BILINEAR AND PAIRING-AWARE CONTINUAL MODEL MERGING

Anonymous authors
Paper under double-blind review

ABSTRACT

Continual model merging (CMM) combines independently fine-tuned task models as they arrive, without returning to earlier checkpoints or task data. Preserving the parameter directions of past tasks is only part of this problem. The merger must also retain enough information to correct later updates. We introduce **BiPaCo** (**Bi**linear admission with **Pa**iring-Aware **Co**rrection), a training-free method that stores consolidated input and output bases and a bounded ledger of paired spectral records. These records preserve the within-task coupling between input and output directions. Bilinear admission discards the incoming component aligned with both previous bases, while pairing-aware correction re-queries surviving records in the spectral blocks of the provisional delta and adjusts selected blocks toward historical targets in closed form. Using one configuration, BiPaCo leads training-free methods in accuracy across all nine CLIP-ViT settings and achieves better backward transfer than the baselines designed for continual model merging in each setting. In the measured 8-task ViT-B/32 setting, merging is faster than the optimization-based baselines. Correct pairings improve accuracy in every tested order of the 8- and 20-task ViT-B/32 streams. Flan-T5 and Gemma-2-2B results extend the evaluation to language models.

1 INTRODUCTION

Continual learning has traditionally been formulated as the problem of learning sequentially arriving task data while preserving performance on previously learned tasks (Parisi et al., 2019; De Lange et al., 2022; Mai et al., 2022; Wang et al., 2024). Most continual learning approaches therefore assume direct access to task datasets and rely on gradient-based optimization to update a single model incrementally (Rusu et al., 2016; Kirkpatrick et al., 2016; Rebuffi et al., 2017; Chaudhry et al., 2019).

However, the recent paradigm shift toward foundation models has changed how task knowledge is accumulated and distributed. Instead of storing knowledge in raw task data, modern machine learning pipelines often represent task-specific knowledge as independently fine-tuned models or parameter updates built upon a shared pretrained backbone (Bommasani et al., 2021; Dosovitskiy et al., 2021; Radford et al., 2021; Ilharco et al., 2023). As a result, recent research has increasingly focused on continual model merging (CMM) (Tang et al., 2025), where the objective is to sequentially integrate multiple task-specific models while retaining previously acquired capabilities. In this setting, the fundamental unit of continual learning shifts from task data to the models themselves.

At every arrival, the new task is integrated on top of the current merged model without further access to its raw training data. The initial method OPCM (Tang et al., 2025) showed that training-free consolidation is feasible through projection-based sequential merging alone. Later methods gain accuracy using gradient-based optimization, adapter training, or test-time signals (Yang et al., 2025; Qiu et al., 2025; 2026). For a merger that uses none of these steps, the central question is what it should remember from earlier tasks.

OPCM and NUFILT (Qiu et al., 2026) estimate past-task directions from a layer-wise SVD of the accumulated merged delta, which no longer indicates which task contributed each direction. As new tasks arrive, it becomes increasingly unclear which directions should be preserved and which components should be removed. Recording the input and output directions of each task at arrival avoids estimating them from this aggregate, but the resulting bases capture only which directions

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.

have appeared, not which input direction was paired with which output direction within a task, or how strongly. A running aggregate such as the mean task vector can still supply a correction target, but not how many earlier tasks support it.

Building on these observations, we propose BiPaCo, a training-free CMM method. Bilinear admission removes only the incoming component jointly aligned with the previous input and output bases. Because admission constrains only the incoming update, it cannot correct drift already present in the merged delta. Pairing-aware correction therefore re-queries earlier paired spectral records in the current spectral blocks of the provisional delta and moves selected blocks toward targets computed from those records in closed form. The correction grows stronger when more relevant records support a block. The bases guide admission, while the records guide correction. We keep the ledger within a fixed budget by discarding the oldest records first. The procedure requires no task data, gradients, trainable adapters, test-time samples, or routing.

Our contributions are twofold. **(i)** We identify within-task pairings of input and output directions as historical information lost by separate bases and not recoverable per task from an aggregate delta. We propose BiPaCo, which uses these records for correction after bilinear admission, the unique minimum-change update under the joint protection constraint (Proposition 3.2). Holding directions and coefficients fixed, the correct pairings yield higher accuracy than scrambled pairings in every tested order of the 8- and 20-task ViT-B/32 streams. **(ii)** With a single configuration, BiPaCo achieves the highest average accuracy (ACC) among training-free methods on CLIP (Radford et al., 2021) ViT-B/32, B/16, and L/14 across 8/14/20-task streams and on Flan-T5 (Chung et al., 2024) over GLUE (Wang et al., 2018). We also report exploratory results on Gemma-2-2B (Gemma Team et al., 2024). Across the nine CLIP-ViT settings, BiPaCo attains higher backward transfer than every continual baseline.

2 Related Works

Offline model merging. Offline model merging combines task-specific models independently fine-tuned from a shared pretrained backbone in parameter space. Task Arithmetic (Ilharco et al., 2023) showed that summing their task vectors yields a multi-task model. Subsequent work explores sign conflict resolution (Yadav et al., 2023), importance-weighted averaging (Matena & Raffel, 2022; Jin et al., 2023), and adaptive coefficient learning (Yang et al., 2024). SVD-based methods reduce interference by aligning task singular directions in a shared basis (Stoica et al., 2025), suppressing their overlap (Gargiulo et al., 2025), or flattening the merged spectrum (Marczak et al., 2025). These methods assume that all task models are available at once, and adaptive variants also rely on test-time samples. Their accuracy falls when the same operators are applied task by task (Section 4.3). In such a stream, KnOTS and TSV-M use each task's singular directions at its arrival. BiPaCo retains the directions and their pairings so they can inform later merges.

Continual model merging. OPCM (Tang et al., 2025) performs orthogonal projection of incoming task vectors with respect to the accumulated merged delta, enabling data-free and training-free CMM. DOP (Yang et al., 2025) optimizes projection-based stability and plasticity objectives with MGDA (Désidéri, 2012), and NUFILT (Qiu et al., 2026) filters incoming task vectors in the null space of the merged delta and trains LoRA (Hu et al., 2022) adapters with a data-free surrogate loss. DOP and NUFILT are data-free but not training-free, since they require gradient computation for the current task. Methods using additional data take a different route. MINGLE (Qiu et al., 2025) trains a mixture of null-space gated low-rank experts using a small set of unlabeled test samples. ODE-M (Lin & Kang, 2026) uses loss gradients on a calibration set of earlier tasks to construct an ODE-guided merge trajectory.

Subspace projection in continual learning. Protecting earlier tasks by projecting updates away from stored subspaces is well established in gradient-based continual learning. OGD (Farajtabar et al., 2020) and Adam-NSCL (Wang et al., 2021), for example, constrain training updates on task data. CMM methods carry this idea into parameter space, often estimating protected subspaces from the merged delta. BiPaCo instead records the directions of each task at arrival and preserves how they were paired within that task alongside the input and output bases. The bases guide admission, while the paired records guide correction of the accumulated merge.

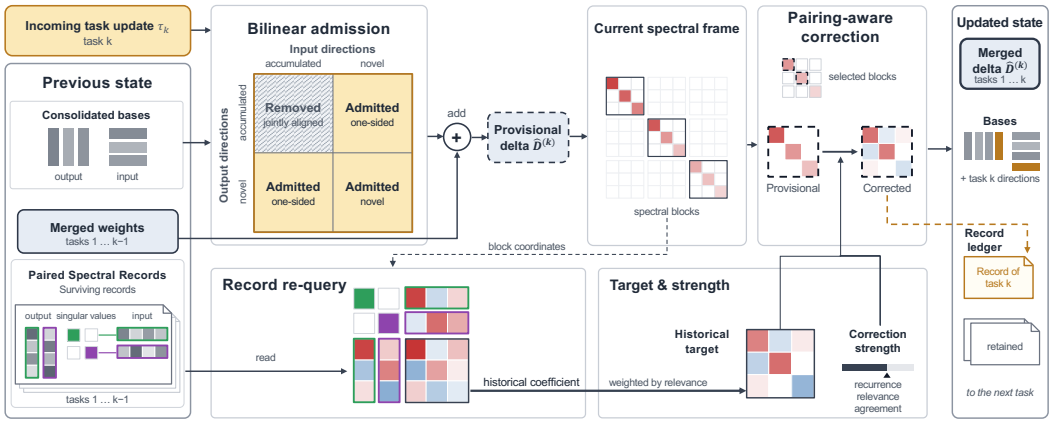

Figure 1: **Overview of BiPaCo.** Bilinear admission removes only the component of the incoming task vector jointly aligned with the previous consolidated bases. After admission, the bases shown in the Updated state panel are extended and used to re-query the paired spectral records in the current spectral blocks of the provisional delta. Pairing-aware correction then moves the selected blocks toward their historical targets. The record of task k is appended to the ledger only after correction. Blue and red entries denote negative and positive coefficients, and matching green or purple outlines connect an output direction, singular value, and input direction paired within the same record. Amber marks the current task k, hatching marks the component removed by admission, and dashed outlines mark blocks selected for correction.

3 Method

3.1 Problem Setup

We aim to address the problem of training-free continual model merging, where task-specific models arrive sequentially and must be integrated without access to training data or additional optimization. We follow the CMM setting of Tang et al. (2025), which carries only the merged model and a few scalars between steps. We additionally allow the merger to keep an explicit historical state.

Definition 3.1 (Training-free continual model merging)**.** *Given a pretrained model θ_0, a new fine-tuned model θ_k, independently trained from θ_0, arrives at each task k. A CMM algorithm $\mathcal{A}$ updates the merged model and its historical state as*

$$(\theta_{\text{merged}}^{(k)}, \mathcal{H}^{(k)}) = \mathcal{A}(\theta_{\text{merged}}^{(k-1)}, \mathcal{H}^{(k-1)}, \theta_k, \theta_0). \tag{1}$$

(i) Data-free. *No access to task training or calibration data.* ***(ii) Historical state.*** *$\mathcal{H}^{(k)}$ may retain information derived from each task model at its arrival, but the task model itself is not kept afterwards.* ***(iii) Strict online.*** *No access to future tasks.* ***(iv) Training-free.*** *No gradient-based optimization or trainable parameters are used.*

3.2 Overview of BiPaCo

Notation. Throughout, $\tau_k \in \mathbb{R}^{m\times n}$ denotes the task vector at a given linear layer, defined as the difference between the corresponding weight matrices of θ_k and θ_0, where m and n are the output and input dimensions of that layer. Following prior work (Ilharco et al., 2023; Tang et al., 2025), we retain the term "vector" even though τ_k is a matrix at each layer. We initialize $\theta_{\text{merged}}^{(0)} = \theta_0$ and $D^{(0)} = 0$. We write $D^{(k)} = \theta_{\text{merged}}^{(k)} - \theta_0$ for the corrected merged delta after step k, and $\hat{D}^{(k)}$ for the provisional delta obtained after bilinear admission but before correction.

Figure 1 illustrates one arrival, in which BiPaCo performs five steps in order. It (1) applies bilinear admission to τ_k with the previous bases (Section 3.4), (2) extends the bases with the leading singular directions of τ_k (Section 3.3), (3) re-queries the surviving records of earlier tasks in the spectral

blocks of $\hat{D}^{(k)}$ (Section 3.5), (4) corrects the selected blocks to obtain $D^{(k)}$ (Section 3.6), and (5) appends the record of task k to the ledger and enforces its budget. We omit layer indices below. The ledger budget and the selection of corrected blocks are shared across layers. Algorithm 1 in Appendix A.7 gives the complete procedure.

3.3 Consolidated Bases and Paired Spectral Records

Per-task direction extraction. For each nonzero task vector $\tau_k = U_k \Sigma_k V_k^\top$ with singular values $\sigma_{k,1} \geq \cdots \geq \sigma_{k,p} \geq 0$ and $p = \min(m, n)$, we retain the smallest rank whose cumulative singular-value mass exceeds α,

$$r_k = \min\Big\{ r : \textstyle\sum_{j \leq r} \sigma_{k,j} \big/ \sum_{j \leq p} \sigma_{k,j} > \alpha \Big\}, \tag{2}$$

together with the leading singular vectors U_{k,r_k} and V_{k,r_k} and the singular values $s_k = (\sigma_{k,1}, \ldots, \sigma_{k,r_k})$, where singular values tied at the boundary are also retained (Appendix A.1).

Consolidated bases. Each layer maintains orthonormal bases U_c and V_c for past output and input directions, both initially empty. After admission, we append only the components of the retained vectors that the bases do not already cover,

$$U_c^{(k)} = \big[\, U_c^{(k-1)},\ \mathrm{orth}\big((I - U_c^{(k-1)} U_c^{(k-1)\top})\, U_{k,r_k}\big)\big], \tag{3}$$

and analogously for $V_c^{(k)}$. Here $\mathrm{orth}(\cdot)$ returns an orthonormal basis for the numerically independent columns of its argument, computed by a thin SVD (Appendix A.1). Existing columns are never modified, and each basis is capped at p columns.

Paired spectral records. The bases preserve the input and output directions seen so far, but lose their within-task associations and strengths. We keep these associations in a record for each task i,

$$E_i = (\bar{U}_i, s_i, \bar{V}_i), \qquad \bar{U}_i = U_c^{(i)\top} U_{i,r_i}, \quad \bar{V}_i = V_c^{(i)\top} V_{i,r_i}, \tag{4}$$

The record expresses the retained singular vectors in the bases available at task i. Later bases append columns without changing earlier ones. At step $k > i$, we therefore decode E_i using the first d_i^U and d_i^V columns of $U_c^{(k)}$ and $V_c^{(k)}$, where d_i^U and d_i^V denote the basis widths at task i,

$$\widehat{U}_i = U_c^{(k)}[:, 1{:}d_i^U]\, \bar{U}_i, \qquad \widehat{V}_i = V_c^{(k)}[:, 1{:}d_i^V]\, \bar{V}_i, \qquad \widehat{\tau}_i = \widehat{U}_i\, \mathrm{Diag}(s_i) \widehat{V}_i^\top. \tag{5}$$

If the bases contain the retained vectors, decoding recovers them exactly. Otherwise, it recovers their projections. The ledger stores these coordinates and singular values without retaining a past task vector.

Record ledger. Records are ordered by arrival in a ledger. After step k, its surviving task indices form $\mathcal{I}_k \subseteq \{1, \ldots, k\}$. Together with the bases, this ledger is the historical state $\mathcal{H}^{(k)}$ in Definition 3.1. We append the record of task k after the correction in Section 3.6, preventing a task from correcting itself. The ledger is capped at $B_\rho = \max\{1, \lfloor \rho\, \mathrm{mem}(\tau) \rfloor\}$ bytes, where $\mathrm{mem}(\tau)$ is the storage of one dense task vector over the merged layers. If the cap is exceeded, we delete the oldest records across all layers until the ledger fits. This does not change the bases or the merged delta.

3.4 Bilinear Admission

Bilinear admission uses the previous bases $U_c = U_c^{(k-1)}$ and $V_c = V_c^{(k-1)}$. For $\tau \in \mathbb{R}^{m \times n}$, the matrix $\Gamma_c(\tau) = U_c^\top \tau V_c$ collects the coefficients with which τ maps input directions in $\mathrm{col}(V_c)$ to output directions in $\mathrm{col}(U_c)$. We remove only this jointly aligned component $U_c \Gamma_c(\tau_k) V_c^\top$ and retain components aligned with only one side,

$$\tau_k^\perp = \tau_k - U_c\, \Gamma_c(\tau_k)\, V_c^\top, \qquad \hat{D}^{(k)} = D^{(k-1)} + \tau_k^\perp. \tag{6}$$

With empty bases, $\tau_k^\perp = \tau_k$.

Proposition 3.2 (Minimum-change admission)**.** *For orthonormal U_c and V_c, $\tau_k^\perp$ is the unique minimizer of $\|X - \tau_k\|_F$ subject to $\Gamma_c(X) = 0$, with $\|\tau_k - \tau_k^\perp\|_F = \|\Gamma_c(\tau_k)\|_F$. In particular, it removes no more Frobenius mass than the one-sided updates $P_L(\tau_k) = \tau_k - U_c U_c^\top \tau_k$ and $P_R(\tau_k) = \tau_k - \tau_k V_c V_c^\top$, which satisfy the same constraint.*

The proof is given in Appendix A.3.

3.5 RE-QUERYING RECORDS IN SPECTRAL BLOCKS

After admission and basis extension, write $U_c = U_c^{(k)}$ and $V_c = V_c^{(k)}$. Each arrival can change the spectral frame of the provisional delta. We therefore express every surviving record E_i, $i \in \mathcal{I}_{k-1}$, in the frame of the current merge, obtained from

$$U_c^\top \hat{D}^{(k)} V_c = \widetilde{P}\,\mathrm{Diag}(\widetilde{\sigma})\widetilde{Q}^\top, \tag{7}$$

We group directions with numerically tied singular values into blocks g and treat each untied direction as a singleton block. For block g, the output and input directions are $\Phi_g = U_c\widetilde{P}_g$ and $\Psi_g = V_c\widetilde{Q}_g$. Its current coefficients are $C_g = \Phi_g^\top \hat{D}^{(k)}\Psi_g = \mathrm{Diag}(\widetilde{\sigma}_g)$. Treating tied directions together keeps the correction independent of the singular-vector choice within a tie (Appendix A.6).

For each record and block, we compute a relevance w_{ig} and a coefficient prescription T_{ig},

$$w_{ig} = \ell_{ig}^{\mathrm{L}}\,\ell_{ig}^{\mathrm{R}}, \qquad T_{ig} = \Phi_g^\top \widehat{U}_i\,\mathrm{Diag}(s_i)\widehat{V}_i^\top\Psi_g, \tag{8}$$

Here $\ell_{ig}^{\mathrm{L}} = \|\widehat{U}_i^\top\Phi_g\|_F^2/|g|$ and $\ell_{ig}^{\mathrm{R}} = \|\widehat{V}_i^\top\Psi_g\|_F^2/|g|$ measure average coverage on the two sides, and $|g|$ is the block width. Multiplying the coverages assigns little relevance when a record overlaps a block on only one side. The prescription T_{ig} reads the record's coefficients in that block. Thus w_{ig} reflects the directions spanned by a record, while T_{ig} also reflects their pairings. For example, coupling u_1 to v_1 and u_2 to v_2 gives a diagonal prescription, whereas crossing the pairings gives an antidiagonal one for the same block (Appendix A.8).

3.6 PAIRING-AWARE CORRECTION

Target and strength. We combine the records relevant to block g into a historical target μ_g and a correction strength λ_g,

$$\mu_g = \frac{\sum_i w_{ig}T_{ig}}{\sum_i w_{ig}}, \qquad \lambda_g = \underbrace{\max(N_g^{\mathrm{eff}} - 1, 0)}_{\text{recurrence}} \cdot \underbrace{\bar{w}_g}_{\text{relevance}} \cdot \underbrace{A_g}_{\text{agreement}}, \tag{9}$$

where $N_g^{\mathrm{eff}} = (\sum_i w_{ig})^2/\sum_i w_{ig}^2$ is the effective number of relevant records, $\bar{w}_g = \sum_i w_{ig}^2/\sum_i w_{ig}$ is their typical relevance, and $A_g \in [0,1]$ equals one when all records prescribe the same coefficients and decreases with their weighted dispersion around μ_g. When q agreeing records have the same relevance w, the strength becomes $\lambda_g = (q-1)w$. A single supporting record therefore produces no correction. Section 4.2 examines recurrence and agreement.

Block correction. For block g, we minimize the objective function

$$J_g(X) = \tfrac{1}{2}\|X - C_g\|_F^2 + \tfrac{\lambda_g}{2}\|X - \mu_g\|_F^2, \tag{10}$$

which balances the provisional coefficients against the historical target. Its unique minimizer interpolates between the two matrices,

$$C_g^* = (1-\gamma_g)\,C_g + \gamma_g\,\mu_g, \qquad \gamma_g = \frac{\lambda_g}{1+\lambda_g} \in [0,1). \tag{11}$$

Block selection. We do not adjust every block. Replacing C_g with C_g^* reduces J_g by $b_g/2$, where

$$b_g = \frac{\lambda_g^2}{1+\lambda_g}\|C_g - \mu_g\|_F^2 \tag{12}$$

is large when the provisional block lies far from a well-supported target. We pool b_g across all layers and select the smallest set $\mathcal{S}$ of highest-scoring blocks whose scores sum to more than an α fraction of the total. This reuses the α of Eq. (2), with selection details in Appendix A.5. Each layer then commits

$$D^{(k)} = \hat{D}^{(k)} + \sum_{g\in\mathcal{S}} \Phi_g(C_g^* - C_g)\Psi_g^\top, \tag{13}$$

Selected blocks receive coefficients C_g^*. The remaining parts of $\hat{D}^{(k)}$, including components outside the spans of U_c and V_c, are unchanged.

4 EXPERIMENTS

Settings. For vision, we use CLIP-ViT-B/32, B/16, and L/14 on 8, 14, and 20 task streams. For NLP, we use Flan-T5-base (Chung et al., 2024) on 8 GLUE (Wang et al., 2018) tasks and Gemma-2-2B (Gemma Team et al., 2024) on five MergeBench (He et al., 2025) domains. We use ten task orders G1 to G10 for vision (following NUFILT (Qiu et al., 2026)) and NLP, and three orders G1 to G3 for Gemma. Following Definition 3.1, step k accesses only $(\theta_0, \theta_k, D^{(k-1)}, \mathcal{H}^{(k-1)})$. Checkpoints, datasets, and orders are given in Appendices B.1 and B.2.

Metrics. We report two standard metrics, average accuracy (ACC) and backward transfer (BWT), where $a_{k,i}$ denotes the accuracy on task i after the k-th merge (Spearman's $\rho \times 100$ for STS-B).

$$\mathrm{ACC} = \frac{1}{K}\sum_{i=1}^{K} a_{K,i}, \qquad \mathrm{BWT} = \frac{1}{K-1}\sum_{i=1}^{K-1}(a_{K,i} - a_{i,i}). \tag{14}$$

ACC and BWT are reported in percent as mean $\pm$ standard deviation (std) over task orders.

Baselines. Baselines are all data-free and fall into three categories. Static merging methods, applied sequentially to each arriving task, include Weight Averaging (Izmailov et al., 2018), Task Arithmetic (Ilharco et al., 2023), TIES-Merging (Yadav et al., 2023), ISO-C (Marczak et al., 2025), KnOTS-TIES (Stoica et al., 2025), TSV-M (Gargiulo et al., 2025), and DC-Merge (Zhang et al., 2026). Optimization-based continual methods include DOP (Yang et al., 2025) and NUFILT (Qiu et al., 2026). The training-free continual baseline is OPCM (Tang et al., 2025). Implementation details are given in Appendix B.3.

Hyperparameters. The method uses α for both per-task rank selection and correction coverage, and ρ for the ledger budget. We choose $(\alpha, \rho) = (0.4, 0.5)$ using three ViT-B/32 orders, G11 to G13, that are separate from the ten evaluation orders. This setting is then held fixed across backbones, task scales, and orders (Appendix B.5).

Table 1: Continual merging on 8/14/20 vision tasks across three CLIP-ViT backbones. ACC (%) and BWT (%) are mean $\pm$ std over G1 to G10. Baseline details are in Appendix B.3. **Bold** indicates the best training-free ACC.

Method	Training-free	ViT-B/32			ViT-B/16			ViT-L/14		
		8 tasks	14 tasks	20 tasks	8 tasks	14 tasks	20 tasks	8 tasks	14 tasks	20 tasks
Pre-Trained	-	48.1	56.9	55.6	55.4	62.0	59.8	64.9	69.1	65.6
Individual	-	90.4	89.3	89.8	92.4	91.3	91.6	94.3	93.4	93.5
C. Fine-Tuned	-	79.8	67.4	62.6	82.9	72.2	68.2	90.0	70.9	77.7
Weight Averaging	✓	66.3±0.0	65.4±0.0	61.1±0.0	72.3±0.0	69.7±0.0	64.8±0.0	80.0±0.0	77.5±0.0	71.1±0.0
Task Arithmetic	✓	67.5±0.0	66.5±0.0	60.6±0.0	77.1±0.0	70.9±0.0	64.2±0.0	82.1±0.0	77.9±0.0	70.3±0.0
TIES-Merging	✓	49.0±10.2	66.2±0.6	59.9±0.7	66.8±3.7	70.5±0.8	63.0±1.6	64.3±7.0	78.0±0.6	68.3±0.9
ISO-C	✓	71.7±1.2	73.2±1.8	67.6±0.8	78.5±1.2	79.7±1.3	73.0±1.1	86.9±0.5	86.9±1.8	80.9±0.8
KNOTS-TIES	✓	54.4±6.9	67.8±0.4	60.5±1.4	57.9±8.4	71.6±0.3	62.7±1.1	68.3±5.7	78.8±0.3	69.7±0.8
TSV-M	✓	68.2±4.8	63.3±4.8	58.8±3.3	75.4±4.0	69.2±3.1	63.1±1.3	82.2±3.6	78.1±3.6	70.5±1.2
DC-Merge	✓	64.8±1.9	65.2±2.4	60.0±1.3	70.8±1.5	70.0±1.6	64.1±1.1	79.6±1.7	77.9±1.5	70.6±1.5
OPCM	✓	75.5±0.5	71.9±0.3	65.7±0.2	81.8±0.3	77.1±0.5	70.3±0.2	87.0±0.4	83.5±0.2	76.0±0.2
BiPaCo (ours)	✓	**84.0±0.5**	**79.1±0.6**	**72.2±0.9**	**87.8±0.5**	**82.8±0.7**	**77.2±0.8**	**91.7±0.3**	**89.3±0.3**	**83.6±0.7**
Optimization-based references										
DOP	✗	78.3±1.6	73.0±1.2	68.5±1.6	84.5±0.8	76.2±1.2	72.2±1.8	88.3±0.7	85.5±0.8	80.7±0.9
NUFILT	✗	83.6±0.2	78.0±0.2	71.0±0.9	87.3±0.1	83.1±0.3	78.1±0.9	91.6±0.1	89.2±0.1	84.7±0.8
Weight Averaging	✓	-11.5±2.2	-8.0±1.3	-7.1±2.1	-9.7±1.5	-7.1±1.4	-7.3±1.7	-7.3±1.4	-5.8±1.0	-6.4±1.5
Task Arithmetic	✓	-9.6±1.5	-1.3±0.3	-3.4±0.4	-4.2±1.0	-1.3±0.4	-3.6±0.4	-7.1±0.8	-1.8±0.3	-3.3±0.3
TIES-Merging	✓	-15.3±8.0	1.9±0.6	-1.5±0.7	-5.5±0.4	1.4±0.7	-1.5±1.2	-13.0±5.7	1.1±0.4	-2.9±1.0
ISO-C	✓	-10.2±1.2	-10.4±1.9	-10.3±1.4	-6.7±0.5	-7.1±1.1	-9.9±1.7	-3.7±0.6	-3.7±1.7	-5.5±1.3
KNOTS-TIES	✓	-12.6±3.9	1.9±0.5	-1.3±0.7	-13.5±5.5	1.0±0.1	-2.3±0.6	-11.6±3.6	0.3±0.3	-2.3±0.7
TSV-M	✓	-24.0±5.6	-26.7±4.9	-31.7±3.4	-18.5±4.6	-22.7±3.1	-28.2±1.8	-13.0±4.0	-15.6±3.9	-23.3±1.3
DC-Merge	✓	-27.7±2.2	-24.7±2.4	-30.2±1.4	-23.1±1.8	-21.9±1.6	-28.0±1.1	-15.6±1.8	-15.7±1.5	-23.3±1.5
OPCM	✓	-6.3±1.1	-6.0±1.0	-7.8±1.5	-4.8±0.7	-5.1±1.4	-6.3±2.2	-2.6±1.0	-4.3±0.7	-6.5±1.8
BiPaCo (ours)	✓	2.0±2.0	0.6±2.3	-3.2±1.8	1.3±1.3	0.9±2.1	-2.4±1.7	0.9±0.8	-0.1±0.8	-2.2±1.2
Optimization-based references										
DOP	✗	-6.1±2.3	-8.8±2.1	-12.5±2.8	-3.9±1.0	-7.5±2.2	-11.4±2.8	-3.2±1.2	-4.6±1.0	-8.4±1.4
NUFILT	✗	-2.7±0.7	-5.7±0.9	-8.9±2.3	-1.6±0.5	-3.5±0.6	-7.1±1.9	-1.1±0.3	-2.0±0.3	-4.6±0.7

Table 2: Continual merging on Flan-T5-base over 8 GLUE tasks. STS-B is scored by Spearman correlation $\times 100$ and the others by accuracy. Values are mean $\pm$ std over G1 to G10. **Bold** marks the best training-free score for each task and ACC.

Method	Training-free	CoLA	MNLI	MRPC	QNLI	QQP	RTE	SST-2	STS-B	ACC (%)	BWT (%)
Pre-Trained	-	69.1	56.5	76.2	88.4	82.1	80.1	91.2	62.2	75.7	-
Individual	-	75.0	83.4	87.5	91.5	85.4	85.9	93.6	88.7	86.4	-
Task Arithmetic	✓	70.5±0.0	57.8±0.0	78.4±0.0	90.2±0.0	83.6±0.0	80.5±0.0	92.3±0.0	77.8±0.0	78.9±0.0	−1.8±1.1
TIES-Merging	✓	42.7±23.5	7.5±16.6	80.3±2.4	85.9±3.6	78.2±2.7	71.8±4.7	76.0±24.2	40.2±39.4	60.3±8.3	−15.3±8.0
ISO-C	✓	69.3±0.2	63.8±4.4	77.9±1.2	89.7±0.3	83.9±0.3	81.9±1.0	91.8±0.2	69.5±1.8	78.5±0.5	−2.2±1.0
KNOTS-TIES	✓	63.8±6.2	17.2±25.0	**81.5±1.9**	88.7±1.1	80.9±1.4	75.7±1.9	90.9±2.3	60.1±27.7	69.9±5.8	−7.3±6.5
TSV-M	✓	70.1±1.2	70.4±11.9	77.1±3.0	90.1±0.8	83.6±0.8	81.4±1.5	92.1±0.8	74.5±4.8	79.9±1.3	−6.3±1.5
DC-Merge	✓	69.5±0.3	67.4±11.3	77.1±1.8	89.7±0.6	83.4±0.7	81.3±1.7	91.8±0.6	69.7±5.4	78.7±1.2	−6.2±1.4
OPCM	✓	68.6±2.0	65.0±12.2	78.7±2.6	89.9±0.6	83.5±0.8	81.1±2.1	92.1±1.0	77.9±5.0	79.6±1.5	−0.6±0.7
BiPaCo (ours)	✓	**70.7±0.7**	**78.4±3.3**	78.0±2.5	**90.4±0.7**	**84.1±0.5**	**83.4±2.5**	**93.1±0.5**	**84.3±4.3**	**82.8±0.6**	−0.2±1.0
Optimization-based references											
DOP	✗	72.6±0.5	80.8±1.0	80.7±0.8	90.5±0.2	83.5±0.3	81.1±0.5	93.0±0.3	86.3±0.5	83.6±0.2	−1.4±0.3
NUFILT	✗	72.4±0.8	83.0±0.5	80.4±0.9	90.9±0.2	84.0±0.4	80.4±0.9	93.2±0.2	86.8±0.6	83.9±0.2	−1.2±0.4

Table 3: Gemma-2-2B results across five evaluation domains on G1 to G3. Values are mean $\pm$ std over three orders. M-MMLU averages French, Spanish, German, and Russian. **Bold** marks the best score in each domain and ACC.

Method	Instruction (IFEval)	Math (GSM8K)	Multilingual (M-MMLU)	Coding (HumanEval+)	Safety (WildGuardTest)	ACC (%)	BWT (%)
Task Arithmetic	23.2 ± 0.3	42.4 ± 0.2	$\mathbf{43.5 \pm 0.0}$	27.0 ± 0.1	49.0 ± 0.2	37.0 ± 0.1	$+1.8 \pm 1.3$
TIES-Merging	16.5 ± 1.2	$\mathbf{51.3 \pm 4.2}$	43.1 ± 0.5	27.4 ± 3.4	44.6 ± 2.5	36.6 ± 1.7	$+5.0 \pm 2.7$
OPCM	15.3 ± 2.0	39.5 ± 5.3	42.4 ± 0.6	22.8 ± 2.9	45.9 ± 2.5	33.2 ± 1.6	-1.8 ± 2.5
BiPaCo (ours)	$\mathbf{26.6 \pm 0.5}$	47.1 ± 4.5	42.3 ± 0.4	$\mathbf{33.4 \pm 4.5}$	$\mathbf{65.1 \pm 2.1}$	$\mathbf{42.9 \pm 1.8}$	$+0.8 \pm 3.2$

4.1 Main Results

CMM on Vision. Table 1 shows that BiPaCo achieves the highest mean ACC among training-free methods in all nine CLIP-ViT settings, improving over OPCM by 4.7 to 8.5%. It exceeds continual fine-tuning on task data in every setting and DOP by 2.9 to 6.6% without merge-time optimization. Relative to NUFILT, its ACC ranges from -1.1 to $+1.2\%$ and is higher in six settings. Its BWT exceeds that of OPCM, DOP, and NUFILT in all nine settings, including by 2.4 to 4.7% in the three settings where its ACC trails NUFILT. BWT is positive in five settings, including all three eight-task settings. On 20-task ViT-B/32, where all four methods have their lowest BWT, BiPaCo reaches -3.2%, compared with -7.8, -12.5, and -8.9%, respectively. Consistent with growing support from earlier records, removing correction changes BWT from 2.0 to -5.3% on 8 tasks and from -3.2 to -5.2% on 20 tasks (Table 4).

CMM on Language. On Flan-T5/GLUE (Table 2), BiPaCo achieves the best training-free ACC of 82.8%, 3.2% above OPCM, while DOP and NUFILT retain advantages of 0.8 and 1.1%. It leads the training-free methods on seven of the eight tasks, exceeds both optimization-based references on QQP and RTE, and attains the highest BWT in the table (-0.2%). Its largest margins over the other training-free methods appear on MNLI and STS-B, at 8.0 and 6.4%, respectively.

CMM on Gemma-2-2B. On Gemma-2-2B (Table 3), BiPaCo achieves the highest mean ACC at 42.9%, 5.9% above Task Arithmetic. It leads on instruction following, coding, and safety, while TIES leads on math and Task Arithmetic on multilingual understanding. The largest margin appears on safety, where BiPaCo exceeds the next best method by 16.1%. Task Arithmetic and TIES attain higher BWT, but their final ACC remains at least 5.9% below that of BiPaCo.

4.2 Component Analysis

Pairing. Scrambling the within-task pairings by permuting right directions at query time, while keeping the stored records, admission, and ledger unchanged, lowers ACC by 4.6% at both task scales and in every order. Its less negative BWT on 20 tasks reflects a 5.9% lower arrival score rather than better retention (Appendix B.4).

Table 4: Component controls on ViT-B/32 over G1 to G10 with $\alpha = 0.4$ and $\rho = 0.5$. Values are mean $\pm$ std. Fixed strength and w/o recurrence change only the strength of the blocks selected by the full method.

		8 tasks		20 tasks	
Component	Variant	ACC (%)	BWT (%)	ACC (%)	BWT (%)
	Full	84.0 ± 0.5	2.0 ± 2.0	72.2 ± 0.9	-3.2 ± 1.8
Pairing	Scrambled	79.4 ± 1.8	0.1 ± 1.9	67.6 ± 2.5	-1.8 ± 1.9
Correction	Fixed $\lambda = 0.5$	83.6 ± 0.5	1.8 ± 2.1	72.0 ± 0.5	-4.0 ± 2.1
	W/o recurrence	80.5 ± 1.3	0.6 ± 1.6	68.6 ± 2.4	-1.9 ± 1.9
	None	66.2 ± 4.6	-5.3 ± 1.3	52.7 ± 4.3	-5.2 ± 2.2
Admission	Left-only	81.6 ± 1.0	1.7 ± 2.2	70.2 ± 1.4	-2.0 ± 1.6
	Right-only	75.8 ± 2.3	3.6 ± 2.7	67.9 ± 1.5	1.1 ± 2.0
	No projection	69.5 ± 3.5	-7.3 ± 7.6	48.3 ± 4.1	-23.6 ± 6.6

Correction. Correction uses earlier task records to adjust the provisional delta containing past merges, rather than restoring the component removed by admission. Removing correction lowers ACC by 17.8 and 19.5% at the two task scales. Removing the recurrence factor lowers ACC by 3.5 and 3.6%, showing the value of increasing strength with the number of relevant records. A fixed $\lambda = 0.5$ keeps ACC within 0.4% of the full method but lowers BWT by 0.2 and 0.8%. Deriving the strength from the records thus costs no ACC and requires no additional strength hyperparameter. Removing agreement or replacing the target by a simple mean changes ACC by at most 0.3% at both scales and lowers BWT by 0.3% on 20 tasks, suggesting that these choices act mainly on retention (Appendix B.4). Spectral blocks are not merely singletons and reach widths of up to 185 on ViT-B/32.

Admission. Bilinear admission gives the highest ACC at both task scales. Left-only and right-only projections lower ACC by 2.4 and 8.2% on 8 tasks and by 2.0 and 4.3% on 20 tasks. Removing projection lowers it by 14.5 and 23.9%. The higher BWT of right-only projection reflects over-protection, since it discards more of each incoming task. The BWT of left-only projection stays within one standard deviation of Full.

4.3 Offline Reference and Ledger Budget

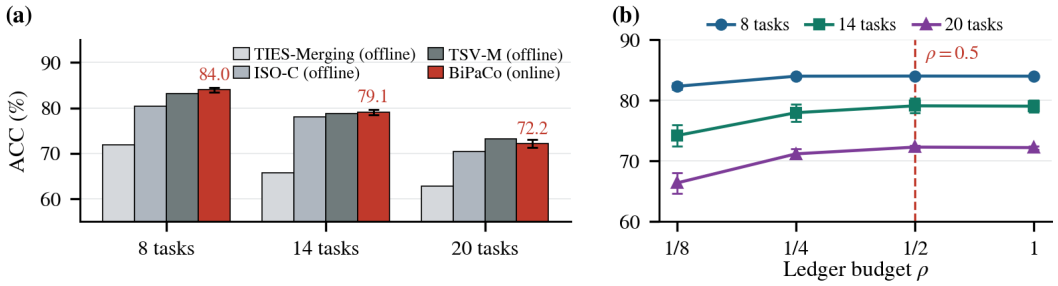

Figure 2: (a) ViT-B/32 ACC of BiPaCo, which receives tasks sequentially, averaged over G1 to G10, and of TIES-Merging, ISO-C, and TSV-M applied offline to all task vectors at once, which does not depend on the order. (b) ViT-B/32 ACC of BiPaCo for different ledger budgets ρ on three task orders G11 to G13 other than the evaluation orders. The dashed line marks the chosen $\rho = 0.5$. Error bars show std over orders.

Figure 2(a) compares BiPaCo on ViT-B/32 with TIES-Merging, ISO-C, and TSV-M applied offline to all task vectors at once. Although BiPaCo receives the tasks one at a time, it exceeds all three on 8 and 14 tasks and trails only TSV-M, by 1.0%, on 20 tasks. Applied sequentially, the same methods lose up to 22.9% (Table 1), so their gap to BiPaCo in Table 1 stems mainly from the sequential arrival of tasks rather than from their merging operators.

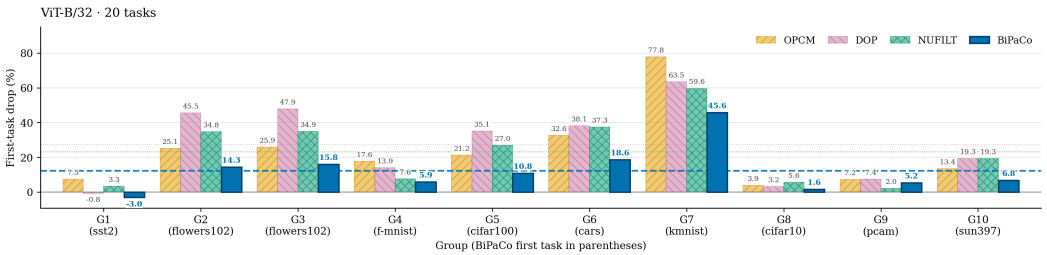

Figure 3: Relative first-task drop on ViT-B/32 with 20 tasks across G1 to G10. Lower is better, and negative bars denote improvement. Dashed lines mark method means.

Figure 2(b) varies the ledger budget on the three tuning orders G11 to G13. ACC changes by at most 0.1% between $\rho = 0.5$ and $\rho = 1$, whereas $\rho = 1/8$ costs 5.9% on 20 tasks, so $\rho = 0.5$ matches the larger budget with a ledger of 147 instead of 308 MiB (Appendix B.5).

4.4 Robustness to Task Ordering

CMM can hide forgetting behind strong initial merging, so high initial accuracy does not imply good retention. Figure 3 reports the relative first-task drop $100(a_{1,1} - a_{20,1})/a_{1,1}$ on ViT-B/32 with 20 tasks. BiPaCo has a mean drop of 12.2%, compared with 23.2% for OPCM, 27.3% for DOP, and 23.1% for NUFILT, and a lower drop than OPCM and DOP in every order and than NUFILT in nine of the ten. Results on 8 and 14 tasks are given in Appendix B.7.

4.5 Computational Overhead

On 8-task CLIP ViT-B/32, BiPaCo takes 46.9 s (Table 5), slightly longer than OPCM but about 1.9× faster than NUFILT and 20× faster than DOP. Compared with TIES-Merging, its extra per-arrival work consists of an SVD of the incoming task vector, basis extension, an SVD of the coordinate core $U_c^\top \hat{D}^{(k)} V_c$ of size at most $p \times p$, and a query over surviving records. The layer-rank limit and ledger cap bound per-arrival work and persistent state. FIFO begins deleting the oldest records on longer streams. On 20-task ViT-B/32, the bases, ledger, and merged delta occupy 949 MiB, about 15% of the 6480 MiB needed for all twenty task vectors (Appendix B.6). Inference uses a single backbone-sized model.

Table 5: Wall-clock time for constructing the merged model on CLIP ViT-B/32 with 8 tasks. Evaluation time is excluded for all methods. Values are single runs on order G1.

Method	Training-free	Iter.	Time (s)
TIES-Merging	✓	✗	27.9
OPCM	✓	✗	41.8
BiPaCo (ours)	✓	✗	46.9
NUFILT	✗	50	89.9
DOP	✗	200	952.7

5 Conclusion

We introduced BiPaCo, a training-free CMM method that retains each task's paired input and output directions instead of reconstructing history from an aggregate delta. Bilinear admission removes the incoming component jointly aligned with previous bases, and pairing-aware correction re-queries surviving records to adjust the provisional delta. Across nine CLIP-ViT settings, BiPaCo attains the highest ACC among training-free methods, performs on par with optimization-based references, and has higher BWT than OPCM, DOP, and NUFILT. It also gives the best training-free ACC on Flan-T5 and, in exploratory runs over three orders, on Gemma-2-2B. Scrambling the pairings alone lowers ACC by 4.6% and, on 20 tasks, the final score of the last task by 12.6%, so the records help admit new tasks as well as retain old ones. The bases grow toward layer rank and take the largest share of stored state, while our Gemma-2-2B evaluation covers three orders. Next steps include retiring basis directions unused by surviving records, selecting α and ρ from the incoming stream, and evaluating larger language and multimodal models.

AI USE STATEMENT

We used generative AI tools to check the grammar and edit author-drafted text, review whether the implementation code matches the equations in Section 3, and check the derivations in Appendix A, including the proof of Proposition 3.2. All AI-assisted text, code, and proofs were verified by the authors.

ETHICS STATEMENT

This work proposes a training-free algorithm for merging fine-tuned models and does not involve human subjects, the collection of new data, or personally identifiable information. All experiments use publicly available datasets and model checkpoints (Appendix B.1), in accordance with their respective licenses.

REPRODUCIBILITY STATEMENT

Section 3 defines every step of BiPaCo, and Appendices A.1 to A.6 describe the implementation details, including tie handling, boundary cases, and ledger accounting. Proposition 3.2 is proved in full in Appendix A.3. All checkpoints are publicly available FusionBench and MergeBench models (Appendix B.1), and all evaluation task orders are listed in Appendix B.2. A single configuration, $(\alpha, \rho) = (0.4, 0.5)$, is used throughout. We selected it on three tuning orders, G11 to G13, that do not overlap with the evaluation orders (Appendix B.5). Baseline settings and their sources are described in Appendix B.3. All experiments were run on an NVIDIA Quadro RTX 8000 GPU, and the timing protocol is described in Appendix B.6. Algorithm 1 in Appendix A.7 presents the complete procedure in pseudocode, and we will release the source code upon publication.

REFERENCES

Rishi Bommasani, Drew A. Hudson, Ehsan Adeli, Russ B. Altman, et al. On the opportunities and risks of foundation models. *arXiv preprint arXiv:2108.07258*, 2021.

Lukas Bossard, Matthieu Guillaumin, and Luc Van Gool. Food-101: Mining discriminative components with random forests. In *Proceedings of the European Conference on Computer Vision*, pp. 446–461, 2014.

Arslan Chaudhry, Marc'Aurelio Ranzato, Marcus Rohrbach, and Mohamed Elhoseiny. Efficient lifelong learning with A-GEM. In *International Conference on Learning Representations*, 2019.

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

Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Yunxuan Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, Albert Webson, Shixiang Shane Gu, Zhuyun Dai, Mirac Suzgun, Xinyun Chen, Aakanksha Chowdhery, Alex Castro-Ros, Marie Pellat, Kevin Robinson, Dasha Valter, Sharan Narang, Gaurav Mishra, Adams Yu, Vincent Y. Zhao, Yanping Huang, Andrew M. Dai, Hongkun Yu, Slav Petrov, Ed H. Chi, Jeff Dean, Jacob Devlin, Adam Roberts, Denny Zhou, Quoc V. Le, and Jason Wei. Scaling instruction-finetuned language models. *Journal of Machine Learning Research*, 25(70):1–53, 2024.

Mircea Cimpoi, Subhransu Maji, Iasonas Kokkinos, Sammy Mohamed, and Andrea Vedaldi. Describing textures in the wild. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3606–3613, 2014.

Tarin Clanuwat, Mikel Bober-Irizar, Asanobu Kitamoto, Alex Lamb, Kazuaki Yamamoto, and David Ha. Deep learning for classical japanese literature. *arXiv preprint arXiv:1812.01718*, 2018.

Adam Coates, Andrew Y. Ng, and Honglak Lee. An analysis of single-layer networks in unsupervised feature learning. In *Proceedings of the International Conference on Artificial Intelligence and Statistics*, pp. 215–223, 2011.

Gregory Cohen, Saeed Afshar, Jonathan Tapson, and André van Schaik. EMNIST: Extending MNIST to handwritten letters. In *International Joint Conference on Neural Networks*, pp. 2921–2926, 2017.

Matthias De Lange, Rahaf Aljundi, Marc Masana, Sarah Parisot, Xu Jia, Ales Leonardis, Gregory G. Slabaugh, and Tinne Tuytelaars. A continual learning survey: Defying forgetting in classification tasks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(7):3366–3385, 2022.

Jean-Antoine Désidéri. Multiple-gradient descent algorithm (MGDA) for multiobjective optimization. *Comptes Rendus. Mathématique*, 350(5–6):313–318, 2012.

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

Mehrdad Farajtabar, Navid Azizan, Alex Mott, and Ang Li. Orthogonal gradient descent for continual learning. In Silvia Chiappa and Roberto Calandra (eds.), *Proceedings of the International Conference on Artificial Intelligence and Statistics*, pp. 3762–3773, 2020.

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

Gemma Team, Morgane Riviere, Shreya Pathak, Pier Giuseppe Sessa, Cassidy Hardin, Surya Bhupatiraju, et al. Gemma 2: Improving open language models at a practical size. *arXiv preprint arXiv:2408.00118*, 2024.

Ian J. Goodfellow, Dumitru Erhan, Pierre Luc Carrier, Aaron C. Courville, Mehdi Mirza, Benjamin Hamner, William Cukierski, Yichuan Tang, David Thaler, Dong-Hyun Lee, Yingbo Zhou, Chetan Ramaiah, Fangxiang Feng, Ruifan Li, Xiaojie Wang, Dimitris Athanasakis, John Shawe-Taylor, Maxim Milakov, John Park, Radu Tudor Ionescu, Marius Popescu, Cristian Grozea, James Bergstra, Jingjing Xie, Lukasz Romaszko, Bing Xu, Chuang Zhang, and Yoshua Bengio. Challenges in representation learning: A report on three machine learning contests. In *Neural Information Processing*, pp. 117–124, 2013.

Yifei He, Siqi Zeng, Yuzheng Hu, Rui Yang, Tong Zhang, and Han Zhao. MergeBench: A benchmark for merging domain-specialized LLMs. In *Advances in Neural Information Processing Systems*, 2025.

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

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2022.

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

Pavel Izmailov, Dmitrii Podoprikhin, Timur Garipov, Dmitry P. Vetrov, and Andrew Gordon Wilson. Averaging weights leads to wider optima and better generalization. In *Proceedings of the Conference on Uncertainty in Artificial Intelligence*, pp. 876–885, 2018.

Xisen Jin, Xiang Ren, Daniel Preotiuc-Pietro, and Pengxiang Cheng. Dataless knowledge fusion by merging weights of language models. In *International Conference on Learning Representations*, 2023.

James Kirkpatrick, Razvan Pascanu, Neil C. Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A. Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, Demis Hassabis, Claudia Clopath, Dharshan Kumaran, and Raia Hadsell. Overcoming catastrophic forgetting in neural networks. *arXiv*, abs/1612.00796, 2016.

Jonathan Krause, Michael Stark, Jia Deng, and Li Fei-Fei. 3D object representations for fine-grained categorization. In *Proceedings of the IEEE International Conference on Computer Vision Workshops*, pp. 554–561, 2013.

Alex Krizhevsky and Geoffrey Hinton. Learning multiple layers of features from tiny images. Technical report, University of Toronto, 2009.

Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

Lihong Lin and Haidong Kang. Unlocking the potential of continual model merging: An ODE perspective. In *International Conference on Machine Learning*, 2026.

Zheda Mai, Ruiwen Li, Jihwan Jeong, David Quispe, Hyunwoo Kim, and Scott Sanner. Online continual learning in image classification: An empirical survey. *Neurocomputing*, 469:28–51, 2022.

Daniel Marczak, Simone Magistri, Sebastian Cygert, Bartłomiej Twardowski, Andrew D. Bagdanov, and Joost van de Weijer. No task left behind: Isotropic model merging with common and task-specific subspaces. In *Proceedings of the International Conference on Machine Learning*, pp. 43177–43199, 2025.

Michael Matena and Colin Raffel. Merging models with fisher-weighted averaging. In Sanmi Koyejo, S. Mohamed, A. Agarwal, Danielle Belgrave, K. Cho, and A. Oh (eds.), *Advances in Neural Information Processing Systems*, 2022.

Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Bo Wu, and Andrew Y. Ng. Reading digits in natural images with unsupervised feature learning. In *NIPS Workshop on Deep Learning and Unsupervised Feature Learning*, 2011.

Maria-Elena Nilsback and Andrew Zisserman. Automated flower classification over a large number of classes. In *Proceedings of the Indian Conference on Computer Vision, Graphics and Image Processing*, pp. 722–729, 2008.

German Ignacio Parisi, Ronald Kemker, Jose L. Part, Christopher Kanan, and Stefan Wermter. Continual lifelong learning with neural networks: A review. *Neural Networks*, 113:54–71, 2019.

Omkar M. Parkhi, Andrea Vedaldi, Andrew Zisserman, and C. V. Jawahar. Cats and dogs. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3498–3505, 2012.

Zihuan Qiu, Yi Xu, Chiyuan He, Fanman Meng, Linfeng Xu, Qingbo Wu, and Hongliang Li. MINGLE: Mixture of null-space gated low-rank experts for test-time continual model merging. In *Advances in Neural Information Processing Systems*, 2025.

Zihuan Qiu, Lei Wang, Yang Cao, Runtong Zhang, Bing Su, Yi Xu, Fanman Meng, Linfeng Xu, Qingbo Wu, and Hongliang Li. Null-space filtering for data-free continual model merging: Preserving stability, promoting plasticity. In *International Conference on Learning Representations*, 2026.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In Marina Meila and Tong Zhang (eds.), *Proceedings of the International Conference on Machine Learning*, pp. 8748–8763, 2021.

Sylvestre-Alvise Rebuffi, Alexander Kolesnikov, Georg Sperl, and Christoph H. Lampert. icarl: Incremental classifier and representation learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 5533–5542, 2017.

Andrei A. Rusu, Neil C. Rabinowitz, Guillaume Desjardins, Hubert Soyer, James Kirkpatrick, Koray Kavukcuoglu, Razvan Pascanu, and Raia Hadsell. Progressive neural networks. *arXiv preprint arXiv:1606.04671*, 2016.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Y. Ng, and Christopher Potts. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pp. 1631–1642, 2013.

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

George Stoica, Pratik Ramesh, Boglarka Ecsedi, Leshem Choshen, and Judy Hoffman. Model merging with SVD to tie the knots. In *International Conference on Learning Representations*, 2025.

Anke Tang, Li Shen, Yong Luo, Han Hu, Bo Du, and Dacheng Tao. Fusionbench: A comprehensive benchmark of deep model fusion. *arXiv preprint arXiv:2406.03280*, 2024.

Anke Tang, Enneng Yang, Li Shen, Yong Luo, Han Hu, Lefei Zhang, Bo Du, and Dacheng Tao. Merging on the fly without retraining: A sequential approach to scalable continual model merging. In *Advances in Neural Information Processing Systems*, 2025.

Bastiaan S. Veeling, Jasper Linmans, Jim Winkens, Taco Cohen, and Max Welling. Rotation equivariant CNNs for digital pathology. In *Medical Image Computing and Computer-Assisted Intervention – MICCAI*, pp. 210–218, 2018.

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.

Liyuan Wang, Xingxing Zhang, Hang Su, and Jun Zhu. A comprehensive survey of continual learning: Theory, method and application. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 46(8):5362–5383, 2024.

Shipeng Wang, Xiaorong Li, Jian Sun, and Zongben Xu. Training networks in null space of feature covariance for continual learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 184–193, 2021.

Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-MNIST: A novel image dataset for benchmarking machine learning algorithms. *arXiv preprint arXiv:1708.07747*, 2017.

Jianxiong Xiao, James Hays, Krista A. Ehinger, Aude Oliva, and Antonio Torralba. SUN database: Large-scale scene recognition from abbey to zoo. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3485–3492, 2010.

Prateek Yadav, Derek Tam, Leshem Choshen, Colin A. Raffel, and Mohit Bansal. Ties-merging: Resolving interference when merging models. In Alice Oh, Tristan Naumann, Amir Globerson, Kate Saenko, Moritz Hardt, and Sergey Levine (eds.), *Advances in Neural Information Processing Systems*, 2023.

Enneng Yang, Zhenyi Wang, Li Shen, Shiwei Liu, Guibing Guo, Xingwei Wang, and Dacheng Tao. AdaMerging: Adaptive model merging for multi-task learning. In *International Conference on Learning Representations*, 2024.

Enneng Yang, Anke Tang, Li Shen, Guibing Guo, Xingwei Wang, Xiaochun Cao, and Jie Zhang. Continual model merging without data: Dual projections for balancing stability and plasticity. In *Advances in Neural Information Processing Systems*, 2025.

Han-Chen Zhang, Zi-Hao Zhou, Mao-Lin Luo, Shimin Di, Min-Ling Zhang, and Tong Wei. DC-Merge: Improving model merging with directional consistency. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2026.