Distilling Linearized Behavior into Non-Linear Fine-Tuning for Effective Task Arithmetic

Thomas Sommariva [1] **Francesca Morandi** [1 2] **Simone Calderara** [1] **Angelo Porrello** [1]

Abstract

Task vector composition has emerged as a promising paradigm for editing pre-trained models, enabling model merging through addition and unlearning through subtraction. Fine-tuning in the tangent space of a pre-trained model (*linear fine-tuning*) has proven effective, as it produces task vectors that are naturally disentangled and resistant to interference. However, linearized models suffer from limited expressivity during training and incur higher computational costs at inference time, which restrict their practical applicability. In this work, we bridge the gap between linear and standard non-linear fine-tuning. We show that linearity with respect to weight perturbations, a property defined in parameter space, can be enforced through constraints in activation space during training. Concretely, we distill hidden representations from a curvature-regularized linearized teacher into a non-linear student trained via conventional fine-tuning. We find that the resulting model inherits key properties of linearized models for task arithmetic, enabling effective composition of task vectors and achieving strong performance across vision and language benchmarks without incurring any inference-time overhead.

1. Introduction

As deep models continue to grow in scale and complexity, retraining from scratch or even fine-tuning them is becoming increasingly impractical. This trend has motivated a growing body of research toward mechanisms that enable the composition (Liu & Soatto, 2023), the modification (Ilharco et al., 2022b; Fierro & Roger, 2026), and reuse (Rinaldi et al., 2025; 2026a;b) of existing models, rather than training new ones. In this context, practitioners can reuse learned capabilities, rapidly customize models, and deploy tailored systems under strict computational or data constraints.

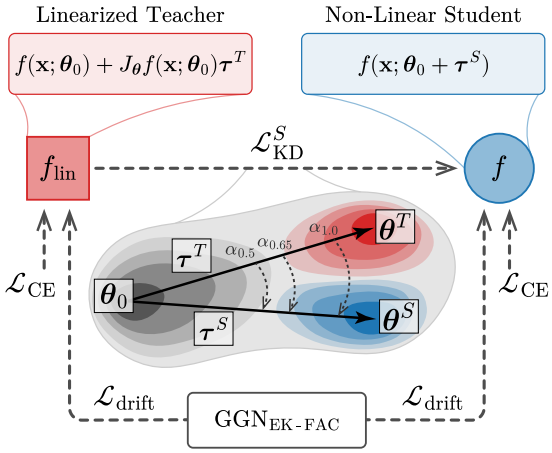

Figure 1. **Overview.** To improve weight disentanglement, a non-linear student model is fine-tuned by distilling a linearized teacher. Both models are trained with curvature-aware regularization based on an approximation of the Generalized Gauss-Newton matrix.

Within this field, task arithmetic (Ilharco et al., 2022a) enables model editing through simple algebraic operations in weight space. Given task-specific models $\{\boldsymbol{\theta}_t = \boldsymbol{\theta}_0 + \boldsymbol{\tau}_t\}_t$ fine-tuned from a common pre-trained model $\boldsymbol{\theta}_0$, the corresponding update vectors $\boldsymbol{\tau}_t$ (**task vectors**) can be composed to create a single multi-task model (*addition*), or to selectively remove task-specific behaviors (*subtraction*).

However, the effectiveness of task arithmetic crucially depends on how task vectors are learned. Recent work has shown that fine-tuning in the tangent space of a pre-trained model, commonly referred to as ***linear fine-tuning*** (Ortiz-Jimenez et al., 2023), yields task vectors that are more naturally disentangled and substantially less prone to interference – a property commonly referred to as ***weight disentanglement***. Moreover, since models in this regime are linear in weight space, edits in parameter space induce predictable changes in the output space. This property enables the use of explicit and efficient regularization penalties (Yoshida et al., 2025; Porrello et al., 2025b) to further promote disentanglement and task compatibility. In contrast, standard

[1]AImageLab, Department of Engineering "Enzo Ferrari", University of Modena and Reggio Emilia, Italy [2]University of Pisa, Italy. Correspondence to: Angelo Porrello <angelo.porrello@unimore.it>.

Proceedings of the 43^{rd} International Conference on Machine Learning, Seoul, South Korea. PMLR 306, 2026.

non-linear fine-tuning does not exhibit these properties; as a result, advanced and tailored post-hoc model merging strategies (Gargiulo et al., 2025; Marczak et al., 2025; Panariello et al., 2025; Buzzega et al., 2025) are required to mitigate interference effects, with mixed results.

While linearization offers several advantages, there is no free lunch. The computational cost of a single forward pass increases significantly, making deployment more expensive (Ortiz-Jimenez et al., 2023). Moreover, constraining optimization to the tangent space of the pre-trained model may limit the expressivity of the model. Given the complementary strengths and weaknesses of linearized and standard non-linear training regimes, a natural question arises:

> Is there a sweet spot in fine-tuning that mitigates the drawbacks of both linearized and non-linear regimes while preserving *weight disentanglement*, *expressiveness*, and *inference efficiency*?

We show that linearity with respect to weight perturbations – a property defined in *parameter space* – can be induced in a conventional non-linear model by imposing tailored learning objectives in *activation space*. Specifically, we show that linearized behavior and weight disentanglement can be distilled (Hinton et al., 2015) by matching the activations of a linearized model. As illustrated in Fig. 1, we distill intermediate activations from a teacher model trained in tangent space, which guides a student model trained in the standard, non-linear fine-tuning regime. This yields task vectors that can be efficiently composed via addition and subtraction within a standard, deployment-friendly non-linear model.

Building on this insight, we propose **DistillEd Linearized Task Arithmetic** (**DELTA**). First, we incorporate curvature-aware regularization (Porrello et al., 2025b;a) to promote disentanglement. Unlike prior work relying on data or statistics from other tasks, we estimate these regularization terms using a third-party **reference dataset**, yielding a task-agnostic training scheme. Second, instead of distilling from a single teacher–student pair, we sample their weights along the linear path connecting the pre-trained weights to their current values during optimization. This **along-path distillation** exposes the student to an ensemble of linearly interpolated teachers, enabling a richer approximation of the linearized dynamics and promoting the transfer of linear behavior.

We empirically show that task arithmetic does not require strict linearization, but rather localized and approximately linear update directions. These properties yield a model that preserves the composability of linear fine-tuning while benefiting from the expressivity of standard training; as a result, the student **outperforms its teacher**. Finally, DELTA achieves strong performance across vision and language benchmarks and can be applied to generative LLM settings.

2. Background

Notation. Let $f : \mathcal{X} \times \Theta \to \mathbb{R}^d$ be a neural network with L layers, mapping inputs $\boldsymbol{x} \in \mathcal{X} \subseteq \mathbb{R}^D$ and weights $\boldsymbol{\theta} \in \Theta \subseteq \mathbb{R}^P$ to an intermediate representation in $\mathbb{R}^d$. The final task output is obtained via an additional linear transformation $\phi : \mathbb{R}^d \to \mathcal{Y} \subseteq \mathbb{R}^C$, with the overall model given by $\phi(f(\boldsymbol{x};\boldsymbol{\theta}))$. We consider a collection of T downstream tasks, where each task t is defined by a triplet $(\mathcal{D}_t, \mu_t, f_t)$: a data support $\mathcal{D}_t \subseteq \mathcal{X}$, an input distribution μ_t with $\text{supp}(\mu_t) = \mathcal{D}_t$, and a target function $f_t : \mathcal{D}_t \to \mathcal{Y}$.

Task vectors. For each task t, the model is fine-tuned on $\mathcal{D}_t$ starting from the pre-trained weights $\boldsymbol{\theta}_0$, yielding task-specific parameters $\boldsymbol{\theta}_t$. The update of the pre-trained model to task t can be represented through the corresponding *task vector* $\boldsymbol{\tau}_t := \boldsymbol{\theta}_t - \boldsymbol{\theta}_0$. *Task arithmetic* (TA) (Ilharco et al., 2022a) posits that these vectors can be combined via simple linear operations in parameter space to edit model functionality. *Task addition* constructs a multi-task model by linearly combining task vectors, resulting in parameters $\boldsymbol{\theta}_0 + \sum_{t=1}^{T} \alpha_t \boldsymbol{\tau}_t$. *Task negation* aims to forget a task by subtracting the corresponding task vector from $\boldsymbol{\theta}_0$.

A property key to task arithmetic is ***weight disentanglement***.

Definition 2.1 (*Weight disentanglement – informal*)**.** A set of task vectors $\{\boldsymbol{\tau}_t\}_{t=1}^{T}$ is disentangled if, for each task t, applying $\boldsymbol{\tau}_t$ induces negligible changes in the predictions for inputs outside the support of task t. In this case, the function f can be decomposed into a sum of spatially localized components vanishing outside a given region.

Linear fine-tuning. Ortiz-Jimenez et al. (2023) empirically demonstrated that **linearized neural networks** exhibit stronger disentanglement than standard non-linear fine-tuning, with improved task arithmetic performance. Formally, a linearized model $f_{\text{lin}}(\boldsymbol{x};\boldsymbol{\theta})$ is defined via a first-order Taylor expansion around pre-trained parameters $\boldsymbol{\theta}_0$:

$$f_{\text{lin}}(\boldsymbol{x};\boldsymbol{\theta}) = f(\boldsymbol{x};\boldsymbol{\theta}_0) + \text{J}_{\boldsymbol{\theta}} f(\boldsymbol{x};\boldsymbol{\theta}_0)(\boldsymbol{\theta} - \boldsymbol{\theta}_0), \quad (1)$$

where $\text{J}_{\boldsymbol{\theta}} f(\boldsymbol{x};\boldsymbol{\theta}_0) \in \mathbb{R}^{d \times P}$ denotes the Jacobian of the model prediction at input $\boldsymbol{x}$ evaluated at $\boldsymbol{\theta}_0$.

Regularization in linearized models. Prior work (Yoshida et al., 2025; Porrello et al., 2025b) showed that even linear fine-tuning admits residual interference between task vectors. A key advantage of the linear regime, however, is that such interference can be analyzed in *closed form*, a property unavailable under standard fine-tuning. In fact, given an example $\boldsymbol{x}$ from the dataset $\mathcal{D}_t$, we can compute the **representation drift**, *i.e.,* the change of the last layer activation when editing the model with the task vector $\boldsymbol{\tau}_{t'}$:

$$\Delta_{t \to t,t'}(\boldsymbol{x}) := \|\boldsymbol{z}_{t,t'} - \boldsymbol{z}_t\|_2^2 \propto \|\text{J}_{\boldsymbol{\theta}} f(\boldsymbol{x};\boldsymbol{\theta}_0)\, \boldsymbol{\tau}_{t'}\|_2^2, \quad (2)$$

where $\boldsymbol{z}_t = f_{\text{lin}}(\boldsymbol{x};\boldsymbol{\theta}_0 + \alpha\boldsymbol{\tau}_t)$ and $\boldsymbol{z}_{t,t'} = f_{\text{lin}}(\boldsymbol{x};\boldsymbol{\theta}_0 + \alpha\boldsymbol{\tau}_t + \alpha\boldsymbol{\tau}_{t'})$ denote the last-layer representations of $\boldsymbol{x}$ before

Table 1. Comparison of training regimes for task arithmetic.

	Non-Linear FT	Linear FT	Distilled (ours)
Task arithmetic	✗	✓	✓
Curvature-aware reg.	✗	✓	✓
Robustness to scaling α	✗	✓	✓
Expressivity	✓	✗	✓
Efficiency (inference)	✓	✗	✓
Efficiency (training)	✓	✗	✗

and after the addition of $\boldsymbol{\tau}_{t'}$, respectively. This analytical characterization of interference is exploited by Yoshida et al. (2025) to introduce a regularizer that explicitly penalizes representation drift in linearized models. While effective, this approach requires direct access to the training data of external tasks, which is often restricted by privacy or storage constraints in decentralized settings.

In this respect, Porrello et al. (2025b) showed that the dependence on external task data can be avoided by leveraging the **generalized Gauss–Newton (GGN) matrix** (Schraudolph, 2003; Martens, 2020), a tool widely used in the curvature-aware optimization literature. Under this lens, for linearized models, the representation drift on examples from an external dataset $\mathcal{D}_t$ has a closed-form quadratic expression:

$$\mathcal{L}_{\text{drift}}(\boldsymbol{\theta}_{t'}) \propto (\boldsymbol{\theta}_{t'} - \boldsymbol{\theta}_0)^\top \boldsymbol{G}_t(\boldsymbol{\theta}_0)(\boldsymbol{\theta}_{t'} - \boldsymbol{\theta}_0). \quad (3)$$

Here, $\boldsymbol{G}_t(\boldsymbol{\theta}_0) = \frac{1}{|\mathcal{D}_t|}\sum_{\boldsymbol{x}\in\mathcal{D}_t} \mathrm{J}_{\boldsymbol{\theta}} f(\boldsymbol{x};\boldsymbol{\theta}_0)^\top \mathrm{J}_{\boldsymbol{\theta}} f(\boldsymbol{x};\boldsymbol{\theta}_0)$ is the GGN matrix computed on the dataset $\mathcal{D}_t$, which, once pre-computed, does not require further access to task data. Since the full GGN matrix is intractable – it scales quadratically with the number of parameters – Porrello et al. (2025b) resort to *Kronecker-Factored Approximate Curvature* (KFAC) (Martens & Grosse, 2015) – for a tutorial covering both theory and implementation, see Dangel et al. (2025). KFAC approximates the GGN with a block-diagonal structure, where each layer l is represented as a Kronecker product $\boldsymbol{A}^l \otimes \boldsymbol{G}^l$, with $\boldsymbol{A}^l$ and $\boldsymbol{G}^l$ denoting the Gram matrices of the input activations and output gradients, respectively. Notably, incorporating the KFAC approximation into Eq. (3) enables dataless optimization, as it requires models to share only the Kronecker factors $\boldsymbol{A}^l, \boldsymbol{G}^l$ rather than raw data.

2.1. Discussion and Limitations

To sum up, linearized models offer several **advantages** for task arithmetic. In particular, they naturally yield task vectors with improved *weight disentanglement* (Ortiz-Jimenez et al., 2023) and admit an exact, closed-form characterization of task interference (Yoshida et al., 2025). This latter property enables the design of dataless regularizers that minimize *representation drift* through curvature-aware approximations. Finally, linearized models have also been observed (Porrello et al., 2025b) to be more robust to the choice of scaling coefficients $\{\alpha_t\}_{t=1}^T$, a property that may facilitate deployment without requiring extensive tuning of scaling parameters on a held-out validation set.

Despite these advantages, the linear regime also presents notable **limitations**. Its reliance on Jacobian-vector products (Eq. (1)) incurs substantial overhead, doubling the cost during both training and inference (see Sec. A). Second, constraining the model to remain on the tangent plane around the pre-trained parameters limits expressivity, potentially leading to inferior performance on individual tasks.

Taken together, the drawbacks of the linear regime summarized in Tab. 1 could hinder its practical deployment in many settings. Hence, our work positions itself in this direction: retaining the inference-time efficiency and flexibility of non-linear fine-tuning, while inducing learning directions compatible with task arithmetic and model merging.

3. Proposed Method: `DELTA`

Building on the discussion in Sec. 2.1, we seek a training strategy that operates in the standard non-linear setting while retaining the favorable properties of linearized models. To this end, we rely on knowledge distillation (Hinton et al., 2015) and propose **DistillEd Linearized Task Arithmetic** (**`DELTA`**)[1]: we train a model via conventional fine-tuning while encouraging it to match the activations of a curvature-regularized *linearized model*, which serves as its teacher.

The key hypothesis is that mimicking the activations of such a teacher biases optimization toward solutions in parameter space that exhibit similar behavior in the student, including linearity to weight perturbations and enhanced weight disentanglement. This is corroborated by Fig. 2: compared with conventional non-linear fine-tuning, ours greatly reduces the **disentanglement error** (Ortiz-Jimenez et al., 2023)

$$\xi(\alpha_1,\alpha_2) = \sum_{t=1}^{2} \mathbb{E}_{\boldsymbol{x}\sim\mu_t}\Big[\operatorname{dist}\big(\phi(f(\boldsymbol{x};\boldsymbol{\theta}_0+\alpha_t\boldsymbol{\tau}_t)), \phi(f(\boldsymbol{x};\boldsymbol{\theta}_0+\alpha_1\boldsymbol{\tau}_1+\alpha_2\boldsymbol{\tau}_2))\big)\Big], \quad (4)$$

i.e., the discrepancy between the predictions of merged and individual models, with $\operatorname{dist}(y_1,y_2) = \mathbb{1}\{y_1 \neq y_2\}$.

3.1. Teacher–Student Training Setup

For each task $t = 1,\dots,T$, we consider two models built upon the **same pre-trained initialization** $\boldsymbol{\theta}_0$: a non-linear student $f(\boldsymbol{x};\boldsymbol{\theta}_t^S)$ and a linearized teacher $f_{\text{lin}}(\boldsymbol{x};\boldsymbol{\theta}_t^T)$. The teacher corresponds to the first-order linearization of the model around $\boldsymbol{\theta}_0$ (see Eq. (1)), while the student operates in the standard non-linear regime. We define the corresponding task vectors as $\boldsymbol{\tau}_t^S = \boldsymbol{\theta}_t^S - \boldsymbol{\theta}_0$ and $\boldsymbol{\tau}_t^T = \boldsymbol{\theta}_t^T - \boldsymbol{\theta}_0$.

[1] https://github.com/apanariello4/merge-and-rebase.

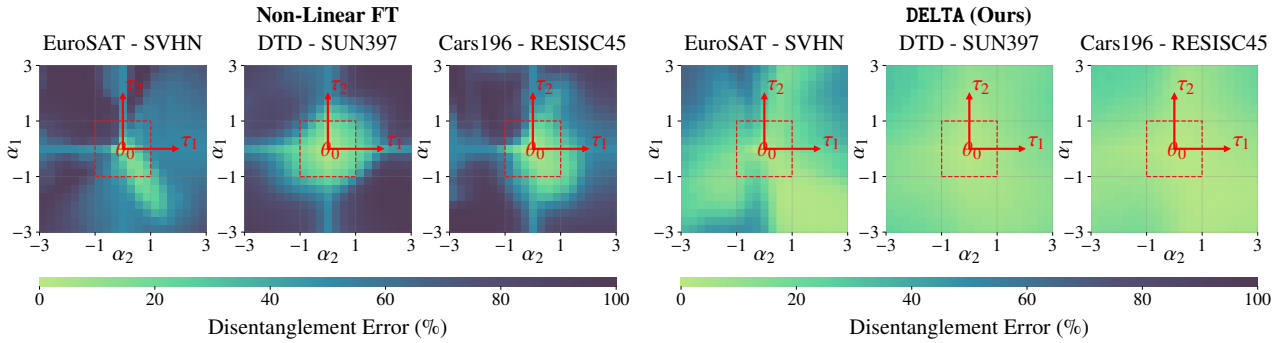

Figure 2. The heatmaps show the disentanglement error (Ortiz-Jimenez et al., 2023) of a non-linear CLIP ViT-B/32 (left) and the non-linear distilled student (right) on several task pairs. The light regions denote areas of the weight space where weight disentanglement is stronger.

Online feature-level distillation. The two models are trained jointly in an **online fashion**, eliminating the need for separate training stages and yielding a single, unified optimization process. Furthermore, distillation is enforced directly in the **feature space**: concretely, we align the activations of the last hidden layer before the final projection head, encouraging the non-linear student to match the linear representations produced by the teacher. As a distillation criterion, we adopt a mean squared error (MSE) loss between teacher and student features, which will later be generalized to a novel proposed along-path distillation objective.

In the following sections, we detail the loss functions applied to the teacher (Sec. 3.2) and the student (Sec. 3.3).

3.2. Training the Linearized Teacher

As in standard distillation frameworks, the linearized teacher is first trained to solve the task under consideration. Accordingly, its most immediate learning signal is the task loss itself (*e.g.,* cross-entropy for classification). However, to act as an effective teacher in the context of task arithmetic, we promote **weight disentanglement** through curvature-aware regularization. As discussed in Sec. 2, operating in the linear regime enables explicit, closed-form control of representation drift, thereby encouraging task-specific update directions that are well separated and suited for task arithmetic. We can thus summarize the twofold objective of the teacher as a composite loss $\mathcal{L}_t^T(\mathcal{X}; \boldsymbol{\theta}_t^T)$, defined as:

Teacher loss

$$\underbrace{\mathcal{L}_{\text{CE}}\left(\mathcal{X};\, \boldsymbol{\theta}_t^T\right)}_{\textbf{Task loss}} + \beta^T \underbrace{\mathcal{L}_{\text{drift}}\left(\boldsymbol{\theta}_t^T\right)}_{\textbf{Drift Eq. (6)}} \tag{5}$$

where $\mathcal{X} := \{(\boldsymbol{x}_i, f_t^\star(\boldsymbol{x}_i))\}_{i=1}^B$ denotes a batch of B examples sampled from task t and β^T is a hyperparameter.

Curvature-Aware weight disentanglement. As discussed in Sec. 2, prior work (Porrello et al., 2025b) minimizes representation drift (Yoshida et al., 2025) via a KFAC approximation of the generalized Gauss-Newton (GGN) matrix. This strategy, however, requires access *during training* to the KFAC factors of all other tasks to be merged. As a consequence, it assumes that the full set of tasks is known *a priori*, which constitutes a severe limitation in realistic settings. Moreover, when a new task is introduced after training, all previously learned task vectors must be retrained to restore disentanglement with respect to the newly introduced task.

To overcome this limitation, we aim to promote disentanglement in a more task-agnostic manner, producing task vectors that are disentangled with respect to *any* potential input distribution rather than a fixed and predefined set of tasks. To this end, we hypothesize that disentanglement can be achieved by regularizing on a proxy dataset that is sufficiently large and diverse to approximate the underlying data manifold. Concretely, we pre-compute a single, shared curvature matrix on a ***reference dataset*** $\mathcal{D}_\Omega$. This dataset is intended to capture a broad range of input distributions spanning many possible downstream tasks. Specifically, for vision tasks we estimate curvature on ImageNet-21k (Deng et al., 2009; Ridnik et al., 2021), using a randomly sampled 15% subset of the original training set. For textual tasks, we instead rely on the C4 corpus (Raffel et al., 2020), employing 10^5 randomly sampled examples[2].

By regularizing each task vector against the curvature matrix derived from the reference dataset $\mathcal{D}_\Omega$, we encourage updates that avoid parameter directions likely to be relevant for other tasks. To obtain a precise estimate of the curvature induced by $\mathcal{D}_\Omega$, we adopt the Eigenvalue-Corrected Kronecker-Factored Approximate Curvature (EK-FAC) (George et al., 2018), which provides a more accurate approximation of the GGN than standard KFAC. While KFAC approximates the GGN as a Kronecker product of two second-moment matrices, $\boldsymbol{A}^l \otimes \boldsymbol{G}^l$, EK-FAC further refines this approximation by explicitly modeling the eigenvalues of the curvature in the Kronecker-factored eigenbasis.

[2]See Sec. C in the appendix for an ablation on the sensitivity to the choice of the reference dataset $\mathcal{D}_\Omega$.

With EK-FAC factors pre-computed on $\mathcal{D}_\Omega$, the representation drift for any task vector $\boldsymbol{\tau}$ can be minimized via

$$\mathcal{L}_{\text{drift}}(\boldsymbol{\theta}_t) = \sum_{l=1}^{L} (\boldsymbol{\theta}_t^l - \boldsymbol{\theta}_0^l)^\top \text{GGN}^l_{\text{EK-FAC}} (\boldsymbol{\theta}_t^l - \boldsymbol{\theta}_0^l), \quad (6)$$

where $\text{GGN}^l_{\text{EK-FAC}} = (U_A^l \otimes U_G^l)\, S^l\, (U_A^l \otimes U_G^l)^\top$ denotes the EK-FAC approximation of the GGN; here, U_A^l and U_G^l are the Kronecker-factored eigenbases and S^l the diagonal matrix of corrected eigenvalues.

3.3. Training the Non-Linear Student

We train the student model $f(\boldsymbol{x}; \boldsymbol{\theta}_t^S)$ in the conventional non-linear fine-tuning regime. As detailed in Sec. 3.1, transfer is performed in an online fashion, with the two models learning simultaneously. The resulting student loss is:

Student loss

$$\underbrace{\mathcal{L}_{\text{CE}}(\mathcal{X};\, \boldsymbol{\theta}_t^S)}_{\text{Task loss}} + \beta^S \underbrace{\mathcal{L}_{\text{drift}}(\boldsymbol{\theta}_t^S)}_{\text{Drift Eq. (6)}} + \gamma \underbrace{\mathcal{L}_{\text{KD}}(\mathcal{X}; \boldsymbol{\theta}_t^S)}_{\text{Transf. Eq. (8)}} \quad (7)$$

The last term corresponds to a tailored modification of the standard MSE-based distillation objective, which transfers intermediate representations along the teacher's trajectory (see next paragraph). Finally, we emphasize that feature-level distillation constrains the student to operate close to a linear regime. In this setting, curvature-aware regularization is well defined; accordingly, we incorporate an EK-FAC-based penalty into the student loss in Eq. (7), *i.e.,* $\mathcal{L}_{\text{drift}}(\boldsymbol{\theta}_t^S)$, computed from the GGN on the reference dataset $\mathcal{D}_\Omega$, to further promote weight disentanglement.

Along-Path Knowledge Distillation. Rather than distilling a single teacher model $f_{\text{lin}}(\boldsymbol{x}; \boldsymbol{\theta}_t^T)$, we further exploit its linear structure and perform distillation over a continuum of teacher models, each obtained by interpolating the task vector along the linear path originating from the origin $\boldsymbol{\theta}_0$. The Along-Path Knowledge Distillation (**APKD**) loss is:

$$\mathcal{L}_{\text{KD}}(\mathcal{X}; \boldsymbol{\theta}_t^S) = \mathbb{E}_{\alpha \sim \mathcal{U}(0.5,1)} \left[\frac{1}{B} \sum_{i=1}^{B} \left\| f(\boldsymbol{x}_i; \boldsymbol{\theta}_0 + \alpha\, \boldsymbol{\tau}_t^S) - \texttt{SG}\left[f_{\text{lin}}(\boldsymbol{x}_i; \boldsymbol{\theta}_0 + \alpha\, \boldsymbol{\tau}_t^T) \right] \right\|_2^2 \right]. \quad (8)$$

In practice, we approximate the expectation by sampling a single $\alpha \sim \mathcal{U}(0.5, 1)$ per optimization step and using the corresponding teacher $f_{\text{lin}}(\boldsymbol{x}; \boldsymbol{\theta}_0 + \alpha\, \boldsymbol{\tau}_t^T)$. Also, in Eq. (8), gradients are prevented from propagating through the teacher via the stop-gradient operator $\texttt{SG}[\cdot]$. This along-path distillation objective encourages the student to inherit the inductive bias of the linearized model not at a single point, but along the linear path between $\boldsymbol{\theta}_0$ and $\boldsymbol{\theta}_t^T = \boldsymbol{\theta}_0 + \boldsymbol{\tau}_t^T$, yielding representations robust to rescaling.

4. Experiments

Vision tasks. We evaluate our method on two multi-task image classification benchmarks. First, we test `DELTA` on the standard **`8-Vision`** benchmark (Ilharco et al., 2022a), which comprises eight heterogeneous visual classification tasks. To assess scalability to larger task pools, we further evaluate on the **`14-Vision`** benchmark (Gargiulo et al., 2025), which extends the former with six additional vision tasks, substantially increasing task diversity and difficulty.

Language tasks. Following (Porrello et al., 2025b), we evaluate our framework on the **`6-NLI`** benchmark (Stoica et al., 2025), which comprises six Natural Language Inference datasets spanning diverse linguistic domains.

Backbones. For vision experiments, we use two variants of CLIP (Radford et al., 2021), employing ViT-B/32 and ViT-L/14 as visual encoders. For each task, we fine-tune the visual encoder while keeping the text encoder frozen. For language experiments, we adopt the T5-base model (Raffel et al., 2020) as backbone for all `6-NLI` tasks. Unless otherwise specified, all methods – both teacher and student in our case – employ full fine-tuning as the learning strategy.

Metrics. Following the original setup of (Ortiz-Jimenez et al., 2023), we employ absolute and normalized accuracy. We further analyze the role of the rescaling coefficient $\boldsymbol{\alpha}$: *(i)* fixing $\alpha_t = \alpha = 1$ for all tasks, *i.e.,* plain summation of task vectors, and *(ii)* tuning α on a cross-task validation set.

4.1. Comparison with the state-of-the-art

We compare against a broad set of existing methods, including both **in-training** approaches that mitigate task interference during optimization and **post-hoc** strategies that operate solely at merging time (after training).

Task Addition. Considering in-training approaches, we first compare against *Non-Linear Fine-Tuning* (Ilharco et al., 2022a) and *Linear Fine-Tuning* (Ortiz-Jimenez et al., 2023), in which task-specific models are optimized independently. Within the class of non-linear methods, we evaluate *TaLoS* (Iurada et al., 2025) and *Attention-Only Fine-Tuning* (Jin et al., 2025). The former identifies and updates a sparse subset of parameters with low Fisher sensitivity, while the latter fine-tunes only attention layers. Within the class of linearized models, we compare against $\boldsymbol{\tau}Jp$ (Yoshida et al., 2025), a data-dependent regularization method that minimizes representation drift. Finally, we consider TAK (Porrello et al., 2025b), a curvature-aware regularization method based on KFAC that reformulates the penalty of $\boldsymbol{\tau}Jp$ to avoid reliance on data from other tasks.

In Tab. 2, we compare our method with state-of-the-art in-training approaches for **task addition**. Notably, in terms of absolute accuracy, `DELTA` consistently achieves strong per-

Table 2. **Task Addition.** Performance comparison against in-training merging methods. *Abs.* denotes the absolute accuracy, while *Norm.* represents accuracy normalized by the accuracies of individually fine-tuned models.

Method	α	8-Vision				14-Vision		6-NLI	
		ViT-B/32		ViT-L/14		ViT-B/32		T5-Base	
		Abs.	Norm.	Abs.	Norm.	Abs.	Norm.	Abs.	Norm.
Pre-trained	—	48.4	—	65.0	—	57.8	—	61.7	—
Individual	—	92.8	—	95.8	—	90.2	—	85.9	—
Linearized models									
Linear FT (Ortiz-Jimenez et al., 2023)	1.0	77.4	88.0	88.0	94.8	73.7	83.4	76.0	92.9
	Best	78.9	89.8	88.0	94.8	76.7	87.0	76.4	93.5
τJp (Yoshida et al., 2025)	1.0	85.0	97.4	90.9	98.3	85.3	97.0	82.5	100.0
	Best	85.6	98.2	91.1	98.5	85.4	**97.1**	**82.5**	**100.0**
TAK (Porrello et al., 2025b)	1.0	86.0	97.7	91.6	99.3	84.3	95.6	79.1	98.4
	Best	86.1	97.8	91.6	99.3	84.7	96.0	79.5	98.8
Non-Linear models									
Non-Linear FT (Ilharco et al., 2022a)	1.0	32.0	32.9	45.3	47.5	15.6	16.6	42.0	49.7
	Best	73.5	80.4	84.5	89.7	68.9	76.1	78.2	91.6
TaLoS (Iurada et al., 2025)	1.0	53.3	59.7	46.1	50.8	33.5	37.3	61.7	72.4
	Best	77.9	87.7	84.7	91.1	74.9	84.4	76.7	89.8
Attn. Only FT (Jin et al., 2025)	1.0	22.5	23.3	66.2	69.7	13.8	15.1	51.6	61.6
	Best	78.2	86.3	88.2	93.8	73.4	81.5	76.7	91.4
DELTA (ours)	1.0	88.3	98.3	92.7	99.5	85.9	95.9	82.3	95.5
	Best	**88.3**	**98.3**	**92.7**	**99.5**	**86.0**	96.0	82.4	95.5

formance, outperforming all competing approaches across settings, while remaining competitive in terms of normalized accuracy. The comparison with **non-linear methods** is particularly revealing: despite our model being fine-tuned in a regime that is, in principle, non-linear, we observe substantial gains over standard non-linear fine-tuning, with improvements of up to approximately $+15$ absolute accuracy on ViT-B/32 in the 8-Vision setting. This result highlights the impact of distillation from the linear regime, an aspect we will further dissect in the following.

The gap with **linearized methods** narrows, yet our approach remains superior. Notably, this advantage comes with improved inference-time efficiency and greater flexibility, as our method does not require access to task-specific data such as τJp (Yoshida et al., 2025) nor KFAC statistics (Porrello et al., 2025b) from other tasks.

Task Negation. We adopt the protocol introduced in (Ilharco et al., 2022a) to forget a target task by subtracting its task vector, and evaluate performance on both the target and a general control task. As shown in Tab. 3, on the 8-Vision benchmark with ViT-B/32, standard non-linear fine-tuning starts from a target accuracy of 20.4% and a control accuracy of 60.5%. In contrast, DELTA reduces the target accuracy to 9.6% while maintaining a high control accuracy of 62.1% – outperforming both Attn. Only FT and TaLoS. Notably, our method is outperformed by TAK and τJp, which rely on the linear regime, suggesting that this regime retains residual disentanglement that is more effectively exploited in task subtraction than in addition.

Table 3. **Task Negation.** Unlearning performance on target versus control (ImageNet-1K) tasks.

Method	8-Vision				14-Vision	
	ViT-B/32		ViT-L/14		ViT-B/32	
	Targ.	Cont.	Targ.	Cont.	Targ.	Cont.
Pre-trained	48.4	63.3	65.0	75.5	57.8	63.3
Linear FT	9.3	60.5	7.1	72.1	19.9	60.6
τJp	3.0	60.6	**1.8**	74.2	**4.5**	60.8
TAK	**2.8**	61.5	3.2	73.6	5.6	60.9
Non-Linear FT	20.4	60.5	18.1	72.3	30.5	60.6
TaLoS	18.4	61.1	23.0	74.1	27.3	61.1
Attn. Only FT	16.7	60.8	17.9	73.3	27.2	60.9
DELTA (ours)	9.6	**62.1**	11.7	**74.7**	19.1	**62.1**

Table 4. **LoRA-based Task Addition.** Results using LoRA re-parameterization with different merging techniques.

Method	8-Vision				14-Vision	
	ViT-B/32		ViT-L/14		ViT-B/32	
	Abs.	Norm.	Abs.	Norm.	Abs.	Norm.
Non-Linear FT	72.6	79.5	85.3	90.5	68.4	75.7
Linear FT	75.9	86.2	87.1	94.0	75.5	85.5
Iso-C	70.6	77.9	85.3	90.7	71.9	80.0
TSV-M	76.4	83.9	88.9	94.4	74.3	82.1
Core + Iso-C	73.6	81.2	87.4	93.0	72.3	80.0
Core + TSV-M	77.9	85.6	89.1	94.6	74.5	82.5
DELTA (ours)	**87.5**	**97.9**	**92.2**	**99.1**	**85.7**	**96.3**

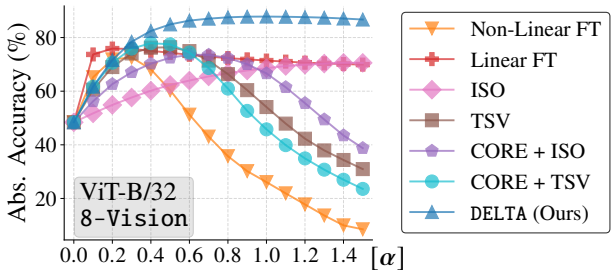

Figure 3. Sensitivity to the scaling coefficient in LoRA merging. Additional results are provided in Fig. 12 of Sec. D.

LoRA-based Task Addition. Our approach supports different parameterizations for the teacher and the student. In particular, the teacher can be trained via full fine-tuning, while the student relies on parameter-efficient methods such as LoRA (Hu et al., 2022). This design allows the teacher to explore richer task-specific directions in weight space, while constraining the student to a low-rank, efficient subspace.

To further examine this, we compare the DELTA LoRA student with three state-of-the-art model merging methods: Iso-C (Marczak et al., 2025), TSV-M (Gargiulo et al., 2025), and Core Space (Panariello et al., 2025). We remark that these approaches operate **post hoc**, after standard fine-tuning, and are thus conceptually different from ours, which acts **during training** and directly produces disentangled task vectors. Nevertheless, this comparison helps position DELTA relative to widely used merging techniques.

As shown in Tab. 4, our method outperforms existing approaches by a large margin. Moreover, the α-sweep sensitivity analysis in Fig. 3 shows that, unlike most baselines (except linear fine-tuning), our model is robust to rescaling of the coefficient α used to scale the merged task vector. This is particularly desirable when α cannot be tuned, *e.g.,* in the absence of a validation set. Taken together, these results suggest that in-training regularization can play a pivotal role in enabling effective fusion of low-rank updates.

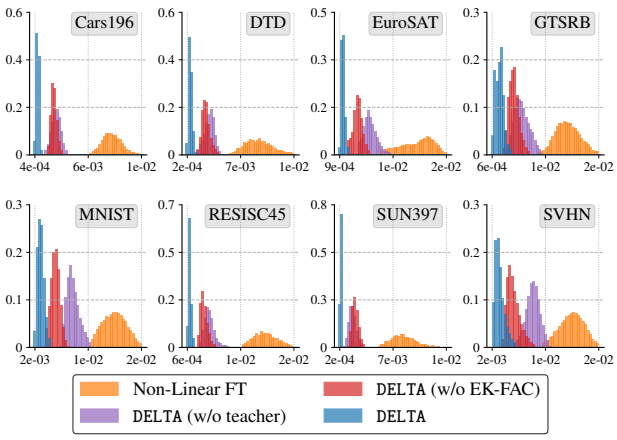

Figure 4. Histogram of the linearization error on 8-Vision.

5. Model Analysis

Sec. 4.1 showed that DELTA outperforms conventional fine-tuning on Task Addition and Negation. Here, we investigate the reasons behind these gains. We focus on two properties learned by the student: the transfer of *linearized behavior* and the emergence of *support localization*. We then assess their relative contributions to task arithmetic performance.

Linearization error. We verify whether the student – despite operating in a non-linear regime – exhibits linear behavior under weight perturbations. To this end, we define the *linearization error* on a single example $\boldsymbol{x}$ as the discrepancy between the average activation of perturbed models and the activation at the mean weights $\bar{\boldsymbol{\theta}}^S = \frac{1}{T}\sum_{t=1}^{T} \boldsymbol{\theta}_t^S$:

$$\xi_{\text{lin}}(\boldsymbol{x}) = \left\| \frac{1}{T}\sum_{t=1}^{T} f(\boldsymbol{x};\boldsymbol{\theta}_t^S) - f(\boldsymbol{x};\bar{\boldsymbol{\theta}}^S) \right\|_1 . \quad (9)$$

This error is exactly zero in the linear regime; thus, a low error indicates an effective transfer of linear behavior.

In Fig. 4, we report the distribution of this error across each dataset of 8-Vision. We compare: *(i)* standard non-linear fine-tuning, *(ii)* our student model without teacher distillation, *(iii)* our student model with distillation but without curvature regularization, and *(iv)* our full method DELTA. The histograms show that our full method achieves **near-zero linearization error**, substantially reducing the discrepancy relative to the non-linear baseline. Moreover, teacher distillation is the primary driver of this effect: in terms of linearization error, removing distillation leads to a larger degradation than removing curvature regularization.

Support localization. Ortiz-Jimenez et al. (2023) showed that strict linearity is not required for task arithmetic. Instead, the key condition is *weight disentanglement* (see Sec. 2), whereby a task-specific update primarily affects predictions on its own data distribution while leaving out-of-domain representations largely unchanged.

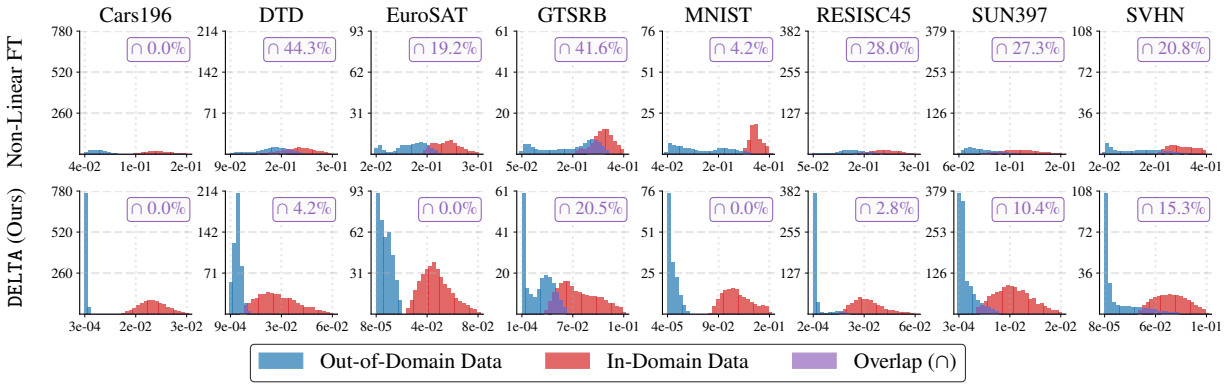

Figure 5. **Support Localization.** Histograms of the activation MSE between the pre-trained and individually fine-tuned models. In our method, the in-domain and out-of-domain distributions are markedly more separated, indicating stronger weight disentanglement.

To assess support localization, for each model t we measure the *edit distance*, *i.e.,* the mean squared error (MSE) between pre- and post-fine-tuning activations, $\frac{1}{d}\|f(\boldsymbol{x};\boldsymbol{\theta}_t^S) - f(\boldsymbol{x};\boldsymbol{\theta}_0)\|_2^2$, on both in-domain training examples and out-of-domain examples drawn from other tasks. As shown in Fig. 5, our student exhibits **stronger support localization** than conventional fine-tuning: in DELTA, the edit distance is higher for in-domain examples while remaining near zero on out-of-domain data, indicating a reversion to the pre-trained model for those examples. Importantly, results in Sec. D (Fig. 16) show that curvature regularization, rather than distillation, is the primary driver of such localization.

With these results, we show that the student converges to regions of parameter space that behave linearly and are task-localized. This behavior is induced by an objective defined in *function space*, which, perhaps surprisingly, leads to tangible effects in *parameter space*. We conjecture that this transfer from function space to parameter space arises because the objective keeps optimization near the pre-trained model, where a first-order Taylor approximation remains accurate. Furthermore, consistent with the simplicity-bias perspective (Huh et al., 2024), distilling a linearized teacher may drive optimization toward the simplest mechanism that fits this behavior – *i.e.,* solutions with approximately linear responses to parameter perturbations.

Impact on task arithmetic. We analyze the relative contributions of the two aforementioned properties to task arithmetic. To this end, Fig. 6 reports per-task performance of the merged model under varying configurations. As shown, distillation from the linear regime alone yields a substantial improvement over conventional fine-tuning. This highlights a possible, more practical variant of our approach that requires no curvature estimation while still delivering meaningful gains. Similarly, applying curvature regularization alone yields performance closest to our full method (though still inferior), confirming that enforcing localized support is the **most critical condition** for task arithmetic.

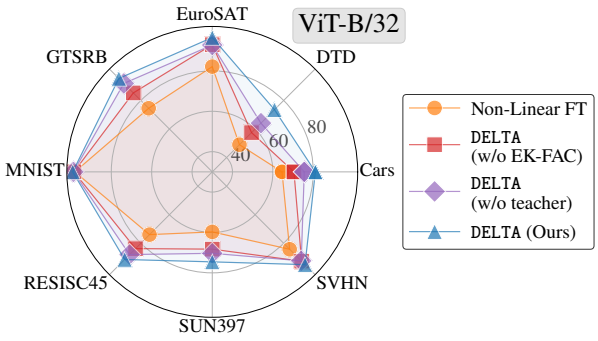

Figure 6. **Impact on task arithmetic**. Per-task accuracy of the merged model under different configurations. Similar trends persist for ViT-L/14 and T5-Base, as shown in Fig. 13 of Sec. D.

On Along-Path Knowledge Distillation (APKD). Unlike conventional KD, our method distills knowledge from multiple models (Sec. 3.3), exposing the student to a continuum of linearized, interpolated teachers, and promoting the transfer of linear dynamics. To validate this, Fig. 7 reports the linearization error when the teacher is fixed at $\alpha = 1$ (*i.e.,* without along-path sampling). As shown, fixing the teacher increases this error across datasets, supporting our claims. Furthermore, we evaluate how teacher sampling influences performance via an α-sweep analysis, reporting in Fig. 8 the accuracy of the merged model as a function of the rescaling coefficient α. Especially on T5, removing teacher sampling leads to significantly less robust performance across α values, highlighting the key role of APKD in transferring the robustness of the linear regime to task vector rescaling.

On expressivity. A somewhat surprising finding is that the student consistently surpasses its linearized teacher in average accuracy after merging (see Fig. 9, *left*). We attribute this to the greater expressivity of the non-linear regime, which allows the student to better fit individual tasks than its teacher. This is corroborated by Fig. 9 (*right*), which reports per-task performance **before merging** for T5 (additional results are provided in Fig. 14 of Sec. D): the student out-

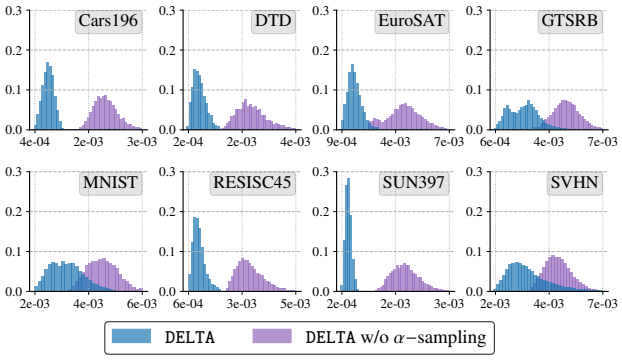

Figure 7. Effect of APKD on linearization error.

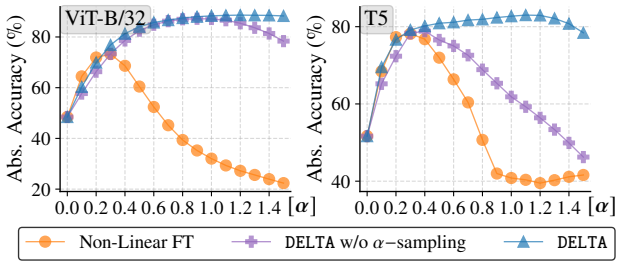

Figure 8. Effect of APKD on robustness to α-sweep.

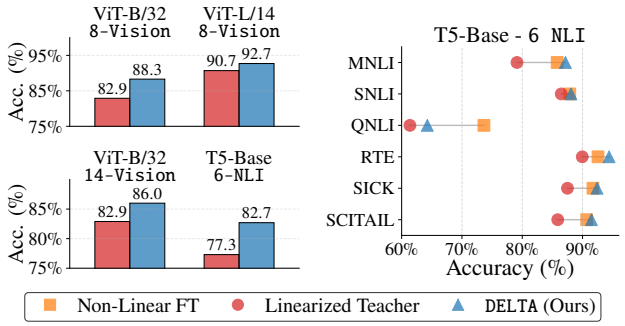

Figure 9. **Teacher-Student comparison.** Left: Accuracy of the merged models. Right: Accuracy of the single task fine-tunings.

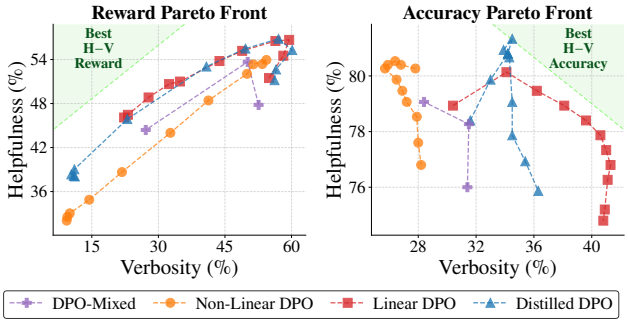

Figure 10. **Multi-Objective Alignment in Generative LLMs.** Pareto frontiers of the trade-off between helpfulness and verbosity.

performs the linearized teacher on all individual tasks. This suggests that DELTA does not merely replicate linearized behavior; rather, the student operates in an intermediate, quasi-linear regime that combines the capacity of the non-linear regime with the disentanglement of the linear one.

Extension to Generative LLMs. Following Erdogan (2026), we go beyond classification and investigate whether task arithmetic can balance multiple preference dimensions in LLMs – such as *helpfulness* and *verbosity* – in a controllable manner. Unlike methods such as Direct Preference Optimization (DPO) (Rafailov et al., 2023), which collapse multiple axes into a single scalar objective, task arithmetic enables training models specialized along each preference axis and combining them at inference time. This allows trading off between objectives without further optimization.

We adopt LLaMA-3.2-1B-Instruct (Grattafiori et al., 2024) and fine-tune two separate models using DPO on UltraFeedback (Cui et al., 2024) and HelpSteer2 (Wang et al., 2024) for helpfulness and verbosity, respectively. The resulting models are merged using the *Affine-2* formulation (Erdogan, 2026): $\boldsymbol{\theta}_{\text{mix}} = \boldsymbol{\theta}_0 + \boldsymbol{\tau}_{\text{help}} + \lambda_2 \boldsymbol{\tau}_{\text{verb}}$. We fix the helpfulness direction and sweep the verbosity coefficient $\lambda_2 \in [0, 5]$, progressively favoring more concise responses. We assess the mixed models using both reward scores (via a Mistral-7B-based reward model) and pairwise preference accuracy.

We conduct a preliminary study by applying DELTA to DPO – called *Distilled DPO* – using only distillation from the linearized teacher, without curvature regularization. We compare against: *(i) DPO-Mixed*, which optimizes a single scalarized objective over the combined dataset; *(ii) Non-Linear DPO*, and *(iii) Linear DPO*, which combine standard and linear fine-tuning with task arithmetic, respectively. As shown in Fig. 10, Distilled DPO closely matches the reward-score Pareto frontier of Linear DPO, achieving a favorable trade-off between helpfulness and verbosity while retaining inference-time efficiency. In terms of preference accuracy, our method surpasses both DPO-Mixed and Non-Linear DPO, although it slightly trails Linear DPO.

6. Conclusions and Limitations

We show that linearized behavior can be transferred to ordinary non-linear models, preserving the benefits of standard fine-tuning while improving the composability of task vectors. We describe two complementary effects: distillation induces quasi-linear behavior under weight perturbations, while curvature-aware regularization promotes support localization and reduces task interference. Together, these properties yield models that are easier to compose, tune, and deploy. Preliminary results further suggest that these principles can support controllable multi-objective alignment. The main drawback is the **training footprint** (Sec. A): training time increases by a factor of three, while memory usage doubles compared to conventional training. Improving training efficiency is an important direction for future work.

Acknowledgements

We acknowledge the CINECA award under the ISCRA initiative, for the availability of high-performance computing resources and support. This work is supported by the Horizon Europe Chips Joint Undertaking under the NexTArc project (HORIZON-JU-Chips-2024-2-RIA). NexTArc – Next Generation Open Innovations in Trustworthy Embedded AI Architectures for Smart Cities, Mobility and Logistics (Grant Agreement ID: 101194287, DOI: 10.3030/101194287). Additionally, the research activities of Angelo Porrello have been partially supported by the Department of Engineering "Enzo Ferrari" through the program FAR2025DIP (CUP E93C25000370005). We also gratefully acknowledge Symboolic s.r.l. for funding the PhD position of Thomas Sommariva. Finally, special thanks to Pietro Buzzega for the many valuable discussions, thoughtful suggestions, and insightful feedback throughout this work.

Impact Statement

Model merging is often treated as a purely post-hoc operation applied to independently fine-tuned checkpoints. In this view, users may download models from public repositories, such as Hugging Face, without having controlled their fine-tuning process. This work contributes instead to a **training-aware view** of model merging, which applies to scenarios where the same entity controls both fine-tuning and merging. For instance, an organization may train a library of composable-by-design models specialized along different behavioral axes and later combine them to provide personalized models without further optimization. Similarly, merging can be used for rehearsal-free continual learning, where new capabilities can be incrementally added to a base model by composing task vectors. In such settings, where the entity performing merging also controls the training process, restricting model merging to post-hoc procedures is unnecessarily limiting: one can instead introduce in-training regularization objectives that explicitly encourage learned updates to be more composable.

In broader terms, this work may support the democratization of AI by making task-specific model updates more modular, reusable, and composable, thereby enabling rapid, low-cost adaptation. Moreover, controllable composition of task vectors may support **more pluralistic AI systems**, where different capabilities, preferences, or behavioral axes can be combined to better reflect diverse user and application requirements. Furthermore, by improving task subtraction over standard non-linear fine-tuning, DELTA may facilitate *machine unlearning*. This is a relevant mechanism for complying with data privacy regulations (*e.g.,* the "right to be forgotten"), mitigating systemic biases, and efficiently removing toxic concepts from deployed models. Finally, by preserving the inference-time efficiency of standard non-linear fine-tuning, DELTA may make task arithmetic more practical in resource-constrained settings.

More effective model editing and merging can be a **double-edged sword**. The same mechanisms that allow benign customization could also be used to combine unsafe capabilities or weaken safety-relevant behaviors, increasing the risk of proliferating harmful AI systems. If task vectors encode undesirable, biased, or unsafe behaviors, improved composability may make it easier to transfer or combine such behaviors across models. In multi-objective alignment settings, composing preference directions could provide useful control, but may also produce unintended trade-offs if the objectives are misspecified or insufficiently evaluated.

Moreover, DELTA introduces additional training-time cost due to teacher–student optimization and curvature precomputation, with associated computational and environmental considerations. On the other hand, this cost is incurred once during training, while the resulting student has the same inference-time efficiency as a standard non-linear model. Whether this trade-off is beneficial depends on the deployment scenario: it is most favorable when a merged model is reused many times or when avoiding linearized inference substantially reduces serving cost.

References

Bowman, S. R., Angeli, G., Potts, C., and Manning, C. D. A large annotated corpus for learning natural language inference. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2015.

Buzzega, P., Salami, R., Porrello, A., and Calderara, S. Rethinking layer-wise model merging through chain of merges. *arXiv preprint arXiv:2508.21421*, 2025.

Cheng, G., Han, J., and Lu, X. Remote sensing image scene classification: Benchmark and state of the art. *Proceedings of the IEEE*, 2017.

Cimpoi, M., Maji, S., Kokkinos, I., Mohamed, S., and Vedaldi, A. Describing textures in the wild. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, 2014.

Coates, A., Ng, A., and Lee, H. An analysis of single-layer networks in unsupervised feature learning. In *International Conference on Artificial Intelligence and Statistics*, 2011.

Cui, G., Yuan, L., Ding, N., Yao, G., Zhu, W., Ni, Y., Xie, G., Liu, Z., and Sun, M. Ultrafeedback: Boosting language models with high-quality feedback. In *International Conference on Machine Learning*, 2024.

Dangel, F., Eschenhagen, R., Mucsányi, B., and Weber, T. Kfac from scratch. *arXiv*, 2025.

DataCanary, hilfialkaff, Jiang, L., Risdal, M., Dandekar, N., and tomtung. Quora question pairs, 2017. Kaggle dataset.

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L. Imagenet: A large-scale hierarchical image database. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, 2009.

Erdogan, M. Tangent space fine-tuning for directional preference alignment in large language models. *arXiv preprint arXiv:2602.01128*, 2026.

Fierro, C. and Roger, F. Steering language models with weight arithmetic. In *International Conference on Learning Representations*, 2026.

Gargiulo, A. A., Crisostomi, D., Bucarelli, M. S., Scardapane, S., Silvestri, F., and Rodola, E. Task singular vectors: Reducing task interference in model merging. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, 2025.

George, T., Laurent, C., Bouthillier, X., Ballas, N., and Vincent, P. Fast approximate natural gradient descent in a kronecker factored eigenbasis. *Advances in Neural Information Processing Systems*, 2018.

Goodfellow, I. J., Erhan, D., Carrier, P. L., Courville, A., Mirza, M., Hamner, B., Cukierski, W., Tang, Y., Thaler, D., Lee, D.-H., et al. Challenges in representation learning: A report on three machine learning contests. In *International conference on neural information processing*, 2013.

Grattafiori, A., Dubey, A., Jauhri, A., Pandey, A., Kadian, A., Al-Dahle, A., Letman, A., Mathur, A., Schelten, A., Vaughan, A., et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

Helber, P., Bischke, B., Dengel, A., and Borth, D. Eurosat: A novel dataset and deep learning benchmark for land use and land cover classification. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 2019.

Hinton, G., Vinyals, O., and Dean, J. Distilling the knowledge in a neural network, 2015.

Hu, E. J., Shen, Y., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., Wang, L., Chen, W., et al. Lora: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2022.

Huh, M., Cheung, B., Wang, T., and Isola, P. Position: The platonic representation hypothesis. In *International Conference on Machine Learning*, 2024.

Ilharco, G., Ribeiro, M. T., Wortsman, M., Schmidt, L., Hajishirzi, H., and Farhadi, A. Editing models with task arithmetic. In *International Conference on Learning Representations*, 2022a.

Ilharco, G., Wortsman, M., Gadre, S. Y., Song, S., Hajishirzi, H., Kornblith, S., Farhadi, A., and Schmidt, L. Patching open-vocabulary models by interpolating weights. In *Advances in Neural Information Processing Systems*, 2022b.

Iurada, L., Ciccone, M., and Tommasi, T. Efficient model editing with task-localized sparse fine-tuning. In *International Conference on Learning Representations*, 2025.

Jin, R., Hou, B., Xiao, J., Su, W., and Shen, L. Fine-tuning attention modules only: Enhancing weight disentanglement in task arithmetic. In *International Conference on Learning Representations*, 2025.

Khot, T., Sabharwal, A., and Clark, P. Scitail: A textual entailment dataset from science question answering. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2018.

Krause, J., Stark, M., Deng, J., and Fei-Fei, L. 3d object representations for fine-grained categorization. In *Proceedings of the IEEE international conference on computer vision workshops*, 2013.

Krizhevsky, A., Hinton, G., et al. Learning multiple layers of features from tiny images. *Master's thesis, University of Toronto*, 2009.

LeCun, Y., Bottou, L., Bengio, Y., and Haffner, P. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 2002.

Liu, T. Y. and Soatto, S. Tangent model composition for ensembling and continual fine-tuning. In *IEEE International Conference on Computer Vision*, 2023.

Loshchilov, I. and Hutter, F. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2017.

Marczak, D., Magistri, S., Cygert, S., Twardowski, B., Bagdanov, A. D., and van de Weijer, J. No task left behind: Isotropic model merging with common and task-specific subspaces. In *International Conference on Machine Learning*, 2025.

Marelli, M., Bentivogli, L., Baroni, M., Bernardi, R., Menini, S., and Zamparelli, R. 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)*, 2014.

Martens, J. New insights and perspectives on the natural gradient method. *Journal of Machine Learning Research*, 2020.

Martens, J. and Grosse, R. Optimizing neural networks with kronecker-factored approximate curvature. In *International Conference on Machine Learning*, 2015.

Netzer, Y., Wang, T., Coates, A., Bissacco, A., Wu, B., and Ng, A. Y. Reading digits in natural images with unsupervised feature learning. In *Proceedings of the NIPS Workshop on Deep Learning and Unsupervised Feature Learning*, 2011.

Nilsback, M.-E. and Zisserman, A. Automated flower classification over a large number of classes. In *Indian Conference on Computer Vision, Graphics and Image Processing*, 2008.

Ortiz-Jimenez, G., Favero, A., and Frossard, P. Task arithmetic in the tangent space: Improved editing of pre-trained models. *Advances in Neural Information Processing Systems*, 2023.

Panariello, A., Marczak, D., Magistri, S., Porrello, A., Twardowski, B., Bagdanov, A. D., Calderara, S., and van de Weijer, J. Accurate and efficient low-rank model merging in core space. In *Advances in Neural Information Processing Systems*, 2025.

Panariello, A., Rinaldi, F., Porrello, A., van de Weijer, J., and Calderara, S. merge-and-rebase. https://github.com/apanariello4/merge-and-rebase, 2026. Version 0.1.0.

Parkhi, O. M., Vedaldi, A., Zisserman, A., and Jawahar, C. Cats and dogs. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, 2012.

Porrello, A., Bonicelli, L., Buzzega, P., Millunzi, M., Calderara, S., and Cucchiara, R. A second-order perspective on model compositionality and incremental learning. In *International Conference on Learning Representations*, 2025a.

Porrello, A., Buzzega, P., Dangel, F., Sommariva, T., Salami, R., Bonicelli, L., and Calderara, S. Dataless weight disentanglement in task arithmetic via kronecker-factored approximate curvature. In *International Conference on Learning Representations*, 2025b.

Radford, A., Kim, J. W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., et al. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning*, 2021.

Rafailov, R., Sharma, A., Mitchell, E., Manning, C. D., Ermon, S., and Finn, C. Direct preference optimization: Your language model is secretly a reward model. In *Advances in Neural Information Processing Systems*, 2023.

Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W., and Liu, P. J. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 2020.

Ridnik, T., Ben-Baruch, E., Noy, A., and Zelnik-Manor, L. Imagenet-21k pretraining for the masses. In *Neural Information Processing Systems Datasets and Benchmarks Track*, 2021.

Rinaldi, F., Capitani, G., Bonicelli, L., Crisostomi, D., Bolelli, F., Ficarra, E., Rodola, E., Calderara, S., and Porrello, A. Update your transformer to the latest release: Re-basin of task vectors. In *International Conference on Machine Learning*, 2025.

Rinaldi, F., Panariello, A., Salici, G., Liu, F., Ciccone, M., Porrello, A., and Calderara, S. Gradient-sign masking for task vector transport across pre-trained models. In *International Conference on Learning Representations*, 2026a.

Rinaldi, F., Panariello, A., Salici, G., Porrello, A., and Calderara, S. Transporting task vectors across different architectures without training. In *International Conference on Machine Learning*, 2026b.

Schraudolph, N. N. Fast curvature matrix-vector products for second-order gradient descent. In *International Conference on Artificial Intelligence and Statistics*, 2003.

Stallkamp, J., Schlipsing, M., Salmen, J., and Igel, C. The german traffic sign recognition benchmark: a multi-class classification competition. In *The 2011 international joint conference on neural networks*, 2011.

Stoica, G., Ramesh, P., Ecsedi, B., Choshen, L., and Hoffman, J. Model merging with svd to tie the knots. In *International Conference on Learning Representations*, 2025.

Veeling, B. S., Linmans, J., Winkens, J., Cohen, T., and Welling, M. Rotation equivariant cnns for digital pathology. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, 2018.

Wang, A., Singh, A., Michael, J., Hill, F., Levy, O., and Bowman, S. R. Glue: A multi-task benchmark and analysis platform for natural language understanding. In *International Conference on Learning Representations*, 2018.

Wang, Z., Dong, Y., Delalleau, O., Zeng, J., Shen, G., Egert, D., Zhang, J. J., Sreedhar, M. N., and Kuchaiev, O. Helpsteer 2: Open-source dataset for training top-performing reward models. In *Advances in Neural Information Processing Systems*, 2024.

Williams, A., Nangia, N., and Bowman, S. A broad-coverage challenge corpus for sentence understanding through inference. In *North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 2018.

Xiao, J., Ehinger, K. A., Hays, J., Torralba, A., and Oliva, A. Sun database: Exploring a large collection of scene categories. *International Journal of Computer Vision*, 2016.

Yoshida, K., Naraki, Y., Horie, T., Yamaki, R., Shimizu, R., Saito, Y., McAuley, J., and Naganuma, H. Mastering task arithmetic: τjp as a key indicator for weight disentanglement. In *International Conference on Learning Representations*, 2025.

Zhang, Y., Baldridge, J., and He, L. Paws: Paraphrase adversaries from word scrambling. In *North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 2019.

Appendix

The appendix is organized as follows:

- Sec. A provides an analysis of the computational footprint, including memory requirements, training overhead, and inference costs.
- Sec. B details the implementation of our methods, with separate discussions for the vision and text domains.
- Sec. C presents an ablation study investigating the robustness of our method to the choice of the reference dataset $\mathcal{D}_\Omega$.
- Sec. D presents additional experiments and extended evaluations.

A. Computational Costs

Table 5. Comparison of training and inference costs across different methods and architectures. Training time is measured in ms/iteration (forward + backward). Note that at inference, our method employs the non-linear student architecture, matching its efficiency.

Method	Training				Inference			
	ViT-B/32		ViT-L/14		ViT-B/32		ViT-L/14	
	Mem (GB)	Time (ms)	Mem (GB)	Time (ms)	Time (ms)	Mem (GB)	Time (ms)	Mem (GB)
Linear FT	10.8	497	34.3	1914	320	3.33	1152	9.36
Non-Linear FT	7.7	223	26.2	988	160	3.03	364	8.27
DELTA (ours)	15.3	876	54.4	3562	160	3.03	364	8.27

In this section, we detail the memory footprint, training time, and inference costs associated with our approach.

Training overhead and memory footprint. The training process involves the joint optimization of both the student and teacher models. In Tab. 5, we report the training speeds. Our approach requires forward and backward passes through both models, which are executed concurrently. As a result, the training time is approximately three times that of the non-linear fine-tuning regime, representing a practical limitation of our approach.

Regarding peak GPU memory usage, our approach requires approximately 1.5–$2\times$ the memory of standard linear fine-tuning and approximately $2\times$ that of standard non-linear fine-tuning, as detailed in Tab. 5.

It is important to note that these measurements reflect a baseline implementation without any specific memory optimizations. The training footprint can be substantially mitigated in practice: since the backward passes of the student and teacher models are independent, either model can be offloaded from the GPU. Additional savings can be achieved by restricting fine-tuning to a subset of layers or by integrating parameter-efficient fine-tuning (PEFT) strategies, such as LoRA. Moreover, the distillation and curvature-regularization terms could be applied only periodically, rather than at every optimization step, or activated only after an initial phase of fine-tuning, further reducing training time.

Inference costs and deployment efficiency. The increased training requirements represent a *one-time cost* in the lifecycle of the resulting models. At deployment time, this cost is amortized by the efficiency of our merging pipeline.

As reported in Tab. 5, the distilled non-linear student model achieves significant latency improvements during a forward pass compared to the linearized teacher model. Measured on a single A100 GPU, the student model yields a $2\times$ speedup on ViT-B/32 and a $3.2\times$ speedup on ViT-L/14, alongside a reduction in peak memory usage. The computational overhead of the linearized model primarily stems from the computationally expensive Jacobian-vector products required during inference, which our distilled student model circumvents entirely.

Ultimately, our approach enables highly scalable model composition via simple task arithmetic. It completely bypasses the need for complex, post-hoc procedures (*e.g.,* SVD-based methods) or expensive linearization operations at deployment. This efficiency is particularly advantageous when merging large-scale models, where traditional SVD methods become prohibitively expensive and must be repeatedly re-computed for different weight configurations. By resolving these

Algorithm 1 DistillEd Linearized Task Arithmetic (DELTA)

Input: Pre-trained initialization $\boldsymbol{\theta}_0$, target task t, reference dataset $\mathcal{D}_\Omega$, hyper-parameters $\beta^T, \beta^S, \gamma, \eta$
Pre-computation: Compute EK-FAC curvature matrices (GGN factors U_A^l, U_G^l, S^l) on the reference dataset $\mathcal{D}_\Omega$
Initialize linearized teacher $\boldsymbol{\theta}_t^T \leftarrow \boldsymbol{\theta}_0$ and non-linear student $\boldsymbol{\theta}_t^S \leftarrow \boldsymbol{\theta}_0$
Define task vectors: $\boldsymbol{\tau}_t^T \leftarrow \boldsymbol{\theta}_t^T - \boldsymbol{\theta}_0$ and $\boldsymbol{\tau}_t^S \leftarrow \boldsymbol{\theta}_t^S - \boldsymbol{\theta}_0$
while training not converged **do**
  Sample a batch $\mathcal{X} = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^B$ from task t
  Sample interpolation scalar $\alpha \sim \mathcal{U}(0.5, 1)$
  $\mathcal{L}_{\text{KD}}^S \leftarrow \frac{1}{B}\sum_{i=1}^B \left\| f(\boldsymbol{x}_i; \boldsymbol{\theta}_0 + \alpha\boldsymbol{\tau}_t^S) - \texttt{SG}\left[f_{\text{lin}}(\boldsymbol{x}_i; \boldsymbol{\theta}_0 + \alpha\boldsymbol{\tau}_t^T)\right] \right\|_2^2$
  $\mathcal{L}^T \leftarrow \mathcal{L}_{\text{CE}}(\mathcal{X}; \boldsymbol{\theta}_t^T) + \beta^T \mathcal{L}_{\text{drift}}(\boldsymbol{\theta}_t^T)$
  $\mathcal{L}^S \leftarrow \mathcal{L}_{\text{CE}}(\mathcal{X}; \boldsymbol{\theta}_t^S) + \beta^S \mathcal{L}_{\text{drift}}(\boldsymbol{\theta}_t^S) + \gamma \mathcal{L}_{\text{KD}}(\mathcal{X}; \boldsymbol{\theta}_t^S)$
  $\boldsymbol{\theta}_t^T \leftarrow \boldsymbol{\theta}_t^T - \eta \nabla_{\boldsymbol{\theta}_t^T} \mathcal{L}^T$
  $\boldsymbol{\theta}_t^S \leftarrow \boldsymbol{\theta}_t^S - \eta \nabla_{\boldsymbol{\theta}_t^S} \mathcal{L}^S$
end while

bottlenecks, our framework effectively supports dynamic settings – such as pluralistic alignment – where model compositions must be efficiently adjusted on-the-fly at inference time to match varying user preferences.

B. Implementation Details

This section details the datasets and implementation specifics used in the experiments presented in the paper. The overall training procedure is summarized in Alg. 1. The code for replicating our method is available at https://github.com/apanariello4/merge-and-rebase (Panariello et al., 2026).

Vision domain. We evaluate our framework on the **8-Vision** benchmark (Ilharco et al., 2022a) which comprises eight heterogeneous image classification datasets: Stanford Cars (Krause et al., 2013), DTD (Cimpoi et al., 2014), EuroSAT (Helber et al., 2019), GTSRB (Stallkamp et al., 2011), MNIST (LeCun et al., 2002), RESISC45 (Cheng et al., 2017), SUN397 (Xiao et al., 2016), and SVHN (Netzer et al., 2011). The **14-Vision** benchmark (Gargiulo et al., 2025) serves as an extension for scalability analysis, incorporating six additional classification datasets: CIFAR100 (Krizhevsky et al., 2009), STL10 (Coates et al., 2011), Flowers102 (Nilsback & Zisserman, 2008), Oxford-IIIT Pet (Parkhi et al., 2012), PCAM (Veeling et al., 2018), and FER2013 (Goodfellow et al., 2013).

For training vision task vectors, we followed the setup of previous works (Ilharco et al., 2022a; Ortiz-Jimenez et al., 2023; Yoshida et al., 2025), adopting a batch size of 128. We used the AdamW optimizer (Loshchilov & Hutter, 2017) with a learning rate of 1×10^{-5}, weight decay of 0.1, and a cosine annealing learning rate scheduler. For TAK (Porrello et al., 2025b), we follow the original paper and use a learning rate of 3×10^{-4}. Unlike prior approaches, we did not apply gradient clipping during training. We weight the regularization terms in the losses Eq. (5) and Eq. (7) by $\beta^T = \beta^S = 500$. The distillation loss term in Eq. (7) is weighted by $\gamma = 1.0$. We grid the coefficient α in $(0; 1]$ for task addition and in $(0; 2]$ for task negation.

Language domain. Natural Language Inference (NLI) experiments are tested on the **6-NLI** benchmark (Stoica et al., 2025), which includes six datasets: SNLI (Bowman et al., 2015), MultiNLI (Williams et al., 2018), SICK (Marelli et al., 2014), which are three-way classification tasks where the relation between a premise and a hypothesis must be identified as entailment, contradiction, or neutral. In contrast, SciTail (Khot et al., 2018), RTE (Wang et al., 2018), and QNLI (Wang et al., 2018) are binary entailment tasks, and therefore fine-tuning and evaluation are restricted to two labels.

For training language task vectors, we adopted a batch size of 128, using an AdamW optimizer (Loshchilov & Hutter, 2017) with a learning rate of 3×10^{-4}, an iteration-based cosine-annealing scheduler and a weight decay of 0.01. Like in vision tasks, we did not apply gradient clipping during training. The regularization term in the losses Eq. (5) and Eq. (7) is set to $\beta^T = \beta^S = 20$ and the distillation loss term in Eq. (7) is weighted by $\gamma = 0.01$.

Table 6. **Robustness to the choice of $\mathcal{D}_\Omega$.** Performance comparison across Vision and Text modalities when different reference datasets are used for our regularization technique. *Abs.* denotes the absolute accuracy, while *Norm.* represents accuracy normalized by the accuracies of individually fine-tuned models.

Method	α	Accuracy	
		Abs.	Norm.
Vision (ViT-B/32)			
Non-Linear FT	1.0	32.0	32.9
	Best	73.5	80.4
DELTA	1.0	88.3	98.3
Reg. ImageNet-21k (Ours)	Best	88.3	98.3
DELTA	1.0	87.5	96.0
Reg. ImageNet-1k	Best	88.0	97.3

Method	α	Accuracy	
		Abs.	Norm.
Text (T5-Base)			
Non-Linear FT	1.0	42.0	49.7
	Best	78.2	91.6
DELTA	1.0	82.3	95.5
Reg. C4 (Ours)	Best	82.4	95.5
DELTA	1.0	80.3	93.4
Reg. PAWS	Best	81.3	92.3
DELTA	1.0	81.6	93.4
Reg. QQP	Best	82.0	94.0

C. Ablation of the reference dataset $\mathcal{D}_\Omega$

To investigate the robustness to the choice of the reference data $\mathcal{D}_\Omega$, Tab. 6 evaluates how varying the dataset used for curvature estimation affects performance. Specifically, we substitute ImageNet-21k with the smaller ImageNet-1k (Deng et al., 2009) for vision tasks, and replace the broad C4 (Raffel et al., 2020) corpus with smaller datasets, such as PAWS (Zhang et al., 2019) and QQP (DataCanary et al., 2017), for language tasks. As shown, while the choice of reference dataset has a measurable impact, the overall performance remains highly robust. In all cases, the student model maintains state-of-the-art results, outperforming all competing methods from Tab. 2 across nearly all evaluation settings.

D. Additional experiments.

Weight Disentanglement. In Fig. 11, we extend our evaluation and include Attention-Only Fine-Tuning (Jin et al., 2025), as well as Linear Fine-Tuning (Ortiz-Jimenez et al., 2023). This comparison demonstrates that `DELTA` achieves a significantly lower disentanglement error compared to all other evaluated fine-tuning techniques.

Sensitivity to the scaling coefficient. Fig. 12 reports the merged models' sensitivity to the scaling coefficient, extending our analysis to ViT-L/14 on the `8-Vision` benchmark and ViT-B/32 on the `14-Vision` benchmark. While the peak accuracy of competing methods is occasionally comparable, `DELTA` consistently achieves higher overall accuracy across all architectures and benchmarks. Crucially, our method uniquely maintains robust performance across the entire α-sweep, effectively eliminating the need for an α grid search.

Impact on Task Arithmetic. In Fig. 13, we analyze the contribution of each loss component to per-task accuracy, extending our evaluation to ViT-L/14 on the `8-Vision` benchmark and T5-Base on the `6-NLI` benchmark. Additionally, Tab. 7 reports the corresponding aggregate results, covering both task addition and task negation. In the task addition setting, the vision results show that the curvature-regularization objective alone is already highly effective at improving final accuracy. This effect is particularly pronounced for ViT-L/14, where linearization appears to play a more marginal role than in ViT-B/32. A possible explanation is that, due to its larger width, ViT-L/14 may operate closer to an infinite-width training regime, in which the first-order Taylor approximation remains accurate even along naturally non-linear training trajectories. In the textual domain, instead, the results more clearly highlight the complementary role of the two objectives. In the task negation setting, the complementarity between the two objectives becomes even more evident.

Student-Teacher comparison. Fig. 14 reports the single-task performance of the student versus the teacher before merging. In addition to the T5 results on the `6-NLI` benchmark reported in the main paper, we extend this comparison to ViT-L/14 on the `8-Vision` benchmark, as well as ViT-B/32 on both the `8-Vision` and `14-Vision` benchmarks. `DELTA` consistently outperforms its linearized teacher across all architectures and datasets. This confirms that the greater expressivity of the non-linear regime is effectively exploited to learn richer representations, rather than simply mimicking the teacher's activations.

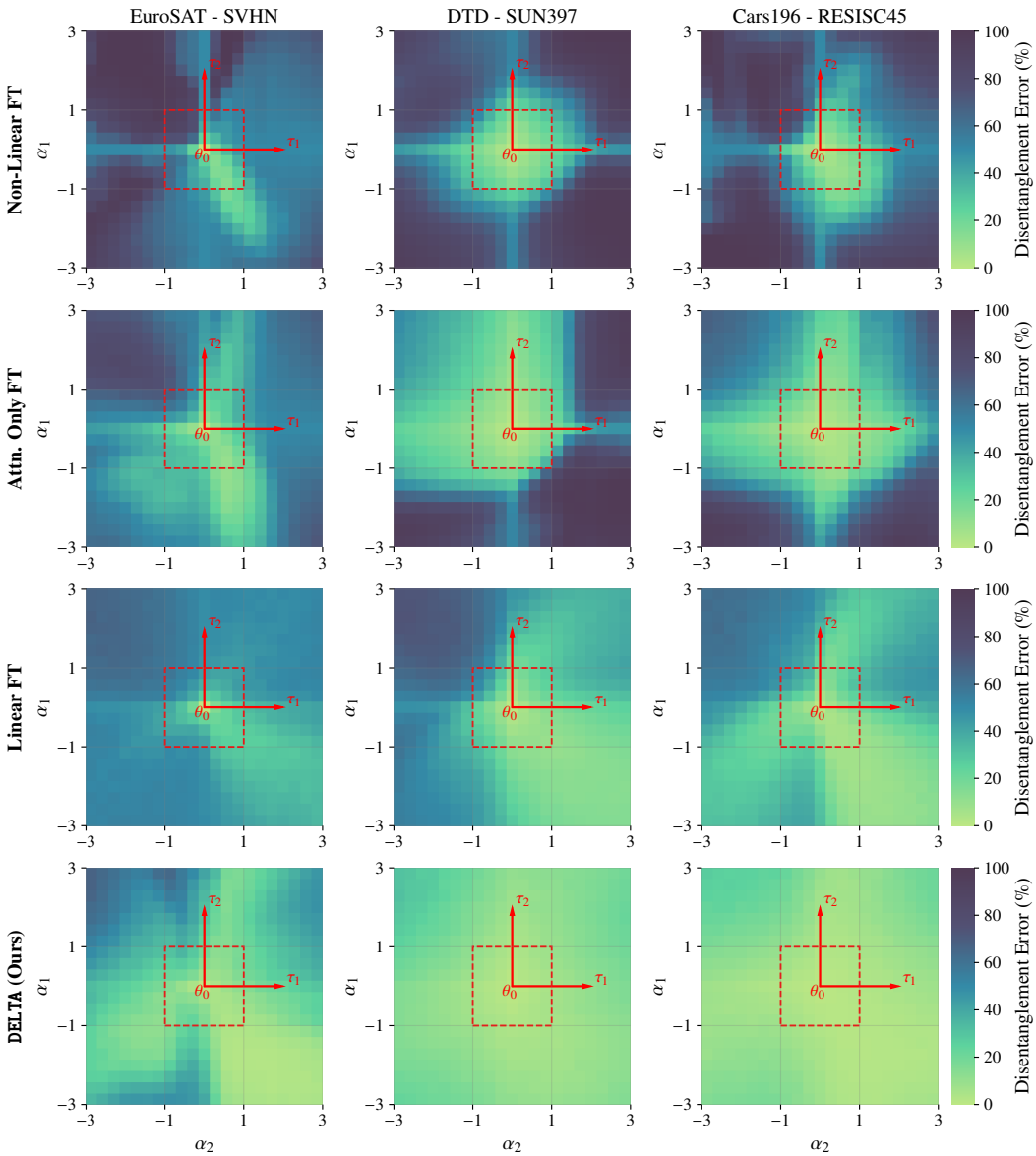

Figure 11. Weight Disentanglement (Ortiz-Jimenez et al., 2023) for Non-Linear FT, Linear FT, Attention-Only FT (Jin et al., 2025) and our distillation-based approach.

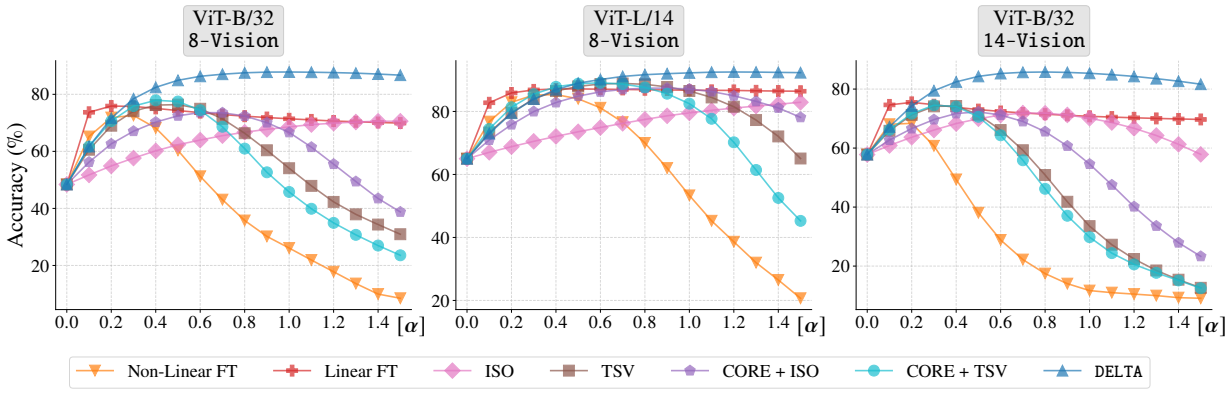

Figure 12. Sensitivity of the merged model to the scaling coefficient on LoRA checkpoints.

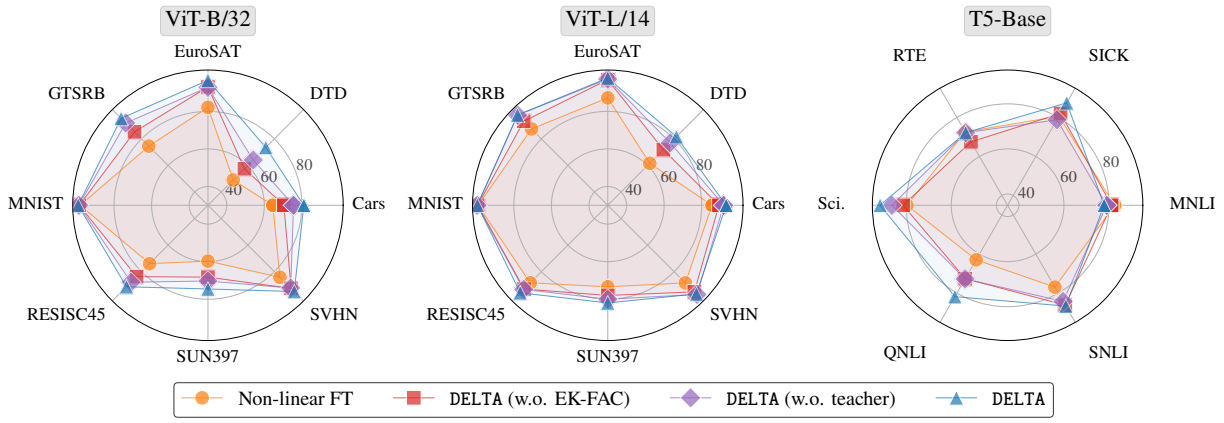

Figure 13. **Individual task performance.** Accuracy of the merged model on each task, illustrating the impact of our distillation framework across different architectures and benchmarks.

Table 7. **Ablation on task addition and task negation.** Left: loss ablation for task addition, reporting the mean absolute accuracy of the merged model across architectures and benchmarks. Right: the same study for task negation, reporting unlearning performance on target tasks and retention on control tasks (ImageNet-1K).

Method	8-Vision B/32	8-Vision L/14	6-NLI T5
Non-Linear FT	73.5	84.5	78.2
DELTA (w/o EK-FAC)	81.1	89.6	78.5
DELTA (w/o teacher)	84.2	91.7	78.9
DELTA (ours)	**88.3**	**92.7**	**82.4**

Method	8-Vision B/32 Targ.	8-Vision B/32 Cont.	8-Vision L/14 Targ.	8-Vision L/14 Cont.	14-Vision B/32 Targ.	14-Vision B/32 Cont.
Non-Linear FT	20.4	60.5	18.1	72.3	30.5	60.6
DELTA (w/o EK-FAC)	14.2	61.1	13.2	72.9	25.2	61.0
DELTA (w/o teacher)	16.8	62.4	19.0	74.9	25.3	62.2
DELTA (ours)	**9.6**	**62.1**	**11.7**	**74.7**	**19.1**	**62.1**

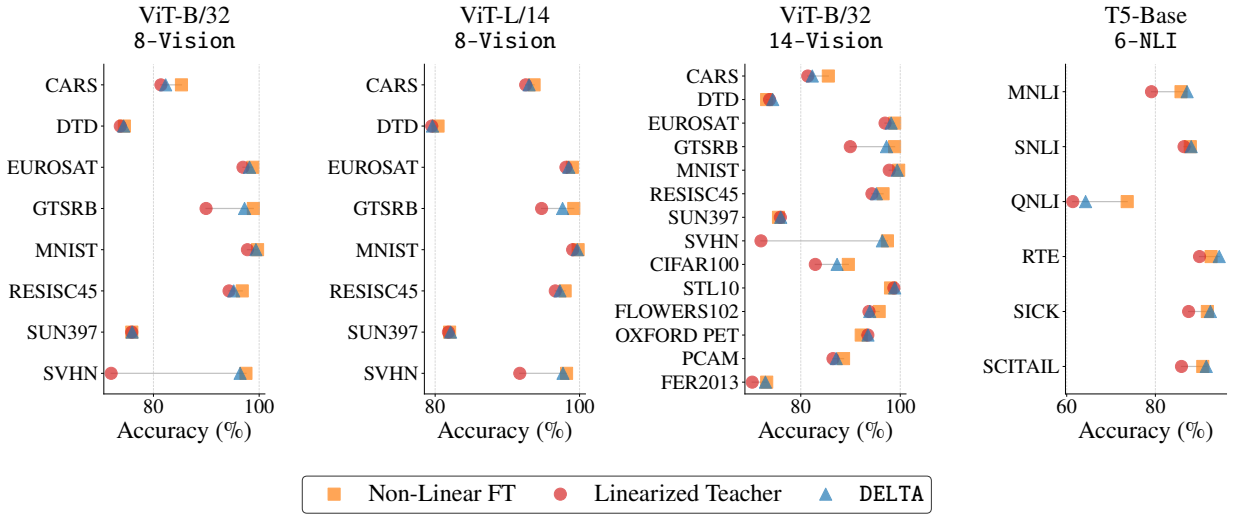

Figure 14. **Student - Teacher comparison.** Accuracy of the single task fine-tuning on the corresponding dataset

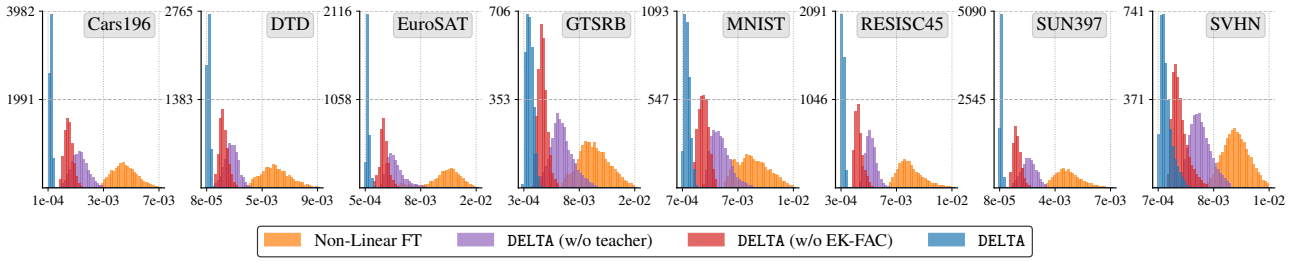

Figure 15. Histogram of the linearization error with the ViT-L/14 backbone.

Linearization error. In Fig. 15, we plot the linearization error of our method on the ViT-L/14 backbone for the `8-Vision` benchmark. The results confirm that our full protocol achieves a near-zero linearization error, substantially reducing the discrepancy relative to the non-linear baseline. Our analysis also reveals that teacher distillation is the core driver of this alignment; without it, the linearization error degrades much more severely than when curvature regularization is omitted.

Support localization. Fig. 16 illustrates the impact of each individual loss component on the Mean Squared Error (MSE) between the activations of the pre-trained and individually fine-tuned models. We replicate this analysis for the ViT-L/14 backbone in Fig. 17. Consistent with the findings in our main text, these ablations confirm that `DELTA` achieves **stronger support localization** than conventional Non-Linear FT. Specifically, the edit distance is substantially higher for In-Domain Data while remaining near zero for Out-of-Domain Data, indicating a strict reversion to the pre-trained model for out-of-distribution inputs.

Crucially, these supplementary results isolate the exact source of this phenomenon: ablating curvature regularization (DELTA w/o Reg) severely disrupts this clean separation, whereas removing distillation (DELTA w/o Distill) largely preserves it. This verifies that curvature regularization, rather than distillation, is the primary driver of support localization.

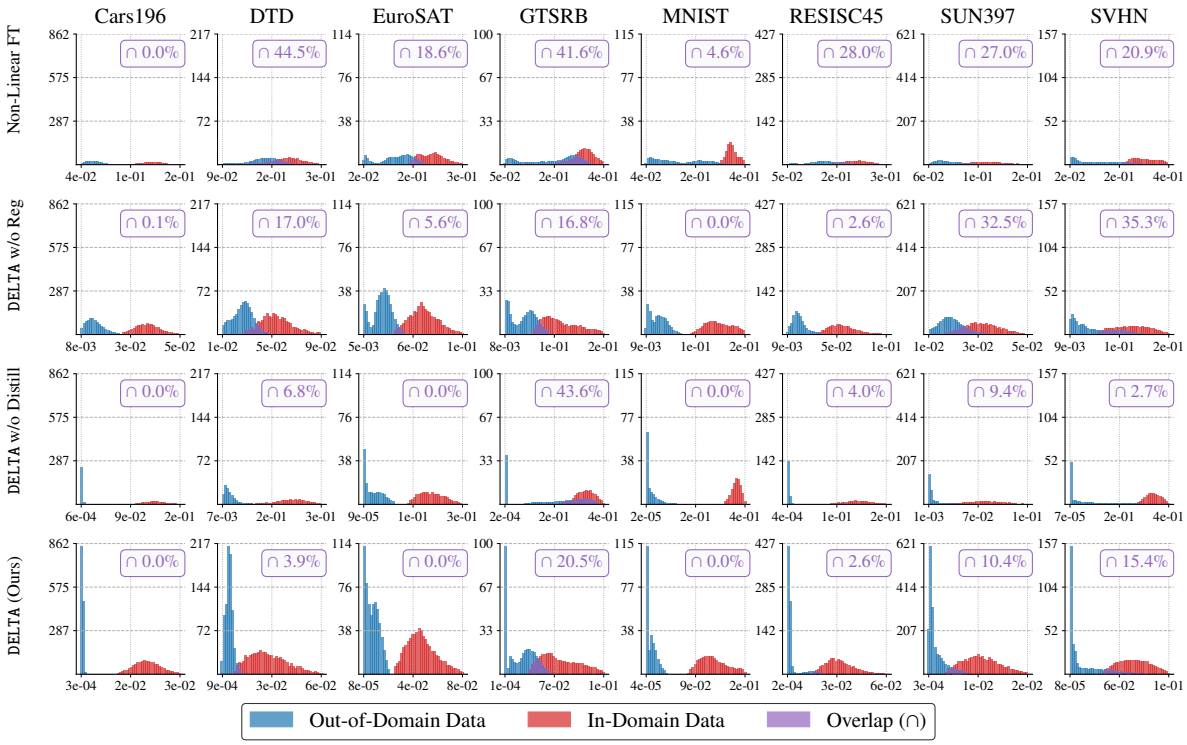

Figure 16. **Activation MSE Distributions.** Extended histograms of the activation MSE between the pre-trained and fine-tuned models, as defined in Fig. 5. Evaluated on the ViT-B/32 backbone, we compare our full DELTA protocol with its ablations (*w/o Distill* and *w/o Reg*) alongside standard Non-Linear FT.

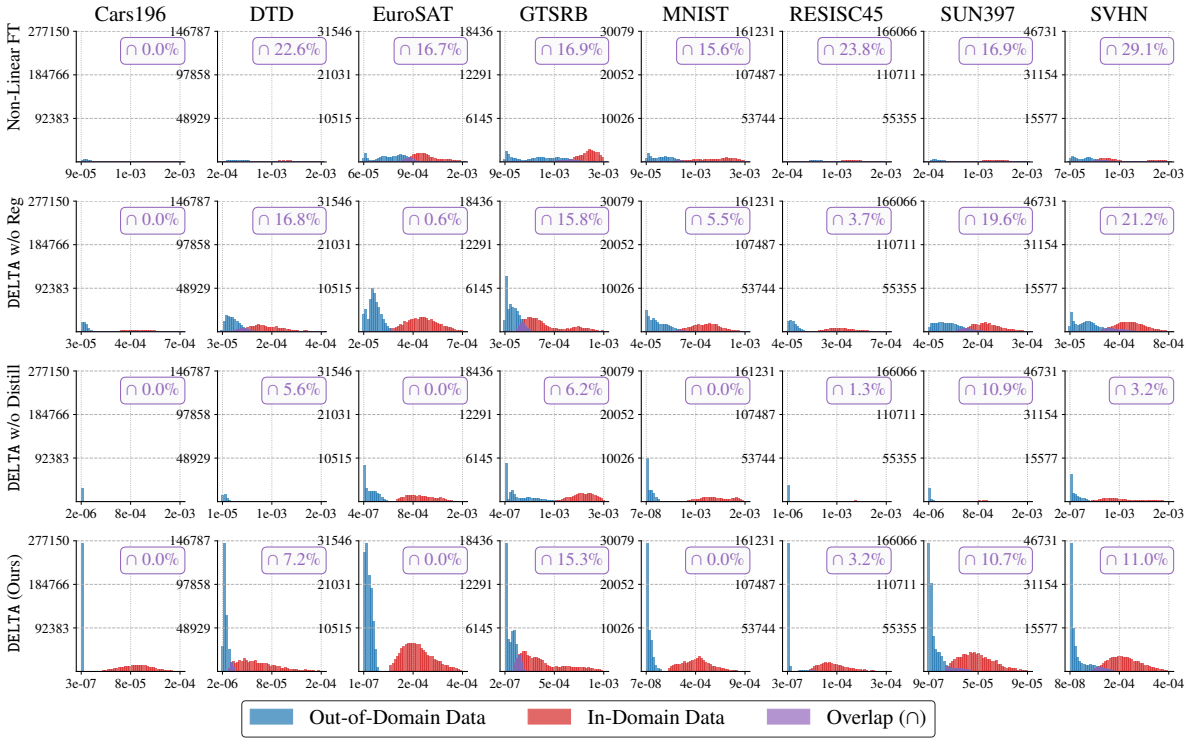

Figure 17. **Activation MSE Distributions on ViT-L/14.** Histograms of the activation MSE evaluated on the ViT-L/14 backbone. Consistent with the base architecture, the ablation of our loss components confirms that curvature regularization remains the primary driver of support localization at scale.