augmxnt/shisa-gamma-7b-v1.

Ethan Chern, Haoyang Zou, Xuefeng Li, Jiewen Hu, Kehua Feng, Junlong Li, and Pengfei Liu. 2023. Generative ai for math: Abel. https://github.com/GAIR-NLP/abel.

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

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

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2021. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*.

Myeongjun Jang, Dohyung Kim, Deuk Sin Kwon, and Eric Davis. 2022. Kobest: Korean balanced evaluation of significant tasks. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 3697–3708.

Albert Q Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, et al. 2023a. Mistral 7b. *arXiv preprint arXiv:2310.06825*.

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2023b. Mistral 7b.

Yanis Labrak, Adrien Bazoge, Emmanuel Morin, Pierre-Antoine Gourraud, Mickael Rouvier, and Richard Dufour. 2024. Biomistral: A collection of open-source pretrained large language models for medical domains. *arXiv preprint arXiv:2402.10373*.

Viet Dac Lai, Chien Van Nguyen, Nghia Trung Ngo, Thuat Nguyen, Franck Dernoncourt, Ryan A Rossi, and Thien Huu Nguyen. 2023. Okapi: Instruction-tuned large language models in multiple languages with reinforcement learning from human feedback. *arXiv preprint arXiv:2307.16039*.

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.

Arindam Mitra, Hamed Khanpour, Corby Rosset, and Ahmed Awadallah. 2024. Orca-math: Unlocking the potential of slms in grade school math.

Satoshi Sekine, Maya Ando, Michiko Goto, Hisami Suzuki, Daisuke Kawahara, Naoya Inoue, and Kentaro Inui. 2024. ichikara-instruction: Constructing a japanese instruction dataset for llms. In *In Proceedings of The Thirtieth Annual Meeting of The Association for Natural Language Processing (NLP2024)*, pages 1508–1513. In Japanese.

Viraj Shah, Nataniel Ruiz, Forrester Cole, Erika Lu, Svetlana Lazebnik, Yuanzhen Li, and Varun Jampani. 2023. Ziplora: Any subject in any style by effectively merging loras. *arXiv preprint arXiv:2311.13600*.

Freda Shi, Mirac Suzgun, Markus Freitag, Xuezhi Wang, Suraj Srivats, Soroush Vosoughi, Hyung Won Chung, Yi Tay, Sebastian Ruder, Denny Zhou, Dipanjan Das, and Jason Wei. 2022. Language models are multilingual chain-of-thought reasoners.

Sidak Pal Singh and Martin Jaggi. 2020. Model fusion via optimal transport. *Advances in Neural Information Processing Systems*, 33:22045–22055.

Shamane Siriwardhana, Mark McQuade, Thomas Gauthier, Lucas Atkins, Fernando Fernandes Neto, Luke Meyers, Anneketh Vij, Tyler Odenthal, Charles Goddard, Mary MacCarthy, et al. 2024. Domain adaptation of llama3-70b-instruct through continual pretraining and model merging: A comprehensive evaluation. *arXiv preprint arXiv:2406.14971*.

George Stoica, Daniel Bolya, Jakob Bjorner, Taylor Hearn, and Judy Hoffman. 2023. Zipit! merging models from different tasks without training. *arXiv preprint arXiv:2305.03053*.

Sainbayar Sukhbaatar, Olga Golovneva, Vasu Sharma, Hu Xu, Xi Victoria Lin, Baptiste Rozière, Jacob Kahn, Daniel Li, Wen-tau Yih, Jason Weston, et al. 2024. Branch-train-mix: Mixing expert llms into a mixture-of-experts llm. *arXiv preprint arXiv:2403.07816*.

Rohan Taori, Ishaan Gulrajani, Tianyi Zhang, Yann Dubois, Xuechen Li, Carlos Guestrin, Percy Liang, and Tatsunori B Hashimoto. 2023. Alpaca: A strong, replicable instruction-following model. *Stanford Center for Research on Foundation Models. https://crfm. stanford. edu/2023/03/13/alpaca. html*, 3(6):7.

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.

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.

Enneng Yang, Zhenyi Wang, Li Shen, Shiwei Liu, Guibing Guo, Xingwei Wang, and Dacheng Tao. 2023. Adamerging: Adaptive model merging for multi-task learning. *arXiv preprint arXiv:2310.02575*.

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

Longhui Yu, Weisen Jiang, Han Shi, Jincheng Yu, Zhengying Liu, Yu Zhang, James T Kwok, Zhenguo Li, Adrian Weller, and Weiyang Liu. 2023. Metamath: Bootstrap your own mathematical questions for large language models. *arXiv preprint arXiv:2309.12284*.

Victor Zhong, Caiming Xiong, and Richard Socher. 2017. Seq2sql: Generating structured queries from natural language using reinforcement learning. *CoRR*, abs/1709.00103.