OPTIMAL TRANSPORT LAYER ALIGNMENT FOR CROSS-ARCHITECTURE TASK VECTOR RE-BASIN

Anonymous authors
Paper under double-blind review

ABSTRACT

Task vectors encode task-specific adaptations, but transferring them across architectures is difficult when source and target models differ in width and depth. Existing re-basin methods address width mismatches through coordinate alignment, yet handle depth mismatches using fixed layer correspondences. We formulate task vector re-basin using correspondence weights to combine source representations for alignment and task vector blocks for transport. To estimate these weights, we introduce Optimal Transport Layer Alignment (OTLA), a training-free method that learns adaptive, weighted correspondences between layers of heterogeneous models. Using a small unlabeled calibration set, OTLA combines representational similarity measured by Centered Kernel Alignment with a relative-depth prior in a relaxed optimal transport formulation. The resulting transport plan jointly determines which source layers contribute to each target layer. The weighted formulation supports expansion and compression without modifying the source architecture or fine-tuning the target model. Across eight vision tasks, three Vision Transformer scales, and both model expansion and compression, OTLA achieves the highest average accuracy when integrated into existing re-basin methods.

1 INTRODUCTION

Pre-trained model families offer a range of sizes for different computational budgets, yet task-specific adaptation is typically tied to a single member of the family. Task vectors, defined as the difference between fine-tuned and pre-trained parameters, provide a natural representation of reusable expertise (Ilharco et al., 2023), but they cannot be directly transferred to larger or smaller counterparts, even within the same Transformer family (Dosovitskiy et al., 2021; Raffel et al., 2020). Enabling such transfer would eliminate the need for repeated fine-tuning as deployment requirements evolve, for example when switching to a smaller model to reduce memory use or latency, or to a larger model when additional compute becomes available. Achieving this requires answering two complementary questions: *how* should the source update be represented in the target model, and *which* source layers should contribute to each target layer?

Recent work has made substantial progress on the first question. While permutation-based re-basin[1] methods align models with the same architecture (Ainsworth et al., 2023), Theseus (Rinaldi et al., 2026b) and BiCo (Son et al., 2026) transport a source task vector to a target model with different feature dimensions without additional training. Both estimate coordinate transformations from calibration data, learning layer-wise rotations between the source and target representation spaces via the solution to a Procrustes problem. However, these transformations assume that the source layer corresponding to each target layer is unique and already known. When model *depths* differ, this correspondence is imposed via a depth-based heuristic like Block Extension (Rinaldi et al., 2026b) which interpolates the source Transformer blocks until source and target have the same depth. Transferring from a 12-block ViT-B/16 to a 24-block ViT-L/14 therefore first constructs a 24-block source model and subsequently aligns layers one-to-one (see Fig. 1, left). The choice of *which* source layers contribute to each target layer is therefore fixed by relative depth before representation alignment.

This assumption is restrictive because corresponding representations need not occur at the same relative depth across models. Studies of representation similarity across models show that models with

[1] *Re-basin* refers to aligning models into a common parameter-space basin (Ainsworth et al., 2023).

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

different depths or architectures can organize their internal representations differently (Kornblith et al., 2019; Nguyen et al., 2021; Raghu et al., 2021). A deeper model does not necessarily place corresponding representations at proportionally deeper positions in the network. Consistent with this observation, recent work has defined cross-model layer correspondences directly from internal representations and found that a layer in a shallower network may correspond to multiple layers in a deeper one (Shah & Khosla, 2026). These results motivate replacing depth-based correspondences with assignments inferred from internal representations. For task vector re-basin, however, estimating such correspondences is only part of the problem: when multiple source layers contribute to a target layer, their contributions must be combined for both Procrustes alignment and task vector transport. We therefore assign each target layer *correspondence weights* indicating how strongly it is matched to each source layer. We use these weights to combine source activations for alignment and task vector blocks for transport. Unlike Block Extension, this weighted formulation operates on the source model without extending blocks (Fig. 1, Right, top).

To determine these weights, we propose *Optimal Transport Layer Alignment* (OTLA), a training-free method for estimating layer correspondences across architectures (Fig. 1, Right, bottom). Independently matching each target layer can repeatedly favor the same source layers, introducing similar representations at different target depths. OTLA instead formulates correspondence as a joint optimal transport problem whose cost combines Centered Kernel Alignment (CKA) similarity with a relative-depth prior, accounting for both representational similarity and network position. We adopt a *relaxed* formulation that encourages assignments to be distributed across source and target layers without enforcing equal use of either. The resulting weights are used consistently to combine both the source representations used for alignment and the source task-vector updates transported to each target layer, operating directly on the original architectures. Our contributions are threefold:

- we introduce a weighted formulation of task vector re-basin that uses shared correspondence weights to combine source representations for coordinate alignment and source task vector updates for transport. Unlike Block Extension, this formulation supports both expansion and compression without modifying the source architecture;
- we introduce OTLA, a training-free layer-matching method that combines representational similarity and relative depth in its transport cost and relaxes source and target marginals to regularize layer usage; and
- we integrate OTLA into existing re-basin methods and show that, for both model expansion and compression, explicitly optimizing layer correspondence improves task vector transport without additional training or calibration data.

2 Related Work

Model Re-Basin and Heterogeneous Alignment. Model merging treats task-specific updates as objects that can be composed in parameter space (Ilharco et al., 2023; Yadav et al., 2023; Gargiulo et al., 2025; Marczak et al., 2025). Re-basin methods exploit parameter symmetries to align independently trained models, enabling update transfer, but generally assume compatible architectures (Ainsworth et al., 2023; Rinaldi et al., 2025). Beyond hard permutations, OT Fusion supports unequal model widths through neuron-level optimal transport (Singh & Jaggi, 2020), while Transformer Fusion extends alignment to attention-based models (Imfeld et al., 2024). ZipIt! and PLeaS use feature correspondences for model fusion (Stoica et al., 2024; Nasery et al., 2025).

Recent work addresses width and depth mismatches across architectures. Cui et al. (2026) learn feature- and layer-level optimal transport couplings to fuse heterogeneous LLM weights before post-fusion fine-tuning. GradFix transfers task vectors to models with different pre-training using target-model gradients to select compatible update coordinates (Rinaldi et al., 2026a). Theseus and BiCo transport task vectors across widths and use prescribed source-target layer correspondences for depth mismatch (Rinaldi et al., 2026b; Son et al., 2026). We instead learn soft cross-depth correspondences for task vector transfer. These correspondences replace prescribed depth mappings in existing re-basin methods and jointly weight calibration signals and task vector blocks during transport.

Representation Similarity Across Depth. Tools include CKA (Kornblith et al., 2019), Procrustes metrics (Williams et al., 2021), and OT-based soft matching (Khosla & Williams, 2024; Shah & Khosla, 2026). Studies examine depth organization (Raghu et al., 2021), shared features across

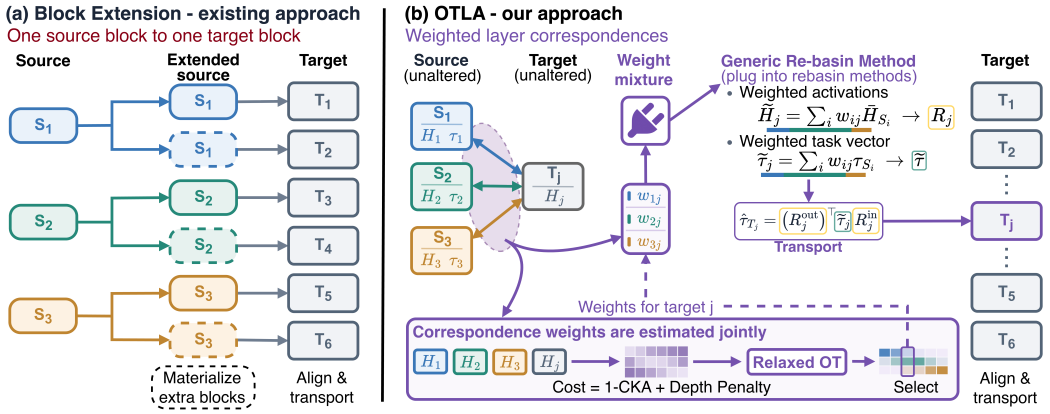


Figure 1: **Weighted Task Vector Re-basin and Optimal Transport Layer Assignment (OTLA) Left**: Block Extension (Rinaldi et al., 2026b) duplicates source transformer blocks to match target depth before pairing each target with one source block. **Right, top**: We propose a weighted formulation for the transport problem. The weight w_{ij} specifies how strongly source layer S_i contributes to target layer T_j. These weights combine source activations for alignment and task vector blocks for transport, without extending the source model. This formulation plugs into re-basin methods. **Right, bottom**: OTLA jointly estimates w_{ij} through relaxed optimal transport using a cost based on CKA similarity between source and target activations and their relative depths.

backbones (Dravid et al., 2023; Huh et al., 2024), and sensitivity to training objectives (Ciernik et al., 2025). Model stitching tests functional compatibility between intermediate representations (Bansal et al., 2021; Csiszárik et al., 2021). Adaptive layer matching supports distillation through OT or attention (Li et al., 2020; Passban et al., 2021). Our relaxed layer transport formulation combines CKA and a depth prior to transfer task updates without student training or neuron-level OT.

3 Background and Motivation

Let θ_S^0 and θ_T^0 be pre-trained source and target ViTs with L_S and L_T layers, and let θ_S^{ft} be the source model θ_S^0 fine-tuned on a downstream task. Its task vector (Ilharco et al., 2023) is:

$$\tau_S = \theta_S^{\text{ft}} - \theta_S^0. \tag{1}$$

Re-basin approaches seek a target update $\hat{\tau}_T$ that transfers without fine-tuning the task-specific information encoded in τ_S into the parameter space of the target model, which may differ in depth and hidden dimension (width). The transported update is scaled by a coefficient α_T, selected on a validation set, yielding the adapted model:

$$\hat{\theta}_T^{\text{ft}} = \theta_T^0 + \alpha_T \hat{\tau}_T. \tag{2}$$

Orthogonal Procrustes for Re-basin. Theseus (Rinaldi et al., 2026b) casts re-basin as a functional matching problem over activations: given a calibration set, it aligns the source and target architectures by solving an orthogonal Procrustes problem between their activations.

Fix a source layer i and a target layer j, and let $\star \in \{\text{in}, \text{out}\}$ indicate the input or output activation of the layer. Given a calibration set of N samples, let $H_{S,i}^\star \in \mathbb{R}^{N \times \ell_S \times d_S^\star}$ and $H_{T,j}^\star \in \mathbb{R}^{N \times \ell_T \times d_T^\star}$ be the corresponding source and target activations, where ℓ_S and ℓ_T are their token sequence lengths and $d_S^\star$ and $d_T^\star$ their feature dimensions. When $\ell_S \neq \ell_T$, Theseus first aligns the token dimension by reshaping patch tokens onto their spatial grids and applying bilinear interpolation. The sample and token dimensions are then flattened, yielding $H_{S,i}^\star \in \mathbb{R}^{m \times d_S^\star}$ and $H_{T,j}^\star \in \mathbb{R}^{m \times d_T^\star}$ with a common number of observations $m = N \times \max(\ell_T, \ell_S)$.

Let $\bar{H}$ denote the centered version of H. The alignment map from i to j solves

$$R_{i \to j}^\star = \underset{R\,:\,RR^\top = I}{\arg\min} \left\| \bar{H}_{S,i}^\star R - \bar{H}_{T,j}^\star \right\|_F^2. \tag{3}$$

Writing the SVD of the cross-covariance matrix as $(\bar{H}^{\star}_{S,i})^{\top}\bar{H}^{\star}_{T,j} = U^{\star}\Sigma^{\star}(V^{\star})^{\top}$, The two maps $R^{\text{in}}_{i\to j}$ and $R^{\text{out}}_{i\to j}$ then transport the slice $\tau_{S,i}$ of the source task vector onto target layer j:

$$\hat{\tau}_{T,j} = \left(R^{\text{out}}_{i\to j}\right)^{\top}\tau_{S,i}\,R^{\text{in}}_{i\to j}. \tag{4}$$

BiCo (Son et al., 2026) adopts the same formulation but estimates the output map $R^{\text{out}}_{i\to j}$ from gradients with respect to the layer outputs rather than from the outputs themselves; the two methods thus share an alignment framework and differ only in this choice of representation.

Handling Depth Mismatch in Model Re-basin. Before computing activations and solving the Procrustes problems in Eq. (3), a correspondence between source layer i and target layer j must be established when the models differ in depth (*e.g.*, from ViT-B/16 to ViT-L/14). Theseus (Rinaldi et al., 2026b) uses a *Block Extension* heuristic, duplicating source blocks via nearest-neighbor interpolation until source and target have the same depth, thereby enabling a one-to-one layer correspondence for solving Eq. (3). A similar strategy was also adopted by BiCo (Son et al., 2026).

However, this strategy has several limitations. It supports only shallow-to-deep transfer, not the opposite direction. It also alters source model activations by artificially duplicating blocks, yielding suboptimal activations for the Procrustes problems. Moreover, it assumes one-to-one layer correspondences that may not hold across architectures. These limitations highlight a broader issue: re-basin across architecture with different depth requires deciding *which* source layers should contribute to each target layer, yet existing approaches treat this as an architectural heuristic.

4 Optimal Transport Layer Alignment for Re-basin Across Different Depths

We introduce *Optimal Transport Layer Alignment* (OTLA), which, instead of assigning each target layer to a single source layer after duplicating source blocks, represents its correspondences as a weighted combination of source layers.

4.1 Weighted Layer Correspondences for Model Re-basin

Rather than assigning each target layer to a single source layer, we represent its correspondence through normalized weights over source layers sharing the same computational role.

Role-Constrained Weighted Layer Correspondences. Transformer blocks contain repeated *submodules* with distinct *computational roles*, such as query, key, value, attention output and MLP projections. We restrict correspondences to submodules with the same role, whose activations are expected to be more comparable for Eq. (3). Let r denote one such role, and let $\mathcal{S}_r \subseteq \{1,\dots,L_S\}$ and $\mathcal{T}_r \subseteq \{1,\dots,L_T\}$ denote the ordered sets of source and target layers associated with it.

For each target layer $j \in \mathcal{T}_r$, we assign normalized weights to source layers:

$$\mathcal{M}(j) = \{w_{ij}\}_{i\in\mathcal{S}_r}, \qquad w_{ij} \geq 0, \qquad \sum_{i\in\mathcal{S}_r} w_{ij} = 1. \tag{5}$$

Thus, $\mathcal{M}(j)$ defines a soft correspondence from multiple source layers performing the same role to target layer j. A hard correspondence is recovered when one weight equals one and the rest are zero. In Section 4.2 we describe how these weights are determined.

Model Re-basin via Weighted Layer Correspondences. For a target layer j, let $\mathcal{M} = \mathcal{M}(j)$ denote its matching weight set. We construct the weighted source activations:

$$\widetilde{H}^{\text{in}}_{\mathcal{M}\to j} = \sum_{i\in\mathcal{S}_r} w_{ij}\bar{H}^{\text{in}}_{S,i}, \qquad \widetilde{H}^{\text{out}}_{\mathcal{M}\to j} = \sum_{i\in\mathcal{S}_r} w_{ij}\bar{H}^{\text{out}}_{S,i}. \tag{6}$$

These weighted activations replace the single source layer activations normally used to estimate the input and output Procrustes maps in Eq. (3). We denote the resulting input and output maps as $R^{\text{in}}_{\mathcal{M},j}$ and $R^{\text{out}}_{\mathcal{M},j}$, respectively. The same weights w_{ij} are applied to the source task vector blocks:

$$\widetilde{\tau}_{\mathcal{M}\to j} = \sum_{i\in\mathcal{S}_r} w_{ij}\tau_{S,i}, \tag{7}$$

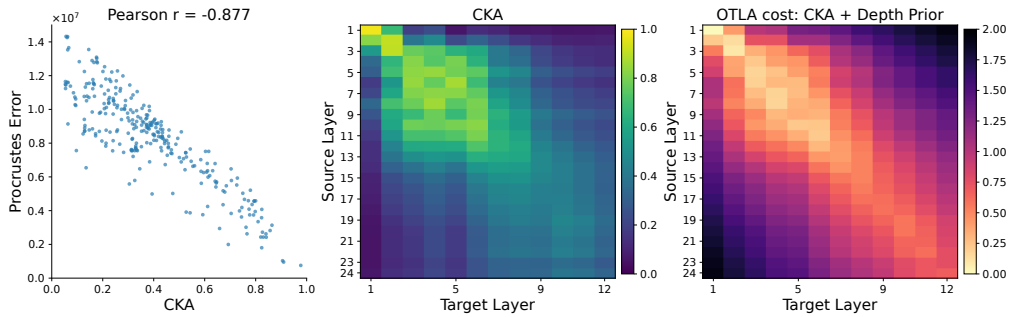


Figure 2: **Motivation for the OT cost.** Left: For source-target query layer pairs on SVHN, CKA similarity is negatively correlated with Procrustes alignment error ($r = -0.877$), supporting its use for matching. Center: For a given target (query) layer, several source layers can have similar CKA scores, leaving their correspondence ambiguous. Right: Adding relative depth distinguishes these candidates in the resulting cost matrix. Analysis on more datasets and layers is provided in Sec. F.6.

The final transported update for target layer j is then

$$\hat{\tau}_{T,j} = \left(R^{\text{out}}_{\mathcal{M},j}\right)^{\top} \widetilde{\tau}_{\mathcal{M}\to j} R^{\text{in}}_{\mathcal{M},j}. \tag{8}$$

OTLA replaces the discrete choice of a single source layer required by existing re-basin methods with a weighted source representation, while leaving the subsequent Procrustes-based transport unchanged. Importantly, OTLA operates directly on the activations and task vector blocks of the original source model, without requiring the duplication or materialization of additional blocks, as in the *Block Extension* approach (Rinaldi et al., 2026b). The remaining problem is to determine the correspondence weights w_{ij}. We formulate their estimation as an optimal transport problem.

4.2 Learning Layer Correspondences with Relaxed Optimal Transport

We aim to favor source layers with representations similar to a target layer, while also accounting for their relative position in the network. We combine these two criteria into a pairwise matching cost, then formulate a *relaxed optimal transport* problem to find the correspondence weights $w_{i,j}$.

Transport Cost. Given a small unlabeled calibration set $\mathcal{D}$, we evaluate the frozen source and target models and collect the output activations of every layer associated with computational role r. For $i \in \mathcal{S}_r$ and $j \in \mathcal{T}_r$, let $H^{\text{out}}_{S,i} \in \mathbb{R}^{m\times d_S}$ and $H^{\text{out}}_{T,j} \in \mathbb{R}^{m\times d_T}$ denote the activation matrices.

The correspondence weights determine which source representations are used to estimate the Procrustes maps (see Eq. (3)). We therefore seek source layers whose representations admit low alignment error. Interestingly, across multiple datasets, we observe that higher Centered Kernel Alignment (CKA) similarity is generally associated with lower Procrustes alignment error (see Fig. 2, left). We thus measure pairwise representational similarity using linear CKA:

$$s_{ij} = \text{CKA}\left(H^{\text{out}}_{S,i}, H^{\text{out}}_{T,j}\right), \qquad i \in \mathcal{S}_r, \quad j \in \mathcal{T}_r. \tag{9}$$

We use $s_{i,j}$ to construct a transport cost matrix; the explicit CKA formula is provided in Appendix B , and its computation for Theseus and BiCo is detailed in Sec. E.1.

As transformer representations evolve across depth, source layers at different depths can have similar CKA score for a target layer (Fig. 2, center). We therefore complement CKA with a *prior on relative depth* favoring pairs at comparable stages of their respective models. For $i \in \mathcal{S}_r$ and $j \in \mathcal{T}_r$, we define their normalized depths as:

$$p_i = \frac{i-1}{|\mathcal{S}_r|-1}, \qquad q_j = \frac{j-1}{|\mathcal{T}_r|-1}. \tag{10}$$

Finally, we combine CKA similarity s_{ij} and relative depth into the pairwise transport cost

$$C_{ij} = 1 - s_{ij} + |p_i - q_j|, \tag{11}$$

where $1 - s_{ij}$ measures representation dissimilarity and $|p_i - q_j|$ penalizes depth displacement. As shown in Fig. 2 (right), the depth term distinguishes layer pairs with similar CKA scores according to their relative positions.

Relaxed Optimal Transport for Joint Assignment. The pairwise cost matrix $C \in \mathbb{R}^{|\mathcal{S}_r|\times|\mathcal{T}_r|}$ identifies compatible layer pairs. However, assigning each target *independently* can repeatedly select the same source layers, whose updates are then transported to different target layers via Procrustes (Eq. (8)). We therefore formulate layer matching as an *optimal transport* (OT) problem (Peyré & Cuturi, 2019), identifying all source–target correspondences together and accounting for how each source layer is used across the target model.

Let $P \in \mathbb{R}_{+}^{|\mathcal{S}_r|\times|\mathcal{T}_r|}$ denote the transport plan, where P_{ij} measures the assignment between source layer i and target layer j. OT minimizes the total matching cost $\langle P, C\rangle$, while fixing the plan row $P\mathbf{1}$ and column $P^\top\mathbf{1}$ sums to prescribed uniform marginals (see Appendix C for details). If these marginals are fixed to be uniform, every source layer must contribute the same total mass and every target layer must receive the same total mass, even when C favors some layers over others.

We instead treat source and target marginals $a = \mathbf{1}/|\mathcal{S}_r|$ and $b = \mathbf{1}/|\mathcal{T}_r|$ as soft constraints and propose to solve a *relaxed* OT problem:

$$P^* = \arg\min_{P\geq 0} \langle P, C\rangle - \varepsilon H(P) + \varepsilon \left[\mathrm{KL}(P\mathbf{1} \,\|\, a) + \mathrm{KL}(P^\top\mathbf{1} \,\|\, b)\right], \tag{12}$$

where $H(P) = -\sum_{ij} P_{ij}(\log P_{ij} - 1)$ is the entropy term, encouraging softer correspondences. The KL terms penalize deviations from uniform source and target usage without enforcing the marginals exactly. This allows the plan to favor layer pairs with lower costs while discouraging excessive concentration. We use the same coefficient ε for entropy and the marginal penalties to avoid an additional tuning parameter, and solve it with log-domain Sinkhorn (Chizat et al., 2018).

From Transport Plan to Correspondence Weights. For each target layer j, we convert the optimal transport plan P^* into normalized correspondence weights:

$$w_{i,j} = \frac{P^*_{i,j}}{\sum_{i'\in\mathcal{S}_r} P^*_{i',j}}, \qquad i \in \mathcal{S}_r, \tag{13}$$

such that $\sum_{i\in\mathcal{S}_r} w_{i,j} = 1$. Since entropy regularization yields dense correspondences, we retain only the most relevant source layers. We sort the weights in descending order and select the smallest number of source layers whose cumulative weight accounts for at least a fraction $p \in (0, 1]$:

$$k_j = \min\left\{k : \sum_{\ell=1}^{k} w_{(\ell),j} \geq p\right\}, \tag{14}$$

where $w_{(1),j} \geq w_{(2),j} \geq \cdots$ denote the sorted correspondence weights for target layer j. We keep the k_j source layers associated with the largest weights, set the remaining weights to zero, and renormalize the selected ones. This allows the number of matched source layers to adapt across target layers and computational roles, without imposing a one-to-one or many-to-one correspondence.

See Appendix A for a detailed description of the OTLA Algorithm.

5 Experimental Results

We evaluate layer matching for task vector transport across both depth and width mismatches, with a particular focus on the former. We study both model expansion and compression, compare OTLA against alternative layer-matching baselines, and ablate its individual components.

5.1 Experimental Setup

Experimental Setting. Following prior work (Rinaldi et al., 2026b; Son et al., 2026), we report per-task and mean accuracy on eight classification tasks (Dataset details in Sec. E.5). We transport task vectors in both directions between OpenCLIP ViT-B/16 and ViT-L/14 pre-trained on DataComp-XL (12 and 24 blocks), and between ViT-B/16 and the 32-block ViT-H/14 models pre-trained on LAION-2B (Cherti et al., 2023). These settings cover both *expansion* and *compression* across changes in model depth and width. We integrate OTLA into Theseus (Rinaldi et al., 2026b) and BiCo (Son et al., 2026), changing only the layer-matching strategy while retaining their Procrustes-based transport procedures. We build on the `merge-and-rebase` codebase (Panariello et al.,

Table 1: **Cross-architecture vision transport.** Average test accuracy (%) across eight tasks. The columns group model expansion (shallow-to-deep) and compression (deep-to-shallow). Best results within each transport method are in **bold** and second-best underlined. These results use $\mathcal{B} = 10$.

		Expansion		Compression	
Transport	**Matching**	B→L	B→H	L→B	H→B
Target zero-shot		73.86	70.97	66.96	62.51
Theseus (Rinaldi et al., 2026b)	Block Extension (Rinaldi et al., 2026b)	75.16	72.99	–	–
	Random	76.01	72.98	67.84	65.01
	Kernel	80.69	76.52	72.97	66.77
	Nearest Depth	80.88	75.31	72.41	65.96
	OTLA	**82.78**	**79.12**	**74.88**	**68.54**
BiCo (Son et al., 2026)	Block Extension (Rinaldi et al., 2026b)	78.00	76.68	–	–
	Random	86.10	86.75	81.30	78.47
	Kernel	87.09	87.33	80.72	79.31
	Nearest Depth	87.39	87.47	81.74	80.42
	OTLA	**88.59**	**88.35**	**83.84**	**82.98**

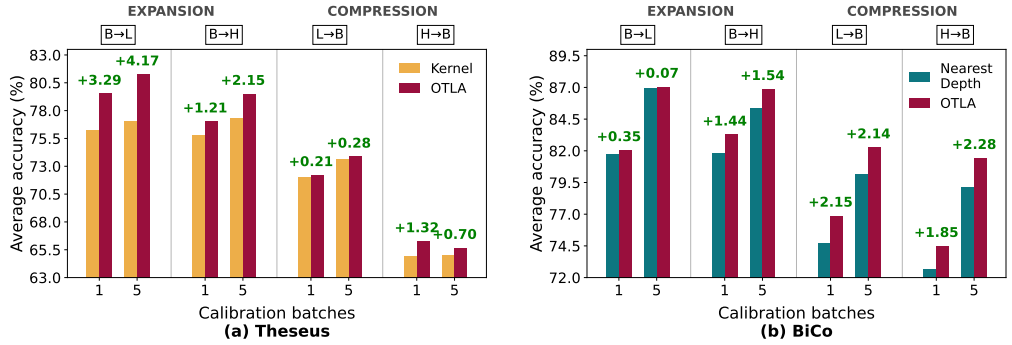


Figure 3: **Scalability across model sizes and transport directions.** Average accuracy on eight vision tasks for re-basin between ViT-B/16 (B), ViT-L/14 (L), and ViT-H/14 (H), using $\mathcal{B} \in \{1, 5\}$ calibration batches. We compare OTLA with the strongest layer-matching baseline for **(a)** Theseus (Kernel) and **(b)** BiCo (Nearest Depth), on both model expansion (B→L, B→H) and compression (L→B, H→B). Numbers above each pair report OTLA accuracy relative to the baseline.

2026), which provides implementations of both methods. Unless stated otherwise, we use $\mathcal{B} = 10$ calibration batches of 32 examples, select α_T from Eq. (2) on a validation set, and perform no target-model optimization after transport. We fix $\varepsilon = 0.005$ and $p = 0.99$ across all experiments. Further dataset, model, and implementation details are provided in Appendix E.

Baselines. We compare OTLA against several layer-matching baselines. *Block Extension* (Rinaldi et al., 2026b) materializes a source network matching the target depth by interpolating blocks before alignment. Since existing re-basin methods lack alternative matching strategies, we additionally design three baselines operating directly on the original source architecture. *Random* selects one source layer uniformly per target layer, *Nearest Depth* selects the source layer minimizing $|p_i - q_j|$ (see Eq. 10), and *Kernel* combines nearby source layers using Gaussian weights over relative depth. Further details are provided in Appendix D.

5.2 Comparison with Block Extension and Other Assignment Strategies

Assignment Strategies vs Block Extension. In Tab. 1, we compare layer-matching strategies for *model expansion* from ViT-B/16 to ViT-L/14 (B→L) and ViT-H/14 (B→H), and *model compression*, from ViT-L/14 to ViT-B/16 (L→B) and from ViT-H/14 to ViT-B/16 (H→B), reporting average test accuracy over eight tasks. *Block Extension* (Rinaldi et al., 2026b) supports network expansion by design, but not compression. Overall, direct layer assignment is more effective than extending the

Table 2: **CKA and relative depth ablation.** Theseus accuracy (%) for ViT-B/16 ↔ ViT-L/14 ($\mathcal{B} = 10$). Best values are in bold, and second-best underlined in each direction and column.

Direction	CKA	Prior	EUROSAT	SVHN	GTSRB	RESISC45	DTD	CARS	MNIST	SUN397	AVG
B→L	✗	✓	85.97	83.47	**77.94**	**76.44**	66.67	92.99	96.67	73.76	81.74
	✓	✗	84.57	**85.51**	75.38	75.40	69.86	**93.17**	**97.74**	**74.32**	81.99
	✓	✓	**89.14**	85.08	77.25	76.07	**71.63**	92.99	96.07	73.98	**82.78**
L→B	✗	✓	68.93	76.11	**63.65**	70.42	**62.29**	89.01	89.54	70.14	73.76
	✓	✗	64.36	77.01	62.73	70.21	58.69	**89.06**	**91.20**	70.13	72.92
	✓	✓	**76.83**	**79.65**	62.93	**71.92**	60.34	88.99	87.97	**70.41**	**74.88**

Table 3: **Independent assignment versus Optimal Transport.** Theseus accuracy (%) on ViT-B/16 → ViT-L/14 ($\mathcal{B} = 10$). Both use the same matching cost; only OTLA jointly optimizes assignments.

Setting	EUROSAT	SVHN	GTSRB	RESISC45	DTD	CARS	MNIST	SUN397	AVG
Cost Only	73.50	74.56	63.92	72.93	66.37	92.98	93.38	73.56	76.40
OTLA	**89.14**	**85.08**	**77.25**	**76.07**	**71.63**	**92.99**	**96.07**	**73.98**	**82.78**

source architecture: *Random Assignment* matches Block Extension in Theseus and improves over it by more than 9 points in BiCo. Adding structure in the assignment strategy helps: the many-to-one Gaussian Kernel assignment substantially improves over the Random one in Theseus, but yields marginal gains in BiCo, where Nearest Depth is the strongest baseline. Across all architecture pairs and both re-basin methods, OTLA performs best by learning adaptive weighted correspondences. Compared with Block Extension, it improves Theseus by 7.62 points on $B \to L$ and 6.13 on $B \to H$, and BiCo by 11.13 points on average across the two expansion settings. Compared to Nearest Depth, OTLA improves by 1.95 points on average in expansion and 2.43 points in compression. Per-task results for all architecture pairs are reported in Appendix G.

OTLA scalability. We vary the batch size $\mathcal{B} \in \{1, 5\}$ to assess OTLA scalability with limited calibration data. Fig. 3 compares it with the nearest competitors: Kernel Assignment in Theseus and Nearest Depth in BiCo; per task results are reported in Appendix G. For Theseus, OTLA consistently improves over Kernel, with the largest gains in expansion. Improvements are smaller under compression, particularly for L→B. With BiCo, OTLA also outperforms Nearest Depth for all directions and calibration budgets, with the largest gains on B→H and the two compression settings.

5.3 Ablations

In this section we ablate the components of OTLA using Theseus with $\mathcal{B} = 10$ calibration batches. Additional ablations and detailed analyses are reported in Appendix F.

CKA and Relative Depth. Tab. 2 ablates the two terms of the transport cost in Eq. (11). Using both yields the highest average accuracy in expansion and compression, with the highest improvement in compression. Among the single-term variants, CKA performs better in expansion, while the relative-depth prior performs better in compression. This suggests that the two signals are complementary, with their relative importance depending on the transfer direction.

Independent Assignment versus Optimal Transport. Tab. 3 compares OTLA with an independent assignment using the same CKA- and depth-based cost matrix in Eq. (11) and cumulative threshold p. In expansion (B→L), OTLA improves on every task, with the largest gains on EuroSAT, SVHN, and GTSRB. These consistent gains support jointly optimizing layer correspondences via optimal transport rather than assigning each target independently. Similar results for compression (L→B) are reported in the Sec. F.3.

Marginal Constraints. Tab. 4 compares standard OT, which fixes both marginals, with our formulation, which penalizes deviations from uniform source and target usage through KL terms. Relaxation improves average accuracy by 1.59 points in expansion and 0.39 points in compression. The compression results are more task-dependent: gains on SVHN and EuroSAT are partly offset by drops on MNIST and GTSRB. Overall, relaxing the marginals is most beneficial for expansion.

Table 4: **Optimal Transport Formulation Ablation.** Theseus test accuracy (%) for ViT-B/16 ↔ ViT-L/14 with $\mathcal{B} = 10$. Average gains are relative to balanced OT in the same direction.

Direction	OT Formulation	EUROSAT	SVHN	GTSRB	RESISC45	DTD	CARS	MNIST	SUN397	AVG
B→L	Standard	84.12	82.81	76.30	**76.19**	66.61	**92.99**	**96.53**	73.94	81.19
	Relaxed	**89.14**	**85.08**	**77.25**	76.07	**71.63**	**92.99**	96.07	**73.98**	**82.78**
L→B	Standard	74.03	74.76	**65.88**	70.97	57.92	88.88	**93.37**	70.13	74.49
	Relaxed	**76.83**	**79.65**	62.93	**71.92**	**60.34**	**88.99**	87.97	**70.41**	**74.88**

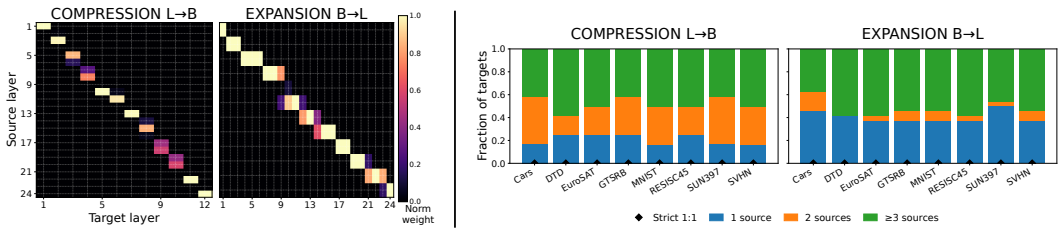


Figure 4: **Transport structure and matching cardinality under expansion and compression.** Left: normalized transport plan (see Eq. 13) for SVHN query projections before cumulative selection, for ViT-L/14 → ViT-B/16 and ViT-B/16 → ViT-L/14. Expansion often reuses a source layer across consecutive targets, while compression combines multiple source layers for the same target. Right: fractions of target query layers selecting one, two, or at least three source layers after cumulative selection ($p = 0.99$) across eight tasks. Black diamonds indicate strict one-to-one matches

5.4 Analysis of the Optimal Transport Plan

We analyze the correspondences learned by OTLA for ViT-B/16 ↔ ViT-L/14 in Fig. 4. The left heatmaps show the normalized correspondence weights obtained by solving the optimal plan in Eq. (13) for SVHN query projections; brighter cells indicate larger weights. Unlike nearest-depth matching, which assigns a single source layer to each target, OTLA learns structured many-to-one and one-to-many correspondences. *During compression* ($L \to B$), weights often extend vertically, indicating that a target combines multiple source layers. *During expansion* ($B \to L$), horizontal runs show that the same source layer is reused across consecutive targets.

The right panels quantify the number of sources selected per target across all eight datasets. Most targets select multiple sources in both directions, with matches involving at least three sources remaining common. Compression generally exhibits fewer single-source and more two-source matches than expansion. These results show that OTLA adapts the matching cardinality to each target rather than enforcing a fixed one-to-one assignment. Additional visualizations are provided in Sec. F.5.

6 Conclusions

In this paper we introduced a weighted formulation for task vector re-basin that replaces fixed layer assignments with adaptive source-target correspondences. The same weights combine source representations for alignment and task vector blocks for transport, supporting both model expansion and compression without architecture modification. OTLA is a training-free method that learns correspondences through relaxed optimal transport using representational similarity and relative depth. Across ViT expansion and compression settings and combined with recent re-basin methods, learned correspondences consistently outperform fixed assignments and Block Extension, highlighting layer correspondence as a key component of task vector re-basin.

Limitations and Future Work. Our evaluation is limited to vision models, leaving other modalities for future work. Compared with heuristic matching strategies, OTLA introduces a slight computational overhead and additional CPU memory requirements to store calibration activations for CKA-based correspondence estimation.

REFERENCES

Samuel Ainsworth, Jonathan Hayase, and Siddhartha Srinivasa. Git re-basin: Merging models modulo permutation symmetries. In *The Eleventh International Conference on Learning Representations*, 2023.

Yamini Bansal, Preetum Nakkiran, and Boaz Barak. Revisiting model stitching to compare neural representations. In *Advances in Neural Information Processing Systems 34 (NeurIPS 2021)*, pp. 225–236, 2021.

Gong Cheng, Junwei Han, and Xiaoqiang Lu. Remote sensing image scene classification: Benchmark and state of the art. *Proceedings of the IEEE*, 105(10):1865–1883, 2017. doi: 10.1109/JPROC.2017.2675998.

Mehdi Cherti, Romain Beaumont, Ross Wightman, Mitchell Wortsman, Gabriel Ilharco, Cade Gordon, Christoph Schuhmann, Ludwig Schmidt, and Jenia Jitsev. Reproducible scaling laws for contrastive language-image learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 2818–2829, 2023.

Lénaïc Chizat, Gabriel Peyré, Bernhard Schmitzer, and François-Xavier Vialard. Scaling algorithms for unbalanced optimal transport problems. *Mathematics of Computation*, 2018.

Laure Ciernik, Lorenz Linhardt, Marco Morik, Jonas Dippel, Simon Kornblith, and Lukas Muttenthaler. Objective drives the consistency of representational similarity across datasets. In *Forty-second International Conference on Machine Learning*, 2025.

Mircea Cimpoi, Subhransu Maji, Iasonas Kokkinos, Sammy Mohamed, and Andrea Vedaldi. Describing textures in the wild. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2014.

Adrián Csiszárik, Péter Kőrösi-Szabó, Ákos Matszangosz, Gergely Papp, and Dániel Varga. Similarity and matching of neural network representations. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan (eds.), *Advances in Neural Information Processing Systems*, volume 34, pp. 5656–5668. Curran Associates, Inc., 2021.

Chenhang Cui, Binyun Yang, Fei Shen, Yuxin Chen, Jingnan Zheng, Xiang Wang, An Zhang, and Tat-Seng Chua. Transport and merge: Cross-architecture merging for large language models. In *Forty-third International Conference on Machine Learning*, 2026.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*, 2021.

Amil Dravid, Yossi Gandelsman, Alexei A. Efros, and Assaf Shocher. Rosetta neurons: Mining the common units in a model zoo. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 1934–1943, 2023. doi: 10.1109/ICCV51070.2023.00185.

Antonio Andrea Gargiulo, Donato Crisostomi, Maria Sofia Bucarelli, Simone Scardapane, Fabrizio Silvestri, and Emanuele Rodolà. Task singular vectors: Reducing task interference in model merging. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 18695–18705, June 2025.

Patrick Helber, Benjamin Bischke, Andreas Dengel, and Damian Borth. Eurosat: A novel dataset and deep learning benchmark for land use and land cover classification. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 12(7):2217–2226, 2019. doi: 10.1109/JSTARS.2019.2918242.

Minyoung Huh, Brian Cheung, Tongzhou Wang, and Phillip Isola. Position: The platonic representation hypothesis. In Ruslan Salakhutdinov, Zico Kolter, Katherine Heller, Adrian Weller, Nuria Oliver, Jonathan Scarlett, and Felix Berkenkamp (eds.), *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 20617–20642. PMLR, 21–27 Jul 2024.

Gabriel Ilharco, Marco Tulio Ribeiro, Mitchell Wortsman, Ludwig Schmidt, Hannaneh Hajishirzi, and Ali Farhadi. Editing models with task arithmetic. In *The Eleventh International Conference on Learning Representations*, 2023.

Moritz Imfeld, Jacopo Graldi, Marco Giordano, Thomas Hofmann, Sotiris Anagnostidis, and Sidak Pal Singh. Transformer fusion with optimal transport. In B. Kim, Y. Yue, S. Chaudhuri, K. Fragkiadaki, M. Khan, and Y. Sun (eds.), *International Conference on Learning Representations*, volume 2024, pp. 29313–29336, 2024.

Meenakshi Khosla and Alex H Williams. Soft matching distance: A metric on neural representations that captures single-neuron tuning. In Marco Fumero, Emanuele Rodolá, Clementine Domine, Francesco Locatello, Karolina Dziugaite, and Caron Mathilde (eds.), *Proceedings of UniReps: the First Workshop on Unifying Representations in Neural Models*, volume 243 of *Proceedings of Machine Learning Research*, pp. 326–341. PMLR, 15 Dec 2024.

Simon Kornblith, Mohammad Norouzi, Honglak Lee, and Geoffrey Hinton. Similarity of neural network representations revisited. In Kamalika Chaudhuri and Ruslan Salakhutdinov (eds.), *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 3519–3529. PMLR, 09–15 Jun 2019.

Jonathan Krause, Michael Stark, Jia Deng, and Li Fei-Fei. 3d object representations for fine-grained categorization. In *2013 IEEE International Conference on Computer Vision Workshops*, pp. 554–561, 2013. doi: 10.1109/ICCVW.2013.77.

Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998. doi: 10.1109/5.726791.

Jianquan Li, Xiaokang Liu, Honghong Zhao, Ruifeng Xu, Min Yang, and Yaohong Jin. BERT-EMD: Many-to-many layer mapping for BERT compression with earth mover's distance. In Bonnie Webber, Trevor Cohn, Yulan He, and Yang Liu (eds.), *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 3009–3018, Online, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-main.242.

Daniel Marczak, Simone Magistri, Sebastian Cygert, Bartłomiej Twardowski, Andrew D. Bagdanov, and Joost van de Weijer. No task left behind: Isotropic model merging with common and task-specific subspaces. In *Forty-second International Conference on Machine Learning*, 2025.

Anshul Nasery, Jonathan Hayase, Pang Wei Koh, and Sewoong Oh. Pleas - merging models with permutations and least squares. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 30493–30502, June 2025.

Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Baolin Wu, Andrew Y Ng, et al. Reading digits in natural images with unsupervised feature learning. In *NIPS workshop on deep learning and unsupervised feature learning*, volume 2011, pp. 4. Granada, 2011.

Thao Nguyen, Maithra Raghu, and Simon Kornblith. Do wide and deep networks learn the same things? uncovering how neural network representations vary with width and depth. In *International Conference on Learning Representations*, 2021.

Aniello Panariello, Filippo Rinaldi, Angelo Porrello, Joost van de Weijer, and Simone Calderara. Merge-and-rebase: A unified framework and evaluation benchmark for fine-tuning, model merging, and rebasin, 2026.

Peyman Passban, Yimeng Wu, Mehdi Rezagholizadeh, and Qun Liu. Alp-kd: Attention-based layer projection for knowledge distillation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pp. 13657–13665, 2021. doi: 10.1609/aaai.v35i15.17610.

Gabriel Peyré and Marco Cuturi. Computational Optimal Transport. *Foundations and Trends in Machine Learning*, 11:355–607, 2019. doi: 10.1561/2200000073.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67, 2020.

Maithra Raghu, Thomas Unterthiner, Simon Kornblith, Chiyuan Zhang, and Alexey Dosovitskiy. Do vision transformers see like convolutional neural networks? In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan (eds.), *Advances in Neural Information Processing Systems*, volume 34, pp. 12116–12128. Curran Associates, Inc., 2021.

Filippo Rinaldi, Giacomo Capitani, Lorenzo Bonicelli, Donato Crisostomi, Federico Bolelli, ELISA FICARRA, Emanuele Rodolà, Simone Calderara, and Angelo Porrello. Update your transformer to the latest release: Re-basin of task vectors. In *Forty-second International Conference on Machine Learning*, 2025.

Filippo Rinaldi, Aniello Panariello, Giacomo Salici, Fengyuan Liu, Marco Ciccone, Angelo Porrello, and Simone Calderara. Gradient-sign masking for task vector transport across pre-trained models. In C. Vondrick, B. Hariharan, C. Raffel, L. Pinto, D. Yang, and A. Faust (eds.), *International Conference on Learning Representations*, volume 2026, pp. 54653–54675, 2026a.

Filippo Rinaldi, Aniello Panariello, Giacomo Salici, Angelo Porrello, and Simone Calderara. Transporting task vectors across different architectures without training. In *Forty-third International Conference on Machine Learning*, 2026b.

Shaan Shah and Meenakshi Khosla. Representational alignment across model layers and brain regions with multi-level optimal transport. In *International Conference on Learning Representations*, volume 2026, pp. 83083–83111, 2026.

Sidak Pal Singh and Martin Jaggi. Model fusion via optimal transport. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 22045–22055. Curran Associates, Inc., 2020.

Jungyong Son, Jinwook Jung, Minhee Park, and Sungyong Baik. Bilinear coordinate alignment for training-free task-vector transfer. *arXiv preprint arXiv:2605.28444*, 2026.

Johannes Stallkamp, Marc Schlipsing, Jan Salmen, and Christian Igel. The german traffic sign recognition benchmark: A multi-class classification competition. In *The 2011 International Joint Conference on Neural Networks*, pp. 1453–1460, 2011. doi: 10.1109/IJCNN.2011.6033395.

George Stoica, Daniel Bolya, Jakob Brandt Bjorner, Pratik Ramesh, Taylor Hearn, and Judy Hoffman. Zipit! merging models from different tasks without training. In *The Twelfth International Conference on Learning Representations*, 2024.

Alex H Williams, Erin Kunz, Simon Kornblith, and Scott Linderman. Generalized shape metrics on neural representations. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan (eds.), *Advances in Neural Information Processing Systems*, volume 34, pp. 4738–4750. Curran Associates, Inc., 2021.

Jianxiong Xiao, Krista A Ehinger, James Hays, Antonio Torralba, and Aude Oliva. Sun database: Exploring a large collection of scene categories. *International Journal of Computer Vision*, 119 (1):3–22, 2016.

Prateek Yadav, Derek Tam, Leshem Choshen, Colin Raffel, and Mohit Bansal. TIES-merging: Resolving interference when merging models. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023.

AI USE STATEMENT

We used generative AI tools solely for limited language editing, including stylistic refinement and grammar correction. We did not use these tools for literature retrieval or discovery, research ideation or execution, drafting sections of the paper, generating synthetic data, or deriving mathematical claims. All AI-assisted edits were reviewed by the authors, who take full responsibility for the paper's final content, claims, and artifacts.

REPRODUCIBILITY STATEMENT

We have taken steps to ensure the reproducibility of our work. The full source code and scripts to reproduce the results in the paper will be published after the review phase is completed. All experiments were performed on publicly available datasets, and details of model architectures, and main training hyperparameters are given in the main paper with additional details included in the Appendix.

APPENDIX

The appendices are organized as follows:

- (A) The OTLA Algorithm
- (B) Linear Centered Kernel Alignment
- (C) Optimal Transport
- (D) Evaluated Depth Matching Strategies
- (E) Implementation and Evaluation Details
  - (E.1) CKA Computation for OTLA with Theseus and BiCo
  - (E.2) Calibration and Matching
  - (E.3) Implementation Details
  - (E.4) Validation-based Updated Scaling
  - (E.5) Dataset Details
- (F) Additional Ablations & Analyses
  - (F.1) Sensitivity Analysis to Regularization and Cumulative Selection
  - (F.2) Role-specific Correspondences
  - (F.3) Independent Assignment versus Optimal Transport
  - (F.4) Local Kernel Alternatives
  - (F.5) Extended Analysis of the Optimal Transport Plan
  - (F.6) Procrustes Residual Analysis
- (G) Calibration Budget Analysis

APPENDIX A THE OTLA ALGORITHM

The OTLA Algorithm groups source and target layers by computational role and collects their activations from a calibration set. It then constructs a pairwise cost matrix between source and target models in Eq. (11) by combining CKA similarity with relative depth, and solves the relaxed optimal transport problem in Eq. (12). The resulting transport plan is converted in correspondence weights through Eq. (13), from which the most relevant source layers are selected according to the cumulative criterion in Eq. (14). The resulting weights are used to build the weighted activations and task vector blocks in Eqs. 6 and 7, which are then used by any Procrustes-based re-basin procedure (*e.g.* Theseus or BiCo, see Eq. 8). See algorithm 1 for pseudocode.

Algorithm 1 Optimal Transport Layer Alignment (OTLA)

Require: Source task vector τ_S, calibration set $\mathcal{D}$, roles $\mathcal{R}$, regularization $\varepsilon > 0$, threshold $p \in (0, 1]$, method $m \in \{\text{THESEUS}, \text{BICO}\}$

Ensure: Transported target task vector $\hat{\tau}_T$

for each computational role $r \in \mathcal{R}$ **do**
  Collect paired input activations H^{in} and output signal Z^{out} on $\mathcal{D}$,
  where $Z^{\text{out}} = H^{\text{out}}$ for THESEUS (out acts), and $Z^{\text{out}} = G^{\text{out}}$ for BICO (out grads)
  For each source $i \in \mathcal{S}_r$ and target $j \in \mathcal{T}_r$ layer sets, compute Eqs. (9) to (11)
  Set $a_i = 1/|\mathcal{S}_r|$, $b_j = 1/|\mathcal{T}_r|$ and solve Eq. (12) to obtain P^*
  for each target layer $j \in \mathcal{T}_r$ **do**
    Normalize the transport column $w_{ij} \leftarrow \frac{P^*_{ij}}{\sum_{i' \in \mathcal{S}_r} P^*_{i'j}}$ ▷ Eq. (13)
    Sort w_{ij} in descending order ▷ Eq. (14)
    Retain and renormalize w_{ij}
    Form weighted source representations and task vector:
    $(\widetilde{H}^{\text{in}}, \widetilde{Z}^{\text{out}}, \widetilde{\tau})_{\mathcal{M}\to j} \leftarrow \sum_{i \in \mathcal{S}_r} w_{ij} (\bar{H}^{\text{in}}_{S,i}, \bar{Z}^{\text{out}}_{S,i}, \tau_{S,i})$ ▷ Eqs. (6) and (7)
    Estimate $R^{\text{in/out}}_{\mathcal{M},j}$ from $(\widetilde{H}^{\text{in/out}}_{\mathcal{M}\to j}, \bar{H}^{\text{in/out}}_{T,j})$ via Procrustes ▷ Eq. (3)
    Transport the update: $\hat{\tau}_{T,j} \leftarrow (R^{\text{out}}_{\mathcal{M},j})^\top \widetilde{\tau}_{\mathcal{M}\to j} R^{\text{in}}_{\mathcal{M},j}$ ▷ Eq. (8)
  end for
end for

APPENDIX B LINEAR CENTERED KERNEL ALIGNMENT

For two row-aligned activation matrices U and V, let $\bar{U}$ and $\bar{V}$ denote their column-centered versions. Linear Centered Kernel Alignment (CKA) (Kornblith et al., 2019) is defined as:

$$\mathrm{CKA}(U, V) = \frac{\|\bar{U}^\top \bar{V}\|_F^2}{\|\bar{U}^\top \bar{U}\|_F \, \|\bar{V}^\top \bar{V}\|_F}, \tag{15}$$

and measures representational similarity even when U and V have different feature dimensions.

APPENDIX C OPTIMAL TRANSPORT

Given a pairwise cost matrix $C \in \mathbb{R}^{|\mathcal{S}_r| \times |\mathcal{T}_r|}$, let $P \in \mathbb{R}+^{|\mathcal{S}_r| \times |\mathcal{T}_r|}$ denote a transport plan, where P_{ij} represents the assignment mass between source layer i and target layer j. Let $a = \frac{\mathbf{1}}{|\mathcal{S}_r|}$ and $b = \frac{\mathbf{1}}{|\mathcal{T}_r|}$ denote uniform source and target marginals. Entropy-regularized optimal transport is defined as

$$P^* = \arg\min_{P \geq 0} \langle P, C \rangle - \varepsilon H(P) \quad \text{s.t.} \quad P\mathbf{1} = a, \qquad P^\top \mathbf{1} = b, \tag{16}$$

where $H(P) = -\sum_{ij} P_{ij}(\log P_{ij} - 1)$ is the entropy term and $\varepsilon > 0$ controls the softness of the transport plan. The marginal constraints prescribe how much total assignment weight each source layer can distribute and each target layer must receive. In the relaxed formulation presented in the main paper (see Eq. (12)), these equalities are replaced by KL penalties, allowing the total assignment weights to deviate from the prescribed marginals a and b when favored by the transport cost.

APPENDIX D EVALUATED DEPTH MATCHING STRATEGIES

We compare layer matching strategies within Theseus (Rinaldi et al., 2026b) and BiCo (Son et al., 2026), separating the choice of layer correspondence from the downstream transport method. Except for materialized Block Extension, the baseline inherited from prior work, the remaining strategies provide controlled comparisons that use the correspondence weights $\mathcal{M}(j) = \{w_{ij}\}_{i \in \mathcal{S}_r}$, for each target layer j, where $\mathcal{S}_r$ contains the source layers with a compatible role.

Block Extension. This baseline explicitly expands the source architecture to the target depth before applying re-basin. Alignment is therefore estimated using the expanded source network. It supports only expansion and does not natively handle compression.

Nearest-depth matching. Each target layer j is assigned to the source layer with the closest normalized depth:

$$i^*(j) = \arg\min_{i \in \mathcal{S}_r} |p_i - q_j|, \qquad w_{ij} = \mathbb{I}[i = i^*(j)]. \tag{17}$$

Random Matching: For each target layer j, independently sample a source layer uniformly from the compatible set, with replacement:

$$I_j \sim \mathrm{Uniform}(\mathcal{S}_r), \qquad w_{ij} = \mathbb{I}[i = I_j]. \tag{18}$$

Thus, a source layer may be assigned to multiple targets.

Kernel: This depth-based soft matching assigns weights w_{ij} using a Gaussian kernel of size 6 centered at the target's relative depth q_j, restricted to a local source-layer neighborhood and normalized to sum to one. Comparisons between Uniform and Gaussian kernel with different kernel sizes are reported in Sec. F.4.

APPENDIX E IMPLEMENTATION AND EVALUATION DETAILS

E.1 CKA COMPUTATION FOR OTLA WITH THESEUS AND BICO

For integrating OTLA into Theseus, we compute linear CKA between the output activations $H_{S,i}^{\mathrm{out}}$ and $H_{T,j}^{\mathrm{out}}$, as in Eq. (9). For BiCo, we instead compute CKA on the corresponding output gradients, following the gradient-based signal used by BiCo. Apart from the signal used to compute s_{ij}, the rest of the OTLA pipeline remains unchanged.

E.2 Calibration and Matching

The main comparisons use $\mathcal{B} = 10$ batches of 32 training examples, sampled uniformly at random to construct the calibration set, following the protocol of Theseus (Rinaldi et al., 2026b). OTLA uses the same calibration set required for Procrustes by both Theseus and BiCo (Son et al., 2026). Source task vectors are computed from the fine-tuned checkpoints used by the Theseus evaluation pipeline. The source and target networks retain their original depth for all matching strategies except for Block Extension. For different patch sizes, we use Theseus spatial bilinear interpolation to align patch-token grids before flattening the sample and token axes. A shared reservoir sample preserves row correspondence across the source and target layers. Without sampling, the four directions (B→L, L→B, B→H, H→B) yield respectively 82,240, 63,040, 41,120, and 63,040 rows, respectively. A row is one token's activation for one image at one layer. For CKA, the reservoir retains at most 40,960 rows in either B/16↔L/14 direction and 16,384 rows in either B/16↔H/14 direction to limit system memory use.

E.3 Implementation Details

We solve the relaxed problem in double precision with log-domain Sinkhorn updates. It checks convergence every ten iterations, using tolerance 10^{-4}, with a cap of 10,000 iterations.

E.4 Validation-based Update Scaling

Following the protocol of Theseus (Rinaldi et al., 2026b) and BiCo (Son et al., 2026), for each task and matching strategy, α is selected by validation accuracy along a grid with step 0.1, starting at zero. All the configurations cap this grid at 20, with early-stopping patience parameters of 10. Test data are used only to evaluate the selected update.

E.5 Dataset Details

We evaluate the transferability of our approach across a heterogeneous collection of vision benchmarks spanning several recognition domains, including satellite imagery, traffic signs, handwritten and natural digits, textures, fine-grained object categories, and scene understanding:

- **EuroSAT (Helber et al., 2019).** A land-use and land-cover recognition benchmark derived from Sentinel-2 satellite imagery. It contains approximately 27,000 images grouped into 10 semantic classes.
- **GTSRB (Stallkamp et al., 2011).** The German Traffic Sign Recognition Benchmark is a traffic-sign classification dataset comprising 51,839 images distributed over 43 sign categories.
- **SVHN (Netzer et al., 2011).** The Street View House Numbers dataset targets digit recognition in natural images. It contains 73,257 images of numerical characters extracted from Google Street View, covering the 10 digit classes from 0 to 9.
- **RESISC45 (Cheng et al., 2017).** A remote-sensing scene recognition benchmark built from satellite and aerial imagery, with 31,500 images belonging to 45 scene categories.
- **DTD (Cimpoi et al., 2014).** The Describable Textures Dataset focuses on visual texture recognition and includes 5,640 images organized into 47 texture classes.
- **Cars (Krause et al., 2013).** A fine-grained visual recognition benchmark designed to distinguish different car types according to attributes such as model, manufacturer, and production year. The dataset contains 16,185 images spanning 196 categories.
- **MNIST (Lecun et al., 1998).** A standard handwritten-digit recognition benchmark containing 70,000 grayscale images of digits from 0 to 9, collected from multiple writers and divided into 10 classes.
- **SUN397 (Xiao et al., 2016).** A large-scale scene recognition benchmark comprising 108,754 images assigned to 397 diverse scene categories.

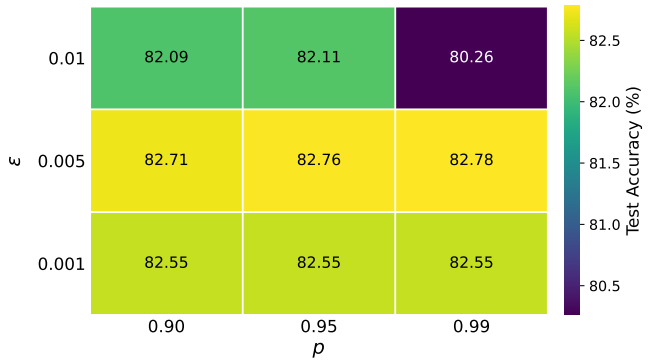


Figure 5: **Sensitivity to ϵ OT regularization and cumulative threshold p.** Mean test accuracy for Theseus with OTLA on ViT-B/16 $\rightarrow$ ViT-L/14 across eight tasks, varying ε and p. Performance remains stable across the considered values, indicating limited sensitivity to these hyperparameters.

Table 5: **Role-specific correspondence ablation.** Test accuracy (%) for Theseus with OTLA on ViT-B/16 $\leftrightarrow$ ViT-L/14 using $\mathcal{B} = 10$, comparing a single correspondence plan shared across transformer parameter roles with independent role-specific transport plans. AVG is the unweighted mean across the eight tasks, and per-task gains are measured relative to the shared-plan variant.

Direction	Grouping strategy	EUROSAT	SVHN	GTSRB	RESISC45	DTD	CARS	MNIST	SUN397	AVG
B→L	Shared plan across roles	88.44	84.83	76.54	75.77	67.38	92.97	96.37	74.17	82.06 (+0.00)
	Independent role (ours)	89.14	85.08	77.25	76.07	71.63	92.99	96.07	73.98	**82.78** (+0.72)
L→B	Shared plan across roles	72.55	79.65	62.17	70.76	59.04	89.17	92.40	70.42	74.52 (+0.00)
	Independent role (ours)	76.83	79.65	62.93	71.92	60.34	88.99	87.97	70.41	**74.88** (+0.36)

APPENDIX F ADDITIONAL ABLATIONS & ANALYSES

F.1 SENSITIVITY ANALYSIS TO REGULARIZATION AND CUMULATIVE SELECTION

We study the interaction between the regularization coefficient ε and the cumulative threshold p for Theseus with OTLA on ViT-B/16 $\rightarrow$ ViT-L/14. Fig. 5 reports mean accuracy across the eight tasks for $\varepsilon \in \{0.001, 0.005, 0.01\}$ and $p \in \{0.9, 0.95, 0.99\}$. Overall, performance remains stable throughout the explored range, indicating that OTLA is robust to these hyperparameters and requires little to no tuning for $\varepsilon \in [0.001, 0.01]$ and $p \in [0.9, 0.99]$. We fix $(\varepsilon, p) = (0.005, 0.99)$ across all our experiments; tuning these hyperparameters separately for each setting could potentially yield further improvements.

F.2 ROLE-SPECIFIC CORRESPONDENCES

Tab. 5 tests whether computational roles benefit from separate correspondence plans (see Section 4.1 for role definitions). Our default formulation solves a separate OT problem for each role. In contrast, the shared variant averages the CKA matrices of query, key, value, attention output, and both MLP projections, and solves a single OT problem whose correspondences are shared across roles.

Role-specific plans improve the mean in both expansion and compression, by 0.72 and 0.36 points, respectively. The gains occur across most tasks, although they are not uniform, with MNIST notably favoring shared correspondences under compression. These results support allowing different computational roles to learn distinct correspondences across depth, while also showing that a shared plan remains competitive.

F.3 INDEPENDENT ASSIGNMENT VERSUS OPTIMAL TRANSPORT

We investigate whether the benefit of jointly optimizing layer correspondences also holds under compression. As shown in Tab. 6, OTLA improves the average accuracy over independent cost-

Table 6: **Independent assignment versus optimal transport.** Test accuracy (%) for Theseus on ViT-L/14 → ViT-B/16 with $\mathcal{B} = 10$, comparing OTLA with an independent cost-based assignment that retains the same matching cost and cumulative sparsification rule but does not jointly couple source and target layers through optimal transport. Reported gains are relative to the cost only assignment.

Setting	EUROSAT	SVHN	GTSRB	RESISC45	DTD	CARS	MNIST	SUN397	AVG
Cost Only	65.23	70.24	64.55	70.18	57.51	88.89	80.89	70.22	70.96 (+0.00)
OTLA	76.83	79.65	62.93	71.92	60.34	88.99	87.97	70.41	**74.88** (+3.92)

Table 7: **OTLA kernel ablation.** Test accuracy (%) for uniform and Gaussian kernels of size 4 and 6. AVG is the unweighted mean across the eight tasks.

Kernel size	Kernel type	EUROSAT	SVHN	GTSRB	RESISC45	DTD	CARS	MNIST	SUN397	AVG
4	Uniform	73.21	80.78	68.80	73.76	66.43	93.02	96.01	73.57	78.20
	Gaussian	**84.36**	**84.84**	68.73	**75.06**	68.20	93.02	**96.88**	73.84	80.62
6	Uniform	75.72	75.97	67.02	73.23	66.43	**93.04**	93.03	73.61	77.26
	Gaussian	82.55	84.70	**70.49**	74.62	**69.27**	**93.04**	96.68	**74.19**	**80.69**

based assignment by 3.92 points. This is consistent with the expansion results in the main paper, supporting joint optimization of layer correspondences across both transport directions.

F.4 Local Kernel Alternatives

Tab. 7 tests whether a fixed local average is sufficient by varying its neighborhood size and weighting. Gaussian weighting improves the mean over uniform weighting at both reported sizes. Increasing the Gaussian neighborhood from four to six layers changes the mean only slightly, and both remain below full OTLA in Tab. 2. These results support adapting the correspondence weights beyond a fixed depth-based kernel; the reported sizes alone do not establish a plateau for larger neighborhoods.

F.5 Extended Analysis of the Optimal Transport Plan

We extend the transport-plan analysis in the main paper to all eight vision tasks, considering query projections for both ViT-B/16 → ViT-L/14 expansion and ViT-L/14 → ViT-B/16 compression.

Fig. 6 shows the learned correspondences in the expansion setting. Across datasets, the transport plans exhibit a consistent depth-ordered structure, with most of the mass concentrated around mappings that preserve the relative progression through the network. Since the target model contains more layers than the source, the same source representation is frequently reused across multiple consecutive target layers, resulting in characteristic horizontal bands. At the same time, their exact extent and weight vary across tasks, showing that the learned correspondence is not a fixed depth-based assignment but adapts to the representations induced by each dataset.

Fig. 7 reports the corresponding analysis under compression. The global depth ordering remains visible across all tasks, but the geometry of the correspondence changes because a larger set of source layers must be mapped to a smaller number of target layers. Consequently, multiple neighboring source layers can contribute to the same target, producing vertically distributed correspondences, particularly in intermediate and deeper portions of the network. As in expansion, the precise allocation varies across datasets while retaining the same overall hierarchical organization.

F.6 Procrustes Residual Analysis

Fig. 2 in the main paper motivates the design of the OTLA cost by examining the relationship between representational similarity and alignment quality on SVHN for ViT-B/16 → ViT-L/14. Fig. 8 extends this analysis to the query projections of EuroSAT and GTSRB, considering both model expansion (ViT-B/16 → ViT-L/14) and compression (ViT-L/14 → ViT-B/16).

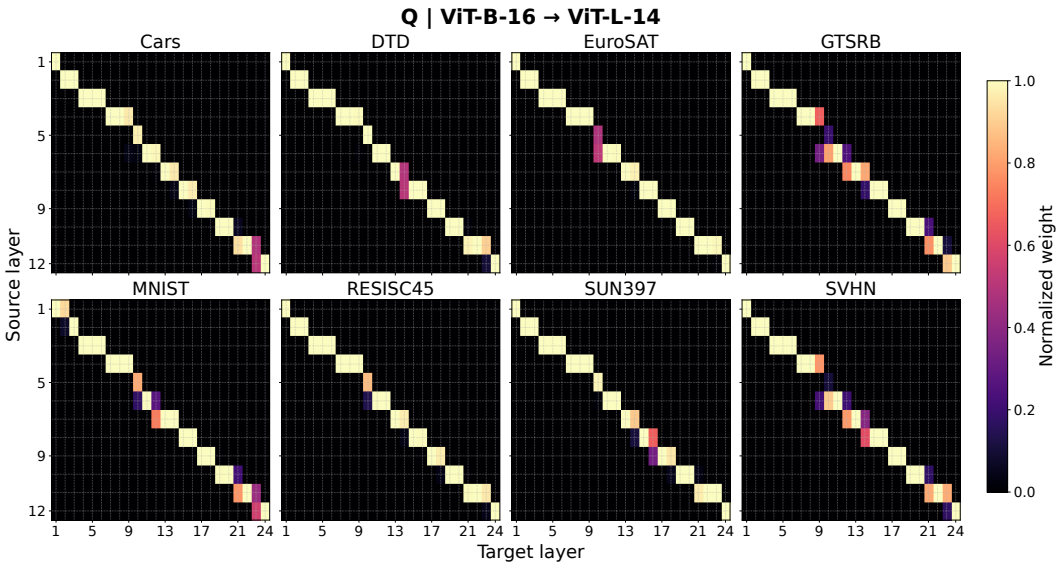


Figure 6: **OTLA correspondences under model expansion.** Correspondence weights for query projections across the eight vision tasks for ViT-B/16 → ViT-L/14. Brighter cells indicate larger weights. The plans preserve a global depth-ordered structure while frequently reusing the same source layer across multiple consecutive target layers.

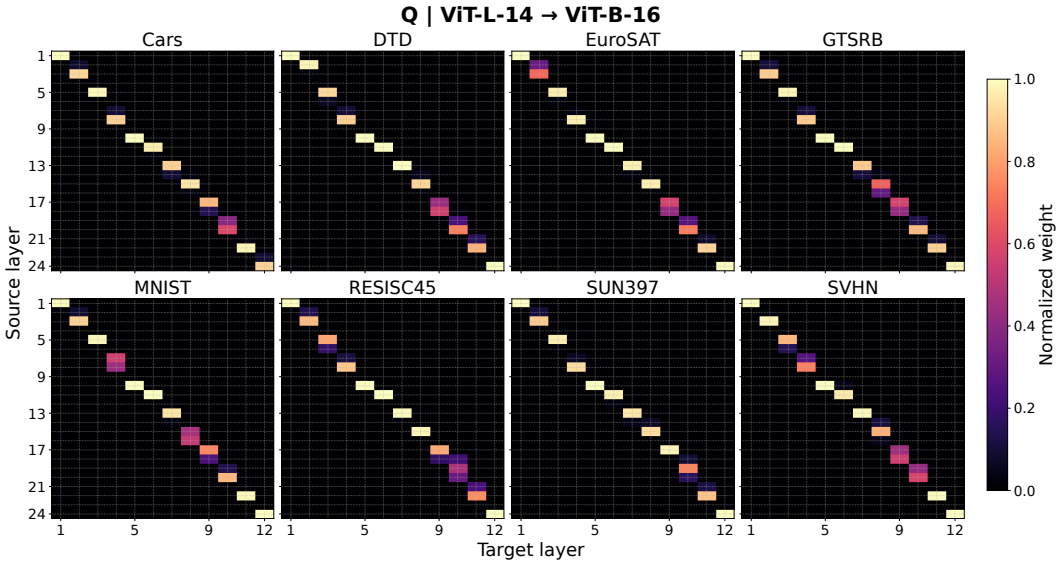


Figure 7: **OTLA correspondences under model compression.** Correspondence weights for query projections across the eight vision tasks for ViT-L/14 → ViT-B/16. Brighter cells indicate larger weights. Multiple neighboring source layers can contribute to the same target layer while the correspondence retains a clear global depth ordering.

Across all settings, CKA exhibits a strong negative correlation with the Procrustes residual, indicating that representations with higher similarity generally require a smaller orthogonal alignment correction. This relationship is consistent across datasets and transport directions, with Pearson correlations of -0.814 and -0.816 for expansion on EuroSAT and GTSRB, respectively, and slightly stronger correlations of -0.847 and -0.850 under compression. At the same time, the CKA matrices contain broad regions of similarly high similarity, making layer correspondences potentially ambiguous when CKA is used alone. Incorporating the relative-depth prior into the OTLA cost in-

troduces additional structure, helping discriminate among otherwise similarly plausible correspondences.

The analysis above focuses on query projections. Fig. 9 further examines all linear roles of the transformer block on SUN397 for ViT-B/16 $\rightarrow$ ViT-L/14. The strength of the relationship between CKA and Procrustes residual varies substantially across roles: query and key exhibit the strongest negative correlations ($r = -0.819$ and $r = -0.810$), while value and MLP expansion retain substantial correlations ($r = -0.647$ and $r = -0.634$). In contrast, attention output and MLP projection show considerably weaker relationships ($r = -0.239$ and $r = -0.427$, respectively). This role-dependent behavior further motivates OTLA's role-wise formulation: different projections exhibit distinct similarity and alignability patterns and can therefore induce different layer correspondences. Transporting an entire transformer block according to a single shared correspondence would discard this role-specific information, as further analyzed in Sec. F.2.

APPENDIX G CALIBRATION BUDGET ANALYSIS

We vary $\mathcal{B}$ for ViT-B/16 $\leftrightarrow$ ViT-L/14 and ViT-B/16 $\leftrightarrow$ ViT-H/14 to determine whether the benefits of learned layer correspondences depend on a large calibration set. Tabs. 8 to 11 retain all task-wise results at each budget, including the $\mathcal{B} = 10$ reference. These experiments measure sensitivity to calibration data, rather than wall-clock or memory scalability.

G.1 EXPANSION

Tabs. 8 and 9 shows that OTLA remains effective with limited calibration data. With Theseus, it outperforms the strongest heuristic at all reported budgets, with larger gains as the budget increases, particularly for B$\rightarrow$L. It also outperforms BiCo. The advantage is smaller at low budget but becomes clearer with additional calibration data, especially for B$\rightarrow$H. Overall, OTLA does not require a large calibration set to improve over fixed matching strategies.

G.2 COMPRESSION

Tabs. 10 and 11 show that OTLA also remains effective under compression with limited calibration data. For L$\rightarrow$B, the gains over the strongest heuristic are modest with Theseus but consistent and larger with BiCo. For the more challenging H$\rightarrow$B setting, OTLA provides clearer improvements, particularly as the calibration budget increases. These results indicate that learned correspondences remain useful even with few calibration batches.

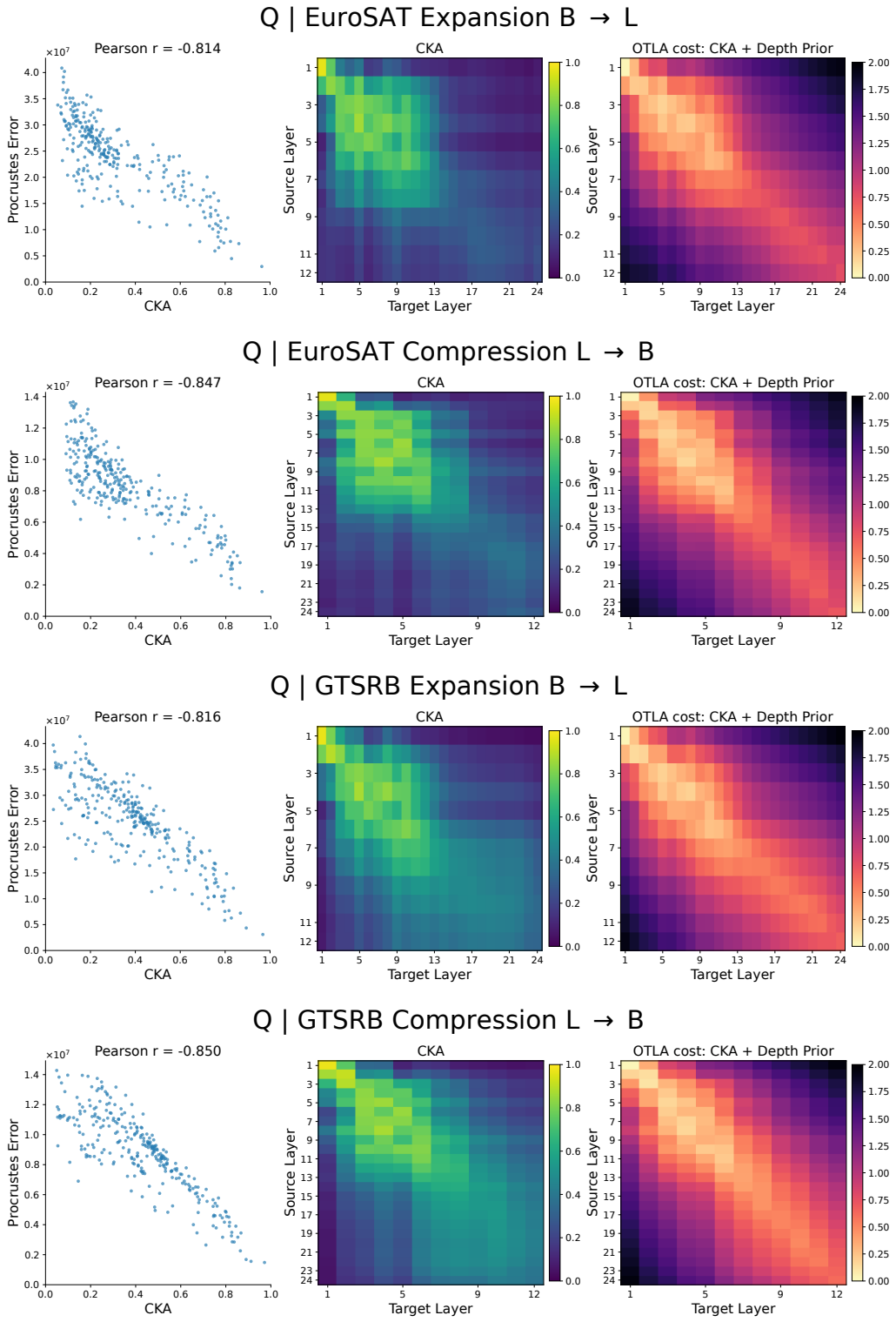


Figure 8: **Procrustes residual analysis across transport directions.** Relationship between CKA similarity and Procrustes residual for query projections on EuroSAT and GTSRB under ViT-B/16 ↔ ViT-L/14 transport. Higher CKA consistently corresponds to lower Procrustes residual, with slightly stronger negative correlations under compression. While CKA alone produces broad regions of comparable similarity, incorporating the relative-depth prior yields a more structured OTLA cost that better differentiates candidate layer correspondences.

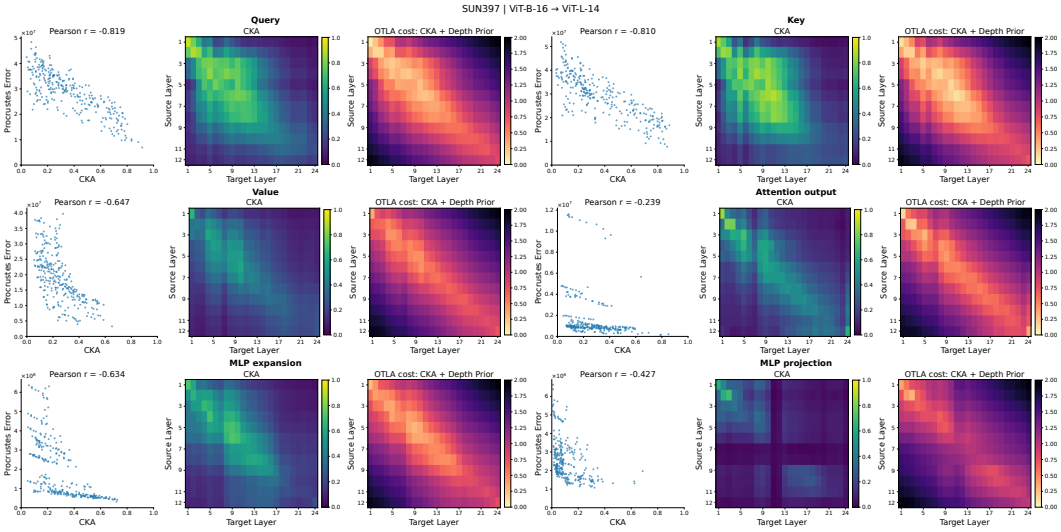


Figure 9: **Role-wise Procrustes residual analysis.** Relationship between CKA similarity, Procrustes residual, and the resulting OTLA cost for the query, key, value, attention-output, MLP-expansion, and MLP-projection roles on SUN397 under ViT-B/16 → ViT-L/14 transport. The strength of the CKA–Procrustes relationship varies substantially across roles, from strong negative correlations for query and key to weaker correlations for attention output and MLP projection. These differences motivate computing correspondences independently for each role rather than enforcing a single block-level layer assignment.

Table 8: **Scalability: B/16 to L/14** Task-vector transport from a ViT-B/16 model to a ViT-L/14 model, both pre-trained on `DataComp-XL`. This setting involves simultaneous depth and width mismatches and employs naive layer interpolation to align transformer depth prior to transport. Results are reported for different alignment budgets $\mathcal{B}$. ΔAcc denotes the average accuracy improvement over the zero-shot baseline of model θ_B.

Model	Layer Matching	$\mathcal{B}$	EUROSAT	SVHN	GTSRB	RESISC45	DTD	CARS	MNIST	SUN397	AVG (ΔAcc)
θ_B *zero-shot*	–	–	72.51	67.73	58.55	72.54	66.37	93.02	86.64	73.55	73.86 (+0.00)
θ_A *fine-tune*	–	–	99.26	97.75	98.76	98.85	83.63	92.62	99.66	76.72	93.41 (+19.55)
θ_B *fine-tune*	–	–	99.38	98.20	99.00	99.22	84.75	94.76	99.79	79.87	94.37 (+20.51)
Theseus	Block Extension	1	73.09	70.68	59.51	72.54	66.37	93.05	90.37	73.50	74.89 (+1.03)
	Random	1	72.63	68.18	59.05	72.54	66.84	93.02	87.94	73.68	74.24 (+0.38)
	Kernel	1	72.80	73.33	65.45	72.70	66.37	93.05	92.50	73.59	76.22 (+2.36)
	Nearest Depth	1	72.22	74.74	72.46	73.17	66.67	93.02	92.53	73.60	77.30 (+3.44)
	OTLA	1	78.89	78.35	75.80	74.37	67.02	93.01	94.76	73.85	**79.51 (+5.64)**
	Block Extension	5	72.59	70.93	63.59	73.81	66.55	93.01	90.96	73.60	75.63 (+1.77)
	Random	5	74.20	68.58	62.30	72.56	66.61	93.01	88.90	73.75	74.99 (+1.13)
	Kernel	5	79.67	73.79	65.65	72.86	66.78	92.99	90.69	74.21	77.08 (+3.22)
	Nearest Depth	5	84.81	74.91	73.27	75.70	70.39	93.01	92.67	73.81	79.82 (+5.96)
	OTLA	5	88.60	81.70	75.91	75.17	66.84	93.04	95.17	73.60	**81.25 (+7.39)**
	Block Extension	10	73.09	69.94	63.68	72.79	66.43	93.02	88.82	73.54	75.16 (+1.30)
	Random	10	78.27	70.12	61.77	72.73	68.50	93.02	89.43	74.23	76.01 (+2.15)
	Kernel	10	82.55	84.70	70.49	74.62	69.27	93.04	96.68	74.19	80.69 (+6.83)
	Nearest Depth	10	87.61	76.46	74.34	74.66	70.98	92.99	96.08	73.92	80.88 (+7.02)
	OTLA	10	89.14	85.08	77.25	76.07	71.63	92.99	96.07	73.98	**82.78 (+8.92)**
BiCo	Block Extension	1	76.67	74.53	73.38	72.58	68.79	93.02	97.03	73.74	78.72 (+4.84)
	Random	1	76.83	69.05	78.16	79.42	68.62	93.22	96.89	73.78	79.49 (+5.63)
	Kernel	1	79.14	75.04	78.16	79.58	71.99	93.09	97.82	74.09	81.11 (+7.25)
	Nearest Depth	1	80.86	78.09	79.19	79.93	70.86	93.13	97.83	73.68	81.70 (+7.84)
	OTLA	1	81.73	77.76	78.68	80.00	72.70	93.08	97.89	74.56	**82.05 (+8.18)**
	Block Extension	5	88.97	72.86	82.90	80.30	74.47	93.08	97.91	73.74	83.03 (+9.17)
	Random	5	90.62	81.28	84.36	83.92	74.82	93.64	98.18	74.39	85.15 (+11.29)
	Kernel	5	92.76	83.42	86.22	85.43	77.13	93.64	98.87	73.64	86.39 (+12.53)
	Nearest Depth	5	91.03	86.49	87.07	85.73	78.01	93.42	98.78	75.28	86.98 (+13.12)
	OTLA	5	91.77	86.14	87.07	85.01	78.55	93.62	98.86	75.40	**87.05 (+13.18)**
	Block Extension	10	73.70	73.71	71.43	73.12	66.90	93.19	98.38	73.60	78.00 (+4.14)
	Random	10	91.52	83.38	86.53	84.60	76.77	93.63	98.37	73.96	86.10 (+12.24)
	Kernel	10	89.14	88.55	87.26	84.86	79.79	93.60	98.22	75.27	87.09 (+13.23)
	Nearest Depth	10	88.72	89.12	87.93	84.96	80.67	93.56	98.77	75.43	87.39 (+13.53)
	OTLA	10	91.48	91.06	88.57	87.37	81.86	93.67	98.88	75.81	**88.59 (+14.72)**

Table 9: **Scalability: ViT-B/16 → ViT-H/14.** Test accuracy (%) for task-vector transport from a ViT-B/16 source θ_A to a ViT-H/14 target θ_B, both pre-trained on `LAION-2B`, under simultaneous depth and width mismatches. Results compare layer matching strategies within Theseus and BiCo for alignment budgets $\mathcal{B} \in \{1, 5, 10\}$. AVG is the unweighted mean across the eight tasks; ΔAcc is the improvement in percentage points over the zero-shot target. Bold denotes the highest available average within each method and budget, selected before rounding. A dash indicates missing results.

Model	Layer Matching	$\mathcal{B}$	EUROSAT	SVHN	GTSRB	RESISC45	DTD	CARS	MNIST	SUN397	AVG (ΔAcc)
θ_B *zero-shot*	–	–	71.93	52.41	58.41	75.73	68.03	93.45	72.76	75.03	70.97 (+0.00)
θ_A *fine-tune*	–	–	98.64	97.49	98.68	98.59	83.04	87.92	99.43	76.96	92.59 (+21.63)
θ_B *fine-tune*	–	–	98.93	98.19	99.26	97.35	85.82	95.81	99.68	82.95	94.75 (+23.78)
Theseus	Block Extension	1	72.02	60.25	58.89	75.73	68.03	93.38	73.14	75.00	72.06 (+1.09)
	Random	1	71.89	54.30	58.65	75.86	68.14	93.31	82.47	74.93	72.45 (+1.48)
	Kernel	1	76.38	74.32	62.56	76.49	68.03	93.42	80.12	75.12	75.81 (+4.84)
	Nearest Depth	1	73.74	62.58	64.42	76.24	68.32	93.56	82.81	75.09	74.60 (+3.63)
	OTLA	1	74.20	74.93	68.68	77.28	68.44	93.60	83.91	75.11	**77.02 (+6.05)**
	Block Extension	5	73.29	53.96	59.73	75.86	68.09	93.42	74.30	75.00	71.71 (+0.74)
	Random	5	72.47	56.36	59.65	76.54	68.14	93.46	81.31	75.06	72.88 (+1.91)
	Kernel	5	76.46	75.34	62.97	78.02	69.44	93.53	87.79	75.26	77.35 (+6.39)
	Nearest Depth	5	73.70	64.76	64.22	77.74	69.44	93.42	80.20	75.27	74.85 (+3.88)
	OTLA	5	75.60	80.20	74.22	79.08	69.56	93.06	89.04	75.19	**79.50 (+8.53)**
	Block Extension	10	72.22	59.73	60.13	76.12	67.85	93.37	79.41	75.09	72.99 (+2.02)
	Random	10	73.74	58.31	58.89	77.02	67.44	93.45	79.81	75.16	72.98 (+2.01)
	Kernel	10	71.44	76.46	64.61	78.31	69.68	93.38	82.62	75.68	76.52 (+5.55)
	Nearest Depth	10	73.99	67.41	64.27	77.51	69.15	93.44	81.18	75.51	75.31 (+4.34)
	OTLA	10	76.42	77.90	70.12	80.51	69.39	93.30	89.63	75.72	**79.12 (+8.16)**
BiCo	Block Extension	1	72.84	56.55	58.92	75.78	67.97	93.46	81.62	75.03	72.77 (+1.81)
	Random	1	84.40	80.15	73.80	80.05	69.92	93.49	92.76	75.00	81.20 (+10.23)
	Kernel	1	83.17	80.83	75.34	79.26	69.98	93.49	93.61	75.38	81.38 (+10.41)
	Nearest Depth	1	84.16	81.15	76.34	79.75	70.69	93.28	93.83	75.52	81.84 (+10.87)
	OTLA	1	83.66	84.51	79.02	80.48	73.46	93.55	96.03	75.52	**83.28 (+12.32)**
	Block Extension	5	73.99	56.28	58.58	75.86	68.03	93.52	84.91	75.03	73.27 (+2.31)
	Random	5	85.06	87.29	81.82	82.75	72.28	93.40	96.56	76.11	84.41 (+13.44)
	Kernel	5	85.72	88.13	82.93	83.17	74.47	93.56	96.67	76.12	85.10 (+14.13)
	Nearest Depth	5	86.21	88.14	84.09	84.27	73.52	93.26	97.09	76.05	85.33 (+14.36)
	OTLA	5	86.42	90.47	86.02	85.96	78.19	93.63	97.69	76.60	**86.87 (+15.91)**
	Block Extension	10	81.69	63.97	61.93	79.72	68.09	93.44	89.46	75.16	76.68 (+5.71)
	Random	10	90.95	88.34	85.97	85.64	77.66	93.59	95.66	76.17	86.75 (+15.78)
	Kernel	10	90.06	91.05	85.42	86.72	77.25	93.38	97.97	76.84	87.33 (+16.36)
	Nearest Depth	10	90.45	90.23	85.89	86.88	78.50	93.60	97.76	76.45	87.47 (+16.50)
	OTLA	10	90.21	92.14	88.20	86.93	80.38	93.60	97.78	77.61	**88.35 (+17.38)**

Table 10: **Scalability: L/14 to B/16** Task-vector transport from a ViT-L/14 model to a ViT-B/16 model, both pre-trained on `DataComp-XL`. This setting involves simultaneous depth and width mismatches and employs naive layer interpolation to align transformer depth prior to transport. Results are reported for different alignment budgets $\mathcal{B}$. ΔAcc denotes the average accuracy improvement over the zero-shot baseline of model θ_B.

Model	Layer Matching	$\mathcal{B}$	EUROSAT	SVHN	GTSRB	RESISC45	DTD	CARS	MNIST	SUN397	AVG (ΔAcc)
θ_B *zero-shot*	–	–	55.88	62.58	55.21	69.19	57.45	88.88	76.48	70.01	66.96 (+0.00)
θ_A *fine-tune*	–	–	99.38	98.20	99.00	99.22	84.75	94.76	99.79	79.87	94.37 (+20.51)
θ_B *fine-tune*	–	–	99.26	97.75	98.76	98.85	83.63	92.62	99.66	76.72	93.41 (+19.55)
Theseus	Block Extension	1	–	–	–	–	–	–	–	–	–
	Random		55.97	62.64	55.27	69.07	57.57	88.88	76.88	69.97	67.03 (+0.07)
	Kernel		67.45	71.61	61.51	70.32	58.16	89.17	87.64	70.09	71.99 (+5.03)
	Nearest Depth		67.61	72.42	60.75	69.58	59.57	88.46	84.02	70.25	71.58 (+4.62)
	OTLA		63.37	77.45	60.76	70.02	57.92	88.48	89.56	70.03	**72.20 (+5.24)**
	Block Extension	5	–	–	–	–	–	–	–	–	–
	Random		58.02	62.64	55.70	69.52	57.51	88.97	80.91	70.09	67.92 (+0.96)
	Kernel		71.98	72.44	65.63	71.27	60.52	89.14	87.40	70.29	73.58 (+6.62)
	Nearest Depth		68.60	71.97	62.58	70.37	62.00	89.08	86.01	70.22	72.60 (+5.64)
	OTLA		71.07	78.62	60.87	70.79	61.17	89.15	89.03	70.21	**73.86 (+6.90)**
	Block Extension	10	–	–	–	–	–	–	–	–	–
	Random		56.34	62.96	56.00	69.82	57.57	88.82	80.96	70.25	67.84 (+0.88)
	Kernel		69.18	70.92	65.66	71.48	59.63	89.24	87.34	70.31	72.97 (+5.99)
	Nearest Depth		68.44	70.64	62.73	70.62	61.82	88.85	86.06	70.10	72.41 (+5.45)
	OTLA		76.83	79.65	62.93	71.92	60.34	88.99	87.97	70.41	**74.88 (+7.92)**
BiCo	Block Extension	1	–	–	–	–	–	–	–	–	–
	Random		71.23	67.14	69.69	71.31	58.69	88.90	88.72	69.99	73.21 (+6.25)
	Kernel		71.32	68.29	70.93	72.10	58.22	88.89	90.96	70.01	73.84 (+6.88)
	Nearest Depth		73.13	68.89	71.34	73.09	59.52	88.95	92.72	70.23	74.73 (+7.77)
	OTLA		73.62	73.45	75.56	75.45	62.71	88.92	95.12	70.23	**76.88 (+9.92)**
	Block Extension	5	–	–	–	–	–	–	–	–	–
	Random		80.78	72.85	77.05	75.47	64.30	88.82	92.87	70.22	77.79 (+10.83)
	Kernel		82.22	76.82	77.63	77.18	60.64	88.93	93.99	70.48	78.49 (+11.53)
	Nearest Depth		83.17	79.29	79.04	78.80	66.31	89.06	94.78	70.63	80.13 (+13.17)
	OTLA		86.71	83.81	80.84	80.11	69.50	88.95	97.30	70.92	**82.27 (+15.31)**
	Block Extension	10	–	–	–	–	–	–	–	–	–
	Random		87.86	79.24	82.26	78.10	66.78	87.85	97.17	71.13	81.30 (+14.34)
	Kernel		83.58	81.75	81.05	77.62	66.67	89.03	95.16	70.90	80.72 (+13.76)
	Nearest Depth		84.61	84.06	82.19	79.47	67.91	89.13	95.71	70.86	81.74 (+14.78)
	OTLA		89.42	86.03	83.58	82.05	71.39	89.01	97.63	71.60	**83.84 (+16.88)**

Table 11: **Scalability: H/14 to B/16** Task-vector transport from a ViT-H/14 model to a ViT-B/16 model, both pre-trained on `Laion-2b`. This setting involves simultaneous depth and width mismatches and employs naive layer interpolation to align transformer depth prior to transport. Results are reported for different alignment budgets $\mathcal{B}$. ΔAcc denotes the average accuracy improvement over the zero-shot baseline of model θ_B.

Model	Layer Matching	$\mathcal{B}$	EUROSAT	SVHN	GTSRB	RESISC45	DTD	CARS	MNIST	SUN397	AVG (ΔAcc)
θ_B *zero-shot*	–	–	52.72	50.16	48.13	67.88	56.80	88.38	65.80	70.20	62.51 (+0.00)
θ_A *fine-tune*	–	–	98.93	98.19	99.26	97.35	85.82	95.81	99.68	82.95	94.75 (+23.78)
θ_B *fine-tune*	–	–	98.64	97.49	98.68	98.59	83.04	87.92	99.43	76.96	92.59 (+21.62)
Theseus	Block Extension	1	–	–	–	–	–	–	–	–	–
	Random		54.94	49.81	48.32	68.06	56.44	88.37	71.03	70.22	63.40 (+0.89)
	Kernel		53.58	56.09	48.17	67.83	56.15	88.32	79.00	70.38	64.94 (+2.43)
	Nearest Depth		50.91	56.23	48.17	68.02	55.38	88.30	78.94	70.25	64.52 (+2.01)
	OTLA		56.21	60.35	54.55	68.15	56.56	88.41	75.59	70.22	**66.26 (+3.75)**
	Block Extension	5	–	–	–	–	–	–	–	–	–
	Random		56.42	54.85	48.79	68.48	56.91	88.30	76.58	70.13	65.06 (+2.55)
	Kernel		53.13	55.69	51.82	68.73	56.56	88.38	74.70	70.65	64.96 (+2.45)
	Nearest Depth		52.63	53.62	53.80	69.10	56.32	88.32	69.23	70.47	64.19 (+1.68)
	OTLA		57.98	63.22	52.56	69.61	57.80	88.37	65.34	70.38	**65.66 (+3.15)**
	Block Extension	10	–	–	–	–	–	–	–	–	–
	Random		54.86	57.03	48.94	68.02	56.62	88.21	77.06	69.34	65.01 (+2.50)
	Kernel		53.13	59.93	54.75	68.36	56.74	88.34	82.13	70.80	66.77 (+4.26)
	Nearest Depth		53.70	56.02	54.17	68.89	56.62	88.32	79.33	70.60	65.96 (+3.45)
	OTLA		58.72	68.12	51.47	70.02	58.63	88.34	82.84	70.14	**68.54 (+6.03)**
BiCo	Block Extension	1	–	–	–	–	–	–	–	–	–
	Random		74.36	64.51	69.32	69.68	56.80	88.42	89.40	70.30	72.85 (+10.34)
	Kernel		68.07	65.85	68.86	70.58	56.32	88.41	90.41	70.53	72.38 (+9.87)
	Nearest Depth		67.45	67.15	70.08	70.49	56.68	88.45	90.29	70.50	72.64 (+10.13)
	OTLA		73.37	65.70	73.13	72.86	59.10	88.53	92.66	70.55	**74.49 (+11.98)**
	Block Extension	5	–	–	–	–	–	–	–	–	–
	Random		81.07	81.69	74.46	73.76	60.46	88.39	93.90	70.53	78.03 (+15.52)
	Kernel		74.65	82.14	74.93	75.73	59.69	88.48	94.60	71.12	77.67 (+15.16)
	Nearest Depth		79.84	83.41	77.04	76.19	61.88	88.53	95.43	70.89	79.15 (+16.64)
	OTLA		85.06	81.33	79.96	81.08	68.20	88.48	95.77	71.59	**81.43 (+18.92)**
	Block Extension	10	–	–	–	–	–	–	–	–	–
	Random		82.67	81.61	74.51	73.26	61.82	88.52	94.87	70.50	78.47 (+15.96)
	Kernel		82.67	81.78	77.84	75.61	62.29	88.45	94.72	71.12	79.31 (+16.80)
	Nearest Depth		86.38	84.12	80.19	76.77	60.70	88.53	95.37	71.34	80.42 (+17.91)
	OTLA		89.42	85.00	82.99	81.09	68.91	88.49	95.98	71.96	**82.98 (+20.47)**