Task Singular Vectors: Reducing Task Interference in Model Merging

Antonio Andrea Gargiulo* Donato Crisostomi* Maria Sofia Bucarelli
Simone Scardapane Fabrizio Silvestri Emanuele Rodolà

Sapienza University of Rome

gargiulo.1769185@studenti.uniroma1.it {crisostomi, rodola}@di.uniroma1.it
simone.scardapane@uniroma1.it {bucarelli, fsilvestri}@diag.uniroma1.it

Abstract

Task Arithmetic has emerged as a simple yet effective method to merge models without additional training. However, by treating entire networks as flat parameter vectors, it overlooks key structural information and is susceptible to task interference. In this paper, we study task vectors at the layer level, focusing on task layer matrices and their singular value decomposition. In particular, we concentrate on the resulting singular vectors, which we refer to as Task Singular Vectors *(TSV). Recognizing that layer task matrices are often low-rank, we propose TSV-Compress (TSV-C), a simple procedure that compresses them to 10% of their original size while retaining 99% of accuracy. We further leverage this low-rank space to define a new measure of task interference based on the interaction of singular vectors from different tasks. Building on these findings, we introduce TSV-Merge (TSV-M), a novel model merging approach that combines compression with interference reduction, significantly outperforming existing methods.*

1. Introduction

The widespread availability of pre-trained models and public repositories has driven the development of techniques for efficiently combining and reusing existing models. Among these techniques, model merging approaches enable the creation of multi-task models without additional training. One popular approach, Task Arithmetic (TA) [22], stands out for its simplicity and effectiveness. However, by treating entire networks as high-dimensional vectors, TA and subsequent works overlook crucial structural information [9, 10, 24, 32, 38, 51, 53]. This flattened view limits these approaches to coarse-grained measures like cosine similarity for assessing inter-task interactions.

⋆ denotes equal contribution.

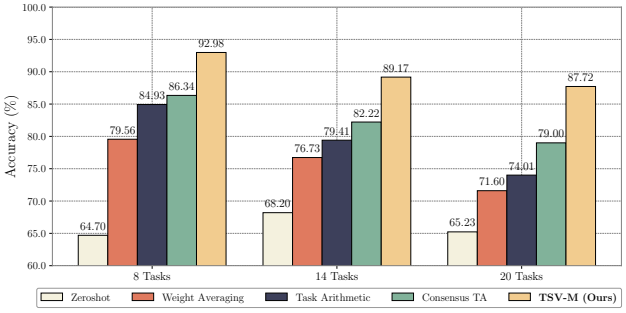

Figure 1. Mean accuracy of a ViT-L-14 merged over 8, 14, and 20 tasks respectively. By significantly surpassing existing methods, TSV-M establishes the new state of the art in model merging.

In this work, we instead focus on preserving and leveraging the natural structure of neural networks by examining weight differences at the layer level while retaining their matrix form wherever possible. Our approach begins by examining per-layer task *matrices*, which represent weight changes for each task, through singular value decomposition (SVD). This decomposition yields singular vectors and singular values that capture the most significant directions of variation within each layer. We term these singular vectors *Task Singular Vectors* (TSV), as they provide an interpretable basis for assessing task-specific contributions at each layer. Importantly, in accordance with recent literature on PEFT (cfr. Sec. 2), our analysis confirms that task matrices are inherently low-rank, meaning that only a small portion of TSVs is sufficient to represent the layer's function with high fidelity.

Building on this insight, we introduce TSV-Compress (TSV-C), a simple yet effective procedure to compress task vectors down to 10% of their original size while maintaining 99% of the original accuracy. Focusing on only a fraction of the most relevant TSVs for each task, TSV-Compress enables efficient storage and processing

of task-specific information without waiving performance.

Beyond compression, examining the interplay of TSVs across different tasks provides a geometrically informed framework for analyzing task interference at the individual layer level. By assessing how singular vectors from different tasks align or diverge within each layer, this approach offers a significantly more fine-grained understanding of inter-task interactions, going beyond global vector similarity metrics like cosine similarity. Building upon these principles, we introduce `TSV-Merge` (`TSV-M`), a novel model merging method that combines compression and task interference reduction. This is achieved by discarding the irrelevant singular vectors for each task and then reducing their inter-task interference with a whitening transformation over their similarity matrix. We empirically show our approach to effectively reduce task interference, and the reduction to be complementary to the compression step. We then evaluate our approach across several benchmarks and show it outperforms existing methods by an average of $\sim 15\%$ accuracy points, establishing the new state of the art by a large margin. We release all our code for research purposes[1].

In summary, our contribution is 4-fold:

- We study the singular value decomposition of per-layer task matrices, showing them to be low-rank and measuring the interplay of singular vectors from different tasks;
- We introduce `TSV-C`, a method that builds on this insight to compress task vectors by a factor of $10\times$ while preserving 99% of their original accuracy;
- We propose `TSV-M`, a model merging technique that complements compression with a task interference reduction step by applying a whitening transformation to decorrelate singular vectors across tasks;
- We conduct extensive experiments on multiple computer vision datasets, showing that `TSV-M` significantly outperforms existing model merging methods and provide in-depth analyses to uncover the underlying factors contributing to its success.

2. Related Work

Model Merging offers an efficient alternative to ensembling by combining existing models without the need for further training. Several approaches address this by determining the neuron permutations that align the models into a shared optimization basin, after which the models are combined using a straightforward averaging method. [1, 8, 25, 36, 43]. An alternative line of approaches focuses on the multi-task scenario, where a single pre-trained model is fine-tuned for different tasks [10, 22, 32, 40, 46, 48, 51, 53, 54]. The prerequisites for methods in this category are outlined in Table 1. `Task Arithmetic` [22] introduces the concept of *task vectors*, which are the weight differences between fine-tuned models and the pre-trained base model. By averaging these vectors, a merged multi-task model can be created; conversely, negating a task vector allows for forgetting a specific task. `TIES` [51] addresses redundancies in model parameters by first selecting the top-k most significant parameter changes and then constructing a sign vector based on the majority sign across all models. The latter is used to merge the task vectors disjointly, meaning the average is not computed when a parameter is zero or when parameters disagree in sign. Similarly, `DARE` [53] randomly resets redundant parameters to their pre-trained values and rescales the remaining parameters by a factor proportional to the dropped ones, aiming to reduce interference among tasks. `Fisher Merging` [32] and `RegMean` [24] merge models by performing weighted averaging, utilizing the Fisher information matrix and the inner product of input vectors, respectively. `Model Breadcrumbs` [10] focuses on merging only significant weights by discarding outliers and both minor and large perturbations in the fine-tuned parameters. More recently, Wang et al. [46] observed that tasks often utilize non-overlapping sets of weights. They propose two methods: the first, `TALL-Mask`, uses binary masks to activate important task-specific weights, requiring extra storage (as does our method `TSV-C`) for the masks. The second, `Consensus Merging`, leverages these masks to remove parameters that are important to less than two tasks. Like our approach, `TwinMerging` [31] and `SMILE` [44] apply SVD to layer task matrices. However, neglecting singular vector interference, they require a router to selectively activate a subset during inference.

Unlike existing approaches, our methods explicitly address task interference by leveraging the geometric structure of singular vectors to minimize interactions between task-specific parameters. Rather than averaging or selectively merging parameters as prior techniques do, we decorrelate singular vectors to directly reduce interference across tasks.

Table 1. Model merging approaches and their requirements.

Method	Additional training	Extra storage	Validat. data inputs	Validat. data labels
Weight Avg. [48]	×	×	×	×
Fisher-Merg. [32]	×	×	✓	×
RegMean [24]	×	×	✓	×
EMR-Merging [20]	×	✓	×	×
TALL-Mask [46]	×	✓	×	×
TSV-C (Ours)	×	✓	×	×
Task Arith. [22]	×	×	✓	✓
Ties-Merging [51]	×	×	✓	✓
AdaMerging [52]	✓	×	✓	×
Consensus-TA [46]	×	×	✓	✓
TSV-M (Ours)	×	×	×	×

[1] https://github.com/AntoAndGar/task_singular_vectors

Additionally, we achieve targeted compression by isolating each task's unique components, storing only essential parts, and ensuring minimal interference, a step beyond traditional SVD-based compression approaches.

SVD for Model Compression A significant body of work explores low-rank decompositions for fully-connected and convolutional layers [11, 12, 16, 23, 27], as well as tensor decompositions [12, 16, 26]. These approaches typically achieve a layer-by-layer factorization of a trained network, focusing on minimizing the difference between the original and the low-rank approximated weight matrices. This is accomplished using techniques like SVD [12, 16, 23, 27] or iterative optimization [23, 26]. While other methods such as weight quantization, knowledge distillation, and pruning exist, we utilize SVD not only for low-rank approximation and compression but also because its matrix decomposition properties make it particularly effective in analyzing and mitigating interference.

3. Task Singular Vectors

In this section, we introduce key concepts for understanding our approach, including a new measure of task interference that is central to our method.

3.1. Background

We build on `Task Arithmetic` (TA), which defines task vectors capturing the differences in model weights for individual tasks. Formally, the weights θ_{MT} of a multi-task model for T tasks are computed by aggregating the task-specific weight differences, or task vectors, as follows:

$$\theta_{\text{MT}} = \theta_{\text{pre}} + \alpha \frac{\sum_{i=1}^{T} \tau_i}{T}\,, \tag{1}$$

where θ_{pre} is the set of pretrained model weights, α is a scaling factor, and $\tau_i = \theta_{\text{ft}_i} - \theta_{\text{pre}}$ is the task vector for task i, with θ_{ft_i} being the fine-tuned weights for the task. Differently from TA, however, we consider these operations at the layer level. From this perspective, Eq. (1) becomes

$$\theta_{\text{MT}}^{(l)} = \theta_{\text{pre}}^{(l)} + \alpha \frac{\sum_{i=1}^{T} \Delta_i^{(l)}}{T}, \tag{2}$$

where $\theta_{\text{pre}}^{(l)}$ encodes the pretrained weights for layer l, and $\Delta_i^{(l)} = \theta_{\text{ft}_i}^{(l)} - \theta_{\text{pre}}^{(l)}$ is the task-specific weight difference for task i at layer l. When layer l has a matrix structure, we call its corresponding ${\Delta_i}^l$ the *per-layer task matrix* for task i. When this is not the case, our framework defaults to standard TA. For brevity, we will generally omit the layer index and refer to the layer-l task matrix Δ_i^l as Δ_i.

Decomposing layer task matrices

Treating layer-wise weights as structured entities rather than flattened vectors enables us to analyze their SVD, revealing low-rank properties and deeper insight into the inter-task interactions. Given two tasks i, j, we consider the SVD of their task matrices Δ_i and Δ_j at a generic layer:

$$\Delta_i = U_i \Sigma_i V_i^\top, \qquad \Delta_j = U_j \Sigma_j V_j^\top$$

where U_i, U_j and V_i, V_j are the matrices of left and right singular vectors respectively, and Σ_i, Σ_j are diagonal matrices of singular values. Due to their role in assessing task-specific contributions, we term the obtained singular vectors *Task Singular Vectors* (TSV).

Importantly for our treatment, we can equivalently write the aggregated task matrices in Eq. (2) in terms of their singular vectors and values. By defining $U = [U_1 \cdots U_T]$ as the column-wise concatenation of all the left TSVs, $V = [V_1 \cdots V_T]$ as the row-wise concatenation of the right TSVs, and Σ as the block diagonal matrix with $\{\Sigma_i\}_{i=1}^{T}$ along its diagonal, we can write:

$$M = U \Sigma V^\top = \sum_{i=1}^{T} U_i \Sigma_i V_i^\top = \sum_{i=1}^{T} \Delta_i\,. \tag{3}$$

In the simple case where $T = 2$, M would be given by:

$$M = \begin{bmatrix} U_1 & U_2 \end{bmatrix} \begin{bmatrix} \Sigma_1 & 0 \\ 0 & \Sigma_2 \end{bmatrix} \begin{bmatrix} V_1^\top \\ V_2^\top \end{bmatrix} = \Delta_1 + \Delta_2\,.$$

When concatenating singular components from different tasks, we violate certain properties of the SVD. Specifically, the matrices U and V become non-orthogonal because singular vectors from different tasks may overlap. Additionally, the singular values Σ_i from different tasks can vary significantly in magnitude, which may bias the merging process toward tasks with larger singular values.

3.2. Low-rank nature of layer task matrices

We start our analysis by studying the low-rank properties of per-layer task matrices. Interpreting SVD as a sum of rank-one matrices, for each task i, by Eckart-Young's theorem [14] we get the best approximation (in Frobenius norm) of each task matrix Δ_i by retaining only the top-k singular values and their corresponding vectors:

$$\hat{\Delta}_i = \sum_{j=1}^{k} \sigma_j^i u_j^i v_j^{i\top}. \tag{4}$$

As we show in Fig. 2, we find task matrices to be inherently low-rank: a small subset of TSVs is sufficient to represent the layer's function with high fidelity. In particular, even when preserving only 3% of the singular components per task, the mean accuracy drops by merely 1.5%. This is

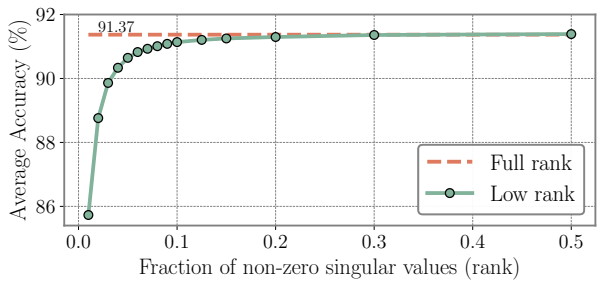

Figure 2. Mean absolute accuracy of the `ViT-B-32` model across increasing fractions of retained singular components, averaged over 20 tasks. The red line represents the average accuracy of the original fine-tuned models with full-rank task matrices, while the green line shows the accuracies using low-rank approximations.

noticeable considering that 97% of singular components in each layer matrix are discarded. Building on this insight, we propose in Sec. 4.1 a compression algorithm that maintains 99% of the accuracy while shrinking the task vectors to 10% of their original size.

Given this low-rank structure, it is natural to aggregate task matrices within their subspaces. We, therefore, obtain a reduced version of the aggregation matrix M (Eq. (3)) as:

$$\hat{M} = \sum_{i=1}^{T} \hat{U}_i \hat{\Sigma}_i \hat{V}_i^\top = \sum_{i=1}^{T} \sum_{j=1}^{k} \sigma_j^i u_j^i v_j^{i\top}, \tag{5}$$

where $\hat{U}_i$ and $\hat{V}_i$ contain the top-k left and right singular vectors for task i, respectively, and $\hat{\Sigma}_i$ is the diagonal matrix of the top-k singular values σ_j^i. This formulation highlights that the low-rank approximation $\hat{M}$ utilizes the most relevant singular components from each task to effectively combine the task-specific weight differences.

3.3. Singular Task interference

We hereby introduce a score of task interference based on the interplay of TSVs from different tasks, which we term *Singular Task Interference* (STI)

$$\text{STI}\left(\{\Delta_i\}_{i=1}^T\right) = \|(U^\top U - I)\Sigma(V^\top V - I)\|_1\,, \tag{6}$$

where U, Σ and V are obtained by concatenating the singular value decompositions $(\{U_i\}_{i=1}^T, \{\Sigma_i\}_{i=1}^T \{V_i\}_{i=1}^T)$ of per-layer task matrices $\{\Delta_i\}_{i=1}^T$ as detailed in Sec. 3.1. In the expression above, high inner product values for $U^\top U$ and $V^\top V$ imply a higher likelihood of interference, with minimal interference ideally yielding identity matrices.

The underlying intuition is that overlapping singular vectors suggest shared features in the weight space across tasks. Such overlap can introduce interference when models are merged, ultimately degrading performance on individual tasks. We refer to Fig. 3 for a real example over eight tasks.

4. Approach

We demonstrate below how TSVs can be used for both compression and task interference reduction.

4.1. TSV for compression

When the task is known or inferred (e.g. with a router, as in Mixture-of-Experts [15, 39] techniques), we can leverage the low-rank structure of per-layer task matrices (see Sec. 3.2) to effectively compress these ones while discarding task singular vectors from different tasks. In particular, given a known or inferred task index h, we set other task components to zero, reducing Eq. (5) to:

$$\hat{M} = \sum_{i=1}^{T} \mathbb{1}_{[i=h]} \sum_{j=1}^{k} \sigma_j^i u_j^i v_j^{i\top} = \sum_{j=1}^{k} \sigma_j^h u_j^h v_j^{h\top} = \hat{\Delta}_i\,. \tag{7}$$

This formula yields a low-rank approximation of Δ_h, where only the top k singular components of the task-specific matrix are retained. Here, the same number of components k is taken from each task to ensure that each layer is compressed by a factor of $\frac{1}{T}$ relative to the original dimension. Increasing k improves approximation but reduces the compression.

4.2. TSV for model merging

In scenarios where the task identity is not known beforehand, we address the standard model merging problem by combining Task Singular Vectors (TSVs) to create a single model that performs well jointly across all tasks.

To reduce our measure of task interference, we decorrelate the TSVs of different tasks (encoded as columns in $\hat{U}$ and $\hat{V}$) by whitening these matrices to minimize their correlations. This can be achieved by applying the transformation $X \mapsto X(X^\top X)^{-\frac{1}{2}}$ to both $\hat{U}$ and $\hat{V}$. For improved numerical stability, we reformulate this whitening as an orthogonal Procrustes problem, seeking the orthogonal matrix $\hat{U}_\perp$ that minimizes the projection error:

$$\min_{\hat{U}_\perp} \|\hat{U}_\perp - \hat{U}\|_F \quad \text{s.t. } \hat{U}_\perp^\top \hat{U}_\perp = I\,, \tag{8}$$

and similarly for $\hat{V}$. This problem admits a closed-form solution via the SVD $\hat{U} = PDQ^\top$, yielding $\hat{U}_\perp = PQ^\top$ (similarly for $\hat{V}_\perp$).

Proposition 4.1. *The transformations $X \mapsto X(X^\top X)^{-\frac{1}{2}}$ (whitening) and $X \mapsto PQ^\top$ (Procrustes), where $X = PDQ^\top$ is the SVD of X, are equivalent.*

Proof. Given $X = PDQ^\top$ and recalling that $D^\top D = D^2$ for diagonal matrices, simple algebraic manipulation yields $X^\top X = QD^2Q^\top$. It follows that $(X^\top X)^{-1/2} = QD^{-1}Q^\top$, therefore the whitening transformation can be rewritten as $X(X^\top X)^{-1/2} = (PDQ^\top)(QD^{-1}Q^\top) = PDID^{-1}Q^\top = PQ^\top$, completing the proof. □

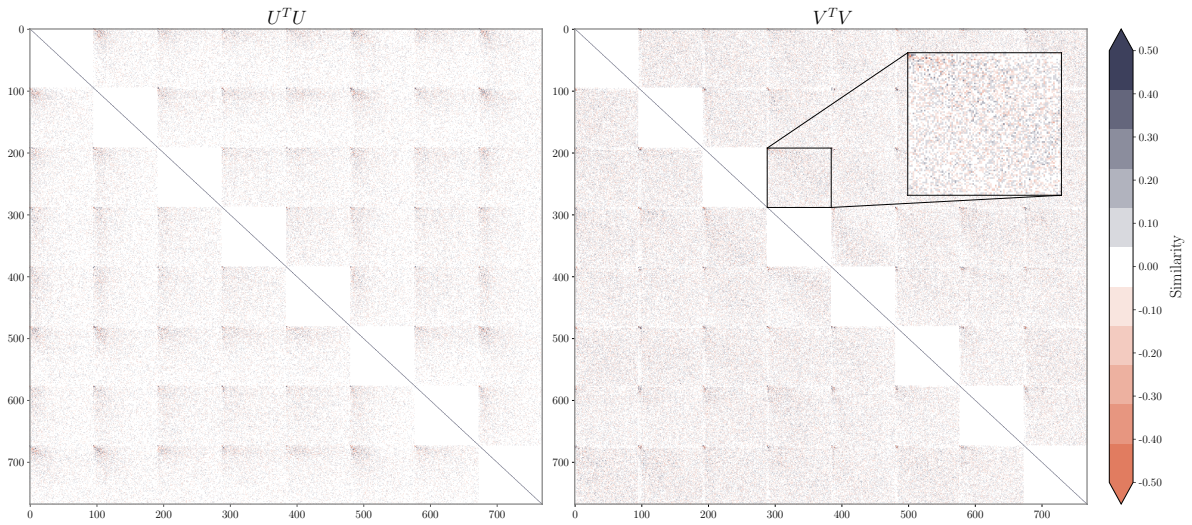

Figure 3. Visualization of task interference among 8 tasks computed on the first attention layer of a `ViT-B-32`. The diagonal blocks display intra-task similarities, while the off-diagonal blocks illustrate inter-task similarities. The zoomed-in section highlights the interaction between the right singular vectors of the 3rd and 4th tasks.

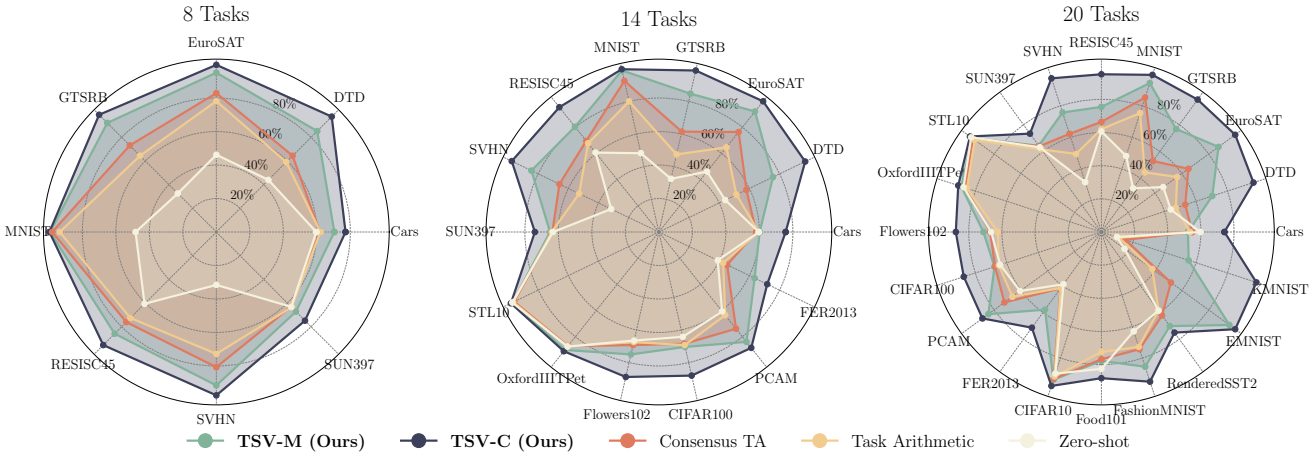

Figure 4. Absolute accuracy of a `ViT-B-32` merged over 8, 14, and 20 tasks, respectively.

Once the (rank-reduced) TSV matrices $\hat{U}$ and $\hat{V}$ are decorrelated, we reconstruct the merged layer as in Eq. (5); see Algorithm 1 for the complete steps.

5. Results

We evaluate our approaches over three different suites of tasks having cardinality 8, 14, and 20, respectively. The first one, introduced in [22], consists of datasets: `Cars` [28], `DTD` [4], `EuroSAT` [19], `GTSRB` [42], `MNIST` [30], `RESISC45` [3], `SUN397` [50], and `SVHN` [33]. The benchmark with 14 tasks builds on the preceding one, incorporating six additional datasets: `CIFAR100` [29], `STL10` [6], `Flowers102` [34], `OxfordIIITPet` [35], `PCAM` [45], and `FER2013` [18]. Finally, the 20-tasks benchmark includes the preceding 14 plus the following six: `EMNIST` [7], `CIFAR10` [29], `Food101` [2], `FashionMNIST` [49], `RenderedSST2` [41], and `KMNIST` [5].

5.1. Model merging results

We evaluate our method on three variants of the `CLIP` [37] model, each employing a different size of `ViT` [13] visual encoder: `ViT-B-32`, `ViT-B-16`, and `ViT-L-14`. The main benchmarks involve merging 8, 14, and 20 tasks, mirroring the experimental setup described in Wang et al. [46]. We compare our approach against several training-free model merging methods, including weight averaging, `Task Arithmetic` [22], and `Consensus Merging` [46]. For reference, we include the performance of zero-

Method	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
Zeroshot	$48.26_{(53.59)}$	$57.21_{(63.69)}$	$56.10_{(62.41)}$	$55.34_{(59.34)}$	$61.28_{(66.19)}$	$59.73_{(64.52)}$	$64.70_{(68.00)}$	$68.20_{(72.15)}$	$65.23_{(68.99)}$
Weight Averaging	$66.34_{(72.13)}$	$64.34_{(71.12)}$	$61.04_{(67.53)}$	$72.22_{(76.60)}$	$69.46_{(74.82)}$	$65.31_{(70.36)}$	$79.56_{(83.15)}$	$76.73_{(81.10)}$	$71.60_{(75.60)}$
Task Arithmetic	$70.79_{(76.55)}$	$65.32_{(72.09)}$	$60.52_{(66.79)}$	$75.41_{(79.58)}$	$70.52_{(75.89)}$	$65.78_{(70.76)}$	$84.93_{(88.65)}$	$79.41_{(83.95)}$	$74.01_{(78.07)}$
Consensus TA	$75.03_{(80.84)}$	$70.39_{(77.36)}$	$65.43_{(71.98)}$	$79.39_{(83.86)}$	$74.39_{(79.92)}$	$69.76_{(74.93)}$	$86.34_{(90.08)}$	$82.22_{(86.94)}$	$79.00_{(83.22)}$
TSV-M (Ours)	$\mathbf{85.86_{(92.31)}}$	$\mathbf{80.06_{(87.88)}}$	$\mathbf{77.07_{(84.29)}}$	$\mathbf{89.01_{(93.94)}}$	$\mathbf{84.58_{(91.01)}}$	$\mathbf{80.57_{(86.45)}}$	$\mathbf{92.98_{(96.98)}}$	$\mathbf{89.17_{(94.43)}}$	$\mathbf{87.72_{(92.50)}}$

Table 2. Average absolute accuracy results on model merging benchmarks; subscript (in parentheses) is the normalized average accuracy.

Algorithm 1 TSV-Merge.

Require: Task matrices $\Delta_1, \ldots, \Delta_T$, scaling factor α
Ensure: Merged model weights θ_{MT}

for $i = 1$ to T **do**
  Compute SVD: $\Delta_i = U_i \Sigma_i V_i^\top$
  Retain first $\frac{1}{T}$ singular components of U_i, Σ_i, and V_i
end for
Concatenate the matrices:
  $U \leftarrow [U_1 \,|\, U_2 \,|\, \ldots \,|\, U_T]$
  $\Sigma \leftarrow \text{block-diag}(\Sigma_1, \Sigma_2, \ldots, \Sigma_T)$
  $V \leftarrow [V_1 \,|\, V_2 \,|\, \ldots \,|\, V_T]$
Compute the SVD of U and V:
  $U = P_U D_U Q_U^\top \qquad V = P_V D_V Q_V^\top$
Obtain the orthogonal matrices:
  $U_\perp = P_U Q_U^\top \qquad V_\perp = P_V Q_V^\top$
Reconstruct the merged matrix:
  $\hat{M} \leftarrow U_\perp \Sigma V_\perp^\top$
Construct merged model weights:
  $\theta_{\text{MT}} \leftarrow \theta_{\text{pre}} + \alpha \hat{M}$
return θ_{MT}

shot models in Table 2 to represent the minimum achievable accuracy, and the average of individually fine-tuned models in Table 3 to indicate the maximum potential gains. Performance metrics were assessed using both the average absolute accuracy and the average normalized accuracy, calculated as detailed in Appendix B.1.

As presented in Table 2, our method achieves state-of-the-art results across all benchmarks, regardless of the ViT size or the number of tasks involved. Notably, the most significant improvements were observed with the smaller ViT-B-32 encoder, where our approach outperforms Task Arithmetic and Consensus TA by an average of $+15.45\%$ and $+10.71\%$ absolute accuracy, respectively. Furthermore, when utilizing the ViT-L-14 model, our method attains an average normalized accuracy of 96.98%. This indicates that we can effectively replace eight individual task-specific models with a single merged model, incurring only a -3.02% reduction in average accuracy. These results highlight the promise of our model merging technique as a cost-efficient alternative to mixtures of experts, multi-task learning, and ensembling methods. We report in Fig. 4 the per-dataset accuracies.

5.2. Compression results

In Table 3 we report the results for our compression algorithm TSV-C. We compare our method with TALL-masks [46], which stores very sparse binary masks for each task. As shown in Table 3, our method always retains more than 99% of the original accuracy for all considered benchmarks and models. The results of TSV-C are comparable to those of TALL-masks [46], with the former slightly underperforming TALL-masks for small ViTs and the latter gaining the upper hand in the larger-scale one. In general, however, the two approaches differ by less than 1% accuracy on average. Regarding the storage, our method stores only the top $\frac{1}{T}$ singular vectors per task, which leads to a fixed storage requirement of approximately twice the size of the original model, regardless of the number of tasks. By storing binary masks, TALL-masks instead results in a storage size that varies with their compressibility, increasing with the number of tasks. This variability can lead to unpredictable storage requirements and may diminish compression benefits when masks are less compressible.

Our results show that TSV-C effectively balances compression and performance by leveraging the most significant TSVs. Maintaining near-original accuracy, our approach is particularly suitable for scenarios where storage constraints are critical but high model performance is still required.

6. Analysis

In this section, we begin by conducting an ablation study to assess the contributions of interference reduction and low-rank compression to the overall performance. Next, we study how task interference varies across layers of different depths. Finally, we empirically show that our method does not require tuning a scaling coefficient.

6.1. Ablation study

TSV-M combines low-rank approximation with task interference reduction. To evaluate the contribution of each component, we conducted an ablation study summarized in Table 4. Applying low-rank approximation alone to the layer task matrices results in worse performance than vanilla Task Arithmetic, shown in the first row where

Method	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
Finetuned	$92.83_{(100)}$	$90.88_{(100)}$	$91.37_{(100)}$	$94.64_{(100)}$	$92.76_{(100)}$	$93.17_{(100)}$	$95.81_{(100)}$	$94.29_{(100)}$	$94.73_{(100)}$
TALL-Mask+TIES	$93.13_{(100.37)}$	$90.92_{(100.04)}$	$91.11_{(99.70)}$	$94.68_{(100.04)}$	$92.69_{(99.90)}$	$93.05_{(99.87)}$	$95.96_{(100.15)}$	$93.40_{(99.09)}$	$93.91_{(99.16)}$
TSV-C (Ours)	$92.62_{(99.74)}$	$90.29_{(99.28)}$	$90.64_{(99.14)}$	$94.47_{(99.79)}$	$92.25_{(99.41)}$	$92.53_{(99.27)}$	$95.68_{(99.85)}$	$94.04_{(99.72)}$	$94.42_{(99.66)}$

Table 3. Average absolute accuracy results across all compression benchmarks for different models and varying number of tasks, subscript (in parentheses) the normalized average accuracies. Our TSV-C compression method consistently achieves over 99% of the original finetuned models' accuracy while significantly reducing storage requirements.

Low-rank approx.	Interf. reduction	ViT-B-32		
		8 tasks	14 tasks	20 tasks
×	×	76.5 (+0.0)	72.1 (+0.0)	66.8 (+0.0)
✓	×	75.2 (-1.3)	71.0 (-1.1)	66.3 (-0.5)
×	✓	82.6 (+7.4)	75.7 (+4.7)	69.9 (+3.6)
✓	✓	92.3 (+9.7)	87.9 (+12.2)	84.3 (+14.4)

Table 4. Comparison of different versions of Task Arithmetic, comprising either the low-rank approximation step, the interference reduction step, or both. The method performing both corresponds to the proposed TSV-Merge.

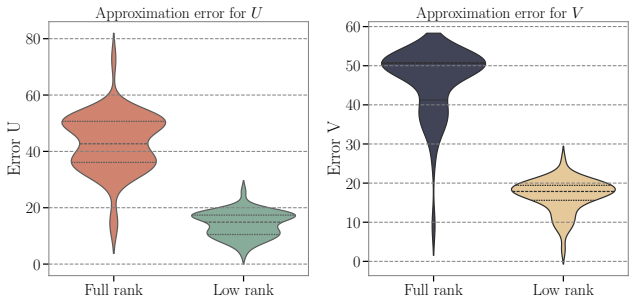

Figure 5. Approximation error from the orthogonalization of the TSVs through Procrustes for the ViT-B-32 model across 8 tasks. The violin plots represent layer-wise approximation error distributions for U and V in both full-rank and low-rank cases.

neither component is applied. In contrast, applying task interference reduction alone, while keeping full-rank matrices, significantly improves performance, with gains ranging from $+3.1\%$ to $+6.1\%$. However, the best results are achieved only when both steps are combined, yielding substantial performance improvements of $+15.8\%$ to $+17.5\%$.

To explain why interference reduction applied to low-rank approximations in TSV-M outperforms its application to full-rank matrices, we analyze the errors introduced during the orthogonalization of the full-rank U and V matrices. As shown in Fig. 5, orthogonalizing full-rank matrices incurs significant approximation errors, measured by the sum of the Frobenius norms of reconstruction errors for the concatenated U and V matrices across all layers.

For example, in the ViT-B-32 model with 8 tasks, the full-rank setting exhibits a wider error distribution with a higher average, indicating greater variability and larger approximation discrepancies. In contrast, the low-rank setting produces a more compact and lower error distribution, suggesting better consistency in approximation across layers. In the following, we prove that this is not specific to the chosen model but a general property of our approach. The proof can be found in Appendix C.3.

Theorem 6.1. *Let $T \in \mathbb{N}$ such that $T > 4$. Define $U = [U_1, \ldots, U_T]$ as the matrix obtained by concatenating T orthogonal matrices U_i, each of shape $n \times n$. Let $\widehat{U} = [\widehat{U}_1, \ldots, \widehat{U}_T]$ be the matrix formed by truncating each U_i to its first k columns. Denote by X and $\widehat{X}$ the matrices resulting from Procrustes orthonormalization of U and $\widehat{U}$, respectively. If $k \leq n\frac{T-2\sqrt{T}}{T}$, then*

$$\|U - X\|_F \geq \|\widehat{U} - \widehat{X}\|_F \,.$$

Intuitively, the theorem asserts that the approximation error introduced by orthogonalization via Procrustes is smaller when applied to the concatenation of truncated matrices, compared to the concatenation of the original full-rank matrices, provided the rank of the truncated matrices satisfies certain conditions. To better understand the statement of the theorem, consider the case $T = 10$, as in our scenario. The term $\frac{T-2\sqrt{T}}{T} = \frac{10-2\sqrt{10}}{10} \geq \frac{1}{3}$. In our case, we choose $k \approx \frac{n}{T}$. For $T = 10$, this gives $k \approx \frac{n}{10}$, which is indeed smaller than $\frac{n}{3}$, satisfying the condition $k \leq \frac{1}{3}n$.

We emphasize that other orthogonalization methods, such as Gram-Schmidt, are ineffective in this context because they do not preserve the overall structure of the original data. Gram-Schmidt sequentially orthogonalizes vectors without minimizing deviation from the original set, potentially resulting in significant distortions.

Finally, the substantial reduction in approximation error suggests that, while the advantages of interference reduction are considerable, they may be offset by the errors introduced when applied to full-rank matrices. In contrast, starting with low-rank approximations captures the essential information of each task, making the orthogonalization step less costly in terms of approximation error.

6.2. Per-layer task interference

We analyze task interference on a per-layer basis using Eq. (6). As shown in Fig. 6, interference is highest in the

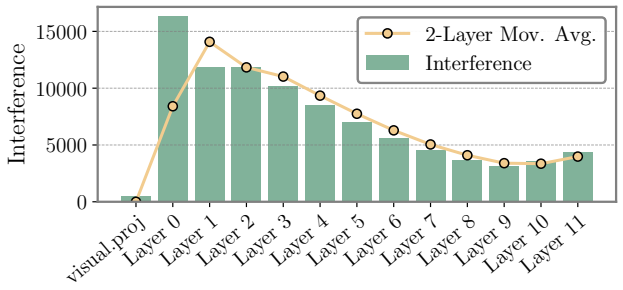

Figure 6. Singular Task Interference (STI) across layers in a `ViT-B-32` for 20 tasks. STI is high in early layers sharing common knowledge and lower in more specialized ones.

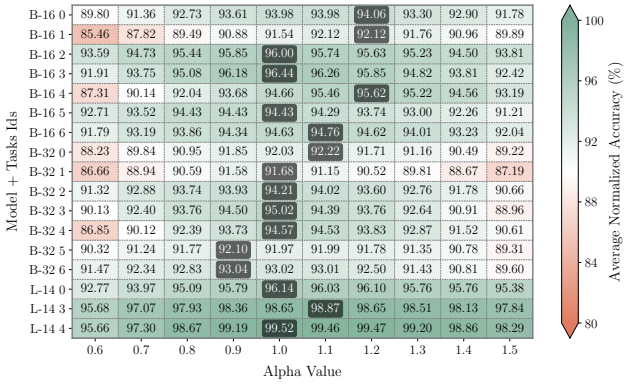

Figure 7. Best average normalized accuracy for different alpha values. The vertical labels indicate different sets of 8 tasks.

initial transformer layers and decreases significantly in the deeper layers. This aligns with the understanding that early, generalized layers capture common features across tasks, increasing the potential for conflict, while deeper layers are more specialized for specific tasks, reducing shared representation and interference. For brevity, we grouped layers within each transformer block in our analysis; each "Layer n" in Fig. 6 includes two attention matrices and two MLP matrices. A layer-by-layer analysis is provided in Fig. 14.

6.3. Choice of the interpolation coefficient

The aggregation in Eq. (2) involves a scaling coefficient α, typically tuned using a validation set, that can have a significant impact on the accuracy of the final model. However, as shown in Fig. 7, our approach consistently achieves the best results with $\alpha = 1.0$, indicating that no additional scaling is necessary. Although purely empirical, this finding spares further tuning and validation data, thereby enhancing the practicality and efficiency of our method.

6.4. Effect of task interference reduction

We report in Fig. 8 the interference of a `ViT-B-32` model both before and after the application of our `TSV-M` pipeline,

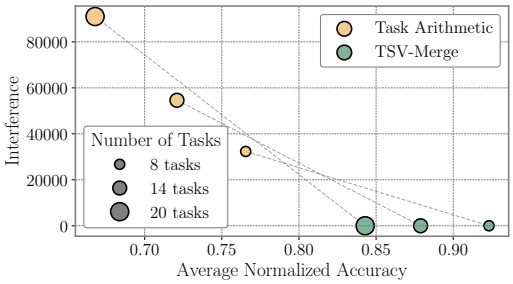

Figure 8. Singular Task Interference (STI) and average normalized accuracy for `Task Arithmetic` and `TSV-Merge` on the `ViT-B-32` model, evaluated across merges of 8, 14, and 20 tasks.

assessed across three distinct sets of tasks with different cardinalities. In all instances, we observed that a reduction in interference is closely associated with a significant gain in accuracy. This provides empirical evidence for the effectiveness of our approach in minimizing interference and, ultimately, enhancing the merging process.

7. Conclusions

In this paper, we tackled the problem of model merging by analyzing the SVD of per-layer task matrices, confirming their inherent low-rank structure and leveraging their singular vectors to define task interference.

Building on the former insight, we introduced `TSV-Compress` (`TSV-C`), a model compression algorithm that reduces task vectors to 10% of their original size while retaining 99% of the original accuracy. Unlike existing methods, our approach maintains a constant storage requirement, independent of the number of tasks. We then developed `TSV-Merge` (`TSV-M`), combining low-rank approximation with interference reduction to create a novel model merging technique. Achieving normalized accuracy up to 97%, our method offers a storage-efficient alternative to MoEs, ensembles, and multi-task learning.

Our extensive evaluation revealed that while interference reduction contributes most to performance improvements, the optimal results are achieved in combination with compression. We also observed that task interference decreases in deeper layers, and importantly, that our method does not require tuning a scaling coefficient.

Exploring alternative methods for task importance and rank approximation could be a valuable direction for future work. Currently, we use a uniform rank across tasks, approximating each task vector to $\frac{1}{T}$ of the layer's dimensionality. However, techniques like those from Idelbayev and Carreira-Perpinán [21] or optimal hard thresholds from Gavish and Donoho [17] could be applied to select the optimal rank for each task individually.

Acknowledgments

This work was partially supported by PRIN 2022 project 20227YET9B "AdVVent" CUP code B53D23012830006 and by projects FAIR (PE0000013) and SERICS (PE00000014) under the MUR National Recovery and Resilience Plan funded by the European Union - NextGenerationEU. This work was moreover partially supported by the project NEREO (Neural Reasoning over Open Data) project funded by the Italian Ministry of Education and Research (PRIN) Grant no. 2022AEFHAZ.

References

[1] Samuel Ainsworth, Jonathan Hayase, and Siddhartha Srinivasa. Git re-basin: Merging models modulo permutation symmetries. In *The Eleventh International Conference on Learning Representations*, 2023. 2

[2] Lukas Bossard, Matthieu Guillaumin, and Luc Van Gool. Food-101 – Mining Discriminative Components with Random Forests. In *Computer Vision – ECCV 2014*, pages 446–461, Cham, 2014. Springer International Publishing. 5

[3] 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. Conference Name: Proceedings of the IEEE. 5

[4] Mircea Cimpoi, Subhransu Maji, Iasonas Kokkinos, Sammy Mohamed, and Andrea Vedaldi. Describing Textures in the Wild. In *2014 IEEE Conference on Computer Vision and Pattern Recognition*, pages 3606–3613, Columbus, OH, USA, 2014. IEEE. 5

[5] Tarin Clanuwat, Mikel Bober-Irizar, Asanobu Kitamoto, Alex Lamb, Kazuaki Yamamoto, and David Ha. Deep learning for classical japanese literature. *CoRR*, abs/1812.01718, 2018. 5

[6] Adam Coates, Andrew Ng, and Honglak Lee. An Analysis of Single-Layer Networks in Unsupervised Feature Learning. In *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, pages 215–223. JMLR Workshop and Conference Proceedings, 2011. ISSN: 1938-7228. 5

[7] Gregory Cohen, Saeed Afshar, Jonathan Tapson, and André van Schaik. EMNIST: Extending MNIST to handwritten letters. In *2017 International Joint Conference on Neural Networks (IJCNN)*, pages 2921–2926, 2017. ISSN: 2161-4407. 5

[8] Donato Crisostomi, Marco Fumero, Daniele Baieri, Florian Bernard, and Emanuele Rodolà. c^2m^3: Cycle-consistent multi-model merging. In *Advances in Neural Information Processing Systems*, 2025. 2

[9] Nico Daheim, Thomas Möllenhoff, Edoardo Ponti, Iryna Gurevych, and Mohammad Emtiyaz Khan. Model merging by uncertainty-based gradient matching. In *The Twelfth International Conference on Learning Representations*, 2024. 1

[10] MohammadReza Davari and Eugene Belilovsky. Model breadcrumbs: Scaling multi-task model merging with sparse masks. In *European Conference on Computer Vision*, pages 270–287. Springer, 2025. 1, 2

[11] Misha Denil, Babak Shakibi, Laurent Dinh, Marc'Aurelio Ranzato, and Nando De Freitas. Predicting parameters in deep learning. *Advances in neural information processing systems*, 26, 2013. 3

[12] Emily L Denton, Wojciech Zaremba, Joan Bruna, Yann LeCun, and Rob Fergus. Exploiting linear structure within convolutional networks for efficient evaluation. *Advances in neural information processing systems*, 27, 2014. 3, 18

[13] 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, 2021. 5

[14] Carl Eckart and G. Marion Young. The approximation of one matrix by another of lower rank. *Psychometrika*, 1:211–218, 1936. 3

[15] David Eigen, Marc'Aurelio Ranzato, and Ilya Sutskever. Learning factored representations in a deep mixture of experts. In *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Workshop Track Proceedings*, 2014. 4

[16] Timur Garipov, Dmitry Podoprikhin, Alexander Novikov, and Dmitry Vetrov. Ultimate tensorization: compressing convolutional and fc layers alike. *arXiv preprint arXiv:1611.03214*, 2016. 3

[17] Matan Gavish and David L. Donoho. The optimal hard threshold for singular values is $4/\sqrt{3}$. *IEEE Transactions on Information Theory*, 60(8):5040–5053, 2014. 8

[18] Ian J. Goodfellow, Dumitru Erhan, Pierre Luc Carrier, Aaron Courville, Mehdi Mirza, Ben Hamner, Will 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 Ionescu, Marius Popescu, Cristian Grozea, James Bergstra, Jingjing Xie, Lukasz Romaszko, Bing Xu, Zhang Chuang, and Yoshua Bengio. Challenges in Representation Learning: A Report on Three Machine Learning Contests. In *Neural Information Processing*, pages 117–124, Berlin, Heidelberg, 2013. Springer. 5

[19] 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. Conference Name: IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing. 5

[20] Chenyu Huang, Peng Ye, Tao Chen, Tong He, Xiangyu Yue, and Wanli Ouyang. EMR-merging: Tuning-free high-performance model merging. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. 2

[21] Yerlan Idelbayev and Miguel A Carreira-Perpinán. Low-rank compression of neural nets: Learning the rank of each layer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8049–8059, 2020. 8

[22] Gabriel Ilharco, Marco Tulio Ribeiro, Mitchell Wortsman, Ludwig Schmidt, Hannaneh Hajishirzi, and Ali Farhadi. Editing models with task arithmetic. In *The Eleventh International Conference on Learning Representations*, 2023. 1, 2, 5, 13

[23] Max Jaderberg, Andrea Vedaldi, and Andrew Zisserman. Speeding up convolutional neural networks with low rank expansions. In *British Machine Vision Conference, BMVC 2014, Nottingham, UK, September 1-5, 2014*. BMVA Press, 2014. 3

[24] Xisen Jin, Xiang Ren, Daniel Preotiuc-Pietro, and Pengxiang Cheng. Dataless knowledge fusion by merging weights of language models. In *The Eleventh International Conference on Learning Representations*, 2023. 1, 2

[25] Keller Jordan, Hanie Sedghi, Olga Saukh, Rahim Entezari, and Behnam Neyshabur. REPAIR: REnormalizing permuted activations for interpolation repair. In *The Eleventh International Conference on Learning Representations*, 2023. 2

[26] Yong-Deok Kim, Eunhyeok Park, Sungjoo Yoo, Taelim Choi, Lu Yang, and Dongjun Shin. Compression of deep convolutional neural networks for fast and low power mobile applications. In *Proc. of the 4th Int. Conf. Learning Representations (ICLR 2016), San Juan, Puerto Rico, May 2–4*, 2016. 3

[27] Tamara G Kolda and Brett W Bader. Tensor decompositions and applications. *SIAM review*, 51(3):455–500, 2009. 3

[28] Jonathan Krause, Michael Stark, Jia Deng, and Li Fei-Fei. 3D Object Representations for Fine-Grained Categorization. In *2013 IEEE International Conference on Computer Vision Workshops*, pages 554–561, Sydney, Australia, 2013. IEEE. 5

[29] Alex Krizhevsky and Geoffrey Hinton. Learning multiple layers of features from tiny images. Technical Report 0, University of Toronto, Toronto, Ontario, 2009. 5

[30] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998. 5

[31] Zhenyi Lu, Chenghao Fan, Wei Wei, Xiaoye Qu, Dangyang Chen, and Yu Cheng. Twin-merging: Dynamic integration of modular expertise in model merging. *Advances in Neural Information Processing Systems*, 37:78905–78935, 2024. 2

[32] Michael S Matena and Colin A Raffel. Merging models with fisher-weighted averaging. *Advances in Neural Information Processing Systems*, 35:17703–17716, 2022. 1, 2

[33] 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*, 2011. 5

[34] Maria-Elena Nilsback and Andrew Zisserman. Automated Flower Classification over a Large Number of Classes. In *2008 Sixth Indian Conference on Computer Vision, Graphics & Image Processing*, pages 722–729, 2008. 5

[35] Omkar M Parkhi, Andrea Vedaldi, Andrew Zisserman, and C. V. Jawahar. Cats and dogs. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pages 3498–3505, 2012. ISSN: 1063-6919. 5

[36] Fidel A Guerrero Peña, Heitor Rapela Medeiros, Thomas Dubail, Masih Aminbeidokhti, Eric Granger, and Marco Pedersoli. Re-basin via implicit sinkhorn differentiation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023. 2

[37] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR, 2021. 5

[38] Alexandre Rame, Matthieu Kirchmeyer, Thibaud Rahier, Alain Rakotomamonjy, Patrick Gallinari, and Matthieu Cord. Diverse weight averaging for out-of-distribution generalization. *Advances in Neural Information Processing Systems*, 35:10821–10836, 2022. 1

[39] Noam Shazeer, *Azalia Mirhoseini, *Krzysztof Maziarz, Andy Davis, Quoc Le, Geoffrey Hinton, and Jeff Dean. Outrageously large neural networks: The sparsely-gated mixture-of-experts layer. In *International Conference on Learning Representations*, 2017. 4

[40] Li Shen, Anke Tang, Enneng Yang, Guibing Guo, Yong Luo, Lefei Zhang, Xiaochun Cao, Bo Du, and Dacheng Tao. Efficient and effective weight-ensembling mixture of experts for multi-task model merging. *arXiv preprint arXiv:2410.21804*, 2024. 2

[41] Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew 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*, pages 1631–1642, Seattle, Washington, USA, 2013. Association for Computational Linguistics. 5

[42] Johannes Stallkamp, Marc Schlipsing, Jan Salmen, and Christian Igel. The German Traffic Sign Recognition Benchmark: A multi-class classification competition. In *The 2011 International Joint Conference on Neural Networks*, pages 1453–1460, 2011. ISSN: 2161-4407. 5

[43] George Stoica, Daniel Bolya, Jakob Brandt Bjorner, Pratik Ramesh, Taylor Hearn, and Judy Hoffman. Zipit! merging models from different tasks without training. In *The Twelfth International Conference on Learning Representations*, 2024. 2

[44] Anke Tang, Li Shen, Yong Luo, Shuai Xie, Han Hu, Lefei Zhang, Bo Du, and Dacheng Tao. Smile: Zero-shot sparse mixture of low-rank experts construction from pre-trained foundation models. *arXiv preprint arXiv:2408.10174*, 2024. 2

[45] 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 2018*, pages 210–218, Cham, 2018. Springer International Publishing. 5

[46] Ke Wang, Nikolaos Dimitriadis, Guillermo Ortiz-Jimenez, François Fleuret, and Pascal Frossard. Localizing task information for improved model merging and compression. In *Forty-first International Conference on Machine Learning*, 2024. 2, 5, 6, 12, 13

[47] B. P. Welford. Note on a method for calculating corrected sums of squares and products. *Technometrics*, 4(3):419–420, 1962. 12

[48] Mitchell Wortsman, Gabriel Ilharco, Samir Ya Gadre, Rebecca Roelofs, Raphael Gontijo-Lopes, Ari S Morcos, Hongseok Namkoong, Ali Farhadi, Yair Carmon, Simon Kornblith, and Ludwig Schmidt. Model soups: averaging weights of multiple fine-tuned models improves accuracy without increasing inference time. In *Proceedings of the 39th International Conference on Machine Learning*, pages 23965–23998. PMLR, 2022. 2

[49] Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-MNIST: a Novel Image Dataset for Benchmarking Machine Learning Algorithms, 2017. arXiv:1708.07747 [cs, stat]. 5

[50] Jianxiong Xiao, Krista A. Ehinger, James Hays, Antonio Torralba, and Aude Oliva. SUN Database: Exploring a Large Collection of Scene Categories. *International Journal of Computer Vision*, 119(1):3–22, 2016. 5

[51] Prateek Yadav, Derek Tam, Leshem Choshen, Colin A Raffel, and Mohit Bansal. Ties-merging: Resolving interference when merging models. *Advances in Neural Information Processing Systems*, 36, 2024. 1, 2

[52] Enneng Yang, Zhenyi Wang, Li Shen, Shiwei Liu, Guibing Guo, Xingwei Wang, and Dacheng Tao. Adamerging: Adaptive model merging for multi-task learning. In *The Twelfth International Conference on Learning Representations*, 2024. 2, 18

[53] Le Yu, Bowen Yu, Haiyang Yu, Fei Huang, and Yongbin Li. Language models are super mario: Absorbing abilities from homologous models as a free lunch. In *Forty-first International Conference on Machine Learning*, 2024. 1, 2

[54] Luca Zhou, Daniele Solombrino, Donato Crisostomi, Maria Sofia Bucarelli, Fabrizio Silvestri, and Emanuele Rodolà. Atm: Improving model merging by alternating tuning and merging, 2024. 2