Mitigating Parameter Interference in Model Merging via Sharpness-Aware Fine-Tuning

Yeoreum Lee[1], Jinwook Jung[1], Sungyong Baik[1,2†]

[1] Dept. of Artificial Intelligence, [2] Dept. of Data Science

{leeyeoreum01, jjw970517, dsybaik}@hanyang.ac.kr

[†] Corresponding author

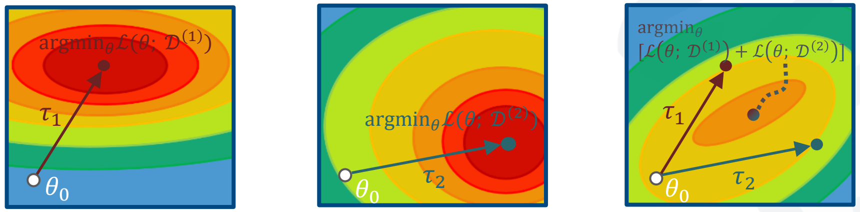
HANYANG UNIVERSITY

CONTENTS

ICLR

HANYANG UNIVERSITY

Parameter interference between task-specific models can degrade the performance of the merged multi-task model on individual tasks



(a) fine-tuning on task 1 (b) fine-tuning on task 2 (c) merging task 1 and 2

Figure 1: Loss landscapes of the task-specific models (a and b) and the merged model (c). Since task-specific models converge to distant minima, parameter interference arises after merging due to differences in parameter magnitude and sign

Successful model merging requires both *(1) less performance gap between a merged model and each fine-tuned model (i.e., less parameter interference)* **and** *(2) performance of each fine-tuned model*

Successful model merging requires both *(1) less performance gap between a merged model and each fine-tuned model (i.e., less parameter interference)* **and** *(2) performance of each fine-tuned model*

* $\theta_{merge}(\theta)$ is to demonstrate that θ_{merge} changes as θ is optimized, while considering parameters for other tasks to be fixed

$$\boldsymbol{\theta}_t = \underset{\boldsymbol{\theta}}{\operatorname{argmin}} \; \underbrace{\mathcal{L}(\boldsymbol{\theta}_{\text{merge}}(\boldsymbol{\theta}); \mathcal{D}^{(t)}) - \mathcal{L}(\boldsymbol{\theta}; \mathcal{D}^{(t)})}_{\text{Objective (1)}} + \underbrace{\mathcal{L}(\boldsymbol{\theta}; \mathcal{D}^{(t)})}_{\text{Objective (2)}}$$

Successful model merging requires both *(1) less performance gap between a merged model and each fine-tuned model (i.e., less parameter interference)* **and** *(2) performance of each fine-tuned model*

* $\theta_{merge}(\theta)$ is to demonstrate that θ_{merge} changes as θ is optimized, while considering parameters for other tasks to be fixed

$$\boldsymbol{\theta}_t = \operatorname*{argmin}_{\boldsymbol{\theta}} \underbrace{\mathcal{L}(\boldsymbol{\theta}_{\text{merge}}(\boldsymbol{\theta}); \mathcal{D}^{(t)}) - \mathcal{L}(\boldsymbol{\theta}; \mathcal{D}^{(t)})}_{\text{Objective (1)}} + \underbrace{\mathcal{L}(\boldsymbol{\theta}; \mathcal{D}^{(t)})}_{\text{Objective (2)}}$$

$$\boldsymbol{\theta}_t = \operatorname*{argmin}_{\boldsymbol{\theta}} \mathcal{L}(\boldsymbol{\theta} + \sum_{s \neq t} \alpha_s \boldsymbol{\tau}_s + (\alpha_t - 1)\boldsymbol{\tau}; \mathcal{D}^{(t)})$$

Successful model merging requires both *(1) less performance gap between a merged model and each fine-tuned model (i.e., less parameter interference)* **and** *(2) performance of each fine-tuned model*

* $\theta_{merge}(\theta)$ is to demonstrate that θ_{merge} changes as θ is optimized, while considering parameters for other tasks to be fixed

$$\boldsymbol{\theta}_t = \operatorname*{argmin}_{\boldsymbol{\theta}} \underbrace{\mathcal{L}(\boldsymbol{\theta}_{\text{merge}}(\boldsymbol{\theta}); \mathcal{D}^{(t)}) - \mathcal{L}(\boldsymbol{\theta}; \mathcal{D}^{(t)})}_{\text{Objective (1)}} + \underbrace{\mathcal{L}(\boldsymbol{\theta}; \mathcal{D}^{(t)})}_{\text{Objective (2)}}$$

$$\boldsymbol{\theta}_t = \operatorname*{argmin}_{\boldsymbol{\theta}} \mathcal{L}(\boldsymbol{\theta} + \underline{\sum_{s \neq t} \alpha_s \boldsymbol{\tau}_s + (\alpha_t - 1)\boldsymbol{\tau}}; \mathcal{D}^{(t)})$$

parameter offsets that would be introduced after model merging

→ <u>unknown perturbation that would cause parameter interference</u>

02 Our Method

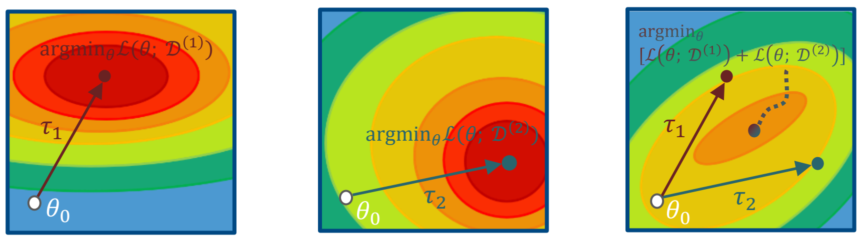
How to solve parameter interference?

$$\boldsymbol{\theta}_t = \underset{\boldsymbol{\theta}}{\operatorname{argmin}}\, \mathcal{L}(\boldsymbol{\theta} + \underline{\sum_{s \neq t} \alpha_s \boldsymbol{\tau}_s + (\alpha_t - 1)\boldsymbol{\tau}}; \mathcal{D}^{(t)})$$

parameter offsets that would be introduced after model merging

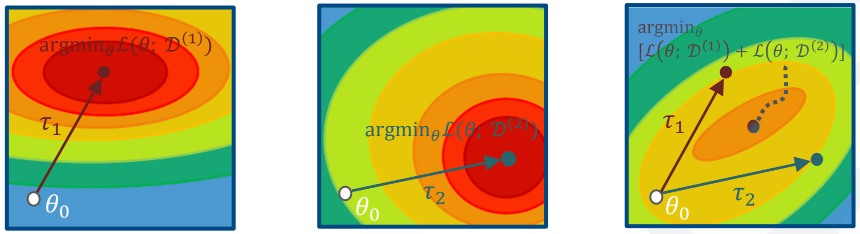
→ <u>unknown perturbation that would cause parameter interference</u>

This perturbation would take a merged model away from the found local minimum of each task to be merged



(a) fine-tuning on task 1 (b) fine-tuning on task 2 (c) merging task 1 and 2

This perturbation would take a merged model away from the found local minima of each task to be merged



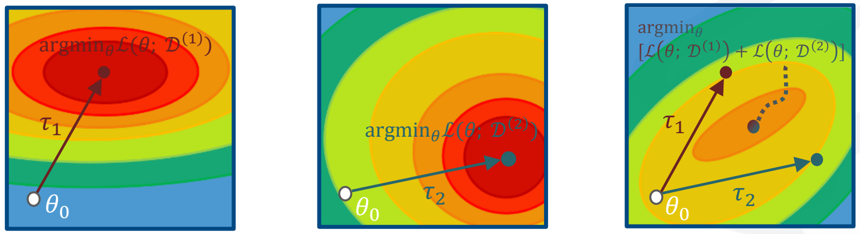
(a) fine-tuning on task 1 (b) fine-tuning on task 2 (c) merging task 1 and 2

<u>If the local minima of each task are not flat enough</u>, the new location (i.e., merged model parameters) brought by perturbations will most likely have a higher loss, <u>resulting in parameter interference</u>

02 Our Method

How to solve parameter interference?

If the local minima of each task are not flat enough, the new location (i.e., merged model parameters) brought by perturbations will most likely have a higher loss, resulting in parameter interference



(a) fine-tuning on task 1 (b) fine-tuning on task 2 (c) merging task 1 and 2

"Since flat minima can effectively prevent the loss from increasing after parameter perturbations (e.g., model merging), we use the perturbation of Adaptive Sharpness-Aware Minimization (ASAM) $\hat{\epsilon}_{ASAM}$ as a surrogate of the unknown perturbation $\sum_{s\neq t}\alpha_s\tau_s + (\alpha_t - 1)\tau$"

"Since flat minima can effectively prevent the loss from increasing after parameter perturbations (e.g., model merging), we use the perturbation of Adaptive Sharpness-Aware Minimization (ASAM) $\hat{\epsilon}_{ASAM}$ as a surrogate of the unknown perturbation $\sum_{s \neq t} \alpha_s \tau_s + (\alpha_t - 1)\tau$"

$$\boldsymbol{\theta}_t = \operatorname*{argmin}_{\boldsymbol{\theta}} \mathcal{L}(\boldsymbol{\theta} + \underline{\sum_{s \neq t} \alpha_s \boldsymbol{\tau}_s + (\alpha_t - 1)\boldsymbol{\tau}}; \mathcal{D}^{(t)})$$

$$\hat{\boldsymbol{\epsilon}}_{\text{ASAM}} = \rho \frac{\boldsymbol{\theta}^2 \nabla_{\boldsymbol{\theta}} \mathcal{L}(\boldsymbol{\theta}; \mathcal{D})}{\|\nabla_{\boldsymbol{\theta}} \mathcal{L}(\boldsymbol{\theta}; \mathcal{D})\|}$$

"Our proposed method: Sharpness-Aware Fine-Tuning (SAFT)"

ICLR

Weight disentanglement, which indicates the output difference between the merged model and task-specific models, indirectly measure parameter interference [1]
→ SAFT can strengthen weight disentanglement

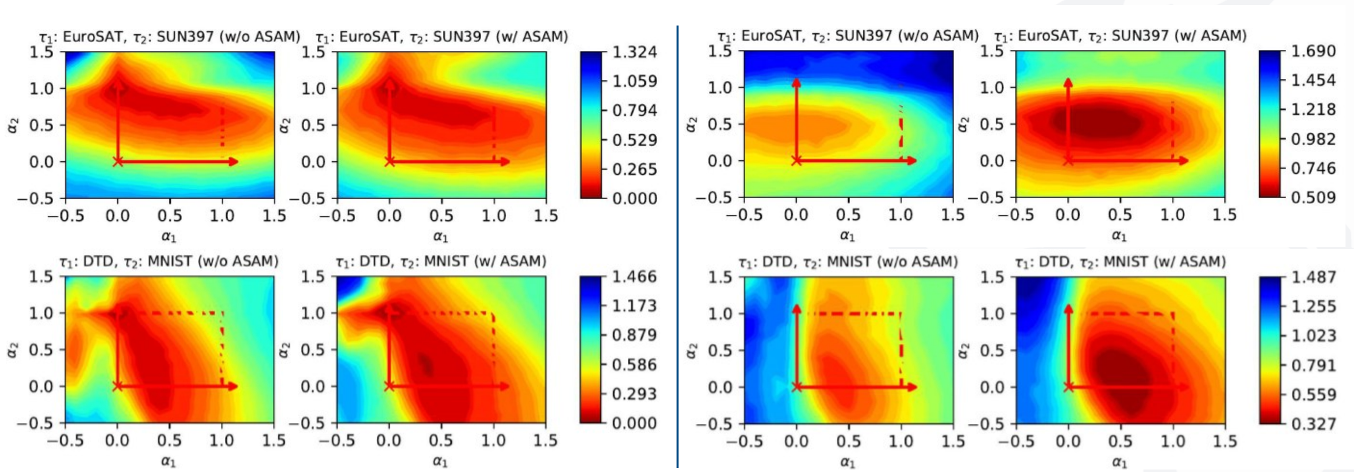


Figure 2: Disentanglement error visualization of two-task-merged model $\xi(\alpha_1, \alpha_2)$ (left) and eight-task-merged model $\xi_{\text{all}}(\alpha_1, \alpha_2)$ (right) across two tasks

$$\xi(\alpha_1,\alpha_2)=\sum_{t=1}^{2}\mathbb{E}_{\boldsymbol{x}\in X^{(t)}}[\mathrm{dist}(f(\boldsymbol{x};\boldsymbol{\theta}_0+\alpha_t\boldsymbol{\tau}_t), f(\boldsymbol{x};\boldsymbol{\theta}_0+\alpha_1\boldsymbol{\tau}_1+\alpha_2\boldsymbol{\tau}_2))]$$

Disentanglement error $\xi(\alpha_1, \alpha_2)$ between of a two-task-merged model and task-specific models across two tasks [1]

$$\xi_{\text{all}}(\alpha_1,\alpha_2)=\sum_{t=1}^{2}\mathbb{E}_{\boldsymbol{x}\in X^{(t)}}\left[\mathrm{dist}\left(f(\boldsymbol{x};\boldsymbol{\theta}_0+\alpha_t\boldsymbol{\tau}_t), f(\boldsymbol{x};\boldsymbol{\theta}_0+\alpha_1\boldsymbol{\tau}_1+\alpha_2\boldsymbol{\tau}_2+\sum_{s\notin\{1,2\}}\alpha_s\boldsymbol{\tau}_s)\right)\right]$$

Disentanglement error $\xi_{\text{all}}(\alpha_1, \alpha_2)$ between of an eight-task-merged model and task-specific models across two tasks

HANYANG UNIVERSITY

[1] Ortiz-Jimenez et al., Task Arithmetic in the Tangent Space: Improved Editing of Pre-Trained Models, NeurIPS, 2023.

If Cross-Task Linearity (CTL) holds between the merged model and the task-specific models, the merged model can be disentangled into each task-specific model, leading to improved weight disentanglement [1]
→ SAFT can strengthen CTL

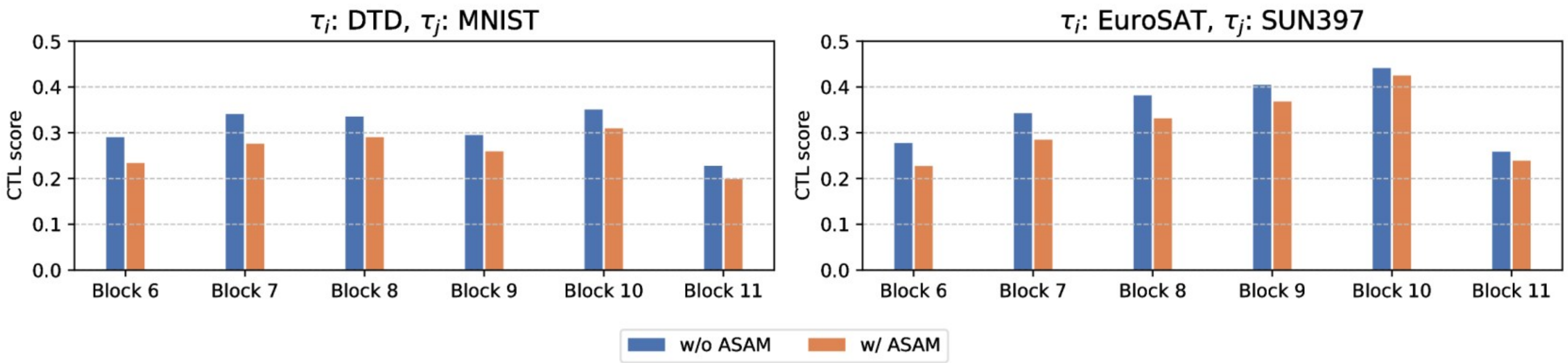


Figure 3: Verification of CTL between merged model and task-specific models

$$\cos^{(\ell)}(\boldsymbol{x}; 2\lambda\boldsymbol{\tau}_s, 2\lambda\boldsymbol{\tau}_t)$$
$$= \cos\left[f^{(\ell)}(\boldsymbol{x}; \boldsymbol{\theta}_0 + \lambda(\boldsymbol{\tau}_s + \boldsymbol{\tau}_t)), \frac{1}{2}f^{(\ell)}(\boldsymbol{x}; \boldsymbol{\theta}_0 + 2\lambda\boldsymbol{\tau}_s) + \frac{1}{2}f^{(\ell)}(\boldsymbol{x}; \boldsymbol{\theta}_0 + 2\lambda\boldsymbol{\tau}_t)\right]$$

To calculate CTL score, the cosine similarity between the layer output of a merged model and the averaged layer outputs of the task-specific models is used [1]

[1] Zhou et al., On the Emergence of Cross-Task Linearity in Pretraining-Finetuning Paradigm, ICML, 2024.

ICLR

We demonstrate that SAFT finds flatter minima on the joint-task loss landscape by proving joint-task loss linearity.

→ **A visualization of the joint-task loss landscape provides further empirical support for this.**

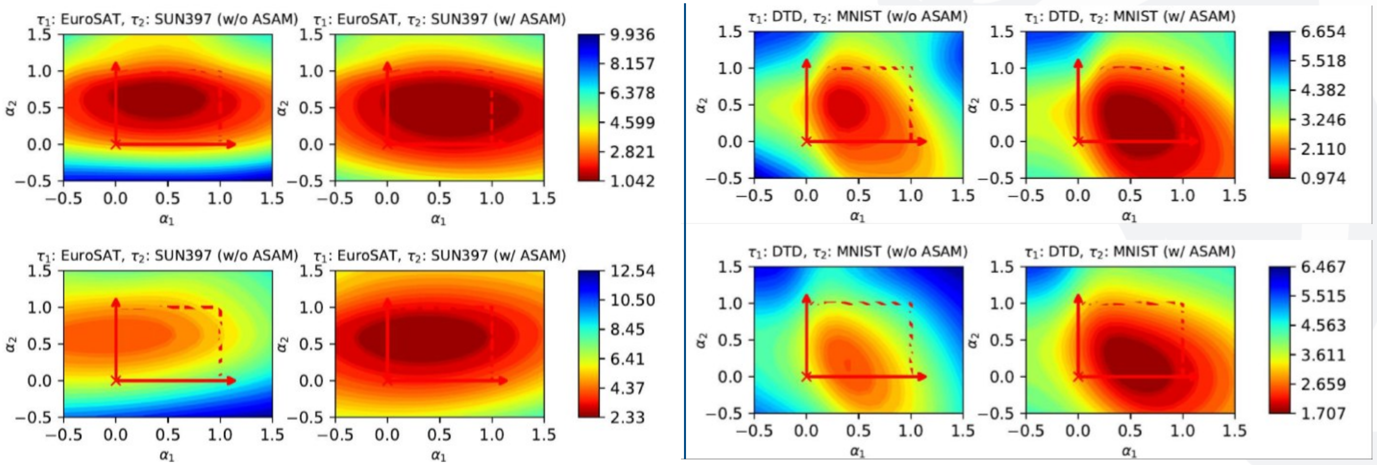


Figure 4: Joint-task loss landscape of two-task-merged model (left) and eight-task-merged model (right) across two tasks

SAFT exhibits synergy with various finetuning methods, model merging methods, and models.

Fine-tuning method (→)	SGD		FTTS		FTLO	
	Abs.	Norm.	Abs.	Norm.	Abs.	Norm.
w/o SAFT-ASAM	68.23	75.47	78.35	86.83	75.93	85.74
w/ SAFT-ASAM (Ours)	**69.45**	**76.32**	**79.38**	**87.72**	**77.49**	**88.77**

Table 1: Multi-task performance across different fine-tuning methods

Merging method (→)	Weight averaging		Task arithmetic		TIES merging	
	Abs.	Norm.	Abs.	Norm.	Abs.	Norm.
	ViT-B/32					
SGD	65.72	72.91	68.23	75.47	74.57	82.29
SAFT-ASAM (Ours)	**66.76**	**73.62**	**69.45**	**76.32**	**75.45**	**82.86**
	ViT-B/16					
SGD	71.58	77.37	73.40	79.31	77.94	84.04
SAFT-ASAM (Ours)	**71.84**	**77.53**	**76.77**	**82.50**	**80.14**	**86.23**

Table 2: Multi-task performance across different model merging methods and image encoders

Motivation: If fine-tuned task-specific models converge to flat minima, a multi-task model merged from these models is less affected by parameter interference.

Method: We propose a novel objective function for multi-task model merging and, by connecting it to Sharpness-Aware Minimization (SAM), introduce Sharpness-Aware Fine-Tuning (SAFT).

Contribution: We demonstrate that SAFT can mitigate parameter interference by showing that our method can enhance weight disentanglement, Cross-Task Linearity (CTL), and joint-task loss linearity.