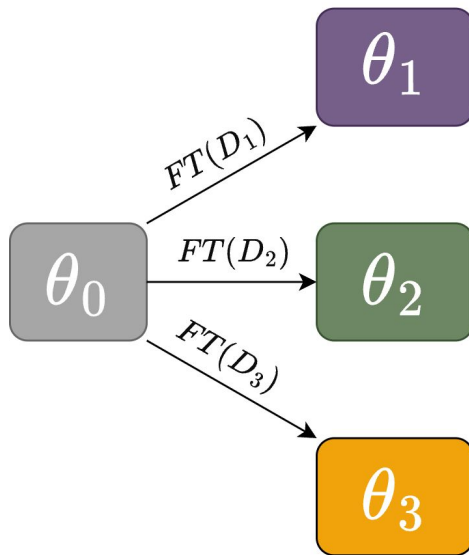

No Task Left Behind: Isotropic Model Merging with Common and Task-Specific Subspaces

**Daniel Marczak
Bartłomiej Twardowski**

**Simone Magistri Sebastian Cygert
Andrew D. Bagdanov Joost van de Weijer**

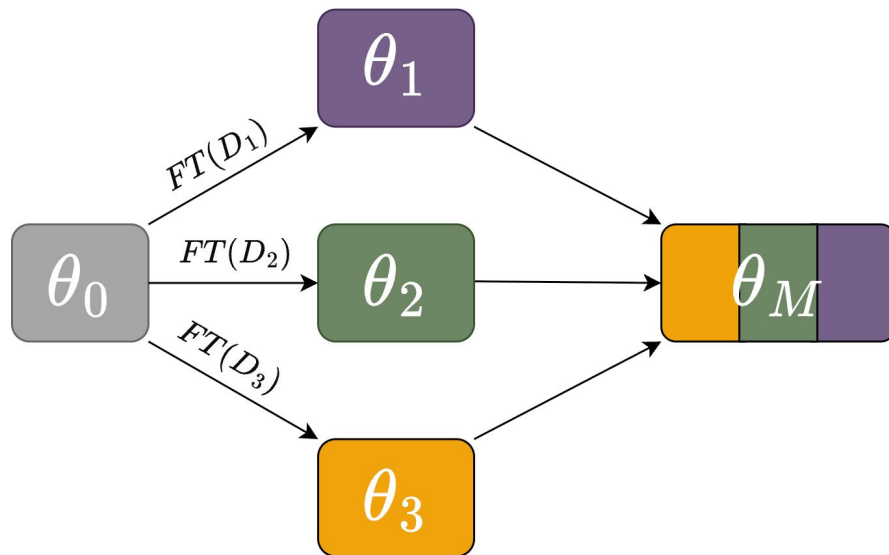
Model merging

- Given: **multiple task-specific models** – fine-tunings of the same pre-trained model on different tasks



Model merging

- *Given:* **multiple task-specific models** – fine-tunings of the same pre-trained model on different tasks
- *Objective:* **combine the weights** of task-specific models into a **single multi-task model**



Research gap

- *Previous works:* operate on *task vectors*
- *Problem:* **overlooks crucial structural information**

$$\tau_t = \text{vec}(\theta_t - \theta_0)$$

Research gap

- *Previous works*: operate on *task vectors*
- *Problem*: **overlooks crucial structural information**

$$\tau_t = \text{vec}(\theta_t - \theta_0)$$

- *Our approach*: operates on per-layer task matrices
- *Advantage*: **leverages the structure of weight update matrices**

$$\Delta_t^{(\ell)} = \theta_t^{(\ell)} - \theta_0^{(\ell)}$$

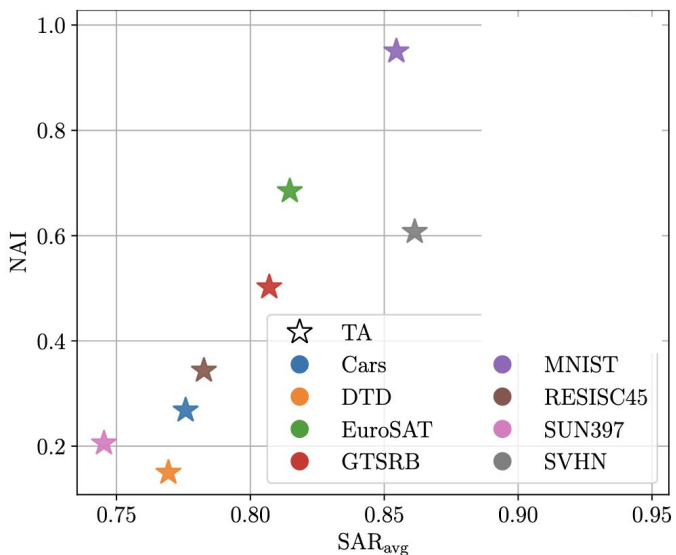
Subspace Alignment Ratio (SAR)

How much of a task matrix is contained in the subspace spanned by top components of merged matrix?

$$\text{SAR}(\Delta_t, \Delta_M; k_M) = \frac{||\Pi_{k_M, M} \Delta_t||_F}{||\Delta_t||_F}$$

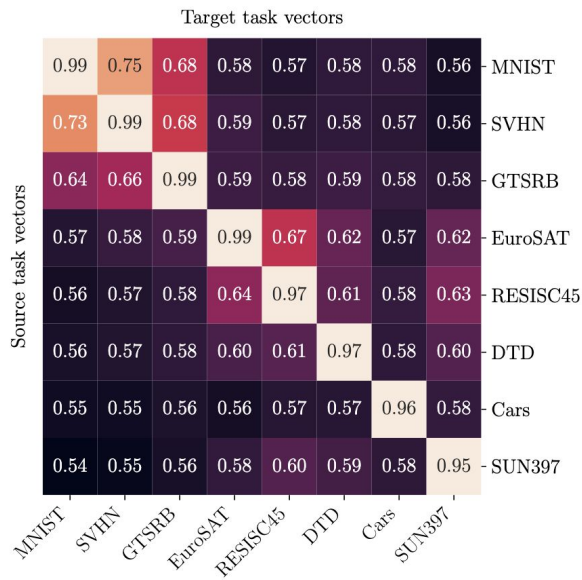
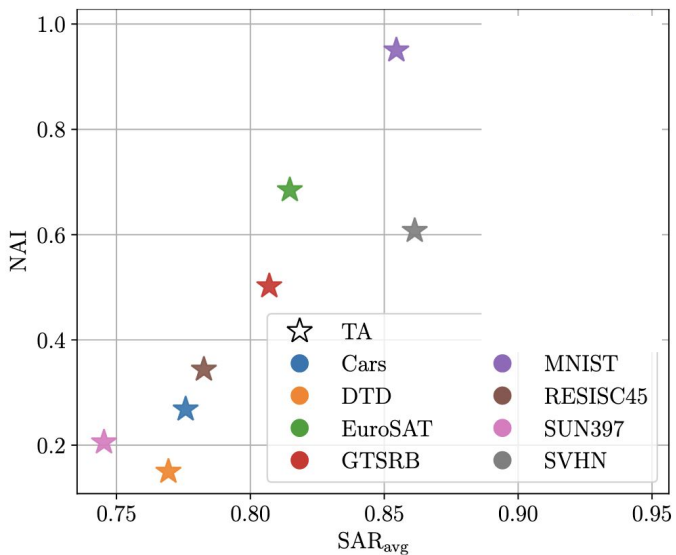
← projection of Δ_t onto subspace spanned by top components of Δ_M

Subspace Alignment vs Performance Improvement

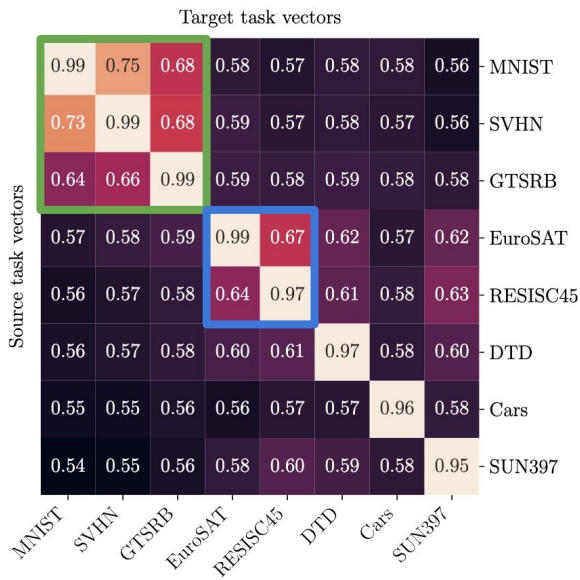
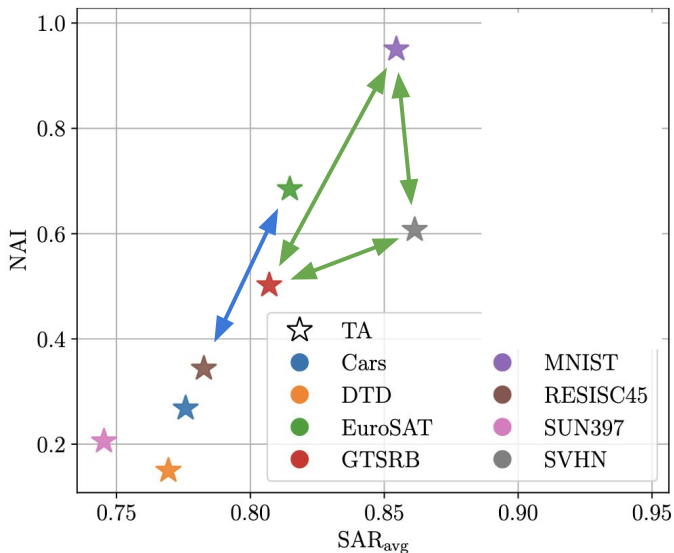


Alignment with merged model (x-axis)
varies across datasets and **highly**
correlates with performance
improvements (y-axis)

Subspace Alignment vs Performance Improvement



Subspace Alignment vs Performance Improvement

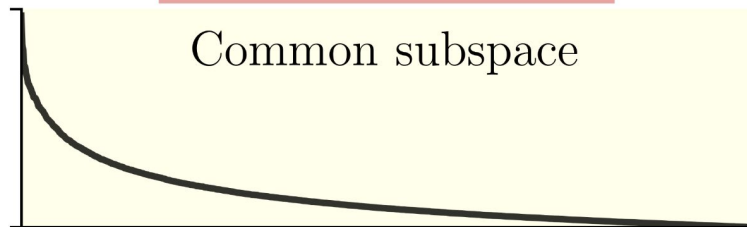


- clusters of datasets highly aligned with each other
- also highly aligned with merged model
→ high performance
- not aligned datasets
→ low performance

How to improve the alignment?

spectrum of singular values after merging with

Task Arithmetic

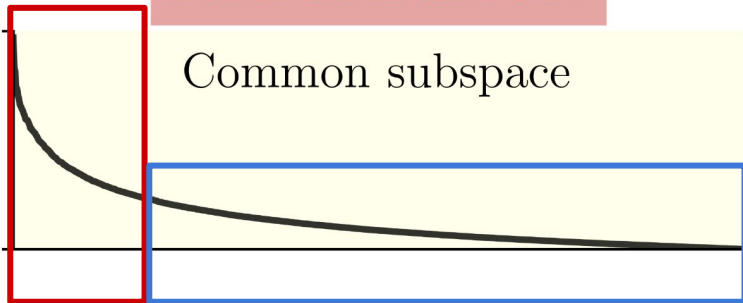


How to improve the alignment?

spectrum of singular values after merging with

Task Arithmetic

Common subspace



↑
this part corresponds to
highly aligned tasks
(amplified directions)

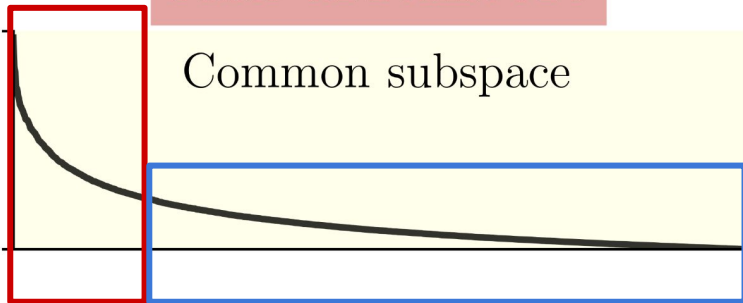
↑
this part corresponds to less correlated
tasks (underrepresented directions)

How to improve the alignment?

spectrum of singular values after merging with

Task Arithmetic

Common subspace

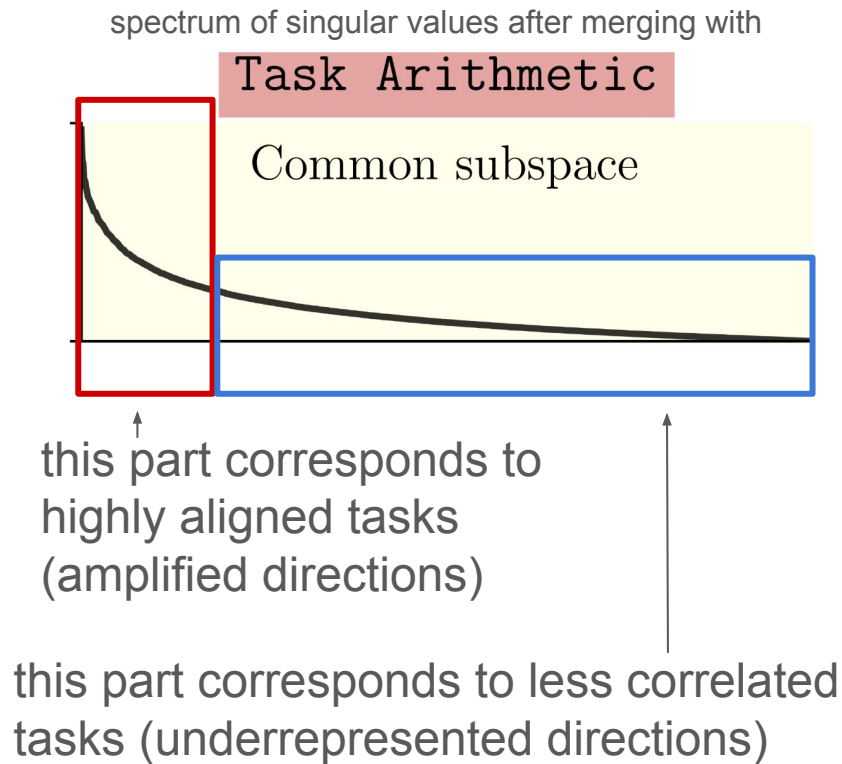


↑
this part corresponds to
highly aligned tasks
(amplified directions)

↑
this part corresponds to less correlated
tasks (underrepresented directions)

How can we reduce the skewness
towards dominant directions?

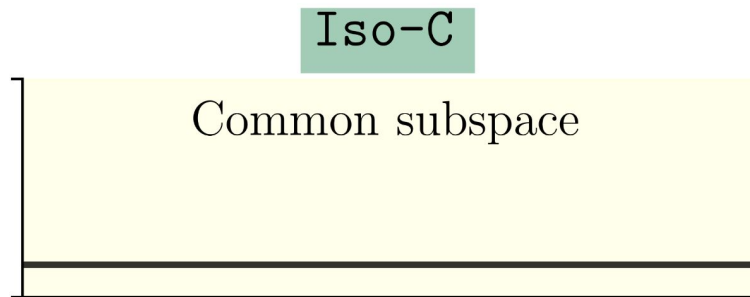
How to improve the alignment?



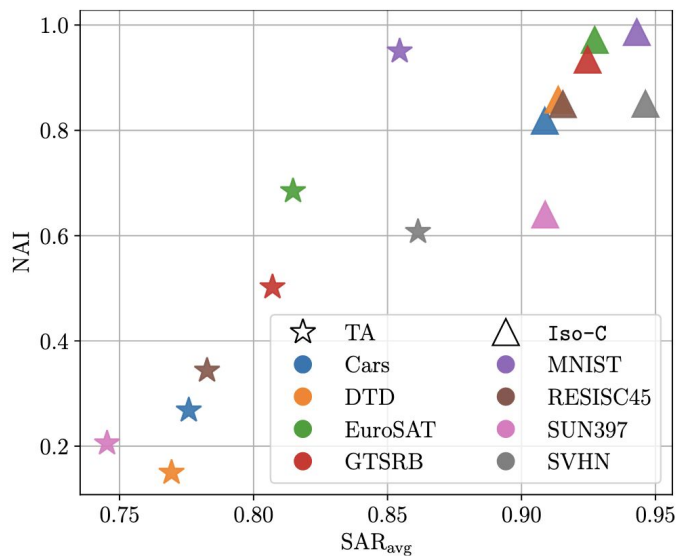
How can we reduce the skewness towards dominant directions?

We propose to use uniform singular values ensuring that the transformation is isotropic

$$\bar{\sigma} = \frac{1}{r} \sum_{i=1}^r \sigma_i$$



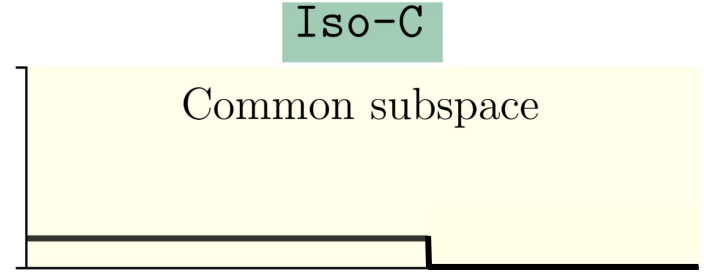
Subspace Alignment vs Performance Improvement



- Iso-C improves the alignment between each task matrix and merged matrix...
- ... and **performance improvement follows**

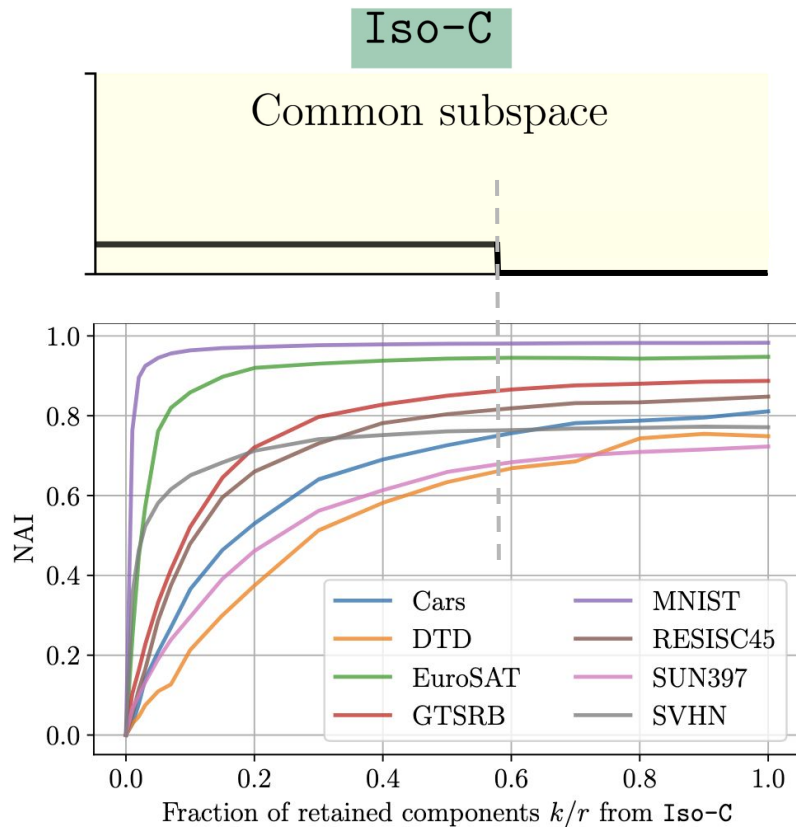
Are all of the directions important for the performance?

- Let's start from Iso-C
- Truncate the spectrum keeping k leftmost directions (associated with the highest singular values of Δ_{TA})



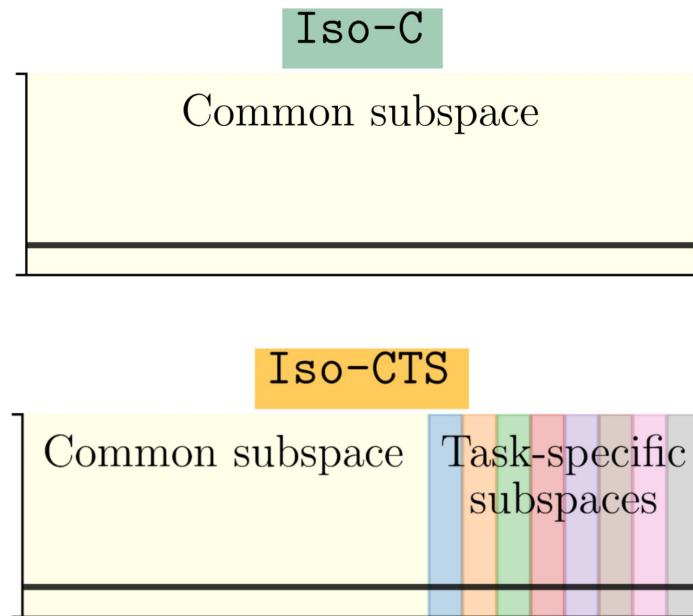
Are all of the directions important for the performance?

- Let's start from Iso-C
- Truncate the spectrum keeping k leftmost directions (associated with the highest singular values of Δ_{TA})
- Leftmost directions are responsible for most of the performance increase
- Rightmost directions do **not contribute** that much
- Can we **utilize the subspace** of rightmost components **more effectively**?



Isotropic Merging in Common and Task-Specific Subspaces

- Let's keep the useful components and call it common subspace (as it is based on summation of task-specific matrices)
- Let's replace not very useful bottom components with the task-specific components that are orthogonal to the common subspace



Fully fine-tuned vision models


Method	ViT-B/32			ViT-B/16			ViT-L/14		
	8 tasks	14 tasks	20 tasks	8 tasks	14 tasks	20 tasks	8 tasks	14 tasks	20 tasks
Zero-shot	48.3	57.2	56.1	55.3	61.3	59.7	64.7	68.2	65.2
Fine-tuned	92.8	90.9	91.3	94.6	92.8	93.2	95.8	94.3	94.7
Weight Averaging	66.3 _(72.1)	64.3 _(71.1)	61.0 _(67.5)	72.2 _(76.6)	69.5 _(74.8)	65.3 _(70.4)	79.6 _(83.2)	76.7 _(81.1)	71.6 _(75.6)
Task Arithmetic	70.8 _(76.5)	65.3 _(72.1)	60.5 _(66.8)	75.4 _(79.6)	70.5 _(75.9)	65.8 _(70.8)	84.9 _(88.7)	79.4 _(84.0)	74.0 _(78.1)
TIES	75.1 _(81.0)	68.0 _(74.8)	63.4 _(69.9)	79.7 _(84.3)	73.2 _(78.7)	68.2 _(73.3)	86.9 _(90.7)	79.5 _(84.1)	75.7 _(79.8)
Consensus TA	75.0 _(80.8)	70.4 _(77.4)	65.4 _(72.0)	79.4 _(83.9)	74.4 _(79.9)	69.8 _(74.9)	86.3 _(90.1)	82.2 _(86.9)	79.0 _(83.2)
TSV-M	85.9 _(92.3)	80.1 _(87.9)	77.1 _(84.3)	89.0 _(93.9)	84.6 _(91.0)	80.6 _(86.5)	93.0 _(97.0)	89.2 _(94.4)	87.7 _(92.5)
Iso-C (Ours)	86.3_(92.9)	80.3_(88.1)	75.5_(82.5)	90.6_(95.6)	84.8_(91.1)	79.6_(85.4)	94.2_(98.3)	89.3_(94.5)	87.6_(92.2)
Iso-CTS (Ours)	86.2_(92.8)	81.7_(89.7)	78.1_(85.5)	91.1_(96.1)	86.4_(92.8)	82.4_(88.4)	94.7_(98.8)	91.0_(96.3)	90.1_(94.9)

state-of-the-art results for all evaluated settings

LoRA-adapted vision models

Method	ViT-B/32	ViT-L/14
TA	63.7	74.4
TIES	63.7	75.2
DARE-TIES	63.7	74.7
KnOTS-TIES	68.0	78.2
KnOTS-DARE-TIES	63.9	75.6
Iso-C (Ours)	<u>73.6</u>	<u>83.7</u>
Iso-CTS (Ours)	73.7	85.3

LoRA-specific
merging method



Language models

Method	8 tasks	7 tasks
	(Zhou et al., 2022)	(Yadav et al., 2023)
Fine-tuned	80.7	85.9
Weight Averaging	56.4	60.5
Task Arithmetic	63.8	69.2
TIES	62.8	71.9
Fisher Merging	57.7	61.0
RegMean	69.1	74.3
MaTS	72.5	81.5
Iso-C (Ours)	75.6	83.3
Iso-CTS (Ours)	<u>75.2</u>	<u>82.8</u>

Summary

- Iso-C **flattens the spectrum of singular values** of common space
- Iso-CTS adds **task-specific subspaces** on top of Iso-C
- Both methods achieve **state-of-the-art** results across vision and language tasks, both fully and LoRA fine-tuned

