REFRACT: MODEL RE-BASIN THROUGH DECOMPOSABLE FINE-TUNING RESIDUALS

Anonymous authors
Paper under double-blind review

ABSTRACT

Model re-basin enables the transfer of task-specific capabilities between models, mitigating the challenges of full retraining. However, existing parameter-space approaches typically rely on weight correspondences that are difficult to establish across different architectures. To address this limitation, we shift the focus to *activation space* and transfer the functional residual, *i.e.*, the change in last-layer hidden representations induced by fine-tuning. Under standard non-linear fine-tuning, this residual entangles the contributions of individual parameter updates through non-linear interactions, making them difficult to isolate and transport to another model. We show that, when the donor model is fine-tuned in the neural tangent kernel (NTK) regime, the residual admits an exact additive decomposition over parameters, without higher-order interactions. Like a prism separating light into its constituent components, we isolate the functional contribution of each layer and transport it to corresponding layers in another model. In practice, our approach first transports these activation residuals into the target model's feature space and then learns layer-wise predictors that map its frozen intermediate activations to the transported corrections. We achieve state-of-the-art performance in capability transfer across both vision and language models, approaching full-data target fine-tuning using only a few examples from the original dataset.

1 INTRODUCTION

A task vector is the difference between fine-tuned and pre-trained weights, encoding a learned capability directly in parameter space. Task vectors can be added, negated, and composed, enabling gradient-free model editing and knowledge transfer (Ilharco et al., 2023). These benefits motivate transferring task vectors across models, allowing learned capabilities to be reused, for instance, in a newly released foundation model without repeating task-specific fine-tuning.

Model re-basin and related transfer approaches enable task vectors learned on one pre-trained model to be transferred to another, by establishing mappings between their parameter spaces (Ainsworth et al., 2023; Rinaldi et al., 2025; 2026a). However, this weight-centric strategy faces a fundamental challenge: as architectures differ in width, depth, or family, a meaningful one-to-one correspondence between their parameters may no longer exist or may be difficult to infer from limited data.

Instead of directly aligning the two parameter sets, we shift our focus to the final *activation space*, where we describe how fine-tuning steers model representations toward the downstream task. Formally, for a source model f_A with pre-trained parameters $\boldsymbol{\theta}_A^0$ and task vector $\Delta\boldsymbol{\theta}_A = \boldsymbol{\theta}_A^{\text{ft}} - \boldsymbol{\theta}_A^0$, we attempt to transfer the **functional residual** $D_A(\boldsymbol{x}) = f_A(\boldsymbol{x}; \boldsymbol{\theta}_A^0 + \Delta\boldsymbol{\theta}_A) - f_A(\boldsymbol{x}; \boldsymbol{\theta}_A^0)$ to another model $f_B(\boldsymbol{x}; \boldsymbol{\theta}_B^0)$. This functional perspective reframes cross-model transfer in terms of changes in model representations rather than changes in weight space. Although activations remain model-specific, relating them across models does not require identical internal architectures or one-to-one correspondences between their weights. The final activation space therefore could provide a natural interface for transferring task-specific capabilities across heterogeneous backbones.

Building on this formulation, our research question is: *can the functional residual induced in one model be encoded in the weights of another model with a different pre-trained initialization or architecture?* One obstacle is that standard fine-tuning produces an update whose functional contributions are entangled across parameters: in other words, changes to subsequent layers interact through the

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.

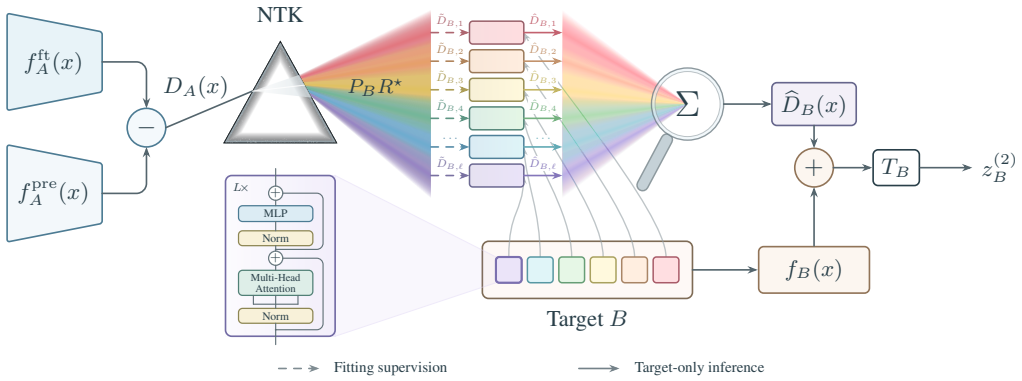

Figure 1: **Overview of REFRACT.** Linearized fine-tuning acts as a *prism*, decomposing the source functional residual into layer-wise contributions. These are transported and predicted from the target's intermediate activations, then recombined to steer its output at inference time.

model's nonlinear computations, so their functional effects cannot be isolated or transferred independently. The functional residual thus remains tightly coupled to the source model's parameters, activation functions, and architecture. To facilitate transfer to a structurally different target model, we aim to disentangle these contributions so that they can be easily remapped across architectures.

A beam of white light appears uniform, yet a prism can reveal the colors that compose it. Similarly, we aim to separate the functional residual into the contributions of individual layers to facilitate their transfer across models. In this paper, we show that **linearized fine-tuning** (Ortiz-Jimenez et al., 2023) can play the role of this *prism*. This approach trains in the tangent space of a pre-trained model, defined by its Jacobian and associated with the *neural tangent kernel* (NTK) (Jacot et al., 2018). The output change is then **linear** in the parameter displacement, so the residual $D_A(\boldsymbol{x})$ admits an exact decomposition into additive contributions from individual parameter updates.

We exploit this property of linearized fine-tuning to build REFRACT, a *divide-et-impera* approach to functional residual transfer. Instead of treating the residual $D_A(\boldsymbol{x})$ as a single object to transfer, we decompose the problem into smaller subproblems, each associated with an individual source layer. In practice, this is achieved through two stages. First, we learn a linear map R that projects the global source residual into the target model's representation space. Second, we decompose the transferred residual and fit ridge-based linear predictors to reconstruct its components from the target model's pre-trained intermediate activations. These predictors are fitted sequentially, allowing each to compensate for errors left by the preceding ones. At inference, their outputs are summed to obtain a correction to the target representation, without requiring the source model.

We show that combining functional residual decomposition with activation steering yields state-of-the-art performance on vision and language benchmarks, outperforming recent model re-basin approaches. Beyond these empirical gains, our main contribution is to motivate fine-tuning strategies that are *rebasable by design*: namely, approaches that structure task-specific updates during learning to facilitate their transfer, departing from the predominantly post-hoc perspective of existing work.

2 Background

Fine-tuning in tangent space. A pretrained network is a nonlinear function of its parameters, so the change in its output induced by a task vector $\Delta\boldsymbol{\theta}$ is a nonlinear function of $\Delta\boldsymbol{\theta}$. Tangent-space fine-tuning (Ortiz-Jimenez et al., 2023) removes this nonlinearity by replacing the network with its first-order Taylor expansion around the pretrained weights, so that the output changes linearly with the weight update. Let $f(\boldsymbol{x};\boldsymbol{\theta}) \in \mathbb{R}^d$ be a network mapping an input $\boldsymbol{x}$ to a d-dimensional representation, with pretrained parameters $\boldsymbol{\theta}_0 \in \mathbb{R}^P$, and let $J_\theta f(\boldsymbol{x};\boldsymbol{\theta}_0) \in \mathbb{R}^{d\times P}$ be its Jacobian at $\boldsymbol{\theta}_0$. Tangent-space fine-tuning learns the task vector $\Delta\boldsymbol{\theta}$ by training the linearized model

$$f_{\text{lin}}(\boldsymbol{x};\Delta\boldsymbol{\theta}) = f(\boldsymbol{x};\boldsymbol{\theta}_0) + J_\theta f(\boldsymbol{x};\boldsymbol{\theta}_0)\Delta\boldsymbol{\theta}. \tag{1}$$

In this linearized regime, the functional residual is $D(\boldsymbol{x}) = J_\theta f(\boldsymbol{x};\boldsymbol{\theta}_0)\Delta\boldsymbol{\theta}$ and can be computed with a single Jacobian–vector product (JVP), without materializing the Jacobian. Under standard

non-linear fine-tuning, the residual is instead

$$D(\boldsymbol{x}) = f(\boldsymbol{x}; \boldsymbol{\theta}_0 + \Delta\boldsymbol{\theta}) - f(\boldsymbol{x}; \boldsymbol{\theta}_0), \tag{2}$$

which requires two forward passes, one through the fine-tuned model and one through the pre-trained model. In both cases, fine-tuning can be interpreted as *activation steering*: the residual $D(\boldsymbol{x})$ is added to the pre-trained representation to obtain the adapted features. Finally, we assume a fixed linear projection head $T \in \mathbb{R}^{C\times d}$ that maps these features to task logits:

$$\boldsymbol{z}(\boldsymbol{x}) = T\big(f(\boldsymbol{x}; \theta_0) + D(\boldsymbol{x})\big). \tag{3}$$

For CLIP (Radford et al., 2021) open-vocabulary classification, the rows of T are the normalized zero-shot text embeddings of class descriptions. For a Large Language Model (LLM), T is the language modeling head, which maps each token's final representation to logits over the vocabulary.

Linearized fine-tuning, rooted in the neural tangent kernel view of neural networks (Jacot et al., 2018; Lee et al., 2020), has recently been studied in task arithmetic (Ilharco et al., 2023) and continual learning (Liu & Soatto, 2023). By enhancing *weight disentanglement* (Ortiz-Jimenez et al., 2023), it produces task vectors that are more amenable to composition through addition (*model merging*) and subtraction (*task unlearning*) (Ortiz-Jimenez et al., 2023). This perspective has also motivated regularization strategies that explicitly promote weight disentanglement by penalizing representation drift (Yoshida et al., 2025) through approximations of the generalized Gauss–Newton (GGN) matrix (Porrello et al., 2026; Sommariva et al., 2026). While prior work primarily exploits linearization to improve parameter-space disentanglement, we leverage it enabling task-specific update to be transferred and absorbed block-wise by another model. To the best of our knowledge, prior work has not investigated whether functional updates learned through linearized fine-tuning can be transferred across models with different pre-trained initializations or architectures.

3 Proposed method: REFRACT

3.1 Linearized fine-tuning as a functional prism

Our approach builds on the observation that linearized fine-tuning (Ortiz-Jimenez et al., 2023) yields an exact additive decomposition of the learned functional residual. Let θ_A^0 denote the pretrained source parameters and $\Delta\theta_A$ an update learned in tangent space. Given a partition of the parameters into L disjoint groups, such as attention and feed-forward modules in Transformer-like architectures, write $\Delta\theta_A = (\Delta\theta_{A,1}, \ldots, \Delta\theta_{A,L})$. The functional residual $D_A(\boldsymbol{x})$ then decomposes exactly as:

$$D_A(\boldsymbol{x}) = J_{\theta_A} f_A(\boldsymbol{x}; \theta_A^0)\, \Delta\theta_A = \sum_{\ell=1}^{L} J_{\theta_{A,\ell}} f_A(\boldsymbol{x}; \theta_A^0)\, \Delta\theta_{A,\ell} = \sum_{\ell=1}^{L} D_{A,\ell}(\boldsymbol{x}), \tag{4}$$

where each $D_{A,\ell}(\boldsymbol{x})$ captures the contribution of one parameter group to the source functional residual, rather than the change in that group's intermediate activations.

Computation. To compute each contribution $D_{A,\ell}(\boldsymbol{x})$, we construct a masked task vector $\Delta\theta_A^{(\ell)} = (0, \ldots, 0, \Delta\theta_{A,\ell}, 0, \ldots, 0)$, replacing all update tensors outside group ℓ with zero tensors, and evaluate the corresponding linearized model:

$$f_A^{\text{lin}}(\boldsymbol{x}; \Delta\theta_A^{(\ell)}) = f_A(\boldsymbol{x}; \theta_A^0) + J_{\theta_A} f_A(\boldsymbol{x}; \theta_A^0)\, \Delta\theta_A^{(\ell)}. \tag{5}$$

The resulting functional variation $f_A^{\text{lin}}(\boldsymbol{x}; \Delta\theta_A^{(\ell)}) - f_A(\boldsymbol{x}; \theta_A^0)$ is exactly $D_{A,\ell}(\boldsymbol{x})$ and can be computed directly through the standard Jacobian-vector product (JVP). Repeating this procedure for all groups yields the complete decomposition $\{D_{A,\ell}(\boldsymbol{x})\}_{\ell=1}^{L}$. We emphasize that standard nonlinear fine-tuning does not generally admit this exact decomposition into independent block contributions evaluated at the pretrained parameters: higher-order terms introduce nonlinear effects within blocks and interactions across blocks.

REFRACT uses this property as a *functional prism*, splitting the source functional residual into additive components and adopting a **divide-and-conquer** approach to transfer. Specifically, we transport the source functional residual into the target representation space (Sec. 3.2) and learn a predictor from the corresponding frozen target activations (Sec. 3.3). In this latter phase, we exploit the decomposition in Eq. (4) to distribute the prediction task across blocks; at inference, we combine their predictions into a single correction to the target model.

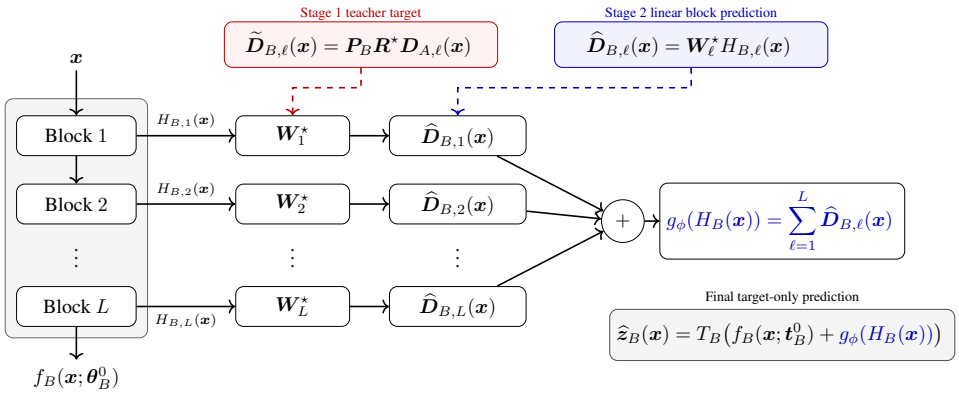

Figure 2: **Block-wise residual prediction in REFRACT.** Stage 1 provides block-level transported residual targets, while Stage 2 learns independent linear predictors from frozen receiver activations. The resulting block-wise predictions are summed to reconstruct the estimated residual $g_\phi(H_B(\boldsymbol{x}))$.

3.2 Stage 1: Constructing the transfer targets

Setup. Let $f_A : \mathcal{X} \to \mathbb{R}^{d_A}$ and $f_B : \mathcal{X} \to \mathbb{R}^{d_B}$ be the pretrained source and target representations, with fixed linear heads T_A and T_B for the same C classes or outputs. Write $D_A(\boldsymbol{x}) = f_A^{\text{ft}}(\boldsymbol{x}) - f_A(\boldsymbol{x})$ for the source's functional residual, and let $\boldsymbol{z}_A^{\text{ft}}(\boldsymbol{x}) = T_A f_A^{\text{ft}}(\boldsymbol{x})$ and $\boldsymbol{z}_B^0(\boldsymbol{x}) = T_B f_B(\boldsymbol{x})$ denote the logits of the fine-tuned source and pretrained target, respectively. We assume access to a calibration set of m examples $\{\boldsymbol{x}_i\}_{i=1}^m$ for learning the transport.

We first aim to transfer the source functional residual $D_A(\boldsymbol{x})$ to the target model. Although the feature coordinates of the two models need not correspond, their final projections map these representations into a shared logit space, whose coordinates correspond to the same classes. We use this shared space to learn a correction that brings the target logits into agreement with those of the fine-tuned source. Since the target already produces $\boldsymbol{z}_B^0(\boldsymbol{x})$, the desired correction is $\boldsymbol{e}(\boldsymbol{x}) = \boldsymbol{z}_A^{\text{ft}}(\boldsymbol{x}) - \boldsymbol{z}_B^0(\boldsymbol{x})$. We therefore learn a linear transport R that maps the source functional residual $D_A(\boldsymbol{x})$ to this correction, so that $\boldsymbol{z}_B^0(\boldsymbol{x}) + RD_A(\boldsymbol{x}) \approx \boldsymbol{z}_A^{\text{ft}}(\boldsymbol{x})$. Formally, we collect calibration columns as $D_A = [D_A(\boldsymbol{x}_i)]_{i=1}^m$ and $E = [\boldsymbol{e}(\boldsymbol{x}_i)]_{i=1}^m$ then solve

$$\min_R \ \|E - RD_A\|_F^2 + \lambda \|RD_A\|_F^2 \quad \to \quad R^\star = \tfrac{1}{1+\lambda} E D_A^\dagger, \tag{6}$$

where $\dagger$ denotes the Moore-Penrose pseudoinverse, and $R^\star$ is the minimum-norm minimizer. Unlike standard ridge regularization, the penalty is not applied directly to the coefficients of R, but to the induced correction RD_A in the target logit space. This discourages large deviations from the frozen target predictions and improves the stability of the transported correction.

Once the correction has been estimated in logit space, we lift it back to the target feature space. Specifically, we left-compose the logit-space correction $R^\star D_A(\boldsymbol{x})$ with the pseudoinverse $P_B = T_B^\dagger$ of the target head T_B. This yields the following **Stage 1 formulation** of REFRACT:

$$\boldsymbol{z}_B^{(1)}(\boldsymbol{x}) = T_B\big(f_B(\boldsymbol{x}) + \widetilde{D}_B^{(1)}(\boldsymbol{x})\big) \quad \text{where} \quad \widetilde{D}_B^{(1)}(\boldsymbol{x}) = P_B R^\star D_A(\boldsymbol{x}). \quad \text{(Stage 1)} \tag{7}$$

We refer the reader to Sec. C for a discussion of this lifting operation and Stage 1 more broadly. In particular, as shown in Sec. C.2, the transported residual $\widetilde{D}_B^{(1)}(\boldsymbol{x})$ is the minimum-norm displacement in the target feature space that realizes the attainable logit-space correction $T_B P_B R^\star D_A(\boldsymbol{x})$.

Once fitted, Stage 1 requires computing the source residual $D_A(\boldsymbol{x})$ at inference time and transporting it as $\widetilde{D}_B^{(1)}(\boldsymbol{x}) = P_B R^\star D_A(\boldsymbol{x})$. Empirically, as shown in Sec. 3.2, this correction often recovers much of the performance achieved by directly fine-tuning the target model. However, it remains expensive, since computing $D_A(\boldsymbol{x})$ still requires evaluating the source model: two forward passes for non-linear fine-tuning, or one forward pass plus a Jacobian-vector product in the linearized setting.

3.3 STAGE 2: ABSORBING THE TRANSFER TARGETS

Stage 2 removes this inference-time dependency on the source models by absorbing the transported functional displacement into the target model. To this end, we equip the target backbone with lightweight residual adapters and train them to reproduce $\widetilde{D}_B^{(1)}(\boldsymbol{x})$ from the target activations. After this compilation step, the source base and fine-tuned models are no longer required at inference time.

Formally, let $H_B(\boldsymbol{x})$ denote a collection of features extracted from one or more layers of the frozen pre-trained target model; for instance, $H_B(\boldsymbol{x})$ may simply correspond to its final-layer activation. Given the m calibration examples, we fit a predictor $g_\phi(\cdot)$ with parameters ϕ to reconstruct the transported target residual $\widetilde{D}_B^{(1)}(\boldsymbol{x}) = P_B R^\star D_A(\boldsymbol{x})$ from the target features $H_B(\boldsymbol{x})$:

$$\min_{\phi} \frac{1}{m} \sum_{i=1}^{m} \|g_\phi(H_B(\boldsymbol{x}_i)) - \widetilde{D}_B^{(1)}(\boldsymbol{x}_i)\|_2^2 + \beta\Omega(\phi), \tag{8}$$

where Ω regularizes the predictor. In our experiments, we consider two simple baseline choices for g_ϕ: a multilayer perceptron (MLP) and a linear ridge regressor. The deployed model then requires only the target base model and the fitted adapter:

$$\boldsymbol{z}_B^{(2)}(\boldsymbol{x}) = T_B\big(f_B(\boldsymbol{x}) + g_\phi(H_B(\boldsymbol{x}))\big) \quad \text{(Stage 2)} \tag{9}$$

3.4 LEVERAGING THE NTK PRISM FOR MODEL RE-BASIN

The previous steps apply to both linearized and non-linear fine-tuning. However, when the source model is trained with linearized fine-tuning, we can further exploit its additive decomposition property (Sec. 3.1) to construct a more robust predictor $g_\phi(\cdot)$ as an ensemble of local predictors, as illustrated in Fig. 2. Rather than predicting the entire transferred functional residual with a single predictor, we indeed decompose the prediction problem into L **additive subtargets**:

$$\widetilde{D}_B^{(1)}(\boldsymbol{x}) = P_B R^\star D_A(\boldsymbol{x}) = P_B R^\star \sum_{\ell=1}^{L} D_{A,\ell}(\boldsymbol{x}) = \sum_{\ell=1}^{L} \widetilde{D}_{B,\ell}(\boldsymbol{x}) \tag{10}$$

where $\widetilde{D}_{B,\ell}(\boldsymbol{x}) = P_B R^\star D_{A,\ell}(\boldsymbol{x})$ denotes the transported residual component associated with block ℓ. Leveraging this decomposition, we train one local predictor per block. Each predictor is learned through a simple closed-form ridge regression and targets the corresponding component $\widetilde{D}_{B,\ell}(\boldsymbol{x})$:

$$W_\ell^\star = \arg\min_{W} \frac{1}{m} \sum_{i=1}^{m} \|W H_{B,\ell}(\boldsymbol{x}_i) - \widetilde{D}_{B,\ell}(\boldsymbol{x}_i)\|_2^2 + \beta\|W\|_F^2, \tag{11}$$

where $H_{B,\ell}(\boldsymbol{x})$ denotes the activation extracted at block ℓ of the target model. This yields a collection of block-level predictors that distributes the learning problem across depth. At inference time, the predicted components are summed to recover the full correction, yielding the forward pass:

$$\boldsymbol{z}_B^{(2)}(\boldsymbol{x}) = T_B\big(f_B(\boldsymbol{x}) + g_\phi(H_B(\boldsymbol{x}))\big) \quad g_\phi(H_B(\boldsymbol{x})) = \sum_{\ell=1}^{L} W_\ell^\star H_{B,\ell}(\boldsymbol{x}) \tag{12}$$

This formulation can accommodate source and target architectures with different depths by grouping receiver blocks and concatenating their activations. For example, with a 12-block donor and a 24-block receiver, we define $h_{B,\ell}(x)$ by concatenating the activations after blocks $2\ell - 1$ and 2ℓ. The parameter partition and activation extraction points are specified in Secs. D.1 and D.3.

Block smoothing. Although the block-wise decomposition simplifies the prediction problem, each local regression can leave a fitting residual. Instead of discarding this error, we allow subsequent predictors to compensate for it by progressively carrying a decayed residual state across blocks. Specifically, we fit the predictors sequentially and initialize $S_0(\boldsymbol{x}_i) = 0$. At block ℓ, the target is augmented with the residual left by the preceding blocks:

$$W_\ell^\star = \arg\min_{W} \frac{1}{m} \sum_{i=1}^{m} \Big\| W H_{B,\ell}(\boldsymbol{x}_i) - \Big[\widetilde{D}_{B,\ell}(\boldsymbol{x}_i) + S_{\ell-1}(\boldsymbol{x}_i)\Big] \Big\|_2^2 + \beta\|W\|_F^2, \tag{13}$$

Table 1: **Cross-architecture task transfer.** Top-1 accuracy (%) for transfer from ViT-B/16 (A) to ViT-L/14 (B), both pretrained on DataComp-XL. Bold denotes the best AVG $\mathcal{K}$-shot result.

Model	$\mathcal{K}$	Grad. free	Tuning free	EUROSAT	SVHN	GTSRB	RESISC45	DTD	CARS	MNIST	SUN397	AVG
θ_B *zero-shot*	–	✓	–	72.25	67.63	58.42	72.65	66.54	93.13	86.70	73.75	73.88
θ_B *ft. non linear*	–	×	–	99.00	98.22	98.42	96.73	86.88	94.80	99.76	82.94	94.59
θ_A *ft. non linear*	–	×	–	99.38	98.05	98.75	98.82	82.92	92.44	99.64	76.87	93.36
θ_A *ft. linear*	–	×	–	98.27	92.03	95.18	95.40	84.81	93.31	99.47	80.26	92.34
Theseus	1	✓	×	72.40	**69.43**	61.77	72.85	66.51	93.15	88.56	73.69	74.80
BiCo	1	×	×	75.80	69.02	65.77	75.99	67.49	93.09	**92.13**	74.22	76.69
REFRACT - *MLP*	1	×	✓	77.65	69.07	70.28	79.31	69.33	**93.70**	89.67	76.44	78.18
REFRACT	1	✓	✓	**86.27**	66.06	**75.23**	**84.02**	**72.42**	93.47	89.68	**76.70**	**80.48**
Theseus	5	✓	×	74.32	69.14	62.85	73.00	66.77	93.15	89.49	73.70	75.30
BiCo	5	×	×	80.36	**73.72**	73.70	79.21	70.23	93.01	96.39	74.97	80.20
REFRACT - *MLP*	5	×	✓	88.89	70.15	79.36	82.91	75.18	94.03	93.90	76.51	82.62
REFRACT	5	✓	✓	**94.31**	72.03	**87.09**	**89.22**	**78.05**	**94.22**	**97.19**	**79.14**	**86.41**
Theseus	20	✓	×	73.40	69.13	62.46	73.01	66.59	93.07	91.17	73.65	75.31
BiCo	20	×	×	83.24	**80.43**	77.69	81.14	71.71	93.04	97.44	75.17	82.48
REFRACT - *MLP*	20	×	✓	92.96	72.43	82.48	83.33	77.66	94.24	96.47	76.96	84.57
REFRACT	20	✓	✓	**96.17**	76.31	**91.49**	**91.60**	**81.23**	**94.78**	**98.28**	**79.88**	**88.72**

and the residual passed to the next block is

$$S_\ell(\boldsymbol{x}_i) = \rho \left[\widetilde{D}_{B,\ell}(\boldsymbol{x}_i) + S_{\ell-1}(\boldsymbol{x}_i) - W_\ell^\star H_{B,\ell}(\boldsymbol{x}_i) \right], \qquad 0 \le \rho \le 1. \tag{14}$$

In this way, prediction errors made by earlier regressors can be absorbed by later ones, while the decay factor ρ progressively reduces the influence of errors originating from distant blocks. For $\rho = 0$, no residual is propagated and the method reduces to independent block-wise fitting; for $\rho = 1$, the full uncorrected residual is carried forward. Importantly, this mechanism is used only during fitting: at inference time, the block-wise predictions are still simply summed as in Eq. (12).

Computational costs. At training time, REFRACT fits the local predictors in closed form, without backpropagating through the backbone. Importantly, each predictor operates on the **pooled representation** $H_{B,\ell}(x)$, *i.e.*, , a single feature vector per example rather than the full token sequence processed by the corresponding Transformer block. As a result, the additional fitting cost is small compared with fine-tuning the backbone. At inference time, REFRACT introduces a modest overhead from evaluating the block-wise linear predictors and summing their outputs. Since these predictors again act only on pooled activations, their cost is independent of sequence length and substantially lower than the forward pass. We complement this analysis with measurements of inference time and FLOPs (see Sec. B.6), which confirm that the computational footprint of our method is negligible compared with the cost of the original backbone.

4 Experiments

Vision setup. We first consider an open-vocabulary vision setting based on CLIP classification (Radford et al., 2021). Following Rinaldi et al. (2026b), we evaluate two transfer scenarios: *i)* from **ViT-B/16 DataComp-XL** (Gadre et al., 2023) to **ViT-L/14 DataComp-XL** (Gadre et al., 2023), and *ii)* from **ViT-B/16 DataComp-XL** (Gadre et al., 2023) to **ViT-B/16 LAION-2B** (Schuhmann et al., 2022). Frozen zero-shot classifiers constructed from class-name embeddings ensure that T_A and T_B share the same class semantics. We evaluate on the eight canonical tasks of the 8-Vision protocol (Ilharco et al., 2023). For each task, both stages use the same calibration set of $\mathcal{K} \in \{1, 2, 5, 10, 20\}$ randomly sampled training examples per class.

We compare against GradFix (Rinaldi et al., 2026a), Theseus (Rinaldi et al., 2026b), and BiCo (Son et al., 2026) under the same labeled budget. To maximize the competitors' accuracy, we grid-search the scaling coefficient α applied to their re-based task vectors. Since our method does not require tuning α, we keep it fixed to $\alpha = 1$ and report our results as *Tuning free* in the following tables. We also report REFRACT - *MLP*, a variant of our method that removes the block-wise decomposition and implements Stage 2 with a single MLP predictor, to isolate the contribution of the proposed decomposed prediction scheme. Moreover, *Grad. free* indicates whether fitting or alignment avoids gradient-based optimization; additional details are provided in Sec. D.

Table 2: **Same-architecture task transfer.** Top-1 accuracy (%) for ViT-B/16 transfer from DataComp-XL (A) to LAION-2B (B).

Model	$\mathcal{K}$	Grad. free	Tuning free	EUROSAT	SVHN	GTSRB	RESISC45	DTD	CARS	MNIST	SUN397	AVG
θ_B *zero-shot*	–	✓	–	52.51	50.10	48.30	68.32	56.97	88.50	65.71	70.44	62.61
θ_B *ft non linear*	–	×	–	98.70	97.45	98.65	95.66	83.19	88.10	99.60	77.20	92.32
θ_A *ft non linear*	–	×	–	99.38	98.05	98.75	98.82	82.92	92.44	99.64	76.87	93.36
θ_A *ft linear*	–	×	–	98.27	92.03	95.18	95.40	84.81	93.31	99.47	80.26	92.34
GradFix	1	×	×	69.02	65.34	61.77	69.57	58.13	87.74	87.33	71.20	71.26
Theseus	1	✓	×	62.79	**69.33**	58.78	71.54	59.04	**88.80**	80.90	71.30	70.31
BiCo	1	×	×	62.41	62.77	**67.50**	74.84	63.63	88.35	**86.48**	72.36	**72.29**
REFRACT - *MLP*	1	×	✓	71.23	41.59	55.32	78.04	**65.60**	88.02	79.87	71.94	68.95
REFRACT	1	✓	✓	**75.17**	44.67	61.51	**79.59**	65.48	88.78	74.77	**73.15**	70.39
GradFix	5	×	×	72.62	68.89	65.34	70.83	58.82	86.37	88.41	71.62	72.86
Theseus	5	✓	×	68.31	68.80	61.76	72.05	60.30	88.78	92.11	71.50	72.95
BiCo	5	×	×	79.44	**74.67**	75.09	81.72	71.77	88.55	**94.69**	73.01	79.87
REFRACT - *MLP*	5	×	✓	85.19	42.71	69.33	84.44	**73.58**	88.97	90.17	73.08	75.93
REFRACT	5	✓	✓	**92.39**	58.20	**77.57**	**86.79**	73.31	**90.30**	94.39	**75.62**	**81.07**
GradFix	20	×	×	69.05	70.08	65.54	72.11	60.83	85.86	88.64	71.57	72.96
Theseus	20	✓	×	72.46	69.23	63.55	74.57	62.07	88.87	94.26	71.66	74.58
BiCo	20	×	×	82.41	**76.13**	78.38	84.73	73.88	88.55	95.78	73.22	81.64
REFRACT - *MLP*	20	×	✓	90.99	48.21	77.51	87.88	**79.08**	88.89	95.07	73.72	80.17
REFRACT	20	✓	✓	**95.14**	67.38	**85.33**	**89.34**	77.17	**91.23**	**96.89**	**76.16**	**84.83**

Tabs. 1 and 2 report results for $\mathcal{K} \in \{1, 5, 20\}$; results for the remaining $\mathcal{K}$ settings are provided in the appendix, along with the zero-shot target and the fully fine-tuned target and source models (θ_B^{ft} and θ_A^{ft}). In the *cross-architecture setting*, REFRACT leads at every budget: it outperforms BiCo by 3.79 points at $\mathcal{K} = 1$ and reaches 88.72% at $\mathcal{K} = 20$. GradFix cannot be applied in the cross-architecture setting. In the *same-architecture setting*, REFRACT reaches the highest average accuracy from $\mathcal{K} = 5$ onward, attaining 84.83% at $\mathcal{K} = 20$. The largest improvements over the zero-shot target occur on EuroSAT and GTSRB (42.63 and 37.03 points, respectively). Importantly, REFRACT also consistently improves over the REFRACT - *MLP* variant, which uses the same transfer mechanism but predicts the full residual with a single MLP. This gap supports the benefit of the proposed divide-and-conquer decomposition: breaking the functional residual into block-wise components yields simpler transfer targets that can be more effectively absorbed by the receiver. Overall, these results show that the method remains effective across changes in model width and depth without requiring parameter correspondence.

Text setup. We consider two transfer settings, using T5-base as the donor A and either **T5-large** (Raffel et al., 2020) or **RoBERTa-base** (Liu et al., 2019) as the target B. These pairs evaluate transfer both within and across model families. By design, we use T5 in an encoder-only configuration. We evaluate on six NLI tasks: MNLI (Williams et al., 2018), SNLI (Bowman et al., 2015), RTE (Wang et al., 2018), SciTail (Khot et al., 2018), QNLI (Wang et al., 2018), and SICK (Marelli et al., 2014). For each task, both stages use the same calibration set of $\mathcal{K} \in \{200, 300, 500\}$ examples per class. Encoder outputs are mean-pooled over tokens. Since these architectures do not provide a zero-shot classification head as in CLIP, we construct the head in the target model from the calibration set using a nearest-mean classifier. Specifically, each classifier row is defined as the mean representation of the calibration examples belonging to the corresponding class. The head in the source model is instead learned during fine-tuning.

Full results are reported in Fig. 3 and Tab. 12. For T5-base-to-T5-large, REFRACT leads at every budget and reaches 80.17% average accuracy at $\mathcal{K} = 500$, 5.26 points above the REFRACT - *MLP*. The gap widens for T5-base-to-RoBERTa-base, where REFRACT outperforms linear probing by 10.55 points. These results show effective transfer across both model scales and model families.

Cross-model mathematical reasoning. We further evaluate REFRACT on GSM8K by transferring mathematical reasoning from a tangent-space fine-tuned **Llama-3.2-3B** donor to a frozen **Llama-3.2-1B-Instruct** receiver. Using 1,024 calibration sequences, REFRACT improves flexible exact match from 32.8% to 39.3% in the 5-shot setting and from 36.6% to 39.8% in the 8-shot setting. These results show that task-specific reasoning improvements can be transferred across model scales without updating the receiver backbone. Additional details are in Sec. B.5.

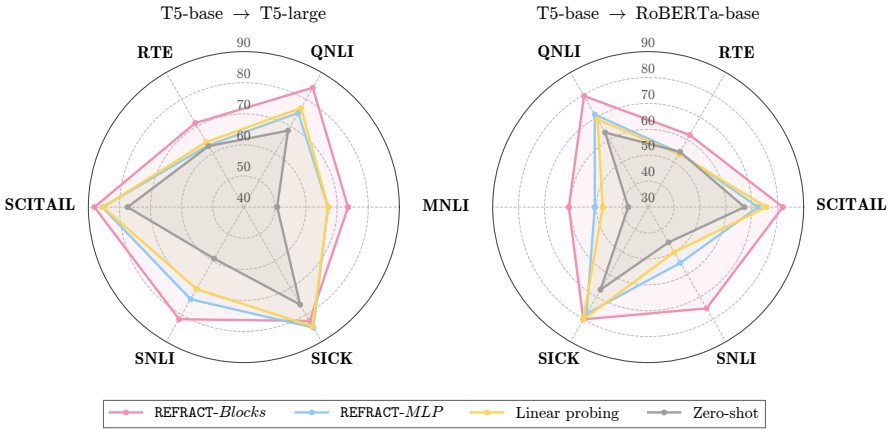

Figure 3: Accuracies on text classification in different cross-architectures settings for $\mathcal{K} = 500$.

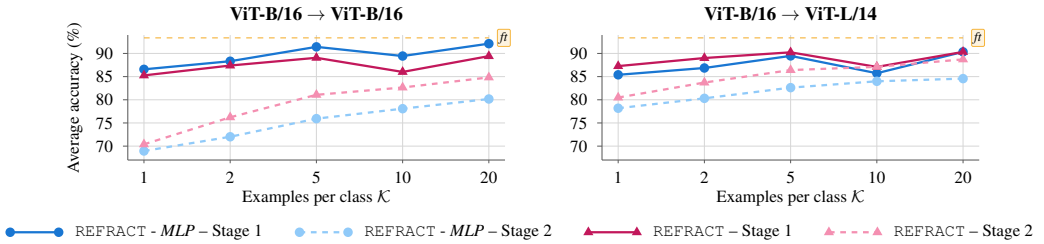

Figure 4: Comparison between Stage 1 oracle accuracy and Stage 2 rebased accuracy as a function of the number of examples per class $\mathcal{K}$. Results are averaged over eight datasets and three seeds.

5 Model Analysis

Stage 1 *vs.* Stage 2. In Fig. 4, we compare Stage 1, which transfers the functional residual without absorbing it into the target, with Stage 2, which also absorbs it. We evaluate both under non-linear and linearized fine-tuning; Stage 2 uses an MLP predictor in the former and our block-wise formulation in the latter. Two findings emerge. *i)* In both regimes, Stage 1 approaches the performance of fine-tuning the source model, supporting the transferability of functional residuals through simple linear mappings across models. *ii)* When the residual is absorbed into the target, linearized fine-tuning's block-wise structure is more robust across different shot budgets than a global MLP predictor. This supports our hypothesis that decomposing functional updates makes them easier to model and transfer across models.

The analyses in Fig. 4 refer to the vision settings. In Sec. B.1, we report the corresponding numerical results in tabular form, together with the same Stage 1 *vs.* Stage 2 analysis for T5-base→T5-large and T5-base→RoBERTa-base. We also compare against a simple global ridge predictor to assess whether the same trends persist without the block-wise decomposition. These additional experiments confirm the trends observed in vision.

Block Smoothing. Errors unresolved by one block can be propagated to subsequent blocks, which may correct them while addressing their own errors. In Sec. B.4, we compare `REFRACT`'s default $\rho = 0.9$ with $\rho = 0$, which treats blocks independently. Results in Tab. 13 show that decay-weighted error propagation improves the rebased model's performance.

On the granularity. We define functional contributions per Transformer block, grouping its attention and feed-forward parameters. Alternatively, attention and MLP sublayers can have separate contributions. As shown in Sec. B.2, this finer decomposition improves transfer by up to 2 points on average, but quadruples the number of block-wise predictors and increases fitting and inference costs. Thus, one contribution per block offers the best performance–efficiency trade-off.

Transport and prediction error. The deployed logit error splits into transport and prediction errors:

$$\boldsymbol{z}_A^{\mathrm{ft}}(\boldsymbol{x}) - \boldsymbol{z}_B^{(2)}(\boldsymbol{x}) = \boldsymbol{z}_A^{\mathrm{ft}}(\boldsymbol{x}) - \boldsymbol{z}_B^{(1)}(\boldsymbol{x}) \quad + \quad T_B\big(\widetilde{D}_B^{(1)}(\boldsymbol{x}) - g_\phi(H_B(\boldsymbol{x}))\big). \tag{15}$$

Consequently,

$$\|\boldsymbol{z}_A^{\mathrm{ft}}(\boldsymbol{x}) - \boldsymbol{z}_B^{(2)}(\boldsymbol{x})\|_2 \le \underbrace{\|\boldsymbol{z}_A^{\mathrm{ft}}(\boldsymbol{x}) - \boldsymbol{z}_B^{(1)}(\boldsymbol{x})\|_2}_{\text{transport error}} \quad + \quad \underbrace{\|T_B\|_2 \|\widetilde{D}_B^{(1)}(\boldsymbol{x}) - g_\phi(H_B(\boldsymbol{x}))\|_2}_{\text{feature-prediction error bound}}. \tag{16}$$

The second term can be tightened to $\|T_B(\widetilde{D}_B^{(1)}(\boldsymbol{x}) - g_\phi(H_B(\boldsymbol{x})))\|_2$, the discrepancy between the Stage 1 and 2 logit corrections. Proposition C.8 proves bound and its mean-squared counterpart.

6 Comparison with Related work and Limitations

Task vectors and model re-basin. Task arithmetic represents a capability as a parameter difference and edits models algebraically (Ilharco et al., 2023); making such a vector meaningful on a *different* pre-trained model defines the re-basin problem. Git Re-Basin aligns two models modulo the permutation symmetries of their hidden units (Ainsworth et al., 2023), following the conjecture that, once these symmetries are accounted for, independently trained solutions are linearly connected (Entezari et al., 2022). Recent work transports task vectors across checkpoints using transformer-specific permutations (Rinaldi et al., 2025), target-side gradient signs (Rinaldi et al., 2026a), or bilinear coordinate alignment (Son et al., 2026). Closest to us, Theseus (Rinaldi et al., 2026b) also moves beyond parameter matching: it characterizes a source task update through its functional effect on intermediate activations and transports it in closed form after aligning source and target representation spaces via orthogonal Procrustes. Our method instead uses the parameter-block contributions of a linearized source model as supervision for predictors of target-side functional corrections. The target backbone remains frozen, and the fitted predictors replace source-model evaluation at inference time. This shifts the transfer problem from constructing a target parameter update to predicting the transferred functional correction from the target model's existing representations.

Correspondence between representations. Model stitching and representation analysis study relationships between the features learned by different networks (Lenc & Vedaldi, 2015; Bansal et al., 2021; Kornblith et al., 2019; Moschella et al., 2022). Our transport is defined through aligned task logits and does not require a correspondence between the full donor and receiver feature spaces. The optional feature lift selects a minimum-norm realization of a logit correction; it does not estimate a general alignment between the models' representations.

Limitations. The receiver backbone remains frozen, and each correction is predicted from its existing activations. Consequently, `REFRACT` cannot introduce task-relevant features that are not already represented by the receiver. Moreover, the exact block-wise decomposition is available only when the donor is fine-tuned in the linearized regime. Checkpoints obtained through standard non-linear fine-tuning can still be transferred through Stage 1 and the global predictor variant, but their functional residual cannot be decomposed into exact per-block contributions. Finally, `REFRACT` introduces a small inference-time overhead due to the additional evaluation of the block-wise predictors.

7 Conclusion

We propose `REFRACT`, a model re-basin approach that transfers the functional effect of fine-tuning. Rather than matching model weights, `REFRACT` exploits linearized fine-tuning to decompose task-specific updates into additive, transferable block-wise contributions. The deployed model consists only of the target model and lightweight linear readouts, without requiring the source at inference time. We demonstrate effective transfer across model scales, architectures, and modalities, including across text-encoder families where parameter correspondences do not exist. More broadly, we show that, in the linearized regime, task-specific functional updates can be exactly decomposed over parameter blocks, potentially offering an interesting perspective for mechanistic interpretability. We also find that the fine-tuning regime itself affects how transferable a learned update is to new backbones. We view this as a first step towards fine-tuning that is *rebasable by design*: structuring task-specific updates during training so that they transfer to future backbones, rather than aligning them after the fact.

AI USE STATEMENT

Generative AI tools assisted with drafting and editing the manuscript, revising mathematical derivations and proofs, providing feedback on experimental methodology and interpretation of results, and preparing plotting scripts, table-generation code, and LaTeX. The authors made all methodological decisions and obtained the reported results through their experimental pipeline. The authors take responsibility for the correctness and integrity of all text, proofs, code, references, and reported results.

ETHICS STATEMENT

This work studies the transfer of an already-learned classification capability between pre-trained checkpoints, using public benchmark datasets and publicly released CLIP and T5 models. It involves no human subjects, no personally identifying data, and no new data collection. The method inherits whatever biases are present in the source model's fine-tuned behaviour: Stage 1 is fitted to reproduce the source's predictions, so biases in those predictions may be carried to the target along with the task. Practitioners should therefore evaluate a re-based model on the same fairness criteria they would apply to the original. Because the method makes a fine-tuned capability cheap to move onto a new backbone, it also lowers the cost of re-deploying a capability that was audited on one model onto a model that was not; we view re-auditing after transfer as necessary rather than optional. We are not aware of other ethical concerns specific to this work.

REPRODUCIBILITY STATEMENT

The transport and predictor objectives are given in Sec. 3, with derivations in Sec. A. The vision model pairs, calibration sampling and three seeds (33, 54, 89) are specified in Sec. D; $\lambda = 1$ is the default value used in all experiments, without requiring manual tuning or grid search. All competitors are reported with their best selection of α hyperparameter, selected via grid search, whereas `REFRACT` has a fixed reference value of $\alpha = 1$ in all experiment configurations. The T5 encoder features and head construction are described in Sec. A.6. Full per-dataset means, standard deviations where available, and stage comparisons are in Sec. B; these summaries are not raw per-seed outputs.

Implementation details, including feature extraction, head construction and predictor fitting, are described in Sec. D. The derivations are in Sec. A, and the error-bound statements and their complete proofs in Sec. C. We intend to release the implementation, calibration splits and per-seed outputs with that record.

REFERENCES

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

Yamini Bansal, Preetum Nakkiran, and Boaz Barak. Revisiting model stitching to compare neural representations. *arXiv preprint*, 2021.

Samuel R Bowman, Gabor Angeli, Christopher Potts, and Christopher D Manning. A large annotated corpus for learning natural language inference. In *Proceedings of the 2015 conference on empirical methods in natural language processing*, pp. 632–642, 2015.

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 *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2818–2829. IEEE, 2023. doi: 10.1109/cvpr52729.2023.00276.

Rahim Entezari, Hanie Sedghi, Olga Saukh, and Behnam Neyshabur. The role of permutation invariance in linear mode connectivity of neural networks. In *International Conference on Learning Representations*, 2022.

Samir Yitzhak Gadre, Gabriel Ilharco, Alex Fang, Jonathan Hayase, Georgios Smyrnis, Thao Nguyen, Ryan Marten, Mitchell Wortsman, Dhruba Ghosh, Jieyu Zhang, Eyal Orgad, Rahim Entezari, Giannis Daras, Sarah Pratt, Vivek Ramanujan, Yonatan Bitton, Kalyani Marathe, Stephen Mussmann, Richard Vencu, Mehdi Cherti, Ranjay Krishna, Pang Wei Koh, Olga Saukh, Alexander Ratner, Shuran Song, Hannaneh Hajishirzi, Ali Farhadi, Romain Beaumont, Sewoong Oh, Alex Dimakis, Jenia Jitsev, Yair Carmon, Vaishaal Shankar, and Ludwig Schmidt. Datacomp: In search of the next generation of multimodal datasets. In *Advances in Neural Information Processing Systems 36*, pp. 27092–27112. Neural Information Processing Systems Foundation, Inc. (NeurIPS), 2023. doi: 10.52202/075280-1179.

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

Arthur Jacot, Franck Gabriel, and Clément Hongler. Neural tangent kernel: Convergence and generalization in neural networks. In *Advances in Neural Information Processing Systems*, 2018.

Tushar Khot, Ashish Sabharwal, and Peter Clark. Scitail: A textual entailment dataset from science question answering. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32, 2018.

Simon Kornblith, Mohammad Norouzi, Honglak Lee, and Geoffrey Hinton. Similarity of neural network representations revisited. *arXiv preprint*, 2019.

Jaehoon Lee, Lechao Xiao, Samuel S Schoenholz, Yasaman Bahri, Roman Novak, Jascha Sohl-Dickstein, and Jeffrey Pennington. Wide neural networks of any depth evolve as linear models under gradient descent. *Journal of Statistical Mechanics Theory and Experiment*, 2020(12): 124002, 2020. doi: 10.1088/1742-5468/abc62b.

Karel Lenc and Andrea Vedaldi. Understanding image representations by measuring their equivariance and equivalence. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 991–999. IEEE, 2015. doi: 10.1109/cvpr.2015.7298701.

Tian Yu Liu and Stefano Soatto. Tangent model composition for ensembling and continual fine-tuning. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 18630–18640. IEEE, 2023.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint*, 2019.

Marco Marelli, Luisa Bentivogli, Marco Baroni, Raffaella Bernardi, Stefano Menini, and Roberto Zamparelli. Semeval-2014 task 1: Evaluation of compositional distributional semantic models on full sentences through semantic relatedness and textual entailment. In *Proceedings of the 8th international workshop on semantic evaluation (SemEval 2014)*, pp. 1–8, 2014.

Luca Moschella, Valentino Maiorca, Marco Fumero, Antonio Norelli, Francesco Locatello, and Emanuele Rodolà. Relative representations enable zero-shot latent space communication. *arXiv preprint*, 2022.

Guillermo Ortiz-Jimenez, Alessandro Favero, and Pascal Frossard. Task arithmetic in the tangent space: Improved editing of pre-trained models. In *Advances in Neural Information Processing Systems*, pp. 66727–66754, 2023. doi: 10.52202/075280-2913.

Angelo Porrello, Pietro Buzzega, Felix Dangel, Thomas Sommariva, Riccardo Salami, Lorenzo Bonicelli, and Simone Calderara. Dataless weight disentanglement in task arithmetic via kronecker-factored approximate curvature. In *International Conference on Learning Representations*, 2026.

Alec Radford, Jong Wook Kim, Chris Hallacy, A. Ramesh, Gabriel Goh, S. Agarwal, G. Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and I. Sutskever. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning*, 2021.

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.

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 *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 *International Conference on Learning Representations*, 2026a.

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

Christoph Schuhmann, Romain Beaumont, Richard Vencu, Cade Gordon, Ross Wightman, Mehdi Cherti, Theo Coombes, Aarush Katta, Clayton Mullis, Mitchell Wortsman, Patrick Schramowski, Srivatsa Kundurthy, Katherine Crowson, Ludwig Schmidt, Robert Kaczmarczyk, and Jenia Jitsev. LAION-5B: An open large-scale dataset for training next generation image-text models. In *Advances in Neural Information Processing Systems 35*, pp. 25278–25294. Neural Information Processing Systems Foundation, Inc. (NeurIPS), 2022. doi: 10.52202/068431-1833.

Thomas Sommariva, Francesca Morandi, Simone Calderara, and Angelo Porrello. Distilling linearized behavior into non-linear fine-tuning for effective task arithmetic. In *International Conference on Machine Learning*, 2026.

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

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R Bowman. Glue: A multi-task benchmark and analysis platform for natural language understanding. In *Proceedings of the 2018 EMNLP workshop BlackboxNLP: Analyzing and interpreting neural networks for NLP*, pp. 353–355, 2018.

Adina Williams, Nikita Nangia, and Samuel R Bowman. A broad-coverage challenge corpus for sentence understanding through inference. In *Proceedings of the 2018 conference of the North American chapter of the association for computational linguistics: human language technologies, volume 1 (long papers)*, pp. 1112–1122, 2018.

Kotaro Yoshida, Yuji Naraki, Takafumi Horie, Ryosuke Yamaki, Ryotaro Shimizu, Yuki Saito, Julian McAuley, and Hiroki Naganuma. Mastering task arithmetic: τ jp as a key indicator for weight disentanglement. In *International Conference on Learning Representations*, volume 2025, pp. 29289–29313, 2025.

A SUPPORTING DERIVATIONS AND VARIANTS

A.1 STAGE 1 SOLUTION AND CALIBRATION ERROR

The minimum-norm solution of Eq. (6) is proved in Proposition C.1. With $\lambda \geq 0$, $a = (1 + \lambda)^{-1}$, $Q_B = T_B P_B = T_B T_B^{\dagger}$, and $\Pi = D_A^{\dagger} D_A$, the realized calibration correction is $aQ_B E\Pi$. Theorem C.2 decomposes its error for an arbitrary-rank head. For $Q_B = I_C$, this reduces to

$$\|E - R^{\star} D_A\|_F^2 = \|E(I - \Pi)\|_F^2 + (1 - a)^2 \|E\Pi\|_F^2. \tag{17}$$

The first term cannot be fitted from the calibration residuals, while the second is due to output shrinkage. This identity concerns calibration data; Theorem C.4 treats evaluation inputs separately.

A.2 STAGE 2 PREDICTION ERROR

The method gives

$$z_B^{(1)}(\boldsymbol{x}) - z_B^{(2)}(\boldsymbol{x}) = T_B\big(\widetilde{D}_B^{(1)}(\boldsymbol{x}) - g_{\phi}(H_B(\boldsymbol{x}))\big).$$

Inserting $z_B^{(1)}(\boldsymbol{x})$ into $z_A^{\mathrm{ft}}(\boldsymbol{x}) - z_B^{(2)}(\boldsymbol{x})$ and applying the triangle inequality gives Eq. (16). The proof and mean-squared version are in Proposition C.8; Proposition C.9 gives the corresponding block-error identity.

A.3 ADAPTIVE DUAL-DECAY RESIDUAL PROPAGATION

We also describe an example-dependent variant, motivated by the possibility that for some calibration examples the correction from preceding blocks already recovers the source model's class prediction, in which case propagating a large residual state is unnecessary. Writing $\widehat{D}_{B,<\ell}(\boldsymbol{x}) = \sum_{j<\ell} W_j^{\star} H_{B,j}(\boldsymbol{x})$ for the cumulative correction so far, we set

$$\rho_{\ell}(\boldsymbol{x}) = \begin{cases} \rho_{\mathrm{forget}}, & \arg\max T_B\big(f_B(\boldsymbol{x}) + \widehat{D}_{B,<\ell}(\boldsymbol{x})\big) = \arg\max T_A\big(f_A(\boldsymbol{x}) + D_A(\boldsymbol{x})\big), \\ \rho_{\mathrm{retain}}, & \text{otherwise}, \end{cases} \tag{18}$$

with $0 \leq \rho_{\mathrm{forget}} < \rho_{\mathrm{retain}} \leq 1$, and use $\rho_{\ell}(\boldsymbol{x})$ in place of ρ in Eq. (14). Agreement with the teacher suppresses accumulated state quickly, whereas a remaining disagreement preserves more of it so that later blocks can compensate. The mechanism is confined to fitting; inference is unchanged. The fixed-decay identity in Proposition C.9 applies when $\rho_{\ell}(\boldsymbol{x}_i) = \rho$ for all blocks. For adaptive decay, writing $\varepsilon_j(\boldsymbol{x}_i) = \widetilde{D}_{B,j}(\boldsymbol{x}_i) - W_j^{\star} H_{B,j}(\boldsymbol{x}_i)$, the corresponding state is

$$S_{\ell}(\boldsymbol{x}_i) = \sum_{j=1}^{\ell} \left(\prod_{k=j}^{\ell} \rho_k(\boldsymbol{x}_i) \right) \varepsilon_j(\boldsymbol{x}_i).$$

A.4 EXACTNESS OF THE HEAD-AWARE LIFT

Proposition. Let $T_B \in \mathbb{R}^{C \times d_B}$ be a linear head with full row rank and let $P_B = T_B^{\dagger}$. For any displacement $v \in \mathbb{R}^C$ and feature $f_B(\boldsymbol{x}) \in \mathbb{R}^{d_B}$, the correction $\delta = P_B v$ satisfies $T_B(f_B(\boldsymbol{x}) + \delta) = T_B f_B(\boldsymbol{x}) + v$. This follows from $T_B T_B^{\dagger} = I_C$ and linearity. For an arbitrary-rank head, the realised displacement is instead $T_B T_B^{\dagger} v$, the orthogonal projection of v onto $\mathrm{col}(T_B)$. It equals v exactly when v lies in that space. This covers singular and zero heads as well; dimension compatibility alone never establishes full row rank.

Scoring conventions. The identity is for unnormalised linear scores. A fixed affine head $T_B \boldsymbol{f} + \boldsymbol{b}_B$, with $\boldsymbol{b}_B \in \mathbb{R}^C$, has the same displacement identity because its offset cancels. For a bias-free head and a nonzero corrected feature, replacing $f_B(\boldsymbol{x}) + \delta$ by its unit-normalised version rescales all class scores by the same positive factor, preserving the predicted class but not raw logits. This argument does not extend to class-dependent offsets or a zero corrected feature, whose unit normalisation is undefined. A fixed positive global temperature also preserves class order.

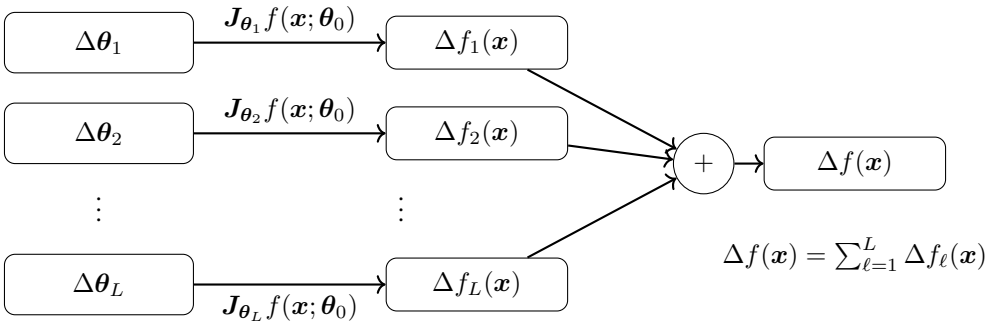

Figure 5: Additive block decomposition in NTK fine-tuning. Each block update $\Delta\boldsymbol{\theta}_\ell$ induces an independent functional contribution $\Delta f_\ell(\boldsymbol{x})$, and the total prediction change is their sum.

A.5 Kernel view of the source residual

For a tangent-space model fitted by squared loss with quadratic regularisation of $\Delta\theta_A$, the residual has a kernel-ridge representation under $K_A(\boldsymbol{x}, \boldsymbol{x}') = J_A(\boldsymbol{x})J_A(\boldsymbol{x}')^\top$, where $J_A(\boldsymbol{x}) = J_{\theta_A} f_A(\boldsymbol{x}; \theta_A^0)$. This is a conditional theoretical interpretation, not a statement of the loss used in the reported fine-tuning runs. For a scalar output, with $\boldsymbol{X}^{\mathrm{tr}}$ the full training set used to learn the source update and $\boldsymbol{r}_A$ the discrepancy between the desired task outputs and the pre-trained source predictions on it, the residual at a new input is

$$D_A(\boldsymbol{x}) = K_A(\boldsymbol{x}, \boldsymbol{X}^{\mathrm{tr}})\big(K_A(\boldsymbol{X}^{\mathrm{tr}}, \boldsymbol{X}^{\mathrm{tr}}) + \lambda_{\mathrm{K}} I\big)^{-1} \boldsymbol{r}_A = K_A(\boldsymbol{x}, \boldsymbol{X}^{\mathrm{tr}})\, \boldsymbol{\alpha}_A, \tag{19}$$

where $\lambda_{\mathrm{K}} > 0$. The vector-valued case uses the corresponding matrix-valued kernel. This representation requires the stated squared-loss training problem; linearization alone does not imply it for cross-entropy fine-tuning.

A.6 Transfer between language models

We evaluate T5-base to T5-large transfer (Raffel et al., 2020) on MNLI, SNLI, RTE, SciTail, QNLI and SICK, reporting accuracy. The representation is the encoder output averaged over sequence tokens, followed by a task-classification head. Both nonlinear and tangent-space training use full fine-tuning; the donor's classification head is fitted during that training. The receiver's head is constructed during calibration using a nearest-mean classifier. Its class scores are dot products between the pooled feature and the corresponding class means. Thus the rows of T_B are the calibration class means and C is the task's number of classes, not its vocabulary size.

B ADDITIONAL RESULTS

Unless otherwise stated, all results in this section are averaged over three seeds. All REFRACT results use a fixed correction scale $\alpha = 1$, whereas competing methods tune α on validation data. Throughout the tables, $\mathcal{K}$ denotes the number of calibration examples per class, AVG denotes the unweighted mean accuracy across the evaluated datasets, and $\pm$ indicates the sample standard deviation across seeds.

Table 3: Full calibration-budget sweep for ViT-B/16 DataComp-XL→ViT-B/16 LAION-2B transfer.

Model	$\mathcal{K}$	Grad. free	Tuning free	EUROSAT	SVHN	GTSRB	RESISC45	DTD	CARS	MNIST	SUN397	AVG
θ_B *zero-shot*	–	✓	–	52.51	50.10	48.30	68.32	56.97	88.50	65.71	70.44	62.61
θ_B *ft. non linear*	–	×	–	98.70	97.45	98.65	95.66	83.19	88.10	99.60	77.20	92.32
θ_A *ft. non linear*	–	×	–	99.38	98.05	98.75	98.82	82.92	92.44	99.64	76.87	93.36
θ_A *ft. linear*	–	×	–	98.27	92.03	95.18	95.40	84.81	93.31	99.47	80.26	92.34
GradFix	1	×	×	69.02 ± 2.25	65.34 ± 7.08	61.77 ± 0.67	69.57 ± 0.52	58.13 ± 0.34	87.74 ± 0.33	87.33 ± 0.76	71.20 ± 0.16	71.26 ± 1.51
Theseus	1	✓	×	62.79 ± 0.98	69.33 ± 1.05	58.78 ± 2.33	71.54 ± 0.13	59.04 ± 0.50	88.80 ± 0.07	80.90 ± 4.23	71.30 ± 0.22	70.31 ± 1.19
BiCo	1	×	×	62.41 ± 4.36	62.77 ± 11.80	67.50 ± 4.43	74.84 ± 2.01	63.63 ± 1.21	88.35 ± 0.12	86.48 ± 4.98	72.36 ± 0.17	**72.29 ± 3.64**
REFRACT - *MLP*	1	×	✓	71.23 ± 3.33	41.59 ± 2.70	55.32 ± 2.16	78.04 ± 1.27	65.60 ± 3.14	88.02 ± 0.16	79.87 ± 2.98	71.94 ± 0.82	68.95 ± 1.02
REFRACT	1	✓	✓	75.17 ± 1.34	44.67 ± 2.97	61.51 ± 1.64	79.59 ± 0.41	65.48 ± 0.84	88.78 ± 0.35	74.77 ± 5.97	73.15 ± 0.11	70.39 ± 0.98
GradFix	2	×	×	63.75 ± 2.42	67.85 ± 5.28	62.88 ± 1.26	70.25 ± 0.37	58.31 ± 0.12	87.48 ± 0.21	88.10 ± 1.07	71.23 ± 0.15	71.23 ± 1.36
Theseus	2	✓	×	64.01 ± 0.47	68.91 ± 0.65	59.87 ± 0.55	71.61 ± 0.28	59.02 ± 0.97	88.81 ± 0.06	86.05 ± 3.67	71.28 ± 0.04	71.20 ± 0.84
BiCo	2	×	×	68.31 ± 6.07	66.93 ± 9.13	69.98 ± 4.52	79.19 ± 0.27	67.36 ± 1.80	88.43 ± 0.15	92.14 ± 1.95	72.86 ± 0.12	75.65 ± 3.00
REFRACT - *MLP*	2	×	✓	78.27 ± 3.03	40.57 ± 2.99	62.52 ± 0.72	81.32 ± 1.05	68.09 ± 2.64	88.47 ± 0.31	84.40 ± 2.45	72.49 ± 0.07	72.02 ± 0.77
REFRACT	2	✓	✓	86.97 ± 1.60	50.22 ± 1.04	69.12 ± 0.89	83.19 ± 0.33	69.86 ± 0.62	88.98 ± 0.45	86.81 ± 2.98	74.60 ± 0.06	**76.22 ± 0.79**
GradFix	5	×	×	72.62 ± 2.03	68.89 ± 11.65	65.34 ± 1.04	70.83 ± 0.56	58.82 ± 0.70	86.37 ± 0.39	88.41 ± 1.06	71.62 ± 0.08	72.86 ± 2.19
Theseus	5	✓	×	68.31 ± 1.51	68.80 ± 0.85	61.76 ± 1.27	72.05 ± 1.47	60.30 ± 1.09	88.78 ± 0.04	92.11 ± 1.07	71.50 ± 0.03	72.95 ± 0.92
BiCo	5	×	×	79.44 ± 4.22	74.67 ± 5.98	75.09 ± 1.81	81.72 ± 2.34	71.77 ± 0.46	88.55 ± 0.01	94.69 ± 0.55	73.01 ± 0.21	79.87 ± 1.95
REFRACT - *MLP*	5	×	✓	85.19 ± 2.64	42.71 ± 3.12	69.33 ± 1.16	84.44 ± 0.47	73.58 ± 2.23	88.97 ± 0.16	90.17 ± 1.49	73.08 ± 0.49	75.93 ± 0.10
REFRACT	5	✓	✓	92.39 ± 0.46	58.20 ± 1.24	77.57 ± 0.43	86.79 ± 0.45	73.31 ± 1.04	90.30 ± 0.12	94.39 ± 0.28	75.62 ± 0.09	**81.07 ± 0.42**
GradFix	10	×	×	70.56 ± 3.17	65.36 ± 5.48	66.65 ± 1.42	72.41 ± 0.36	60.79 ± 0.17	86.30 ± 0.32	88.65 ± 0.59	71.41 ± 0.60	72.77 ± 1.51
Theseus	10	✓	×	71.24 ± 2.55	69.32 ± 1.50	63.61 ± 0.50	73.88 ± 0.77	61.26 ± 0.56	88.89 ± 0.07	93.14 ± 1.00	71.49 ± 0.02	74.10 ± 0.87
BiCo	10	×	×	80.08 ± 2.86	70.62 ± 9.61	76.88 ± 0.68	84.22 ± 1.16	72.32 ± 0.89	88.53 ± 0.04	94.70 ± 1.93	73.20 ± 0.24	80.07 ± 2.18
REFRACT - *MLP*	10	×	✓	88.52 ± 0.80	44.09 ± 1.62	73.90 ± 1.47	86.24 ± 0.40	76.95 ± 0.25	89.05 ± 0.51	92.30 ± 1.35	73.60 ± 0.16	78.08 ± 0.30
REFRACT	10	✓	✓	93.81 ± 0.31	62.25 ± 0.24	81.48 ± 0.96	87.78 ± 0.35	73.19 ± 0.45	90.93 ± 0.24	95.79 ± 0.62	75.96 ± 0.07	**82.65 ± 0.36**
GradFix	20	×	×	69.05 ± 0.18	70.08 ± 2.35	65.54 ± 0.65	72.11 ± 0.31	60.83 ± 1.16	85.86 ± 0.17	88.64 ± 0.60	71.57 ± 0.18	72.96 ± 0.70
Theseus	20	✓	×	72.46 ± 3.59	69.23 ± 1.03	63.55 ± 0.76	74.57 ± 0.53	62.07 ± 0.99	88.87 ± 0.05	94.26 ± 0.71	71.66 ± 0.10	74.58 ± 0.97
BiCo	20	×	×	82.41 ± 2.16	76.13 ± 7.28	78.38 ± 1.61	84.73 ± 1.05	73.88 ± 0.84	88.55 ± 0.02	95.78 ± 0.54	73.22 ± 0.17	81.64 ± 1.71
REFRACT - *MLP*	20	×	✓	90.99 ± 0.97	48.21 ± 2.27	77.51 ± 0.90	87.88 ± 0.20	79.08 ± 1.53	88.89 ± 0.31	95.07 ± 0.82	73.72 ± 0.19	80.17 ± 0.40
REFRACT	20	✓	✓	95.14 ± 0.12	67.38 ± 0.69	85.33 ± 0.18	89.34 ± 0.31	77.17 ± 0.80	91.23 ± 0.10	96.89 ± 0.53	76.16 ± 0.12	**84.83 ± 0.18**

Vision results. Sec. 4 considers two source–target pairs: ViT-B/16 DataComp-XL→ViT-B/16 LAION-2B, which changes the pre-training distribution, and ViT-B/16→ViT-L/14 on DataComp-XL, which additionally changes model width and depth. Tabs. 3 and 4 extend the main results to the complete calibration-budget sweep. In the same-architecture setting, REFRACT becomes the best-performing method at $\mathcal{K} = 2$, reaching 76.22 ± 0.79 compared with 75.65 ± 3.00 for BiCo, and increases its margin to 2.58 points at $\mathcal{K} = 10$. In the cross-architecture setting, it leads throughout the sweep, exceeding the strongest baseline by 5.33 points at $\mathcal{K} = 2$ and 2.65 points at $\mathcal{K} = 10$. Absolute results across the two settings are not directly comparable because the ViT-L/14 receiver has a stronger zero-shot starting point (73.88% versus 62.61%).

Text results. Tabs. 5 and 6 evaluate NLI transfer from T5-base to T5-large and RoBERTa-base using the protocol of Sec. D. θ_B's head is initialized via Nearest Mean Classifier (NMC). We compare with the receiver reference, linear probing under the same calibration budget, and Theseus (Rinaldi et al., 2026b).

REFRACT achieves the highest AVG at every budget for both receivers. Relative to linear probing, its gains on T5-large are 7.0, 5.9, and 5.5 points for $\mathcal{K} \in \{200, 300, 500\}$; on RoBERTa-base, they are 10.6, 10.3, and 10.6 points. The largest improvements occur on SNLI, QNLI, and MNLI. By contrast, REFRACT - *MLP*, which predicts the correction only from the receiver's final representation, remains within approximately one point of linear probing, showing that most of the improvement comes from exploiting intermediate receiver activations.

Theseus remains below the receiver reference on T5-large and degrades substantially on RoBERTa-base. RTE is the least stable task because its validation and test splits contain only 28 and 249

Table 4: Full calibration-budget sweep for ViT-B/16→ViT-L/14 transfer on DataComp-XL.

Model	$\mathcal{K}$	Grad. free	Tuning free	EUROSAT	SVHN	GTSRB	RESISC45	DTD	CARS	MNIST	SUN397	AVG
θ_B *zero-shot*	–	✓	–	72.25	67.63	58.42	72.65	66.54	93.13	86.70	73.75	73.88
θ_B *ft. non linear*	–	×	–	99.00	98.22	98.42	96.73	86.88	94.80	99.76	82.94	94.59
θ_A *ft. non linear*	–	×	–	99.38	98.05	98.75	98.82	82.92	92.44	99.64	76.87	93.36
θ_A *ft. linear*	–	×	–	98.27	92.03	95.18	95.40	84.81	93.31	99.47	80.26	92.34
Theseus	1	✓	×	72.40 ± 0.29	69.43 ± 0.84	61.77 ± 0.52	72.85 ± 0.82	66.51 ± 0.12	93.15 ± 0.09	88.56 ± 1.68	73.69 ± 0.07	74.80 ± 0.55
BiCo	1	×	×	75.80 ± 1.87	69.02 ± 0.59	65.77 ± 3.19	75.99 ± 0.52	67.49 ± 0.43	93.09 ± 0.13	92.13 ± 2.20	74.22 ± 0.46	76.69 ± 1.17
REFRACT - *MLP*	1	×	✓	77.65 ± 4.65	69.07 ± 1.87	70.28 ± 1.72	79.31 ± 0.75	69.33 ± 0.66	93.70 ± 0.56	89.67 ± 0.60	76.44 ± 0.21	78.18 ± 0.68
REFRACT	1	✓	✓	86.27 ± 1.99	66.06 ± 3.95	75.23 ± 2.67	84.02 ± 0.54	72.42 ± 0.19	93.47 ± 0.04	89.68 ± 4.25	76.70 ± 0.08	**80.48 ± 1.71**
Theseus	2	✓	×	72.81 ± 0.56	68.74 ± 0.87	62.21 ± 0.96	72.73 ± 0.27	66.63 ± 0.24	93.15 ± 0.04	89.19 ± 2.40	73.72 ± 0.05	74.90 ± 0.67
BiCo	2	×	×	77.34 ± 2.49	70.78 ± 3.04	70.10 ± 2.16	76.93 ± 0.51	69.15 ± 0.12	93.02 ± 0.00	95.17 ± 1.71	74.65 ± 0.11	78.39 ± 1.27
REFRACT - *MLP*	2	×	✓	83.09 ± 1.97	69.77 ± 0.70	75.53 ± 1.23	80.69 ± 0.91	71.99 ± 1.76	93.82 ± 0.29	90.97 ± 1.68	76.64 ± 0.26	80.31 ± 0.35
REFRACT	2	✓	✓	92.13 ± 0.77	68.34 ± 1.44	81.24 ± 0.19	86.48 ± 0.13	75.93 ± 0.52	93.50 ± 0.16	93.96 ± 1.60	78.15 ± 0.03	**83.72 ± 0.61**
Theseus	5	✓	×	74.32 ± 1.32	69.14 ± 1.33	62.85 ± 0.52	73.00 ± 0.49	66.77 ± 0.18	93.15 ± 0.06	89.49 ± 1.83	73.70 ± 0.09	75.30 ± 0.73
BiCo	5	×	×	80.36 ± 3.85	73.72 ± 3.41	73.70 ± 1.85	79.21 ± 0.41	70.23 ± 0.38	93.01 ± 0.06	96.39 ± 0.51	74.97 ± 0.14	80.20 ± 1.33
REFRACT - *MLP*	5	×	✓	88.89 ± 2.58	70.15 ± 0.39	79.36 ± 0.93	82.91 ± 0.61	75.18 ± 1.40	94.03 ± 0.18	93.90 ± 0.83	76.51 ± 0.46	82.62 ± 0.46
REFRACT	5	✓	✓	94.31 ± 0.52	72.03 ± 0.85	87.09 ± 0.46	89.22 ± 0.17	78.05 ± 0.51	94.22 ± 0.21	97.19 ± 0.53	79.14 ± 0.19	**86.41 ± 0.43**
Theseus	10	✓	×	73.83 ± 1.20	69.18 ± 1.12	62.18 ± 0.34	73.02 ± 0.30	66.53 ± 0.12	93.13 ± 0.02	89.66 ± 1.08	73.80 ± 0.03	75.17 ± 0.53
BiCo	10	×	×	82.50 ± 0.55	75.64 ± 1.55	75.62 ± 1.84	80.52 ± 0.38	70.94 ± 0.46	93.09 ± 0.07	96.84 ± 0.25	75.13 ± 0.14	81.29 ± 0.66
REFRACT - *MLP*	10	×	✓	90.99 ± 0.17	70.56 ± 1.13	81.71 ± 1.15	83.86 ± 0.88	77.48 ± 1.25	94.44 ± 0.12	95.93 ± 0.68	76.91 ± 0.13	83.99 ± 0.25
REFRACT	10	✓	✓	95.64 ± 0.25	73.58 ± 0.62	89.42 ± 0.45	89.97 ± 0.28	76.52 ± 0.63	94.54 ± 0.02	97.58 ± 0.40	79.65 ± 0.08	**87.11 ± 0.34**
Theseus	20	✓	×	73.40 ± 0.76	69.13 ± 0.78	62.46 ± 0.19	73.01 ± 0.33	66.59 ± 0.25	93.07 ± 0.08	91.17 ± 0.51	73.65 ± 0.03	75.31 ± 0.37
BiCo	20	×	×	83.24 ± 3.08	80.43 ± 1.82	77.69 ± 1.16	81.14 ± 0.66	71.71 ± 0.35	93.04 ± 0.07	97.44 ± 0.22	75.17 ± 0.16	82.48 ± 0.94
REFRACT - *MLP*	20	×	✓	92.96 ± 0.52	72.43 ± 0.36	82.48 ± 0.52	83.33 ± 0.47	77.66 ± 0.75	94.24 ± 0.21	96.47 ± 0.54	76.96 ± 0.35	84.57 ± 0.11
REFRACT	20	✓	✓	96.17 ± 0.27	76.31 ± 0.56	91.49 ± 0.33	91.60 ± 0.15	81.23 ± 0.49	94.78 ± 0.10	98.28 ± 0.05	79.88 ± 0.15	**88.72 ± 0.26**

Table 5: NLI task transfer from T5-base to T5-large.

Model	$\mathcal{K}$	MNLI	SICK	RTE	SciTail	QNLI	SNLI	AVG
θ_B *zero-shot*	300	50.72	76.22	62.65	77.39	68.39	59.11	65.75
θ_B *fine-tuned*	–	90.40	93.40	88.80	92.10	94.70	89.40	91.47
θ_A *fine-tuned*	–	86.40	85.60	73.65	91.55	91.40	91.20	86.63
Theseus	200	47.22	71.22	64.26	77.70	68.37	39.11	61.31
Linear Probing	200	61.43	81.39	61.04	81.28	73.24	65.57	70.66
REFRACT - *MLP*	200	61.17	81.67	59.17	81.41	71.65	70.13	70.86
REFRACT	200	69.17	80.06	69.48	85.85	82.43	79.00	**77.67**
Theseus	300	46.37	69.37	65.19	77.48	67.72	38.09	60.71
Linear Probing	300	64.33	82.33	61.71	83.70	75.61	67.70	72.57
REFRACT - *MLP*	300	63.96	82.81	61.58	83.87	74.22	72.57	73.17
REFRACT	300	70.91	81.02	69.48	86.39	82.43	80.26	**78.42**
Theseus	500	47.28	72.07	65.33	77.50	67.93	37.61	61.29
Linear Probing	500	67.11	84.28	64.12	85.30	76.80	70.41	74.67
REFRACT - *MLP*	500	67.06	84.74	62.92	85.50	75.00	74.22	74.91
REFRACT	500	73.52	82.28	71.22	88.06	84.33	81.63	**80.17**

examples, respectively: on RoBERTa-base, linear probing and REFRACT - *MLP* remain at the receiver reference of 54.6%, whereas REFRACT reaches approximately 61-62%.

B.1 Comparison between Stage 1 and Stage 2

Tabs. 7 and 8 report the per-dataset results, while Fig. 4 summarizes the AVG trends. Stage 1 evaluates the donor-dependent transported correction before amortisation; Stage 2 predicts that correction from receiver activations and is the final target-only model. A stronger Stage 1 teacher does not necessarily produce a better deployed model. In the same-architecture setting at $\mathcal{K} = 20$, REFRACT - *MLP* decreases from 92.11% to 80.17%, an amortisation gap of 11.94 points. The block-wise variant instead retains 84.83% from a weaker 89.42% teacher, reducing the gap to 4.59 points and outperforming the deployed MLP by 4.66 points. At $\mathcal{K} = 5$, block-wise prediction similarly reduces

Table 6: Cross-family NLI task transfer from T5-base to RoBERTa-base.

Model	$\mathcal{K}$	MNLI	SICK	RTE	SciTail	QNLI	SNLI	AVG
θ_B *zero-shot*	300	37.72	66.83	54.62	67.17	63.22	45.72	55.88
θ_B *fine-tuned*	–	87.94	89.59	87.36	93.51	90.77	91.41	90.10
θ_A *fine-tuned*	–	86.40	85.60	73.65	91.55	91.40	91.20	86.63
Theseus	200	34.94	16.44	48.73	61.67	51.67	33.93	41.23
Linear Probing	200	43.52	78.44	51.81	72.31	65.13	49.57	60.13
REFRACT - *MLP*	200	46.96	76.96	53.01	70.44	68.00	51.81	61.20
REFRACT	200	57.93	77.78	61.18	78.87	77.31	71.52	70.76
Theseus	300	34.94	16.44	46.18	61.67	51.67	33.93	40.81
Linear Probing	300	46.00	78.69	54.62	74.13	66.87	50.07	61.73
REFRACT - *MLP*	300	49.56	77.63	54.22	71.72	70.20	53.80	62.85
REFRACT	300	59.83	79.13	61.98	79.80	78.04	73.48	72.04
Theseus	500	34.87	16.44	51.27	61.67	51.67	34.67	41.76
Linear Probing	500	47.52	79.98	53.68	75.67	69.09	50.15	62.68
REFRACT - *MLP*	500	50.59	77.52	54.22	72.43	71.28	54.85	63.48
REFRACT	500	60.63	80.04	62.12	81.98	79.50	75.11	73.23

Table 7: Stage 1 teacher and Stage 2 target-only accuracies for same-architecture vision transfer. The 8-Vision averages are summarised in Fig. 4.

Stage	$\mathcal{K}$	EuroSAT	SVHN	GTSRB	RESISC45	DTD	Cars	MNIST	SUN397	AVG
REFRACT - *MLP*										
S1 *Oracle*	1	98.70	75.75	94.37	94.50	77.48	86.54	95.61	69.63	86.57
S2 *Rebase*		71.23	41.59	55.32	78.04	65.60	88.02	79.87	71.94	68.95
S1 *Oracle*	2	99.01	85.18	97.72	97.40	80.40	74.41	97.97	74.51	88.32
S2 *Rebase*		78.27	40.57	62.52	81.32	68.09	88.47	84.40	72.49	72.02
S1 *Oracle*	5	99.38	92.01	98.29	97.97	81.29	86.90	99.47	76.02	91.42
S2 *Rebase*		85.19	42.71	69.33	84.44	73.58	88.97	90.17	73.08	75.93
S1 *Oracle*	10	99.45	93.45	96.30	94.60	67.65	88.44	99.52	76.09	89.44
S2 *Rebase*		88.52	44.09	73.90	86.24	76.95	89.05	92.30	73.60	78.08
S1 *Oracle*	20	99.41	93.79	98.41	98.17	82.56	88.87	99.49	76.21	92.11
S2 *Rebase*		90.99	48.21	77.51	87.88	79.08	88.89	95.07	73.72	80.17
REFRACT										
S1 *Oracle*	1	94.06	75.56	88.45	89.54	72.12	91.68	98.09	72.44	85.26
S2 *Rebase*		75.17	44.67	61.51	79.59	65.48	88.78	74.77	73.15	70.39
S1 *Oracle*	2	96.38	79.25	91.97	91.67	76.01	88.15	98.82	76.85	87.39
S2 *Rebase*		86.97	50.22	69.12	83.19	69.86	88.98	86.81	74.60	76.22
S1 *Oracle*	5	97.72	81.87	93.14	92.67	77.24	92.52	99.20	78.11	89.06
S2 *Rebase*		92.39	58.20	77.57	86.79	73.31	90.30	94.39	75.62	81.07
S1 *Oracle*	10	97.93	82.19	89.21	87.85	60.52	92.96	99.17	78.28	86.01
S2 *Rebase*		93.81	62.25	81.48	87.78	73.19	90.93	95.79	75.96	82.65
S1 *Oracle*	20	98.02	81.53	93.30	93.09	78.96	93.08	99.16	78.27	89.42
S2 *Rebase*		95.14	67.38	85.33	89.34	77.17	91.23	96.89	76.16	84.83

the gap from 15.49 to 7.99 points. These results are consistent with the motivation of REFRACT: additive block targets create simpler local prediction problems, and block smoothing allows later predictors to compensate for earlier errors.

Table 8: Stage 1 teacher and Stage 2 target-only accuracies for cross-architecture vision transfer. The 8-Vision averages are summarised in Fig. 4.

Stage	$\mathcal{K}$	EuroSAT	SVHN	GTSRB	RESISC45	DTD	Cars	MNIST	SUN397	AVG
REFRACT - *MLP*										
S1 *Oracle*	1	94.82	76.02	90.08	88.95	74.41	92.64	95.47	70.74	85.39
S2 *Rebase*		77.65	69.07	70.28	79.31	69.33	93.70	89.67	76.44	78.18
S1 *Oracle*	2	96.63	78.78	93.29	91.38	77.38	85.07	96.08	76.30	86.86
S2 *Rebase*		83.09	69.77	75.53	80.69	71.99	93.82	90.97	76.64	80.31
S1 *Oracle*	5	97.33	84.65	94.56	92.99	78.74	92.84	96.78	77.89	89.47
S2 *Rebase*		88.89	70.15	79.36	82.91	75.18	94.03	93.90	76.51	82.62
S1 *Oracle*	10	97.97	85.82	89.45	85.25	58.15	93.51	97.70	78.01	85.73
S2 *Rebase*		90.99	70.56	81.71	83.86	77.48	94.44	95.93	76.91	83.99
S1 *Oracle*	20	98.02	84.84	95.63	93.89	80.14	93.77	98.34	78.08	90.34
S2 *Rebase*		92.96	72.43	82.48	83.33	77.66	94.24	96.47	76.96	84.57
REFRACT										
S1 *Oracle*	1	94.46	80.16	89.64	89.80	76.04	94.38	98.16	75.53	87.27
S2 *Rebase*		86.27	66.06	75.23	84.02	72.42	93.47	89.68	76.70	80.48
S1 *Oracle*	2	96.75	81.94	92.40	91.94	78.88	91.85	98.72	79.60	89.01
S2 *Rebase*		92.13	68.34	81.24	86.48	75.93	93.50	93.96	78.15	83.72
S1 *Oracle*	5	97.58	83.37	93.32	92.99	80.20	94.58	99.01	80.87	90.24
S2 *Rebase*		94.31	72.03	87.09	89.22	78.05	94.22	97.19	79.14	86.41
S1 *Oracle*	10	97.83	83.45	90.19	88.15	62.29	94.83	99.03	80.95	87.09
S2 *Rebase*		95.64	73.58	89.42	89.97	76.52	94.54	97.58	79.65	87.11
S1 *Oracle*	20	97.86	82.52	93.51	93.36	80.52	94.79	99.09	81.01	90.33
S2 *Rebase*		96.17	76.31	91.49	91.60	81.23	94.78	98.28	79.88	88.72

The same pattern holds across architectures. For the block-wise method, the Stage 1–Stage 2 gap decreases from 6.79 points at $\mathcal{K} = 1$ to 1.61 points at $\mathcal{K} = 20$, compared with 5.77 points for REFRACT - *MLP* at the latter budget. At $\mathcal{K} = 10$, Stage 2 marginally exceeds its teacher (87.11% versus 87.09%). This does not imply a negative residual-prediction error: Stage 2 approximates transported features, whereas the tables measure classification accuracy, and regularization can occasionally improve the resulting decision boundary. Stage 1 is therefore a diagnostic of transport quality rather than a strict upper bound on target-only accuracy.

As shown in Fig. 4, Stage 1 drops between $\mathcal{K} = 5$ and $\mathcal{K} = 10$, largely because of DTD, whereas every Stage 2 curve improves monotonically.

B.2 Effect of block granularity

Block granularity determines the resolution at which the donor residual is decomposed and reconstructed by Stage 2. Each source group produces an additive component through a masked JVP, which is transported to the receiver space and predicted by a separate ridge regressor. Coarser partitions reduce computational and statistical complexity but discard information about the origin of the update; finer partitions preserve greater locality at the cost of additional JVPs and regressors.

We examine this trade-off for ViT-B/16→ViT-B/16, where components can be matched one-to-one. REFRACT - *Model* places all visual parameters in one group and predicts their combined residual from the final receiver embedding. REFRACT uses the 12 transformer residual blocks and the output projection, yielding 13 groups. REFRACT - *Layers* uses 50 operator-level groups: the convolutional stem; packed QKV, attention output, MLP expansion, and MLP projection within each transformer block; and the final visual projection. Q, K, and V remain grouped because OpenCLIP stores them

in a single packed projection, while embeddings and normalization parameters are assigned to the semantically following operator.

Table 9: Effect of block granularity on the final target-only top-1 accuracy (%) for ViT-B/16 DataComp-XL→ViT-B/16 LAION-2B.

Granularity	$\mathcal{K}$	EuroSAT	SVHN	GTSRB	RESISC45	DTD	Cars	MNIST	SUN397	AVG
REFRACT - *Model*	1	57.53	50.06	58.09	73.70	60.64	88.81	67.43	74.66	66.36
REFRACT	1	75.17	44.67	61.51	79.59	65.48	88.78	74.77	73.15	70.39
REFRACT - *Layers*	1	76.91	49.24	61.55	80.32	66.31	88.52	80.25	73.75	72.11
REFRACT - *Model*	2	64.44	52.85	61.47	76.40	64.89	89.18	79.20	75.30	70.47
REFRACT	2	86.97	50.22	69.12	83.19	69.86	88.98	86.81	74.60	76.22
REFRACT - *Layers*	2	86.42	55.99	68.54	84.02	71.45	88.62	85.73	76.36	77.14
REFRACT - *Model*	5	76.05	56.96	69.25	81.69	70.04	89.97	86.93	76.00	75.86
REFRACT	5	92.39	58.20	77.57	86.79	73.31	90.30	94.39	75.62	81.07
REFRACT - *Layers*	5	92.22	58.75	78.07	87.88	73.23	90.17	93.87	76.68	81.36
REFRACT - *Model*	10	83.83	58.65	74.74	83.81	71.99	90.34	90.03	76.57	78.75
REFRACT	10	93.81	62.25	81.48	87.78	73.19	90.93	95.79	75.96	82.65
REFRACT - *Layers*	10	94.07	62.86	82.11	88.20	72.34	90.92	95.94	77.23	82.96
REFRACT - *Model*	20	88.02	61.51	78.65	85.61	73.94	90.55	91.63	76.71	80.83
REFRACT	20	95.14	67.38	85.33	89.34	77.17	91.23	96.89	76.16	84.83
REFRACT - *Layers*	20	95.56	68.52	86.75	89.89	76.60	91.38	97.03	77.53	85.41

Tab. 9 shows that finer partitions consistently improve AVG accuracy. At $\mathcal{K} = 1$, the REFRACT - *Model*, REFRACT, and REFRACT - *Layers* variants obtain 66.36%, 70.39%, and 72.11%, respectively; at $\mathcal{K} = 20$, they reach 80.83%, 84.83%, and 85.41%. The operator-level advantage over residual blocks is largest at $\mathcal{K} = 1$ (1.72 points), driven mainly by SVHN and MNIST, but narrows to 0.29–0.58 points from $\mathcal{K} = 5$ onward. A global predictor therefore loses substantial information about the internal origin of the update, whereas residual-block decomposition captures most of the available gain. Since the operator-level variant requires 50 local JVPs and regressors instead of 13, residual blocks provide the best accuracy–efficiency trade-off, while the finest partition offers a modest gain when computational cost is secondary.

B.3 REFRACT VARIANTS

Tabs. 10 to 12 compare four variants of REFRACT in three settings. The first is a language setting (T5-base → T5-large). The other two are vision settings: one changes the pre-training data (DataComp-XL → LAION-2B), and the other changes the model scale (ViT-B/16 → ViT-L/14). Three trends hold across all settings.

(i) REFRACT achieves the best average accuracy in every setting and for every budget $\mathcal{K}$. On T5 it outperforms the second-best variant by 3.8–4.5 points, and it improves over the zero-shot target model by up to $+14.4$ points (80.17 vs. 65.75). On vision its margin reaches $+1.6$ points.

(ii) Enforcing linearity helps. The linear MLP outperforms its non-linear counterpart in every setting and for every budget, by up to $+4.05$ points (ViT-B → ViT-B, $\mathcal{K}$=5) and $+2.63$ points (ViT-B → ViT-L, $\mathcal{K}$=20). The extra flexibility of a non-linear fine-tuning therefore does not improve transfer, and linear fine-tuning is sufficient and a more robust choice.

(iii) Performance improves as the calibration budget $\mathcal{K}$ increases for all variants, while their relative ranking remains broadly stable. The differences between methods are most pronounced in the low-data regime, where accurately reconstructing the transferred residual is particularly challenging. Overall, REFRACT transfers capabilities reliably across modalities, pre-training distributions, and model scales, and provides the strongest default performance even with only a few calibration examples per class.

Table 10: Ablation of donor fine-tuning and predictor variants for ViT-B/16 DataComp-XL→ViT-B/16 LAION-2B.

Model	Linear	κ	EuroSAT	SVHN	GTSRB	RESISC45	DTD	Cars	MNIST	SUN397	AVG
REFRACT - *MLP*	×	1	71.23	41.59	55.32	78.04	65.60	88.02	79.87	71.94	68.95
REFRACT - *MLP*	✓	1	70.12	45.88	62.60	78.99	65.78	89.22	72.40	74.98	70.00
REFRACT - *Ridge*	✓	1	57.53	50.06	58.09	73.70	60.64	88.81	67.43	74.66	66.36
REFRACT	✓	1	75.17	44.67	61.51	79.59	65.48	88.78	74.77	73.15	70.39
REFRACT - *MLP*	×	2	78.27	40.57	62.52	81.32	68.09	88.47	84.40	72.49	72.02
REFRACT - *MLP*	✓	2	80.37	48.87	69.07	83.49	68.97	89.84	84.63	75.99	75.15
REFRACT - *Ridge*	✓	2	64.44	52.85	61.47	76.40	64.89	89.18	79.20	75.30	70.47
REFRACT	✓	2	86.97	50.22	69.12	83.19	69.86	88.98	86.81	74.60	76.22
REFRACT - *MLP*	×	5	85.19	42.71	69.33	84.44	73.58	88.97	90.17	73.08	75.93
REFRACT - *MLP*	✓	5	90.25	53.26	77.07	87.83	73.23	90.59	91.07	76.54	79.98
REFRACT - *Ridge*	✓	5	76.05	56.96	69.25	81.69	70.04	89.97	86.93	76.00	75.86
REFRACT	✓	5	92.39	58.20	77.57	86.79	73.31	90.30	94.39	75.62	81.07
REFRACT - *MLP*	×	10	88.52	44.09	73.90	86.24	76.95	89.05	92.30	73.60	78.08
REFRACT - *MLP*	✓	10	91.73	57.52	81.13	87.99	75.71	91.29	93.70	76.47	81.94
REFRACT - *Ridge*	✓	10	83.83	58.65	74.74	83.81	71.99	90.34	90.03	76.57	78.75
REFRACT	✓	10	93.81	62.25	81.48	87.78	73.19	90.93	95.79	75.96	82.65
REFRACT - *MLP*	×	20	90.99	48.21	77.51	87.88	79.08	88.89	95.07	73.72	80.17
REFRACT - *MLP*	✓	20	92.72	60.17	83.90	89.05	77.48	91.00	94.67	76.78	83.22
REFRACT - *Ridge*	✓	20	88.02	61.51	78.65	85.61	73.94	90.55	91.63	76.71	80.83
REFRACT	✓	20	95.14	67.38	85.33	89.34	77.17	91.23	96.89	76.16	84.83

Table 11: Ablation of donor fine-tuning and predictor variants for ViT-B/16→ViT-L/14 transfer.

Model	Linear	κ	EuroSAT	SVHN	GTSRB	RESISC45	DTD	Cars	MNIST	SUN397	AVG
REFRACT - *MLP*	×	1	77.65	69.07	70.28	79.31	69.33	93.70	89.67	76.44	78.18
REFRACT - *MLP*	✓	1	80.00	67.52	75.03	81.48	70.57	93.82	86.73	77.45	79.08
REFRACT - *Ridge*	✓	1	74.32	70.37	70.81	76.72	68.44	93.91	86.93	76.84	77.29
REFRACT	✓	1	86.27	66.06	75.23	84.02	72.42	93.47	89.68	76.70	80.48
REFRACT - *MLP*	×	2	83.09	69.77	75.53	80.69	71.99	93.82	90.97	76.64	80.31
REFRACT - *MLP*	✓	2	84.07	68.61	81.42	85.29	73.58	94.40	91.83	78.49	82.21
REFRACT - *Ridge*	✓	2	77.65	70.16	74.24	79.63	70.57	94.24	90.27	77.28	79.25
REFRACT	✓	2	92.13	68.34	81.24	86.48	75.93	93.50	93.96	78.15	83.72
REFRACT - *MLP*	×	5	88.89	70.15	79.36	82.91	75.18	94.03	93.90	76.51	82.62
REFRACT - *MLP*	✓	5	90.25	69.68	85.56	88.52	77.66	94.73	95.43	78.79	85.08
REFRACT - *Ridge*	✓	5	87.41	71.07	81.16	83.02	73.76	94.44	93.43	78.14	82.80
REFRACT	✓	5	94.31	72.03	87.09	89.22	78.05	94.22	97.19	79.14	86.41
REFRACT - *MLP*	×	10	90.99	70.56	81.71	83.86	77.48	94.44	95.93	76.91	83.99
REFRACT - *MLP*	✓	10	93.70	70.33	88.36	89.37	78.37	95.23	96.77	78.94	86.38
REFRACT - *Ridge*	✓	10	91.36	72.03	84.19	85.82	75.89	94.61	95.30	78.35	84.69
REFRACT	✓	10	95.64	73.58	89.42	89.97	76.52	94.54	97.58	79.65	87.11
REFRACT - *MLP*	×	20	92.96	72.43	82.48	83.33	77.66	94.24	96.47	76.96	84.57
REFRACT - *MLP*	✓	20	95.68	70.65	89.58	90.37	79.79	95.44	97.33	78.79	87.20
REFRACT - *Ridge*	✓	20	93.21	72.20	87.20	87.72	77.48	94.94	97.00	78.29	86.01
REFRACT	✓	20	96.17	76.31	91.49	91.60	81.23	94.78	98.28	79.88	88.72

Table 12: Comparison of REFRACT predictor variants for T5-base→T5-large transfer.

Model	Linear	κ	MNLI	SICK	RTE	SciTail	QNLI	SNLI	AVG
θ_B *zero-shot*	–	–	50.72	76.22	62.65	77.39	68.39	59.11	65.75
θ_A *fine-tuned*	✓	–	86.40	85.60	73.65	91.55	91.40	91.20	86.63
REFRACT - *MLP*	✓	200	64.52	80.37	61.18	81.78	74.22	72.89	72.49
REFRACT - *MLP*	×	200	61.17	81.67	59.17	81.41	71.65	70.13	70.86
REFRACT - *Ridge*	✓	200	65.50	79.80	61.58	82.61	76.13	73.43	73.17
REFRACT	✓	200	69.17	80.06	69.48	85.85	82.43	79.00	77.67
REFRACT - *MLP*	✓	300	66.20	81.26	61.58	84.30	75.91	74.59	73.97
REFRACT - *MLP*	×	300	63.96	82.81	61.58	83.87	74.22	72.57	73.17
REFRACT - *Ridge*	✓	300	67.46	80.87	61.85	84.59	77.85	74.85	74.58
REFRACT	✓	300	70.91	81.02	69.48	86.39	82.43	80.26	78.42
REFRACT - *MLP*	✓	500	68.59	82.20	62.25	85.22	76.41	75.83	75.08
REFRACT - *MLP*	×	500	67.06	84.74	62.92	85.50	75.00	74.22	74.91
REFRACT - *Ridge*	✓	500	69.69	81.28	63.86	85.85	77.74	76.15	75.76
REFRACT	✓	500	73.52	82.28	71.22	88.06	84.33	81.63	80.17

Table 13: Block smoothing effect from T5-base to T5-large.

Model	κ	MNLI	SICK	RTE	SciTail	QNLI	SNLI	AVG
θ_B *zero-shot*	300	50.72	76.22	62.65	77.39	68.39	59.11	65.75
θ_B *fine-tuned*	–	90.40	93.40	88.80	92.10	94.70	89.40	91.47
θ_A *fine-tuned*	–	86.40	85.60	73.65	91.55	91.40	91.20	86.63
REFRACT	200	69.17	80.06	69.48	85.85	82.43	79.00	**77.67**
REFRACT - $\rho = 0$	200	68.02	79.83	68.94	85.72	82.13	78.24	77.15
REFRACT	300	70.91	81.02	69.48	86.39	82.43	80.26	**78.42**
REFRACT - $\rho = 0$	300	69.67	80.24	69.61	85.65	83.13	79.70	78.00
REFRACT	500	73.52	82.28	71.22	88.06	84.33	81.63	**80.17**
REFRACT - $\rho = 0$	500	72.57	81.13	71.49	87.13	84.19	81.02	79.59

B.4 Block Smoothing

Tab. 13 compares independent block fitting with smoothed residual propagation, using $\rho = 0.9$ as the default in REFRACT. Propagating the smoothed residual to subsequent layers improves performance in nearly all configurations, yielding average gains of 0.4–0.6 percentage points.

B.5 Cross-model transfer of mathematical reasoning

We further evaluate REFRACT on the transfer of mathematical reasoning between causal language models with different scales. The donor is Llama-3.2-3B, fine-tuned in tangent space with NTK-LoRA on `dart-math-uniform`, while the receiver is Llama-3.2-1B-Instruct. We evaluate on the complete GSM8K test set of 1,319 question-answer pairs without applying the receiver's chat template. Unlike the preceding experiments, this evaluation uses one fixed calibration seed. To match donor's functional residuals and reciever's hidden features across differing depths, we partition them into subset of contiguous blocks we term "superblocks". Stage 1 maps the donor's linearized functional residual to a receiver logit correction, and Stage 2 predicts that residual from the receiver's activations using one ridge regressor per superblock. The final configuration partitions the receiver into four superblocks, concatenates the attention-output inputs of all layers within each group, and

Table 14: GSM8K flexible exact-match accuracy at $\alpha = 1$.

Model	Stage 2 predictor	5-shot Accuracy	8-shot Accuracy
Receiver B	–	32.8	36.6
REFRACT - *MLP*	Final hidden state	35.9	36.3
REFRACT - *Ridge*	Final hidden state	37.4	**39.9**
REFRACT	Superblock attention taps	**39.3**	39.8
Source A_{ft}	–	49.3	50.0

Table 15: Inference cost on ViT-B/16 (batch size 128). Latency is also reported as a ratio to the zero-shot model; FLOPs are relative to zero-shot.

Method	**Latency (ms)**	**× zero-shot**	**FLOPs ×**	**Extra memory**
Zero-shot	172.7	1.00	1.000	–
REFRACT - *Ridge*	177.3	1.03	1.000	2 MB
REFRACT - *MLP*	176.9	1.02	1.000	8 MB
REFRACT	190.4	1.10	1.002	257 MB
NTK	1611	9.33	6.0	658 MB

fits the regressors on 1,024 calibration sequences. Calibration rows are restricted to answer tokens, while each sequence includes five in-context examples matching the evaluation format.

As shown in Tab. 14, REFRACT consistently improves the target-only receiver at $\alpha = 1$. In the 5-shot setting, flexible exact match increases from 32.8% to 39.3% ($+6.5$ points), with a corresponding strict score of 39.0%. In the 8-shot setting, accuracy increases from 36.6% to 39.8% ($+3.2$ points). The smaller improvement at 8-shot suggests that the transferred correction partly overlaps with information already supplied by additional in-context demonstrations. Nevertheless, the transferred model remains below the fine-tuned 3B donor, which reaches 49.3% and 50.0% in the 5- and 8-shot settings, respectively, leaving substantial room for improving Stage 2 approximation.

The controls clarify which conclusions can be drawn from these gains. At 5-shot, the global ridge predictor reaches 37.4%, below the block-wise result, whereas the REFRACT - ***MLP*** reaches 35.9%. At 8-shot, however, REFRACT - ***Ridge*** and REFRACT prediction are effectively tied at 39.9% and 39.8%. The experiment therefore establishes that the donor's mathematical capability can be transferred to a smaller receiver without updating its backbone parameters, but it does not yet demonstrate that per-superblock prediction is superior to a single global ridge in this setting. Zero-shot results are not emphasized because GSM8K answer-format mismatches make strict exact match identically zero and render flexible matching sensitive to verbose generations.

B.6 Inference costs.

Table 15 shows that REFRACT adds almost no cost at inference time. All variants keep the FLOPs of the zero-shot model ($\leq 1.002\times$) and add at most 257 MB of memory. At batch size 128, the ***Ridge*** and ***MLP*** variants are within 3% of zero-shot latency. The most accurate variant, REFRACT, is only $1.1\times$ slower than zero-shot. The NTK baseline, instead, needs $6\times$ the FLOPs, is $9.3\times$ slower than zero-shot, and uses $2.5\times$ more extra memory than REFRACT. In absolute terms, REFRACT runs about $8.5\times$ faster than NTK (190.4 vs. 1611 ms).

C ERROR BOUNDS AND THE FEATURE-SPACE OBJECTIVE

Setting. We use the construction of Sec. 3. Assume $\lambda \geq 0$. To shorten expressions locally in this appendix, write $d(x) = D_A(x)$, $D = [d(x_i)]_{i=1}^m$, $T = T_B$, $P = P_B = T^\dagger$, and

$$a = (1+\lambda)^{-1}, \qquad Q = Q_B = TP, \qquad \Pi = D^\dagger D, \qquad \Pi_D = DD^\dagger. \tag{20}$$

The matrices Q, Π, and Π_D are orthogonal projectors onto $\mathrm{col}(T)$, $\mathrm{row}(D)$, and $\mathrm{col}(D)$, respectively, with dimensions $C\times C$, $m\times m$, and $d_A\times d_A$. The desired correction is $e(x) = z_A^{\mathrm{ft}}(x) - z_B^0(x)$, with calibration matrix $E = [e(x_i)]_i$. The fitted quantities are

$$\begin{aligned} R^\star &= aED^\dagger, & u(x) &= PR^\star d(x), \\ c(x) &= Tu(x) = QR^\star d(x), & z_B^{(1)}(x) &= z_B^0(x) + c(x). \end{aligned} \tag{21}$$

These abbreviations agree with the method through $u(x) = \widetilde{D}_B^{(1)}(x)$ and $\widehat{u}(x) = g_\phi(H_B(x))$; hence $z_B^{(2)}(x) = z_B^0(x) + T\widehat{u}(x)$. The matrix D is the calibration matrix denoted D_A in the method. We suppress input arguments where all terms are evaluated at the same input. Unless marked F, matrix norms are spectral norms and vector norms are Euclidean. The statements allow rank-deficient and zero matrices. They concern exact pseudoinverses and fixed linear scores, with any fixed biases included in the base logits. Nonlinear normalization after adding a correction changes these scores; Sec. A.4 specifies this distinction. No result assumes that matching the donor is equivalent to fine-tuning the receiver.

C.1 STAGE 1 SOLUTION AND CALIBRATION ERROR

Proposition C.1 (Minimum-norm transport)**.** *All minimizers of Eq.* (6) *have the form*

$$R = aED^\dagger + G(I - \Pi_D), \qquad G \in \mathbb{R}^{C\times d_A}. \tag{22}$$

The unique minimum-Frobenius-norm minimizer is $R^\star = aED^\dagger$.

Proof. Completing the square gives

$$\|E - RD\|_F^2 + \lambda\|RD\|_F^2 = (1+\lambda)\|RD - aE\|_F^2 + (1-a)\|E\|_F^2. \tag{23}$$

Each row of RD lies in $\mathrm{row}(D)$, so its best fit is $RD = aE\Pi$. The matrix $aED^\dagger$ attains this fit. Any difference between minimizers satisfies $ND = 0$, equivalently $N = N(I - \Pi_D)$, which gives Eq. (22). Finally, $R^\star = R^\star\Pi_D$ is Frobenius-orthogonal to $G(I - \Pi_D)$. Thus adding a nonzero such term strictly increases the squared norm. □

Theorem C.2 (Calibration-error decomposition)**.** *The realized calibration correction is* $\widehat{E} = QR^\star D = aQE\Pi$*, and*

$$\|E - \widehat{E}\|_F^2 = \|(I-Q)E\|_F^2 + \|QE(I-\Pi)\|_F^2 + (1-a)^2\|QE\Pi\|_F^2. \tag{24}$$

Proof. Write $E - aQE\Pi = (I-Q)E + QE(I-\Pi) + (1-a)QE\Pi$. The first term is Frobenius-orthogonal to the others because $(I-Q)Q = 0$. The last two are orthogonal because $\Pi(I-\Pi) = 0$, using the cyclic property of the trace. Taking squared norms proves the identity. □

The terms correspond to corrections outside the head's range, corrections not fittable from the calibration residuals, and output shrinkage. If $Q = I_C$, the first term vanishes. If D has full column rank, $\Pi = I_m$ and the second vanishes.

C.2 RECOVERY AND ERROR OUTSIDE CALIBRATION

Theorem C.3 (Recovery under coverage and realizability)**.** *Suppose* $E = MD$ *for a fixed matrix* M*. At an evaluation input* x*, suppose also* $d(x) \in \mathrm{col}(D)$*,* $e(x) = Md(x)$*, and* $Qe(x) = e(x)$*. Then*

$$z_B^{(1)}(x) = (1-a)z_B^0(x) + az_A^{\mathrm{ft}}(x). \tag{25}$$

In particular, $\lambda = 0$ *gives exact donor logits at that input.*

Proof. Since $R^\star = aM\Pi_D$ and $\Pi_D d(x) = d(x)$, $QR^\star d(x) = aQMd(x) = ae(x)$. Adding $z_B^0(x)$ gives the result. □

Coverage and linear realizability are additional assumptions, not consequences of tangent-space fine-tuning. Spanning an r-dimensional residual space with r independent calibration residuals suffices under these assumptions.

For a bound without realizability, fix any $M \in \mathbb{R}^{C\times d_A}$ and define

$$\xi(x) = e(x) - Md(x), \qquad \Xi = E - MD, \qquad w(x) = D^\dagger d(x). \tag{26}$$

These quantities are deterministic; no zero-mean or independence assumption is used.

Theorem C.4 (Pointwise transport error)**.** *For any input, let*

$$b(x) = (1-a)Md(x) + aM(I - \Pi_D)d(x) + \xi(x) - a\Xi w(x). \tag{27}$$

Then

$$e(x) - QR^\star d(x) = (I-Q)e(x) + Qb(x), \tag{28}$$

$$\|e(x) - QR^\star d(x)\|_2^2 = \|(I-Q)e(x)\|_2^2 + \|Qb(x)\|_2^2, \tag{29}$$

$$\begin{aligned}\|Qb(x)\|_2 \le{}& (1-a)\|QMd(x)\|_2 + a\|QM(I-\Pi_D)d(x)\|_2 \\ &+ \|Q\xi(x)\|_2 + a\|Q\Xi\|_2\|w(x)\|_2.\end{aligned} \tag{30}$$

Proof. Substitute $E = MD + \Xi$ into $R^\star$ to obtain $R^\star = aM\Pi_D + a\Xi D^\dagger$. Hence $e - R^\star d = b$. Adding and subtracting Qe proves Eq. (28). Its two terms lie in orthogonal subspaces, giving Eq. (29). The triangle inequality and $\|Q\Xi w\|_2 \le \|Q\Xi\|_2\|w\|_2$ give the bound. □

The four terms within Qb need not be orthogonal. An exact sum of their squared norms would therefore be unjustified. For $D \neq 0$, its compact SVD gives

$$\|w(x)\|_2 \le \frac{\|\Pi_D d(x)\|_2}{\sigma_{\min}^+(D)}, \tag{31}$$

where $\sigma_{\min}^+$ is the smallest positive singular value. If $D = 0$, $w = 0$ instead. The bound identifies missing residual directions and amplification of calibration mismatch, but gives no numerical test guarantee without controlling these quantities.

C.3 Output shrinkage and coefficient regularization

The Stage 1 penalty changes every inverse singular value $1/\sigma$ to a/σ. It does not preferentially suppress poorly determined directions. For comparison, adding a separate coefficient penalty $\mu\|R\|_F^2$, $\mu > 0$, to Eq. (6) gives

$$R_{\lambda,\mu} = ED^\top\big((1+\lambda)DD^\top + \mu I\big)^{-1}. \tag{32}$$

Differentiating the strictly convex objective proves this formula. If $D = U\Sigma V^\top$ is a compact SVD, the corresponding filter is $g(\sigma) = \sigma/((1+\lambda)\sigma^2 + \mu)$. Since $(1+\lambda)\sigma^2 + \mu \ge 2\sqrt{(1+\lambda)\mu}\,\sigma$, perturbing only E gives

$$\|\Delta R_{\lambda,\mu}\|_F \le \frac{\|\Delta E\|_F}{2\sqrt{(1+\lambda)\mu}}. \tag{33}$$

C.4 Feature targets and their loss

Proposition C.5 (Minimum-norm lift)**.** *For any $v \in \mathbb{R}^C$, Pv is the unique minimum-norm minimizer of $\min_u \|Tu - v\|_2^2$, and $TPv = Qv$. Equivalently, it is the minimum-norm feature displacement with head response Qv.*

Proof. Orthogonal projection gives $\|Tu - v\|_2^2 = \|Tu - Qv\|_2^2 + \|(I-Q)v\|_2^2$. The first term vanishes for $u = Pv$. All minimizers are $Pv + n$ with $n \in \ker T$. Since $Pv \in \mathrm{range}(T^\top)$, it is orthogonal to n, so $\|Pv + n\|_2^2 = \|Pv\|_2^2 + \|n\|_2^2$. □

This selects a canonical target from its logit response. Small nonzero singular values of T can also produce large feature targets.

Proposition C.6 (Feature-loss decomposition)**.** *Let $c \in \mathrm{col}(T)$, $u = Pc$, and let $q \in \mathbb{R}^{d_B}$ be any predicted feature displacement. With $N = I - PT$,*

$$\|q - u\|_2^2 = (Tq - c)^\top (TT^\top)^\dagger (Tq - c) + \|Nq\|_2^2. \tag{34}$$

Proof. Decompose $q - Pc = P(Tq - c) + Nq$. The terms lie in $\mathrm{range}(T^\top)$ and $\ker T$, respectively, and are orthogonal. An SVD of T gives $P^\top P = (TT^\top)^\dagger$, proving the identity. □

For a full-row-rank head, the pseudoinverse on the right is an ordinary inverse. Feature supervision therefore weights attainable logit errors by the inverse squared singular values of the head and penalizes classifier-invisible outputs. For an MLP this changes the training objective relative to ordinary logit MSE.

Proposition C.7 (Equivalence for linear ridge prediction)**.** *Let $H \in \mathbb{R}^{r \times m}$ collect fixed predictor inputs and let $Y \in \mathbb{R}^{C \times m}$ collect unprojected transported logit targets. For the global predictor, $H = [H_B(x_i)]_{i=1}^m$ and $Y = R^\star D$. For an independent block predictor, use $H = [H_{B,\ell}(x_i)]_{i=1}^m$ and $Y = [R^\star D_{A,\ell}(x_i)]_{i=1}^m$. Use feature targets $U = PY$ and attainable logit targets $C_Y = QY$. For the two objectives*

$$\min_W \; m^{-1}\|WH - U\|_F^2 + \beta\|W\|_F^2, \qquad \min_V \; m^{-1}\|VH - C_Y\|_F^2 + \beta\|V\|_F^2, \tag{35}$$

with the same $\beta \geq 0$ and minimum-norm solutions at $\beta = 0$, the fitted maps satisfy $TW^\star = V^\star$ and $W^\star = PV^\star$. Thus their logits agree at every predictor input, not just on calibration data.

Proof. For $\beta > 0$, set $A_H = H^\top(HH^\top + m\beta I)^{-1}$; for $\beta = 0$, set $A_H = H^\dagger$. The normal equations or least-squares pseudoinverse give $W^\star = PYA_H$ and $V^\star = QYA_H$. Now use $TP = Q$ and $PQ = P$. □

The result applies to REFRACT - *Ridge* and each independent block fit with matched inputs, target preprocessing, loss scaling, and regularization. It also extends to fixed-decay block smoothing: if the feature state is P times the logit state before a fit, the same calculation preserves this relation for the fitted outputs and the next state. Induction starts from zero states.

C.5 Stage 2 error and block smoothing

Proposition C.8 (Transport and prediction error)**.** *For any Stage 2 predictor $\widehat{u}(x)$ with $z_B^{(2)}(x) = z_B^0(x) + T\widehat{u}(x)$,*

$$z_A^{\mathrm{ft}} - z_B^{(2)} = (z_A^{\mathrm{ft}} - z_B^{(1)}) + T(u - \widehat{u}), \tag{36}$$

$$\begin{aligned} \|z_A^{\mathrm{ft}} - z_B^{(2)}\|_2 &\leq \|z_A^{\mathrm{ft}} - z_B^{(1)}\|_2 + \|T(u - \widehat{u})\|_2 \\ &\leq \|z_A^{\mathrm{ft}} - z_B^{(1)}\|_2 + \|T\|_2\|u - \widehat{u}\|_2. \end{aligned} \tag{37}$$

If, under any evaluation distribution, $\mathbb{E}\|z_A^{\mathrm{ft}} - z_B^{(1)}\|_2^2 \leq \varepsilon_{\mathrm{tr}}^2$ and $\mathbb{E}\|u - \widehat{u}\|_2^2 \leq \eta^2$, then

$$\mathbb{E}\|z_A^{\mathrm{ft}} - z_B^{(2)}\|_2^2 \leq (\varepsilon_{\mathrm{tr}} + \|T\|_2\eta)^2. \tag{38}$$

Proof. Insert $z_B^{(1)}$ and use $z_B^{(1)} - z_B^{(2)} = T(u - \widehat{u})$. The triangle inequality and the induced matrix norm give the pointwise bounds. The L^2 triangle inequality then gives $(\mathbb{E}\|z_A^{\mathrm{ft}} - z_B^{(2)}\|_2^2)^{1/2} \leq \varepsilon_{\mathrm{tr}} + \|T\|_2\eta$. Squaring proves the claim; no independence between the errors is needed. □

Proposition C.9 (Accumulated block error)**.** *Write $u_\ell(x) = \widetilde{D}_{B,\ell}(x)$ and suppress input arguments. Then $u = \sum_{\ell=1}^L u_\ell$ and $\widehat{u} = \sum_{\ell=1}^L W_\ell^\star H_{B,\ell}$, as in Eqs.* (4)*,* (11) *and* (13)*. Define the error against each* original *block target by $\varepsilon_\ell = u_\ell - W_\ell^\star H_{B,\ell}$. For either independent or smoothed fitting,*

$$u - \widehat{u} = \sum_{\ell=1}^L \varepsilon_\ell, \qquad \|u - \widehat{u}\|_2 \leq \sum_{\ell=1}^L \|\varepsilon_\ell\|_2. \tag{39}$$

At calibration inputs x_i (with arguments again suppressed), the fixed-decay state in Eq. (14) *satisfies*

$$S_\ell = \sum_{j=1}^{\ell} \rho^{\ell-j+1} \varepsilon_j. \tag{40}$$

In particular, $S_L = u - \widehat{u}$ when $\rho = 1$.

Proof. Subtract the two sums to obtain Eq. (39). For the state, induction in $S_\ell = \rho(S_{\ell-1} + \varepsilon_\ell)$, starting from $S_0 = 0$, gives Eq. (40). □

Combining the block identity with the Stage 2 bound gives

$$\|z_A^{\mathrm{ft}} - z_B^{(2)}\|_2 \le \|z_A^{\mathrm{ft}} - z_B^{(1)}\|_2 + \|T\|_2 \sum_{\ell=1}^{L} \|\varepsilon_\ell\|_2. \tag{41}$$

At $\rho = 1$, the target $u_\ell + S_{\ell-1}$ asks block ℓ to fit the remaining error in the partial sum. At $0 < \rho < 1$, it carries a discounted history. The errors ε_ℓ themselves depend on ρ through the fitted maps, so the recurrence does not prove that smoothing reduces the final error. The norm bound can also be loose when block errors cancel.

Proposition C.10 (Agreement with the donor)**.** *For $C \ge 2$, let $y_A = \arg\max_j [z_A^{\mathrm{ft}}]_j$, with fixed tie-breaking, and let $\gamma = [z_A^{\mathrm{ft}}]_{y_A} - \max_{j \ne y_A} [z_A^{\mathrm{ft}}]_j$. Any prediction z satisfies*

$$\|z - z_A^{\mathrm{ft}}\|_\infty < \gamma/2 \quad \Longrightarrow \quad \arg\max_j z_j = y_A. \tag{42}$$

Proof. For every $j \ne y_A$, $z_{y_A} - z_j \ge \gamma - 2\|z - z_A^{\mathrm{ft}}\|_\infty > 0$. □

Since $\|v\|_\infty \le \|v\|_2$, either Stage 1 or Stage 2 Euclidean bound being strictly below $\gamma/2$ is sufficient.

D IMPLEMENTATION DETAILS AND REPRODUCIBILITY CHECKLIST

This section records the implementation choices behind the reported results. The implementation follows standard CLIP ViT and T5 backbone operations. We write $\mathrm{n}(\boldsymbol{v}) = \boldsymbol{v}/\|\boldsymbol{v}\|_2$ for ℓ_2 normalisation and $m = \mathcal{K}C$ for the number of calibration examples of a task with C classes. All experiments were conducted using 3 different seeds, i.e. 33, 54, 89.

D.1 MODELS AND CHECKPOINTS

Vision pairs. All vision encoders are OpenCLIP (Cherti et al., 2023) ViTs, loaded by architecture name and pre-training tag; `open_clip` downloads the pre-trained weights from the Hugging Face Hub. The donor A is `ViT-B-16/datacomp_xl_s13b_b90k` (Gadre et al., 2023): 12 transformer blocks, width 768, $d_A = 512$. The receivers B are (i) `ViT-B-16/laion2b_s34b_b88k` (Schuhmann et al., 2022), which has the donor's architecture and a different pre-training distribution ($d_B = 512$), and (ii) `ViT-L-14/datacomp_xl_s13b_b90k`, which has the donor's pre-training data but 24 transformer blocks, width 1024 and $d_B = 768$. Receivers are only used frozen.

Language pairs. T5-base donor (Raffel et al., 2020) with a T5-large or RoBERTa-base (Liu et al., 2019) receiver, using encoder features for classification. The donor is `google-t5/t5-base` (12 encoder blocks, width 768, $d_A = 768$) and the receivers are `google-t5/t5-large` (24 encoder blocks, width 1024, $d_B = 1024$) and `FacebookAI/roberta-base` (12 blocks, width 768, $d_B = 768$), each with its own tokenizer. Only the encoders are used: the T5 decoder is never built, and RoBERTa is loaded without its pooler. Both are wrapped as an encoder followed by a single linear head (`T5EncoderForSequenceClassification` and `RobertaEncoderForSequenceClassification` in the released code), so that no untrained layer sits between the pooled feature and the logits.

D.2 TEXT INPUT ENCODING AND HYPERPARAMETERS.

Text input encoding. Each example is encoded with the tokenizer of the corresponding model as a sentence pair: premise–hypothesis for MNLI, SNLI, RTE, SciTail, and SICK, and question–sentence for QNLI. We use no task prefix or prompt template, truncate sequences to 256 tokens, and apply dynamic padding. For T5, the resulting sequence is `premise </s> hypothesis </s>`; for RoBERTa, it is `<s> premise </s></s> hypothesis </s>`.

Vision hyperparameters. For all vision experiments, we use the same hyperparameters across tasks, receiver models, calibration budgets $\mathcal{K}$, and seeds. We set $\lambda = 1$ in Eq. (6), $\rho = 0.9$ in Eq. (14), and use trace-scaled Stage 2 ridge regularization with $\bar{\beta} = 0.1$:

$$\lambda_\ell = \beta \frac{\mathrm{tr}\big(H_\ell^\top H_\ell\big)}{m}$$

where $H_\ell = [H_{B,\ell}(\boldsymbol{x}_i)]_{i=1}^m \in \mathbb{R}^{r\ell \times m}$ is the matrix of predictor inputs, r is the input feature dimension and m is the number of calibration examples. Since $\mathrm{tr}\big(H_\ell^\top H_\ell\big) = \|H_\ell\|_F^2$, this scaling makes the regularization invariant to the overall feature scale. No hyperparameter is selected separately for individual runs, and the `REFRACT` correction always uses $\alpha = 1$.

Text hyperparameters. We use the same $\lambda = 1$, $\rho = 0.9$, $\bar{\beta} = 0.1$, and $\alpha = 1$ for every text task, receiver, calibration budget, and seed in the `REFRACT` rows. For `REFRACT` - ***Ridge***, we instead use an unscaled ridge coefficient $\lambda_2 = 1$. The `REFRACT` - ***MLP*** rows use the same fixed configuration across all text experiments.

D.3 REPRESENTATIONS, HEADS AND DEPTH GROUPING

Vision. Each vision encoder uses its own OpenCLIP evaluation transform: bicubic resizing of the shorter side to 224 pixels, a 224×224 centre crop, and CLIP mean/std normalisation. All models are evaluated in fp32 mode.

For source and receiver image encoders ϕ_A and ϕ_B, respectively, the representations used throughout Secs. 2 and 3 are the normalised OpenCLIP image embeddings,

$$f_A(\boldsymbol{x};\theta) = \mathrm{n}(\phi_A(\boldsymbol{x};\theta)) \in \mathbb{R}^{d_A}, \qquad f_B(\boldsymbol{x}) = \mathrm{n}(\phi_B(\boldsymbol{x})) \in \mathbb{R}^{d_B},$$

where ϕ includes the class token, output LayerNorm, and projection. For each class, we construct the classifier from the task-specific CLIP prompt ensemble of Ilharco et al. (2023): text embeddings produce frozen, bias-free heads $T_A \in \mathbb{R}^{C\times d_A}$ and $T_B \in \mathbb{R}^{C\times d_B}$ with unit-norm rows.

For block-wise prediction, $H_{B,j}(\boldsymbol{x})$ is the receiver residual-stream activation at the output of transformer block j, before the LayerNorm of the next block. We average this activation over all tokens, including the class token, and do not normalise it.

The vision donor is partitioned into $L = 13$ groups: its 12 transformer blocks and one output group. The output group is always paired with the receiver representation $f_B(\boldsymbol{x})$, so that $r_{13} = d_B$. For equal-depth transfer (ViT-B/16→ViT-B/16), donor block ℓ is paired with receiver block ℓ, giving $r_\ell = 768$. For ViT-B/16→ViT-L/14, the 24 receiver blocks are partitioned into 12 contiguous pairs. Donor block ℓ is paired with receiver blocks $(2\ell-1, 2\ell)$, and their activations are concatenated:

$$H_{B,\ell}(\boldsymbol{x}) = [H_{B,2\ell-1}(\boldsymbol{x}); H_{B,2\ell}(\boldsymbol{x})], \qquad r_\ell = 2048.$$

Text. Text examples are encoded by each model's own tokenizer and the final encoder feature is obtained by mean-pooling the last-layer hidden states over real tokens only; padding tokens are excluded, while separator tokens are retained. For T5, this feature is read after the encoder's final RMS layer norm; for RoBERTa, it is read from the last encoder layer, without using the pooler. Features are left unnormalised. Thus, $d_A = 768$, while $d_B = 1024$ for T5-large and $d_B = 768$ for RoBERTa-base.

The text donor head is the biased $\texttt{Linear}(d, C_{\mathrm{head}})$ classifier stored in the donor checkpoint, with $C_{\mathrm{head}} = 3$. Only its weight matrix T_A enters Eq. (6); the bias is not used in Stage 1. The receiver head is a nearest-class-mean classifier, constructed once for each task and receiver. Given support examples $\mathcal{S}_c$ for class c, its centroid is

$$\mu_c = \mathrm{n}\left(\frac{1}{|\mathcal{S}_c|}\sum_{x\in\mathcal{S}_c} \mathrm{n}(f_B(x))\right).$$

The rows of T_B are these unit-norm centroids and the receiver bias is zero. Scores are computed as $T_B f_B(\boldsymbol{x})$; because the feature norm is common across classes, this has the same argmax as cosine similarity to the centroids.

For text block-wise prediction, we implement two activation sources. The *residual* source uses the residual-stream output of each block, as in vision. The *attention* source, used in the reported `REFRACT` text experiments, reads the input to the attention output projection (`SelfAttention.o` in T5 and `attention.output.dense` in RoBERTa), after concatenating attention heads. Activations are averaged over non-padding tokens only.

The reported text experiments additionally use segment pooling. For each block, we concatenate the mean activation over premise tokens, hypothesis tokens, and all non-padding tokens:

$$H_{B,j}(\boldsymbol{x}) = [\,\bar{H}^{\mathrm{prem}}_{B,j}; \bar{H}^{\mathrm{hyp}}_{B,j}; \bar{H}^{\mathrm{all}}_{B,j}\,].$$

The two segments are separated at the first `</s>` token for T5 and at the double `</s></s>` separator for RoBERTa; RoBERTa's leading `<s>` belongs to neither segment.

The text donor is likewise partitioned into 12 T5-base encoder blocks and one output group. For T5-base→RoBERTa-base, donor block ℓ is paired with RoBERTa block ℓ, yielding $r_\ell = 768$, or 2304 with segment pooling. For T5-base→T5-large, the 24 receiver blocks are grouped into 12 contiguous pairs. We use the `last` grouping from the released code: donor block ℓ is paired only with the deeper receiver block,

$$H_{B,\ell}(\boldsymbol{x}) = H_{B,2\ell}(\boldsymbol{x}),$$

giving $r_\ell = 1024$, or 3072 with segment pooling. The output group is paired with $f_B(\boldsymbol{x})$.

Code base. Our implementation builds on the publicly available merge-and-rebase code base (`https://github.com/apanariello4/merge-and-rebase`), which we extend with the model pairs, text encoders, block-wise predictors and hyperparameters described in this section. All results reported in this paper are produced with this extended code.