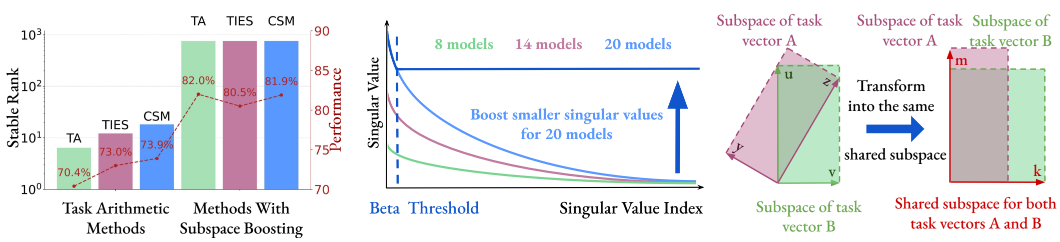
# SUBSPACE-BOOSTED MODEL MERGING

**Anonymous authors**
Paper under double-blind review

## ABSTRACT

Model merging enables the combination of multiple specialized expert models into a single model capable of performing multiple tasks. However, the benefits of merging an increasing amount of specialized experts generally lead to diminishing returns and reduced overall performance gains. In this work, we offer an explanation and analysis from a task arithmetic perspective; revealing that as the merging process (across numerous existing merging methods) continues for more and more experts, the associated task vector space experiences *rank collapse*. To mitigate this issue, we introduce *Subspace Boosting*, which operates on the singular value decomposed task vector space and maintains task vector ranks. *Subspace Boosting* raises merging efficacy for up to 20 expert models by large margins of more than 10% when evaluated on both vision and language benchmarks. Moreover, we propose employing Higher-Order Generalized Singular Value Decomposition to quantify task similarity, offering *a new interpretable perspective on model merging*.




(a) The stable rank and performance for different methods before and after *Subspace Boosting*.

(b) *Subspace Boosting* boosts smaller singular values to increase the effective rank of the model.

(c) Higher-Order Generalized SVD transforms independent task vector subspaces into a shared subspace.

Figure 1: Overview of our contributions. (a) Popular merging methods such as Task Arithmetic (TA) (Ilharco et al., 2023b), TIES (Yadav et al., 2023) and Consensus Merging (CSM) (Wang et al., 2024c), suffer from ***rank collapse***, correlating with low performance. (b) To prevent ***rank collapse***, we introduce *Subspace Boosting*, which mitigates it by boosting neglected singular values, vastly improving performance. (c) Finally, for interpretability, we use HO-GSVD, transforming individual models to share the same subspace, enabling direct comparison.

## 1 INTRODUCTION

Training models at "foundational" scale (Bommasani et al., 2021) has significantly driven progress in the development of general-purpose models across domains, ranging from computer vision to natural language processing (Radford et al., 2021b; 2019; Brown et al., 2020; Devlin et al., 2018; Rombach et al., 2022). Despite their broad use in various downstream tasks, these foundation models still require fine-tuning to effectively adapt to specialized *expert* domains (Roth et al., 2024; Mukhoti et al., 2024), such as particular reasoning, language, or image generation. In addition, deploying and storing a growing number of *expert models* becomes unsustainable (Yadav et al., 2024b;a).

To address these problems, recent advances in model merging (Wortsman et al., 2022b; Yadav et al., 2024b; Dziadzio et al., 2025; Wang et al., 2024a) have shown significant promise. Model merging enables the creation of a ***single model*** from multiple experts (Rofin et al., 2022; Yadav et al., 2024b; 2023), while preserving the foundational generalization capability. This enables improved generalization capabilities across numerous domains, substantially simplified model inference, and the development of decentralized models.

Current approaches in model merging commonly revolve around simple scaled or filtered weight interpolation of experts. The resulting merged models outperform the base model, but, as expected, perform worse than individual task experts on their specific task. However, recent studies (Yadav et al., 2024b; Dziadzio et al., 2025) suggest that the significance of specific merging techniques may be overestimated, as most merging methods often yield comparable performance, especially at scale, indicating a gap in understanding of the underlying merging process and the weight space structure.

To bridge this gap, we first investigate the merged weight space to understand why popular merging techniques yield suboptimal performance. In particular, we investigate the task vectors, since they contain the essential task information. Our results, visualized in Fig. 1a, reveal that the merged task vectors suffer from ***rank collapse***, in which a vast majority of information is captured by the most important singular values and vectors, shown in Fig. 1b. This can be estimated by their *stable rank* (Zhou et al., 2010; Shukla et al., 2024; Sanyal et al., 2020) (which evaluates the "effective" rank of the matrix by disregarding small singular values). Our analysis also reveals that *rank collapse* consistently affects existing methods, seen in Fig. 1a, and that the entire model's task vector space suffers from *rank collapse*, leading to the merged task vectors operating on a constrained subspace.

To address this, we propose *Subspace Boosting*, a method that directly mitigates *rank collapse* by decomposing task vectors via Singular Value Decomposition (SVD), and explicitly boosting underutilized dimensions (Fig. 1b). Therefore, by using these subspaces, our method significantly enhances the model's capability and performance, as shown in Fig. 1a. *Subspace Boosting* shows strong improvements across both vision and language tasks for up to 20 vision transformer experts (Dosovitskiy et al., 2021) and 8 T5 language experts (Raffel et al., 2020) of varying sizes when applied to recent merging techniques (Task Arithmetic (Ilharco et al., 2023b), TIES (Yadav et al., 2023) and Consensus Merging (Wang et al., 2024c)). For both domains, the baseline method's performance is increased by over 10%.

Finally, we use Higher-Order Generalized Singular Value Decomposition (HO-GSVD) to introduce an interpretable model merging variant (Fig. 1c). This approach decomposes task vectors into a shared space containing ***common*** and ***unique*** subspaces. This allows the comparison of task similarity or selection of optimal models via the *Alignment Matrix*, derived from the shared vector space.

Our contributions can be summarized as follows.

- We identify weight-space *rank collapse* as a crucial limitation in task-arithmetic based methods, which reduces generalization of the merged model.
- We introduce *Subspace Boosting*, a general method that mitigates *rank collapse*, which is compatible with several merging techniques, significantly improving merging efficacy across standard model merging vision and language benchmarks.
- Finally, we propose a novel framework using Higher-Order Generalized SVD on task vectors, allowing the *shared* subspace to be identified, interpreted, or used for expert selection.

## 2 Related Work

Model merging (Yadav et al., 2024a; Yang et al., 2024a) has emerged as an important technique to improve post-training capabilities, and as a general toolkit to combine knowledge across different expert models (Wortsman et al., 2022a; Rame et al., 2023; Sanyal et al., 2024; Sung et al., 2023; Pari et al., 2024; Nylund et al., 2023; Zaman et al., 2023; Stoica et al., 2024; Wang et al., 2024c; He et al., 2024; Oh et al., 2024; Shen et al., 2024; Sharma et al., 2024; Tam et al., 2024b; Goddard et al., 2024; Xiong et al., 2024; Yang et al., 2024b; Lu et al., 2024; Zheng & Wang, 2024; Nasery et al., 2024; Rofin et al., 2022; Yadav et al., 2023; Jin et al., 2023; Deep et al., 2024; Marczak et al., 2024), even across time (Roth et al., 2024; Dziadzio et al., 2025). Many model merging techniques leverage the principle of linear mode connectivity, which implies that model weights across separate training runs can be (linearly) interpolated, especially when finetuned from the same base model (Izmailov et al., 2018; Ramé et al., 2024; Neyshabur et al., 2020; Frankle et al., 2020; Ainsworth et al., 2023; Garipov et al., 2018; Entezari et al., 2022). Initial studies mainly focus on simple linear weight interpolation (Wortsman et al., 2022b; Rofin et al., 2022) or spherical linear interpolation (SLERP, (Shoemake, 1985; Ramé et al., 2024)) without particular differentiation between individual weights.

**Task Arithmetic.** Ilharco et al. (2023b) provided a task arithmetic perspective on the interpolation problem, defining finetune-to-base-weight differentials as *task vectors*. Building upon the work of Ilharco et al. (2023b), methods such as Tangent Task Arithmetic ((Ortiz-Jimenez et al., 2023), finetuning on the weight tangent space), TIES ((Yadav et al., 2023), removing low magnitude task vector entries and magnitude-based sign assignment), DARE ((Davari & Belilovsky, 2025), random weight masking over task vectors), Model Stock ((Jang et al., 2024), determining a suitable center of mass across multiple task vectors) or Breadcrumbs ((Davari & Belilovsky, 2025), cutting tail-ended task weights based on the distribution of magnitudes) have been introduced.

**Adaptive Methods.** An orthogonal line of research explores adaptive methods to find the optimal merging parameters (Lee et al., 2025; Yang et al., 2024c). By contrast, we propose a training-free method designed to improve the capabilities of existing model merging techniques. Therefore, we consider adaptive methods (Lee et al., 2025; Yang et al., 2024c) as distantly-related.

**SVD-Based.** Similar to our research, several existing works employ Singular Value Decomposition (SVD) in the context of model merging (Stoica et al., 2024; Choi et al., 2025; Marczak et al., 2025; Gargiulo et al., 2025). For example, Gargiulo et al. (2025) propose TSV-Merge, which reduces task interference by compressing layer-wise task matrices to their essential singular vectors and then decorrelating them. Similarly, Marczak et al. (2025) introduced Iso-C and Iso-CTS, two model merging methods that enhanced the performance of model merging by introducing task-specific subspaces. While these methods leverage SVD, our work is the first to diagnose and quantify the phenomenon of *rank collapse*. By mitigating *rank collapse*, our method substantially improves Task-Arithmetic based methods while being over 6x more time-efficient than the above SVD-based methods. Finally, we are the first to leverage HO-GSVD for merging, introducing a novel framework for interpreting task similarity and enabling principled expert selection.

## 3 Rank Collapse in Model Merging

In this section, we explore the prevalent phenomenon of ***rank collapse*** in the weight space of merged models. We begin by introducing the following notation and background.

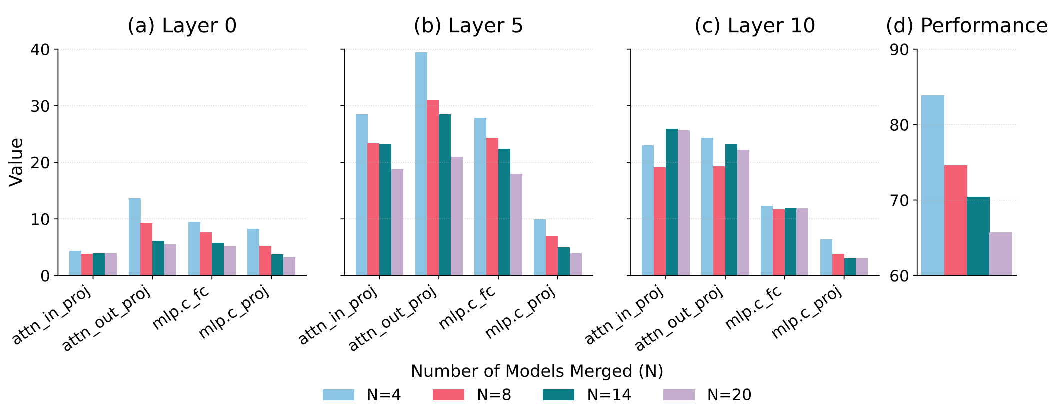



Figure 2: Stable rank in merged ViT-B/16 models. (a-c) The stable rank is decomposed across various attention and MLP sublayers of three layer blocks. (d) As more models are merged, the stable rank decreases across a majority of layers, strongly correlating with the performance.

**Background.** Given a set of $n$ expert models $\{M_1, M_2, \ldots, M_n\}$ finetuned for different tasks from the same pretrained model $M_{base}$, the goal of model merging is to obtain the model $M_m$ that is capable of solving all associated expert tasks. Let $\theta_{base}$ be the parameters of $M_{base}$, and $\theta_i$ be the parameters of each expert $M_i$, $1 \leq i \leq n$. Following Ilharco et al. (2023a), we define the ***task vectors*** as the weight differential $\Delta_i = \theta_i - \theta_{base}$, $1 \leq i \leq n$. Using this notation, the merged parameters $\theta_m$, defined as task arithmetic through linear interpolation, are expressed as: $\theta_m = \theta_{base} + \alpha \sum_i^n \Delta_i$, with $\alpha$ representing the scalar merging coefficient. We also define $\Delta_m = \Delta_1 + \cdots + \Delta_n$ as the total task vector.

**Rank Collapse During Merging.** To determine why model merging performance degrades as more models are merged, we investigate the subspaces spanned by the task vectors, as they isolate the knowledge specific to each task. In particular, we investigate whether merged task vectors suffer from rank collapse. We measure the rank collapse with the *stable rank* and *cumulative energy rank* (in Supplementary Sec. D). The *Stable rank* uses SVD, and for a matrix $A \in \mathbb{R}^{m \times n}$, is defined as:

$$A = U \Sigma V^T, \tag{1}$$

with the orthogonal matrix $U \in \mathbb{R}^{m \times m}$, also denoted as *left* singular vectors, $\Sigma \in \mathbb{R}^{m \times n}$ the diagonal matrix containing singular values $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_n \geq 0$, and the *right* singular vectors $V \in \mathbb{R}^{n \times n}$.

Using this decomposition, the *stable rank* $\mathcal{M}_{\text{stable}}$ denotes the "effective rank" of the weight matrices, similar to computing the rank of the matrix by ignoring the smaller singular values:

$$\mathcal{M}_{\text{stable}} = \frac{\sum_i \sigma_i^2}{\max_i \sigma_i^2}. \tag{2}$$

A small stable rank indicates that most of the information is concentrated in only a few dimensions. This implies that the weight projections operate within a limited subspace, underutilizing the full vector space.

In Fig. 2, we investigate the stable rank of merged ViT-B/16 models with Task Arithmetic. As more models are merged, we clearly observe that the stable rank decreases for a large majority of sublayers. While the problem grows in complexity as the model contains more tasks, the subspace actually contracts, contradicting the expectation that a larger set of tasks would require a higher-rank representation. This establishes a strong correlation between rank collapse and model performance degradation. Similarly, in Fig. 1a, once the stable rank increases, the performance also drastically improves across the baselines: Task Arithmetic (Ilharco et al., 2023b) (in green), TIES (Yadav et al., 2023) (in pink) or Consensus Merging (CSM) (Wang et al., 2024b) (in blue).

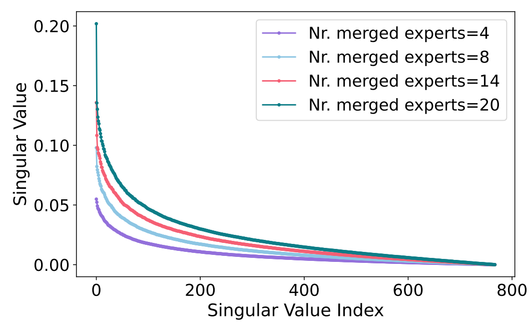



Figure 3: **Evolution of the Singular Value Distribution.** As more experts are merged, higher absolute and relative mass is placed on fewer singular vectors; encouraging the rank collapse. This indicates that information becomes concentrated in fewer dominant dimensions.

Fig. 3 further illustrates rank collapse by visualizing the singular value distributions for the models. The ordering is clear: as more models are merged, the largest singular values increase more drastically compared to the smallest singular values. For example, the largest singular value for 20 merged models ($\sigma \approx 0.20$) is $4\times$ larger than for 4 merged models ($\sigma \approx 0.05$). For more rank collapse results across a number of merged models and merging methods, please see Supplementary Sec. D.

## 4 Model Merging from a Decomposed Subspace Perspective

Building on our prior analysis of rank collapse as well as previous works linking rank collapse (Dziadzio et al., 2025; Milbich et al., 2020) to reduced generalization, we propose two solutions. Firstly, we introduce *Subspace Boosting* (Sec. 4.1) to directly mitigate rank collapse in task vectors. Furthermore, we leverage HO-GSVD (Sec. 4.2) to create an interpretable framework for model merging and expert selection.

### 4.1 *Subspace Boosting* for Model Merging

To directly mitigate the rank collapse issue identified in Sec. 3, we introduce *Subspace Boosting*, a general method that operates on merged task vectors, ensuring compatibility with modern merging techniques that rely on task vectors (Yadav et al., 2023; Ilharco et al., 2023a; Wang et al., 2024b) (see pseudocode provided in Alg. 1).

We posit that preventing rank collapse is essential for effective merging. As more experts are merged, this collapse forces the models to encode an increasing amount of information into progressively constrained task vector subspaces, harming generalization. By addressing this key limitation, *Subspace Boosting* improves the merging process efficacy and final model performance.

*Subspace Boosting* is applied to the merged task vector from any Task-Arithmetic based method such as TA, TIES, or Consensus merging. Each weight matrix in the merged task vector (denoted by "param" in Alg. 1) is independently decomposed through the following steps. (i) Initially, the weight matrix is decomposed via SVD. (ii) Afterwards, the hyperparameter $\beta$ (denoted by "beta" in Alg. 1) is used to determine a cutoff point for the cumulative sum of singular values. All singular values smaller than the one at the cutoff index are "boosted" by clamping them to the cutoff value, visualized in Fig. 1b. (iii) Finally, the new weight matrix is reconstructed using the original singular vectors, but with the new, boosted singular values.

As observed, *Subspace Boosting* requires only one hyperparameter, namely the boosting threshold $\beta$, which determines the cutoff after which the smaller singular values are boosted. Across our experiments, our findings indicate that $\beta$ is highly robust, with minimal tuning required. Also, *Subspace Boosting* provides superior performance at 6x faster wall clock time over state-of-the-art methods (Gargiulo et al., 2025) (results in Supplementary Sec. C).

In addition, Fig. 1a demonstrates that *Subspace Boosting* effectively mitigates the rank collapse issue in task vectors by utilizing the full subspace (of dimensionality 768 for ViT-B/16 models), compared to other existing Task Arithmetic-based methods.

**Algorithm 1:** Pytorch style pseudocode for *Subspace Boosting*

```
def subspace_boosting(param, beta):
    """
    param: weight matrix
    beta: boosting threshold
    """
    U, S, Vh = svd(param)
    t_sum = S.sum()
    c_sum = torch.cumsum(S, dim=0)
    n_sum = c_sum / t_sum
    k = (n_sum >= beta).nonzero()
    idx = k[0].item()
    S_new = torch.clamp(S, min=S[idx])
    new_param = U @ diag(S_new) @ Vh

    return new_param
```

### 4.2 Breaking Down Task Vectors with Higher Order Generalized SVD

While *Subspace Boosting* substantially enhances performance, the underlying relationships between task vectors remain a black box. To create a more interpretable merging framework, we now focus on decomposing task vectors into their **common** and **unique** subspaces.

However, standard SVD is ill-suited for this purpose, as it decomposes each task vector into a unique, expert-specific basis, preventing direct comparison. We therefore leverage **Higher Order Generalized SVD (HO-GSVD)**, a technique that projects multiple matrices into a single, shared subspace. This enables a straightforward comparison of multiple expert weights. As proposed in Ponnapalli et al. (2011); Kempf et al. (2023); Loan (1976); Golub & Van Loan (2013), given a set of $N$ matrices $A_1, \ldots, A_N$, HO-GSVD decomposes each matrix $A_i$ as:

$$A_i = U_i \Sigma_i V^T, \quad i = 1, \ldots, N, \tag{3}$$

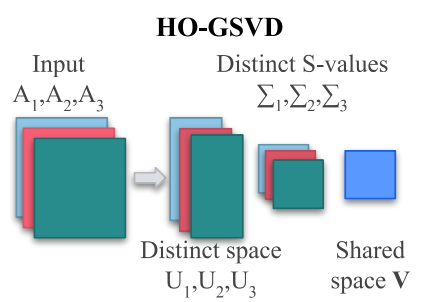



Figure 4: **Higher-Order Generalized SVD (HO-GSVD).** Unlike normal Singular Value Decomposition (SVD) which decomposes matrices into individual $A_i = U_i \Sigma_i V_i$, HO-GSVD allows for decompositions into shared right singular subspaces $V$.

resulting in *distinct* $U_i \in \mathbb{R}^{m_i \times n}$, $\Sigma_i \in \mathbb{R}^{n \times n}$ and a *shared* subspace $V \in \mathbb{R}^{n \times n}$, identical for all factorizations. In our case, the matrices $A_i$ correspond to the weight matrices originating from the independent task vectors $1 \leq i \leq n$, for a certain layer. By establishing a shared subspace $V$, this decomposition enables the direct comparison of different models and identification of **common** or **unique** subspaces for different tasks. The complete details are provided in Supplementary Sec. F.

Table 1: ***Subspace Boosting* significantly improves model merging efficacy.** Accuracy performance results (in %) for merging vision classification benchmarks with 8, 14 and 20 tasks (Wang et al., 2024b) when *Subspace Boosting* is applied to Task Arithmetic (TA) (Ilharco et al., 2023a;a), TIES-Merging (Yadav et al., 2023), Consensus Merging (Wang et al., 2024b), or LiNeS (Wang et al., 2025). Best performing result in each group is indicated in bold, while the second best is underlined.

| **Method** | **LiNeS** | **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 |
| Zero-Shot | – | 48.3 | 57.3 | 56.1 | 55.5 | 61.4 | 59.8 | 64.8 | 68.3 | 65.3 |
| Finetuned | – | 90.5 | 89.5 | 90.4 | 92.6 | 91.6 | 92.3 | 94.0 | 93.3 | 94.0 |
| Task Arithmetic | ✗ | 69.7 | 65.0 | 60.3 | 74.6 | 70.4 | 65.7 | 84.0 | 79.2 | 74.0 |
| | ✓ | 74.2 | 69.1 | 63.4 | 77.6 | 72.7 | 67.7 | 86.5 | 82.2 | 77.1 |
| + *Subspace Boosting* (**ours**) | ✗ | 83.1 | 75.8 | 66.4 | 87.7 | 82.0 | 71.6 | 91.4 | 86.2 | 80.6 |
| | ✓ | **85.6** | **80.8** | **77.2** | **88.8** | **84.7** | **80.0** | **92.6** | **89.3** | 87.2 |
| TIES-Merging | ✗ | 73.6 | 67.6 | 63.1 | 79.1 | 73.0 | 68.1 | 85.6 | 79.3 | 75.6 |
| | ✓ | 77.2 | 72.1 | 67.2 | 79.9 | 75.2 | 71.2 | 88.0 | 82.5 | 79.6 |
| + *Subspace Boosting* (**ours**) | ✗ | 81.8 | 74.4 | 69.8 | 87.0 | 80.5 | 75.9 | 91.1 | 83.6 | 82.0 |
| | ✓ | 83.8 | 79.1 | 75.9 | 87.4 | 83.3 | 79.7 | 91.9 | 86.1 | 85.9 |
| Consensus Merging | ✗ | 74.5 | 70.1 | 65.3 | 78.9 | 73.9 | 70.2 | 85.2 | 81.9 | 78.7 |
| | ✓ | 77.1 | 73.6 | 68.6 | 79.5 | 75.8 | 72.0 | 87.3 | 84.0 | 81.0 |
| + *Subspace Boosting* (**ours**) | ✗ | 82.7 | 77.1 | 73.2 | 87.0 | 81.9 | 77.6 | 91.5 | 86.4 | 84.9 |
| | ✓ | 84.4 | 80.3 | **77.2** | 87.6 | 84.2 | **80.0** | 92.2 | 88.8 | **87.9** |

**Matrix Decomposition Comparison via Alignment Matrices** To evaluate the significance of dimension $V_k$ (of the shared space) for one matrix relative to another, we take the ratio of their corresponding generalized singular values $\sigma_{i,k} / \sigma_{j,k}$. A ratio close to one indicates a **common** subspace, while a ratio deviating highly from one suggests that the dimension is more **unique** or important to one of the matrices. This allows us to express each matrix as a sum of common and unique components:

$$A_i = \underbrace{\sum_{k \in \mathcal{I}_{>1}} \sigma_{i,k} u_{i,k} v_k^T}_{\textbf{common}} + \underbrace{\sum_{k \in \mathcal{I}_1} \sigma_{i,k} u_{i,k} v_k^T}_{\textbf{unique}}, \tag{4}$$

where $\mathcal{I}_{>}1$ and $\mathcal{I}_1$ denote the common and unique subspaces, respectively.

To quantify the degree of interference between two matrices, we introduce the *Alignment Matrix* $\mathbf{A} \in \mathbb{R}^{N \times N}$. We define matrices as *well-aligned* if they rely on different subspaces (low interference) and *poorly aligned* if they share important subspaces (high interference). The alignment between matrices $A_i$ and $A_j$ is calculated as the average log ratio of their generalized singular values across all $L$ weight matrices from the respective layers:

$$(\mathbf{A})_{ij} = \frac{1}{L} \sum_{l=1}^{L} \left( \frac{1}{M_l} \sum_{p=1}^{M_l} \left| \log \left( \frac{\sigma_{i,p}^{(l)} + \epsilon}{\sigma_{j,p}^{(l)} + \epsilon} \right) \right| \right) \tag{5}$$

where $\sigma_{i,p}^{(l)}$ is the $p$-th generalized singular value of model $i$ for $l$-th weight matrix, and $M_l$ is the number of generalized singular values for that matrix, with $\epsilon = 1e^{-12}$ for stability. Unlike other methods such as direct weight comparisons, this approach offers a novel way to *compare task vectors within a shared, decomposed subspace*. This allows us to optimize subspace overlap to select diverse, low interference experts.

Finally, we also introduce *Higher-Order Subspace Boosting* by replacing SVD with HO-GSVD in *Subspace Boosting*. This is the first interpretable model merging method, achieving strong performance. For further details, please refer to Supplementary Sec. F.

# 5 Experiments

**Baselines and Datasets.** As described in Sec. 4.1, we apply *Subspace Boosting* to the task vectors obtained from several state-of-the-art model merging techniques (Ilharco et al., 2023a; Yadav et al., 2023; Wang et al., 2024b). The foundational method, Task Arithmetic (Ilharco et al., 2023a), defines task vectors and merges them via simple averaging. Building on this, TIES-Merging (Yadav et al., 2023) reduces interference by pruning low-magnitude weights from each vector before performing a sign-aware averaging. Consensus Merging (Wang et al., 2024b) further refines this by only keeping weights that are important for at least two of the tasks being merged. We also evaluate compatibilty with *LiNeS* (Wang et al., 2025), an orthogonal post-processing technique that scales updates based on layer depth.

*Datasets.* We follow the previous state-of-the-art methods (Wang et al., 2024b; 2025) and evaluate our model merging techniques on image classification tasks, considering the same grouping of 8, 14, and 20 tasks. For language tasks, we follow Tam et al. (2024a) and evaluate on 8 QA tasks, presented by Zhou et al. (2023) and 7 NLP tasks, provided by Yadav et al. (2023).

**Implementation Details.** Following previous works (Ilharco et al., 2023a; Wang et al., 2024b; 2025; Yadav et al., 2023), we employ the CLIP model (Radford et al., 2021a) with ViT-B/32, ViT-B/16 and ViT-L/14 as vision encoders. We use the pretrained and finetuned checkpoints provided by Wang et al. (2024b) and utilize the official code provided by the original authors for all model merging techniques, including the official hyperparameters. *Subspace Boosting* requires only one hyperparameter $\beta$ that determines the number of dimensions to be boosted. We tune $\beta$ on the validation set by performing a simple search over the set $\{0, 0.01, 0.02\}$. The updated task vector components of the Transformer architecture (Dosovitskiy et al., 2021) are the linear layers and the attention layers. For the language domain, we utilize T5 transformers (Raffel et al., 2020), provided by Tam et al. (2024a) and apply our method to the same layers.

## 5.1 *Subspace Boosting* enhances performance across vision and language tasks

Figure 1a illustrates the impact of *Subspace Boosting* on the stable rank scores of task vectors obtained from TA, TIES and Consensus Merging. Notably, *Subspace Boosting* successfully utilizes the available weight space by increasing the matrix rank. This enables the models to better populate the corresponding subspaces and mitigates rank collapse as more and more experts are incorporated. Additional results are provided in Supplementary Sec D.

Table 2: **Language Results.** Performance comparison (in terms of accuracy in %) on two language task collections. Our *Subspace Boosting* variant with TA and LiNeS achieves the best performance on both benchmarks.

| Method | 8 Tasks | 7 Tasks |
|---|---|---|
| Zero-shot | 33.1 | 44.9 |
| Fine-tuned | 80.7 | 85.9 |
| Task Arithmetic (TA) | 63.8 | 71.9 |
| TIES Merging | 63.0 | 71.6 |
| Consensus Merging | 68.6 | 73.5 |
| TA + LiNeS | 67.6 | 76.4 |
| Subspace Boosting (**ours**) | **75.3** | **83.0** |

**Vision Results.** As shown in Table 1, *Subspace Boosting* significantly improves performance of standard merging techniques across 8, 14 and 20 vision tasks. By enabling the methods to leverage a higher effective subspace, our approach yields substantial gains for all methods; for example it boosts the accuracy of simple Task Arithmetic from 65.0% to 75.8% when merging 14 tasks. Notably, this elevates all baselines to comparable performance.

To demonstrate its generality, we show that *Subspace Boosting* also enhances orthogonal techniques such as LiNeS. When merging 14 ViT-B/32 experts, LiNeS alone improves Task Arithmetic's performance from 65.0% to 69.1%, whereas applying *Subspace Boosting* provides a further, substantial improvement to 80.8%, highlighting its complementarity. Overall, the performance gains remain substantial across methods (as illustrated in Fig. 1a), model sizes, and expert model counts.

**Language Results.** We also evaluate *Subspace Boosting* with TA and LiNeS using popular language benchmarks. For 8 QA tasks (Zhou et al., 2023), *Subspace Boosting* shows the same significant improvements as in the vision domain, improving TA by around 12% when applied with LiNeS. For 7 NLP tasks, provided by Yadav et al. (2023), the improvements are of similar magnitude.

Table 3: ***Subspace Boosting* Ablations.** We analyze the impact of the boosting threshold $\beta$, the layer type, and layer-specific $\beta$. The results are reported for 8 ViT-B/16 models merged via TA (Ilharco et al., 2023a). (a) *Subspace Boosting* is highly robust to the choice of $\beta$, (b) performs best when applied to all layers, and (c) requires no tuning when modifying both weight types.

(a) **Boosting threshold**. *Subspace Boosting* demonstrates robustness to variations of the $\beta$ value.

| $\beta$ | Accuracy (%) |
|---|---|
| 0.00 | 87.7 |
| 0.01 | 87.4 |
| 0.02 | 87.2 |

(b) **Layers**. We observe that both the fully connected (FC) and the attention (Attn) layers contribute to the performance gain.

| Layer | Accuracy (%) |
|---|---|
| FC | 86.5 |
| Attn | 83.9 |
| Both | 87.7 |

(c) **Boosting threshold across layers**. In this setting, the attention (Attn) and fully connected (FC) share the same optimal $\beta$.

| FC | Attn | Accuracy (%) |
|---|---|---|
| 0.00 | 0.00 | 87.7 |
| 0.00 | 0.01 | 87.6 |
| 0.01 | 0.01 | 87.4 |
| 0.02 | 0.01 | 87.4 |

Table 4: **Comparison to State-of-the-Art.** Our best-performing *Subspace Boosting* variant with LiNeS achieves results comparable to other state-of-the-art methods, but at a fraction of the computational overhead and complexity.

| 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 | 90.5 | 89.5 | 90.4 | 92.6 | 91.6 | 92.3 | 94.0 | 93.3 | 94.0 |
| TSV-M† (Gargiulo et al., 2025) | 83.8 | 79.5 | 76.7 | 87.2 | 83.7 | 80.3 | 91.2 | 88.3 | 87.3 |
| Iso-C† (Marczak et al., 2025) | 83.8 | 79.1 | 74.7 | 88.6 | 83.3 | 79.0 | 92.4 | 88.2 | 87.0 |
| Iso-CTS† (Marczak et al., 2025) | <u>83.8</u> | <u>80.2</u> | <u>77.3</u> | **89.1** | <u>85.0</u> | **81.6** | **93.0** | **89.6** | **89.3** |
| TA + *Subspace Boosting* (**ours**) | **85.6** | **81.7** | **77.6** | <u>88.9</u> | **85.1** | <u>81.1</u> | <u>92.7</u> | <u>89.4</u> | <u>88.0</u> |

### 5.2 Ablations

To better understand the behavior of *Subspace Boosting*, we conduct ablation studies reporting the results in Tab. 3. The reported results use 8 ViT-B/16 models merged with TA (Ilharco et al., 2023a).

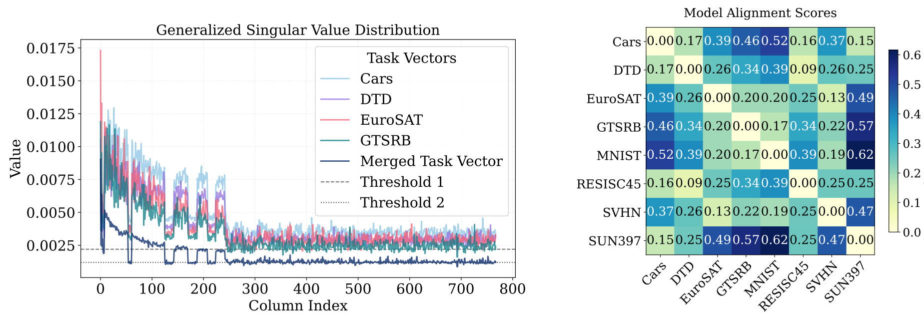
Our analysis investigates three key aspects. Firstly, we investigate the robustness to the boosting parameter $\beta$. As shown in Table 3a, our method is robust to variations of $\beta$, since performance only fluctuates between $87.2\%$ and $87.7\%$ when $\beta \in [0.00, 0.01, 0.02]$. Secondly, we examine the contributions of different layer types, seen in Table 3b. We observe that both the attention and fully connected (FC) layers contribute to improving performance, with the best accuracy of $87.7\%$ achieved when applying *Subspace Boosting* to the full model. Finally, we analyze whether attention layers require a different value for $\beta$ than FC layers. As reported in Table 3c, for *Subspace Boosting*, the attention and FC layers share the same optimal $\beta$ value. This reveals that *Subspace Boosting* can be applied across both weight types without additional tuning.

**State-of-the-Art Comparison.** We compare our method against recent state-of-the-art techniques (Gargiulo et al., 2025; Marczak et al., 2025) in Table 4. We equivalently tune the merging coefficient $\alpha$ over 30 interpolation points. The results demonstrate that simple TA enhanced with LiNeS and *Subspace Boosting* surpasses both TSV-M and Iso-C and matches or surpasses Iso-CTS. This is particularly evident for ViT-B/32 models, outperforming Iso-CTS by almost 2% for 8 models. However, unlike the other methods, *Subspace Boosting* is both compatible with all other TA based methods as well as over 6x more computationally efficient (details in Supplementary Sec. C).

### 5.3 Interpretable *Subspace Boosting* via HO-GSVD

**Comparison of Generalized Singular Values.** A key limitation of modern model merging is its lack of interpretability. To address this, we leverage HO-GSVD to enable a transparent investigation of the merging process and to compare contributions of individual experts. Fig. 5a illustrates this by visualizing the distribution of generalized singular values for an exemplary weight matrix of a ViT-B/16 attention block projection layer. We compare the generalized singular value distribution of 4 task vectors (Cars, DTD, Eurosat, GTSRB), against the averaged task vector over 8 tasks.

Since HO-GSVD operates on a shared decomposition space, we can directly compare generalized singular values across tasks. The plot reveals a large proportion of shared generalized singular vec-




(a) (b)

Figure 5: **(a) Distribution of Generalized Singular Values** across different task vectors and a merged reference of eight task vectors. **(b) Model Alignment via HO-GSVD**, showcasing how HO-GSVD can be used to contrast expert alignments across a shared decomposition space.

tors among the tasks, implying that the top dimensions experience high interference. Furthermore, this visualization clearly illustrates rank collapse without needing additional metrics. For the merged model, a significant number of singular values (in the 0-200 index range) are near zero. In contrast, the generalized singular values for the independent task vectors remain well above this floor (Threshold 1), indicating that the merged task vector occupies a lower-rank subspace. Finally, we observe that the merged task vector is significantly below the others, implying that the used merging coefficient $1/N$ for the average is suboptimal, pointing to a new promising direction for future research: using HO-GSVD to *automatically* choose the optimal merging coefficient.

**Selecting Optimal Experts For Better Performance.** Another benefit of HO-GSVD is comparing the relative significance of a certain subspace for matrix $A_i$ compared to matrix $A_j$, as described in Sec. 4.2. This can be used to select the optimal set of models for merging, which is a combinatorial complex problem (e.g. selecting 6 out of 20 leads to nearly 40,000 possible permutations).

HO-GSVD enables us to utilize the previously defined *Alignment Matrix*, where the higher the value for a pair of models ($A_i$, $A_j$), the easier it is to merge both models with reduced interference. An example of an *Alignment matrix* is presented in Fig. 5b. Higher scores correspond to models that are more well aligned (easier to merge).

Table 5: **HO-GSVD facilitates expert selection.** Choosing experts from a larger pool via HO-GSVD improves/maintains transfer to candidate tasks (*Pool*) *while* maintain transfer to external (*Ext*).

| Method | Accuracy | |
|---|---|---|
| | Pool | Ext |
| Random | 72.9 | 47.6 |
| HO-GSVD (ours) | 75.7 | 47.6 |

To demonstrate the effectivenss of the *Alignment Matrix* for 20 ViT models (more results in Supplementary Sec. G) for expert selection, we design an experiment to construct a merged model of 8 experts. Three of these experts are pre-selected to cover in-distribution (Pool) tasks. The remaining five are chosen from 14 candidates either via a random baseline (averaged over 10 draws) or our proposed method, which selects models that minimize interference with the pre-selected Pool tasks.

The final merged models are evaluated on the Pool tasks as well as a separate set of 3 out-of-distribution (OOD) tasks. As shown in Table 5, our proposed method improves in-distribution performance, boosting accuracy from 72.9% to 75.7% while maintaining performance on OOD tasks. This result demonstrates the efficacy of applying HO-GSVD for model selection.

## 6 Conclusion

In this work, we identified *rank collapse* as a fundamental limitation of task-arithmetic model merging methods Ilharco et al. (2023b); Yadav et al. (2023); Wang et al. (2024c). Consequently, we proposed *Subspace Boosting* to mitigate this limitation, thereby achieving significant performance improvements that exceed 10% in a wide range of settings across vision and language domains. Additionally, we provide a novel framework using HO-GSVD to address the black-box nature of modern model merging techniques. Moreover, we demonstrated that by comparing shared task subspaces via the *Alignment Matrix*, we can effectively evaluate the behavior of model merging and offer a strong approach to select a subset of models that achieve higher performance when merged.

# References


Samuel Ainsworth, Jonathan Hayase, and Siddhartha Srinivasa. Git re-basin: Merging models modulo permutation symmetries. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=CQsmMYmlP5T.

Rishi Bommasani, Drew A. Hudson, Ehsan Adeli, and Russ Altman et al. On the opportunities and risks of foundation models. *ArXiv*, 2021. URL https://crfm.stanford.edu/assets/report.pdf.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, NIPS '20, Red Hook, NY, USA, 2020. Curran Associates Inc. ISBN 9781713829546.

Jiho Choi, Donggyun Kim, Chanhyuk Lee, and Seunghoon Hong. Revisiting weight averaging for model merging, 2025. URL https://arxiv.org/abs/2412.12153.

MohammadReza Davari and Eugene Belilovsky. Model breadcrumbs: Scaling multi-task model merging with sparse masks. In Aleš Leonardis, Elisa Ricci, Stefan Roth, Olga Russakovsky, Torsten Sattler, and Gül Varol (eds.), *Computer Vision – ECCV 2024*, pp. 270–287, Cham, 2025. Springer Nature Switzerland. ISBN 978-3-031-73226-3.

Pala Tej Deep, Rishabh Bhardwaj, and Soujanya Poria. Della-merging: Reducing interference in model merging through magnitude-based sampling. *arXiv preprint arXiv:2406.11617*, 2024.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding, 2018. URL http://arxiv.org/abs/1810.04805. cite arxiv:1810.04805Comment: 13 pages.

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. URL https://openreview.net/forum?id=YicbFdNTTy.

Sebastian Dziadzio, Vishaal Udandarao, Karsten Roth, Ameya Prabhu, Zeynep Akata, Samuel Albanie, and Matthias Bethge. How to merge your multimodal models over time? In *CVPR*, 2025.

Rahim Entezari, Hanie Sedghi, Olga Saukh, and Behnam Neyshabur. The role of permutation invariance in linear mode connectivity of neural networks. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=dNigytemkL.

Jonathan Frankle, Gintare Karolina Dziugaite, Daniel Roy, and Michael Carbin. Linear mode connectivity and the lottery ticket hypothesis. In Hal Daumé III and Aarti Singh (eds.), *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pp. 3259–3269. PMLR, 13–18 Jul 2020. URL https://proceedings.mlr.press/v119/frankle20a.html.

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 *CVPR*, 2025.

Timur Garipov, Pavel Izmailov, Dmitrii Podoprikhin, Dmitry P Vetrov, and Andrew G Wilson. Loss surfaces, mode connectivity, and fast ensembling of dnns. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018. URL https://proceedings.neurips.cc/paper_files/paper/2018/file/be3087e74e9100d4bc4c6268cdbe8456-Paper.pdf.

Charles Goddard, Shamane Siriwardhana, Malikeh Ehghaghi, Luke Meyers, Vlad Karpukhin, Brian Benedict, Mark McQuade, and Jacob Solawetz. Arcee's mergekit: A toolkit for merging large language models. *arXiv preprint arXiv:2403.13257*, 2024.

Gene H. Golub and Charles F. Van Loan. *Matrix Computations - 4th Edition*. Johns Hopkins University Press, 2013. doi: 10.1137/1.9781421407944. URL https://epubs.siam.org/doi/abs/10.1137/1.9781421407944.

Yifei He, Yuzheng Hu, Yong Lin, Tong Zhang, and Han Zhao. Localize-and-stitch: Efficient model merging via sparse task arithmetic. *arXiv preprint arXiv:2408.13656*, 2024.

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*, 2023a. URL https://openreview.net/forum?id=6t0Kwf8-jrj.

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*, 2023b. URL https://openreview.net/forum?id=6t0Kwf8-jrj.

Pavel Izmailov, Dmitrii Podoprikhin, Timur Garipov, Dmitry Vetrov, and Andrew Gordon Wilson. Averaging weights leads to wider optima and better generalization. *arXiv preprint arXiv:1803.05407*, 2018.

Dong-Hwan Jang, Sangdoo Yun, and Dongyoon Han. Model stock: All we need is just a few fine-tuned models. In *Computer Vision – ECCV 2024: 18th European Conference, Milan, Italy, September 29–October 4, 2024, Proceedings, Part XLIV*, pp. 207–223, Berlin, Heidelberg, 2024. Springer-Verlag. ISBN 978-3-031-72783-2. doi: 10.1007/978-3-031-72784-9_12. URL https://doi.org/10.1007/978-3-031-72784-9_12.

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. URL https://openreview.net/forum?id=FCnohuR6AnM.

Idris Kempf, Paul J. Goulart, and Stephen R. Duncan. A higher-order generalized singular value decomposition for rank-deficient matrices. *SIAM Journal on Matrix Analysis and Applications*, 44(3):1047–1072, 2023.

Chanhyuk Lee, Jiho Choi, Chanryeol Lee, Donggyun Kim, and Seunghoon Hong. Adarank: Adaptive rank pruning for enhanced model merging, 2025. URL https://arxiv.org/abs/2503.22178.

Charles F. Van Loan. Generalizing the singular value decomposition. *SIAM Journal on Numerical Analysis*, 13(1):76–83, 1976. ISSN 00361429. URL http://www.jstor.org/stable/2156468.

Jinliang Lu, Ziliang Pang, Min Xiao, Yaochen Zhu, Rui Xia, and Jiajun Zhang. Merge, ensemble, and cooperate! a survey on collaborative strategies in the era of large language models. *arXiv preprint arXiv:2407.06089*, 2024.

Daniel Marczak, Bartłomiej Twardowski, Tomasz Trzciński, and Sebastian Cygert. Magmax: Leveraging model merging for seamless continual learning. *arXiv preprint arXiv:2407.06322*, 2024.

Daniel Marczak, Simone Magistri, Sebastian Cygert, Bartlomiej Twardowski, Andrew D. Bagdanov, and Joost van de Weijer. No task left behind: Isotropic model merging with common and task-specific subspaces. *ICML*, 2025.

Timo Milbich, Karsten Roth, Homanga Bharadhwaj, Samarth Sinha, Yoshua Bengio, Björn Ommer, and Joseph Paul Cohen. Diva: Diverse visual feature aggregation for deep metric learning. In *Computer Vision – ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part VIII*, pp. 590–607, Berlin, Heidelberg, 2020. Springer-Verlag. ISBN 978-3-030-58597-6. doi: 10.1007/978-3-030-58598-3_35. URL https://doi.org/10.1007/978-3-030-58598-3_35.

Jishnu Mukhoti, Yarin Gal, Philip Torr, and Puneet K. Dokania. Fine-tuning can cripple your foundation model; preserving features may be the solution. *Transactions on Machine Learning Research*, 2024. ISSN 2835-8856. URL https://openreview.net/forum?id=kfhoeZCeW7. Featured Certification.

Anshul Nasery, Jonathan Hayase, Pang Wei Koh, and Sewoong Oh. Pleas–merging models with permutations and least squares. *arXiv preprint arXiv:2407.02447*, 2024.

Behnam Neyshabur, Hanie Sedghi, and Chiyuan Zhang. What is being transferred in transfer learning? In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 512–523. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper_files/paper/2020/file/0607f4c705595b911a4f3e7a127b44e0-Paper.pdf.

Kai Nylund, Suchin Gururangan, and Noah A Smith. Time is encoded in the weights of finetuned language models. *arXiv preprint arXiv:2312.13401*, 2023.

Changdae Oh, Yixuan Li, Kyungwoo Song, Sangdoo Yun, and Dongyoon Han. Dawin: Training-free dynamic weight interpolation for robust adaptation. *arXiv preprint arXiv:2410.03782*, 2024.

Guillermo Ortiz-Jimenez, Alessandro Favero, and Pascal Frossard. Task arithmetic in the tangent space: Improved editing of pre-trained models. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL https://openreview.net/forum?id=0A9f2jZDGW.

Jyothish Pari, Samy Jelassi, and Pulkit Agrawal. Collective model intelligence requires compatible specialization. *arXiv preprint arXiv:2411.02207*, 2024.

Sri Priya Ponnapalli, Michael A. Saunders, Charles F. Van Loan, and Orly Alter. A higher-order generalized singular value decomposition for comparison of global mrna expression from multiple organisms. *PLOS ONE*, 6:1–11, 12 2011.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. *OpenAI*, 2019. URL https://cdn.openai.com/better-language-models/language_models_are_unsupervised_multitask_learners.pdf. Accessed: 2024-11-15.

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 *International Conference on Machine Learning*, 2021a. URL https://api.semanticscholar.org/CorpusID:231591445.

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, 2021b. URL https://arxiv.org/abs/2103.00020.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *J. Mach. Learn. Res.*, 21(1), January 2020. ISSN 1532-4435.

Alexandre Rame, Kartik Ahuja, Jianyu Zhang, Matthieu Cord, Leon Bottou, and David Lopez-Paz. Model ratatouille: Recycling diverse models for out-of-distribution generalization. In Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett (eds.), *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 28656–28679. PMLR, 23–29 Jul 2023. URL https://proceedings.mlr.press/v202/rame23a.html.

Alexandre Ramé, Johan Ferret, Nino Vieillard, Robert Dadashi, Léonard Hussenot, Pierre-Louis Cedoz, Pier Giuseppe Sessa, Sertan Girgin, Arthur Douillard, and Olivier Bachem. Warp: On the benefits of weight averaged rewarded policies. *arXiv preprint arXiv:2406.16768*, 2024.

Alexandre Ramé, Matthieu Kirchmeyer, Thibaud Rahier, Alain Rakotomamonjy, Patrick Gallinari, and Matthieu Cord. Diverse weight averaging for out-of-distribution generalization. In *Proceedings of the 36th International Conference on Neural Information Processing Systems*, NIPS '22, 2024. ISBN 9781713871088.

Mark Rofin, Nikita Balagansky, and Daniil Gavrilov. Linear interpolation in parameter space is good enough for fine-tuned language models. *arXiv preprint arXiv:2211.12092*, 2022.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10684–10695, 2022.

Karsten Roth, Vishaal Udandarao, Sebastian Dziadzio, Ameya Prabhu, Mehdi Cherti, Oriol Vinyals, Olivier Hénaff, Samuel Albanie, Matthias Bethge, and Zeynep Akata. A practitioner's guide to continual multimodal pretraining. *arXiv preprint arXiv:2408.14471*, 2024.

Amartya Sanyal, Philip H. Torr, and Puneet K. Dokania. Stable rank normalization for improved generalization in neural networks and gans. In *International Conference on Learning Representations*, 2020. URL `https://openreview.net/forum?id=H1enKkrFDB`.

Sunny Sanyal, Atula Tejaswi Neerkaje, Jean Kaddour, Abhishek Kumar, and sujay sanghavi. Early weight averaging meets high learning rates for LLM pre-training. In *First Conference on Language Modeling*, 2024. URL `https://openreview.net/forum?id=IA8CWtNkUr`.

Peter H. Schönemann. A generalized solution of the orthogonal procrustes problem. *Psychometrika*, 31(1):1–10, Mar 1966. ISSN 1860-0980. doi: 10.1007/BF02289451.

Ekansh Sharma, Daniel M Roy, and Gintare Karolina Dziugaite. The non-local model merging problem: Permutation symmetries and variance collapse. *arXiv preprint arXiv:2410.12766*, 2024.

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.

Ken Shoemake. Animating rotation with quaternion curves. *Proceedings of the 12th annual conference on Computer graphics and interactive techniques*, 1985. URL `https://api.semanticscholar.org/CorpusID:11290566`.

Prarabdh Shukla, Gagan Raj Gupta, and Kunal Dutta. DiffRed: Dimensionality reduction guided by stable rank. In Sanjoy Dasgupta, Stephan Mandt, and Yingzhen Li (eds.), *Proceedings of The 27th International Conference on Artificial Intelligence and Statistics*, volume 238 of *Proceedings of Machine Learning Research*, pp. 3430–3438. PMLR, 02–04 May 2024. URL `https://proceedings.mlr.press/v238/shukla24a.html`.

George Stoica, Pratik Ramesh, Boglarka Ecsedi, Leshem Choshen, and Judy Hoffman. Model merging with svd to tie the knots. *arXiv preprint arXiv:2410.19735*, 2024.

Yi-Lin Sung, Linjie Li, Kevin Lin, Zhe Gan, Mohit Bansal, and Lijuan Wang. An empirical study of multimodal model merging. *arXiv preprint arXiv:2304.14933*, 2023.

Derek Tam, Mohit Bansal, and Colin Raffel. Merging by matching models in task parameter subspaces, 2024a. URL `https://arxiv.org/abs/2312.04339`.

Derek Tam, Yash Kant, Brian Lester, Igor Gilitschenski, and Colin Raffel. Realistic evaluation of model merging for compositional generalization. *arXiv preprint arXiv:2409.18314*, 2024b.

Ke Wang, Nikolaos Dimitriadis, Alessandro Favero, Guillermo Ortiz-Jimenez, Francois Fleuret, and Pascal Frossard. Lines: Post-training layer scaling prevents forgetting and enhances model merging. *arXiv preprint arXiv:2410.17146*, 2024a.

Ke Wang, Nikolaos Dimitriadis, Guillermo Ortiz-Jiménez, François Fleuret, and Pascal Frossard. Localizing task information for improved model merging and compression. In *Proceedings of the 41st International Conference on Machine Learning*, ICML'24, 2024b.

Ke Wang, Nikolaos Dimitriadis, Guillermo Ortiz-Jimenez, François Fleuret, and Pascal Frossard. Localizing task information for improved model merging and compression. *arXiv preprint arXiv:2405.07813*, 2024c.

Ke Wang, Nikolaos Dimitriadis, Alessandro Favero, Guillermo Ortiz-Jimenez, François Fleuret, and Pascal Frossard. Lines: Post-training layer scaling prevents forgetting and enhances model merging. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=J5sUOvlLbQ.

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 Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvari, Gang Niu, and Sivan Sabato (eds.), *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pp. 23965–23998. PMLR, 17–23 Jul 2022a. URL https://proceedings.mlr.press/v162/wortsman22a.html.

Mitchell Wortsman, Gabriel Ilharco, Jong Wook Kim, Mike Li, Simon Kornblith, Rebecca Roelofs, Raphael Gontijo Lopes, Hannaneh Hajishirzi, Ali Farhadi, Hongseok Namkoong, and Ludwig Schmidt. Robust fine-tuning of zero-shot models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 7959–7971, June 2022b.

Feng Xiong, Runxi Cheng, Wang Chen, Zhanqiu Zhang, Yiwen Guo, Chun Yuan, and Ruifeng Xu. Multi-task model merging via adaptive weight disentanglement. *arXiv preprint arXiv:2411.18729*, 2024.

Prateek Yadav, Derek Tam, Leshem Choshen, Colin Raffel, and Mohit Bansal. TIES-merging: Resolving interference when merging models. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL https://openreview.net/forum?id=xtaX3WyCj1.

Prateek Yadav, Colin Raffel, Mohammed Muqeeth, Lucas Caccia, Haokun Liu, Tianlong Chen, Mohit Bansal, Leshem Choshen, and Alessandro Sordoni. A survey on model moerging: Recycling and routing among specialized experts for collaborative learning. *arXiv preprint arXiv:2408.07057*, 2024a.

Prateek Yadav, Tu Vu, Jonathan Lai, Alexandra Chronopoulou, Manaal Faruqui, Mohit Bansal, and Tsendsuren Munkhdalai. What matters for model merging at scale? *arXiv preprint arXiv:2410.03617*, 2024b.

Enneng Yang, Li Shen, Guibing Guo, Xingwei Wang, Xiaochun Cao, Jie Zhang, and Dacheng Tao. Model merging in llms, mllms, and beyond: Methods, theories, applications and opportunities. *arXiv preprint arXiv:2408.07666*, 2024a.

Enneng Yang, Li Shen, Zhenyi Wang, Guibing Guo, Xiaojun Chen, Xingwei Wang, and Dacheng Tao. Representation surgery for multi-task model merging. *arXiv preprint arXiv:2402.02705*, 2024b.

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*, 2024c.

Kerem Zaman, Leshem Choshen, and Shashank Srivastava. Fuse to forget: Bias reduction and selective memorization through model fusion. *arXiv preprint arXiv:2311.07682*, 2023.

Shenghe Zheng and Hongzhi Wang. Free-merging: Fourier transform for model merging with lightweight experts. *arXiv preprint arXiv:2411.16815*, 2024.

Jing Zhou, Zongyu Lin, Yanan Zheng, Jian Li, and Zhilin Yang. Not all tasks are born equal: Understanding zero-shot generalization. In *International Conference on Learning Representations (ICLR)*, 2023. URL https://openreview.net/pdf?id=KGV-GBh8fb.

Zihan Zhou, Xiaodong Li, John Wright, Emmanuel Candes, and Yi Ma. Stable principal component pursuit, 2010. URL https://arxiv.org/abs/1001.2363.

# Appendix Table of Contents



## A Limitations

Current Task Arithmetic-based methods, including *Subspace Boosting* and *Higher-Order Subspace Boosting*, require tuning the best merging coefficient. This requires access to a reasonable validation dataset as well as compute resources in order to tune this hyperparameter. In future work, we consider automating this and removing the necessity for tuning. Another limitation is that as the number of merged models grows, the performance continues to decrease. Therefore, finding optimal methods to prevent this degradation could be another interesting direction for future work.

## B Additional Implementation Details

**Hyperparameters**. The results presented with *Subspace Boosting* are obtained with $\beta$ optimized over the small set $\{0.00, 0.01, 0.02\}$. Given the robustness, for *Higher-Order Subspace Boosting*, we simply kept $\beta = 0.0$. In regards to merging coefficients, we utilized the same range as provided by (Wang et al., 2024c) for all our results $\lambda \in \{0.1, 0.2, ..., 1.0\}$. For additional baseline methods, such as TSV-M (Gargiulo et al., 2025) and Iso-C, Iso-CTS (Marczak et al., 2025), we extend the range to $\lambda \in \{0.1, 0.2, ..., 3.0\}$, as utilized by the original authors, and a range of 30 of $\lambda \in \{0.1, ..., 0.5\}$ for *Subspace Boosting*. We benchmark TSV-M, Iso-C, and Iso-CTS using the previously provided checkpoints, which are slightly inferior to the checkpoints used by (Gargiulo et al., 2025) and Iso-CTS (Marczak et al., 2025). For Consensus Merging (Wang et al., 2024b), all results are evaluated using Task Arithmetic as the baseline method.

**Models and Datasets.** To ensure direct replicability and comparability, we extend the repository provided by Wang et al. (2024c), and use the same baseline checkpoints, finetuned checkpoints and datasets as the authors. For the 8-, 14- and 20-task benchmarks, we use the same datasets as the original authors. For the language domain, we use the repository provided by Tam et al. (2024a) and the respective models and datasets.

## C Computational Resources

For all of our experiments, we leverage a compute cluster equipped with 2 NVIDIA GeForce RTX 2080 Ti with 12 GB VRAM and 6 NVIDIA Quadro RTX 6000 GPUs with 24 GB VRAM; alongside 2 Intel Xeon Silver 416 CPU @ 2.10 Ghz CPUs and 256 GB of RAM.

The running time for our *Subspace Boosting* compared to the state-of-the-art results and the task-arithmetic baselines is reported in Tab. 6. *Subspace Boosting* introduces additional computational overhead, however, we notice that the computational overhead for SVD and boosting is very negligible and in line with other, simpler methods. We notice that the computational overhead of our method is in line with the other, simpler methods such as TIES and over 6x more efficient in terms of clock-time than TSV-M and Iso-CTS (which extends TSV-M).

Table 6: Comparison of execution time (in seconds) of *Subspace Boosting* against other methods for 20 tasks. *Subspace Boosting* is 6$\times$ faster than TSV-M.

| **Method** | **ViT-B/32** | **ViT-B/16** | **ViT-L/14** |
|---|---|---|---|
| Task Arithmetic | 0.2s | 0.28s | 1s |
| Consensus Merging | 1.8s | 2.4s | 20.2s |
| TIES | 6.3s | 4.3s | 25.6s |
| *Subspace Boosting* **(ours)** | 9.7s | 10.5s | 40.1s |
| TSV-M | 62s | 63s | 210s |

## D Additional Rank Collapse Results

We also employ the *cumulative energy rank* to measure the rank collapse. The *cumulative energy rank* $\mathcal{M}_{\text{cer}}$ measures how much of the matrix information is captured by the top singular values, or more precisely how many singular values are required to cover $k\%$ of the energy $\mathcal{E} := \sum_i^n \sigma_i^2$.

We visualize rank collapse using both metrics, namely the stable rank and the cumulative energy rank. In Figure 6, the stable rank is plotted across different layers and components. Rank collapse is evident across the majority of the layers and components. For the projection layer of the MLP component, rank collapse is especially evident across all visualized layers.

Figure 7 shows the stable rank and the normalized value over an increasing number of merged models (4, 8, 14, 20) via Task Arithmetic (Ilharco et al., 2023a). To calculate the normalized values, each component's values are divided by the largest value for the respective component. This keeps the range (0-1) identical across all layers for easier comparison. As more models are merged, the stable rank clearly decreases in all layers and for almost all components. Although it would be expected for a merged model to maintain maximal rank as it must account for multiple tasks simultaneously, due to the rank collapse, naively merged models often rely on increasingly smaller subspaces for the classification of increasing number of tasks. This hinders the model's generalization capability as the task becomes more complex and higher in number.

We also employ the cumulative energy as a metric for the intrinsic dimension of the model across layers to provide another perspective on rank collapse. In Figure 8, the number of components necessary to sum up to 50% of the energy (cumulative energy rank) is visualized. We observe a similar trend to the previous results, and observe that for a vast majority of layers and components the cumulative energy rank decreases as we merge more models.

Moreover, we observe a direct correlation between the performance of various model merging methods and their respective cumulative energy rank (see Figure 9 for 14 merged models). For detailed performance reports, please refer to Table 1. It is noticeable that Task Arithmetic merging ranks are often small or negligible, and while TIES does maintain a higher cumulative energy rank, Consensus TA merging consistently achieves a higher rank - which is tied with increasing overall merging performance.

## E Additional Ablation Results

We perform various ablation experiments for the boosting threshold $\beta$ across different settings. In Table 7, we report the effect of the threshold for Task Arithmetic + Subspace Boosting. For the simple case of $0.0$, the performance is consistently the best. Therefore, we advise the practitioner to set the threshold to $0.0$ by default.

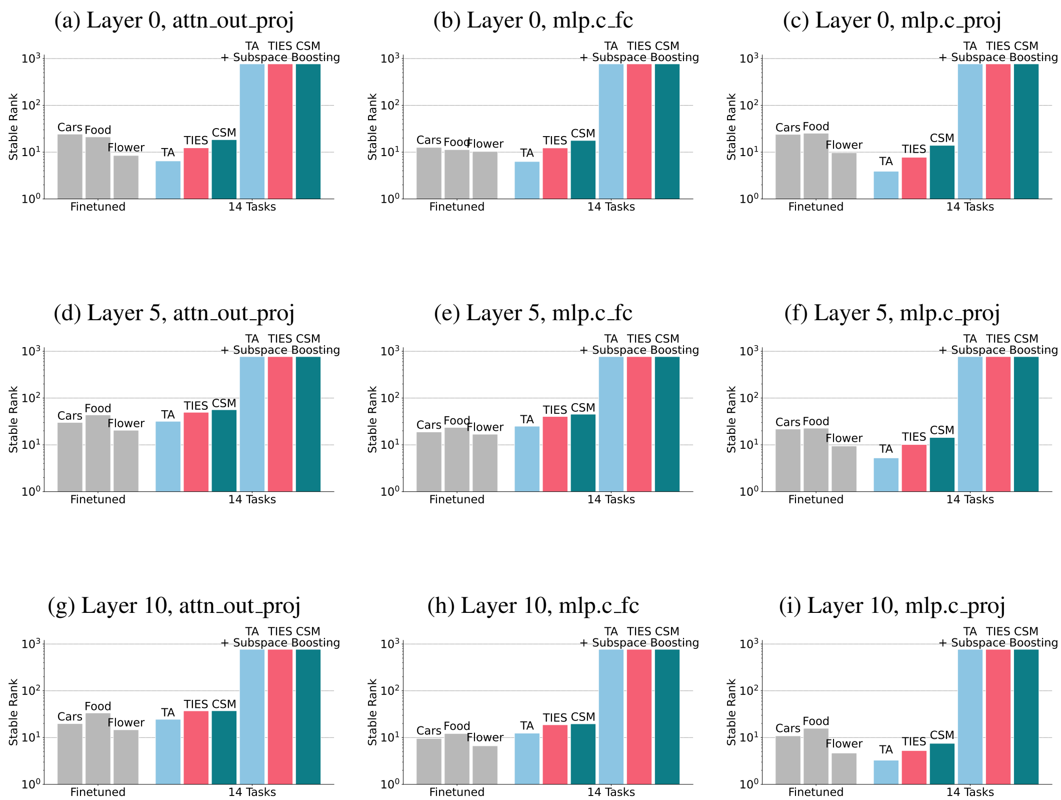
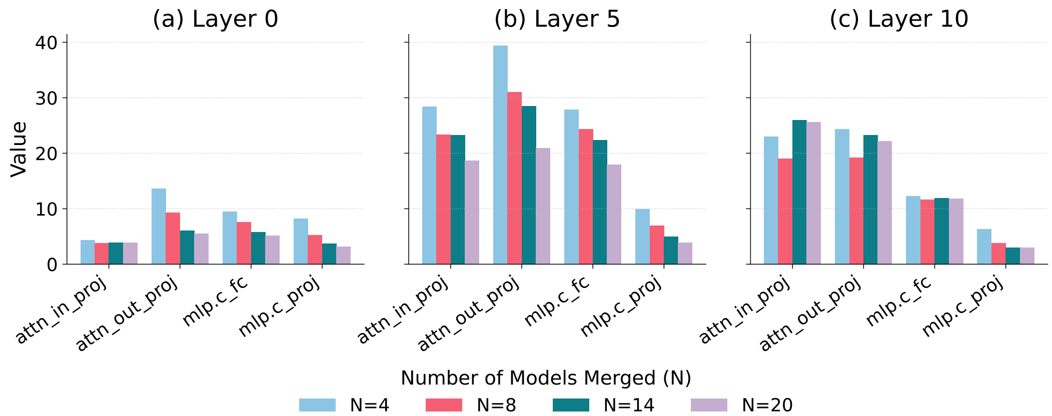
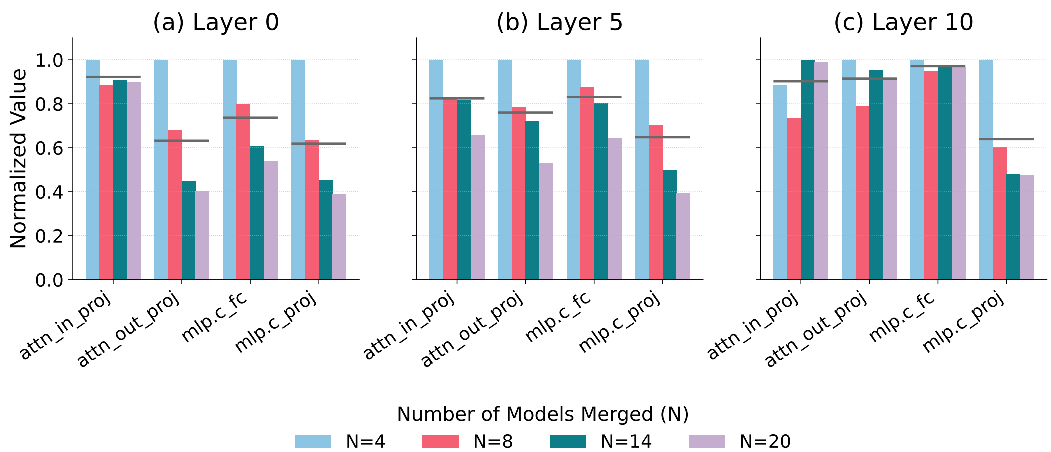



Figure 6: Stable rank visualized across multiple layers and different components. It is noticeable that the baseline methods TA, TIES and CSM exhibit small stable rank values, however by applying *Subspace Boosting* the stable rank score increases considerably. We report the layer and the component on top of each subplot.




(a) Stable rank per component.



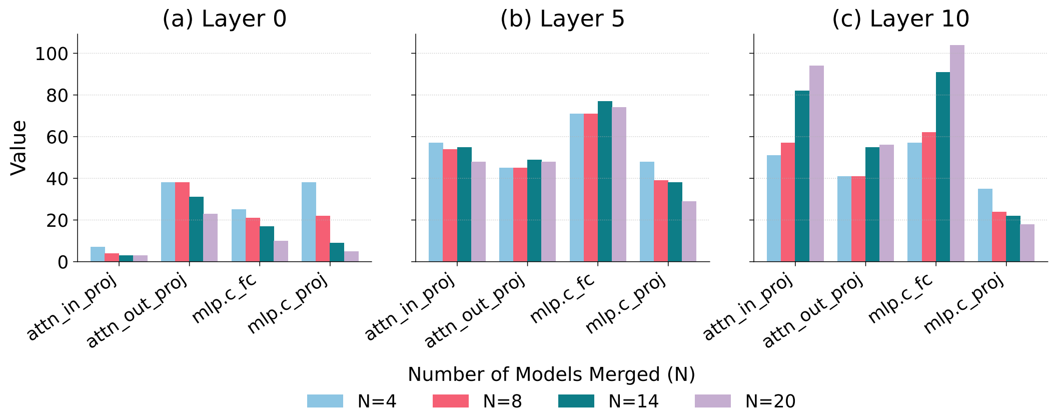

(b) Normalized stable rank per component. The mean is plotted in gray.

Figure 7: Absolute and normalized stable rank across a different number of merged models via Task Arithmetic. It is noticeable that merging more models decreases the rank of task vectors, limiting the expression space of the model.

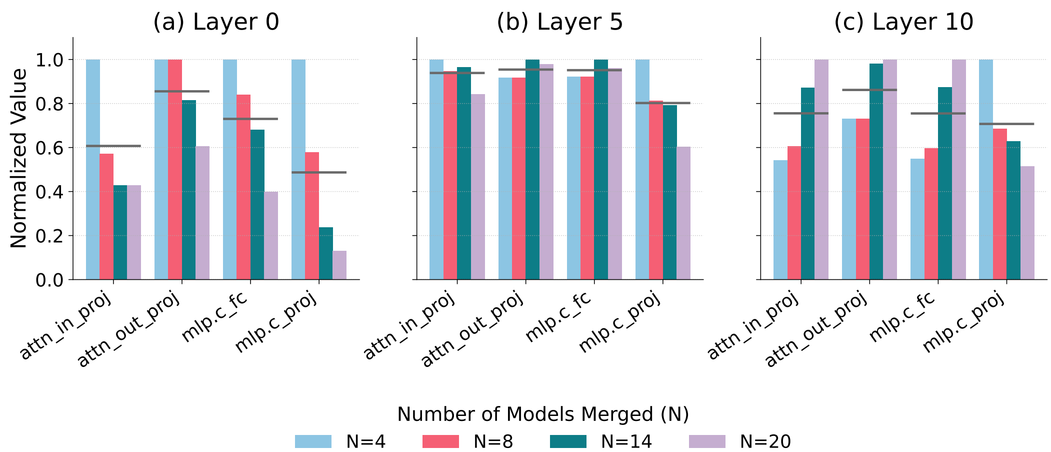
Table 7: Boosting threshold ($\beta$) ablation for Task Arithmetic + Subspace Boosting

| Threshold | 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 |
| $\beta = 0.00$ | **83.1** | **75.8** | 63.7 | **87.7** | **82.0** | 71.3 | **91.4** | **86.2** | **80.6** |
| $\beta = 0.01$ | 82.6 | 75.3 | 63.2 | 87.4 | 81.7 | 70.9 | 90.8 | 85.8 | 80.1 |
| $\beta = 0.02$ | 82.0 | 75.2 | **66.4** | 87.2 | 80.3 | **71.6** | 86.2 | 83.8 | 79.3 |

Table 8 shows the performance for different boosting thresholds $\beta$ for Task Arithmetic + Subspace Boosting + LiNeS, depending on the layer type. For both the fully-connected layers as well as the attention weight layer, we observe that the best performance is often achieved when setting both thresholds to 0.00. This shows that $\beta$ is quite robust to tuning and in all of our other results, we rely on one shared $\beta$ for both layer types.




(a) Absolute cumulative energy rank per component.




(b) Normalized cumulative energy rank per component. The mean is plotted in gray.

Figure 8: Absolute and normalized cumulative energy rank for 50% of the energy when Task Arithmetic is employed. It is noticeable that the cumulative energy rank decreases with the increase of the number of merged tasks in a majority layers, especially earlier ones, leading to rank collapse.

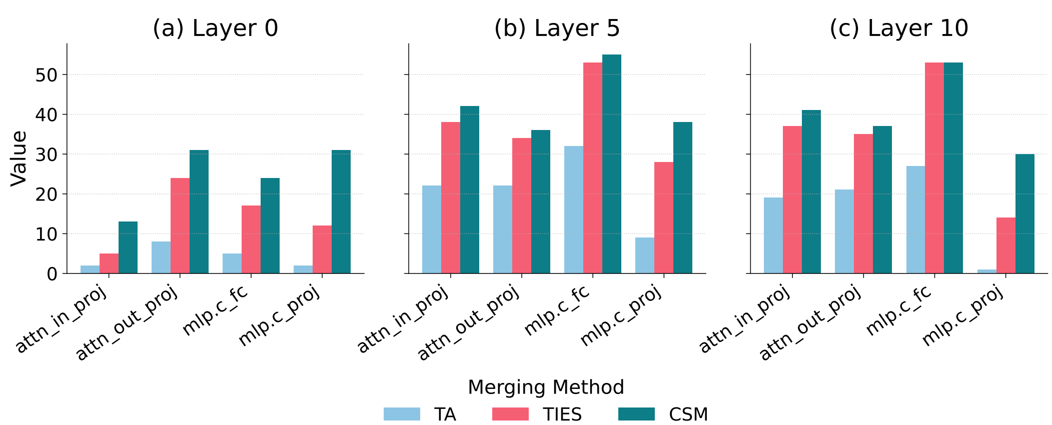



Figure 9: Cumulative energy rank for 30% of the energy for different merging methods when 14 tasks are merged.

Table 8: Ablation of optimal $\beta$ depending on layer.

| FC Layers | Attn Layer | 8 Tasks | 14 Tasks | 20 Tasks |
|---|---|---|---|---|
| 0.00 | 0.00 | **87.7** | **82.0** | 71.3 |
| 0.00 | 0.01 | 87.6 | 82.0 | 71.2 |
| 0.01 | 0.00 | 87.6 | 81.9 | 71.2 |
| 0.01 | 0.01 | 87.4 | 80.9 | 71.0 |
| 0.02 | 0.00 | 87.4 | 80.9 | **71.8** |
| 0.02 | 0.01 | 87.4 | 80.6 | 71.5 |

**Algorithm 2:** HO-GSVD with Subspace Boosting

**Input:** $\Delta_m = \{\Delta_{m_1}, \dots, \Delta_{m_N}\}$ – the individual task vectors for each model.
**Output:** $\Delta_{boost} = \{\Delta_{boost_1}, \dots, \Delta_{boost_K}\}$ – the *updated* merged task vectors for each component in the model.

$\Delta_{boost} \leftarrow \emptyset$; $\triangleleft$ initialize the task vector for merging.
$S_\pi \leftarrow 0$; $\triangleleft$ initialize $S_\pi$ to zero for all pairwise entries.
**foreach** $\Delta_{c_{k,i}} \in \Delta_{m_i}$ **do**
  **foreach** $\Delta_{c_{k,j}} \in \Delta_{m_j}$ **do**
    $S_\pi(i,j) \leftarrow \text{calculateS}({\Delta_{c_{k,i}}}^T \Delta_{c_{k,i}}, {\Delta_{c_{k,j}}}^T \Delta_{c_{k,j}})$; $\triangleleft$ calculate $S_\pi$ for each pair of components $k$.
  $(\Lambda, V_{\text{shared}}) \leftarrow \text{eig}(S_\pi)$ $\triangleleft$ compute the eigendecomposition of $S_\pi$.
  **foreach** $i \in N$ **do**
    $(U_i, \Sigma_i) \leftarrow \text{calculate\_sing\_matrices}(S_\pi, \Delta_{c_{k,i}})$ $\triangleleft$ calculate left singular vector matrix and the singular value matrix.
  $U_{\text{ortho}} \leftarrow \text{calculate\_ortho\_U}(U_1, ..., U_N)$ $\triangleleft$ calculate the orthonormal matrix for all Us via Generalized Procrustes.
  $V_{\text{ortho}} \leftarrow \text{calculate\_ortho\_V}(V_{\text{shared}})$ $\triangleleft$ calculate orthonormal matrix for V via Procrustes.
  $\Sigma_{\text{sum}} \leftarrow sum(\Sigma_1, ..., \Sigma_\text{N})$ $\triangleleft$ sum up the singular values into one shared matrix.
  $\Sigma_{\text{boosted}} \leftarrow \text{subspace\_boosting}(\Sigma_{\text{sum}})$ $\triangleleft$ boost the singular values with subspace boosting.
  $\Delta_{boost_k} \leftarrow U_{\text{ortho}} \cdot \Sigma_{\text{boosted}} \cdot V_{\text{ortho}}^T$; $\triangleleft$ recompute the task vector using new boosted singular values.
  $\Delta_{boost} \leftarrow \Delta_{boost} \cup \Delta_{boost_k}$; $\triangleleft$ add the updated component.
return $\Delta_{boost}$ $\triangleleft$ return merged task vector.

# F HO-GSVD: Additional Information & Details

## F.1 HO-GSVD Details and Higher-Order Subspace Boosting

Given a set of $N$ matrices $A_1, \dots, A_N$, HO-GSVD decomposes each matrix $A_i = U_i \Sigma_i V^T, \quad i = 1, \dots, N$ resulting in *distinct* $U_i \in \mathbb{R}^{m_i \times n}$, $\Sigma_i \in \mathbb{R}^{n \times n}$ and a *shared* subspace $V \in \mathbb{R}^{n \times n}$, identical for all factorizations. To obtain $V$, we solve the eigensystem $S_\pi V = V\Lambda$ of the arithmetic mean $S_\pi$ of all pairwise quotients $D_{i,\pi} D_{j,\pi}^{-1}$:

$$S_\pi = \frac{1}{N(N-1)} \sum_{i=1}^{N} \sum_{j=i+1}^{N} (D_{i,\pi} D_{j,\pi}^{-1} + D_{j,\pi} D_{i,\pi}^{-1}), \tag{6}$$

with $D_{i,\pi}$ defined as:

$$D_{i,\pi} = A_i^T A_i + \pi A^T A, \pi \geq 0, \tag{7}$$

where $A = [A_1^T, \dots, A_N^T]^T$. Following Kempf et al. (2023), we add the previous regularization term $A^T A$ and scale it by $\pi$ to extend HO-GSVD's applicability to *rank-deficient matrices*. This phenomenon frequently occurs for task vectors, as particularly visualized for merged task vectors in Fig. 1a. We set $\pi$ to $10^{-2}$ by default. As previously described, HO-GSVD introduces a shared subspace *identical across all inputs*. This enables the *direct comparison of different models* and the identification of common or unique subspaces for different tasks. HO-GSVD allows us to perform merging in a more interpretable manner by operating compositions over shared subspaces.

However, unlike standard SVD, the left and right singular matrices are generally not orthonormal (Ponnapalli et al., 2011; Kempf et al., 2023). In order to perform subspace boosting with this method, we must first orthonormalize the respective matrices. This is achieved by solving the Generalized Orthogonal Procrustes problem (Golub & Van Loan, 2013; Schönemann, 1966). Afterwards, the weight matrices can be reconstructed using these newly orthonormal matrices while preserving a shared subspace across all models. Please refer to Algorithm 2 for more details.

### F.2 Higher-Order Subspace Boosting Merging Performance

Similarly to its standard SVD counterpart *Subspace Boosting*, HO-GSVD can also be used for highly performant model merging, using Algorithm 2, which we refer to as *Higher-Order Subspace Boosting*. Table 9 showcases the performance of Higher-Order Subspace Boosting + LiNeS against other performant methods. We observe that the method consistently achieves strong performance. For ViT-B/32, for 8 tasks, it is the second best method, whereas for 14 tasks, it achieves the same performance as our best-performing Task Arithmetic variant. This showcases that *Higher-Order Subspace Boosting* is a strong model merging method and a potentially suitable plug-in variant for standard *Subspace Boosting*, but with the additional benefit of operating over shared composition spaces for improved interpretability. Similar results can be seen for ViT-B/16, establishing *Higher-Order Subspace Boosting* as a strong, but also interpretable model merging method.

Table 9: Higher-Order Subspace Boosting (Higher-Order SB) Performance compared against *Subspace Boosting* variants utilizing Task Arithmetic (TA), TIES, and Consensus Merging (CSM).

| Method | ViT-B/32 | | | ViT-B/16 | | |
|---|---|---|---|---|---|---|
| | 8 Tasks | 14 Tasks | 20 Tasks | 8 Tasks | 14 Tasks | 20 Tasks |
| TA + SB + LiNeS | **85.6** | **80.8** | 77.2 | **88.8** | **84.7** | 80.0 |
| TIES + SB + LiNeS | 83.8 | 79.1 | 75.9 | 87.4 | 83.3 | 79.7 |
| CSM + SB + LiNeS | 84.4 | 80.3 | **77.2** | 87.6 | 84.2 | **80.0** |
| Higher-Order SB + LiNeS | 84.5 | 79.8 | 75.6 | 88.5 | **84.7** | 79.1 |

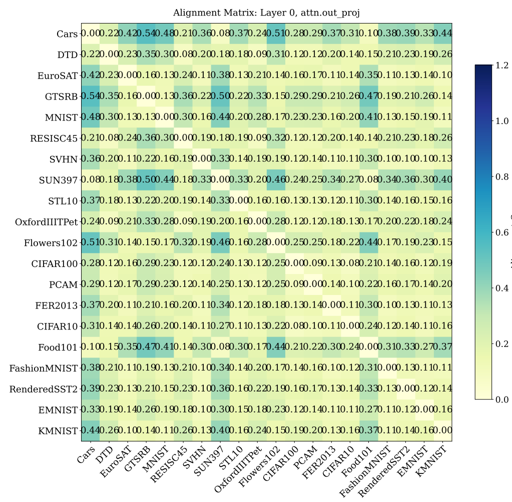
## G Additional Experiments Leveraging HO-GSVD

Leveraging HO-GSVD, we can revisit again in more detail the Alignment Matrices introduced in the main paper. Figure 10 shows the resulting Alignment Matrix for the attention block's projection layer across different layers. Subfigure 10d shows the mean Alignment Matrix across all layers and components. While for layer 0, the values are consistently low and showing high overlap between models, for the deeper layers, the alignment scores start becoming more defined. This confirms that shallow layers learn general features that are shared across multiple models (Wang et al., 2024a), hence the high overlap of important dimensions, whereas the later layers showcase more task-specific information and separation into subspaces.

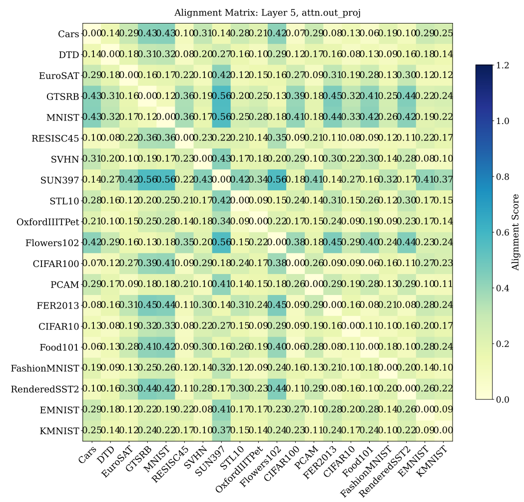
Table 10: Model selection comparison for 8 experts evaluated on all 20 tasks.

| Method | Accuracy | Normalized Accuracy |
|---|---|---|
| Random selection | 67.5 | 72.5 |
| HO-GSVD selection | **69.7** | **75.2** |

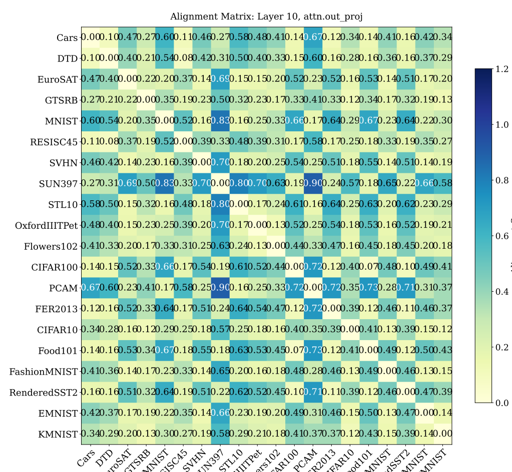
We perform an additional experiment by merging 8 models out of 20 chosen either randomly or via our optimal selection approach. Random selection is performed 20 times, and we report the average. The results are shown in Table 10. The performance of the merged model is evaluated across all 20 tasks (including those not involved in the merging process). When merging via optimal model selection, the method accurately chooses a subset of models that performs better than average. Based on both experiments, one can observe that optimal model selection outperforms average random model selection by several percent, highlighting the effectiveness of using HO-GSVD to select optimal models for merging.



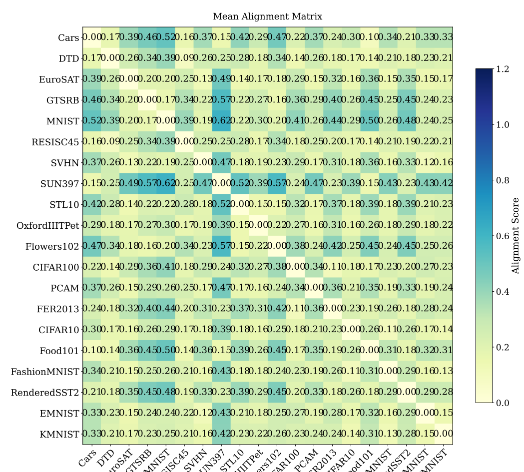

(a) Layer 0, attn_out_proj




(b) Layer 5, attn_out_proj




(c) Layer 10, attn_out_proj




(d) Mean Alignment Matrix for all components and layers.

Figure 10: Alignment matrices for attn_out_proj for different layers as well as the mean Alignment Matrix for all components.