The Twelfth International Conference on Learning Representations

AdaMerging: Adaptive Model Merging for Multi-Task Learning

Enneng Yang, Zhenyi Wang, Li Shen, Shiwei Liu, Guibing Guo,
Xingwei Wang, Dacheng Tao

JDT 京东科技

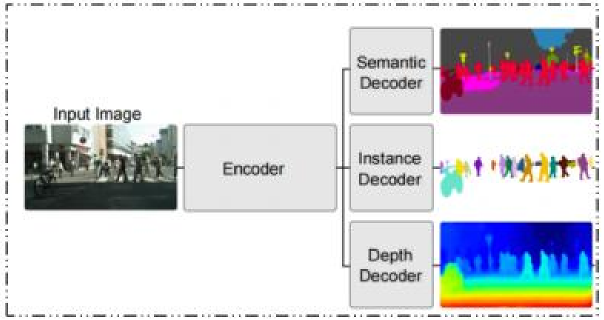
01

Background

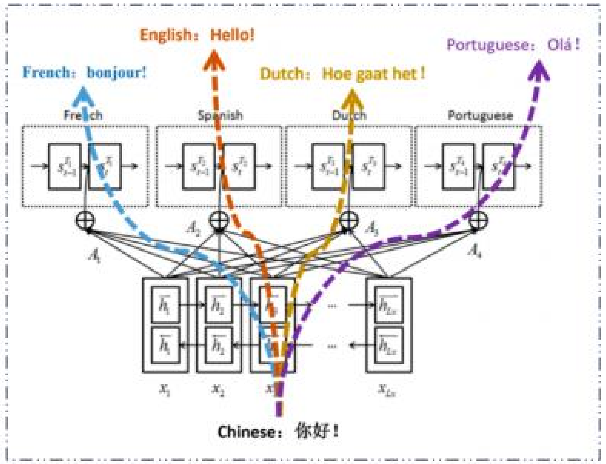
Introduction to MTL

Goal of MTL： to train **a single model** collaboratively using data from multiple tasks to enable knowledge transfer.

- **Computer Vision**



- **Multilingual Translation**



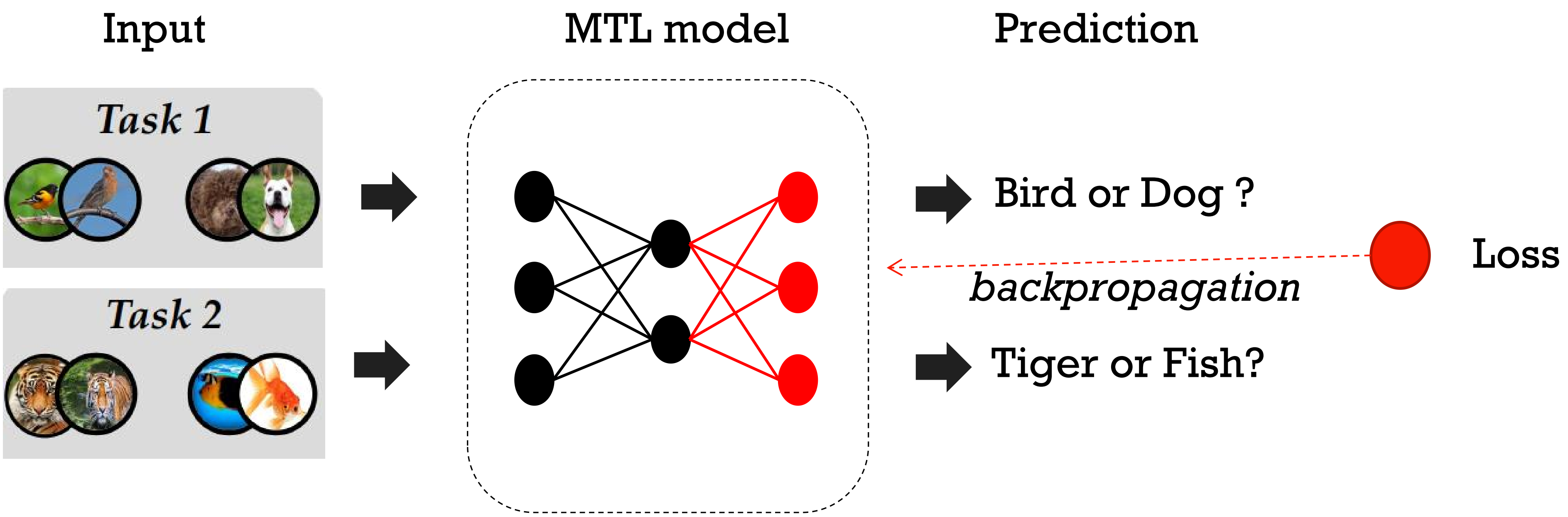
- **Recomendation System**

MTL is widely used in CV, NLP, RecSys, etc.

Introduction to MTL

The core steps of traditional MTL include:

- Collect **training data** for the multi-tasks
- Design a MTL **architecture**
- **Optimize** the parameters



"Learn from Raw Data"

State-of-the-art MTL solution

However, there are two problems with using raw data for MTL:

Learn from Raw Data

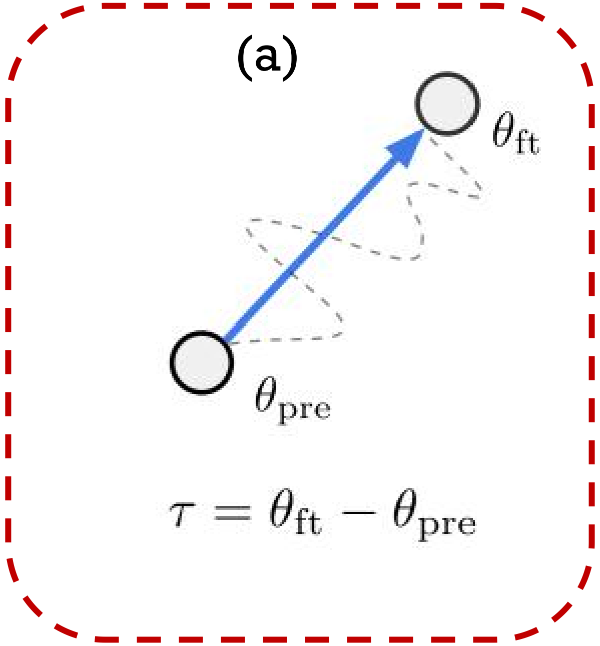
◆ High data management/storage costs

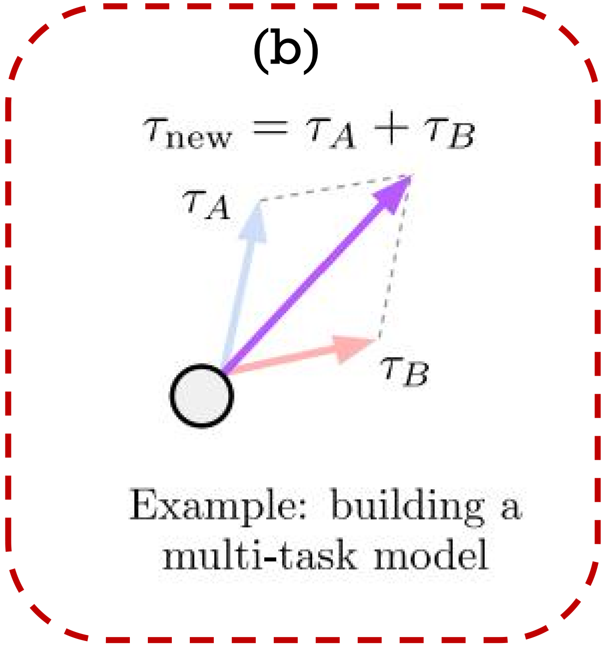
◆ Data privacy risk

"Instead, can we learn from well-trained models?"

State-of-the-art MTL solution

Recent research (called Task Arithmetic) has shown that multi-task learning can be performed by merging independently trained models.





$$\tau_t = \theta_{\text{ft}}^t - \theta_{\text{pre}}$$

$$\tau_{\text{new}} = \sum_i \tau_i$$

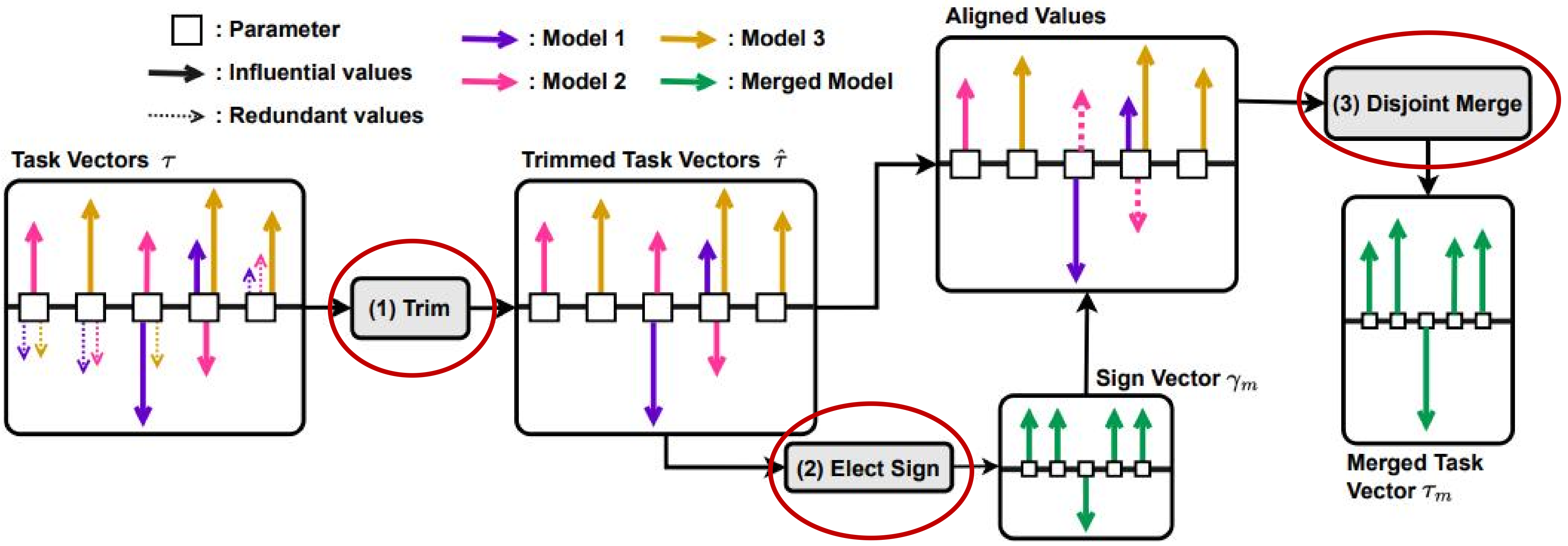
$$\theta_{\text{new}} = \theta + \lambda \tau_{\text{new}}$$

(a) A **task vector** is obtained by subtracting the weights of a pre-trained model from the weights of the same model after fine-tuning.
(b) **Adding** task vectors together perform the multi-task learning.

Ilharco, Gabriel, et al. "Editing models with task arithmetic." ICLR, 2023.

State-of-the-art MTL solution

Based on task vectors, TIES-Merging further removes redundant parameter updating and solves parameter symbol conflicts to alleviate the interference of model merging.



Yadav, Prateek, et al. "Ties-Merging: Resolving interference when merging models." NeurIPS, 2023.

State-of-the-art MTL solution

Table 1: Multi-task performance when merging ViT-B/32 models on eight tasks.

Method	SUN397	Cars	RESISC45	EuroSAT	SVHN	GTSRB	MNIST	DTD	**Avg Acc**
Pretrained	62.3	59.7	60.7	45.5	31.4	32.6	48.5	43.8	48.0
Individual	75.3	77.7	96.1	99.7	97.5	98.7	99.7	79.4	90.5
Traditional MTL	73.9	74.4	93.9	98.2	95.8	98.9	99.5	77.9	88.9
Weight Averaging	65.3	63.4	71.4	71.7	64.2	52.8	87.5	50.1	65.8
Fisher Merging (Matena & Raffel, 2022)	68.6	69.2	70.7	66.4	72.9	51.1	87.9	59.9	68.3
RegMean (Jin et al., 2023)	65.3	63.5	75.6	78.6	78.1	67.4	93.7	52.0	71.8
Task Arithmetic (Ilharco et al., 2023)	55.2	54.9	66.7	78.9	80.2	69.7	97.3	50.4	69.1
Ties-Merging (Yadav et al., 2023)	59.8	58.6	70.7	79.7	86.2	72.1	98.3	54.2	72.4

Gap

However, we observe that there is still **a large performance gap** between the task vector-based model merging approach and the traditional MTL.

Ilharco, Gabriel, et al. "Editing models with task arithmetic." ICLR, 2023.
Yadav, Prateek, et al. "Ties-Merging: Resolving interference when merging models." NeurIPS, 2023.

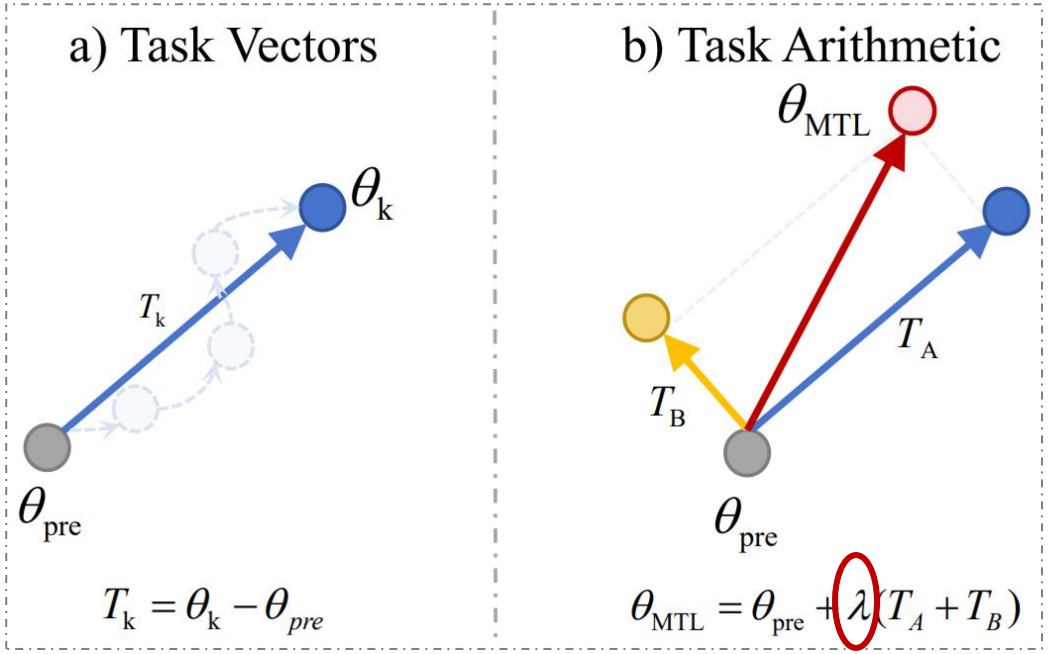
02

AdaMerging: Adaptive Model Merging for Multi-Task Learning.

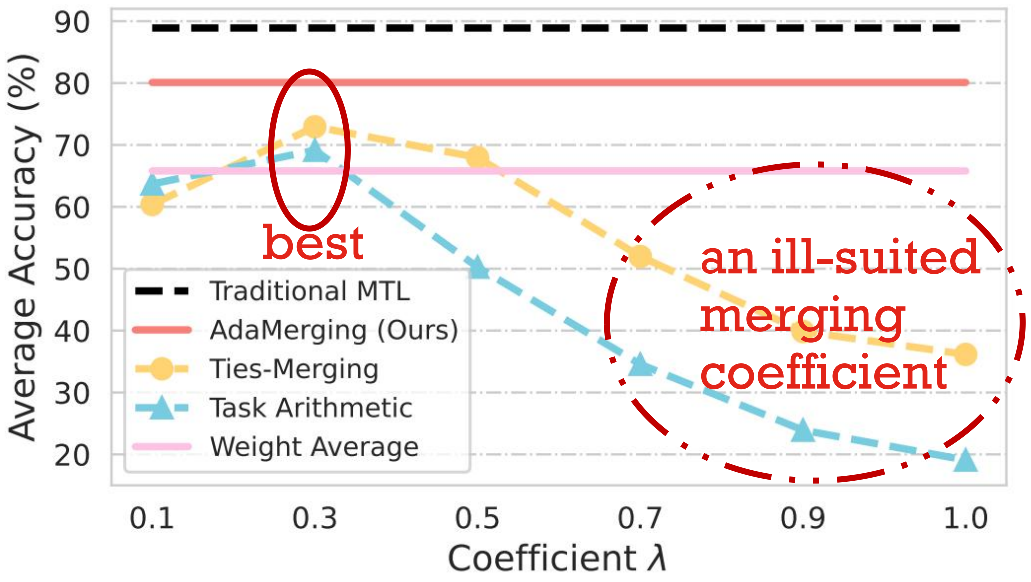
ICLR, 2024.

Our AdaMerging

A critical observation in the analysis of task vector-based MTL methods is the significance of the merging coefficient λ **associated with the task** vector.



(a) Definition of "task vector"
(b) Task Arithmetic (Ilharco et al., 2023)



The impact of coefficient λ on the average accuracy of various MTL methods on eight tasks.

Ilharco, Gabriel, et al. "Editing models with task arithmetic." ICLR, 2023.

Our AdaMerging

Further, one important question is:

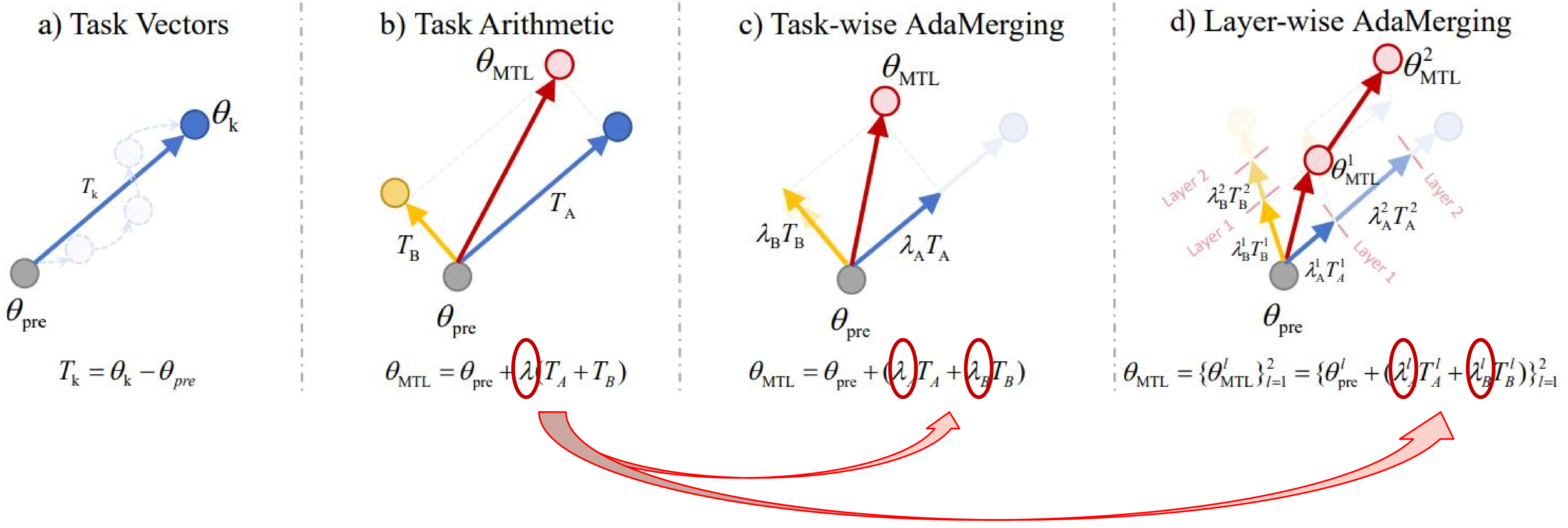
Question: Is it reasonable for all task vectors to share a merge coefficient? And, is it reasonable for all layers of a task vector to share a merge coefficient?

We think the answer is "no".

Task vectors/layers differ greatly and it is not enough to share a single coefficient.

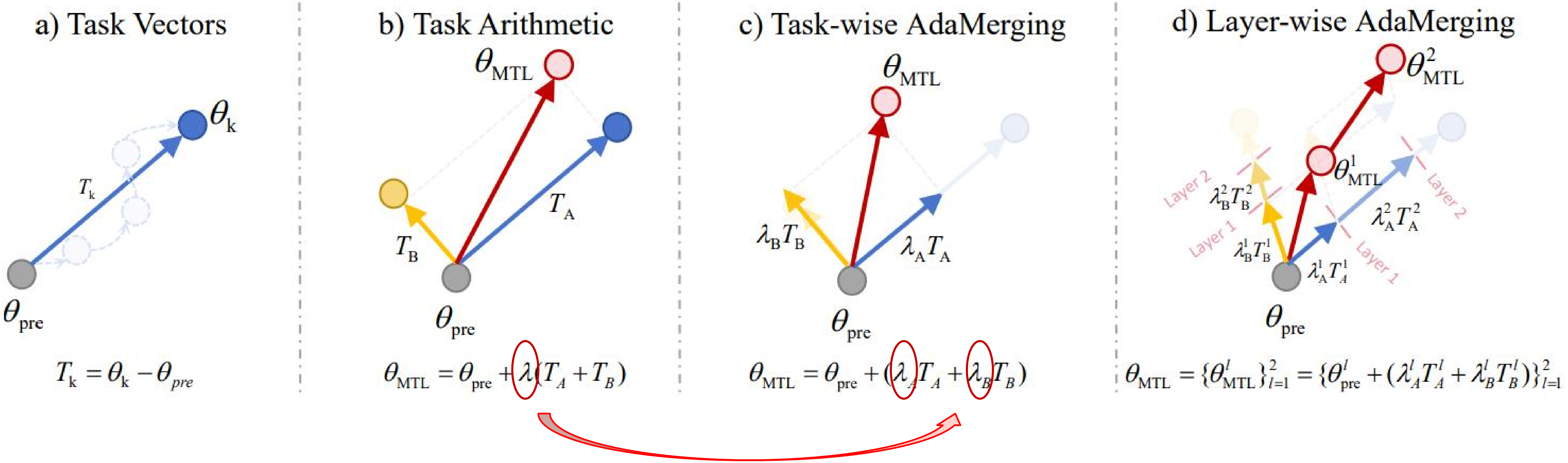
Our AdaMerging

We propose Task-wise model merging and Layer-wise model merging.



Our AdaMerging

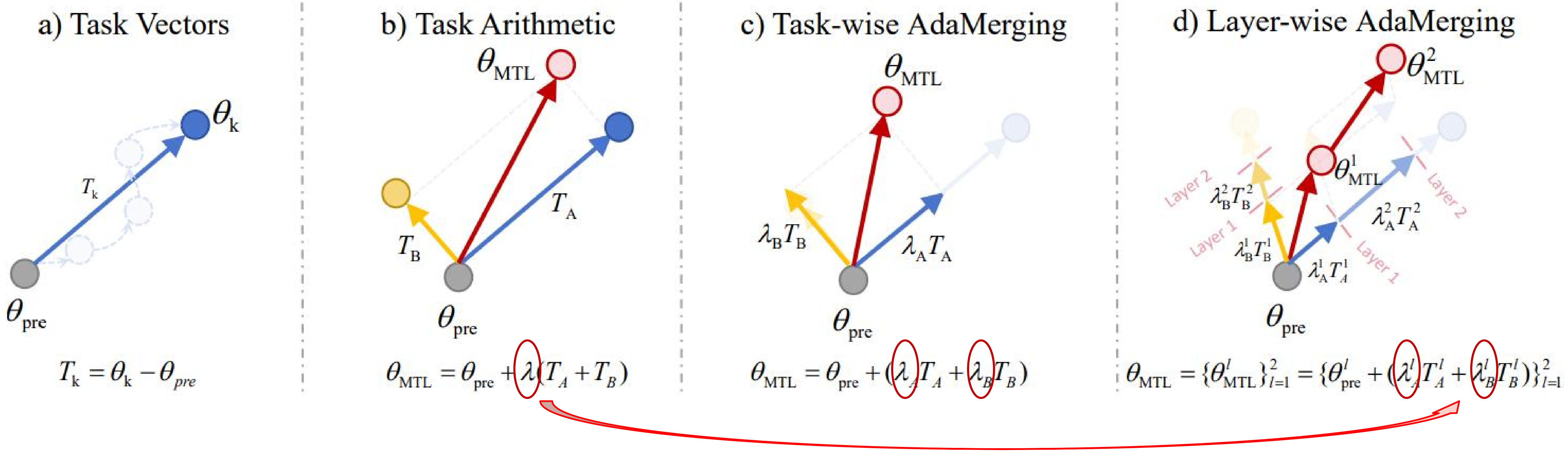
We propose Task-wise AdaMerging for multi-task model merging .



(C) ***Task-wise AdaMerging*** for MTL, which learns a different merging coefficient λ_k to each task vector T_k $(k \in \{A, B\})$.

Our AdaMerging

We propose Layer-wise AdaMerging for multi-task model merging .



(d) ***Layer-wise AdaMerging*** for MTL, which learns a different merging coefficient λ^l_k to each layer l ($l \in \{1, 2\}$) of the task vector T_k ($k \in \{A, B\}$).

Our AdaMerging

Critical Challenge: How to optimize merging coefficients?

Because we don't have the raw training data for the multiple tasks.

Inspired by test-time adaptation (Wang, Dequan, et al), we use entropy minimization of unlabeled test data as a proxy objective function.

Wang, Dequan, et al. "Tent: Fully test-time adaptation by entropy minimization." ICLR, 2021.

Our AdaMerging

Cross-Entropy Loss

$$H(y_i, \hat{y}_i) = -\sum_c^C p(y_{i,c}) \log p(\hat{y}_{i,c})$$

Shannon Entropy:

$$H(\hat{y}_i) = -\sum_c^C p(\hat{y}_{i,c}) \log p(\hat{y}_{i,c})$$

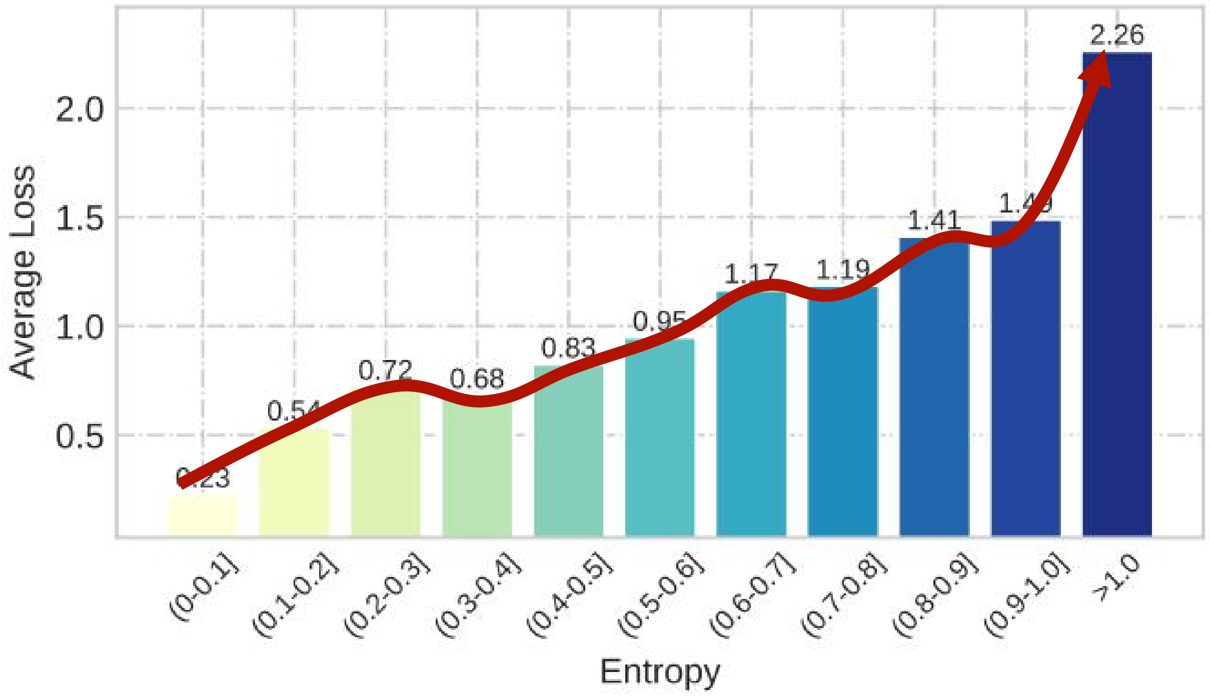
➡ Shannon entropy depends only on the output of the model.

where

- y_i: Real Label (One-hot)
- $\hat{y}_i$: The prediction probability of the model for each class
- C: Total class number

How can we verify that entropy minimization is a reasonable surrogate objective in model merging?

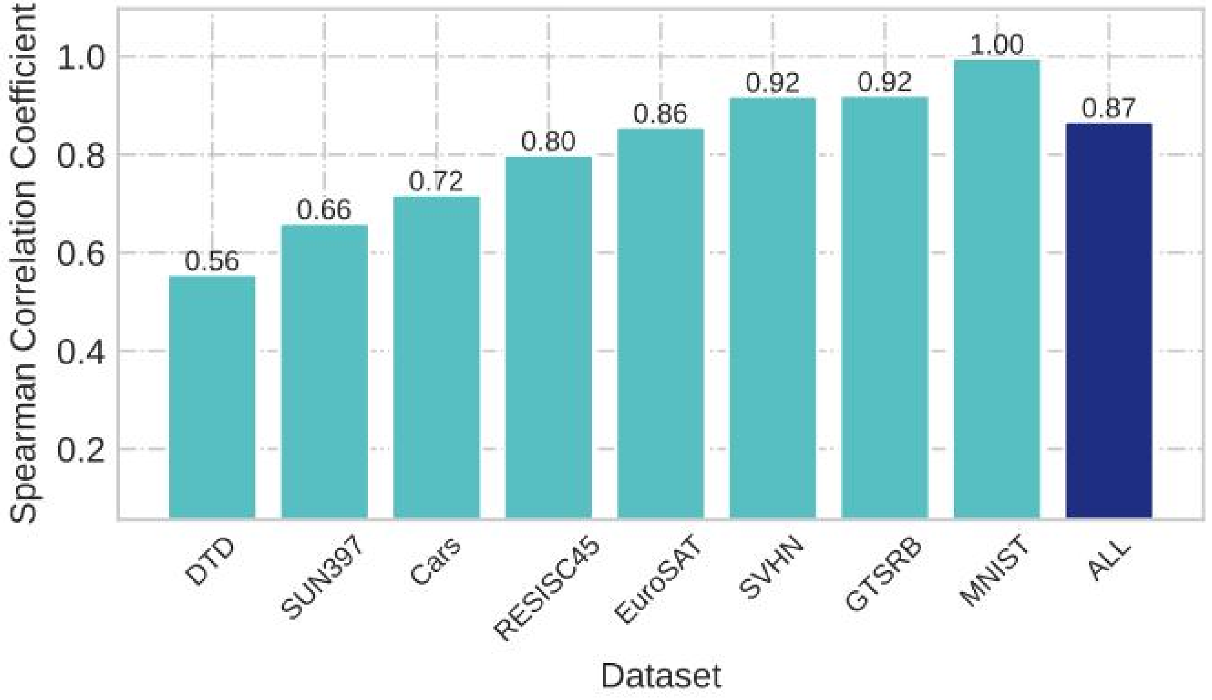
Our AdaMerging



We first group the samples according to the entropy of each sample (a total of 11 groups), and then count the true cross entropy loss in each group.

◆ Evidence 1: Samples with low entropy also have low losses.

Our AdaMerging



We also directly calculated the Spearman correlation coefficient of entropy and prediction loss.

◆ Evidence 2: We observe a higher average correlation between them.

Our AdaMerging

Optimization Objective

Based on the above verification, we take entropy minimization as the optimization proxy goal of the model merging coefficient in our AdaMerging.

$$\min_{\lambda_1,\lambda_2,\ldots,\lambda_K} \sum_{k=1}^{K} \sum_{x_i \in \mathcal{B}_k} H(f_{\theta_{MTL}}(x_i)) \text{ , where } \theta_{MTL} = \theta_{\text{pre}} + \sum_{k=1}^{K} \lambda_k T_k,$$

where B_k represents a batch of unlabeled test samples sampled in task k.

Experiments

◆ Significantly Higher MTL Performance.

Table 1: Multi-task performance when merging ViT-B/32 models on eight tasks.

Method	SUN397	Cars	RESISC45	EuroSAT	SVHN	GTSRB	MNIST	DTD	**Avg Acc**
Pretrained	62.3	59.7	60.7	45.5	31.4	32.6	48.5	43.8	48.0
Individual	75.3	77.7	96.1	99.7	97.5	98.7	99.7	79.4	90.5
Traditional MTL	73.9	74.4	93.9	98.2	95.8	98.9	99.5	77.9	88.9
Weight Averaging	65.3	63.4	71.4	71.7	64.2	52.8	87.5	50.1	65.8
Fisher Merging (Matena & Raffel, 2022)	68.6	69.2	70.7	66.4	72.9	51.1	87.9	59.9	68.3
RegMean (Jin et al., 2023)	65.3	63.5	75.6	78.6	78.1	67.4	93.7	52.0	71.8
Task Arithmetic (Ilharco et al., 2023)	55.2	54.9	66.7	78.9	80.2	69.7	97.3	50.4	69.1
Ties-Merging (Yadav et al., 2023)	59.8	58.6	70.7	79.7	86.2	72.1	98.3	54.2	72.4
Task-wise AdaMerging (Ours)	58.0	53.2	68.8	85.7	81.1	84.4	92.4	44.8	71.1
Task-wise AdaMerging++ (Ours)	60.8	56.9	73.1	83.4	87.3	82.4	95.7	50.1	73.7
Layer-wise AdaMerging (Ours)	64.5	68.1	79.2	93.8	87.0	91.9	97.5	59.1	80.1
Layer-wise AdaMerging++ (Ours)	66.6	68.3	82.2	94.2	89.6	89.0	98.3	60.6	81.1

Table 2: Multi-task performance when merging ViT-L/14 models on eight tasks.

Method	SUN397	Cars	RESISC45	EuroSAT	SVHN	GTSRB	MNIST	DTD	**Avg Acc**
Pretrained	66.8	77.7	71.0	59.9	58.4	50.5	76.3	55.3	64.5
Individual	82.3	92.4	97.4	100	98.1	99.2	99.7	84.1	94.2
Traditional MTL	80.8	90.6	96.3	96.3	97.6	99.1	99.6	84.4	93.5
Weight Averaging	72.1	81.6	82.6	91.9	78.2	70.7	97.1	62.8	79.6
Fisher Merging (Matena & Raffel, 2022)	69.2	88.6	87.5	93.5	80.6	74.8	93.3	70.0	82.2
RegMean (Jin et al., 2023)	73.3	81.8	86.1	97.0	88.0	84.2	98.5	60.8	83.7
Task Arithmetic (Ilharco et al., 2023)	73.9	82.1	86.6	94.1	87.9	86.7	98.9	65.6	84.5
Ties-Merging (Yadav et al., 2023)	76.5	85.0	89.3	95.7	90.3	83.3	99.0	68.8	86.0
AdaMerging (Ours)	79.0	90.3	90.8	96.2	93.4	98.0	99.0	79.9	90.8
AdaMerging++ (Ours)	79.4	90.3	91.6	97.4	93.4	97.5	99.0	79.2	91.0

We verify that the proposed AdaMerging method significantly outperforms existing model merging methods in performance.

Experiments

◆ Substantially Improved Generalization.

We also compare the performance of AdaMerging and task vector-based model merging methods (Task Arithmetic and Ties-Merging) on two sets of unseen tasks. AdaMerging is significantly better.

Table 3: Generalization results on two unseen tasks when merging ViT-B/32 models on six tasks.

	Seen Tasks							Unseen Tasks		
Method	SUN397	Cars	RESISC45	DTD	SVHN	GTSRB	**Avg Acc**	MNIST	EuroSAT	**Avg Acc**
Task Arithmetic (Ilharco et al., 2023)	63.3	62.4	75.1	57.8	84.6	80.4	70.6	77.2	46.2	61.7
Ties-Merging (Yadav et al., 2023)	67.8	66.2	77.2	56.7	77.1	70.9	69.3	75.9	43.3	59.6
AdaMerging (Ours)	65.2	65.9	88.5	61.1	92.2	91.5	77.4	84.0	56.1	70.0
AdaMerging++ (Ours)	68.2	67.6	86.3	63.6	92.6	89.8	78.0	83.9	53.5	68.7
Method	SUN397	Cars	GTSRB	EuroSAT	DTD	MNIST	**Avg Acc**	RESISC45	SVHN	**Avg Acc**
Task Arithmetic (Ilharco et al., 2023)	64.0	64.0	75.2	87.7	57.0	95.7	73.9	52.3	44.9	51.1
Ties-Merging (Yadav et al., 2023)	68.0	67.1	67.7	78.4	56.5	92.8	71.8	58.7	49.2	53.9
AdaMerging (Ours)	67.1	67.8	94.8	94.4	59.6	98.2	80.3	50.2	60.9	55.5
AdaMerging++ (Ours)	68.9	69.6	91.6	94.3	61.9	98.7	80.8	52.0	64.9	58.5

Experiments

◆ Visual analysis of merging coefficient

Different tasks/layers in AdaMerging learn different merge coefficients.

Table 5: Model merging coefficients $\{\lambda_k\}_{k=1}^{K}$ change with respect to training steps on ViT-B/32.

Method	SUN397	Cars	RESISC45	EuroSAT	SVHN	GTSRB	MNIST	DTD
Task-wise AdaMerging	0.2202	0.1413	0.2826	0.3284	0.2841	0.4003	0.1978	0.1692
Task-wise AdaMerging++	0.3171	0.1698	0.4235	0.5198	0.4386	0.5803	0.2452	0.2885

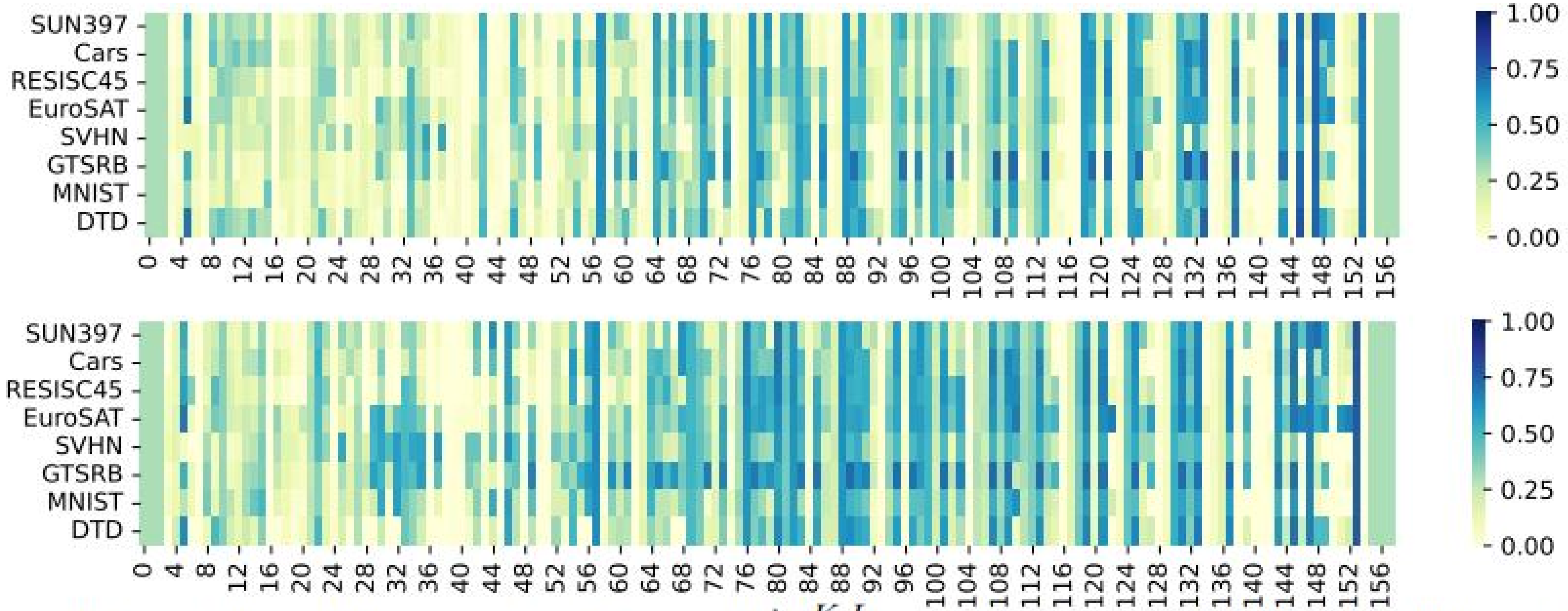


Figure 4: Learned model merging coefficients $\{\lambda_k^l\}_{k=1,l=1}^{K,L}$ of Layer-wise AdaMerging (Above) and AdaMerging++ (Below) on ViT-B/32. The k-th row represents the k-th task vector, the l-th column represents the l-th layer, and the intersection point represents the coefficient λ_k^l.

Thank you!