Evolutionary Optimization of Model Merging Recipes

Cirò G., Lomele M.

30592 - Topics in Computational Modelling: from Information Theory to Evolutionary Models

April 14, 2024

Evolutionary Optimization of Model Merging Recipes

Takuya Akiba, Makoto Shing, Yujin Tang, Qi Sun, David Ha
Sakana AI
Tokyo, Japan
{takiba,mkshing,yujintang,qisun,hadavid}@sakana.ai

Abstract

We present a novel application of evolutionary algorithms to automate the creation of powerful foundation models. While model merging has emerged as a promising approach for LLM development due to its cost-effectiveness, it currently relies on human intuition and domain knowledge, limiting its potential. Here, we propose an evolutionary approach that overcomes this limitation by automatically discovering effective combinations of diverse open-source models, harnessing their collective intelligence without requiring extensive additional training data or compute. Our approach operates in both parameter space and data flow space, allowing for optimization beyond just the weights of the individual models. This approach even facilitates cross-domain merging, generating models like a Japanese LLM with Math reasoning capabilities. Surprisingly, our Japanese Math LLM achieved state-of-the-art performance on a variety of established Japanese LLM benchmarks, even surpassing models with significantly more parameters, despite not being explicitly trained for such tasks. Furthermore, a culturally-aware Japanese VLM generated through our approach demonstrates its effectiveness in describing Japanese culture-specific content, outperforming previous Japanese VLMs. This work not only contributes new state-of-the-art models back to the open-source community, but also introduces a new paradigm for automated model composition, paving the way for exploring alternative, efficient approaches to foundation model development.[1]

Table of Contents

1. The Model Merging Problem
2. Useful Background
   - Task Arithmetic
   - TIES-Merging
   - DARE
   - Frankenmerging
   - CMA-ES
3. Evolutionary Model Merge
   - Parameter Space (PS)
   - Data Flow Space (DFS)
   - Interpolation
4. Results
   - Experiments
   - Natural Language Task
   - Computer Vision Task

Table of Contents

1. The Model Merging Problem
2. Useful Background
   - Task Arithmetic
   - TIES-Merging
   - DARE
   - Frankenmerging
   - CMA-ES
3. Evolutionary Model Merge
   - Parameter Space (PS)
   - Data Flow Space (DFS)
   - Interpolation
4. Results
   - Experiments
   - Natural Language Task
   - Computer Vision Task

Some Context

- A **Pre-Trained Model** (**PTM**) is a huge machine learning model trained on vast amount of data to understand **general patterns**:

$$\theta_{PTM} = [\theta_1, ..., \theta_P]$$

- **Fine Tuning** (**FT**) refers to further training a PTM on task-specific data to improve **task-specific** performance:

$$\theta^t_{FT} = [\theta^t_1, ..., \theta^t_P]$$

where $P = \#params$.

The Model Merging Problem

Combine **multiple PTMs** fine-tuned for T different tasks into a **single new model** which performs well on all T tasks.

$$\{\theta_{FT}^{1}, \ldots, \theta_{FT}^{T}\} \rightarrow \theta_{new}$$

How?

- Simple Average
- RegMean
- Fisher Merging
- Task Arithmetic
- TIES-Merging
- ...
- **Evolutionary Model Merge**

Table of Contents

1 The Model Merging Problem
2 Useful Background
Task Arithmetic
TIES-Merging
DARE
Frankenmerging
CMA-ES
3 Evolutionary Model Merge
Parameter Space (PS)
Data Flow Space (DFS)
Interpolation
4 Results
Experiments
Natural Language Task
Computer Vision Task

Table of Contents

1 The Model Merging Problem
2 Useful Background
 Task Arithmetic
 TIES-Merging
 DARE
 Frankenmerging
 CMA-ES
3 Evolutionary Model Merge
 Parameter Space (PS)
 Data Flow Space (DFS)
 Interpolation
4 Results
 Experiments
 Natural Language Task
 Computer Vision Task

Task Arithmetic

Ilharco et al. (Dec 2022) introduced **task vectors** τ as the difference in model's parameters before and after **fine-tuning**:

$$\tau^t = \theta^t_{FT} - \theta_{PTM}$$

Combining multiple task vectors with basic **vector operations** yields a well-performing **multi-task** model:

$$\theta_{new} = \theta_{PTM} + \lambda \sum_{t=1}^{T} \tau^t$$

where λ is a scaling hyper-parameter.

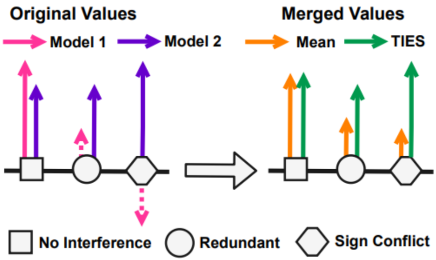
Table of Contents

1 The Model Merging Problem
2 Useful Background
Task Arithmetic
TIES-Merging
DARE
Frankenmerging
CMA-ES
3 Evolutionary Model Merge
Parameter Space (PS)
Data Flow Space (DFS)
Interpolation
4 Results
Experiments
Natural Language Task
Computer Vision Task

TIES-Merging

Yadav et al. (Jun 2023) showed there is **interference** in delta parameters when **combining task vectors**, due to:

- Redundant Parameters Value
- Sign Conflicts



TIES-Merging

They propose Trim, Elect Sign and Disjoint Merge (**TIES**):

1. **Trim** $\rightarrow$ keep only top-k% delta parameters by magnitude;
2. **Elect Sign** $\rightarrow$ by highest total magnitude;

$$sign^* = [sign_1^*, \ldots, sign_P^*]$$

$$sign_p^* = sign\left(\sum_{t=1}^{T} \tau_p^t\right), p \in \{1, \ldots, P\}$$

3. **Disjoint Merge** $\rightarrow$ compute mean of delta parameters whose signs agree with the elected sign.

$$\tau_p^m = \frac{1}{|A^p|} \sum_{t \in A^p} \tau_p^t$$

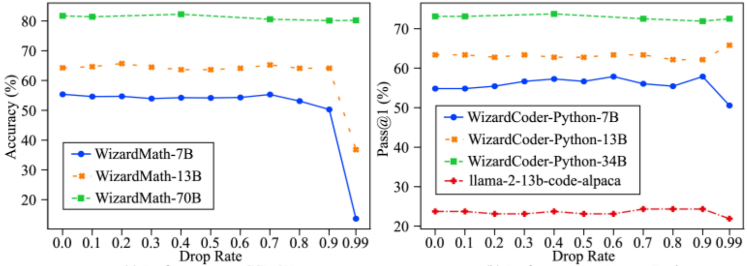
where $A^p = \{t \in [T] : sign_p^t = sign_p^*\}$

Table of Contents

1. The Model Merging Problem
2. Useful Background
    - Task Arithmetic
    - TIES-Merging
    - DARE
    - Frankenmerging
    - CMA-ES
3. Evolutionary Model Merge
    - Parameter Space (PS)
    - Data Flow Space (DFS)
    - Interpolation
4. Results
    - Experiments
    - Natural Language Task
    - Computer Vision Task

DARE

Yu et al. (Nov 2023) showed that most (**90-99%**)of the change in parameters happening during **fine-tuning** is actually **redundant**.



(a) Performance on GSM8K.

(b) Performance on HumanEval.

The **same performance** can be achieved by **resetting** most FT **parameters** to their original PTM values and **scaling** the rest accordingly.

Table of Contents

1 The Model Merging Problem
2 Useful Background
- Task Arithmetic
- TIES-Merging
- DARE
- Frankenmerging
- CMA-ES

3 Evolutionary Model Merge
- Parameter Space (PS)
- Data Flow Space (DFS)
- Interpolation

4 Results
- Experiments
- Natural Language Task
- Computer Vision Task

Frankenmerging

So far, we moved in the **Parameter Space** (**PS**) by interpolating weights of multiple models, keeping the architecture fixed.

We can also move in the **Data Flow Space** (**DFS**) by changing how the input flows through the model, i.e. changing the architecture.

Frankenmerging $\rightarrow$ stack different layers from different models together to create new architectures, while keeping the weights fixed.

Table of Contents

1 The Model Merging Problem
2 Useful Background
- Task Arithmetic
- TIES-Merging
- DARE
- Frankenmerging
- CMA-ES

3 Evolutionary Model Merge
- Parameter Space (PS)
- Data Flow Space (DFS)
- Interpolation

4 Results
- Experiments
- Natural Language Task
- Computer Vision Task

Evolutionary Algorithms

There are 4 main dialects of Evolutionary Algorithms (EA), whose main difference lies in the data structure used to encode candidate solutions:

- Genetic Algorithms (GA) $\rightarrow$ string over a finite alphabet
- **Evolution Strategies (ES)** $\rightarrow$ **real-valued vectors**
- Evolutionary Programming (EP) $\rightarrow$ finite state machines
- Genetic Programming (GP) $\rightarrow$ trees

Evolution Strategies

Given a starting population of size λ of candidate solutions, apply selection, recombination and mutation to improve fitness, minimizing an objective function.

- Candidate Solution: $\mathbf{x}_i \in \mathbb{R}^n$
- Population: $\{\mathbf{x}_1, ..., \mathbf{x}_\lambda\}$
- Objective Function $f : \mathbb{R}^n \to \mathbb{R}$

We will focus on **selection** and **mutation**.

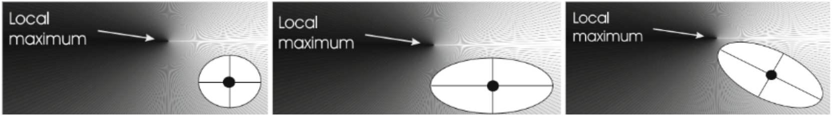
Selection and Mutation of Real-valued Vectors

At each generation g, select top μ candidate solutions according to fitness (**exploitation**) and apply mutation by adding Gaussian noise (**exploration**).

$$\mathbf{x}_i^{g+1} = \mathbf{x}_i^g + \mathbf{z}_i$$

where $\mathbf{z}_i \sim \mathcal{N}(0, C_i) \in \mathbb{R}^n$.

Modify the covariance matrix C_i to explore the search space in different ways:

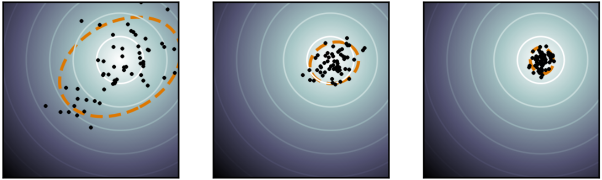


Estimation of Distributions Algorithms

Traditional Evolution Strategies (**ES**s) evolve an **implicit distribution** over the solution space by mutating each individual.

Estimation of Distribution Algorithms (**EDA**s) evolve an **explicit distribution** by mutating the parameters of a multivariate search distribution.

ESs maintain the entire population, EDAs preserve only the population statistics.



CMA-ES

Hansen (Apr 2016) introduced **Covariance Matrix Adaptation** (**CMA**) **Evolution Strategy** (**ES**), a novel approach to explore the solution space by evolving the statistics of a Search Distribution.

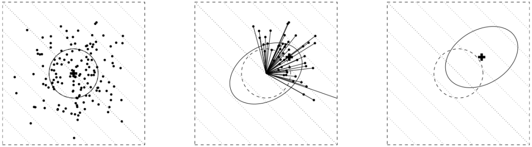
CMA-ES

At each generation $g = 1, 2, \ldots$ sample candidate solutions:

$$\mathbf{x}_k^{(g+1)} \sim \mathbf{m}^{(g)} + \sigma^{(g)}\mathcal{N}(0, C^{(g)})$$

$$\text{for } k = 1, \ldots, \lambda, \quad \mathbf{m}^{(g)} \in \mathbb{R}^n, \quad C^{(g)} \in \mathbb{R}^{n \times n}$$

Apply selection and adapt parameters (**mean** m, **step-size** σ, **covariance matrix** C):



Adapting the Mean

The new mean $\mathbf{m}^{g+1}$ is a weighted average of μ selected points from the sample $\mathbf{x}_1^{g+1}, \ldots, \mathbf{x}_\lambda^{g+1}$:

$$\mathbf{m}^{(g+1)} = \sum_{i=1}^{\mu} w_i \mathbf{x}_{i:\lambda}^{(g+1)}$$

$$\sum_{i=1}^{\mu} w_i = 1 \qquad \text{for } w_1 \geq w_2 \geq \ldots \geq w_\mu > 0.$$

$\mathbf{x}_{i:\lambda}$ is the i-th fittest solution.

Adapting the Covariance Matrix

The new covariance matrix $C^{(g+1)}$ is obtained as follows:

$$\boldsymbol{C}^{(g+1)} = \underbrace{(1 - c_1 - c_\mu \textstyle\sum w_j)}_{\text{can be close or equal to } 0} \boldsymbol{C}^{(g)} + c_1 \underbrace{\boldsymbol{p}_c^{(g+1)} \boldsymbol{p}_c^{(g+1)\mathsf{T}}}_{\text{rank-one update}} + c_\mu \underbrace{\sum_{i=1}^{\lambda} w_i \, \boldsymbol{y}_{i:\lambda}^{(g+1)} \left(\boldsymbol{y}_{i:\lambda}^{(g+1)}\right)^{\mathsf{T}}}_{\text{rank-}\mu\text{ update}}$$

Rank-μ update

$$\boldsymbol{C}^{(g+1)} = \underbrace{(1 - c_1 - c_\mu \textstyle\sum w_j)}_{\text{can be close or equal to } 0} \boldsymbol{C}^{(g)} + c_1 \underbrace{\boldsymbol{p}_c^{(g+1)} \boldsymbol{p}_c^{(g+1)^{\mathsf{T}}}}_{\text{rank-one update}} + c_\mu \underbrace{\sum_{i=1}^{\lambda} w_i \, \boldsymbol{y}_{i:\lambda}^{(g+1)} \left(\boldsymbol{y}_{i:\lambda}^{(g+1)}\right)^{\mathsf{T}}}_{\text{rank-}\mu\text{ update}}$$

where

$$\mathbf{y}_{i:\lambda}^{(g+1)} = (\mathbf{x}_{i:\lambda}^{(g+1)} - \mathbf{m}^{(g)})/\sigma^{(g)}$$

and

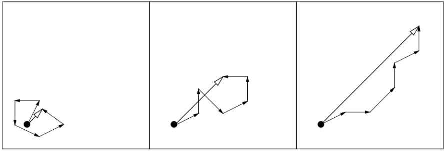
$$w_{1,\dots,\lambda} \in \mathbb{R} : w_1 \geq \dots \geq w_\mu \geq 0 \geq w_{\mu+1} \geq \dots \geq w_\lambda$$

The rank-μ update uses the standardized and weighted sample covariance matrix.

Evolution Path

An **evolution path** is the path traced across the search space by the Search Distribution's **mean** as it **evolves** throughout generations:

$$\frac{\mathbf{m}^{(g+1)}-\mathbf{m}^{(g)}}{\sigma^{(g)}}+\frac{\mathbf{m}^{(g)}-\mathbf{m}^{(g-1)}}{\sigma^{(g-1)}}+\frac{\mathbf{m}^{(g-1)}-\mathbf{m}^{(g-2)}}{\sigma^{(g-2)}}+\ldots+\frac{\mathbf{m}^{(1)}-\mathbf{m}^{(0)}}{\sigma^{(0)}}$$



Rank-one update

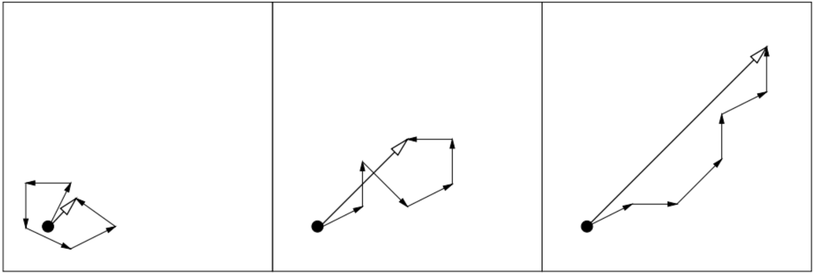
$$\boldsymbol{C}^{(g+1)} = \underbrace{(1 - c_1 - c_\mu \textstyle\sum w_j)}_{\text{can be close or equal to 0}} \boldsymbol{C}^{(g)} + c_1 \underbrace{\boldsymbol{p}_c^{(g+1)} \boldsymbol{p}_c^{(g+1)^{\mathsf{T}}}}_{\text{rank-one update}} + c_\mu \underbrace{\sum_{i=1}^{\lambda} w_i \, \boldsymbol{y}_{i:\lambda}^{(g+1)} \left(\boldsymbol{y}_{i:\lambda}^{(g+1)}\right)^{\mathsf{T}}}_{\text{rank-}\mu\text{ update}}$$

Add scary coefficients and exponentially smooth (discount) the previous generations' paths to get the update:

$$\mathbf{p}_c^{(g+1)} = (1 - c_c)\mathbf{p}_c^{(g)} + \sqrt{c_c(2 - c_c)\mu_{eff}} \frac{\mathbf{m}^{(g+1)} - \mathbf{m}^{(g)}}{c_m \sigma^{(g)}}$$

Adapting the Step-Size

The step-size $\sigma^{(g+1)}$ is updated based on evolution paths:



- Anti-correlated (left) $\rightarrow$ decrease;
- Orthogonal (middle) $\rightarrow$ keep;
- Correlated (right) $\rightarrow$ increase.

Table of Contents

1. The Model Merging Problem
2. Useful Background
   - Task Arithmetic
   - TIES-Merging
   - DARE
   - Frankenmerging
   - CMA-ES
3. Evolutionary Model Merge
   - Parameter Space (PS)
   - Data Flow Space (DFS)
   - Interpolation
4. Results
   - Experiments
   - Natural Language Task
   - Computer Vision Task

Overview

The Open LLM Leaderboard is dominated by merged models, great cost-effectiveness.

However, current model merging relies on **human intuition** and domain knowledge.

Akiba et al. (Mar 2024) from Sakana AI inroduced a **systematic** model composition framework using **evolution**. In particular:

1. Merge in Parameter Space (PS).
2. Merge in Data Flow Space (DFS).
3. Integrate PS and DFS.

Table of Contents

1 The Model Merging Problem
2 Useful Background
   Task Arithmetic
   TIES-Merging
   DARE
   Frankenmerging
   CMA-ES
3 Evolutionary Model Merge
   Parameter Space (PS)
   Data Flow Space (DFS)
   Interpolation
4 Results
   Experiments
   Natural Language Task
   Computer Vision Task

Merge in PS

Sub-goal 1: **Merge** the **weights** of the models **block-wise**, where each block/layer is an input-output embedding module, e.g. a transformer's block.

1. Apply DARE and TIES-Merging.
2. Establish **configuration parameters** for sparsification and weight mixing (e.g. scaling in TIES and drop % in DARE).
3. Set accuracy (or any other critical task-specific metric) as the fitness function.
4. Optimise the parameters with CMA-ES.

Merge in PS visualized

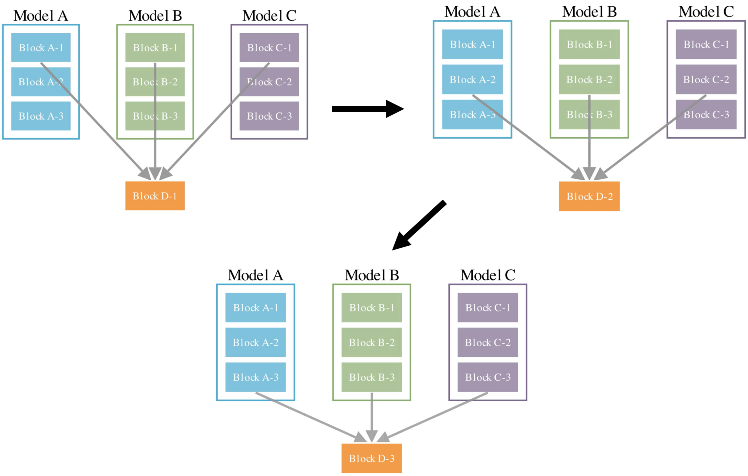


Table of Contents

1. The Model Merging Problem
2. Useful Background
   - Task Arithmetic
   - TIES-Merging
   - DARE
   - Frankenmerging
   - CMA-ES
3. Evolutionary Model Merge
   - Parameter Space (PS)
   - Data Flow Space (DFS)
   - Interpolation
4. Results
   - Experiments
   - Natural Language Task
   - Computer Vision Task

Merge in DFS

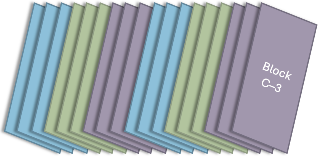
Sub-goal 2: optimise how **tokens flow** across the layers of the models, leaving weights unchanged. Given N models, with a total of M layers, and a budget length T, we search for an **optimal sequence** of layers.

1. Lay out models in sequential order, repeat them r times.
2. Set an **indicator array** $I \in \mathbb{R}^{T=M\times r}$ to deal with the inclusion ($I_i = 1$) and exclusion ($I_i = 0$) of layers.
3. Set a **scaling matrix** $W \in \mathbb{R}^{M\times M}$, where W_{ij} is scaling from layer i to j in the sequence. This because a layer may face an input whose distribution is different from what it is used to.
4. Optimise I and W with CMA-ES.

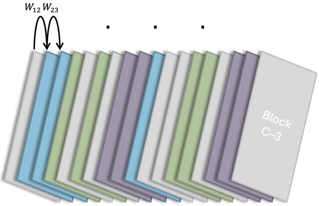
Merge in DFS visualised

Number of models: $N = 3$
Number of total layers: $M = 9$
Number of repetitions: $r = 2$
Budget for sequence length: $T = 18$

1



2 & 3



$I = [0, \quad 1, \quad 1, \quad \dots, \quad 0]$

Table of Contents

1 The Model Merging Problem
2 Useful Background
   Task Arithmetic
   TIES-Merging
   DARE
   Frankenmerging
   CMA-ES
3 Evolutionary Model Merge
   Parameter Space (PS)
   Data Flow Space (DFS)
   Interpolation
4 Results
   Experiments
   Natural Language Task
   Computer Vision Task

Integrate PS and DFS

Combine the two approaches:

1. Merge in PS a collection of models
   $\rightarrow$ new merged model.
2. Insert new merged model into the collection
   $\rightarrow$ updated collection.
3. Merge in DFS the updated collection
   $\rightarrow$ final merged model.

Evolutionary Model Merge visualised

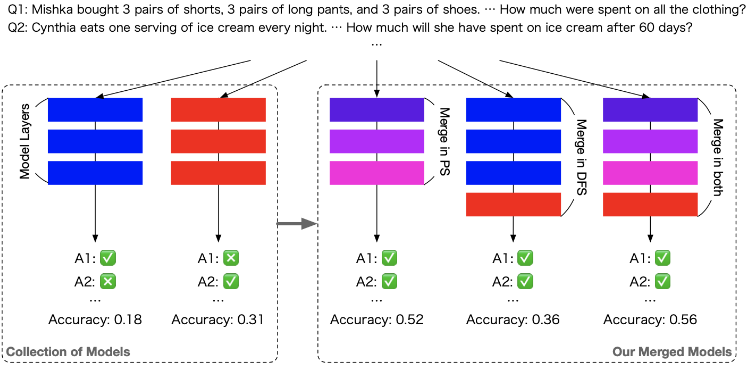


Table of Contents

1. The Model Merging Problem
2. Useful Background
   - Task Arithmetic
   - TIES-Merging
   - DARE
   - Frankenmerging
   - CMA-ES
3. Evolutionary Model Merge
   - Parameter Space (PS)
   - Data Flow Space (DFS)
   - Interpolation
4. Results
   - Experiments
   - Natural Language Task
   - Computer Vision Task

Table of Contents

1. The Model Merging Problem
2. Useful Background
   - Task Arithmetic
   - TIES-Merging
   - DARE
   - Frankenmerging
   - CMA-ES
3. Evolutionary Model Merge
   - Parameter Space (PS)
   - Data Flow Space (DFS)
   - Interpolation
4. Results
   - Experiments
   - Natural Language Task
   - Computer Vision Task

Tasks

- **Natural Language Processing**: Solve math problems in Japanese.
- **Computer Vision**: Recognise and describe elements of the Japanese culture in pictures.

Table of Contents

1. The Model Merging Problem
2. Useful Background
   - Task Arithmetic
   - TIES-Merging
   - DARE
   - Frankenmerging
   - CMA-ES
3. Evolutionary Model Merge
   - Parameter Space (PS)
   - Data Flow Space (DFS)
   - Interpolation
4. Results
   - Experiments
   - Natural Language Task
   - Computer Vision Task

Natural Language Task

Goal: combine **Japanese LLMs** and **Math LLMs** (all based on Mistal-7B) into one that can generate **Japanese answers to Japanese math problems**.

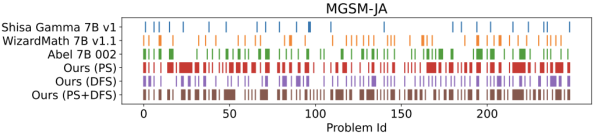
Dataset: MGSM. *Train set*: 1069 samples translated in Japanese. *Test set*: 250 samples original in Japanese.

Optimisation with CMA-ES:

- *PS Merging*
  - $\sigma = 1/6$
  - population size $4 + \lfloor 3 ln(n_{params}) \rfloor$
- *DFS Merging*
  - $M = 64$, $r = 3$
  - population size $= 128$
  - generations $= 100$

Results

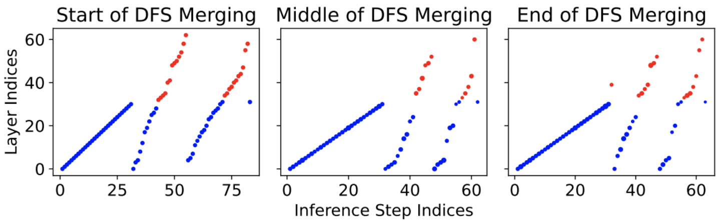
Key finding: merged model retains foundational knowledge from the source models and exhibits the desired **emerging capability**.



Id.	Model	Type	Size	MGSM-JA (acc ↑)	JP-LMEH (avg ↑)
1	Shisa Gamma 7B v1	JA general	7B	9.6	66.1
2	WizardMath 7B v1.1	EN math	7B	18.4	60.1
3	Abel 7B 002	EN math	7B	30.0	56.5
4	**Ours (PS)**	1 + 2 + 3	7B	**52.0**	**70.5**
5	**Ours (DFS)**	3 + 1	10B	**36.4**	**53.2**
6	**Ours (PS+DFS)**	4 + 1	10B	**55.2**	**66.2**
7	Llama 2 70B	EN general	70B	18.0	64.5
8	Japanese StableLM 70B	JA general	70B	17.2	68.3
9	Swallow 70B	JA general	70B	13.6	71.5
10	GPT-3.5	commercial	-	50.4	-
11	GPT-4	commercial	-	78.8	-

Results

A look at **model explainability** in DFS merging with two models.



The evolutionary search includes most layers of the first model, the blue dots, at an early stage. Then, it alternates between layers from both models.

Table of Contents

1. The Model Merging Problem
2. Useful Background
   - Task Arithmetic
   - TIES-Merging
   - DARE
   - Frankenmerging
   - CMA-ES
3. Evolutionary Model Merge
   - Parameter Space (PS)
   - Data Flow Space (DFS)
   - Interpolation
4. Results
   - Experiments
   - Natural Language Task
   - Computer Vision Task

Computer Vision Task

Goal: design a **culturally-specific content aware** Japanese VLM that can describe pictures and give insights about Japan and its culture.

Dataset: Japanese Visual Genome VQA and JA-VLM-Bench-in-the-Wild. The latter is a hand-crafted collection of 42 images with 50 questions and nuanced answers about Japan's culture and its objects.

Optimisation using CMA-ES with identical settings to the previous PS-Merging experiment.

Results

Key finding: **multi-modality** allows merged model to produce more detailed and accurate responses.

Model	JA-VG-VQA-500 (ROUGE-L ↑)	JA-VLM-Bench-In-the-Wild (ROUGE-L ↑)
LLaVA 1.6 Mistral 7B	14.3	41.1
Japanese Stable VLM	-	40.5
Ours	**19.7**	**51.2**

Conclusion

- Novel method for merging models of vastly different domains that achieves state-of-the-art results;
- Showcase for the potential of democratising model development.

Limitations

- *Inheritance* $\rightarrow$ no method of filtering selection of models or blocks of models;
- *Incoherence* $\rightarrow$ no control on logical and factual coherence of responses;
- *Alignment* $\rightarrow$ lack of instruction fine-tuning;
- *Evolutionary Details* $\rightarrow$ it's hard to understand exactly which parameters they're evolving and no insights on the evolution process (only a hint for DFS).

Future Work

- Cross-domain *image generation* $\rightarrow$ evolutionary model merging on diffusion models;
- *Self-improvement* $\rightarrow$ evolutionary search for candidate models instead of manual selection of starting population;
- *Generalisation* $\rightarrow$ merging in more orthogonal spaces (beyond PS & DFS).

Thank You!

→ Generated by the authors with EvoSDXL-JP, an incoming diffusion model created with Evolutionary Model Merge.

Question 1

- *In the NLP task, how does the final model retain performance on the original tasks?*

  From the results table, the performance of the final model (id 6) on general japanese language tasks is measured on JP-LMEH dataset.

  No significant performance drop is observed compared to the original models (1-3), with model 4 (PS) actually improving.

  Unfortunately, the authors do not provide insights on the performance of the merged model on english math, suggesting there might be a loss.

Question 2

- *How does the final model retain performance on English language tasks?*

  The authors do not expand on the performance of the merged models on English tasks. This aspect was not part of the original models' tasks, which were fine-tuned for English math problems and Japanese language.

  This is also evident from the two databases that were used to test and benchmark the final merged model. The first, MGSM-JA, is a set of grade-school math problems translated to Japanese by the authors. The second, JP-LMEH, is an evaluation harness specific for Japanese language models.

Question 3

- *How does this method perform on overlapping or less complementary tasks?*

  In this paper, the authors do not present additional experiments of using their method. Hence, there is no empirical evidence of how this method behaves for overlapping tasks.

  However, in the related research papers (DARE, TIES, Task Arithmetic etc.), it's been shown that merging different models fine-tuned for the same tasks improve both in-domain performance and out-of-domain generalization.

  Moreover, merged models provide a better initialization for fine-tuning.

Bibliography

- Akiba, T., Shing, M., Tang, Y., Sun, Q., & Ha, D. (2023). Evolutionary Optimization of Model Merging Recipes. *arXiv preprint arXiv:2403.13187*. Retrieved from `https://arxiv.org/abs/2403.13187`
- Yu, L., Yu, B., Yu, H., Huang, F., & Li, Y. (2023). Language Models are Super Mario: Absorbing Abilities from Homologous Models as a Free Lunch. *arXiv preprint arXiv:2311.03099*. Retrieved from `https://arxiv.org/abs/2311.03099`
- Ilharco, G., Ribeiro, M.T., Wortsman, M., Gururangan, S., Schmidt, L., Hajishirzi, H., & Farhadi, A. (2022). Editing models with task arithmetic. *arXiv preprint arXiv:2212.04089*. Retrieved from `https://arxiv.org/abs/2212.04089`

Bibliography

- Hansen, N. (2016). The CMA Evolution Strategy: A Tutorial. *arXiv preprint arXiv:1604.00772*. Retrieved from `https://arxiv.org/abs/1604.00772`
- Sakana AI. (2023). Evolutionary Model Merge. Retrieved from `https://sakana.ai/evolutionary-model-merge/`