THE UNIVERSITY of ADELAIDE

Australian Institute for Machine Learning

Funded by the Centre for Augmented Reasoning

NEURAL INFORMATION PROCESSING SYSTEMS

Vancouver, Dec. 2024

Knowledge Composition using Task Vectors with Learned Anisotropic Scaling

Frederic Z. Zhang* Paul Albert* Cristian Rodriguez-Opazo
Anton van den Hengel Ehsan Abbasnejad

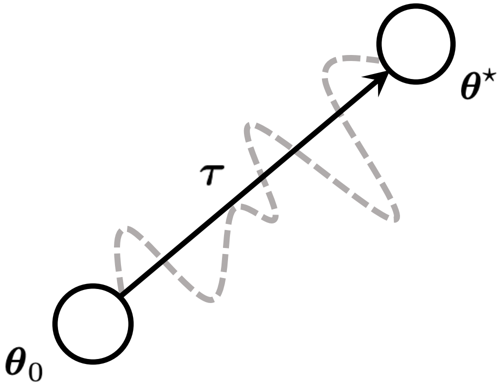
{firstname.lastname@adelaide.edu.au}

Australian Institute for Machine Learning University of Adelaide

December 2024

Task vectors[1]

- Defined as the difference in network weights after fine-tuning
- Characterises the direction and stride of fine-tuning

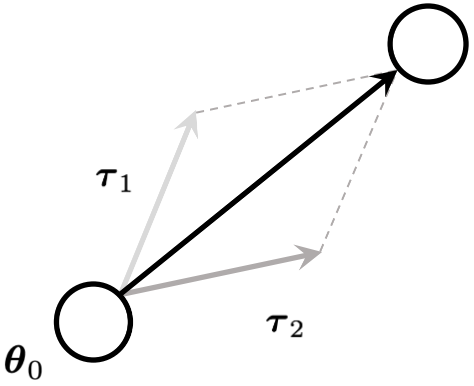


$$\boldsymbol{\theta}^\star = \arg\min_{\boldsymbol{\theta}} \mathcal{L}\left(f\left(\mathbf{x};\boldsymbol{\theta}\right),\mathbf{y}\right)$$

[1] Editing models with task arithmetic, Ilharco et al., ICLR’23

Task arithmetic

- Properties of task vectors that enable model editing via
  - Addition – model merging
  - Negation – remove model bias



Task arithmetic (Cont.)

- Implications
  - Task vectors can serve as knowledge carriers
  - Learning problems may be simplified to learning a combination of task vectors

Proposed method – aTLAS

Task vectors with learned anisotropic scaling

- Task vectors represented as a collection of *m* parameter blocks, with each block represented by a column vector.

$$\boldsymbol{\tau} = \left(\boldsymbol{\tau}^{(1)}, \ldots, \boldsymbol{\tau}^{(m)}\right)$$

Proposed method – aTLAS

NEURAL INFORMATION PROCESSING SYSTEMS
Vancouver, Dec. 2024

Task vectors with learned anisotropic scaling

- Task vectors represented as a collection of *m* parameter blocks, with each block represented by a column vector.
- Anisotropic scaling as a block-diagonal matrix, with each scaling coefficient $\lambda^{(j)} \in \mathbb{R}$ being a learnable parameter.

$$\Lambda = \begin{bmatrix} \lambda^{(1)} I^{(1)} & \dots & \mathbf{0} \\ \vdots & \ddots & \vdots \\ \mathbf{0} & \dots & \lambda^{(m)} I^{(m)} \end{bmatrix}$$

$$\Lambda_i \boldsymbol{\tau}_i = \left(\lambda_i^{(1)} \boldsymbol{\tau}_i^{(1)}, \dots, \lambda_i^{(m)} \boldsymbol{\tau}_i^{(m)} \right)$$

Proposed method – aTLAS

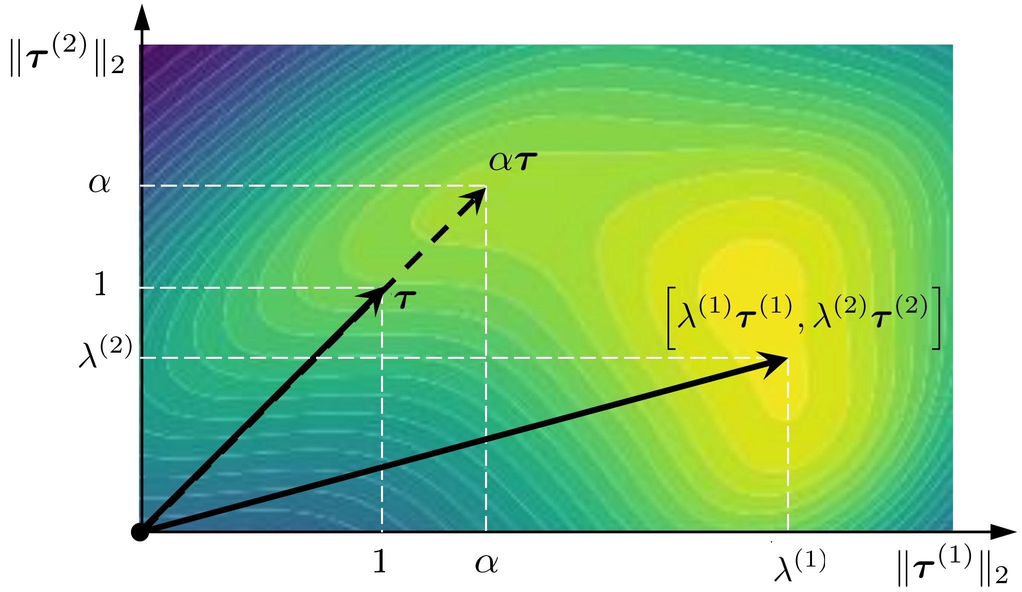
Task vectors with learned anisotropic scaling

- Task vectors represented as a collection of *m* parameter blocks, with each block represented by a column vector.
- Anisotropic scaling as a block-diagonal matrix, with each scaling coefficient $\lambda^{(j)} \in \mathbb{R}$ being a learnable parameter.
- Optimal composition of task vectors

$$\underset{\Lambda_1,\dots,\Lambda_n}{\arg\min}\ \mathbf{E}_{(\mathbf{x},\mathbf{y})\in\mathcal{D}_t}\big[\mathcal{L}\big(f(\mathbf{x};\boldsymbol{\theta}_0+\textstyle\sum_{i=1}^{n}\Lambda_i\boldsymbol{\tau}_i),\mathbf{y}\big)\big]$$

Intuitions

- Isotropic scaling vs. anisotropic scaling



Application 1: Improved task arithmetic

Task arithmetic performance

Task negation

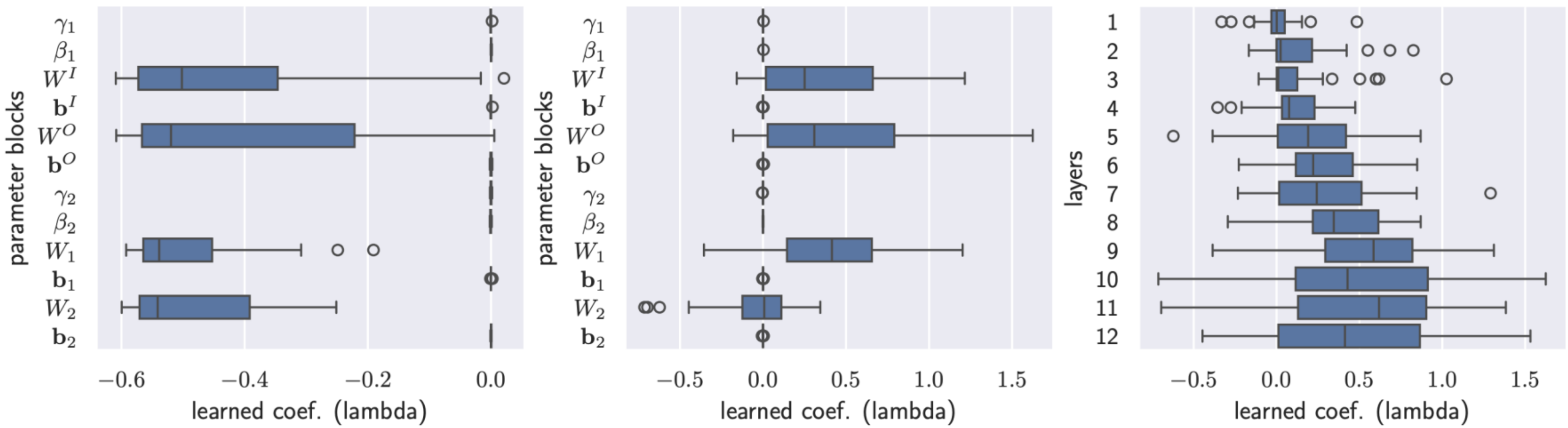
Methods	Models	ViT-B/32		ViT-B/16		ViT-L/14	
		Target (↓)	Control (↑)	Target (↓)	Control (↑)	Target (↓)	Control (↑)
Pre-trained	$f(\mathbf{x};\theta_0)$	48.14	63.35	55.48	68.33	64.89	75.54
Search	$f(\mathbf{x};\theta_0+\alpha\boldsymbol{\tau})$	23.22	60.71	19.38	64.66	19.15	72.05
aTLAS (ours)	$f(\mathbf{x};\theta_0+\Lambda\boldsymbol{\tau})$	**18.76**	**61.21**	**17.34**	**65.84**	**17.75**	**73.28**

Task addition

Methods	Models	ViT-B/32		ViT-B/16		ViT-L/14	
		Abs. (↑)	Rel. (↑)	Abs. (↑)	Rel. (↑)	Abs. (↑)	Rel. (↑)
Pre-trained	$f(\mathbf{x};\theta_0)$	48.14	-	55.48	-	64.89	-
Search	$f\left(\mathbf{x};\theta_0+\alpha\sum_i\boldsymbol{\tau}_i\right)$	70.12	77.24	73.63	79.85	82.93	87.92
aTLAS (ours)	$f\left(\mathbf{x};\theta_0+\sum_i\Lambda_i\boldsymbol{\tau}_i\right)$	**84.98**	**93.79**	**86.08**	**93.44**	**91.36**	**97.07**

Observations

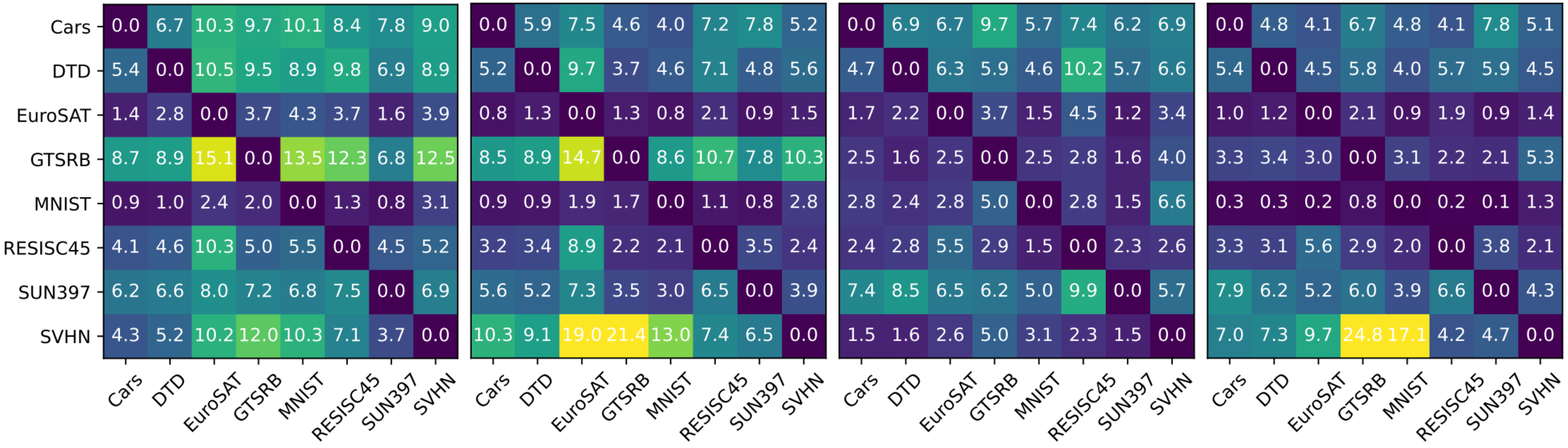
- Learned coefficients concentrate on weight matrices, and on deeper layers.



(a) Coef. of param. blocks (negation) (b) Coef. of param. blocks (addition) (c) Coef. by layer/depth (addition)

Observations (Cont.)

- Learned coefficients concentrate on weight matrices, and on deeper layers.
- Anisotropic scaling can achieve lower disentanglement error, resulting in less conflict between different models during composition.



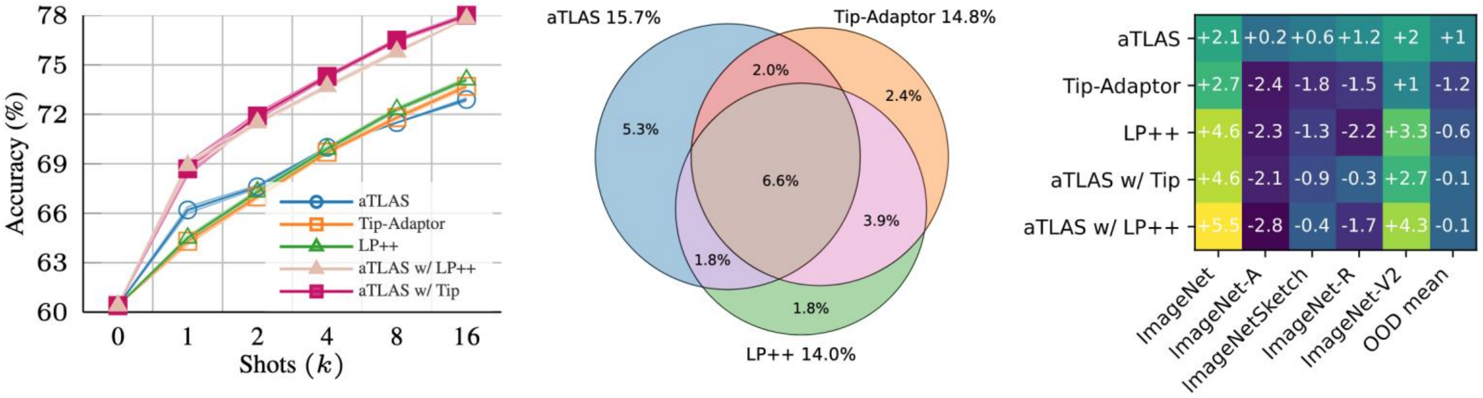
(a) Std. [1], $\bar{\xi} = 6.67\%$ (b) Lin. [2], $\bar{\xi} = 5.69\%$ (c) Std. (ours), $\bar{\xi} = 4.28\%$ (d) Lin. (ours), $\bar{\xi} = 4.39\%$

[2] Task Arithmetic in the Tangent Space: Improved Editing of Pre-Trained Models, Ortiz-Jimenez et al., NeurIPS'23

Application 2:
Knowledge transfer in low-data regimes

Few-shot adaptation

- Complementarity with existing few-shot methods
- Robustness against domain shift



(a) Few-shot recognition performance (b) Images (%) that become correctly classified (c) Improvements on OOD datasets

[3] Tip-Adapter: Training-free CLIP-Adapter for Better Vision-Language Modeling, Zhang et al., ECCV'22
[4] LP++: A Surprisingly Strong Linear Probe for Few-Shot CLIP, Huang et al., CVPR'24

Test-time adaptation

Adapting a model without labelled data, using

- Entropy minimisation
- Contrastive objective
- Pseudo labelling

Method	Zero-shot	Contrastive (SimCLR)		Entropy (SAR)		Pseudo labelling (UFM)	
		LN	aTLAS	LN	aTLAS	LN	aTLAS
Accuracy	60.4	60.4 ± 0.0	62.7 ± 0.1	61.2 ± 0.1	62.9 ± 0.0	62.2 ± 0.1	$\mathbf{66.9} \pm 0.1$

[5] A Simple Framework for Contrastive Learning of Visual Representations, Chen et al., ICML'20
[6] Towards Stable Test-time Adaptation in Dynamic Wild World, Niu et al., ICLR'23

Application 3:
Parameter-efficient fine-tuning (PEFT)

LoRAs [7] as task vectors

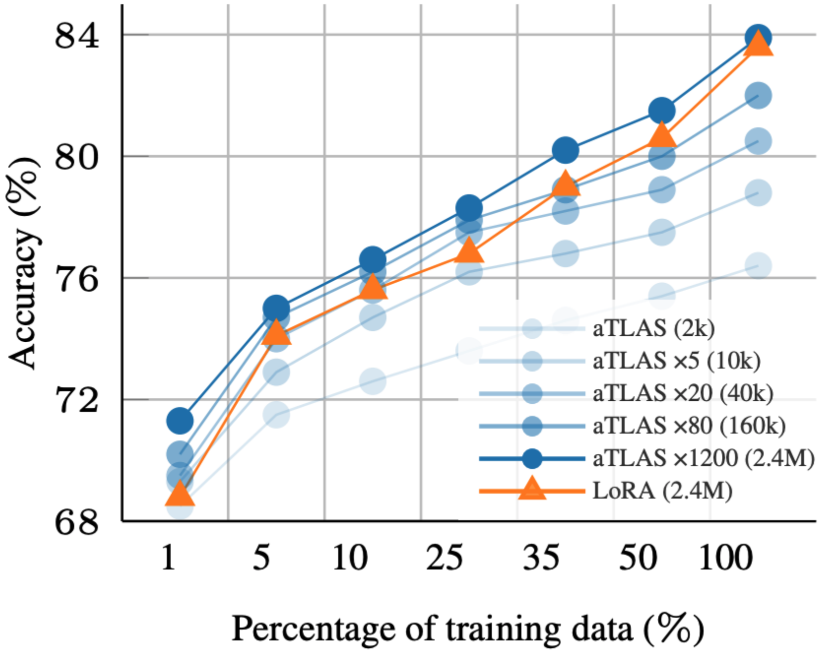
Low-rank adaptations (LoRAs) are sparse task vectors

Shots (k)	Standard task vectors		LoRAs as task vectors		
	All parameter blocks (10.7 GB)	Weight matrices (10.5 GB)	Rank=4 (3.3 GB)	Rank=16 (3.4 GB)	Rank=64 (4.1 GB)
1	66.0 ± 0.2	66.0 ± 0.1	64.4 ± 0.1	64.6 ± 0.1	65.4 ± 0.1
2	67.7 ± 0.1	67.0 ± 0.2	65.7 ± 0.0	66.6 ± 0.2	67.4 ± 0.1
4	70.0 ± 0.0	69.4 ± 0.2	68.2 ± 0.0	68.7 ± 0.1	69.5 ± 0.2
8	71.3 ± 0.1	70.9 ± 0.0	70.2 ± 0.2	70.4 ± 0.1	70.9 ± 0.1
16	72.8 ± 0.1	72.3 ± 0.0	71.7 ± 0.1	71.8 ± 0.1	72.0 ± 0.1

[7] LoRA: Low-Rank Adaptation of Large Language Models,
Hu et al., ICLR’22

Scaling up aTLAS

Higer performance across different percentage of data



Conclusion

- We introduced an algorithm (aTLAS) for task vector composition
- Learned anisotropic scaling results in lower disentanglement error
- Learned coefficients concentrate on weight matrices, and on deeper layers
- aTLAS is complementary to existing few-shot methods
- aTLAS is robust to domain shift
- LoRAs can be integrated into aTLAS for memory efficiency
- aTLAS can be efficiently scaled up for higher performance