Update Your Transformer to the Latest Release: Re-Basin of Task Vectors

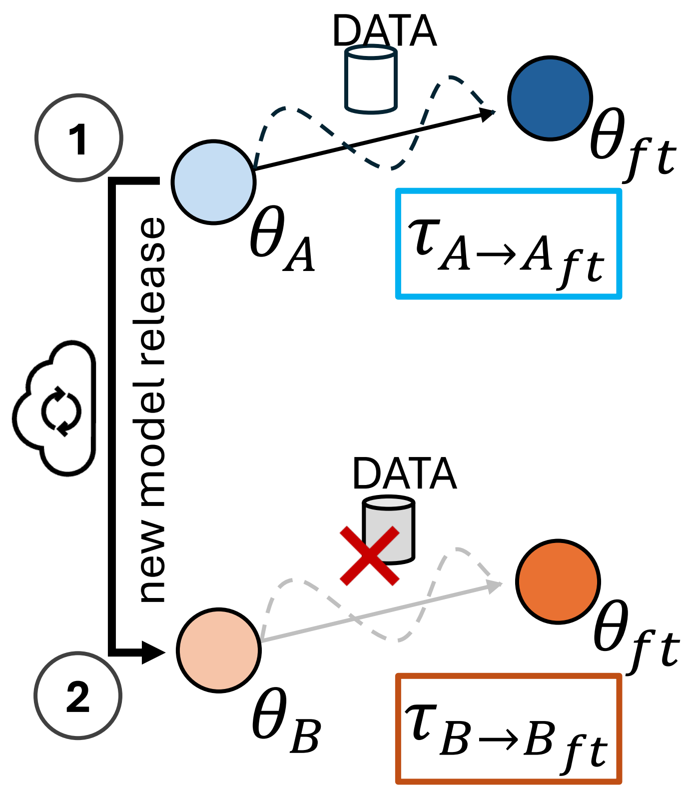
AImage Lab, University of Modena and Reggio Emilia — UNIMORE Università degli Studi di Modena e Reggio Emilia — Sapienza Università di Roma — ICML International Conference On Machine Learning

Filippo Rinaldi*, Giacomo Capitani*, Lorenzo Bonicelli, Donato Crisostomi, Federico Bolelli
Elisa Ficarra, Emanuele Rodolà, Simone Calderara and Angelo Porrello

MOTIVATION

1. Pre-trained CLIP (OpenAI) fine-tuned on a downstream task

2. New pre-trained CLIP (Laion, Meta, etc..) → old fine-tuning is obsolete and re-training is needed

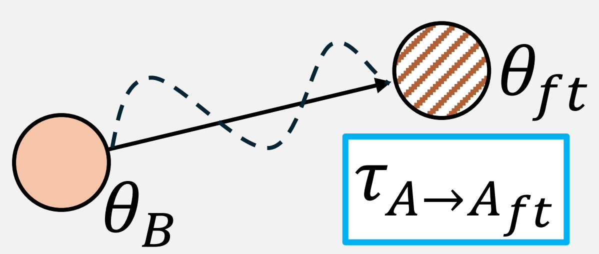


Our Goal

Given a fine-tuned Transformer θ_A^{ft}, we aim to transfer the knowledge of task vector τ_A (computed in 1) into a new release θ_B.

✓ **Data-Free Procedure**

✓ **Applicable to any τ**



BACKGROUND

Model Re-Basin: Exploits permutation symmetry to align different trained models into a shared optimization basin, enabling their interpolation. In our case, we need to align $\boldsymbol{\theta_A}$ with $\boldsymbol{\theta_B}$ to mount $\boldsymbol{\tau_A}$ on $\boldsymbol{\theta_B}$.

Functional Equivalence: NNs exhibit permutation symmetry due to the exchangeability of units within layers. For an MLP layer with activation σ, applying permutation matrix P yields:

$$z_{\ell+1} = \sigma(W_\ell z_\ell + b_\ell) \quad = \quad z_{\ell+1} = P^\top \sigma(P W_\ell z_\ell + P b_\ell)$$

Thus, preserving functional equivalence requires applying consistent permutations across the network:

$$W'_\ell = P W_\ell, \quad b'_\ell = P b_\ell, \quad W'_{\ell+1} = W_{\ell+1} P^\top$$

⚠ **Limitations**: Fail with multi-head attention structure.

RE-BASIN OF TASK VECTORS

TransFusion permutes Transformer weights while preserving attention and residual structures, enabling transfer of task-specific updates across different pretrained model versions.

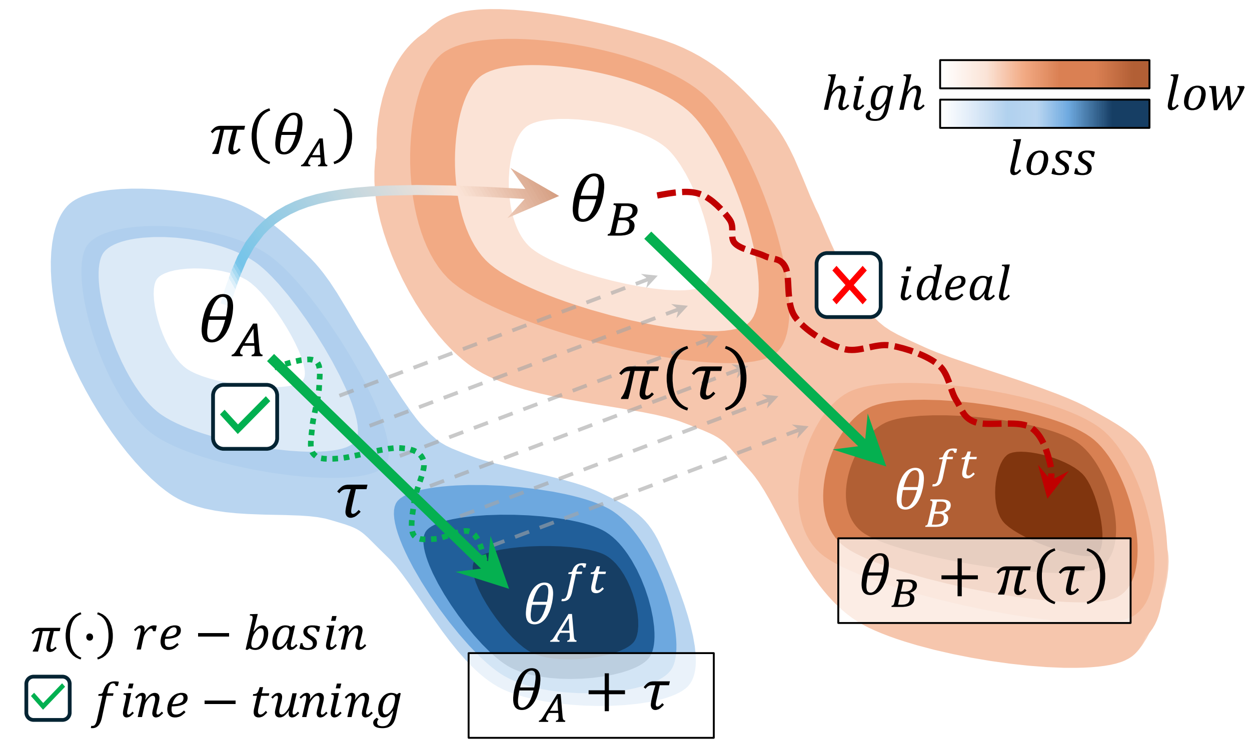
Hybrid Weight Permutation Approach

- **MHA layers:** Custom spectral alignment
- **Other layers:** Git Re-Basin alignment

Task-Vector Transport

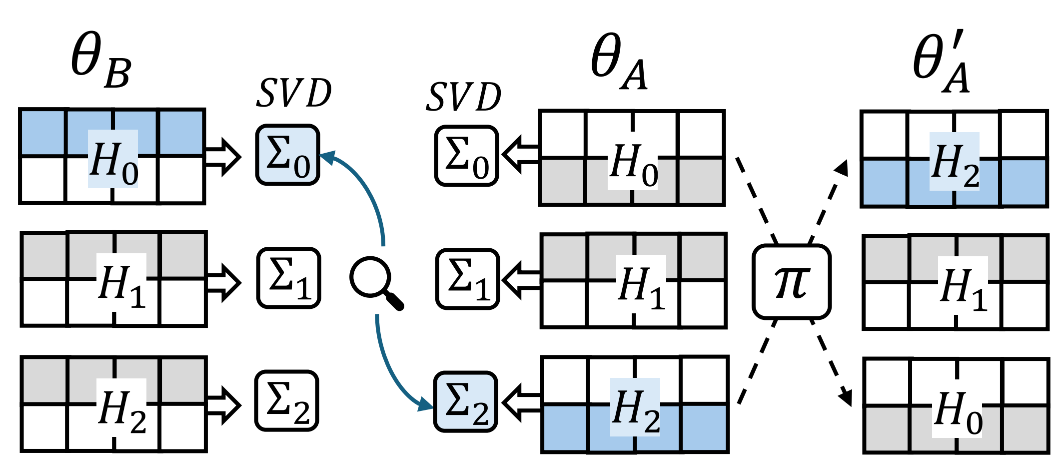
After computing best permutations π, such that $\pi(\theta_A) \approx \theta_B$, we transport the task vector on the new backbone θ_B:

$$\theta_B^{ft} = \theta_B + \pi(\tau_A)$$



WEIGHTS MATCHING

A two-level permutation strategy that first finds optimal mappings between pairs of heads (**Inter-Head matching**), then refines permutations within those matched heads (**Intra-Head matching**).



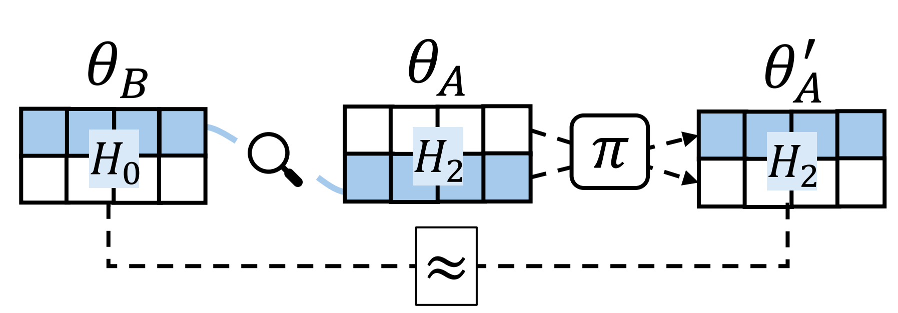
Inter-Head Matching: Find optimal pairing π between attention heads across models using a spectral metric based on singular values.

$$\mathrm{SVD}(W_h) = U_h \Sigma_h V_h^\top$$

$$d\left(h_i^{(A)}, h_j^{(B)}\right) = \left\| \Sigma_i^{(A)} - \Sigma_j^{(B)} \right\|_F$$

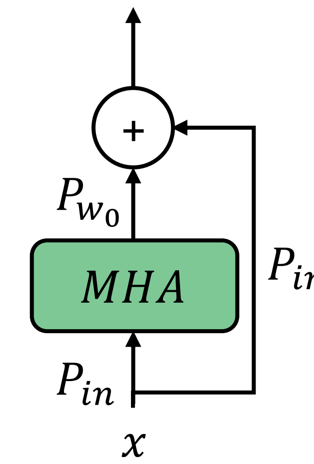
Proposition Let $h \in \mathbb{R}^{m \times n}$ and define $d_p(h_1, h_2) = \|\sigma(h_1) - \sigma(h_2)\|_p$, where $\sigma(h)$ is the vector of singular values. For permutation matrices P_r, P_c, we have:

$$\sigma(P_r h P_c) = \sigma(h) \quad \Rightarrow \quad d_p(h, P_r h P_c) = 0.$$



Intra-Head Matching: Determine permutations π that maximize the inner products across projection weight partitions corresponding to each matched head pair.

FUNCTIONAL EQUIVALENCE



Residual Connection

Ensure consistency by aligning identity paths with attention permutations:

$$\mathbf{z}_i = P_{W_0} \mathbf{z}_{\text{attn}} + \mathcal{I}_i P_{\text{in}} \mathbf{x} = P_{W_0} \mathbf{z}_{\text{attn}} + P_{W_0} \mathbf{x}$$

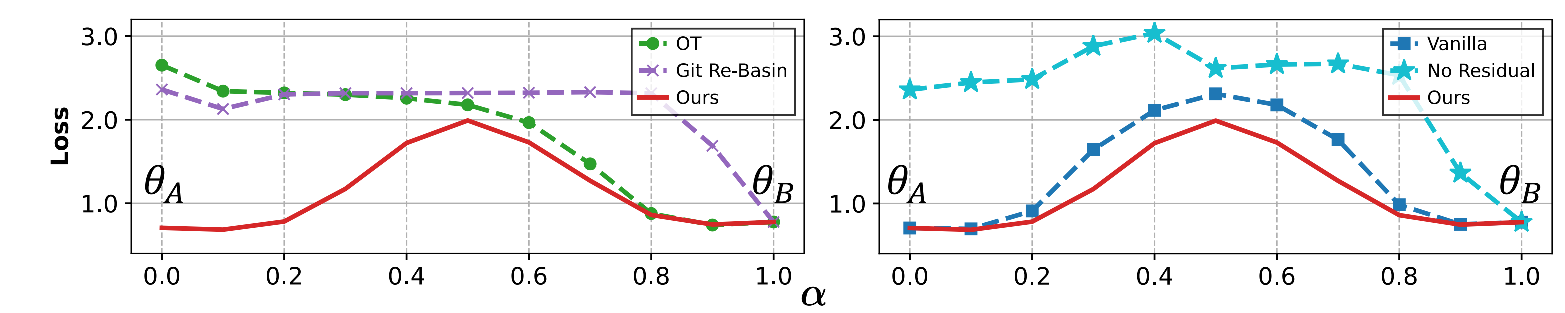
Multi-Head Attention

The output of MHA is **equivariant** to TransFusion permutations:

- Permutation Inter-Head: $P_{\text{inter\_head}} \in \pi$
- Permutation Intra-Head: $P_{\text{intra\_head}} \in \pi, P_{\text{intra\_head}} = \{P^{(i)}\}_{i=1}^H$
- MHA Block Permutation: $P_{\text{attn}} = P_{\text{inter\_head}} \circ P_{\text{intra\_head}} \in \pi$

$$O' = \mathrm{MHA}(X; P_{\text{attn}} W_q, P_{\text{attn}} W_k, P_{\text{attn}} W_v) = O P_{\text{attn}}$$

$$O = O' P_{\text{attn}}^T$$



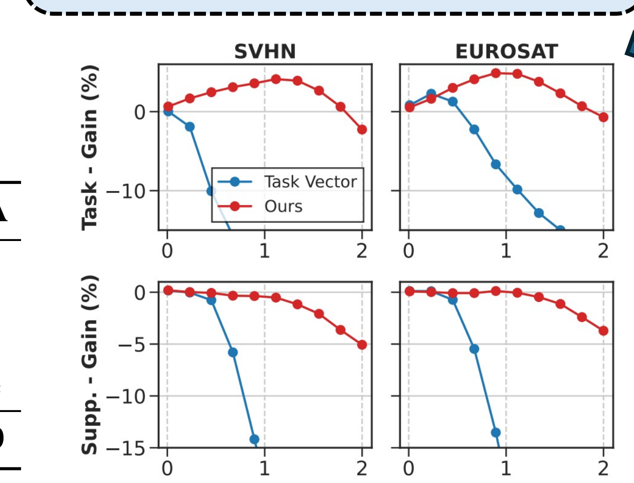
EXPERIMENTS

VISION

Method	EuroSAT Task	EuroSAT Supp.	DTD Task	DTD Supp.	GTSRB Task	GTSRB Supp.	SVHN Task	SVHN Supp.
θ_B *zero-shot*	49.02	68.73	47.50	68.73	43.42	68.73	45.97	68.73
$\theta_B + \tau$	-7.62	-16.15	-0.15	-0.10	-5.39	-0.70	-22.00	-16.45
$\theta_B + \pi(\tau)$ *(Optimal Transport)*	-14.05	-5.28	-0.53	-1.18	-2.43	-1.30	-12.30	-2.70
$\theta_B + \pi(\tau)$ *(GiT Re-Basin)*	+0.95	-0.48	-0.91	**-0.02**	+0.76	**-0.05**	+0.79	**+0.30**
TransFusion (Ours)	**+4.95**	**-0.06**	**+0.21**	-0.08	**+1.10**	-0.40	**+3.64**	-0.48

For all experiments we consider **CLIP ViT-B/16**. We use CommonPool pre-training for θ_A and Datacomp for θ_B. Our method boosts θ_B zero-shot performance and preserves generalization in the updated model.

Zero-shot gain/drop relative to θ_B of naive $\theta_B + \tau$ and our strategy $\theta_B + \pi(\tau)$ varying α.



NLP

Method	QQP	SST2	RTE	CoLA
θ_B	55.00	50.69	54.51	40.94
$\theta_B + \tau$	-8.29	+0.23	-2.53	-0.77
$\theta_B + \tau$ *(OT)*	-8.31	+5.39	-1.08	-1.25
$\theta_B + \tau$ *(GiT Re-Basin)*	+3.58	+5.73	+2.17	+1.44
TransFusion (Ours)	**+6.50**	**+5.96**	**+3.61**	**+2.49**