A. Illustrative example: merging two tasks with rank-1 approximation

Consider merging two distinct tasks by selecting only the first singular vector and singular value from the SVD for each task. This setting yields the following setup for each layer L_i:

$$\begin{bmatrix} | & | \\ u_{1_{L_i}} & u_{2_{L_i}} \\ | & | \end{bmatrix} \begin{bmatrix} \sigma_{1_{L_i}} & 0 \\ 0 & \sigma_{2_{L_i}} \end{bmatrix} \begin{bmatrix} - & v_{1_{L_i}}^T & - \\ - & v_{2_{L_i}}^T & - \end{bmatrix}$$

In this formulation, $u_{1_{L_i}}$ originates from task 1 and $u_{2_{L_i}}$ from task 2, with analogous assignments for the singular vectors v and singular values σ.

To elucidate the interaction between tasks, we examine three distinct cases, considering a single layer, thereby omitting the layer index L_i:

1. **Orthogonal Singular Vectors:** when u_1 and u_2 (respectively v) are orthogonal, the similarity matrix $U^\top U$ (respectively $V^\top V$) is given by:

$$\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

   The zeroes in the off-diagonal elements indicate no interference between the tasks. Consequently, the orthogonal components derived from different tasks operate independently, ensuring that each task does not affect the other.

2. **Collinear Singular Vectors:** when u_1 and u_2 (respectively v) are collinear, either aligned in the same direction (angle of 0 degrees) or in the opposite direction (angle of 180 degrees), the similarity matrix $U^\top U$ (respectively $V^\top V$) takes the form:

$$\begin{bmatrix} 1 & \langle u, \pm u \rangle \\ \langle \pm u, u \rangle & 1 \end{bmatrix}$$

   If the singular vectors are perfectly aligned (0 degrees), then $u_1 = u_2 = u$, simplifying the diagonal elements to $\langle u, u \rangle = \|u\|^2 = 1$. Conversely, if the singular vectors are oppositely aligned (180 degrees), $u_1 = -u_2$, resulting in $\langle u, -u \rangle = -1$ Thus, the similarity matrices becomes:

$$\begin{bmatrix} 1 & \pm 1 \\ \pm 1 & 1 \end{bmatrix}$$

   This structure reveals complete interference between the tasks: a double scaling effect when the vectors agree and complete cancellation when they disagree.

3. **Partially Collinear Singular Vectors:** when u_1 and u_2 (respectively v) are partially collinear, with the angle between them ranging from slightly greater than 0 degrees to less than 90 degrees or slightly more than 90 degrees to less than 180 degrees, similarity matrices expressed as:

$$\begin{bmatrix} 1 & \langle u_1, u_2 \rangle \\ \langle u_2, u_1 \rangle & 1 \end{bmatrix}$$

   In this case, the overlap between singular vectors induces a partial interaction between the tasks. The degree of interference, whether it is constructive or destructive, is proportional to the cosine of the angle between the singular vectors. This partial collinearity leads to subtle interplay, where the tasks influence each other to a degree dictated by their vector alignment.

This example underscores the critical role of singular vector alignment in model merging, highlighting how orthogonality ensures independent task performance, collinearity leads to maximal interference and partial collinearity results in an intermediate level of task interaction.

B. Additional details

B.1. Implementation details and computational resources

Normalized Accuracy To address the varying difficulties of the task, we report both normalized and absolute accuracies in our results. The normalized accuracy provides a relative performance metric by comparing the accuracy of the multi-task model to that of individually fine-tuned models. Specifically, the normalized accuracy is calculated as:

$$\text{Normalized Accuracy} = \frac{1}{T} \sum_{i=1}^{T} \frac{\text{accuracy}(\theta_{MT}, t_i)}{\text{accuracy}(\theta_{ft_i}, t_i)} \quad (9)$$

where T is the total number of tasks, θ_{MT} represents the multi-task model and θ_{ft_i} denotes the individually fine-tuned model for task t_i. This metric allows for a more fair comparison by adjusting for the baseline performance of each task.

Datasets for tasks All benchmarks were performed by integrating the codebase provided by Wang et al. [46]. In line with the principles of PEFT, we reused the already existing model checkpoints in the codebase for both the models and classification heads without additional fine-tuning.

Implementation Our method utilizes the SVD, a matrix decomposition technique applicable to two-dimensional matrices. For layers that are not represented as matrices (e.g., normalization layers) we default to standard `Task Arithmetic`. In particular, we employ Knut's algorithm [47] to compute the average efficiently. This ensures that all fine-tuned model task layers, regardless of their structure, are appropriately integrated into the merged model.

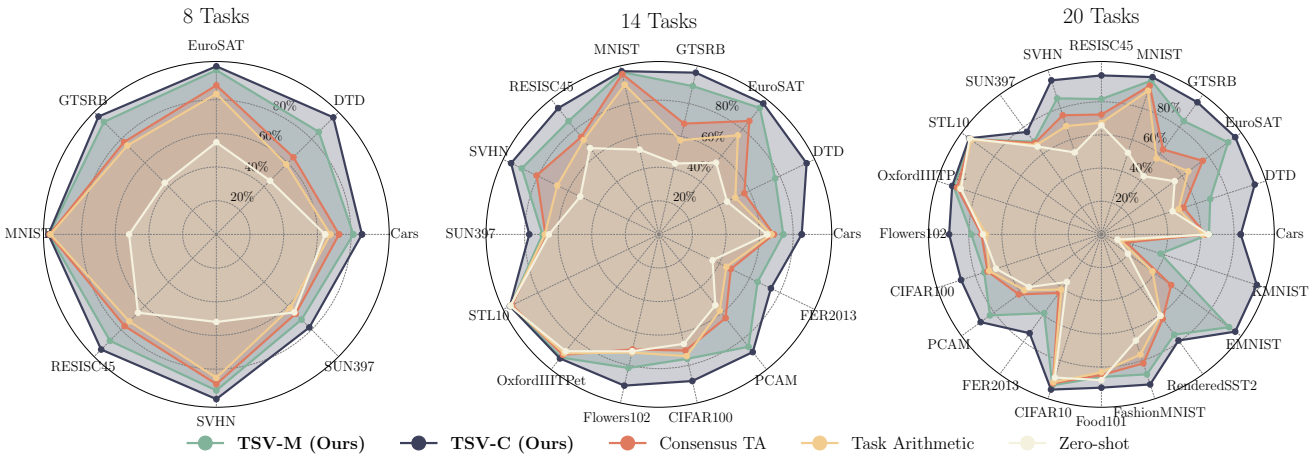

Figure 9. Absolute accuracy of a `ViT-B-16` merged over 8, 14, and 20 tasks, respectively.

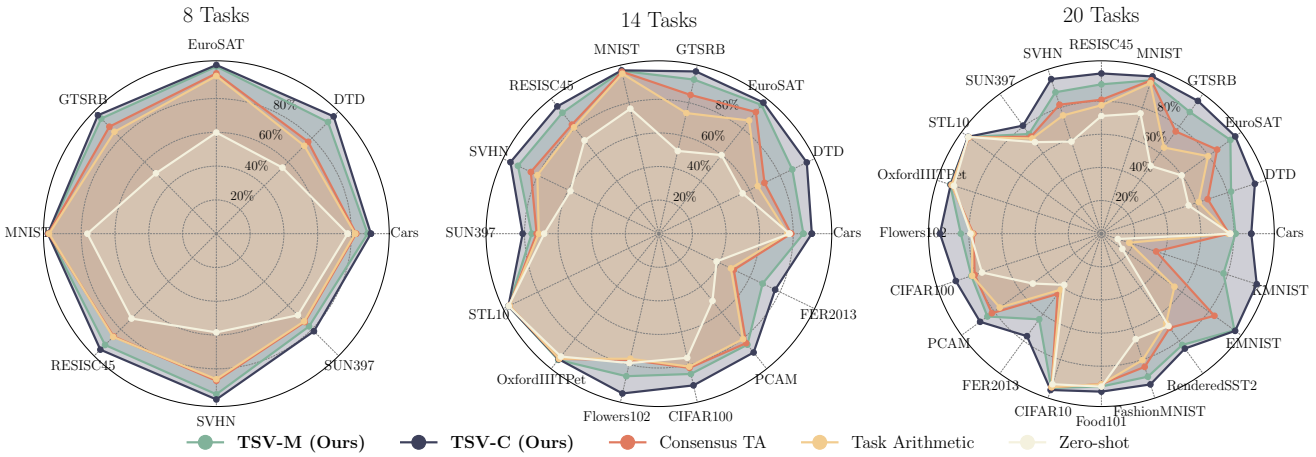

Figure 10. Absolute accuracy of a `ViT-L-14` merged over 8, 14, and 20 tasks, respectively.

Compute Resources We utilize PyTorch as deep learning framework. All the merging and evaluations were conducted on an `NVIDIA 4060Ti` GPU with 16GB of memory, and an `Intel i7-6800K` CPU equipped with 64GB of RAM. For experiments that need more than 64GB of RAM, we resort to a shared `HTCondor` cluster equipped with `NVIDIA P6000` GPUs.

B.2. Hyperparameter Settings

Following `Task Arithmetic` [22] and `Consensus TA` [46], we apply a single scaling factor, α, to adjust the multi-task vector within the model merging techniques outlined in Table 2. This scaling factor is optimized, when feasible, over the range $\{0.0, 0.1, ..., 2.9, 3.0\}$, with the optimal value selected based on the average validation performance across all tasks. However, as discussed in Section 6.3, our experimental findings indicate that the proposed `TSV-Merge` method does not strictly depend on this hyperparameter, as the marginal performance gains from tuning α do not justify the computational resources required for the evaluation on the validation datasets. Consequently, this allows us to eliminate the necessity for validation datasets and the corresponding labels from the prerequisites of the method, further simplifying the practicality and resource usage of the approach. The evaluation is performed on batch of 32 images. To produce Fig. 7 we selected the following subsets of 8 tasks from the 20 available, using the whitening method to speed up computation, the number in the image is the index indicating the subset:

0. `SVHN`, `SUN397`, `STL10`, `OxfordIIITPet`, `Flowers102`, `CIFAR100`, `PCAM`, `FER2013`
1. `PCAM`, `FER2013`, `CIFAR10`, `Food101`, `FashionMNIST`, `RenderedSST2`, `EMNIST`, `KMNIST`
2. `EuroSAT`, `GTSRB`, `MNIST`, `RESISC45`, `SVHN`, `SUN397`, `STL10`, `OxfordIIITPet`

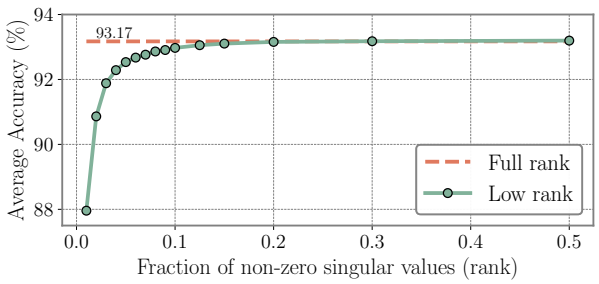

Figure 11. Mean absolute accuracy of the `ViT-B-16` model across increasing fractions of retained singular components, averaged over 20 tasks. The red line represents the average accuracy of the original fine-tuned models with full-rank task matrices, while the green line shows the accuracies using low-rank approximations.

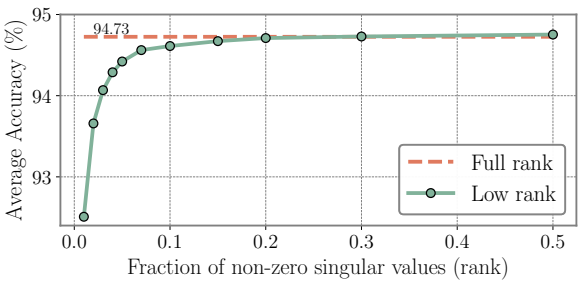

Figure 12. Mean absolute accuracy of the `ViT-L-14` model across increasing fractions of retained singular components, averaged over 20 tasks. The red line represents the average accuracy of the original fine-tuned models with full-rank task matrices, while the green line shows the accuracies using low-rank approximations.

3. `Cars`, `DTD`, `EuroSAT`, `GTSRB`, `MNIST`, `RESISC45`, `SVHN`, `SUN397`
4. `Cars`, `DTD`, `EuroSAT`, `GTSRB`, `FashionMNIST`, `RenderedSST2`, `EMNIST`, `KMNIST`
5. `MNIST`, `RESISC45`, `SVHN`, `SUN397`, `STL10`, `OxfordIIITPet`, `Flowers102`, `CIFAR100`
6. `STL10`, `OxfordIIITPet`, `Flowers102`, `CIFAR100`, `PCAM`, `FER2013`, `CIFAR10`, `Food101`

B.3. Storage cost calculation

Suppose we have a neural network comprising of L two-dimensional layers, each of dimension $d \times m$, and N one-dimensional layers of size c. The total number of parameters in the network is therefore:

$$\text{Params(NN)} = L \times (d \times m) + N \times c.$$

In standard `Task Arithmetic`, one must store the same number of parameters to obtain a task vector. In contrast, our approach provides the flexibility to select the number of parameters to preserve based on storage constraints or the desired needed performance, to adhere to the chosen constraints. Under the above assumptions, our method applies the truncated SVD to each two-dimensional layer. This decomposition yields two matrices of singular vectors, U and V, and a vector of singular values, σ, specifically:

- U of size $d \times k$,
- V of size $k \times m$,
- σ of size k,

where $k = \min(d, m)$. We select a reduced rank $k' \ll k$ to approximate each layer's task matrix. Consequently, the total number of parameters for TSV becomes:

$$\text{Params(TSV)} = L \times ((d \times k') + k' + (k' \times m)) + N \times c$$

To demonstrate that our method results in fewer stored parameters than the original parameter count, we require that $k' < \frac{d \times m}{d+m+1}$. This condition ensures:

$$\text{Params(NN)} > \text{Params(TSV)}$$

Substituting the expressions, yields:

$$L \times (d \times m) + N \times c > L \times ((d \times k') + k' + (k' \times m)) + N \times c$$

Simplifying, we obtain:

$$\begin{aligned} L \times (d \times m) &> L \times ((d \times k') + k' + (k' \times m)) \\ (d \times m) &> ((d \times k') + k' + (k' \times m)) \\ d \times m &> k' \times (d + 1 + m) \\ k' &< \frac{d \times m}{d + m + 1}. \end{aligned} \tag{10}$$

This inequality confirms that our method reduces the storage requirements of a task vector when $k' < \frac{d \times m}{d+m+1}$. Empirical evidence from Figures 2, 11 and 12 suggests that selecting $k' < 0.1 \times k = 0.1 \times \min(d, m)$ is sufficient to preserve most of the task performance, preserving the main requirement of performance. Furthermore, it is easy to prove that when choosing $k' = \frac{k}{T}$, the inequality is always satisfied for $T \geq 3$, respecting the main requirement of limited storage usage.

C. Proofs

We hereby prove the claims outlined in the main manuscript.

C.1. Characterization of the similarity matrices

Proposition C.1. *The matrix $\hat{U}^\top \hat{U}$ is positive definite.*

Proof. We define $\hat{U}^\top \hat{U}$, where $\hat{U}$ is a generic $d \times k$ rectangular matrix. Consequently, $\hat{U}^\top \hat{U}$ is a $k \times k$ square matrix. To establish that $\hat{U}^\top \hat{U}$ is positive definite, it suffices

to demonstrate that for all non-zero vectors $x \in \mathbb{R}^k$, the following inequality holds:

$$x^\top \hat{U}^\top \hat{U} x > 0.$$

This expression can be rewritten as:

$$x^\top (\hat{U}^\top \hat{U}) x = (\hat{U}x)^\top (\hat{U}x) = \|\hat{U}x\|^2.$$

Here, $\|\hat{U}x\|^2$ denotes the squared Euclidean norm of the vector $\hat{U}x$, which is always non-negative. Moreover, assuming that $\hat{U}$ has full column rank, the norm $\|\hat{U}x\|^2$ is strictly positive for any non-zero vector x. Therefore, we have:

$$\|\hat{U}x\|^2 > 0 \quad \text{for all} \quad x \in \mathbb{R}^k,\, x \neq 0.$$

This implies that:

$$x^\top \hat{U}^\top \hat{U} x > 0 \quad \text{for all} \quad x \in \mathbb{R}^k,\, x \neq 0,$$

which confirms that $\hat{U}^\top \hat{U}$ is positive definite. □

Corollary C.2. *Since $\hat{U}^\top \hat{U}$ is positive definite, then $\hat{U}^\top \hat{U}$ is invertible.*

Proof. From Proposition C.1, we have established that $\hat{U}^\top \hat{U}$ is a positive definite matrix. A positive definite matrix, by definition, has all its eigenvalues strictly positive. Let $\lambda_1, \lambda_2, \ldots, \lambda_k$ denote the eigenvalues of $\hat{U}^\top \hat{U}$. Therefore, we have:

$$\lambda_i > 0 \quad \text{for all} \quad i = 1, 2, \ldots, k.$$

The determinant of $\hat{U}^\top \hat{U}$ is the product of its eigenvalues:

$$\det(\hat{U}^\top \hat{U}) = \prod_{i=1}^{k} \lambda_i.$$

Since each λ_i is positive, their product is also positive:

$$\det(\hat{U}^\top \hat{U}) > 0.$$

A matrix is invertible if and only if its determinant is non-zero. Given that $\det(\hat{U}^\top \hat{U}) > 0$, it follows that $\hat{U}^\top \hat{U}$ is invertible.

Therefore, $\hat{U}^\top \hat{U}$ is invertible. □

Low-rank approx.	Interf. reduction	ViT-B-16 8 tasks	ViT-B-16 14 tasks	ViT-B-16 20 tasks
×	×	79.6 (+0.0)	75.9 (+0.0)	70.8 (+0.0)
✓	×	79.6 (+0.0)	74.9 (-1.0)	70.0 (-0.8)
×	✓	84.8 (+5.2)	79.0 (+4.1)	73.2 (+3.2)
✓	✓	93.9 (+9.1)	91.0 (+12.0)	86.5 (+13.3)

Table 5. 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`.

C.2. Observations

Since $\hat{U}^\top \hat{U}$ is a real symmetric matrix, it admits an eigendecomposition of the form

$$\hat{U}^\top \hat{U} = Q \Lambda Q^{-1} = Q \Lambda Q^\top,$$

where:

- Λ is a diagonal matrix containing the real eigenvalues of $\hat{U}^\top \hat{U}$,
- Q is an orthogonal matrix whose columns are the orthonormal eigenvectors of $\hat{U}^\top \hat{U}$, satisfying $Q^\top = Q^{-1}$.

The inverse of $\hat{U}^\top \hat{U}$ exists (see Corollary C.2), and can be expressed using its eigendecomposition as

$$(\hat{U}^\top \hat{U})^{-1} = Q \Lambda^{-1} Q^{-1} = Q \Lambda^{-1} Q^\top.$$

Additionally, since Λ is a diagonal matrix with non-zero diagonal entries (Proposition C.1), its inverse Λ^{-1} is straightforward to compute, with each diagonal element given by

$$\Lambda^{-1} = \operatorname{diag}\left(\frac{1}{\lambda_i}\right),$$

where λ_i are the eigenvalues of $\hat{U}^\top \hat{U}$.

Furthermore, the eigenvalues of $(\hat{U}^\top \hat{U})^{-1}$ are $\frac{1}{\lambda_i}$, each of which is positive since $\lambda_i > 0$ for all i (following by the definition in Proposition C.1). Consequently, $(\hat{U}^\top \hat{U})^{-1}$ is also a positive definite matrix.

These observations confirm that not only is $\hat{U}^\top \hat{U}$ positive definite, but its inverse inherits this property due to the positivity of its eigenvalues.

Low-rank approx.	Interf. reduction	ViT-L-14 8 tasks	ViT-L-14 14 tasks	ViT-L-14 20 tasks
×	×	88.6 (+0.0)	84.0 (+0.0)	78.1 (+0.0)
✓	×	87.9 (-0.7)	83.4 (-0.6)	77.2 (-0.9)
×	✓	92.1 (+4.2)	86.8 (+3.4)	81.0 (+3.8)
✓	✓	97.0 (+4.9)	94.4 (+7.6)	92.5 (+11.5)

Table 6. 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`.

C.3. Proof of Theorem 6.1

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*

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

Proof. Let us consider the SVD decomposition of U and $\widehat{U}$: $U = P_u \Sigma_u P_v$ and $\widehat{U} = R_u \widehat{\Sigma_u} R_v$. X and $\widehat{X}$ obtain as $X = P_u P_v^\top$, $\widehat{X} = R_u R_v^\top$ respectively We first consider the Frobenius norm of $||X - U||_F$. Notice that the singular values of U are the square root of the eigenvalues of $\Sigma_u = UU^\top$.

$UU^\top = \sum_{i=1}^{N} U_i U_i^\top = TI_n$. As a consequence, the eigenvalues of $UU^\top$ are all equal to T and the singular values are all equal to $\sqrt{T}$.

$$\begin{aligned} ||X - U||_F &= ||P_u P_v^\top - P_u \Sigma_u P_v^\top||_F \\ &= ||P_u (I - \Sigma_u) P_v^\top||_F \\ &= ||I_n - \Sigma_u||_F \\ &= ||I_n - \sqrt{T} I_n||_F \\ &= \sqrt{n}(\sqrt{T} - 1). \end{aligned}$$

We are now left to compute $||\widehat{X} - \widehat{U}||_F$. In this case, we are not able to compute the exact norm without other assumptions, but we can provide an upper bound that gives us a sufficient condition to prove our statement. As before

$$\begin{aligned} ||\widehat{X} - \widehat{U}||_F &= ||I_n - \widehat{\Sigma}_u||_F \\ &= \sqrt{\sum_{i=1}^{n} (\hat{\sigma}_i - 1)^2}. \end{aligned}$$

where $\hat{\sigma}$ are the singular values of $\widehat{U}$.

Notice that $\sum_{i=1}^{n} \hat{\sigma}_i^2 = \text{tr}(\widehat{U}\widehat{U}^\top) = \text{tr}(\widehat{U}^\top \widehat{U})$ and $\widehat{U}^\top U$ is a $Tk \times Tk$ matrices with diagonal elements equal to one, so $\text{tr}(\widehat{U}^\top \widehat{U}) = kT$.

Moreover,

$$\sum_{i=1}^{n} \hat{\sigma}_i = \sum_{i=1}^{n} \sqrt{\lambda_i(\widehat{U}\widehat{U}^\top)} \tag{11}$$

$$\geq \sqrt{\sum_{i=1}^{n} \lambda_i(\widehat{U}\widehat{U}^\top)} \tag{12}$$

$$= \sqrt{\text{tr}(\widehat{U}\widehat{U}^\top)} = \sqrt{Tk} \tag{13}$$

Notice that this upper bound is tight, indeed the sum of the singular values of $\hat{U}$ must lie within: $\sqrt{kT} \leq \sum_{i=1}^{n} \hat{\sigma}_i \leq kT$. The minimum $\sqrt{kT}$ is achieved if all matrices U_i are equals, on the other end, the maximum kT is achieved if the kT columns are orthonormal.

Putting everything together,

$$||\widehat{X} - \widehat{U}||_F = \sqrt{\sum_{i=1}^{n} (\hat{\sigma}_i - 1)^2} \tag{14}$$

$$= \sqrt{n + \sum_{i=1}^{n} \hat{\sigma}_i^2 - 2\sum_{i=1}^{n} \hat{\sigma}_i} \tag{15}$$

$$= \sqrt{n + kT - 2\sum_{i=1}^{n} \hat{\sigma}_i} \tag{16}$$

$$\leq \sqrt{n + kT - 2\sqrt{kT}}. \tag{17}$$

So we have to check for what values of k it holds that $\sqrt{n + kT - 2\sqrt{kT}} \leq \sqrt{n}(\sqrt{T} - 1)$.

We have that

$$\sqrt{n + kT - 2\sqrt{kT}} \leq \sqrt{n + kT} \leq \sqrt{n}(\sqrt{T} - 1) \tag{18}$$

Equation 18 is satisifed if

$k \leq n\frac{T - 2\sqrt{T}}{T}$. This concludes the proof. Since k is a positive number the inequaility of meaningful for $T > 4$. □

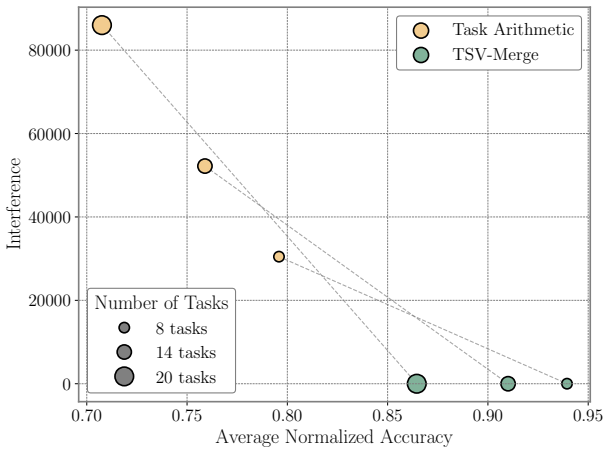

Figure 13. Singular Task Interference (STI) and average normalized accuracy for `Task Arithmetic` and `TSV-Merge` on the `ViT-B-16` model, evaluated across merges of 8, 14, and 20 tasks.

D. Additional experimental results

D.1. Per-dataset performance metrics

In Section 5.1, we present comprehensive results for individual tasks using the `ViT-B-32` model. Here we include analogous radar plots for the `ViT-B-16` model in Figure 9 and the `ViT-L-14` model in Figure 10. The analyses of these models reveal findings consistent with those reported for `ViT-B-32` in the main text.

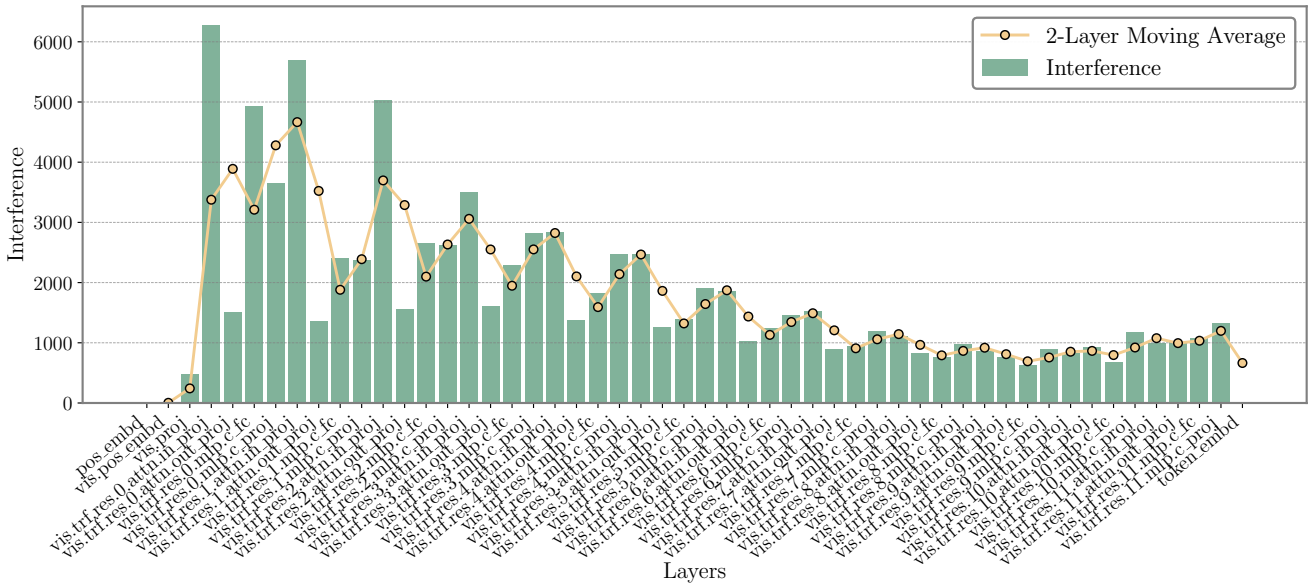

Figure 14. Detailed view of Singular Task Interference (STI) across layers in a `ViT-B-32` for 20 tasks. The interference trend is high in early layers and decreases later. Here, the pattern for each transformer block is observable, the interference first increases and then drops in each attention-out layer.

D.2. Extended analysis

D.2.1. Whitening vs. SVD

As we have seen in Section 4.2, applying a whitening transformation to the matrices of task singular vectors is mathematically equivalent to solving the Orthogonal Procrustes problem. However, implementing these two approaches may yield different results depending on the distinct matrix decomposition algorithms employed. In this study, we used PyTorch to compute both eigendecomposition and SVD, observing slightly different results that may be attributed to numerical errors. To more robustly compute the matrix square root for the eigendecomposition case, we compute

$$\Lambda^{-\frac{1}{2}} = \text{diag}\left(\frac{1}{\sqrt{|\lambda_i|} + \epsilon}\right)$$

where $\epsilon = 1\text{e}{-12}$ prevents division by 0 and the absolute value avoids numerical errors producing small negative values in magnitude less than $1\text{e}{-6}$.

D.2.2. Impact of rank

The Section 3.2 shows that the task matrices of a `ViT-B-32` are inherently low-rank and a small percentage of TSVs is enough to approximate each layer with satisfying results. We here provide the same plots for the models `ViT-B-16` (Figure 11) and `ViT-L-14` (Figure 12), observing analogous findings. In fact, the first shows a decrease of 1.3% mean accuracy at 3% of retained TSVs and the second shows a reduction of 1.1% mean accuracy at 2% of maintained TSVs. We refer to Figure 17 for a breakdown of this analysis.

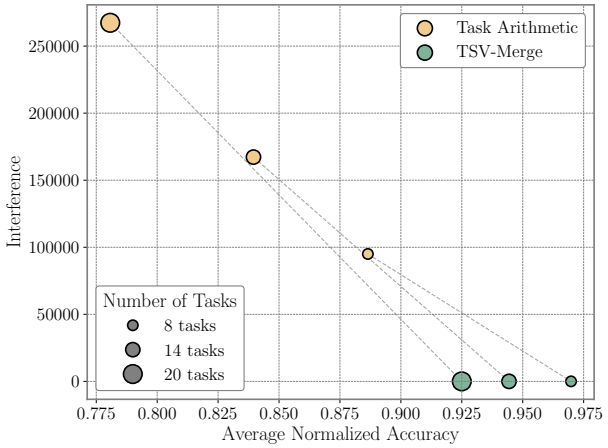

Figure 15. Singular Task Interference (STI) and average normalized accuracy for `Task Arithmetic` and `TSV-Merge` on the `ViT-L-14` model, evaluated across merges of 8, 14, and 20 tasks.

D.2.3. Extended Ablation study

In Section 6.1, we reported an ablation study on the `ViT-B-32` model to evaluate the individual contributions of low-rank approximation and interference reduction to the overall performance of our `TSV-Merge` method. To further mark our findings and demonstrate the generality of

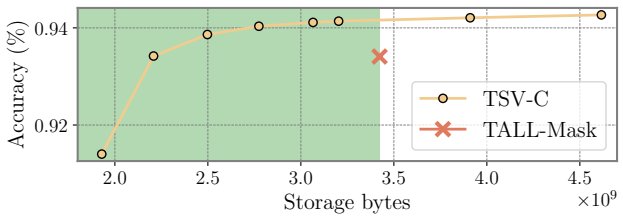

Figure 16. Accuracy with varying compression budgets for ViT-L-14 across 14 tasks.

our approach across different model sizes, we report in Table 5 the ablation study for the ViT-B-16 model and in Table 6 ViT-L-14 model. The experimental setup follows the one described in Section 6.1. We assess the impact of the two key components of TSV-M, low-rank approximation and interference reduction, by considering the following four configurations:

1. **Baseline Task Arithmetic**: the standard TA method without any modification.
2. **Low-Rank Approximation**: apply only low-rank approximation to task matrices without any interference reduction step.
3. **Interference Reduction**: apply interference reduction to the full-rank task matrices without any pre-step of low-rank approximation.
4. **TSV-Merge**: Combining both low-rank approximation and interference reduction.

D.2.4. Effect of task interference

We provide here the same plots shown in Figure 8, for the ViT-B-16 we show it in Figure 13 and respectively for ViT-L-14 in Figure 15. The finding remains valid also for these models, all the instances show a significant gain in accuracy when the interference is removed.

D.2.5. Detailed per-layer task interference

We show in Fig. 14 the per-layer task interference, extending the block-level analysis in Figure 6 in the main manuscript.

D.2.6. Compression analysis

Our experimental results (see Table 3 in main manuscript) demonstrate that TSV outperforms TALL-Mask on the large-scale ViT-L-14 model for 14 and 20 tasks benchmarks, signaling a scaling advantage. With a fixed-budget analysis, we show in Figure 16 that unlike TALL-Mask, which has a fixed requirement defined by model size and number of tasks, we allow flexible compression rates by allowing rank selection. This enables more aggressive compression, as highlighted in the green region in the figure.

D.2.7. Test-time adaptation

We compare our method with AdaMerging [52] for test-time adaptation. On a subset of 7 tasks from the 8 task benchmark, AdaMerging achieves an accuracy of 85.43%, while our TSV-M attains 88.93%, an improvement of approximately 3.5% without requiring any test-time adaptation. Additionally, when integrating an AdaMerging-style test-time adaptation into our framework, the accuracy increases to 89.87%, demonstrating the complementary benefits of combining TSV-M with test-time adaptation techniques.

E. Theoretical motivations and analysis

E.1. Theoretical foundation - Empirical design

The TSV-C method is grounded in the well-established framework of low-rank approximation for compression (e.g., [12]). Instead, TSV-M is motivated by more empirical foundations: it is designed to achieve noise reduction through low-rank approximation and to eliminate interference via orthogonalization. Low-rank truncation serves to filter out insignificant variations, while orthogonalization ensures that task-specific singular vectors remain independent, preserving individual task performance.

E.2. Heuristic interference measure

Given that a formal definition of interference in model merging is not yet established, we adopt an operational definition: interference is any cross-task interaction that hinders the merging process. Our proposed Singular Task Interference measure is empirically validated by the consistent performance improvements observed when its value is minimized. Furthermore, we examine the relationship between overlapping singular vectors and knowledge sharing. Unlike multi-task learning (MTL), which enables coordinated knowledge sharing through joint training, the independent task-wise finetuning in model merging may evolve in destructive overlaps in the activations, resulting in interference rather than beneficial knowledge sharing. By orthogonalizing the singular vectors, our approach effectively mitigates these overlaps, reducing interference and enhancing the performance of the merged model.

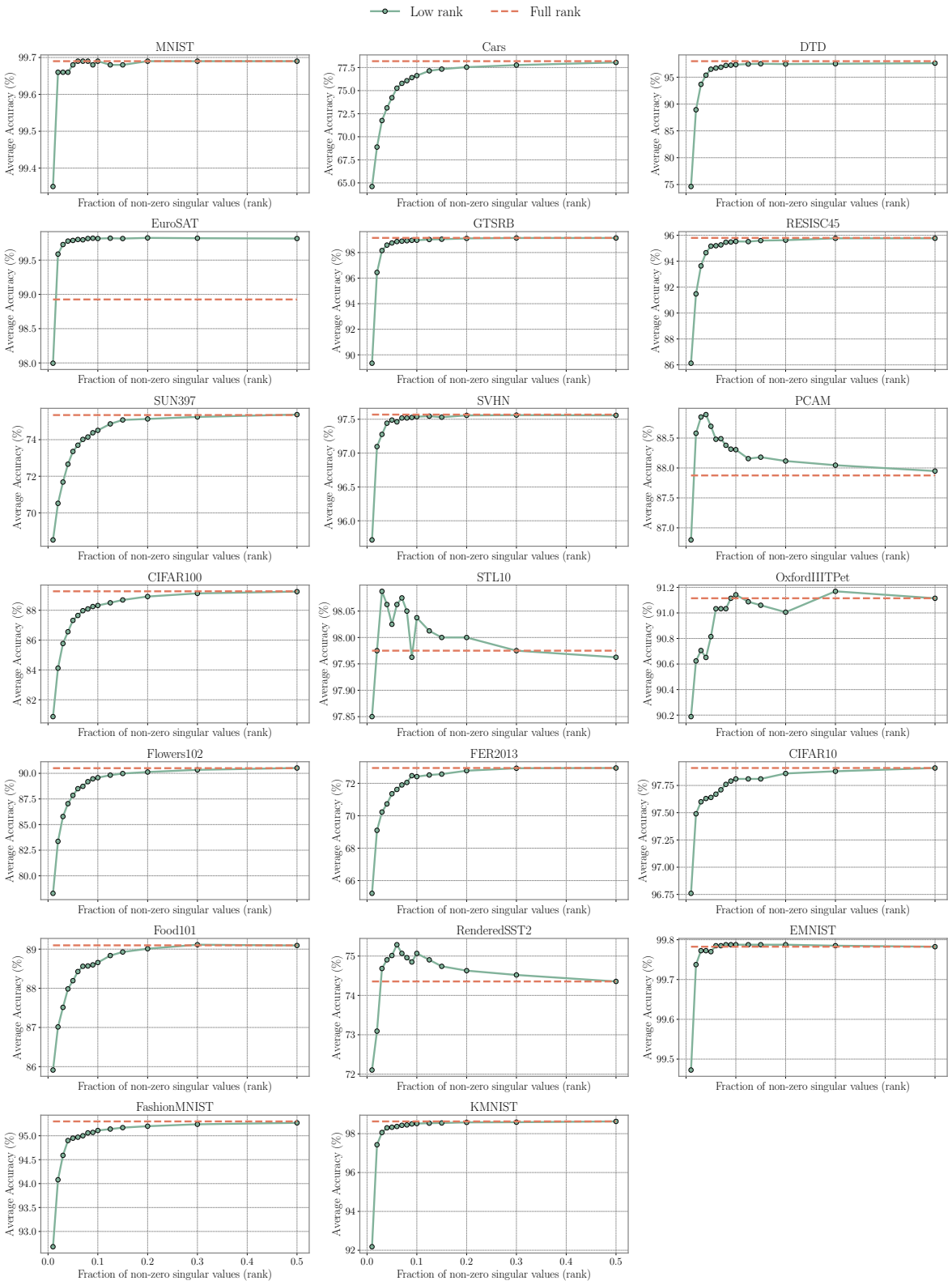

Figure 17. Absolute accuracy of the `ViT-B-32` model across increasing fractions of retained singular components, for each task. The red line represents the accuracy of the original fine-tuned models with full-rank task matrices, while the green line shows the accuracies using low-rank approximations.

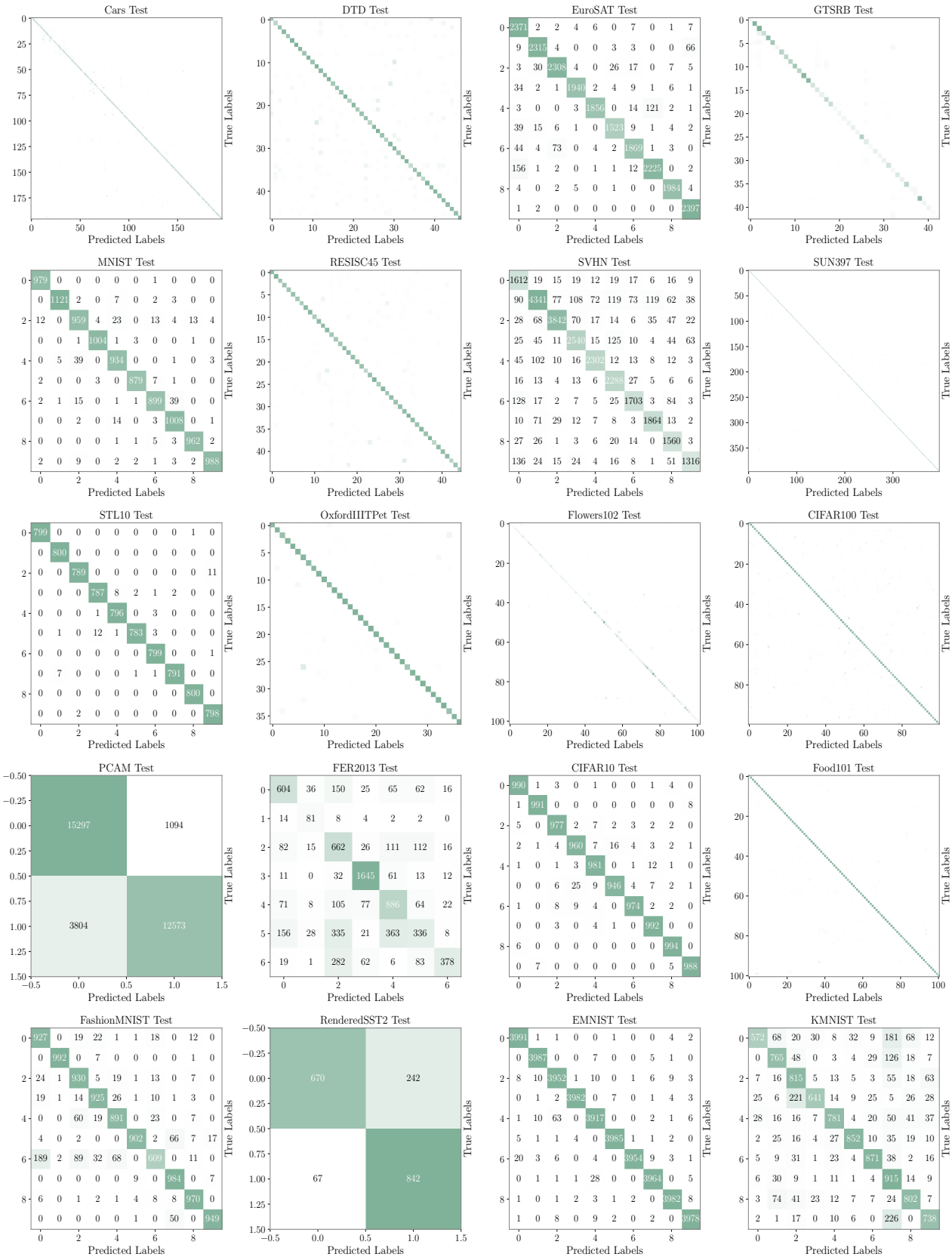

Figure 18. Breakdown of classification accuracy in confusion matrices of a single merged `ViT-L-14` model over 20 tasks. The numbers are omitted when they are too small to display.