ML2026 HW7
Model Merging

TAs: 黃郁涵、陳思齊、董家愷

Email: ntu-ml-2026-spring-ta@googlegroups.com

Deadline: 2026/05/28 23:59:59

Ref: ML2025 HW9 Model Merging

Outline

- Task Description
- Dataset
- Eval Metric and Answer Extraction
- Merging Algorithms
- Paper Reading
- Submission and Grading
- Reference

Links

- Course Website
- NTU COOL
- Colab Sample Code
- Kaggle Sample Code
- Judgeboi

Task Description - overall

Part 1 (60%) - model merging

- Modify model merging config
- Merge in possible (weights, density) pairs and inference on task
- Select the optimal results, save submission.json and submit to Judgeboi
- Submit your .ipynb file to NTUCOOL

Part 2 (40%) - paper reading

- Read the given two papers and answer the questions on NTUCOOL

Task Description

- Goal: Learn to merge models with distinct capabilities at the **parameter level** to build a unified, multi-task model without additional training
- Explore various model merging algorithms to develop a unified model that preserves or improves performance across Japanese understanding and mathematical capability.

Task Description

base models **fine-tuned** for specific domain capabilities

two 7B models pretrained from **Mistral-7B-v1**

1. **augmxnt/shisa-gamma-7b-v1 (Japanese)**
2. **WizardLMTeam/WizardMath-7B-V1.1 (Math)**

merge on base model weights with techniques,
e.g. linear, pruning, ties, SCE

mergekit: a toolkit for merging

Inference

a model with **Japanese Math QAs** capability

Akiba, Takuya, et al. "Evolutionary optimization of model merging recipes." *Nature Machine Intelligence* 7.2 (2025): 195-204.

Task Description

In this homework, students are required to implement model merging using two 7B llms:

- `augmxnt/shisa-gamma-7b-v1`, which is strong in Japanese language understanding
- `WizardLMTeam/WizardMath-7B-V1.1`, which is strong in mathematical reasoning
- Both models are fine-tuned from Mistral-7B-v0.1, as noted in *Evolutionary Optimization of Model Merging Recipes.*

Evaluation

- 20 Japanese Math questions created by TA 董家愷
- Instead of math problems solely translated to Japanese, the questions also require background knowledge related Japanese culture.
- Example:

  "日本の都道府県のうち、東京都、北海道、京都府、大阪府を除いた「県」はいくつありますか？"（日本縣市數量）

  "太郎は平成5年に生まれました。次郎は令和2年に生まれました。太郎は次郎より何歳年上ですか？"（日本年號紀年制度）

Eval Metric and Answer Extraction

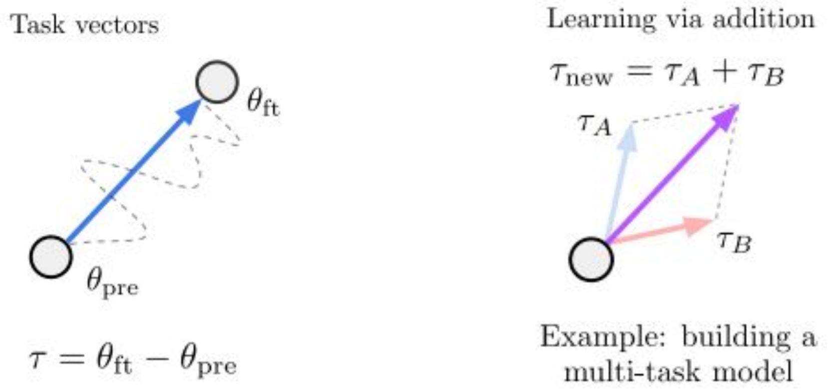
- The final answer is extracted from the model’s generated output using a rule-based approach. The extraction function first searches for a number that appears after the pattern “答え:” or “答え：”
- If the expected pattern is not present, the function falls back to extracting the last numerical value appearing in the generated text
- You can get the accuracy in the output of your notebook (exact same result on Judgeboi)

Paper Reading

- Basic model merging questions (5 questions)
- Latest model merging related paper (3 questions)
    - Soro, Bedionita, et al. "LS-Merge: Merging Language Models in Latent Space." The Fourteenth International Conference on Learning Representations. 2026.
- Answer the multiple choice questions on NTUCOOL

Model Merging

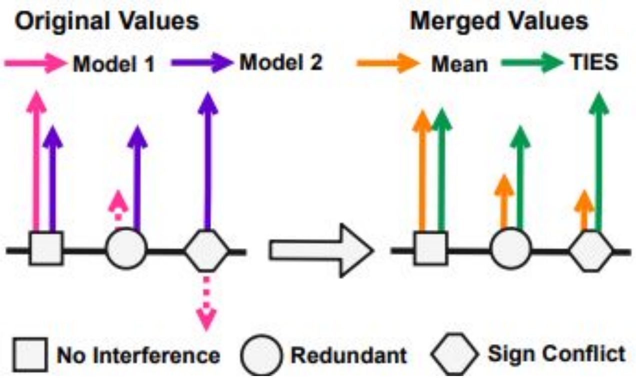
- Def. refer to the process of merging models with **simple arithmetic** on parameters **without retraining** from scratch or accessing original training data, to preserve or integrate capabilities from each source model (e.g., tasks, domains) into a unified model



Ilharco, Gabriel, et al. "Editing models with task arithmetic." *arXiv preprint arXiv:2212.04089* (2022).

Model Merging

- recent emergence of large foundation models and pretraining-finetuning paradigm have motivated more merging techniques → *Multi-Task Learning*
- redundant parameter or sign conflicts in different task vectors ⇒ ***parameter interference*** → degraded performance



Lee, Yeoreum, Jinwook Jung, and Sungyong Baik. "[Mitigating Parameter Interference in Model Merging via Sharpness-Aware Fine-Tuning]." *The Thirteenth International Conference on Learning Representations*.

Terminology in Merging Algorithms

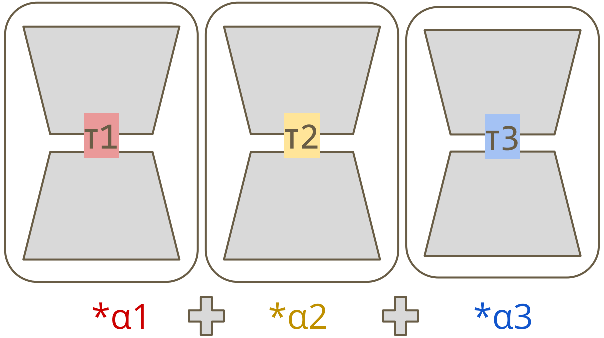
3 common settings in MergeKit...

- input models / base model
  - the source models to merge, plus the base model used as reference when needed
- weights `list(α)`
  - scalar coefficients controlling each model's contribution
- density `d`
  - fraction of values to preserve in a tensor or matrix

Merging Algorithms

- Task Arithmetic / linear (weights)

  linear(A, B; t)= (1−t)A + tB



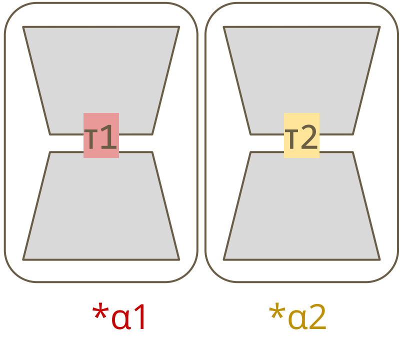
weighted sum of task vectors

Ilharco, Gabriel, et al. "[Editing models with task arithmetic]." *arXiv preprint arXiv:2212.04089* (2022).

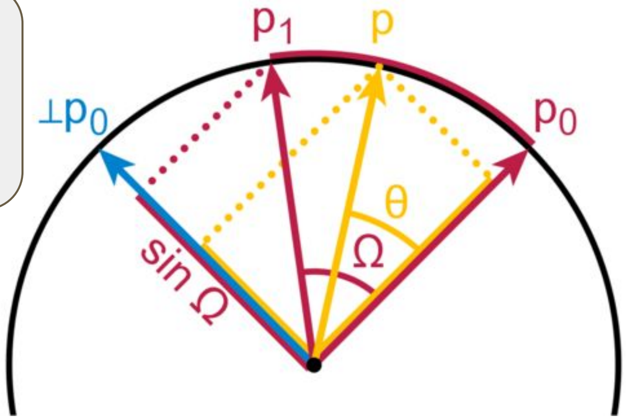
Merging Algorithms

- Slerp (weights)

$$\mathrm{slerp}(p_0, p_1; t) = \frac{\sin[(1-t)\Omega]}{\sin\Omega} p_0 + \frac{\sin[t\Omega]}{\sin\Omega} p_1.$$



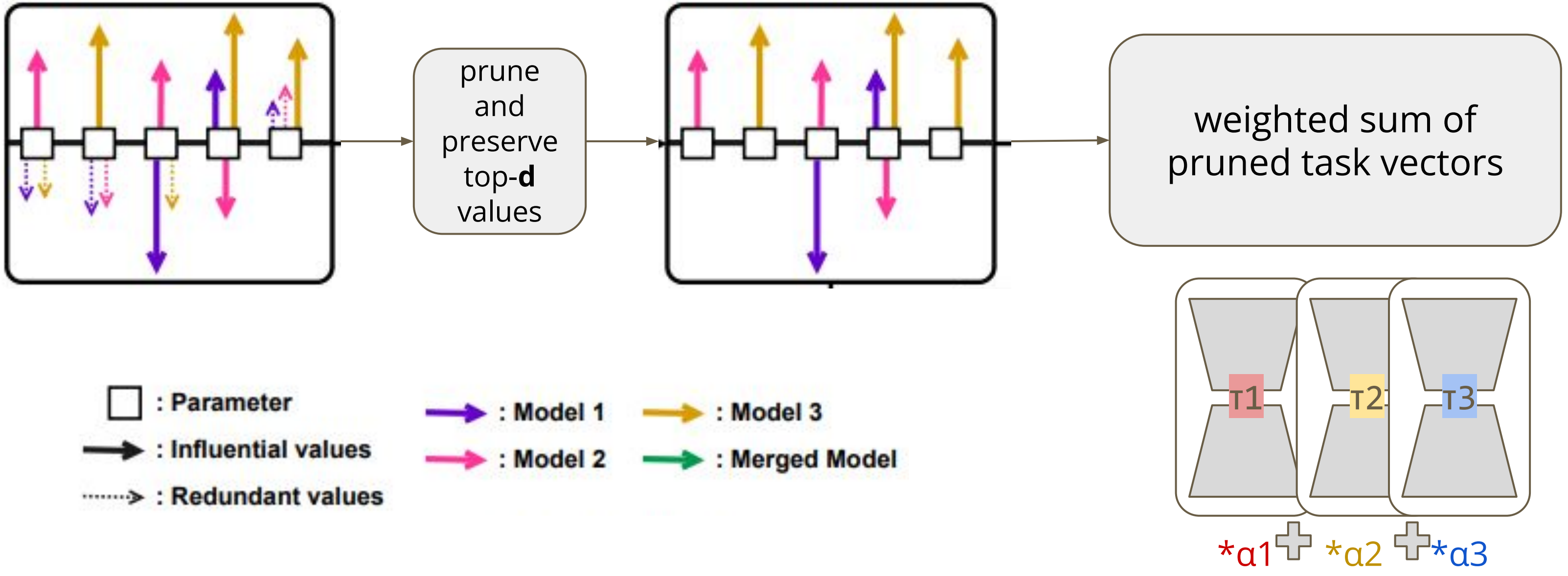
Spherical Linear Interpolation of the weights of two models



Merging Algorithms

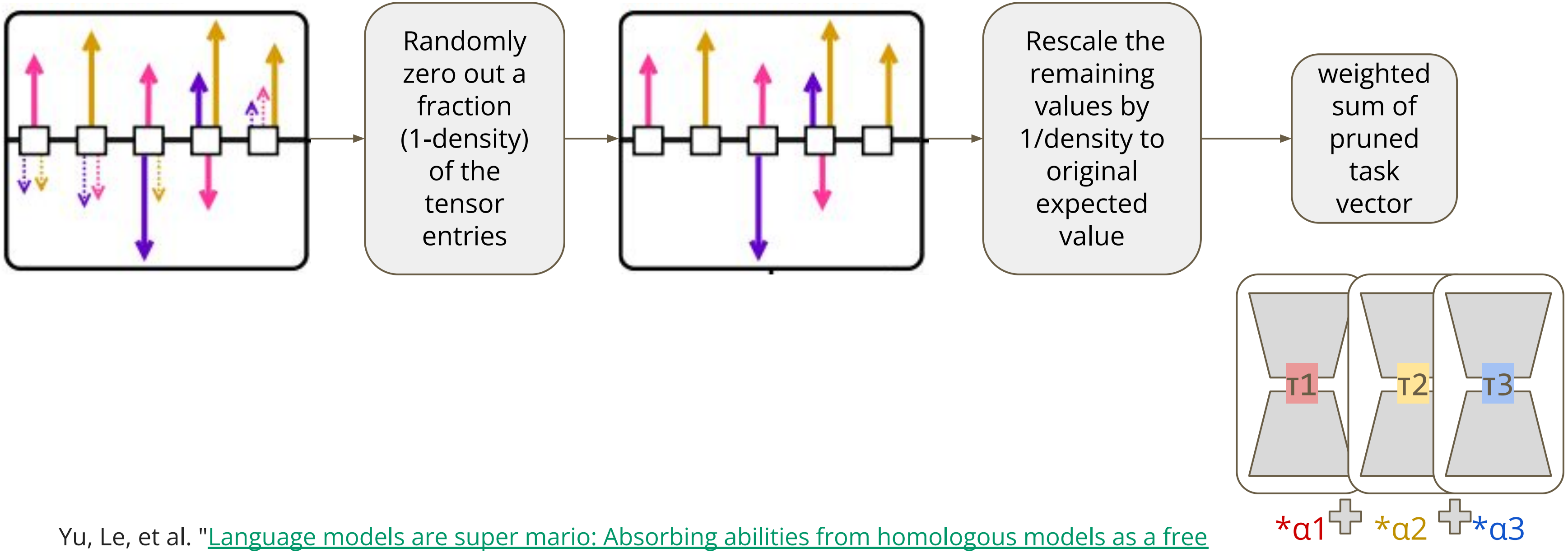
Arrow: direction → sign, length → magnitude

- Magnitude Prune (density, weights)



Merging Algorithms

- DARE Linear (density, weights)



Yu, Le, et al. "Language models are super mario: Absorbing abilities from homologous models as a free lunch." Forty-first International Conference on Machine Learning. 2024.

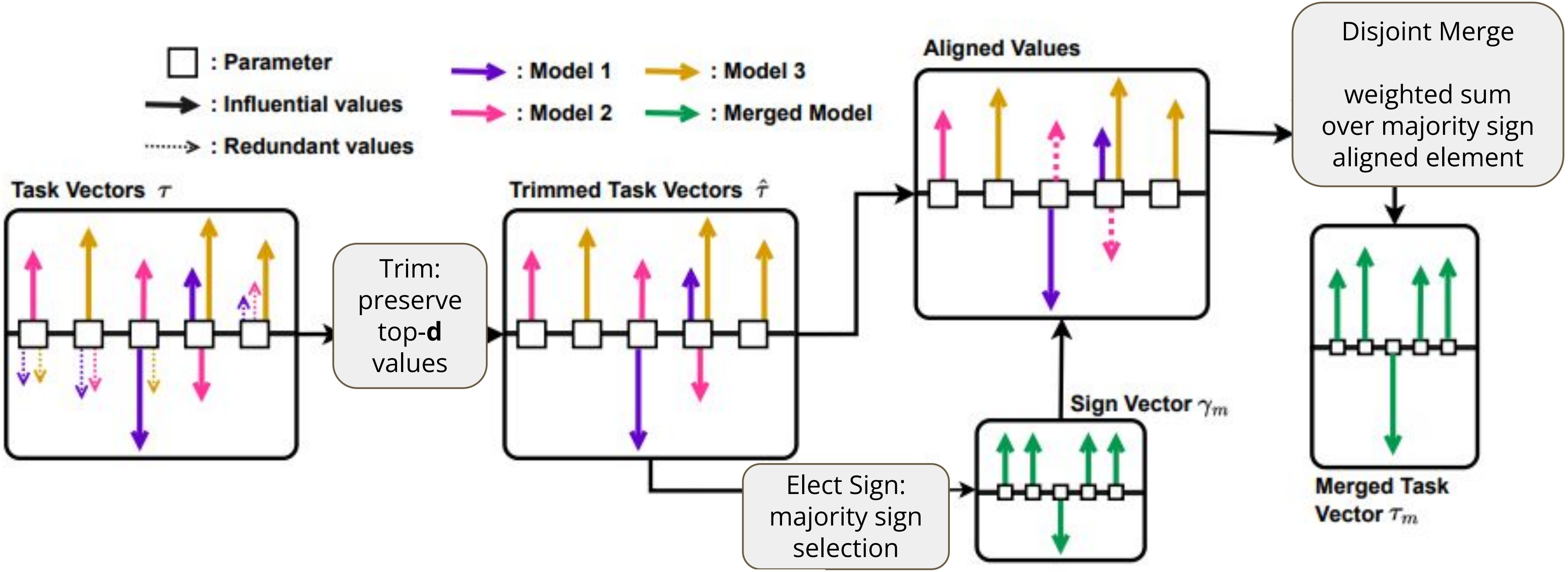
Merging Algorithms

- DARE Linear
  - derive task vectors
  - **D**rop: randomly zero out a fraction (1 - d) (Bernouli(d) masks) of the tensor entries to preserve vector elements with density d
  - **A**nd
  - **RE**scale: rescale remaining ones by 1/d to approximate expected value of the original embeddings
  - weighted sum refined task vectors

Yu, Le, et al. "Language models are super mario: Absorbing abilities from homologous models as a free lunch." Forty-first International Conference on Machine Learning. 2024.

Merging Algorithms

- TIES (density, weights)



Yadav, Prateek, et al. "[Ties-merging: Resolving interference when merging models.](#)" *Advances in Neural Information Processing Systems* 36 (2023): 7093-7115.

Merging Algorithms

- TIES
  - derive task vectors
  - **T**r**I**m: prune task vector by magnitude, preserve top-**d** important parameters
  - **E**lect **S**ign : Determine +/- for each parameter by summing up (total/frequency)
  - Disjoint **Merge**: weighted sum over majority sign aligned elements

Yadav, Prateek, et al. "Ties-merging: Resolving interference when merging models." *Advances in Neural Information Processing Systems* 36 (2023): 7093-7115.

Merging Algorithms

- TIES

Algorithm 1 TIES-MERGING Procedure.

Input: Fine-tuned models $\{\theta_t\}_{t=1}^n$, Initialization θ_{init}, k, and λ.
Output: Merged Model θ_m
forall t ***in*** $1, ..., n$ **do**

```
▷ Create task vectors.
```
$\tau_t = \theta_t - \theta_{\text{init}}$
```
▷ Step 1: Trim redundant parameters.
```
$\hat{\tau}_t \leftarrow \text{keep\_topk\_reset\_rest\_to\_zero}(\tau_t, k)$
$\hat{\gamma}_t \leftarrow sgn(\hat{\tau}_t)$
$\hat{\mu}_t \leftarrow |\hat{\tau}_t|$

end
```
▷ Step 2: Elect Final Signs.
```
$\gamma_m = sgn(\sum_{t=1}^n \hat{\tau}_t)$
```
▷ Step 3: Disjoint Merge.
```
forall p ***in*** $1, ..., d$ **do**

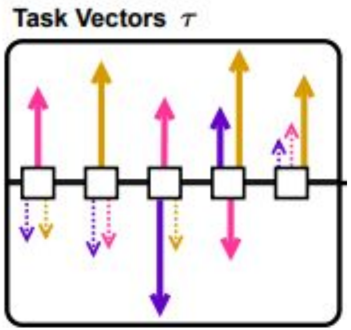
$\mathcal{A}^p = \{t \in [n] \mid \hat{\gamma}_t^p = \gamma_m^p\}$
$\tau_m^p = \frac{1}{|\mathcal{A}^p|} \sum_{t \in \mathcal{A}^p} \hat{\tau}_t^p$

end
```
▷ Obtain merged checkpoint
```
$\theta_m \leftarrow \theta_{\text{init}} + \lambda * \tau_m$
return θ_m

Ref. Yadav, Prateek, et al. "Ties-merging: Resolving interference when merging models." *Advances in Neural Information Processing Systems* 36 (2023): 7093-7115.

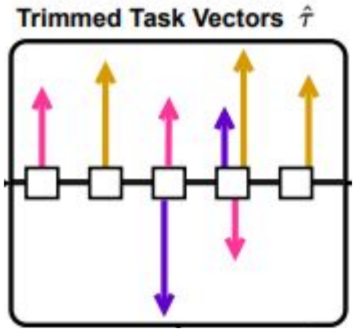
Merging Algorithms

- SCE (density)



Select (prune)

consider element variances between task vectors



$$\eta_{j,m} = \frac{\sum \hat{\delta}_{j,m}^2}{\sum_j \sum \hat{\delta}_{j,m}^2}$$

Calculate coefficient

calculate the sum of squares of elements

Erase minority elements

select majority sign of elements

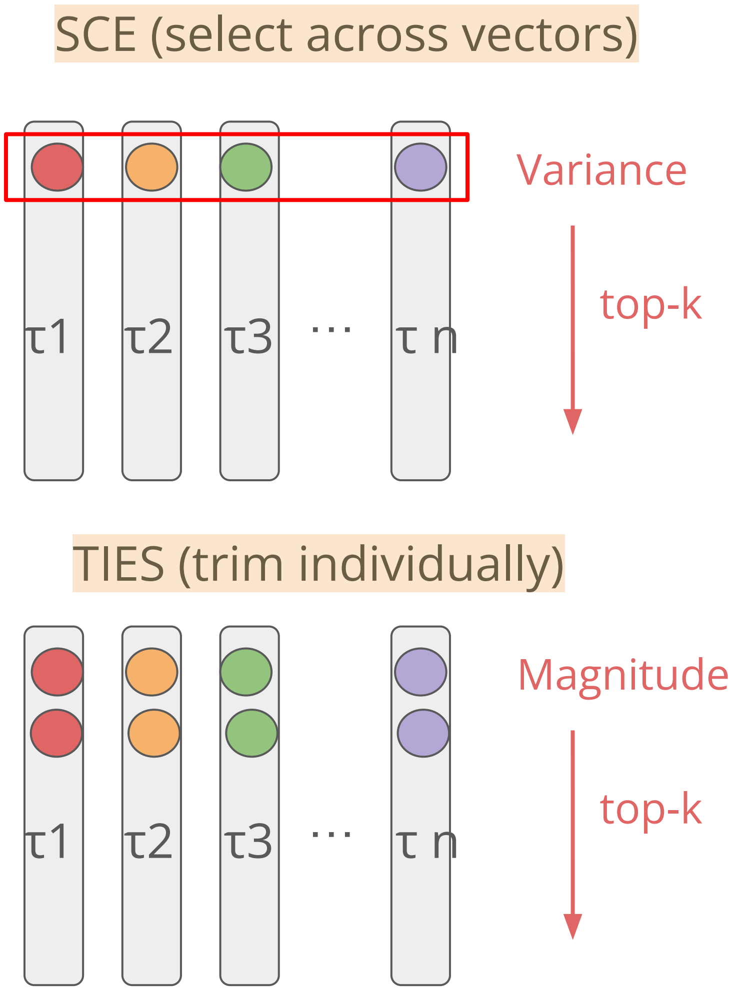
Disjoint Merge

weighted sum over majority sign aligned element

Wan, Fanqi, et al. "Fusechat: Knowledge fusion of chat models." *arXiv preprint arXiv:2408.07990* (2024).

Merging Algorithms

- SCE v.s. TIES
  - Select – similar to pruning, further consider variations across different task vectors
  - Trim – prune each task vector individually



Merging Algorithms

- SCE
  - derive task vectors
  - **S**: select top-k variance elements in matrices (among different task vectors)
    - v.s. TIES (pruning individually)
  - **C**: sum of squares of elements to obtain merging coefficient for each target LLM
  - **E**: filter elements with minority directions

Wan, Fanqi, et al. "Fusechat: Knowledge fusion of chat models."
arXiv preprint arXiv:2408.07990 (2024).

Merging Algorithms

- SCE

Algorithm 1 SCE Procedure

Input: target LLMs parameters $\{\phi_j\}_{j=1}^{K-1}$, pivot LLM parameters θ_v, threshold τ.
Output: merged LLM parameters Φ

▷ `Create fusion vectors`

$$\{\delta_j\}_{j=1}^{K-1} = \{\phi_j - \theta_v\}_{j=1}^{K-1} \quad (5)$$

▷ `Calculate parameter matrix-level merging coefficients`

for $\{\delta_{j,m}\}_{j=1}^{K-1} \in \{\delta_j\}_{j=1}^{K-1}$ **do**

▷ `Step 1: Select salient elements`

$$\{\hat{\delta}_{j,m}\}_{j=1}^{K-1} = \mathrm{Select}(\{\delta_{j,m}\}_{j=1}^{K-1}, \tau) \quad (6)$$

▷ `Step 2: Calculate coefficients`

$$\{\eta_{j,m}\}_{j=1}^{K-1} = \mathrm{Calculate}(\{\hat{\delta}_{j,m}^2\}_{j=1}^{K-1}) \quad (7)$$

$$\eta_{j,m} = \frac{\sum \hat{\delta}_{j,m}^2}{\sum_j \sum \hat{\delta}_{j,m}^2}$$

▷ `Step 3: Erase minority elements`

$$\{\delta'_{j,m}\}_{j=1}^{K-1} = \mathrm{Erase}(\{\hat{\delta}_{j,m}\}_{j=1}^{K-1}) \quad (8)$$

▷ `Update merged LLM parameters`

$$\Phi_m = \theta_{v,m} + \sum_{j=1}^{K-1} \eta_{j,m}\delta'_{j,m} \quad (9)$$

end
return Φ

Ref. Wan, Fanqi, et al. "Fusechat: Knowledge fusion of chat models." *arXiv preprint arXiv:2408.07990* (2024).

Grading and Submission - Baselines & Grading

	JP-Math QA accuracy	Score	Estimated Merge Time 0.5-2hr Inference Time 1-2hr /20q
Public Simple Baseline	≥ 50%	+2	
Public Strong Baseline	≥ 75%	+2	
Code Submission		+2	
Paper Reading (8 questions)		+0.5 /question +4 /total	LS-MERGE: MERGING LANGUAGE MODELS INLATENT SPACE

Grading and Submission - Judgeboi

- Please submit the submission.json to judgeboi. (only .json file is allowed)
- Do not adjust the output file format

- **5** submissions quota per day, reset at 23:59 (UTC+8).

Grading and Submission - NTU COOL

submit before deadline: **2026/05/28 23:59:59 (UTC+8)**.

No late submission is allowed.

- Submit your code to NTU COOL. (2 points)
    - Remember to submit the main **.ipynb file** that cover the **complete inference process.**
    - Your Judgeboi leaderboard score **should be reproducible** from your submitted .ipynb file, **otherwise your score will not count.**
    - Compress your code into _hw7.zip. (e.g. b13901001_hw7.zip)
    - All the English alphabets in your student ID should be in lowercase.

Grading - Regulations

- You should NOT plagiarize, if you use any other resource, you should cite it in the reference.
- Do NOT share codes or prediction files with any living creatures.
- Do NOT use any approaches to submit your results more than 5 times a day.
- Do NOT search for or use additional data for training or the answers for the testing data.
- Do NOT use closed-source LLM APIs like GPT-4, Gemini, etc.
- You should NOT modify your input file or prediction files manually.
- Make sure that TAs can reproduce the predictions using the code you submit. (Fix the random
- seed)
- Your final grade x 0.9 and get a score 0 for that homework if you violate any of the above rules
- first time (within a semester).
- Your will get F for the final grade if you violate any of the above rules multiple (> 1) times.
- Prof. Lee & TAs preserve the rights to change the rules & grades.

If any questions, you can ask us via...

- NTU Cool HW7 作業討論區
    - 如果同學的問題不涉及作業答案或隱私，請一律使用 NTU Cool 討論區)
    - 助教們會優先回答 NTU Cool 討論區上的問題
- Email
    - ntu-ml-2026-spring-ta@googlegroups.com
    - The title should begin with “[ML 2026 Spring HW7]”
- TA Hours
    - Each Friday before / after class
        - (Fri.) 13.20 ~ 14.10 / 17:30~18:00
        - Location: 博理 112

Reference

pretrained model:

[Evolutionary Optimization of Model Merging Recipes]

PEFT related:
https://huggingface.co/docs/peft/index,
https://huggingface.co/docs/peft/developer_guides/model_merging

paper reading:

LS-MERGE: MERGING LANGUAGE MODELS INLATENT SPACE

papers of Merging Algorithms: Task Arithmetic, TIES, DARE, SCE

Mergitkit: https://github.com/arcee-ai/mergekit