

DC-Merge: Improving Model Merging with Directional Consistency

Han-Chen Zhang* Zi-Hao Zhou* Mao-Lin Luo Shimin Di
Min-Ling Zhang Tong Wei†

¹School of Computer Science and Engineering, Southeast University, Nanjing 210096, China

²Key Laboratory of Computer Network and Information Integration (Southeast University)

{hanchenzh, zhouzih, weit}@seu.edu.cn

Abstract

Model merging aims to integrate multiple task-adapted models into a unified model that preserves the knowledge of each task. In this paper, we identify that the key to this knowledge retention lies in maintaining the directional consistency of singular spaces between merged multi-task vector and individual task vectors. However, this consistency is frequently compromised by two issues: i) an imbalanced energy distribution within task vectors, where a small fraction of singular values dominate the total energy, leading to the neglect of semantically important but weaker components upon merging, and ii) the geometric inconsistency of task vectors in parameter space, which causes direct merging to distort their underlying directional geometry. To address these challenges, we propose DC-Merge, a method for directional-consistent model merging. It first balances the energy distribution of each task vector by smoothing its singular values, ensuring all knowledge components are adequately represented. These energy-balanced vectors are then projected onto a shared orthogonal subspace to align their directional geometries with minimal reconstruction error. Finally, the aligned vectors are aggregated in the shared orthogonal subspace and projected back to the original parameter space. Extensive experiments on vision and vision-language benchmarks show that DC-Merge consistently achieves state-of-the-art performance in both full fine-tuning and LoRA settings. The implementation code is available at <https://github.com/Tobeginwith/DC-Merge>.

1. Introduction

Pre-trained models are the foundation of modern machine learning systems [3, 4, 52, 71]. In practice, they are typically fine-tuned for specialization in specific tasks [17, 18, 28, 65]. A growing body of research has focused

on *model merging* [37], which integrates multiple task-adapted models into a unified model while preserving each task’s capability. Many methods have been proposed to improve the effectiveness of model merging by reducing sign conflicts [68], by aligning gradients [11], or through magnitude-based selection [12, 30, 45]. Despite its potential to enable efficient multi-task adaptation without retraining, existing approaches often suffer performance degradation after merging [37], especially when tasks originate from heterogeneous domains [11]. Recent studies aim to reduce the interference among different tasks [6, 20, 43], while the underlying mechanism of how task-specific capabilities are preserved after merging remains underexplored. A central question thus arises: *what property must be preserved to retain each task’s ability after merging?*

Following Task Arithmetic (TA) [29], we define a task vector as the parameter difference between a fine-tuned model and its pre-trained counterpart. Each task vector can be decomposed via singular value decomposition (SVD) into a set of orthogonal *knowledge vectors*, each representing a distinct adaptation direction weighted by its singular value. We term each of these directions as a *knowledge component* and the corresponding singular values reflect the energy distributed across these components. We observe that task performance after model merging primarily depends on the *directional consistency* between the merged and original task vectors. Specifically, as long as the directions of the knowledge components are preserved, the merged model retains most task capabilities, even if their energy distribution changes. In contrast, slight directional deviations significantly degrade performance, indicating that maintaining directional consistency of knowledge components is crucial to maintaining task performance.

To quantify this consistency, we propose *directional similarity* (DirSim), which measures the consistency of directional geometry between two task vectors while discounting the influence of energy distribution. Unlike cosine similarity, which emphasizes consistency of high-energy compo-

*Equal contribution, †Corresponding author

nents, DirSim also accounts for the directional consistency of weaker yet semantically informative components. Empirically, DirSim shows a strongly positive correlation with post-merge task-wise performance, validating it as a reliable indicator of knowledge preservation.

Despite its importance, directional consistency is often violated by two fundamental issues. First, the energy distribution of task vectors is imbalanced, where a few singular values capture most of the energy (as shown in Figure 1), causing the model to overemphasize on high-energy directions and thereby hindering generalization and directional geometry preservation. Second, directly merging task vectors in the original parameter space leads to basis misalignment: different tasks span heterogeneous low-rank subspaces whose orientations are not geometrically aligned. Consequently, the merged task vector fails to preserve the directional geometry of each task vector that characterizes the task’s knowledge. To address these challenges, we propose a new method called *DC-Merge*, which explicitly enforces directional consistency between the merged multi-task vector and each original task vector. DC-Merge consists of two complementary modules: i) *energy smoothing* redistributes the singular values of each task vector to balance the energy distribution of its knowledge components, thereby preventing the merging process from overlooking weaker but semantically rich directions within each task vector. ii) *cover space merging* then projects all smoothed task vectors into a shared orthogonal subspace before aggregation, ensuring that merging occurs under a consistent cover basis without cross-task directional interference. Together, these modules preserve the task directional geometry during merging, enabling stable multi-task compatibility and strong generalization. Extensive experiments on both full fine-tuning (FFT) and LoRA [26] setups show that DC-Merge achieves state-of-the-art results on both vision and vision-language benchmarks while maintaining high directional consistency with original task vectors.

In summary, our key contributions are as follows:

- We correlate the model merging performance with a novel concept *directional consistency* between the merged multi-task vector and individual task vectors.
- We introduce DirSim, a new metric that isolates directional consistency from energy distribution effects. DirSim shows a strong positive correlation with the performance of merged model.
- We propose *DC-Merge*, a method that enhances directional consistency by first balancing energy distribution of task vectors and then merging them within a shared orthogonal subspace.
- Extensive experiments on vision and vision-language benchmarks demonstrate that DC-Merge achieves state-of-the-art performance in both FFT and LoRA settings.

2. Directional Consistency Matters

This section reveals the intrinsic imbalance of energy distribution across knowledge components and presents empirical evidence confirming the importance of directional consistency in model merging. We also introduce a new metric to quantify this consistency. Unless otherwise specified, the experiments in this section are based on a ViT-B-32 visual encoder [15] under LoRA configuration.

2.1. Preliminary

Model Merging. Given a pre-trained parameter set \mathbf{W}_0 and a collection of fine-tuned models $\{\mathbf{W}_i\}_{i=1}^T$ obtained from distinct tasks, model merging seeks a merged parameter set $\widetilde{\mathbf{W}}$ that approximates the behavior of each \mathbf{W}_i on its corresponding task.

Task Vectors. For a FFT model of the i -th task, the task vector is defined as $\Delta\mathbf{W}_i = \mathbf{W}_i - \mathbf{W}_0$, which captures the direction and magnitude of adaptation in the weight space [29]. In the LoRA paradigm, the task-specific update is parameterized explicitly as $\Delta\mathbf{W}_i^{\text{LoRA}} = \mathbf{B}_i\mathbf{A}_i$, where $\mathbf{A}_i \in \mathbb{R}^{r \times d}$ and $\mathbf{B}_i \in \mathbb{R}^{d \times r}$. Thus, LoRA directly produces a compact and structured low-rank task vector.

Unified View and Low-Rank Merging. Although FFT and LoRA differ in parameterization, they are inherently connected under a unified low-rank formulation. Empirically, FFT updates $\Delta\mathbf{W}_i$ tend to reside in a low-dimensional subspace [26] and can be well approximated by a truncated SVD as $\Delta\mathbf{W}_i \approx \mathbf{U}_i\boldsymbol{\Sigma}_i\mathbf{V}_i^\top$. From this perspective, LoRA explicitly constrains the parameter updates to a low-rank subspace, whereas FFT implicitly exhibits a similar low-rank structure that can be revealed through singular value analysis and compression. This insight bridges the two approaches under a unified low-rank adaptation paradigm.

Building on this unified view, model merging can be viewed as an operation on these low-rank matrices. The merging process thus involves: i) extracting low-rank representations for each task vector, ii) merging these low-rank matrices to effectively integrate multi-task knowledge, and iii) constructing the final model by combining the pre-trained weights and merged multi-task vector through $\widetilde{\mathbf{W}} = \mathbf{W}_0 + \Delta\widetilde{\mathbf{W}}$.

2.2. Balanced Energy Enhances Generalization

A New Perspective on Task Vectors. We first provide a new interpretation of task vectors from the viewpoint of their intrinsic low-rank structure. Given a task vector $\Delta\mathbf{W}$ with rank $r \ll d$, we can decompose it using SVD:

$$\Delta\mathbf{W} = \sum_{i=1}^r \sigma^i \mathbf{u}^i \mathbf{v}^{i\top}, \quad (1)$$

where $\{\mathbf{u}^i\}_{i=1}^r$ and $\{\mathbf{v}^i\}_{i=1}^r$ are the left and right singular vectors, respectively, and $\{\sigma_i\}_{i=1}^r$ are the correspond-

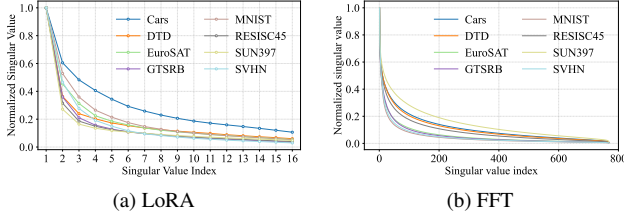


Figure 1. The singular value distribution of task vectors averaged across all layers. We normalize each singular value by the largest one within each dataset to eliminate the magnitude discrepancy among different datasets.

ing singular values. From this decomposition, each term $\sigma^i \mathbf{u}^i \mathbf{v}^{i\top}$ can be regarded as a specific *knowledge vector* for task adaptation, and the adaptation direction $\mathbf{u}^i \mathbf{v}^{i\top}$ is defined as a *knowledge component*. The singular value σ^i quantifies the degree to which the corresponding knowledge component is utilized, thus the distribution of $\boldsymbol{\sigma} = (\sigma^1, \dots, \sigma^r)$ can be interpreted as the *energy distribution* across knowledge components.

Observation: Singular Values Follow a Long-Tailed Distribution. An important empirical observation arises when analyzing the low-rank task vectors obtained from practical training. As shown in Figure 1, the decomposed knowledge vectors $\{\sigma_i \mathbf{u}_i \mathbf{v}_i^\top\}_{i=1}^r$ usually follow a long-tailed energy distribution, where a small fraction of knowledge components dominate the total energy, indicating that the knowledge captured by the task vector is inherently imbalanced. This intrinsic imbalance leads to a potential drawback in model behavior: as a small number of knowledge components dominate the task adaptation, the model tends to overfit to specific patterns while neglecting weaker but semantically important components.

To further illustrate this phenomenon, Figure 2a presents the transfer performance in eight tasks, contrasting the results before and after balancing the internal knowledge of each task vector. The *diagonal elements* of each heatmap represent the degree to which the original task’s capability is preserved, while the *off-diagonal elements* measure the zero-shot transferability to other tasks, reflecting how well the knowledge generalizes beyond its training domain. As shown in Figure 2a (left), directly using the original low-rank task vector obtained from fine-tuning causes severe performance degradation on unrelated tasks, with many off-diagonal entries significantly suppressed, implying that over-concentrated energy distribution harms cross-task generalization. In contrast, Figure 2a (right) displays the results of *energy-balanced task vectors* constructed by a simple averaging strategy:

$$\Delta \bar{\mathbf{W}} = \left(\frac{\sum_{i=1}^r \sigma^i}{r} \right) \left(\sum_{i=1}^r \mathbf{u}^i \mathbf{v}^{i\top} \right). \quad (2)$$

The diagonal elements are quite close to 1.0, indicating that

the vast majority of task capability is preserved, and the off-diagonal elements increase notably, suggesting improved zero-shot transfer and multi-task compatibility.

Revisiting Task Vector Similarity from a Knowledge Decomposition Perspective. To further analyze the underlying reason behind better generalization capabilities after balancing the energy distribution, we revisit the cosine similarity between task vectors [29] from the perspective of knowledge decomposition. We argue that this metric can be interpreted as the *expressive capacity* of one task vector to represent another, i.e., how well the knowledge of task t can be linearly reconstructed by task s .

Proposition 1 Given the knowledge vector decompositions of two task vectors $\Delta \mathbf{W}_s$ and $\Delta \mathbf{W}_t$, their cosine similarity can be equivalently expressed as

$$\begin{aligned} \text{CosSim}(\Delta \mathbf{W}_s, \Delta \mathbf{W}_t) &= \frac{\langle \Delta \mathbf{W}_s, \Delta \mathbf{W}_t \rangle}{\|\Delta \mathbf{W}_s\|_F \|\Delta \mathbf{W}_t\|_F} \\ &= \frac{\boldsymbol{\sigma}_s \mathbf{R}(s, t) (\boldsymbol{\sigma}_t)^\top}{\|\boldsymbol{\sigma}_s\|_2 \|\boldsymbol{\sigma}_t\|_2}, \end{aligned} \quad (3)$$

where $\mathbf{R}(s, t) \in \mathbb{R}^{n \times m}$ is defined entry-wise as:

$$\mathbf{R}_{i,j}(s, t) = (\mathbf{u}_s^i)^\top \mathbf{u}_t^j (\mathbf{v}_t^j)^\top \mathbf{v}_s^i. \quad (4)$$

Remark. The matrix $\mathbf{R}(s, t)$ measures the directional consistency between the two knowledge bases. Each entry $\mathbf{R}_{i,j}(s, t) = (\mathbf{u}_s^i)^\top \mathbf{u}_t^j (\mathbf{v}_t^j)^\top \mathbf{v}_s^i$ quantifies how the j -th knowledge component of task t can be projected onto the i -th knowledge component of task s . Therefore, $\mathbf{R}(s, t)$ can be interpreted as a projection operator that expresses the knowledge geometry of task t in the basis of task s . From this perspective, the overall cosine similarity $\text{CosSim}(\Delta \mathbf{W}_s, \Delta \mathbf{W}_t) = \frac{\boldsymbol{\sigma}_s \mathbf{R}(s, t) (\boldsymbol{\sigma}_t)^\top}{\|\boldsymbol{\sigma}_s\|_2 \|\boldsymbol{\sigma}_t\|_2}$ can be viewed as the weighted aggregation of these projections, reflecting how effectively task s can represent or reconstruct the knowledge of task t . When the rank m of $\Delta \mathbf{W}_t$ is higher than the rank n of $\Delta \mathbf{W}_s$, this interpretation is particularly intuitive: $\Delta \mathbf{W}_s$ spans a lower-dimensional subspace that attempts to encode the richer knowledge geometry of task t . Thus, a higher CosSim indicates stronger expressiveness of task s with respect to task t . We leave further analysis of this perspective in Appendix C.

Energy-Balanced Knowledge Components Enhance Multi-Task Performance. Building upon the cosine similarity analysis above, we can now explain why energy-balanced task vectors achieve stronger multi-task capability. When singular values $\boldsymbol{\sigma}$ are highly skewed, the corresponding task vectors collapse onto a few dominant knowledge components, limiting the span of the subspace and thereby reducing its expressive coverage over other tasks. In contrast, balancing the energy distribution across knowledge components prevents representational collapse and enlarges

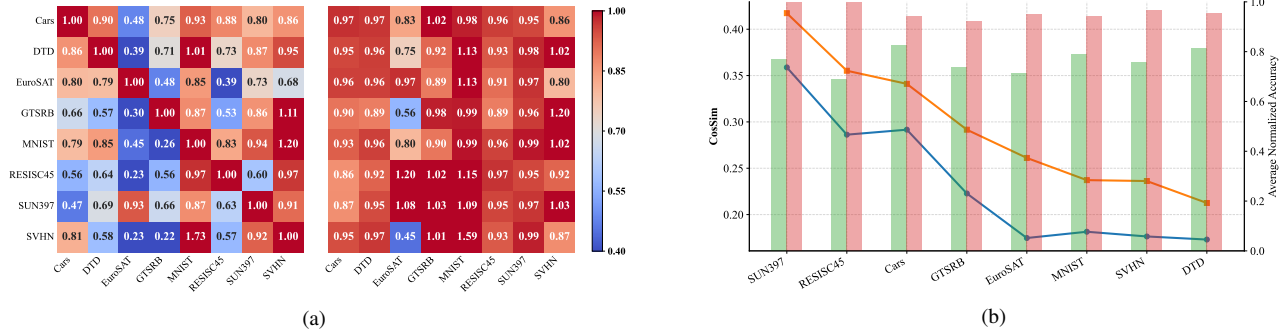


Figure 2. **(a)** Comparison of cross-task transferability before (left) and after (right) balancing the energy distribution. The diagonal elements represent the relative performance of each task with respect to its fine-tuned model, while the off-diagonal elements indicate the relative transfer performance on other tasks, normalized against their zero-shot baseline. **(b)** Each task vector’s cosine similarity with multi-task vector (lines) and average normalized transfer accuracy of each task vector (bars). We compare original task vectors and their energy-balanced counterparts. The energy-balanced task vectors achieve higher CosSim with the multi-task vector (we use $\Delta_{\text{Iso-c}}$ [46] for simplicity) and better cross-task generalization, indicating that balancing the energy distribution across knowledge components enhances multi-task expressiveness.

the subspace, thereby enhancing its ability to encode multiple tasks.

To verify this, we compute the cosine similarity between the single-task vectors and multi-task vector, and compare it with that of the energy-balanced task vectors obtained by Eq. (2). As shown in Figure 2b, the balanced vectors consistently exhibit higher cosine similarity with the multi-task model vector. Empirically, this aligns with higher average normalized transfer accuracy. These results suggest that energy smoothing improves the task vector’s ability to represent multi-task knowledge.

2.3. Measuring Directional Consistency

Directional Knowledge Similarity. We now delve into the underlying mechanism of task knowledge preservation. The cosine similarity can be factorized into two components: i) matrix $\mathbf{R}(s, t)$ quantifies the directional consistency between the two knowledge components, and ii) singular values σ_s, σ_t encode the importance of these knowledge components. While both terms contribute to the overall similarity, we argue that the preservation of task ability primarily depends on the *directional consistency* rather than on the energy distribution. Empirically, this observation is supported by the fact that energy-balanced task vectors can largely maintain the performance of the original task vector, which implies that as long as the relative directions of knowledge vectors are preserved, the model can retain most of its learned behavior.

To isolate this directional factor and quantify it explicitly, we propose a new similarity metric that removes the influence of energy distribution by uniformizing it with $\bar{\sigma}_s = \frac{1_n}{\sqrt{n}}, \bar{\sigma}_t = \frac{1_m}{\sqrt{m}}$. Substituting back to the cosine simi-

ilarity leads to a purely directional consistency measure:

$$\begin{aligned} \text{DirSim}(\Delta\mathbf{W}_s, \Delta\mathbf{W}_t) &\triangleq \bar{\sigma}_s \mathbf{R}(s, t) (\bar{\sigma}_t)^\top \\ &= \frac{1}{\sqrt{nm}} \mathbf{1}_n^\top \mathbf{R}(s, t) \mathbf{1}_m. \end{aligned} \quad (5)$$

DirSim equally considers the similarity between every pair of directions, whereas CosSim is subject to the similarity among the dominant directions (Figure 3a). A higher value DirSim implies that the two task vectors share more directionally consistent knowledge components and that one can better represent the other within its knowledge basis.

To verify the above claim, we conduct controlled perturbation experiments and analyze how the retained task performance correlates with both DirSim and CosSim. As illustrated in Figure 3b, performance decreases as DirSim declines, indicating that directional inconsistency leads to substantial performance loss. In contrast, when only energy is redistributed while the directions remain aligned, performance remains largely stable despite notable changes in CosSim. These results provide strong empirical evidence that preserving knowledge directions is the key to maintaining task ability, while variation of energy distribution has a relatively minor effect. We leave implementation details in Appendix E.3.

Directional Similarity for Post-merge Task-wise Performance. Building upon the previous analysis, we argue that preserving the directions of knowledge components is the key factor for retaining each task’s performance during model merging. To quantify this, one might consider computing the directional similarity (DirSim) between each task vector $\Delta\mathbf{W}_i$ and the merged task vector $\Delta\tilde{\mathbf{W}}$. However, $\Delta\tilde{\mathbf{W}}$ aggregates multiple task vectors, introducing directional redundancy that artificially deflates DirSim.

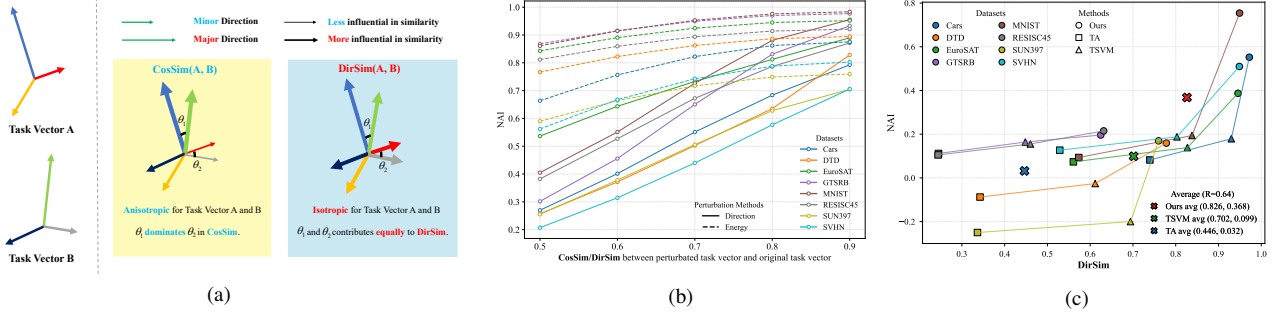


Figure 3. (a) Comparison of CosSim and DirSim. DirSim considers the similarity between every pair of directions equally, whereas CosSim mainly focuses on the similarity among the dominant directions while ignoring the minor ones. (b) Empirical validation of the importance of preserving directional geometry. Solid line: task performance vs. DirSim under random directional perturbations; dashed line: task performance vs. CosSim under energy distribution perturbations. (c) Correlation of task-wise performance with projected DirSim. The overall correlation is positive (Pearson $R = 0.64$) and per-method averages follow the same trend. Similar patterns persist under larger task scales as illustrated in Figure 7. We utilize Normalized Accuracy Improvement (NAI) [46] to measure task-wise performance in (b) and (c).

To address this, we consider the task-specific activation of the merged vector by projecting it onto the low-rank feature subspace. Empirical evidence shows that features from a given task i collapse into a low-rank subspace U_i [13, 19]. We thus project the merged multi-task vector onto the low-rank feature subspace to acquire the task-activated part:

$$\Delta \widetilde{W}_i = U_i U_i^\top \Delta \widetilde{W}. \quad (6)$$

We then compute $\text{DirSim}(\Delta W_i, \Delta \widetilde{W}_i)$ as our metric for task-wise capability retention.

In practice, the exact low-rank feature subspace U_i is inaccessible and we approximate it using the subspace spanned by the left singular vectors of ΔW_i . As shown in Figure 3c, the projected DirSim computed in this manner exhibits a clear monotonic relationship with the normalized accuracy improvement of merged models across different datasets and methods. Points with higher projected DirSim correspond to better task knowledge retention. This confirms that the proposed projected DirSim can measure how well task-specific knowledge is preserved during model merging.

3. The Proposed DC-Merge Approach

The goal of our method is to preserve the complete directional geometry of task vectors during model merging. To this end, we propose two modules, i.e., *energy smoothing* and *cover space merging*.

Energy Smoothing for Balanced Knowledge Representations. As discussed in Section 2.2, the imbalanced energy distribution of knowledge components biases the merging process, potentially ignoring the direction of weaker but semantically rich knowledge components. To mitigate these issues, we balance the energy distribution of each task vector via energy smoothing before merging. For each

Algorithm 1 DC-Merge

- 1: **Input:** Task vectors $\{\Delta_i\}_{i=1}^T$ with $\Delta_i \in \mathbb{R}^{m \times n}$
- 2: **Output:** Merged multi-task vector $\widetilde{\Delta}$
- 3: **Step 1:** Construct cover space for all task vectors
- 4: **for** $i = 1 \rightarrow T$ **do**
- 5: Compute r -rank SVD: $\Delta_i \approx U_i^{(r)} \Sigma_i^{(r)} V_i^{(r)\top}$
- 6: Smoothing $\Sigma_i^{(r)}$ by $\overline{\Sigma}_i^{(r)} = \text{diag}([\overline{\sigma}_i^1; \overline{\sigma}_i^2; \dots; \overline{\sigma}_i^r])$
- 7: Reconstruct $\overline{\Delta}_i = U_i^{(r)} \overline{\Sigma}_i^{(r)} V_i^{(r)\top}$
- 8: **end for**
- 9: Obtain concatenated basis U, V by Eq. (9)
- 10: Whitening U and V respectively to obtain \widetilde{U} and \widetilde{V}
- 11: **Step 2:** Project $\overline{\Delta}_i$ onto cover space and merge
- 12: Project each $\overline{\Delta}_i$ and obtain M_i by Eq. (10)
- 13: $\widetilde{M} \leftarrow \text{Merging}(\{M_i\}_{i=1}^T)$ via TA or TIES
- 14: **Step 3:** Project \widetilde{M} back to parameter space
- 15: Construct mask $\mathcal{M} \leftarrow \text{block-diag}(\mathbf{1}_{r \times r}, \dots, \mathbf{1}_{r \times r})$
- 16: Obtain merged task vector $\widetilde{\Delta}$ by Eq. (11)

task vector Δ_i , we consider its knowledge decomposition $\Delta_i = \sum_{j=1}^r \sigma_i^j \mathbf{u}_i^j \mathbf{v}_i^{j\top}$, where $\sigma_i = (\sigma_i^1, \dots, \sigma_i^r)$ contains the singular values sorted in descending order. Instead of directly using the original energy distribution σ_i , we replace it with a smoothed version $\overline{\sigma}_i$ to redistribute the energy more evenly across the top- r components, alleviating dominance on a few knowledge components.

For example, we consider the simple yet effective form of smoothing by replacing all top- r singular values with their mean: $\overline{\sigma}_i = \left(\frac{1}{r} \sum_{j=1}^r \sigma_i^j\right) \mathbf{1}_r$, which equalizes the contribution of all retained knowledge components. For completeness, additional smoothing strategies are discussed in Appendix E.4. We then perform merging on these energy-balanced task vectors rather than the original ones.

Projection and Merging in the Cover Space. Directly merging task vectors in the original parameter space may distort directional geometry due to misaligned subspaces $\{(U_i, V_i)\}$. To preserve directional consistency across all tasks, we seek a pair of shared orthonormal basis (\tilde{U}, \tilde{V}) that define a *cover space* capturing the directional geometry of all task vectors, and perform merging within the cover space. The objective can be formulated as:

$$\min_{\tilde{U}, \tilde{V}} \sum_{i=1}^T \sum_{j=1}^r \min_{\sigma_i^j \in \mathbb{R}^k} \left\| \mathbf{u}_i^j \mathbf{v}_i^{j\top} - \tilde{U} \text{diag}(\sigma_i^j) \tilde{V}^\top \right\|_F^2, \quad (7)$$

s.t. $\tilde{U}^\top \tilde{U} = \tilde{V}^\top \tilde{V} = \mathbf{I}$.

where $k = rT$. The inner minimization over σ_i^j determines the optimal coefficients along the cover basis. Using Proposition 2(a), we obtain the surrogate objective:

$$\max_{\tilde{U}, \tilde{V}} \sum_{i=1}^T \sum_{j=1}^r \left\| \sigma(\tilde{U}, \tilde{V}, \mathbf{u}_i^j \mathbf{v}_i^{j\top}) \right\|_2^2, \quad (8)$$

s.t. $\tilde{U}^\top \tilde{U} = \tilde{V}^\top \tilde{V} = \mathbf{I}$,

where $\sigma(\mathbf{U}, \mathbf{V}, \Delta) = \text{diag}(\mathbf{U}^\top \Delta \mathbf{V}) \in \mathbb{R}^k$ denotes projection onto the shared dyadic directions.

As directly optimizing Eq. (8) incurs non-trivial computational overhead, we adopt whitening [54] here as it serves as a near-optimal solution to Eq. (8) while being computationally efficient. In Appendix D, we provide an iterative approach for constructing cover basis and theoretically show its relation to the whitening transformation. Specifically, we construct the cover basis (\tilde{U}, \tilde{V}) by whitening the column-wise concatenated per-task knowledge basis:

$$\mathbf{U} = [\mathbf{U}_1^{(r)}, \dots, \mathbf{U}_T^{(r)}], \quad \mathbf{V} = [\mathbf{V}_1^{(r)}, \dots, \mathbf{V}_T^{(r)}]. \quad (9)$$

Thus, $\tilde{U}^\top \tilde{U} = \tilde{V}^\top \tilde{V} = \mathbf{I}$ defines an orthogonal basis that contains the union of all tasks' directional geometry. Each smoothed task vector is then projected onto cover space by:

$$\mathbf{M}_i = \tilde{U}^\top \Delta_i \tilde{V}. \quad (10)$$

This projection ensures that all task vectors are expressed under shared cover basis, which facilitates directionally consistent task vectors aggregation via existing element-wise merging methods, such as TA [29] and TIES-Merging [68], to obtain $\tilde{\mathbf{M}}$. Finally, the merged multi-task vector is reconstructed by projecting $\tilde{\mathbf{M}}$ back to the original parameter space:

$$\tilde{\Delta} = \tilde{U} (\tilde{\mathbf{M}} \odot \mathcal{M}) \tilde{V}^\top, \quad (11)$$

where \mathcal{M} serves as a structural mask. We leave further discussion of the structural mask in Appendix D and summarize the key steps of our approach in Algorithm 1.

4. Experiments

In this section, we evaluate the performance of DC-Merge against existing baselines through extensive experiments using vision models and vision-language models (VLMs) in both FFT and LoRA settings, demonstrating the versatility of DC-Merge. We further perform comprehensive ablation studies to analyze the effectiveness of each key component in DC-Merge.

4.1. Results for Vision Tasks

In this subsection, we investigate the merging of vision models. For fully fine-tuned vision models, following prior works [20, 46], we use 8-task, 14-task, and 20-task benchmarks for evaluation, respectively, and employ three CLIP [52] variants: ViT-B-32, ViT-B-16, and ViT-L-14 as visual encoders [15]. For LoRA fine-tuned vision models, we extend previous evaluations by assessing both our method and existing baselines on a larger number of tasks, specifically 8, 12 and 16. Consistent with prior work [6, 20, 46, 58], we report the average absolute and normalized accuracy of merged models.

In the full parameter fine-tuning setting, we compare our method against Weight Averaging [64], Task Arithmetic [29], TIES-Merging [68], Consensus TA [62], TSV-M [20] and Iso-CTS [46]. For LoRA fine-tuned models, Task Arithmetic, KnOTS-TIES [58], WUDI-Merging [6], TSV-M, and Iso-CTS serve as baselines. We provide details on benchmarks and experimental setups in the Appendix E.

Table 1 shows the results of merging vision models fine-tuned by LoRA. Across three different backbones, the performance of our method consistently surpasses the current state-of-the-art methods. Moreover, the performance gains remain substantial with the growth of tasks. We also conduct experiments on the checkpoints provided by KnOTS [58], where our method still achieves superior performance compared to existing state-of-the-art methods. The corresponding results are reported in Appendix F.1. Table 2 presents the results under the full fine-tuning setting. The results demonstrate that our approach not only exhibits strong capability when merging LoRA fine-tuned models but also achieves state-of-the-art performance under the FFT scenario. Notably, the superiority of our method becomes more significant as the number of tasks increases.

4.2. Results for Vision-Language Tasks

In the multi-modal model merging setting, we compare our method with Task Arithmetic, TIES-Merging, DARE [69], PCB-Merging [16], and RobustMerge [70] on eight multi-modal datasets using LLaVA-v1.5-7B [38] as backbone. Following the experimental setup of RobustMerge, we further evaluate the merged model on four additional datasets to assess its generalization ability to unseen tasks. We adopt

Method	ViT-B-32			ViT-B-16			ViT-L-14		
	8 tasks	12 tasks	16 tasks	8 tasks	12 tasks	16 tasks	8 tasks	12 tasks	16 tasks
Individual	87.82	88.90	87.50	89.71	90.76	89.11	92.36	93.57	92.11
Task Arithmetic	52.80 _(61.73)	60.76 _(69.12)	60.04 _(68.94)	57.70 _(65.30)	64.26 _(71.12)	62.40 _(70.16)	68.29 _(74.38)	73.69 _(78.84)	69.98 _(75.58)
KnOTS-TIES	55.93 _(65.03)	63.03 _(71.43)	61.78 _(70.78)	60.80 _(68.59)	66.35 _(73.38)	64.31 _(72.29)	73.61 _(79.92)	75.64 _(80.90)	72.19 _(77.98)
WUDI-Merging	55.25 _(64.38)	62.20 _(70.64)	61.24 _(70.27)	58.95 _(66.63)	65.29 _(72.29)	64.59 _(72.51)	69.78 _(75.91)	74.25 _(79.45)	71.79 _(77.59)
TSV-M	58.91 _(68.25)	65.30 _(73.86)	63.51 _(72.72)	62.97 _(70.87)	68.92 _(76.06)	67.21 _(75.35)	76.52 _(83.00)	79.67 _(85.13)	74.37 _(80.31)
Iso-CTS	63.01 _(72.71)	66.28 _(75.01)	64.61 _(74.02)	69.06 _(77.38)	71.52 _(78.87)	69.88 _(78.22)	81.64 _(88.31)	81.35 _(86.87)	77.50 _(83.65)
DC-Merge	64.17 _(73.90)	68.40 _(77.22)	66.27 _(75.80)	70.53 _(78.86)	73.12 _(80.56)	70.57 _(78.91)	82.61 _(89.42)	83.62 _(89.31)	79.53 _(85.71)

Table 1. Average absolute accuracy results on vision model merging benchmarks in LoRA setting; subscript (in parentheses) is the average normalized accuracy. The best results are in **bold** and the second-best are underlined.

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
Individual	92.83	90.88	91.37	94.64	92.76	93.17	95.81	94.29	94.73
Weight Averaging	66.34 _(72.13)	64.34 _(71.12)	61.04 _(67.53)	72.22 _(76.60)	69.46 _(74.82)	65.31 _(70.36)	79.56 _(83.15)	76.73 _(81.10)	71.60 _(75.60)
Task Arithmetic	70.79 _(76.55)	65.32 _(72.09)	60.52 _(66.79)	75.41 _(79.58)	70.52 _(75.89)	65.78 _(70.76)	84.93 _(88.65)	79.41 _(83.95)	74.01 _(78.07)
TIES-Merging	75.09 _(81.08)	68.02 _(74.83)	63.38 _(69.90)	79.74 _(84.34)	73.22 _(78.73)	68.18 _(73.26)	86.88 _(90.69)	79.46 _(84.05)	75.71 _(79.80)
Consensus TA	75.03 _(80.84)	70.39 _(77.36)	65.43 _(71.98)	79.39 _(83.86)	74.39 _(79.92)	69.76 _(74.93)	86.34 _(90.08)	82.22 _(86.94)	79.00 _(83.22)
TSV-M	85.86 _(92.31)	80.06 _(87.88)	77.07 _(84.29)	89.01 _(93.94)	84.58 _(91.01)	80.57 _(86.45)	92.98 _(96.98)	89.17 _(94.43)	87.72 _(92.50)
Iso-CTS	86.20 _(91.78)	81.71 _(89.70)	78.05 _(85.48)	90.91 _(95.95)	86.40 _(92.81)	82.38 _(88.36)	94.69 _(98.81)	90.98 _(96.28)	90.05 _(94.88)
DC-Merge	87.05 _(93.55)	82.52 _(90.62)	80.58 _(88.18)	90.78 _(95.83)	87.06 _(93.70)	84.57 _(90.76)	94.31 _(98.38)	91.01 _(96.43)	90.51 _(95.43)

Table 2. Average absolute accuracy results on vision model merging benchmarks in FFT setting; subscript (in parentheses) is the average normalized accuracy. The best results are in **bold** and the second-best are underlined.

the checkpoints released by RobustMerge and provide detailed experimental configurations in Appendix E.

As shown in Table 3, our method notably outperforms existing state-of-the-art methods on both seen and unseen tasks, demonstrating that its applicability is not limited to vision models but can also scale to large multi-modal models. We also evaluate the generalization capability to unseen tasks of our method on vision models and the detailed results are presented in Appendix F.3.

Method	Seen Tasks	Unseen Tasks
Zeroshot	43.37	25.22
Individual	69.23	–
Multi Task	63.62	36.06
Task Arithmetic	53.93	33.31
DARE-Merging	53.84	33.15
TIES-Merging	53.09	33.14
PCB-Merging	53.70	33.53
RobustMerge	<u>57.33</u>	<u>37.99</u>
DC-Merge	59.63	39.84

Table 3. Performance on MM-MergeBench [70], containing eight seen tasks (LoRA fine-tuned) and four unseen tasks. The best results are in **bold** and the second-best are underlined. We report average absolute accuracy. See Appendix F.7 for detailed results.

4.3. Ablations and Analysis

Unless otherwise specified, all experiments in this subsection are conducted in LoRA setting.

The Effectiveness of Energy Smoothing. We investigate the impact of our energy smoothing strategy on the perfor-

mance of our method, with the results summarized in Table 4. The results align well with our observations: applying energy smoothing to each task vector effectively ensures that all the knowledge components can be adequately expressed, leading to a significant improvement in overall performance. Notably, preserving a moderate degree of skewness in the energy distribution (i.e., linear smoothing) can yield better results than averaging. We provide additional comparisons of smoothing strategy on ViT-B-16 and ViT-L-14 in Appendix E.4.

Method	8 tasks	12 tasks	16 tasks
No smoothing	69.27	74.60	74.47
Averaging	73.09 (+3.82)	76.42 (+1.82)	75.51 (+1.04)
Linear smoothing	73.90 (+4.63)	77.22 (+2.62)	75.80 (+1.33)

Table 4. Performance comparison of different smoothing strategies. We report average normalized accuracy using ViT-B-32.

Impact of Performing a Post-hoc Pruning. In Algorithm 1, we perform a post-hoc pruning by applying a mask \mathcal{M} to mitigate directional inconsistency of different tasks before projecting the merged parameter matrix \bar{M} back to the original parameter space. Table 5 presents the effect of such structural pruning on overall performance in both LoRA and FFT settings. The performance degradation becomes more pronounced with the increase of tasks. Moreover, since the number of fine-tuned parameters in the FFT setting is substantially larger than that in LoRA, incorporating masks leads to significant performance gains of up to 10.55% in average normalized accuracy, highlighting the crucial role

that structural pruning plays in preventing cross-task directional inconsistency. We investigate the impact of mask size on the performance in Appendix D.

Method	Tasks	w/o mask	w/ mask
FFT	8 tasks	87.98	93.55 (+5.57)
	14 tasks	82.39	90.50 (+8.11)
	20 tasks	77.63	88.18 (+10.55)
LoRA	8 tasks	73.61	73.90 (+0.29)
	12 tasks	75.94	77.22 (+1.28)
	16 tasks	74.42	75.80 (+1.38)

Table 5. Comparison of performance with and w/o applying masks. We report average normalized accuracy using ViT-B-32.

Impact of Merging in the Shared Cover space. To maintain the directional geometry of each task vector, we project the smoothed task vectors onto a shared subspace prior to model merging. As shown in Table 6, compared to merging in the original parameter space, CSM significantly boosts the performance of both TA [29] and TIES [68]. Moreover, after applying energy smoothing to the task vectors, the performance is further enhanced, indicating that the two main components of our proposed method are complementary. An illustrative example provided in Appendix D further demonstrates the importance of shared cover basis in preserving the directional structure of task vectors.

Method	8 tasks	12 tasks	16 tasks
Vanilla TA	61.73	69.12	68.94
TA + ES	69.12 (+7.39)	74.35 (+5.23)	73.01 (+4.07)
TA + CSM	68.13 (+6.40)	73.92 (+4.80)	72.64 (+3.70)
TA + CSM + ES	73.82 (+12.09)	77.16 (+8.04)	75.73 (+6.79)
Vanilla TIES	62.09	69.30	70.06
TIES + ES	69.94 (+7.85)	74.97 (+5.67)	74.74 (+4.68)
TIES + CSM	69.27 (+7.18)	74.60 (+5.30)	74.47 (+4.41)
TIES + CSM + ES	73.90 (+11.81)	77.22 (+7.92)	75.80 (+5.74)

Table 6. Performance of individually applying energy smoothing (ES) and cover space merging (CSM) as well as combining them to TA or TIES compared with vanilla settings. We report the average normalized accuracy using ViT-B-32.

5. Related Work

Model merging has emerged as a promising approach to integrate expert models fine-tuned on different downstream tasks into a single multi-task model. Task Arithmetic (TA) [29] first introduces the concept of a *task vector*, defined as the difference between an expert and its pre-trained model, and combines them through scaled averaging to construct a merged model. Subsequent studies propose meticulously crafted parameter-wise strategies to mitigate interference during merging. TIES [68] reduces sign conflicts by adopting the majority sign across all models. Consensus Merging [62] applies binary masks to exclude parameters

important to fewer than two tasks. Recent studies WUDI-Merging [6] and FDA [55] optimizes the merged task vector to keep the output of merged model align with each fine-tuned model given the same input of corresponding task.

These merging methods are data-free, producing merged task vectors that can be directly integrated into the pre-trained model. A number of recent approaches, however, focus on creating model with multi-task capabilities by modifying the inference stage. Twin-Merging [42] composes task-specific components at test time, requiring two forward passes. EMR-Merging [27] employs additional per-task masks and rescalers for inference. In this paper, we restrict our study to merging methods which are data-free and leave the inference stage unaffected.

SVD-based Model Merging. Recent data-free model merging methods have incorporated SVD to improve performance [58, 63]. State-of-the-art approaches include TSV-M [20], which enforces orthogonality between task-specific subspaces to reduce task interference, and IsoCTS [46], which standardizes singular values after combining a common subspace constructed by TA [29] and task-specific subspaces. More recently, ESM [36] projects parameter updates into an activation-aware essential subspace and applies polarized scaling to amplify critical weights. In contrast to these methods, our approach prioritizes the directional consistency of each original task vector with the merged vector. We achieve this by balancing the energy distribution of knowledge components and performing the merge process within a shared orthogonal subspace induced by a pair of cover basis.

6. Conclusion and Limitation

Conclusion. In this work, we are the first to identify that preserving the directional consistency of task vectors after merging is crucial for retaining the capabilities of individual tasks. Building upon this insight, we propose DC-Merge, which maintains the directional consistency between the merged multi-task vector and each original task vector by *energy smoothing* and *cover space merging*. Our method achieves state-of-the-art performance in both FFT and LoRA settings.

Limitation. There still exists a noticeable performance gap between merging LoRA fine-tuned models and full parameter fine-tuned models. This phenomenon may arise from the number of knowledge components in each task vector. A larger number of knowledge components provides redundancy that is robust to direction shift, whereas a smaller set makes the task vector more fragile to directional inconsistency. The knowledge components of each LoRA task vector are scarce, even fewer than the LoRA rank due to its long-tailed energy distribution. A potential remedy lies in promoting a balanced energy distribution of the knowledge components during fine-tuning.

References

- [1] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E. Hinton. Layer normalization. *arXiv preprint arXiv: 1607.06450*, 2016. 18
- [2] Lukas Bossard, Matthieu Guillaumin, and Luc Van Gool. Food-101 – Mining Discriminative Components with Random Forests. In *ECCV*, 2014. 16
- [3] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *ECCV*, 2020. 1
- [4] Mathilde Caron, Hugo Touvron, Ishan Misra, Herve Jegou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. *ICCV*, 2021. 1
- [5] Gong Cheng, Junwei Han, and Xiaoqiang Lu. Remote sensing image scene classification: Benchmark and state of the art. *Proceedings of the IEEE*, 2017. 16
- [6] Runxi Cheng, Feng Xiong, Yongxian Wei, Wanyun Zhu, and Chun Yuan. Whoever started the interference should end it: Guiding data-free model merging via task vectors. In *ICML*, 2025. 1, 6, 8, 22
- [7] Mircea Cimpoi, Subhansu Maji, Iasonas Kokkinos, Sammy Mohamed, and Andrea Vedaldi. Describing textures in the wild. In *CVPR*, 2014. 16
- [8] Tarin Clanuwat, Mikel Bober-Irizar, Asanobu Kitamoto, Alex Lamb, Kazuaki Yamamoto, and David Ha. Deep Learning for Classical Japanese Literature. *arXiv preprint arXiv: 1607.06450*, 2018. 16
- [9] Adam Coates, Andrew Ng, and Honglak Lee. An Analysis of Single-Layer Networks in Unsupervised Feature Learning. In *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*. JMLR Workshop and Conference Proceedings, 2011. 16
- [10] Gregory Cohen, Saeed Afshar, Jonathan Tapson, and André van Schaik. EMNIST: Extending MNIST to handwritten letters. In *IJCNN*, 2017. 16
- [11] Nico Daheim, Thomas Möllenhoff, Edoardo M. Ponti, Iryna Gurevych, and Mohammad Emtiyaz Khan. Model merging by uncertainty-based gradient matching. In *ICLR*, 2024. 1
- [12] Mohammad-Javad Davari and Eugene Belilovsky. Model breadcrumbs: Scaling multi-task model merging with sparse masks. In *ECCV*, 2024. 1
- [13] Grégoire Delétang, Anian Ruoss, Paul-Ambroise Duquenne, Elliot Catt, Tim Genewein, Christopher Mattern, Jordi Grau-Moya, Li Kevin Wenliang, Matthew Aitchison, Laurent Orseau, Marcus Hutter, and Joel Veness. Language modeling is compression. *arXiv preprint arXiv:2309.10668*, 2024. 5
- [14] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVPR*, 2009. 17
- [15] 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 *ICLR*, 2021. 2, 6
- [16] Guodong Du, Junlin Lee, Jing Li, Runhua Jiang, Yifei Guo, Shuyang Yu, Hanting Liu, Sim K Goh, Ho-Kin Tang, Daojing He, et al. Parameter competition balancing for model merging. In *NeurIPS*, 2024. 6
- [17] Ziqing Fan, Ruipeng Zhang, Jiangchao Yao, Bo Han, Ya Zhang, and Yanfeng Wang. Federated learning with bilateral curation for partially class-disjoint data. In *Proceedings of the 37th International Conference on Neural Information Processing Systems*, pages 32006–32019, 2023. 1
- [18] Ziqing Fan, Shengchao Hu, Jiangchao Yao, Gang Niu, Ya Zhang, Masashi Sugiyama, and Yanfeng Wang. Locally estimated global perturbations are better than local perturbations for federated sharpness-aware minimization. In *Proceedings of the 41st International Conference on Machine Learning*, pages 12858–12881, 2024. 1
- [19] Giorgio Franceschelli, Claudia Cevenini, and Mirco Musolesi. Training foundation models as data compression: On information, model weights and copyright law. *arXiv preprint arXiv:2407.13493*, 2025. 5
- [20] 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. 1, 6, 8, 18, 22
- [21] Ian J. Goodfellow, Dumitru Erhan, Pierre Luc Carrier, Aaron Courville, Mehdi Mirza, Ben Hamner, Will Cukierski, Yichuan Tang, David Thaler, Dong-Hyun Lee, Yingbo Zhou, Chetan Ramaiah, Fangxiang Feng, Ruifan Li, Xiaojie Wang, Dimitris Athanasakis, John Shawe-Taylor, Maxim Milakov, John Park, Radu Ionescu, Marius Popescu, Cristian Grozea, James Bergstra, Jingjing Xie, Lukasz Romaszko, Bing Xu, Zhang Chuang, and Yoshua Bengio. Challenges in Representation Learning: A Report on Three Machine Learning Contests. *Neural Networks*, 2013. 16, 17
- [22] Yash Goyal, Tejas Khot, Douglas Summers-Stay, Dhruv Batra, and Devi Parikh. Making the v in vqa matter: Elevating the role of image understanding in visual question answering. In *CVPR*, 2017. 17
- [23] Danna Gurari, Qing Li, Abigale J Stangl, Anhong Guo, Chi Lin, Kristen Grauman, Jiebo Luo, and Jeffrey P Bigham. Vizwiz grand challenge: Answering visual questions from blind people. In *CVPR*, 2018. 17
- [24] Patrick Helber, Benjamin Bischke, Andreas Dengel, and Damian Borth. Eurosat: A novel dataset and deep learning benchmark for land use and land cover classification. *Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 2019. 16
- [25] Dan Hendrycks, Steven Basart, Norman Mu, Saurav Kadavath, Frank Wang, Evan Dorundo, Rahul Desai, Tyler Zhu, Samyak Parajuli, Mike Guo, et al. The many faces of robustness: A critical analysis of out-of-distribution generalization. In *ICCV*, 2021. 17
- [26] Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, Weizhu Chen, et al. Lora: Low-rank adaptation of large language models. In *ICLR*, 2022. 2
- [27] Chenyu Huang, Peng Ye, Tao Chen, Tong He, Xiangyu

- Yue, and Wanli Ouyang. Emr-merging: Tuning-free high-performance model merging. In *NeurIPS*, 2024. 8
- [28] Gabriel Ilharco, Mitchell Wortsman, Samir Yitzhak Gadre, Shuran Song, Hannaneh Hajishirzi, Simon Kornblith, Ali Farhadi, and Ludwig Schmidt. Patching open-vocabulary models by interpolating weights. In *NeurIPS*, 2022. 1
- [29] Gabriel Ilharco, Marco Túlio Ribeiro, Mitchell Wortsman, Ludwig Schmidt, Hannaneh Hajishirzi, and Ali Farhadi. Editing models with task arithmetic. In *ICLR*, 2023. 1, 2, 3, 6, 8, 14, 20
- [30] Xisen Jin, Xiang Ren, Daniel Preotiuc-Pietro, and Pengxiang Cheng. Dataless knowledge fusion by merging weights of language models. In *ICLR*, 2023. 1
- [31] Sahar Kazemzadeh, Vicente Ordonez, Mark Matten, and Tamara Berg. Referitgame: Referring to objects in photographs of natural scenes. In *EMNLP*, 2014. 17
- [32] Jonathan Krause, Michael Stark, Jia Deng, and Li Fei-Fei. 3D Object representations for fine-grained categorization. In *ICCV Workshops*, 2013. 16
- [33] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. *University of Toronto*, 2009. 16, 17
- [34] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 1998. 16
- [35] Mario Lezcano-Casado and David Martínez-Rubio. Cheap orthogonal constraints in neural networks: A simple parametrization of the orthogonal and unitary group. In *ICML*, 2019. 14
- [36] Longhua Li, Lei Qi, Qi Tian, and Xin Geng. Model merging in the essential subspace. In *CVPR*, 2026. 8
- [37] Weishi Li, Yong Peng, Miao Zhang, Liang Ding, Han Hu, and Li Shen. Deep model fusion: A survey. *arXiv preprint arXiv: 2309.15698*, 2023. 1
- [38] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. In *NeurIPS*, 2023. 6
- [39] Pan Lu, Liang Qiu, Jiaqi Chen, Tony Xia, Yizhou Zhao, Wei Zhang, Zhou Yu, Xiaodan Liang, and Song-Chun Zhu. Iconqa: A new benchmark for abstract diagram understanding and visual language reasoning. *arXiv preprint arXiv:2110.13214*, 2021. 17
- [40] Pan Lu, Swaroop Mishra, Tanglin Xia, Liang Qiu, Kai-Wei Chang, Song-Chun Zhu, Oyvind Taffjord, Peter Clark, and Ashwin Kalyan. Learn to explain: Multimodal reasoning via thought chains for science question answering. In *NeurIPS*, 2022. 17
- [41] Pan Lu, Liang Qiu, Kai-Wei Chang, Ying Nian Wu, Song-Chun Zhu, Tanmay Rajpurohit, Peter Clark, and Ashwin Kalyan. Dynamic prompt learning via policy gradient for semi-structured mathematical reasoning. *arXiv preprint arXiv:2209.14610*, 2022. 17
- [42] Zhenyi Lu, Chenghao Fan, Wei Wei, Xiaoye Qu, Danyang Chen, and Yu Cheng. Twin-merging: Dynamic integration of modular expertise in model merging. In *NeurIPS*, 2024. 8
- [43] Mao-Lin Luo, Zi-Hao Zhou, Yi-Lin Zhang, Yuanyu Wan, Min-Ling Zhang, and Tong Wei. KeepLoRA: Continual learning with residual gradient adaptation. In *The Fourteenth International Conference on Learning Representations*, 2026. 1
- [44] Junhua Mao, Jonathan Huang, Alexander Toshev, Oana Camburu, Alan L Yuille, and Kevin Murphy. Generation and comprehension of unambiguous object descriptions. In *CVPR*, 2016. 17
- [45] Daniel Marczak, Bartłomiej Twardowski, Tomasz Trzcinski, and Sebastian Cygert. MagMax: Leveraging Model Merging for Seamless Continual Learning. In *ECCV*, 2024. 1
- [46] Daniel Marczak, Simone Magistri, Sebastian Cygert, Bartłomiej Twardowski, Andrew D. Bagdanov, and Joost van de Weijer. No Task Left Behind: Isotropic Model Merging with Common and Task-Specific Subspaces. In *ICML*, 2025. 4, 5, 6, 8, 18, 21, 22
- [47] Anand Mishra, Shashank Shekhar, Ajeet Kumar Singh, and Anirban Chakraborty. Ocr-vqa: Visual question answering by reading text in images. In *ICDAR*, 2019. 17
- [48] Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bisacco, Bo Wu, and Andrew Y Ng. Reading digits in natural images with unsupervised feature learning. In *NeurIPS Workshops*, 2011. 16
- [49] Maria-Elena Nilsback and Andrew Zisserman. Automated Flower Classification over a Large Number of Classes. In *2008 Sixth Indian Conference on Computer Vision, Graphics & Image Processing*, 2008. 16
- [50] Omkar M Parkhi, Andrea Vedaldi, Andrew Zisserman, and C. V. Jawahar. Cats and dogs. In *CVPR*, 2012. 16
- [51] Bryan A Plummer, Liwei Wang, Chris M Cervantes, Juan C Caicedo, Julia Hockenmaier, and Svetlana Lazebnik. Flickr30k entities: Collecting region-to-phrase correspondences for richer image-to-sentence models. In *ICCV*, 2015. 17
- [52] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *ICML*, 2021. 1, 6
- [53] Dustin Schwenk, Apoorv Khandelwal, Christopher Clark, Kenneth Marino, and Roozbeh Mottaghi. A-okvqa: A benchmark for visual question answering using world knowledge. In *ECCV*, 2022. 17
- [54] Peter H. Schönemann. A generalized solution of the orthogonal procrustes problem. *Psychometrika*, 1966. 6, 14
- [55] Kexuan Shi, Yandong Wen, and Weiyang Liu. Model merging with functional dual anchors. *arXiv preprint arXiv:2510.21223*, 2025. 8
- [56] Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Ng, and Christopher Potts. Recursive deep models for semantic compositionality over a sentiment treebank. In *EMNLP*, 2013. 16, 17
- [57] Johannes Stalkamp, Marc Schlipf, Jan Salmen, and Christian Igel. The german traffic sign recognition benchmark: a multi-class classification competition. In *IJCNN*, 2011. 16
- [58] George Stoica, Pratik Ramesh, Boglarka Ecsedi, Leshem Choshen, and Judy Hoffman. Model merging with SVD to tie the Knots. In *ICLR*, 2025. 6, 8, 17, 20

- [59] Vinita Vasudevan and M. Ramakrishna. A hierarchical singular value decomposition algorithm for low rank matrices. *arXiv preprint arXiv: 1710.02812*, 2017. [21](#)
- [60] Bastiaan S. Veeling, Jasper Linmans, Jim Winkens, Taco Cohen, and Max Welling. Rotation Equivariant CNNs for Digital Pathology. In *MICCAI*, 2018. [16](#)
- [61] Bryan Wang, Gang Li, Xin Zhou, Zhouong Chen, Tovi Grossman, and Yang Li. Screen2words: Automatic mobile ui summarization with multimodal learning. In *UIST*, 2021. [17](#)
- [62] Ke Wang, Nikolaos Dimitriadis, Guillermo Ortiz-Jiménez, François Fleuret, and Pascal Frossard. Localizing task information for improved model merging and compression. In *ICML*, 2024. [6](#), [8](#)
- [63] Yongxian Wei, Anke Tang, Li Shen, Chun Yuan, and Xiaochun Cao. Modeling multi-task model merging as adaptive projective gradient descent. *arXiv preprint arXiv:2501.01230*, 2025. [8](#)
- [64] Mitchell Wortsman, Gabriel Ilharco, Samir Yitzhak Gadre, Rebecca Roelofs, Raphael Gontijo-Lopes, Ari S Morcos, Hongseok Namkoong, Ali Farhadi, Yair Carmon, Simon Kornblith, et al. Model soups: averaging weights of multiple fine-tuned models improves accuracy without increasing inference time. In *ICML*, 2022. [6](#)
- [65] 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 *CVPR*, 2022. [1](#)
- [66] Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. *arXiv preprint arXiv: 1708.07747*, 2017. [16](#), [17](#)
- [67] Jianxiong Xiao, Krista A Ehinger, James Hays, Antonio Torralba, and Aude Oliva. Sun database: Exploring a large collection of scene categories. *IJCV*, 2016. [16](#)
- [68] Prateek Yadav, Derek Tam, Leshem Choshen, Colin Raffel, and Mohit Bansal. TIES-merging: Resolving interference when merging models. In *NeurIPS*, 2023. [1](#), [6](#), [8](#), [18](#)
- [69] Le Yu, Bowen Yu, Haiyang Yu, Fei Huang, and Yongbin Li. Language models are super mario: Absorbing abilities from homologous models as a free lunch. In *ICML*, 2024. [6](#)
- [70] Fanhu Zeng, Haiyang Guo, Fei Zhu, Li Shen, and Hao Tang. Robustmerge: Parameter-efficient model merging for mllms with direction robustness. *arXiv preprint arXiv:2502.17159*, 2025. [6](#), [7](#), [17](#), [22](#)
- [71] Xiaohua Zhai, Basil Mustafa, Alexander Kolesnikov, and Lucas Beyer. Sigmoid loss for language image pre-training. In *ICCV*, 2023. [1](#)

DC-Merge: Improving Model Merging with Directional Consistency

Supplementary Material

A. Notations

We present the definitions used in our paper and their corresponding mathematical symbols in Table 7 for ease of reference.

B. Proof of Proposition 1

Proof 1 Let the (reduced) SVDs (knowledge-vector decompositions) of the two task updates be

$$\begin{aligned}\Delta \mathbf{W}_s &= \sum_{i=1}^n \sigma_s^i \mathbf{u}_s^i (\mathbf{v}_s^i)^\top = \mathbf{U}_s \text{diag}(\boldsymbol{\sigma}^s) \mathbf{V}_s^\top, \\ \Delta \mathbf{W}_t &= \sum_{j=1}^m \sigma_t^j \mathbf{u}_t^j (\mathbf{v}_t^j)^\top = \mathbf{U}_t \text{diag}(\boldsymbol{\sigma}^t) \mathbf{V}_t^\top,\end{aligned}\quad (12)$$

where $\{\mathbf{u}_s^i\}_{i=1}^n$ and $\{\mathbf{v}_s^i\}_{i=1}^n$ (resp. $\{\mathbf{u}_t^j\}_{j=1}^m$ and $\{\mathbf{v}_t^j\}_{j=1}^m$) are orthonormal left/right singular vectors of $\Delta \mathbf{W}_s$ (resp. $\Delta \mathbf{W}_t$), and $\boldsymbol{\sigma}_s \in \mathbb{R}_{\geq 0}^n$, $\boldsymbol{\sigma}_t \in \mathbb{R}_{\geq 0}^m$ collect the singular values in descending order. We calculate the numerator and denominator of CosSim as follows:

Numerator. Using the Frobenius inner product $\langle \mathbf{A}, \mathbf{B} \rangle = \text{tr}(\mathbf{A}^\top \mathbf{B})$ and bilinearity, we have:

$$\begin{aligned}\langle \Delta \mathbf{W}_s, \Delta \mathbf{W}_t \rangle &= \left\langle \sum_{i=1}^n \sigma_s^i \mathbf{u}_s^i (\mathbf{v}_s^i)^\top, \sum_{j=1}^m \sigma_t^j \mathbf{u}_t^j (\mathbf{v}_t^j)^\top \right\rangle \\ &= \sum_{i=1}^n \sum_{j=1}^m \sigma_s^i \sigma_t^j \left\langle \mathbf{u}_s^i (\mathbf{v}_s^i)^\top, \mathbf{u}_t^j (\mathbf{v}_t^j)^\top \right\rangle.\end{aligned}\quad (13)$$

For rank-one matrices $\mathbf{a}\mathbf{b}^\top$ and $\mathbf{c}\mathbf{d}^\top$ we have $\langle \mathbf{a}\mathbf{b}^\top, \mathbf{c}\mathbf{d}^\top \rangle = (\mathbf{a}^\top \mathbf{c})(\mathbf{b}^\top \mathbf{d})$. Thus, we have:

$$\begin{aligned}\left\langle \mathbf{u}_s^i (\mathbf{v}_s^i)^\top, \mathbf{u}_t^j (\mathbf{v}_t^j)^\top \right\rangle &= (\mathbf{u}_s^i)^\top \mathbf{u}_t^j \cdot (\mathbf{v}_s^i)^\top \mathbf{v}_t^j \\ &= (\mathbf{u}_s^i)^\top \mathbf{u}_t^j \cdot (\mathbf{v}_t^j)^\top \mathbf{v}_s^i.\end{aligned}\quad (14)$$

Therefore, we have:

$$\begin{aligned}\langle \Delta \mathbf{W}_s, \Delta \mathbf{W}_t \rangle &= \sum_{i=1}^n \sum_{j=1}^m \sigma_s^i \sigma_t^j \underbrace{(\mathbf{u}_s^i)^\top \mathbf{u}_t^j (\mathbf{v}_t^j)^\top \mathbf{v}_s^i}_{\mathbf{R}_{i,j}(s,t)} \\ &= \boldsymbol{\sigma}_t \mathbf{R}(s,t) \boldsymbol{\sigma}_s^\top = \boldsymbol{\sigma}_s \mathbf{R}(s,t) \boldsymbol{\sigma}_t^\top,\end{aligned}\quad (15)$$

where we have defined $\mathbf{R}(s,t) \in \mathbb{R}^{n \times m}$ entrywise by $\mathbf{R}_{i,j}(s,t) = (\mathbf{u}_s^i)^\top \mathbf{u}_t^j (\mathbf{v}_t^j)^\top \mathbf{v}_s^i$.

Denominator. For any matrix \mathbf{A} with SVD $\mathbf{A} = \sum_{\ell} \sigma^\ell \mathbf{u}^\ell \mathbf{v}^{\ell \top}$, the Frobenius norm satisfies $\|\mathbf{A}\|_F^2 = \sum_{\ell} (\sigma^\ell)^2$. Hence

$$\begin{aligned}\|\Delta \mathbf{W}_s\|_F &= \left(\sum_{i=1}^n (\sigma_s^i)^2 \right)^{1/2} = \|\boldsymbol{\sigma}_s\|_2, \\ \|\Delta \mathbf{W}_t\|_F &= \left(\sum_{j=1}^m (\sigma_t^j)^2 \right)^{1/2} = \|\boldsymbol{\sigma}_t\|_2.\end{aligned}\quad (16)$$

Conclusion. Combining the numerator and denominator gives

$$\begin{aligned}\text{CosSim}(\Delta \mathbf{W}_s, \Delta \mathbf{W}_t) &= \frac{\langle \Delta \mathbf{W}_s, \Delta \mathbf{W}_t \rangle}{\|\Delta \mathbf{W}_s\|_F \|\Delta \mathbf{W}_t\|_F} \\ &= \frac{\boldsymbol{\sigma}_s \mathbf{R}(s,t) (\boldsymbol{\sigma}_t)^\top}{\|\boldsymbol{\sigma}_s\|_2 \|\boldsymbol{\sigma}_t\|_2},\end{aligned}\quad (17)$$

which proves the proposition.

C. CosSim as Expressiveness via Coefficient Projections

Setting and notation. Let $\Delta \in \mathbb{R}^{m \times n}$, and let $\mathbf{U} \in \mathbb{R}^{m \times k}$, $\mathbf{V} \in \mathbb{R}^{n \times k}$ have orthonormal columns, i.e., $\mathbf{U}^\top \mathbf{U} = \mathbf{V}^\top \mathbf{V} = \mathbf{I}_k$. Define the bi-directional coefficient vector

$$\boldsymbol{\sigma}(\mathbf{U}, \mathbf{V}, \Delta) \triangleq \text{diag}(\mathbf{U}^\top \Delta \mathbf{V}) \in \mathbb{R}^k.$$

Proposition 2 With the notation above, the following hold:

(A) Least-squares fitting \leftrightarrow maximizing ℓ_2 -coefficients.

$$\min_{\mathbf{U}, \mathbf{V}} \min_{\boldsymbol{\sigma} \in \mathbb{R}^k} \left\| \Delta - \mathbf{U} \text{diag}(\boldsymbol{\sigma}) \mathbf{V}^\top \right\|_F \leftrightarrow \max_{\mathbf{U}, \mathbf{V}} \|\boldsymbol{\sigma}(\mathbf{U}, \mathbf{V}, \Delta)\|_2 \quad (18)$$

(B) Cosine similarity with the unweighted dyadic sum \leftrightarrow maximizing ℓ_1 -coefficients.

$$\max_{\mathbf{U}, \mathbf{V}} \text{CosSim}(\Delta, \mathbf{U}\mathbf{V}^\top) \leftrightarrow \max_{\mathbf{U}, \mathbf{V}} \|\boldsymbol{\sigma}(\mathbf{U}, \mathbf{V}, \Delta)\|_1 \quad (19)$$

The equivalences are up to the usual sign-alignment of column pairs $\{(\mathbf{u}_j, \mathbf{v}_j)\}_{j=1}^k$ (flipping both signs leaves $\mathbf{U}\mathbf{V}^\top$ unchanged), which can be chosen to make all entries of $\boldsymbol{\sigma}(\mathbf{U}, \mathbf{V}, \Delta)$ nonnegative.

Table 7. List of definitions and their corresponding mathematical symbols used in this paper.

Symbol	Definition
$\mathbf{W}_0, \{\mathbf{W}_i\}_{i=1}^T$	The weights of pretrained model
$\{\mathbf{W}_i\}_{i=1}^T$	The weights of T tasks fine-tuned model
$\Delta \mathbf{W}_i$	The task vector of task i
$\Delta \mathbf{W}_i = \sum_{j=1}^r \sigma_i^j \mathbf{u}_i^j \mathbf{v}_i^{j\top}$	Knowledge decomposition of $\Delta \mathbf{W}_i$
$\sigma_i^j \mathbf{u}_i^j \mathbf{v}_i^{j\top}$	j -th knowledge vector of $\Delta \mathbf{W}_i$
$\mathbf{u}_i^j \mathbf{v}_i^{j\top}$	j -th knowledge component of $\Delta \mathbf{W}_i$
$\boldsymbol{\sigma}_i = (\sigma_i^1, \dots, \sigma_i^r)$	Energy distribution of task i
$\{\mathbf{u}_i^j \mathbf{v}_i^{j\top}\}_{j=1}^r$	Directional geometry of task i
$\Delta \bar{\mathbf{W}}_i = \left(\frac{\sum_{j=1}^r \sigma_i^j}{r} \right) \left(\sum_{j=1}^r \mathbf{u}_i^j \mathbf{v}_i^{j\top} \right)$	Energy-balanced task vector of task i
$\tilde{\mathbf{W}}, \Delta \tilde{\mathbf{W}}$	The weights of merged model and merged task vector, respectively
$\mathbf{U}_i^{(r)}, \mathbf{V}_i^{(r)}$	The top- r left and right singular vectors of $\Delta \mathbf{W}_i$
\mathbf{U}, \mathbf{V}	The concatenated per-task knowledge basis
$(\tilde{\mathbf{U}}, \tilde{\mathbf{V}})$	The shared cover basis across all tasks
$\mathbf{M}_i = \tilde{\mathbf{U}}^\top \Delta \mathbf{W}_i \tilde{\mathbf{V}}$	Task vector i under shared cover space induced by $(\tilde{\mathbf{U}}, \tilde{\mathbf{V}})$
$\tilde{\mathbf{M}} = \text{Element-wise Merging}(\{\mathbf{M}_i\}_{i=1}^T)$	The merged vector under shared cover space
$\mathcal{M} = \text{block-diag}(\mathbf{1}_{r \times r}, \dots, \mathbf{1}_{r \times r})$	Structural mask for reconstructing parameters

Proof 2 (A). Fix \mathbf{U}, \mathbf{V} orthonormal and consider

$$\min_{\boldsymbol{\sigma} \in \mathbb{R}^k} \left\| \Delta - \mathbf{U} \text{diag}(\boldsymbol{\sigma}) \mathbf{V}^\top \right\|_F^2.$$

Let $\mathbf{C} = \mathbf{U}^\top \Delta \mathbf{V}$ and $\mathbf{d} = \text{diag}(\mathbf{C}) = \boldsymbol{\sigma}(\mathbf{U}, \mathbf{V}, \Delta)$. Using orthonormality and the bilinearity of the Frobenius inner product,

$$\langle \Delta, \mathbf{U} \text{diag}(\boldsymbol{\sigma}) \mathbf{V}^\top \rangle = \langle \mathbf{C}, \text{diag}(\boldsymbol{\sigma}) \rangle = \sum_{j=1}^k d_j \sigma_j.$$

Therefore

$$\begin{aligned} & \left\| \Delta - \mathbf{U} \text{diag}(\boldsymbol{\sigma}) \mathbf{V}^\top \right\|_F^2 \\ &= \|\Delta\|_F^2 + \|\text{diag}(\boldsymbol{\sigma})\|_F^2 - 2 \sum_{j=1}^k d_j \sigma_j \\ &= \|\Delta\|_F^2 + \sum_{j=1}^k (\sigma_j^2 - 2d_j \sigma_j). \end{aligned} \quad (20)$$

The unique minimizer is $\sigma_j^* = d_j$ for all j , i.e., $\boldsymbol{\sigma}^* = \mathbf{d} = \boldsymbol{\sigma}(\mathbf{U}, \mathbf{V}, \Delta)$, and the minimal value equals

$$\|\Delta\|_F^2 - \sum_{j=1}^k d_j^2 = \|\Delta\|_F^2 - \|\boldsymbol{\sigma}(\mathbf{U}, \mathbf{V}, \Delta)\|_2^2.$$

Hence, after inner minimization over $\boldsymbol{\sigma}$, the outer optimization over \mathbf{U}, \mathbf{V} minimizes the residual iff it maximizes $\|\boldsymbol{\sigma}(\mathbf{U}, \mathbf{V}, \Delta)\|_2$, proving Eq. (18).

(B). Using $\langle \mathbf{A}, \mathbf{B} \rangle = \text{tr}(\mathbf{A}^\top \mathbf{B})$, we have:

$$\langle \Delta, \mathbf{U} \mathbf{V}^\top \rangle = \text{tr}(\mathbf{U}^\top \Delta \mathbf{V}) = \sum_{j=1}^k (\mathbf{U}^\top \Delta \mathbf{V})_{jj}$$

$$= \sum_{j=1}^k \sigma_j(\mathbf{U}, \mathbf{V}, \Delta) = \|\boldsymbol{\sigma}(\mathbf{U}, \mathbf{V}, \Delta)\|_1$$

(21)

after sign alignment (flip $(\mathbf{u}_j, \mathbf{v}_j)$ together if needed). Moreover, $\|\mathbf{U} \mathbf{V}^\top\|_F^2 = \text{tr}(\mathbf{V} \mathbf{U}^\top \mathbf{U} \mathbf{V}^\top) = \text{tr}(\mathbf{V} \mathbf{V}^\top) = k$. Therefore

$$\begin{aligned} \text{CosSim}(\Delta, \mathbf{U} \mathbf{V}^\top) &= \frac{\langle \Delta, \mathbf{U} \mathbf{V}^\top \rangle}{\|\Delta\|_F \|\mathbf{U} \mathbf{V}^\top\|_F} \\ &= \frac{\|\boldsymbol{\sigma}(\mathbf{U}, \mathbf{V}, \Delta)\|_1}{\|\Delta\|_F \sqrt{k}}. \end{aligned} \quad (22)$$

Since the denominator is independent of (\mathbf{U}, \mathbf{V}) , maximizing the cosine is equivalent to maximizing $\|\boldsymbol{\sigma}(\mathbf{U}, \mathbf{V}, \Delta)\|_1$, establishing Eq. (19).

Interpretation. Each entry of $\boldsymbol{\sigma}(\mathbf{U}, \mathbf{V}, \Delta) = \text{diag}(\mathbf{U}^\top \Delta \mathbf{V})$ is a *bi-directional projection* of Δ onto the dyad $\mathbf{u}_j \mathbf{v}_j^\top$ (input- and output-side alignments jointly). Part (A) shows that the best reconstruction by dyads in the basis (\mathbf{U}, \mathbf{V}) is achieved when the coefficients equal these projections, and the residual decreases as the ℓ_2 energy of the projections increases. Part (B) shows that the cosine with the *unweighted* dyadic sum $\mathbf{U} \mathbf{V}^\top$ is (up to a constant)

the ℓ_1 sum of these projections. Consequently, a higher cosine indicates stronger ability of the basis (\mathbf{U}, \mathbf{V}) to *express* (encode) the knowledge contained in Δ —exactly mirroring the main-text view where $\mathbf{R}(s, t)$ aggregates bi-directional projections to quantify how effectively task s represents task t .

D. Theoretical Analysis of Merging in the Cover Space

In this section, we first provide a theoretical justification for whitening-based construction of cover space. The empirical results provide an intuitive and solid validation of the proposed objective in Eq. (8). We also present visualization of task representations in cover space and investigate the effect of structural mask. Finally, we provide a toy example to intuitively demonstrate the crucial role that shared cover space plays in directional preservation.

Theoretical Analysis of Whitening-based Cover Space Construction. First, by expanding and applying the Hadamard norm inequality, we obtain:

$$\begin{aligned}
& \sum_{i=1}^T \sum_{j=1}^r \left\| \sigma(\tilde{\mathbf{U}}, \tilde{\mathbf{V}}, \mathbf{u}_i^j \mathbf{v}_i^{j\top}) \right\|_2^2 \\
&= \sum_{i=1}^T \sum_{j=1}^r \left\| \text{diag} \left((\tilde{\mathbf{U}}^\top \mathbf{u}_i^j) (\mathbf{v}_i^{j\top} \tilde{\mathbf{V}}) \right) \right\|_2^2 \\
&= \sum_{i=1}^T \sum_{j=1}^r \left\| (\tilde{\mathbf{U}}^\top \mathbf{u}_i^j) \odot (\tilde{\mathbf{V}}^\top \mathbf{v}_i^j) \right\|_2^2 \quad (23) \\
&= \left\| (\tilde{\mathbf{U}}^\top \mathbf{U}) \odot (\tilde{\mathbf{V}}^\top \mathbf{V}) \right\|_F^2 \\
&\leq \left\| \tilde{\mathbf{U}}^\top \mathbf{U} \right\|_F^2 \left\| \tilde{\mathbf{V}}^\top \mathbf{V} \right\|_F^2,
\end{aligned}$$

where $\mathbf{U} = ([\mathbf{U}_1^{(r)}, \dots, \mathbf{U}_T^{(r)}])$ and $\mathbf{V} = ([\mathbf{V}_1^{(r)}, \dots, \mathbf{V}_T^{(r)}])$. This shows that the surrogate objective is upper-bounded by two alignment terms $\left\| \tilde{\mathbf{U}}^\top \mathbf{U} \right\|_F^2$ and $\left\| \tilde{\mathbf{V}}^\top \mathbf{V} \right\|_F^2$, which characterize how well $\tilde{\mathbf{U}}$ aligns with all $\mathbf{U}_i^{(r)}$ and how well $\tilde{\mathbf{V}}$ aligns with all $\mathbf{V}_i^{(r)}$.

Next, using the diagonal-trace inequality and arithmetic mean inequality, we have:

$$\begin{aligned}
\left\| \tilde{\mathbf{U}}^\top \mathbf{U} \right\|_F^2 &\geq \left\| \text{diag} \left(\tilde{\mathbf{U}}^\top \mathbf{U} \right) \right\|_2^2 \geq \frac{1}{k} \text{tr} \left(\tilde{\mathbf{U}}^\top \mathbf{U} \right)^2, \\
\left\| \tilde{\mathbf{V}}^\top \mathbf{V} \right\|_F^2 &\geq \left\| \text{diag} \left(\tilde{\mathbf{V}}^\top \mathbf{V} \right) \right\|_2^2 \geq \frac{1}{k} \text{tr} \left(\tilde{\mathbf{V}}^\top \mathbf{V} \right)^2, \quad (24)
\end{aligned}$$

we see that maximizing the surrogate in Eq. (8) effectively corresponds to maximizing these trace terms. Hence, the optimal $(\tilde{\mathbf{U}}, \tilde{\mathbf{V}})$ should maximize the average alignment with each task knowledge directions.

Finally, whitening [54] naturally achieves this objective:

$$\begin{aligned}
\tilde{\mathbf{U}} &= \arg \max_{\tilde{\mathbf{U}}^\top \tilde{\mathbf{U}} = \mathbf{I}} \langle \mathbf{U}, \tilde{\mathbf{U}} \rangle, \\
\tilde{\mathbf{V}} &= \arg \max_{\tilde{\mathbf{V}}^\top \tilde{\mathbf{V}} = \mathbf{I}} \langle \mathbf{V}, \tilde{\mathbf{V}} \rangle. \quad (25)
\end{aligned}$$

This whitening-based alignment maximizes $\text{tr}(\tilde{\mathbf{U}}^\top \mathbf{U})$ and $\text{tr}(\tilde{\mathbf{V}}^\top \mathbf{V})$, which correspond to the lower-bound traces in Eq. (24). Therefore, whitening can be interpreted as an efficient approximate solution to the subspace alignment problem in Eq. (7), providing a computationally simple yet theoretically justified way to construct shared cover basis.

Validation of Objective for Cover Basis. We further verify the rationality of the proposed trace-based objective by directly optimizing it to obtain the cover basis that minimizes reconstruction error, and compare the result with the whitening-based approximation. Our goal is to confirm that whitening indeed provides a close approximation to the optimal cover subspace defined by Eq. (8).

To optimize a set of rank- k orthonormal bases in a d -dimensional space, we employ a differentiable parameterization using skew-symmetric matrices [35]. Specifically, any orthonormal matrix $\tilde{\mathbf{U}}, \tilde{\mathbf{V}} \in \mathbb{R}^{d \times k}$ can be expressed as

$$\tilde{\mathbf{U}} = \exp(\mathbf{A}) \tilde{\mathbf{U}}_0, \quad \tilde{\mathbf{V}} = \exp(\mathbf{B}) \tilde{\mathbf{V}}_0 \quad (26)$$

where $\mathbf{A}, \mathbf{B} \in \mathbb{R}^{d \times d}$ is a skew-symmetric matrix ($\mathbf{A}^\top = -\mathbf{A}, \mathbf{B}^\top = -\mathbf{B}$) and $\tilde{\mathbf{U}}_0, \tilde{\mathbf{V}}_0$ is an initial orthonormal basis. This parameterization utilizes the isomorphic properties of Lie groups and rotation operations [35] and ensures that $\tilde{\mathbf{U}}, \tilde{\mathbf{V}}$ always remains on the *Stiefel manifold* $\mathcal{S}(d, k)$, thus preserving orthogonality during gradient-based optimization.

To further validate the rationality of our optimization target, we optimize the cover basis following Algorithm 2. Starting from a poor initialization $(\tilde{\mathbf{U}}_0, \tilde{\mathbf{V}}_0)$ obtained by decomposing the merged task vector from TA [29] through SVD, we perform iterative optimization toward the objective in Eq. (8). Figure 4 illustrates the optimization trajectory of the alignment score (left) and its logarithmic scale counterpart (right), along with the corresponding accuracy evolution by using corresponding cover basis. Initially, the alignment score is low, indicating poor consistency between the cover basis and the directions of each task. As optimization proceeds, both the alignment score and downstream task accuracy increase steadily and exhibit similar trends, confirming that higher alignment directly translates to improved task retention and generalization. In particular, the convergence of the alignment score to a stable maximum implies that the optimized basis successfully captures the shared directional geometry among all tasks.

Algorithm 2 Optimization of Cover Basis via Skew-Symmetric Parameterization

- 1: **Input:** Task directional geometry $\{\mathbf{u}_i^j \mathbf{v}_i^{j\top}\}_{j=1}^r$, Task vectors $\{\Delta_i\}_{i=1}^T$, learning rate η , maximum iteration steps L
 - 2: **Output:** Optimized cover basis $\tilde{\mathbf{U}}, \tilde{\mathbf{V}}$
 - 3: ▷ Step 1: Initialize cover basis
 - 4: Initialize $\mathbf{A}, \mathbf{B} \in \mathbb{R}^{d \times d}$ as zero matrices
 - 5: $\tilde{\mathbf{U}}_0, \tilde{\mathbf{V}}_0 = \text{SVD}(\sum_{i=1}^T \Delta_i)$
 - 6: ▷ Step 2: Iterative optimization of cover basis
 - 7: **for** $\ell = 1 \rightarrow L$ **do**
 - 8: $\tilde{\mathbf{U}} \leftarrow \exp(\mathbf{A} - \mathbf{A}^\top) \tilde{\mathbf{U}}_0, \tilde{\mathbf{V}} \leftarrow \exp(\mathbf{B} - \mathbf{B}^\top) \tilde{\mathbf{V}}_0$
 - 9: Compute alignment score \mathcal{L} from Eq. (8)
 - 10: Compute gradients $\nabla_{\mathbf{A}} \mathcal{L}, \nabla_{\mathbf{B}} \mathcal{L}$
 - 11: $\mathbf{A} \leftarrow \mathbf{A} + \eta \nabla_{\mathbf{A}} \mathcal{L}, \mathbf{B} \leftarrow \mathbf{B} + \eta \nabla_{\mathbf{B}} \mathcal{L}$
 - 12: **end for**
 - 13: **return** $\tilde{\mathbf{U}}, \tilde{\mathbf{V}}$
-

Notably, the final alignment and accuracy achieved by the optimized cover basis closely match those obtained via our whitening-based construction, validating that whitening provides a strong approximation to the optimal solution of the cover subspace objective. Therefore, this empirical results confirm the theoretical justification that whitening acts as an efficient surrogate optimizer for constructing a shared cover basis.

Visualization of Task Representations in Cover Space.

To provide an intuitive illustration of how each task vector is represented in the cover space constructed by Algorithm 2, we visualize the representation in Figure 14 using a block-wise averaged aggregation. We perform the optimization for 250 steps on ViT-B-32 8-task benchmark in LoRA setting, and the size of each block is $r \times r$.

Effect of Structural Masking. After projection and merging within the cover space, a block-diagonal mask \mathcal{M} is applied:

$$\Delta_M = \tilde{\mathbf{U}} \left(\hat{\mathbf{M}} \odot \mathcal{M} \right) \tilde{\mathbf{V}}^\top.$$

We now investigate the underlying mechanism of the block-diagonal mask. Consider T task vectors $\{\Delta_i\}_{i=1}^T$ that share approximately directional geometry. Suppose that there exist shared orthonormal basis $\tilde{\mathbf{U}}, \tilde{\mathbf{V}}$ such that each task vector can be represented as:

$$\Delta_i \approx \tilde{\mathbf{U}} \text{diag}(\sigma_i) \tilde{\mathbf{V}}^\top, \quad (27)$$

with $\tilde{\mathbf{U}}^\top \tilde{\mathbf{U}} = \tilde{\mathbf{V}}^\top \tilde{\mathbf{V}} = \mathbf{I}$. In this case, the merged model vector obtained by summation or averaging becomes:

$$\sum_{i=1}^T \Delta_i \approx \tilde{\mathbf{U}} \left(\sum_{i=1}^T \text{diag}(\sigma_i) \right) \tilde{\mathbf{V}}^\top, \quad (28)$$

which clearly preserves the original knowledge directions $\tilde{\mathbf{U}}$ and $\tilde{\mathbf{V}}$ and σ_i is the corresponding optimal coefficients under $\tilde{\mathbf{U}}$ and $\tilde{\mathbf{V}}$ for Δ_i . That is, merging under a fixed cover basis ensures that the directional geometry of all task vectors remains unchanged while only the coefficients are combined. We construct a pair of cover basis under which task representations are largely distributed along diagonal blocks. When restricted to these diagonal components, merging does not change task directions. The mask explicitly suppresses off-diagonal elements to remove cross-task directional inconsistency that induces task interference.

The size of mask controls the trade-off between *directional preservation* and *individual task reconstruction fidelity*. In the extreme case where the mask size is 1 (i.e., only diagonal elements retained), merging reduces to purely aggregating the aligned components without mixing any directions, which perfectly preserves subspace directions but sacrifices per-task reconstruction fidelity. As the mask size increases, more cross-task interactions are included, improving reconstruction of each individual task at the cost of slight directional deviation. Therefore, \mathcal{M} implicitly balances between directional consistency and task-specific fidelity.

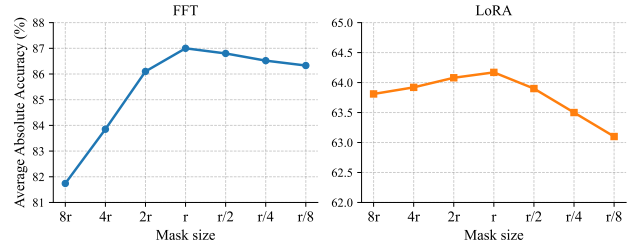


Figure 5. The average absolute accuracy of DC-Merge with mask size $\{8r, 4r, 2r, r, r/2, r/4, r/8\}$ in FFT (left) and LoRA (right) settings. Performance peaks near r for FFT and shows a milder variation for LoRA (slightly improves till r then decreases), indicating that moderate masks retain performance while overly small/large masks degrade it. The results are based on ViT-B-32 8-task benchmark.

Illustrative Example. We provide a simple two-task example to intuitively demonstrate the advantage of merging within the shared cover space. Consider two tasks, each represented by a single knowledge component:

$$\begin{aligned} u_1 = v_1 &= [1, 0]^\top \\ u_2 = v_2 &= [0.1104, 0.9939]^\top. \end{aligned}$$

When directly merging in the parameter space,

$$\Delta_M = u_1 v_1^\top + u_2 v_2^\top = \begin{bmatrix} 1.0121 & 0.1098 \\ 0.1098 & 0.9878 \end{bmatrix}$$

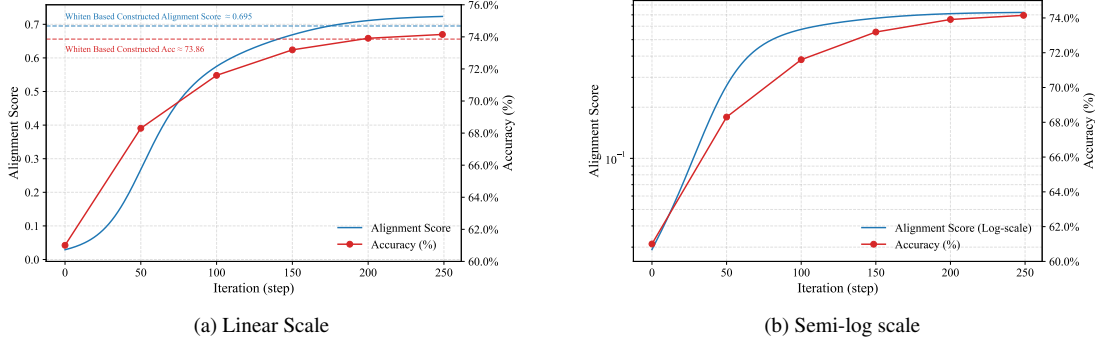


Figure 4. **Optimization trajectory of the cover basis.** We start from a poor initialization $(\tilde{U}_0, \tilde{V}_0)$ obtained by SVD of task vectors from TA and iteratively optimize the surrogate objective in Eq. (8). (a) The optimization trace of alignment score and average normalized accuracy of DC-Merge under linear scale. (b) The same process in semi-log scale for better visibility of early-stage dynamics. Both alignment score and accuracy increase monotonically during training and converge to a stable optimum, demonstrating that higher alignment directly corresponds to stronger task retention. The final accuracy and alignment closely match those of the whitening-based approximation, confirming that whitening yields a near-optimal cover basis.

the resulting SVD gives principal directions:

$$u'_1 = v'_1 = [0.7451, 0.6669]^\top,$$

$$u'_2 = v'_2 = [0.6669, -0.7451]^\top,$$

which significantly deviate from the original task directions, implying that direct merging distorts the directional geometry of task knowledge.

In contrast, when get shared cover basis by whitening the concatenation per-task knowledge basis before merging, we have:

$$\tilde{U} = \tilde{V} = [u'_1, u'_2],$$

where $u'_1 = [0.9985, -0.0533]^\top$, $u'_2 = [0.0533, 0.9985]^\top$. When only using a diagonal mask \mathcal{M} , the resulting directional equal to \tilde{U} , \tilde{V} remains nearly aligned with the original task directions. This demonstrates that whitening preserves the cover geometry of knowledge directions and prevents directional shift during model merging. The visualization of this result can refer to Figure 6.

E. Experimental Details

In this section, we first provide an overview of datasets and evaluation metrics employed in our experiments. We then present the implementation details of our experiments in depth and conclude with a discussion of different smoothing strategies.

E.1. Datasets and Metrics

We present the datasets in the vision model benchmarks as follows. For full fine-tuning setting, the 8-task benchmark comprises the following datasets: Cars [32], DTD [7], EuroSAT [24], GTSRB [57], MNIST [34], RESISC45 [5], SUN397 [67] and SVHN [48]. The 14-task benchmark extends the previous one by incorporating six additional datasets: CIFAR100 [33], STL10 [9],

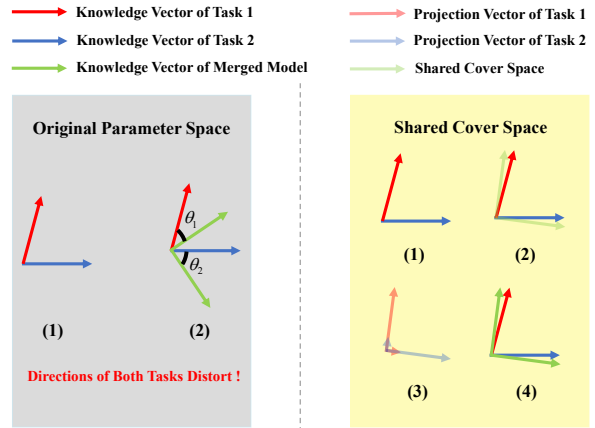


Figure 6. Illustration of mitigating directional shift when merging in the shared cover space. We take u_1 and u_2 as example. Left: (1) The direction of u_1 and u_2 . (2) Directly merging in the original parameter space yields u'_1 and u'_2 , causing severe directional deviation. Right: (1) The direction of u_1 and u_2 . (2) Obtain u'_1 and u'_2 by whitening. (3) Project u_1 and u_2 onto the space spanned by u'_1 and u'_2 . (4) Aggregate the projections along the directions of u'_1 and u'_2 respectively as merged knowledge vectors, which significantly alleviates directional shift.

Flowers102 [49], OxfordIIITPet [50], PCAM [60] and FER2013 [21]. The 20-task benchmark adds another six datasets to the 14-task configuration: EMNIST [10], CIFAR10 [33], Food101 [2], FashionMNIST [66], RenderedSST2 [56] and KMNIST [8].

For LoRA setting, the 8-task benchmark includes the same datasets as in the full fine-tuning setting. The 12-task benchmark extends the previous one by introducing four new datasets: CIFAR100 [33], Flowers102 [49], OxfordIIITPet [50] and STL10 [9]. The 16-task benchmark

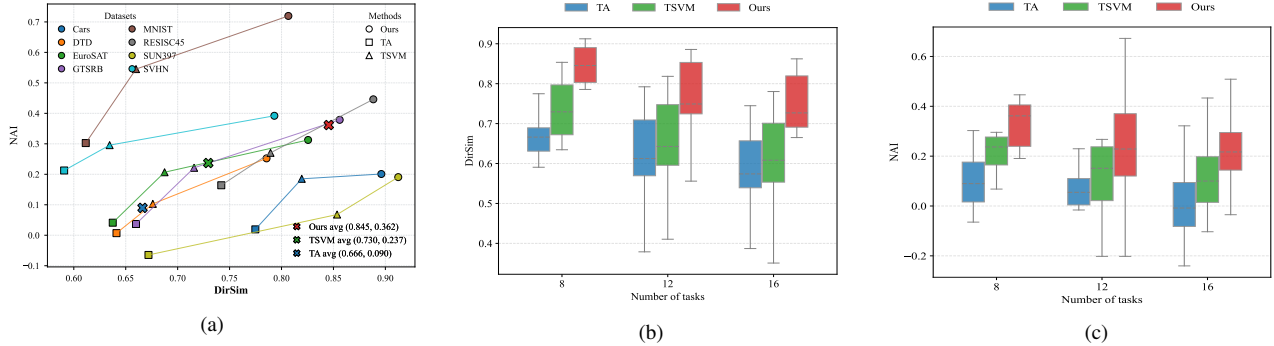


Figure 7. (a) Correlation of task-wise performance with DirSim. The result is based on ViT-B-32 8-task benchmark with our checkpoints, while we utilize checkpoints provided by KnOTS [58] for Figure 3c. (b) Layer-averaged projected DirSim between merged multi-task vector and original task vectors for varying number of tasks. (c) Task-wise NAI of the merged model for varying number of tasks. (b) and (c) are based on merging LoRA fine-tuned ViT-B-32 models.

further adds four datasets to the 12-task configuration: FER2013 [21], CIFAR10 [33], FashionMNIST [66] and RenderedSST2 [56].

The MM-MergeBench [70] contain 8 multimodal datasets as seen tasks: ScienceQA [40], ImageNet [14], VQAv2 [22], REC-COCO [31, 44], OCRVQA [47], Flickr30k [51], VizWiz-caption [23] and IconQA [39], along with 4 additional datasets as unseen tasks for generalizability evaluation: ImageNet-R [25], AOKVQA [53], Screen2Word [61], TabMWP [41].

Since models fine-tuned on different datasets exhibit varying absolute accuracies, we measure the performance of the merged model by its *average normalized accuracy*:

$$\text{Average Normalized Accuracy} = \frac{1}{T} \sum_{i=1}^T \frac{\text{acc}(\theta_M, i)}{\text{acc}(\theta_i, i)} \quad (29)$$

where $\text{acc}(\theta_i, i)$ denotes the absolute accuracy of the model fine-tuned on task i , $\text{acc}(\theta_M, i)$ denotes the absolute accuracy of the merged model on task i , and T is the total number of tasks.

Normalized Accuracy Improvement (NAI) is another commonly-used metric which incorporates zero-shot performance of each task. NAI of task i is defined as

$$\text{NAI}(\theta_M, \theta_i; \theta_0) = \frac{\text{acc}(\theta_M, i) - \text{acc}(\theta_0, i)}{\text{acc}(\theta_i, i) - \text{acc}(\theta_0, i)} \quad (30)$$

where $\text{acc}(\theta_0, i)$ denotes the zero-shot performance on task i . We use NAI to evaluate the task-wise performance of merged models in Figure 3b, 3c and 7.

E.2. Implementation Details

Compute resources. All experiments involving the fine-tuning and merging of vision models are performed on a single NVIDIA RTX 4090, while the merging of vision-language models is conducted on a single NVIDIA A6000.

Choice of rescaling coefficient. For DC-Merge and other baselines which require a rescaling coefficient α before merging, we integrate the merged multi-task vector $\Delta\widetilde{W}$ with the pretrained weights W_0 to obtain the merged model by:

$$\widetilde{W} = W_0 + \alpha\Delta\widetilde{W}, \quad (31)$$

where α is chosen on a held-out validation set. Exceptionally, we adopt a fixed $\alpha = 2.0$ for DC-Merge following RobustMerge [70] in MM-MergeBench.

Choice of checkpoints. We directly utilize the checkpoints released by TSV-M for evaluation in Table 2, which ensures a fair comparison with the state-of-the-art method Iso-CTS and other baselines. Similarly, we adopt the checkpoints released by RobustMerge in MM-Merge-Bench evaluation (Table 3) for fair comparison with baseline methods. Unlike previous works that only consider evaluation for LoRA fine-tuned vision models on the 8-task benchmark, we extend our evaluation to 12-task and 16-task benchmarks. Thus, we perform LoRA fine-tuning of vision models by ourselves and results with respect to LoRA fine-tuned vision models are based on our checkpoints unless otherwise specified. We present the training details as follows.

Fine-tuning vision models with LoRA. We obtain the CLIP visual encoders ViT-B-32, ViT-B-16 and ViT-L-14 from Hugging Face (HF). These models are then fine-tuned by LoRA using `peft` library. Across all our experiments of vision models, we set the LoRA target modules to the query, key, value and output projection weights, which are the only learnable modules. We set the LoRA rank to be 16, LoRA alpha to be 16, LoRA dropout to be 0.1 and disable the use of bias parameters. All vision models are trained using the AdamW optimizer, with a cosine learning rate scheduler using Cross-Entropy loss. We use a standard learning rate of $3e-4$, weight decay of $1e-1$ and label smoothing set to 0 across all datasets and different types of visual encoders. The text encoder in the CLIP model

remains frozen across all our experiments of vision models. **Choice of hyperparameters.** We adopt TIES-Merging (TIES) [68] for aggregation in the shared cover space across all our experiments, and the top- k parameter (i.e., the percentage of elements retained) in TIES is fixed to 0.1. For hyperparameter r , we plot the performance of our method with different r values in Figure 8 for both LoRA and FFT settings. Consequently, in all LoRA fine-tuning scenarios including vision models and vision-language models, we set r to the LoRA rank. For FFT setting, we set r to $\lfloor \frac{\min(m,n)}{T} \rfloor$ (see Algorithm 1).

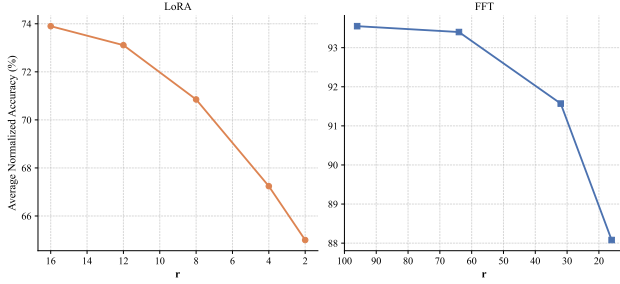


Figure 8. Impact of hyperparameter r in LoRA and FFT settings. We report average normalized accuracy of DC-Merge on ViT-B-32 8-task benchmark. The empirical results align with our motivation to preserve the directional geometry of each task vector after merging as much as possible.

Baseline adaptation in LoRA settings. The state-of-the-art approaches TSV-M [20] and Iso-CTS [46] are also based on SVD. Unfortunately, the former ignores merging models fine-tuned by LoRA and the latter does not specify details of LoRA model merging neither in its paper nor in its code-base. We assume that ranks of the merged vectors $\Delta_{\text{TSV-M}}$ and $\Delta_{\text{Iso-CTS}}$ should be $T \times \text{LoRA_rank}$ where T is the total number of tasks. This setting aligns with our hyperparameter r in LoRA setting.

Merging parameters with non-matrix structure. Some weights of the neural networks are represented by vectors, e.g. bias vectors and parameters of layer normalization [1]. Following previous works [20, 46], we apply simple averaging to merge these parameters.

E.3. Details of Perturbation for Figure 3b

We separately construct controllable energy distribution and directional perturbation to verify that maintaining directional geometry is the key to maintaining performance.

Controllable Energy Distribution Perturbation. When keeping the directions U_i, V_i fixed and perturbing only the energy distribution σ_i to $\hat{\sigma}_i$, we have

$$\text{CosSim}(\Delta_i, \hat{\Delta}_i) = \text{CosSim}(\sigma_i, \hat{\sigma}_i), \quad (32)$$

where $\hat{\Delta}_i = U_i \text{diag}(\hat{\sigma}_i) V_i^\top$. Hence, to obtain a controllable energy distribution perturbation, it suffices to con-

struct $\hat{\sigma}_i$ such that $\text{CosSim}(\sigma_i, \hat{\sigma}_i) = p$, where $p \in [0, 1]$ prescribes the perturbation degree.

Let $\sigma \equiv \sigma_i \in \mathbb{R}_{\geq 0}^r$ and let $\bar{\sigma} \in \mathbb{R}_{\geq 0}^r$ be a *balanced* spectrum (e.g., the averaging or linearly smoothed spectrum that preserves total energy). We define the unit reference $s = \frac{\sigma}{\|\sigma\|_2}$ and extract a component orthogonal to s from $\bar{\sigma}$ by Gram-Schmidt:

$$s^\perp = \frac{\bar{\sigma} - (s^\top \bar{\sigma}) s}{\|\bar{\sigma} - (s^\top \bar{\sigma}) s\|_2} \quad (33)$$

For any target $p \in [0, 1]$, we set

$$\hat{s} = p s + \sqrt{1-p^2} s^\perp, \quad \hat{\sigma} = \|\sigma\|_2 \hat{s}. \quad (34)$$

By definition we have $\|\hat{s}\|_2 = 1$, $\|s\|_2 = 1$, and $s^\top s^\perp = 0$. Therefore,

$$\begin{aligned} \text{CosSim}(\sigma, \hat{\sigma}) &= \frac{\sigma^\top \hat{\sigma}}{\|\sigma\|_2 \|\hat{\sigma}\|_2} = s^\top \hat{s} \\ &= p s^\top s + \sqrt{1-p^2} s^\top s^\perp = p. \end{aligned} \quad (35)$$

Hence Eq. (34) yields an energy distribution perturbation with prescribed cosine p while keeping directions (U_i, V_i) unchanged.

Controllable Directional Perturbation. To investigate the sensitivity of model merging to changes in knowledge directions, we introduce a procedure for applying controllable directional perturbations such that the resulting representation $\hat{\Delta}_i$ satisfies $\text{DirSim}(\Delta_i, \hat{\Delta}_i) = p$. Specifically, we perturb the orthogonal directions U_i and V_i by mixing them with randomly sampled bases from their orthogonal complements.

Let U_i^\perp and V_i^\perp denote the orthogonal complements of U_i and V_i , respectively. We randomly select r orthonormal columns $\hat{U}_i \subset U_i^\perp$ and $\hat{V}_i \subset V_i^\perp$, then construct the perturbed bases $\hat{U}_i(p)$ and $\hat{V}_i(p)$ as linear combinations:

$$\begin{aligned} \hat{U}_i(p) &= \sqrt{p} U_i + \sqrt{1-p} \hat{U}_i, \\ \hat{V}_i(p) &= \sqrt{p} V_i + \sqrt{1-p} \hat{V}_i, \end{aligned} \quad (36)$$

which satisfy the orthonormality $\hat{U}_i(p)^\top \hat{U}_i(p) = \hat{V}_i(p)^\top \hat{V}_i(p) = I$. The perturbed task vector $\hat{\Delta}_i$ with directional geometry $(\hat{U}_i(p), \hat{V}_i(p))$ achieves $\text{DirSim}(\Delta_i, \hat{\Delta}_i) = p$, which can be easily verified by:

$$\begin{aligned} \text{DirSim}(\Delta_i, \hat{\Delta}_i) &= \frac{\text{tr}(U_i^\top \hat{U}_i(p) \hat{V}_i(p)^\top V_i)}{r} \\ &= \frac{\text{tr}(p \cdot I_{r \times r})}{r} = p, \end{aligned} \quad (37)$$

where the second equality holds because $U_i^\top \hat{U}_i(p) = \hat{V}_i(p)^\top V_i = \sqrt{p} \cdot I_{r \times r}$.

Method	ViT-B-16			ViT-L-14			LLaVA-v1.5-7B
	8 tasks	12 tasks	16 tasks	8 tasks	12 tasks	16 tasks	8 tasks
No smoothing	71.64	76.13	75.48	83.96	85.63	80.86	79.10
Linear smoothing	78.69 (+7.05)	80.35 (+4.22)	78.82 (+3.34)	89.22 (+5.26)	89.11 (+3.48)	85.38 (+4.52)	87.58 (+8.48)
Averaging	78.86 (+7.22)	80.56 (+4.43)	78.91 (+3.43)	89.42 (+5.46)	89.31 (+3.68)	85.71 (+4.85)	87.91 (+8.81)

Table 8. Performance with different smoothing strategies. We report average normalized accuracy.

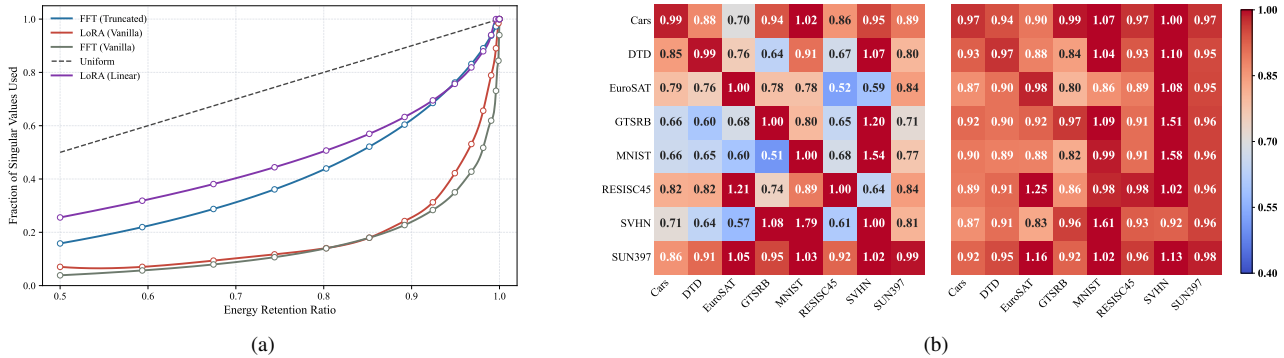


Figure 9. (a) The relationship between energy retention ratios and fractions of singular values averaged across all datasets. The result is based on the ViT-B-32 8-task benchmark. FFT (Truncated) considers reconstructing each task vector using the top- r singular values (and their associated singular vectors) in the full fine-tuning setting without additional smoothing. (b) Comparison of cross-task transferability w/o (left) and w/ (right) further balancing the energy distribution after truncating long-tailed energy parts. Compared with LoRA setting, applying energy smoothing in FFT scenario yields relatively small improvements in generalization.

E.4. Smoothing Strategies

In line 6 of Algorithm 1, we mention the smoothing of $\Sigma_i^{(r)}$. Here, we propose two *explicit* smoothing strategies as follows.

Averaging. We consider the simplest and most effective form of smoothing by replacing all top- r singular values with their average value:

$$\bar{\sigma}_{i,\text{avg}} = \left(\frac{1}{r} \sum_{j=1}^r \sigma_i^j \right) \mathbf{1}_r. \quad (38)$$

Linear Smoothing. Beyond uniform averaging, we further consider a more flexible linear smoothing strategy that controls the energy flattening degree through a single hyperparameter ρ . Specifically, we first constrain the ratio between the largest and smallest smoothed singular values to be no greater than ρ , i.e.,

$$\frac{\bar{\sigma}_{\max}}{\bar{\sigma}_{\min}} = \min \left(\frac{\sigma_{\max}}{\sigma_{\min}}, \rho \right). \quad (39)$$

We then generate a linearly decreasing distribution $\mathbf{w} = [w_1, w_2, \dots, w_r]$ whose entries sum to one, such that $w_1/w_r = \bar{\sigma}_{\max}/\bar{\sigma}_{\min}$. Finally, we scale this distribution by the sum of singular values, yielding the smoothed energy distribution

$$\bar{\sigma}_{i,\text{linear}} = \left(\sum_{j=1}^r \sigma_j \right) \mathbf{w}. \quad (40)$$

This procedure maintains the relative ordering of the singular values while reducing their extreme disparity. We set ρ to 5.0 in all LoRA scenarios, including vision models and vision-language models. We provide a comprehensive comparison between averaging and linear smoothing in Table 4 for ViT-B-32 and Table 8 for ViT-B-16 and ViT-L-14. We report the better result of the two aforementioned smoothing strategies in Table 1 and 3.

Besides, directly utilizing top- r singular values along with their corresponding singular vectors of each task vector to obtain $\bar{\Delta}_i$ without any additional smoothing (i.e., skipping line 6 in Algorithm 1) can also serve as an *implicit* smoothing strategy, as it truncates the long-tailed energy parts. We only apply this truncation strategy to full fine-tuned vision models since we set r to LoRA rank in LoRA setting. Surprisingly, we notice that applying truncation to long-tailed singular values without further smoothing in FFT scenario can generally exhibit similar energy distribution to explicitly linearly smoothing in LoRA setting.

It is worth noting that energy smoothing can enhance cross-task transferability of each task vector at a cost of task performance fidelity (see Figure 2a and 9b), so there may exist a trade-off between its pros and cons. To verify this, we perform a linear interpolation between original singular values of each task vector and averaging smoothed ones by:

$$\bar{\Sigma}_i^{(r)} = \tau \Sigma_{i,\text{orig}}^{(r)} + (1 - \tau) \Sigma_{i,\text{avg}}^{(r)} \quad (41)$$

The results are presented in Figure 10. With τ decreases, the

Method	Datasets								Average
	Cars	DTD	EuroSAT	GTSRB	MNIST	RESISC45	SUN397	SVHN	
Task Arithmetic	82.0	73.6	48.8	42.1	53.1	71.5	97.5	41.2	63.7
KnOTS-TIES	82.7	73.7	49.3	48.9	68.9	70.9	95.5	53.8	68.0
WUDI-Merging	82.4	73.5	48.6	46.8	54.5	72.4	96.0	46.5	65.1
TSV-M	83.9	75.1	52.4	45.5	58.3	73.1	97.6	45.3	66.4
Iso-CTS	83.3	84.7	49.0	79.6	<u>69.4</u>	82.3	99.0	53.2	<u>75.1</u>
DC-Merge	90.9	<u>79.6</u>	65.5	<u>54.0</u>	92.6	<u>75.0</u>	<u>98.2</u>	66.8	77.8

Table 9. Performance on ViT-B-32 8-task benchmark using checkpoints provided by KnOTS. We report normalized accuracy.

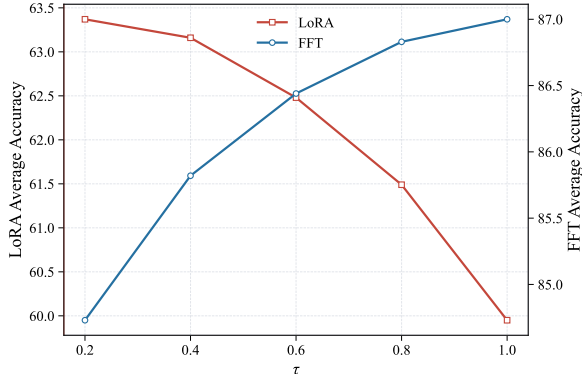


Figure 10. The relationship between average absolute accuracy of DC-Merge in LoRA/FFT setting and interpolation coefficient τ .

energy distribution of each task vector tends to be smoother. We notice a consistent performance improvement as each $\bar{\Sigma}_i^{(r)}$ is smoothed gradually in LoRA setting, while a reverse trend can be observed in FFT setting. This phenomenon suggests that the aforementioned truncation strategy is sufficient to ensure the top- r knowledge components to be adequately expressed within each task vector, while further smoothing induces additional bias for individual tasks and thereby harms overall performance. We thus explicitly apply smoothing to LoRA fine-tuned task vectors across all experiments while obtaining $\bar{\Delta}_i$ solely by retaining top- r knowledge vectors as implicitly smoothing in FFT setting.

F. Additional Experiments and Analyses

In this section, we extend our evaluations of DC-Merge to additional checkpoints, larger task scales, and varying smoothing levels, demonstrating its strong robustness and scalability. Furthermore, we empirically verify that the two key modules in DC-Merge are both effective to preserve directional geometry of task vectors during merging and assess the generalization capability of DC-Merge on vision tasks. Finally, we provide a step-by-step analysis on computational complexity of our method.

F.1. Additional checkpoints

Ideally, the performance of a well-designed model merging method is robust to various checkpoints. For LoRA

setting, we present the performance of DC-Merge along with all baselines on ViT-B-32 checkpoints provided by KnOTS [58] in Table 9. DC-Merge still achieves the state-of-the-art performance, outperforming Iso-CTS by 2.7%. For FFT setting, we provide the results on 8-task benchmarks using checkpoints from Task Arithmetic [29] in Table 10, where the capability of DC-Merge is consistently superior (or comparable) to state-of-the-art methods.

Method	ViT-B-32	ViT-B-16	ViT-L-14
Zeroshot	48.3	55.5	64.7
Individual	90.5	92.6	94.2
Task Arithmetic	70.5	74.6	84.6
Fisher Merging	68.3	71.7	83.7
RegMean	71.8	76.6	82.2
PCB-Merging	76.3	81.5	87.5
TSV-M	83.8	87.2	91.5
Iso-CTS	<u>84.0</u>	<u>88.6</u>	92.9
DC-Merge	85.1	88.8	<u>92.7</u>

Table 10. Performance on 8-task benchmark using checkpoints provided by Task Arithmetic. We report average absolute accuracy.

F.2. Complementary Modules in DC-Merge

We conduct comparative experiments on vanilla Task Arithmetic (TA) [29] to demonstrate that both *energy smoothing* (ES) and *cover space merging* (CSM) are conducive to maintaining the directional consistency between merged vector and original task vectors. As illustrated in Figure 11, individually applying ES and CSM significantly boosts the projected $\text{DirSim}(\Delta \mathbf{W}_i, \Delta \tilde{\mathbf{W}}_i)$ of each task. Furthermore, by combining ES and CSM, we can achieve higher projected DirSim, indicating that the two key modules in DC-Merge are complementary.

We also present the task-wise projected CosSim of TA and DC-Merge on ViT-B-32 8-task benchmark in Figure 12. While TA exhibits higher projected CosSim on each individual task, its projected DirSim is much lower than DC-Merge across all tasks, indicating that TA fails to retain the directions of weaker but semantically important knowledge components within each task vector in the merging process and thereby suffers more performance degradation on each task than our method.

Method	Seen Tasks									Unseen Tasks				
	Cars	DTD	EuroSAT	GTSRB	MNIST	RESISC45	SUN397	SVHN	Average	CIFAR100	Flowers	Pets	STL10	Average
Individual	76.61	67.34	98.19	98.29	99.15	93.97	72.49	96.50	87.82	—	—	—	—	—
Zeroshot	59.49	44.15	44.30	32.27	47.94	60.25	63.21	31.42	47.88	64.61	66.18	87.74	96.77	78.83
Task Arithmetic	59.82	44.31	46.52	34.71	63.43	65.78	62.61	45.25	52.80	43.83	62.30	82.12	96.14	71.10
KnOTS-TIES	61.39	44.79	48.26	43.42	71.24	65.49	63.48	49.36	55.93	47.21	<u>62.11</u>	82.97	96.03	72.08
WUDI-Merging	60.89	45.64	51.82	39.90	66.96	67.40	63.32	46.12	55.25	46.81	62.05	82.46	<u>96.19</u>	71.89
TSV-M	<u>62.66</u>	46.54	55.44	46.88	75.86	69.37	<u>63.84</u>	50.66	58.91	51.13	61.87	84.33	96.27	<u>73.40</u>
Iso-CTS	60.72	52.34	64.26	<u>52.31</u>	<u>82.80</u>	<u>72.87</u>	63.68	<u>55.10</u>	<u>63.01</u>	<u>52.64</u>	60.14	<u>84.37</u>	95.67	73.21
DC-Merge	62.93	<u>50.00</u>	<u>61.15</u>	57.27	84.79	75.29	64.98	56.93	64.17	56.90	61.44	85.08	96.00	74.86

Table 11. Performance of eight seen tasks and four unseen tasks on ViT-B-32 in LoRA setting. We report absolute accuracy.

Method	Seen Tasks									Unseen Tasks				
	Cars	DTD	EuroSAT	GTSRB	MNIST	RESISC45	SUN397	SVHN	Average	CIFAR100	Flowers	Pets	STL10	Average
Individual	89.18	77.93	98.44	99.11	99.28	96.95	80.49	97.53	92.36	—	—	—	—	—
Zeroshot	77.94	55.64	64.68	50.68	76.16	71.33	68.34	58.58	65.42	76.13	79.43	93.32	99.39	87.07
Task Arithmetic	79.01	57.39	66.07	55.72	80.15	74.25	68.90	64.85	68.29	57.28	65.71	83.52	<u>98.95</u>	76.36
KnOTS-TIES	81.39	60.53	71.22	65.98	89.48	79.48	69.41	71.36	73.61	57.50	72.01	90.33	98.84	79.67
WUDI-Merging	79.62	58.19	66.56	57.26	85.03	76.51	68.59	66.49	69.78	60.81	69.65	87.26	99.03	79.19
TSV-M	<u>82.53</u>	62.61	75.74	75.12	88.74	82.52	<u>70.32</u>	74.61	76.52	60.79	73.42	91.55	98.86	<u>81.15</u>
Iso-CTS	81.74	<u>67.34</u>	<u>84.56</u>	<u>88.03</u>	<u>96.96</u>	<u>86.44</u>	69.53	<u>78.50</u>	<u>81.64</u>	<u>61.90</u>	<u>72.68</u>	<u>91.62</u>	98.31	81.13
DC-Merge	83.17	68.35	84.78	88.48	97.06	86.84	71.78	80.45	82.61	65.40	72.07	91.79	98.45	81.93

Table 12. Performance of eight seen tasks and four unseen tasks on ViT-L-14 in LoRA setting. We report absolute accuracy.

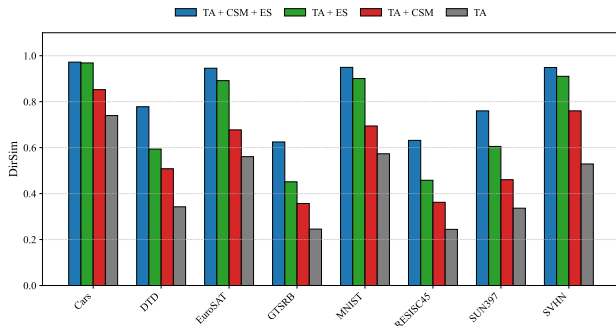


Figure 11. DirSim of TA w/o and w/ CSM and ES on ViT-B-32 8-task benchmark in LoRA setting. Both two modules in DC-Merge enhance DirSim($\Delta \mathbf{W}_i, \Delta \mathbf{W}_i$) of each task.



Figure 12. Projected CosSim of TA and DC-Merge (Ours) on ViT-B-32 8-task benchmark in LoRA setting.

F.3. Generalization Capability on Vision Tasks

We evaluate the generalization capability with four additional vision tasks on both ViT-B-32 and ViT-L-14 8-

task benchmarks in LoRA setting and the results are respectively presented in Figure 11 and 12. Unlike vision-language tasks, all baselines along with DC-Merge exhibit limited generalization capabilities due to the complexity of unseen tasks. Nevertheless, DC-Merge consistently outperforms other methods across both benchmarks.

F.4. Computational Complexity Analysis

We provide a comprehensive computational complexity analysis on DC-Merge in this subsection. For simplicity, we follow previous work [46] to consider a deep neutral network with L layers, and each layer consists of a single task vector $\Delta_i \in \mathbb{R}^{n \times n}$. Besides, we consider the traditional full SVD implementation with a complexity of $\mathcal{O}(n^3)$ [59] and compute r -rank SVD by first computing full SVD and then retaining the top- r singular values and their corresponding singular vectors. Supposing that the number of tasks is T , the step-by-step computational complexity of DC-Merge is:

- Compute r -rank SVD for each task vector Δ_i in each layer, with the total complexity of $\mathcal{O}(TLn^3)$;
- Perform energy smoothing for each task vector in each layer, with the total complexity of $\mathcal{O}(TLr)$;
- Reconstruct energy-balanced task vector Δ_i layer-wise, with the total complexity of $\mathcal{O}(TLn^2r)$;
- Whiten concatenated basis $\mathbf{U} \in \mathbb{R}^{n \times k}$ and $\mathbf{V} \in \mathbb{R}^{n \times k}$ respectively, with the total complexity of $\mathcal{O}(TL(nk^2 + k^3))$;
- Project each energy-balanced task vector Δ_i onto cover space for each layer, with the total complexity of $\mathcal{O}(TL(nk^2 + n^2k))$;
- Merge $\{\mathbf{M}\}_{i=1}^T$ via TA or TIES layer-wise, with the total complexity of $\mathcal{O}(TLk^2)$;

Method	Seen Tasks									Unseen Tasks				
	SciQA	Image	VQA	REC	OCR	VizWiz	Flickr	IconQA	Average	AVQA	Image-R	S2W	TabMWP	Average
Individual	83.74	96.02	67.58	43.40	65.50	64.80	57.29	75.54	69.23	—	—	—	—	—
Zeroshot	61.73	40.87	62.88	36.10	41.16	41.03	49.07	14.09	43.37	51.62	28.27	5.98	15.01	25.22
Multi Task	76.90	74.08	67.05	35.98	65.37	66.67	56.09	66.87	63.62	76.33	41.39	8.34	18.20	36.06
Task Arithmetic	71.94	57.49	67.06	38.90	62.87	44.80	49.20	39.21	53.93	74.78	37.37	7.52	13.57	33.31
DARE-Merging	71.59	57.25	66.26	39.38	62.56	44.93	49.13	39.59	53.84	73.75	37.67	7.56	13.62	33.15
TIES-Merging	71.49	55.88	66.73	39.67	65.12	44.35	47.06	34.46	53.09	73.43	38.44	7.47	13.23	33.14
PCB-Merging	71.10	57.82	67.59	38.22	<u>64.35</u>	44.58	48.90	37.01	53.70	74.57	36.28	7.84	15.44	33.53
TSV-M	75.66	78.61	59.48	41.92	41.71	41.19	52.23	45.97	54.60	77.29	42.72	11.12	13.80	36.23
WUDI-Merging	69.46	<u>78.40</u>	56.84	38.95	38.58	40.23	51.13	38.82	51.55	68.56	41.57	<u>11.16</u>	11.80	33.27
Iso-CTS	77.54	77.01	63.50	<u>45.72</u>	40.75	42.34	<u>54.74</u>	53.25	56.86	78.48	44.81	11.21	13.99	37.12
RobustMerge	<u>73.43</u>	<u>65.54</u>	<u>67.20</u>	44.80	62.97	46.61	52.80	<u>45.90</u>	<u>57.33</u>	<u>79.30</u>	<u>45.79</u>	9.23	<u>17.62</u>	<u>37.99</u>
DC-Merge	<u>77.05</u>	73.33	65.87	47.88	59.66	<u>45.23</u>	55.06	<u>52.92</u>	59.63	79.30	50.24	10.88	18.94	39.84

Table 13. Task-wise absolute accuracy results on MM-MergeBench [70], containing eight seen tasks (LoRA fine-tuned) and four unseen tasks. The best results are in **bold** and the second-best are underlined.

- Project the aggregated \widetilde{M} back to original parameter space for each layer, with the total complexity of $O(TL(nk^2 + n^2k))$.

As $r \leq \lfloor n/T \rfloor$, the overall computational complexity of DC-Merge is $O(TLn^3)$, which shares the same asymptotic computational complexity with the current state-of-art method TSV-M [20] and Iso-CTS [46]. In practical applications, we recommend using randomized SVD for low-rank approximation to enhance computational efficiency.

F.5. Performance under Increasing Task Scales

We compare the performance of DC-Merge with existing state-of-the-art methods under different task scales (i.e., the number of tasks). As illustrated in Table 14, with the increase of task scales, DC-Merge consistently achieves state-of-the-art overall performance while WUDI-Merging [6] suffers severe performance degradation. Furthermore, the superiority of DC-Merge becomes more significant as the task scale increases, which demonstrates the strong robustness of our method to task scales. The same trend can be observed in Table 2 on ViT-B-16 and ViT-L-14.

Method	8 tasks	14 tasks	20 tasks
Zeroshot	48.26	57.21	56.10
Individual	92.83	90.88	91.37
TSV-M	85.86 _(92.31)	80.06 _(87.88)	77.07 _(84.29)
WUDI-Merging	86.47 _(93.05)	78.87 _(86.71)	69.90 _(76.71)
Iso-CTS	86.20 _(91.78)	81.71 _(89.70)	78.05 _(85.48)
DC-Merge	87.05 _(93.55)	82.52 _(90.62)	80.58 _(88.18)

Table 14. Performance on ViT-B-32 benchmarks in FFT setting. We report average absolute accuracy and subscript (in parentheses) is the average normalized accuracy.

F.6. Performance on Varying Smoothing Levels

In Appendix E.4, we propose a flexible linear smoothing strategy that controls the energy flattening degree through a single hyperparameter ρ . Higher ρ indicates more skewness

in energy distribution. We present the performance of DC-Merge on 8-task benchmark in LoRA setting with varying ρ values in Figure 13. Notably, DC-Merge exhibits strong robustness to the choice of ρ , achieving state-of-the-art performance across a wide range of smoothing levels.

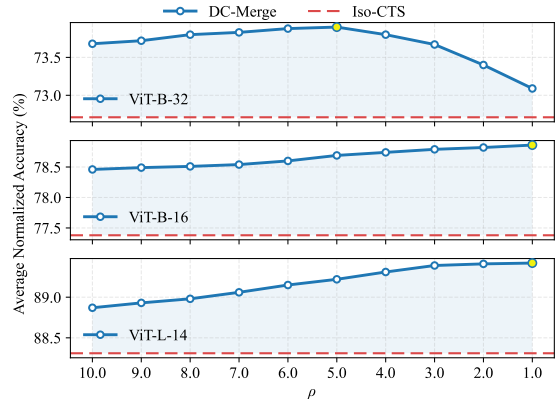
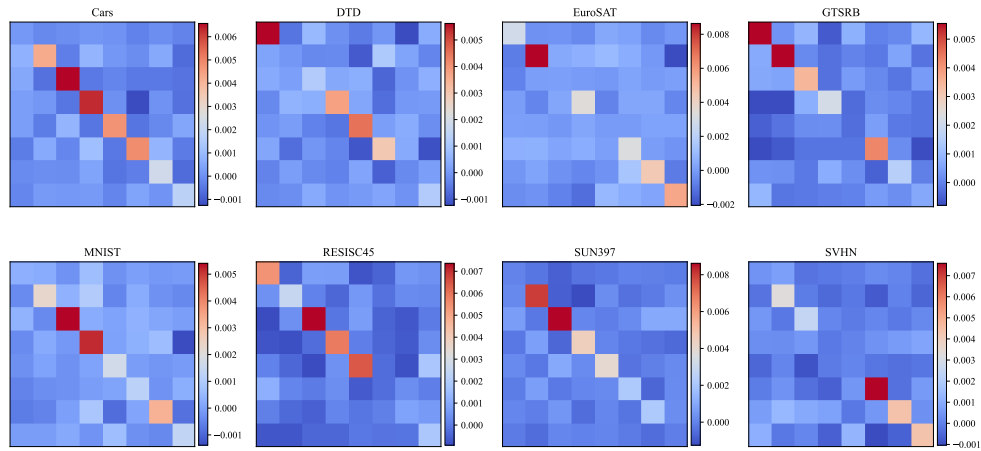


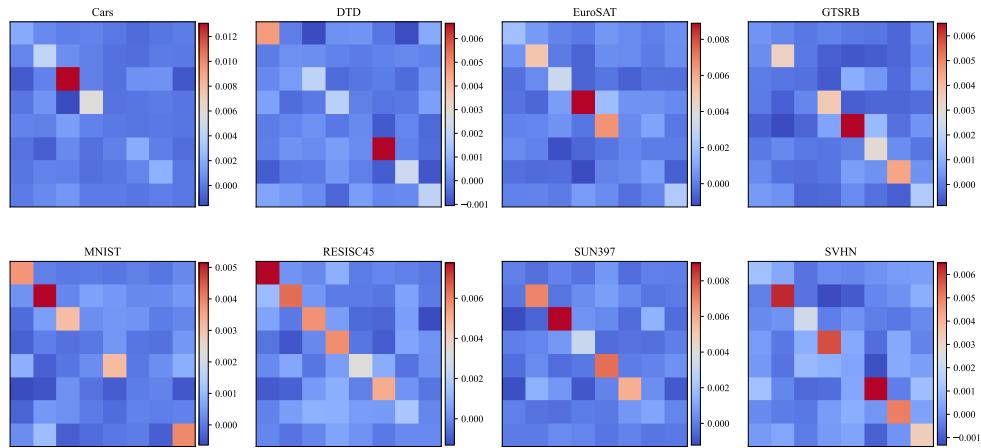
Figure 13. Performance of DC-Merge with respect to various smoothing levels in linear smoothing. We report average normalized accuracy. When $\rho = 1.0$, linear smoothing reduces to simple averaging.

F.7. Scaling to Vision-Language Models

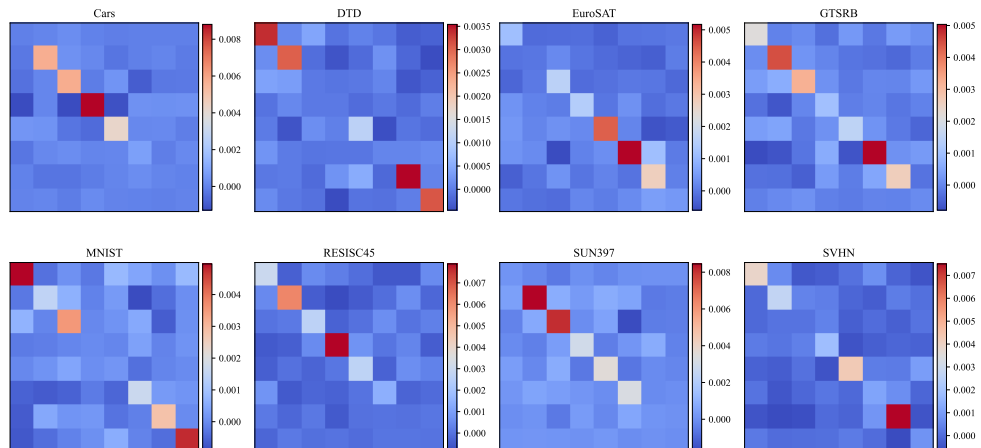
Table 13 presents task-wise performance of existing baselines in Table 3, along with additional state-of-the-art methods TSV-M [20], WUDI-Merging [6] and Iso-CTS [46]. While some state-of-the-art approaches struggle to maintain strong performance, DC-Merge consistently achieves superior results on both seen and unseen tasks, demonstrating that the effectiveness of DC-Merge is not limited to vision models, but naturally extends to large-scale multi-modal models.



(a) encoder.layers.2.self_attn.q_proj.weight

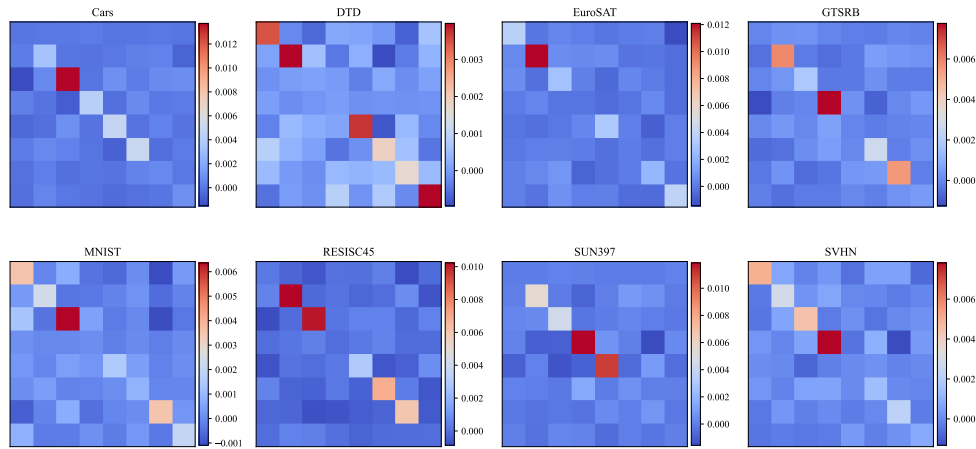


(b) encoder.layers.2.self_attn.k_proj.weight

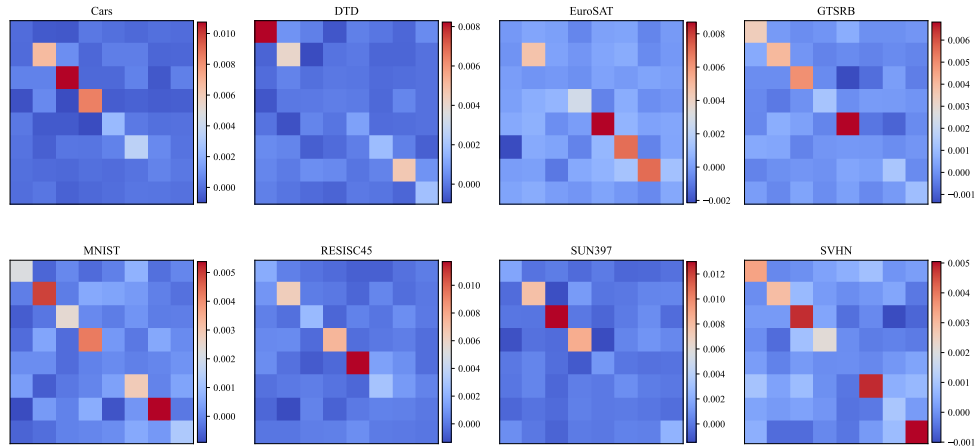


(c) encoder.layers.2.self_attn.v_proj.weight

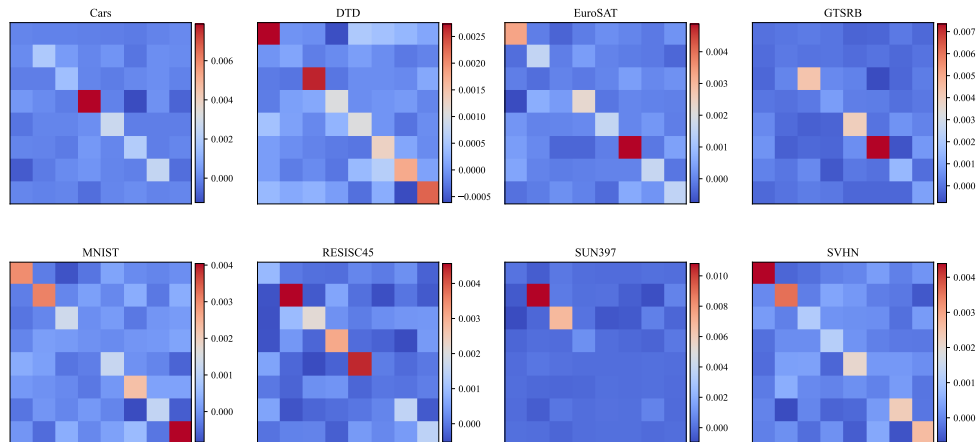
Figure 14. Visualization of task representations in the shared cover space. We apply block-diagonal masks to eliminate cross-task directional interference (off-diagonal blocks) when projecting the aggregated task representations back to parameter space, where the mask size implicitly balances directional preservation and task fidelity. (Continued on next page)



(d) encoder.layers.5.self_attn.q_proj.weight

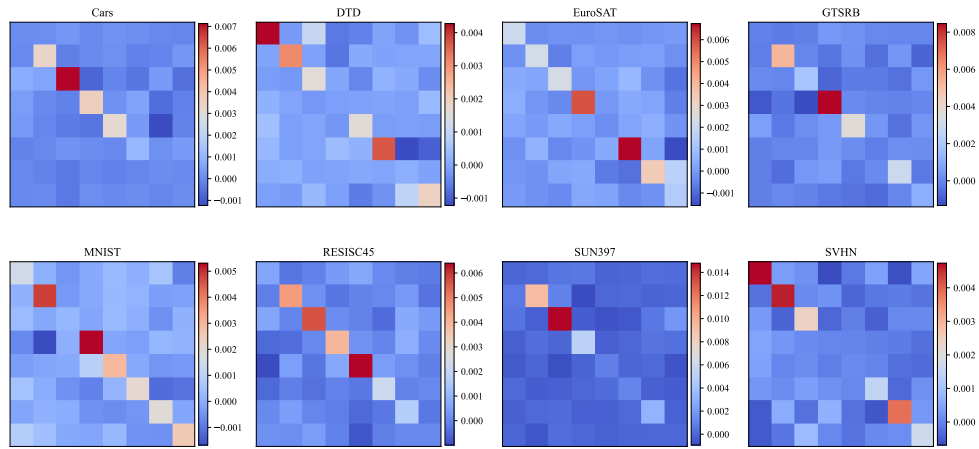


(e) encoder.layers.5.self_attn.k_proj.weight

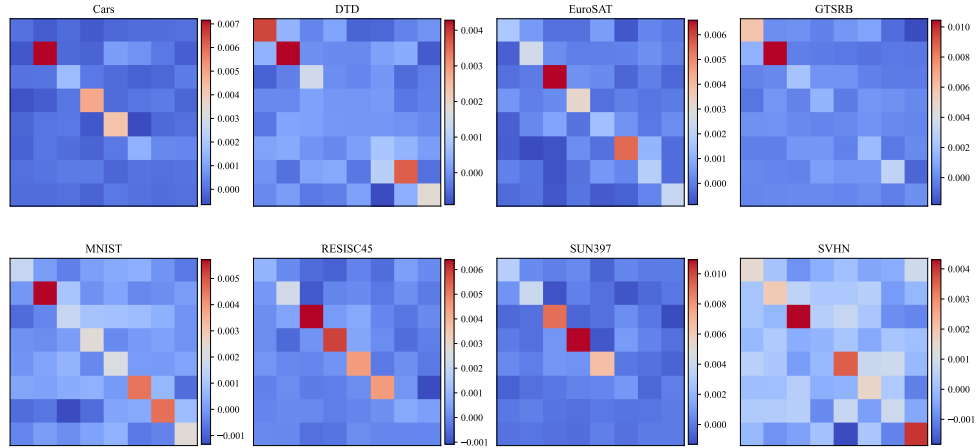


(f) encoder.layers.5.self_attn.v_proj.weight

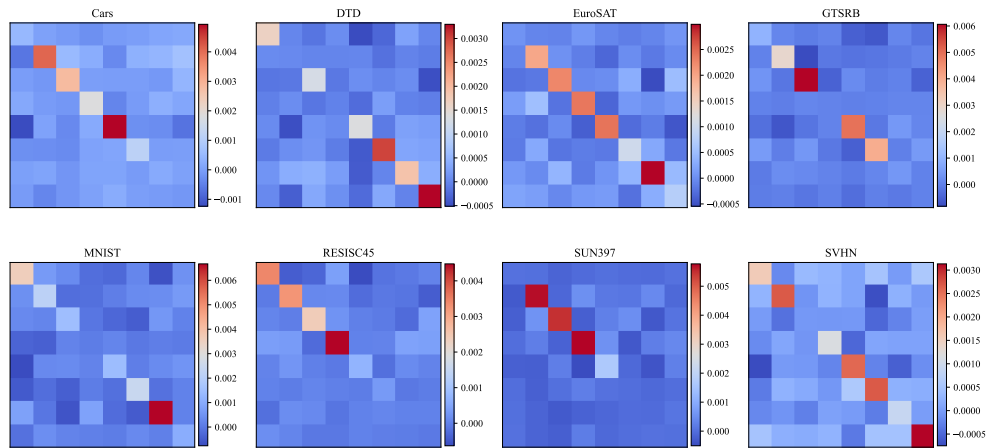
Figure 14. Visualization of task representations in the shared cover space. We apply block-diagonal masks to eliminate cross-task directional interference (off-diagonal blocks) when projecting the aggregated task representations back to parameter space, where the mask size implicitly balances directional preservation and task fidelity. (Continued on next page)



(g) encoder.layers.8.self_attn.q_proj.weight



(h) encoder.layers.8.self_attn.k_proj.weight



(i) encoder.layers.8.self_attn.v_proj.weight

Figure 14. Visualization of task representations in the shared cover space. We apply block-diagonal masks to eliminate cross-task directional interference (off-diagonal blocks) when projecting the aggregated task representations back to parameter space, where the mask size implicitly balances directional preservation and task fidelity.



Figure 15. Layer-wise visualization of projected DirSim with each individual task. The results are based on ViT-B-32 8-task benchmark in LoRA setting. DC-Merge consistently exhibits higher projected DirSim with task vectors across all layers.