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*

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

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

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

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

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

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

Therefore, we have:

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

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

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

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

Conclusion. *Combining the numerator and denominator gives*

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

which proves the proposition.

C. CosSim as Expressiveness via Coefficient Projections

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

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

Proposition 2 *With the notation above, the following hold:*
(A) Least-squares fitting $\leftrightarrow$ maximizing ℓ_2-coefficients.

$$\min_{\boldsymbol{U},\boldsymbol{V}} \min_{\boldsymbol{\sigma} \in \mathbb{R}^k} \Big\| \boldsymbol{\Delta} - \boldsymbol{U} \, \mathrm{diag}(\boldsymbol{\sigma}) \, \boldsymbol{V}^\top \Big\|_F \leftrightarrow \max_{\boldsymbol{U},\boldsymbol{V}} \|\boldsymbol{\sigma}(\boldsymbol{U}, \boldsymbol{V}, \boldsymbol{\Delta})\|_2 \tag{18}$$

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

$$\max_{\boldsymbol{U},\boldsymbol{V}} \mathrm{CosSim}\left(\boldsymbol{\Delta}, \boldsymbol{U}\boldsymbol{V}^\top\right) \leftrightarrow \max_{\boldsymbol{U},\boldsymbol{V}} \|\boldsymbol{\sigma}(\boldsymbol{U}, \boldsymbol{V}, \boldsymbol{\Delta})\|_1 \tag{19}$$

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

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

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

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

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

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

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

Therefore

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

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

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

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

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

$$\begin{aligned} \left\langle \boldsymbol{\Delta},\, \boldsymbol{U}\boldsymbol{V}^\top \right\rangle &= \mathrm{tr}(\boldsymbol{U}^\top \boldsymbol{\Delta}\, \boldsymbol{V}) = \sum_{j=1}^k (\boldsymbol{U}^\top \boldsymbol{\Delta}\, \boldsymbol{V})_{jj} \\ &= \sum_{j=1}^k \sigma_j(\boldsymbol{U}, \boldsymbol{V}, \boldsymbol{\Delta}) = \left\|\boldsymbol{\sigma}(\boldsymbol{U}, \boldsymbol{V}, \boldsymbol{\Delta})\right\|_1 \end{aligned} \tag{21}$$

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

$$\begin{aligned} \mathrm{CosSim}\Big(\boldsymbol{\Delta},\, \boldsymbol{U}\boldsymbol{V}^\top\Big) &= \frac{\langle \boldsymbol{\Delta},\, \boldsymbol{U}\boldsymbol{V}^\top \rangle}{\|\boldsymbol{\Delta}\|_F\, \|\boldsymbol{U}\boldsymbol{V}^\top\|_F} \\ &= \frac{\left\|\boldsymbol{\sigma}(\boldsymbol{U}, \boldsymbol{V}, \boldsymbol{\Delta})\right\|_1}{\|\boldsymbol{\Delta}\|_F\, \sqrt{k}}. \end{aligned} \tag{22}$$

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

Interpretation. Each entry of $\boldsymbol{\sigma}(\boldsymbol{U}, \boldsymbol{V}, \boldsymbol{\Delta}) = \mathrm{diag}(\boldsymbol{U}^\top \boldsymbol{\Delta} \boldsymbol{V})$ is a *bi-directional projection* of $\boldsymbol{\Delta}$ onto the dyad $\boldsymbol{u}_j \boldsymbol{v}_j^\top$ (input- and output-side alignments jointly). Part (A) shows that the best reconstruction by dyads in the basis $(\boldsymbol{U}, \boldsymbol{V})$ 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 $\boldsymbol{U}\boldsymbol{V}^\top$ is (up to a constant)

the ℓ_1 sum of these projections. Consequently, a higher cosine indicates stronger ability of the basis $(\boldsymbol{U}, \boldsymbol{V})$ to *express* (encode) the knowledge contained in $\boldsymbol{\Delta}$—exactly mirroring the main-text view where $\boldsymbol{R}(s,t)$ 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:

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

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

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

$$\begin{aligned}
\left\|\widetilde{\boldsymbol{U}}^\top\boldsymbol{U}\right\|_F^2 &\ge \left\|\mathrm{diag}\left(\widetilde{\boldsymbol{U}}^\top\boldsymbol{U}\right)\right\|_2^2 \ge \frac{1}{k}\mathrm{tr}\left(\widetilde{\boldsymbol{U}}^\top\boldsymbol{U}\right)^2, \\
\left\|\widetilde{\boldsymbol{V}}^\top\boldsymbol{V}\right\|_F^2 &\ge \left\|\mathrm{diag}\left(\widetilde{\boldsymbol{V}}^\top\boldsymbol{V}\right)\right\|_2^2 \ge \frac{1}{k}\mathrm{tr}\left(\widetilde{\boldsymbol{V}}^\top\boldsymbol{V}\right)^2,
\end{aligned} \tag{24}$$

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

Finally, whitening [54] naturally achieves this objective:

$$\begin{aligned}
\widetilde{\boldsymbol{U}} &= \arg\max_{\widetilde{\boldsymbol{U}}^\top\widetilde{\boldsymbol{U}}=\boldsymbol{I}}\left\langle\boldsymbol{U},\,\widetilde{\boldsymbol{U}}\right\rangle, \\
\widetilde{\boldsymbol{V}} &= \arg\max_{\widetilde{\boldsymbol{V}}^\top\widetilde{\boldsymbol{V}}=\boldsymbol{I}}\left\langle\boldsymbol{V},\,\widetilde{\boldsymbol{V}}\right\rangle.
\end{aligned} \tag{25}$$

This whitening-based alignment maximizes $\mathrm{tr}\left(\widetilde{\boldsymbol{U}}^\top\boldsymbol{U}\right)$ and $\mathrm{tr}\left(\widetilde{\boldsymbol{V}}^\top\boldsymbol{V}\right)$, 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 $\widetilde{\boldsymbol{U}},\widetilde{\boldsymbol{V}}\in\mathbb{R}^{d\times k}$ can be expressed as

$$\widetilde{\boldsymbol{U}} = \exp(\boldsymbol{A})\,\widetilde{\boldsymbol{U}}_0, \quad \widetilde{\boldsymbol{V}} = \exp(\boldsymbol{B})\,\widetilde{\boldsymbol{V}}_0 \tag{26}$$

where $\boldsymbol{A},\boldsymbol{B}\in\mathbb{R}^{d\times d}$ is a skew-symmetric matrix ($\boldsymbol{A}^\top = -\boldsymbol{A}, \boldsymbol{B}^\top = -\boldsymbol{B}$) and $\widetilde{\boldsymbol{U}}_0,\widetilde{\boldsymbol{V}}_0$ is an initial orthonormal basis. This parameterization utilizes the isomorphic properties of Lie groups and rotation operations [35] and ensures that $\widetilde{\boldsymbol{U}},\widetilde{\boldsymbol{V}}$ 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 $\left(\widetilde{\boldsymbol{U}}_0,\widetilde{\boldsymbol{V}}_0\right)$ 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

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

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 $\boldsymbol{\mathcal{M}}$ is applied:

$$\boldsymbol{\Delta}_M = \widetilde{\boldsymbol{U}}\left(\hat{\boldsymbol{M}} \odot \boldsymbol{\mathcal{M}}\right)\widetilde{\boldsymbol{V}}^\top.$$

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

$$\boldsymbol{\Delta}_i \approx \widetilde{\boldsymbol{U}}\,\mathrm{diag}(\boldsymbol{\sigma}_i)\,\widetilde{\boldsymbol{V}}^\top, \tag{27}$$

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

$$\sum_{i=1}^T \boldsymbol{\Delta}_i \approx \widetilde{\boldsymbol{U}}\left(\sum_{i=1}^T \mathrm{diag}(\boldsymbol{\sigma}_i)\right)\widetilde{\boldsymbol{V}}^\top, \tag{28}$$

which clearly preserves the original knowledge directions $\widetilde{\boldsymbol{U}}$ and $\widetilde{\boldsymbol{V}}$ and $\boldsymbol{\sigma}_i$ is the corresponding optimal coefficients under $\widetilde{\boldsymbol{U}}$ and $\widetilde{\boldsymbol{V}}$ for $\boldsymbol{\Delta}_i$. 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, $\boldsymbol{\mathcal{M}}$ implicitly balances between directional consistency and task-specific fidelity.

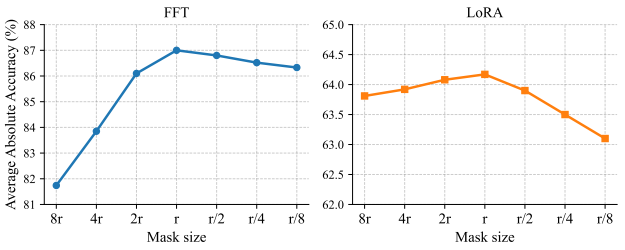


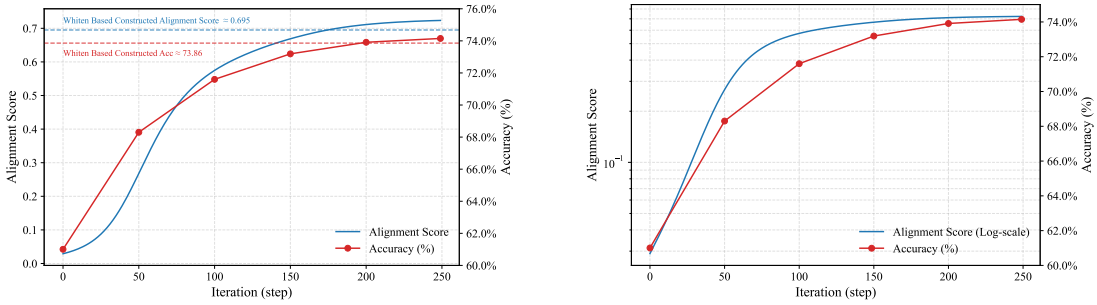
Figure 5. The average absolute accuracy of DC-Merge with mask size $\{8r, 4r, 2r, r, r/2, r/4, r/8\}$ in FFT (left) and LoRA (right) settings. Performance peaks near r for FFT and shows a milder variation for LoRA (slightly improves till r then decreases), indicating that moderate masks retain performance 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:

$$u_1 = v_1 = [1,\, 0]^\top$$
$$u_2 = v_2 = [0.1104,\, 0.9939]^\top.$$

When directly merging in the parameter space,

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



(a) Linear Scale (b) Semi-log scale

Figure 4. **Optimization trajectory of the cover basis.** We start from a poor initialization $\left(\widetilde{\boldsymbol{U}}_0, \widetilde{\boldsymbol{V}}_0\right)$ 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 geometric of task knowledge.

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

$$\widetilde{\boldsymbol{U}} = \widetilde{\boldsymbol{V}} = [u_1', u_2'],$$

where $u_1' = [0.9985, -0.0533]^\top, u_2' = [0.0533, 0.9985]$. When only using a diagonal mask $\mathcal{M}$, the resulting directional equal to $\widetilde{\boldsymbol{U}}, \widetilde{\boldsymbol{V}}$ 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.

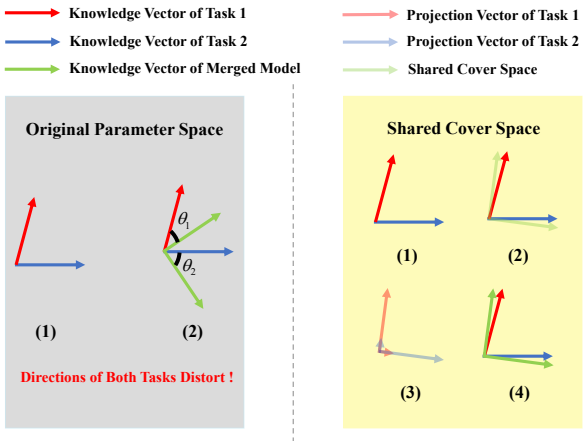


Figure 6. Illustration of mitigating directional shift when merging in the shared cover space. We take u_1 and u_2 as example. Left: (1) The direction of u_1 and u_2. (2) Directly merging in the original parameter space yields u_1' and u_2', causing severe directional deviation. Right: (1) The direction 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.

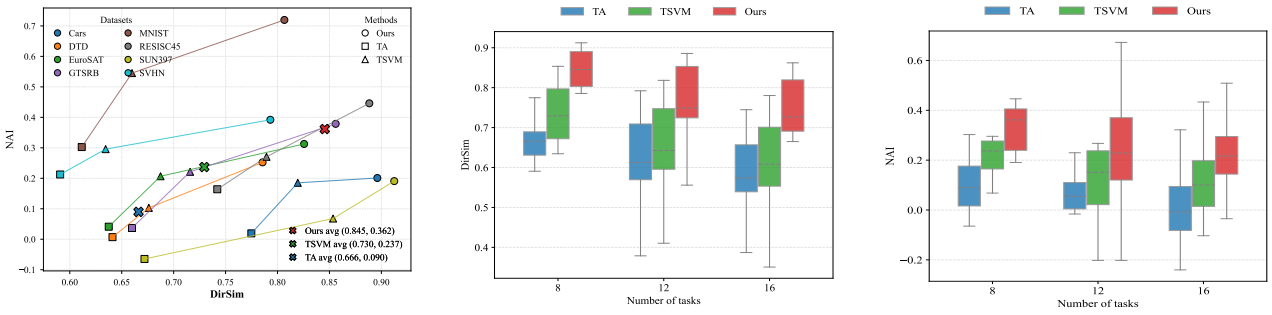
E. Experimental Details

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], 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



(a) (b) (c)

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*:

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

where $acc(\theta_i, i)$ denotes the absolute accuracy of the model fine-tuned on task i, $acc(\theta_M, i)$ 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

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

where $acc(\theta_0, i)$ 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 $\Delta\widetilde{\boldsymbol{W}}$ with the pretrained weights $\boldsymbol{W}_0$ to obtain the merged model by:

$$\widetilde{\boldsymbol{W}} = \boldsymbol{W}_0 + \alpha\Delta\widetilde{\boldsymbol{W}}, \tag{31}$$

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 $\lfloor \frac{min(m,n)}{T} \rfloor$ with each task vector $\boldsymbol{\Delta}_i \in \mathbb{R}^{m\times n}$.

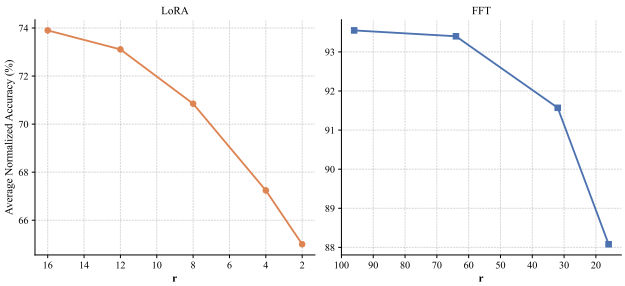


Figure 8. Impact of hyperparameter r in LoRA and FFT settings. We report average normalized accuracy of DC-Merge on `ViT-B-32` 8-task benchmark. The empirical results align with our motivation to preserve the directional geometry of each task vector after merging as much as possible.

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 codebase. We assume that ranks of the merged vectors $\Delta_{\text{TSV-M}}$ and $\Delta_{\text{Iso-CTS}}$ should be $T \times$ 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 $\boldsymbol{U}_i, \boldsymbol{V}_i$ fixed and perturbing only the energy distribution $\boldsymbol{\sigma}_i$ to $\widehat{\boldsymbol{\sigma}}_i$, we have

$$\text{CosSim}(\boldsymbol{\Delta}_i, \widehat{\boldsymbol{\Delta}}_i) = \text{CosSim}(\boldsymbol{\sigma}_i, \widehat{\boldsymbol{\sigma}}_i), \tag{32}$$

where $\widehat{\boldsymbol{\Delta}}_i = \boldsymbol{U}_i \,\text{diag}(\widehat{\boldsymbol{\sigma}}_i)\, \boldsymbol{V}_i^\top$. Hence, to obtain a controllable energy distribution perturbation, it suffices to construct $\widehat{\boldsymbol{\sigma}}_i$ such that $\text{CosSim}(\boldsymbol{\sigma}_i, \widehat{\boldsymbol{\sigma}}_i) = p$, where $p \in [0, 1]$ prescribes the perturbation degree.

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

$$\boldsymbol{s}^\perp = \frac{\overline{\boldsymbol{\sigma}} - \left(\boldsymbol{s}^\top \overline{\boldsymbol{\sigma}}\right) \boldsymbol{s}}{\left\|\overline{\boldsymbol{\sigma}} - \left(\boldsymbol{s}^\top \overline{\boldsymbol{\sigma}}\right) \boldsymbol{s}\right\|_2} \tag{33}$$

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

$$\widehat{\boldsymbol{s}} = p\,\boldsymbol{s} + \sqrt{1 - p^2}\,\boldsymbol{s}^\perp, \qquad \widehat{\boldsymbol{\sigma}} = \|\boldsymbol{\sigma}\|_2\, \widehat{\boldsymbol{s}}. \tag{34}$$

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

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

Hence Eq. (34) yields an energy distribution perturbation with prescribed cosine p while keeping directions $(\boldsymbol{U}_i, \boldsymbol{V}_i)$ 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 $\widehat{\boldsymbol{\Delta}}_i$ satisfies $\text{DirSim}(\boldsymbol{\Delta}_i, \widehat{\boldsymbol{\Delta}}_i) = p$. Specifically, we perturb the orthogonal directions $\boldsymbol{U}_i$ and $\boldsymbol{V}_i$ by mixing them with randomly sampled bases from their orthogonal complements.

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

$$\begin{aligned} \widehat{\boldsymbol{U}}_i(p) &= \sqrt{p}\,\boldsymbol{U}_i + \sqrt{1-p}\,\widehat{\boldsymbol{U}}_i, \\ \widehat{\boldsymbol{V}}_i(p) &= \sqrt{p}\,\boldsymbol{V}_i + \sqrt{1-p}\,\widehat{\boldsymbol{V}}_i, \end{aligned} \tag{36}$$

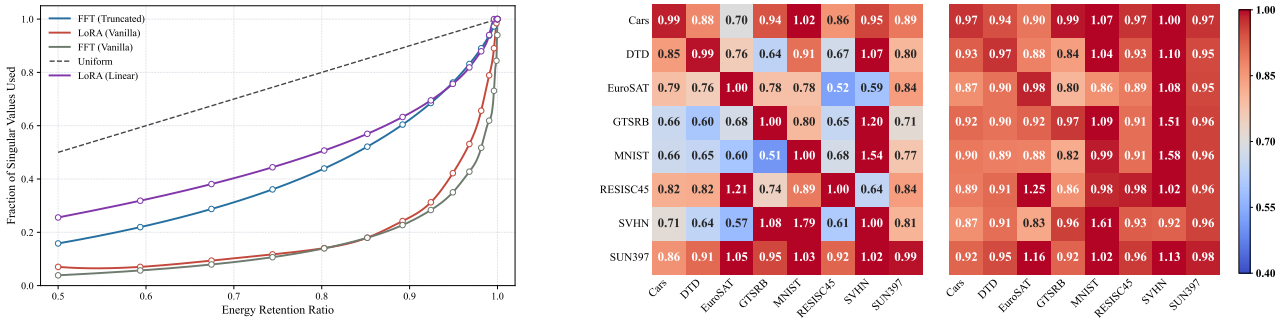
which satisfy the orthonormality $\widehat{\boldsymbol{U}}_i(p)^\top \widehat{\boldsymbol{U}}_i(p) = \widehat{\boldsymbol{V}}_i(p)^\top \widehat{\boldsymbol{V}}_i(p) = \boldsymbol{I}$. The perturbed task vector $\widehat{\boldsymbol{\Delta}}_i$ with directional geometry $(\widehat{\boldsymbol{U}}_i(p),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

$$\begin{aligned} \text{DirSim}(\boldsymbol{\Delta}_i, \widehat{\boldsymbol{\Delta}}_i) &= \frac{\text{tr}\left(\boldsymbol{U}_i^\top \widehat{\boldsymbol{U}}_i(p) \widehat{\boldsymbol{V}}_i(p)^\top \boldsymbol{V}_i\right)}{r} \\ &= \frac{\text{tr}\,(p \cdot \boldsymbol{I}_{r\times r})}{r} = p, \end{aligned} \tag{37}$$

where the second equality holds because $\boldsymbol{U}_i^\top \widehat{\boldsymbol{U}}_i(p) = \widehat{\boldsymbol{V}}_i(p)^\top \boldsymbol{V}_i = \sqrt{p} \cdot \boldsymbol{I}_{r\times r}$.

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.



(a) (b)

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 $\mathbf{\Sigma}_i^{(r)}$. 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:

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

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.,

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

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

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

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 $\overline{\mathbf{\Delta}}_i$ 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:

$$\overline{\mathbf{\Sigma}}_i^{(r)} = \tau \mathbf{\Sigma}_{i,\text{orig}}^{(r)} + (1-\tau) \mathbf{\Sigma}_{i,\text{avg}}^{(r)} \tag{41}$$

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

Method	Datasets								
	Cars	DTD	EuroSAT	GTSRB	MNIST	RESISC45	SUN397	SVHN	**Average**
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**	69.4	**82.3**	**99.0**	53.2	75.1
DC-Merge	**90.9**	79.6	**65.5**	54.0	**92.6**	75.0	98.2	**66.8**	**77.8**

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

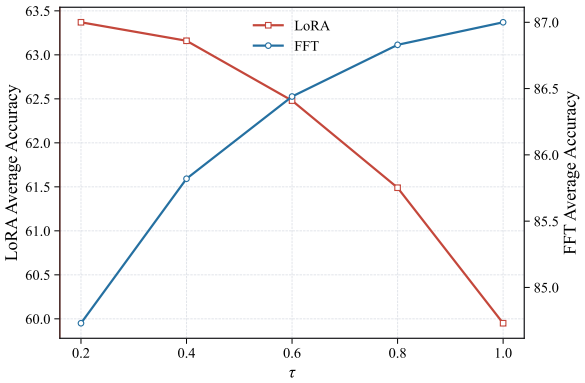


Figure 10. The relationship between average absolute accuracy of DC-Merge in LoRA/FFT setting and interpolation coefficient τ.

energy distribution of each task vector tends to be smoother. We notice a consistent performance improvement as each $\overline{\mathbf{\Sigma}}_i^{(r)}$ 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 $\overline{\mathbf{\Delta}}_i$ 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	84.0	88.6	**92.9**
DC-Merge	**85.1**	**88.8**	92.7

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 Arithemetic (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 $\mathrm{DirSim}(\Delta \boldsymbol{W}_i, \Delta \widetilde{\boldsymbol{W}}_i)$ 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	62.11	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	96.19	71.89
TSV-M	62.66	46.54	55.44	46.88	75.86	69.37	63.84	50.66	58.91	51.13	61.87	84.33	**96.27**	73.40
Iso-CTS	60.72	**52.34**	**64.26**	52.31	82.80	72.87	63.68	55.10	63.01	52.64	60.14	84.37	95.67	73.21
DC-Merge	**62.93**	50.00	61.15	**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	98.95	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	82.53	62.61	75.74	75.12	88.74	82.52	70.32	74.61	76.52	60.79	**73.42**	91.55	98.86	81.15
Iso-CTS	81.74	67.34	84.56	88.03	96.96	86.44	69.53	78.50	81.64	61.90	72.68	91.62	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.

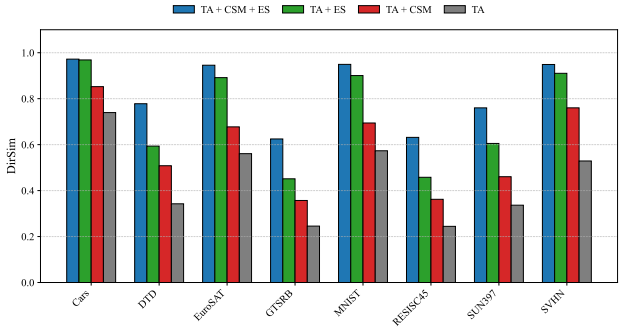


Figure 11. DirSim of TA w/o and w/ CSM and ES on ViT-B-32 8-task benchmark in LoRA setting. Both two modules in DC-Merge enhance $\text{DirSim}(\Delta \boldsymbol{W}_i, \Delta \widetilde{\boldsymbol{W}}_i)$ of each task.

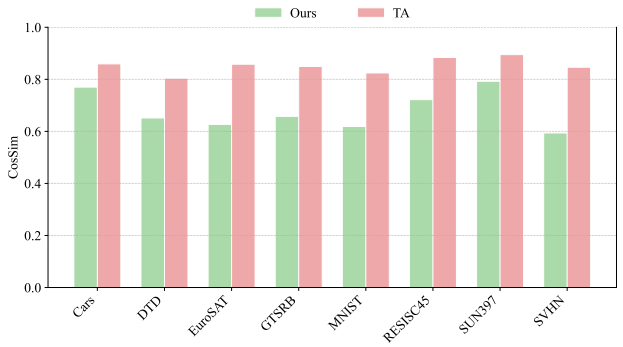


Figure 12. Projected CosSim of TA and DC-Merge (Ours) on ViT-B-32 8-task benchmark in LoRA setting.

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 $\boldsymbol{\Delta}_i \in \mathbb{R}^{n \times n}$. 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 $\boldsymbol{\Delta}_i$ 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 $\boldsymbol{\Delta}_i$ layer-wise, with the total complexity of $\mathcal{O}(TLn^2r)$;
- Whiten concatenated basis $\boldsymbol{U} \in \mathbb{R}^{n \times k}$ and $\boldsymbol{V} \in \mathbb{R}^{n \times k}$ respectively, with the total complexity of $\mathcal{O}(TL(nk^2 + k^3))$;
- Project each energy-balanced task vector $\boldsymbol{\Delta}_i$ onto cover space for each layer, with the total complexity of $\mathcal{O}(TL(nk^2 + n^2k))$;
- Merge $\{\boldsymbol{M}\}_{i=1}^{T}$ 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	64.35	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	78.40	56.84	38.95	38.58	40.23	51.13	38.82	51.55	68.56	41.57	11.16	11.80	33.27
Iso-CTS	**77.54**	77.01	63.50	45.72	40.75	42.34	54.74	**53.25**	56.86	78.48	44.81	**11.21**	13.99	37.12
RobustMerge	73.43	65.54	67.20	44.80	62.97	**46.61**	52.80	45.90	57.33	79.30	45.79	9.23	17.62	37.99
DC-Merge	77.05	73.33	65.87	**47.88**	59.66	45.23	**55.06**	52.92	**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 $\mathcal{O}(TL(nk^2 + n^2k))$.

As $r \leq \lfloor n/T \rfloor$, the overall computational complexity of DC-Merge is $\mathcal{O}(TLn^3)$, 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	$\mathbf{87.05}_{(\mathbf{93.55})}$	$\mathbf{82.52}_{(\mathbf{90.62})}$	$\mathbf{80.58}_{(\mathbf{88.18})}$

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.

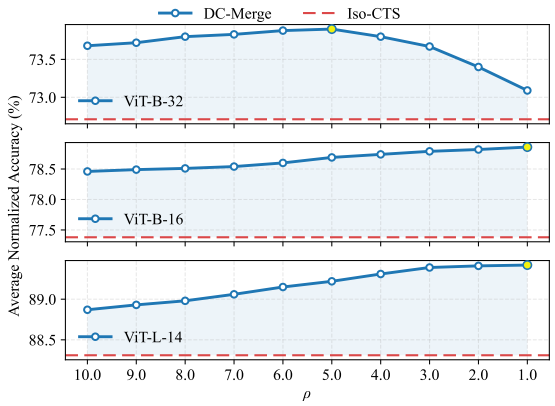
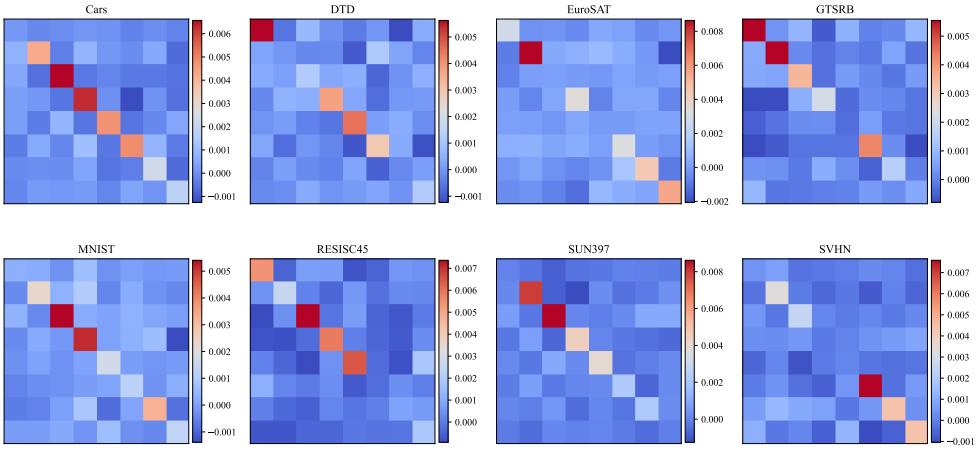


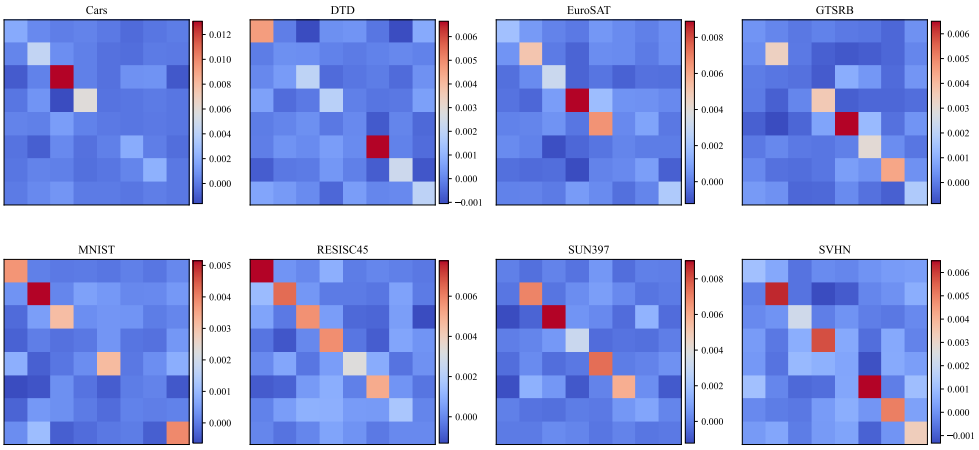
Figure 13. Performance of DC-Merge with respect to various smoothing levels in linear smoothing. We report average normalized accuracy. When $\rho = 1.0$, linear smoothing reduces to simple averaging.

F.7. Scaling to Vision-Language Models

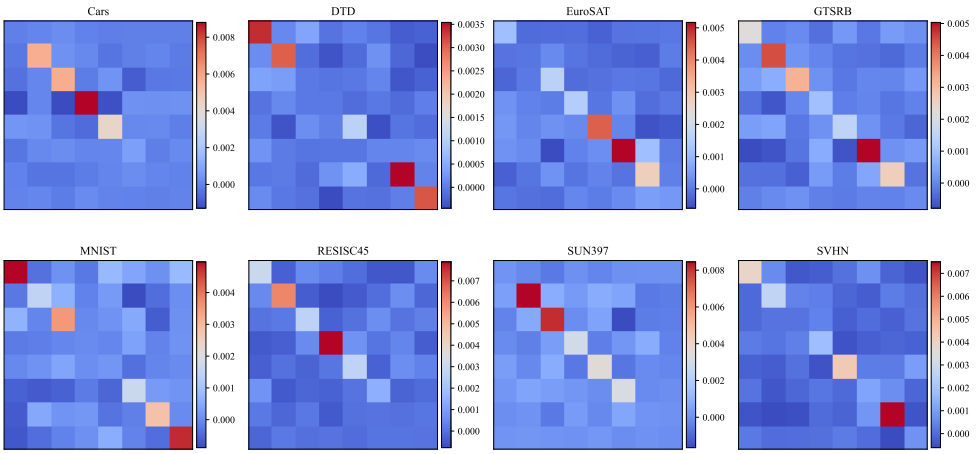
Table 13 presents task-wise performance of existing baselines in Table 3, along with additional state-of-the-art methods TSV-M [20], WUDI-Merging [6] and Iso-CTS [46]. While some state-of-the-art approaches struggle to maintain strong performance, DC-Merge consistently achieves superior results on both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



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

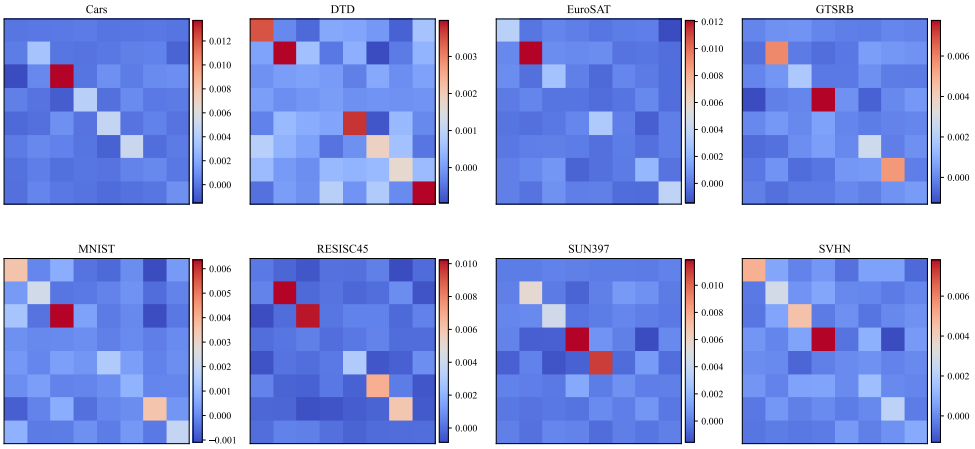


(b) encoder.layers.2.self_attn.k_proj.weight

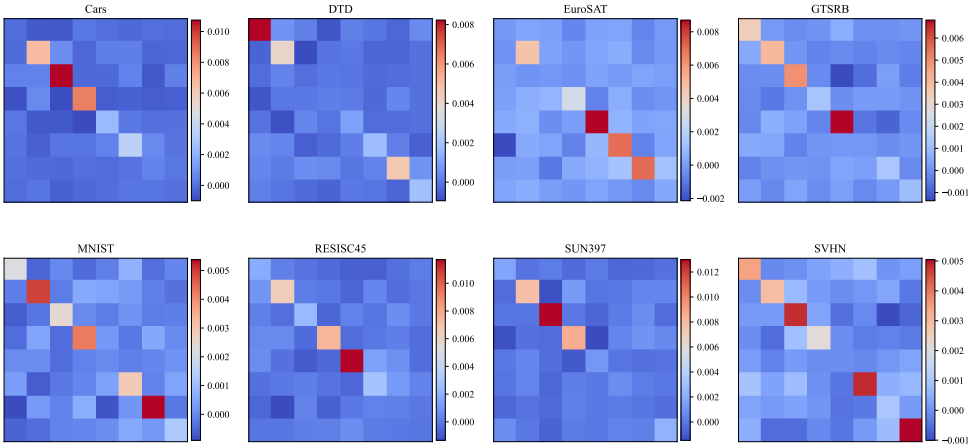


(c) encoder.layers.2.self_attn.v_proj.weight

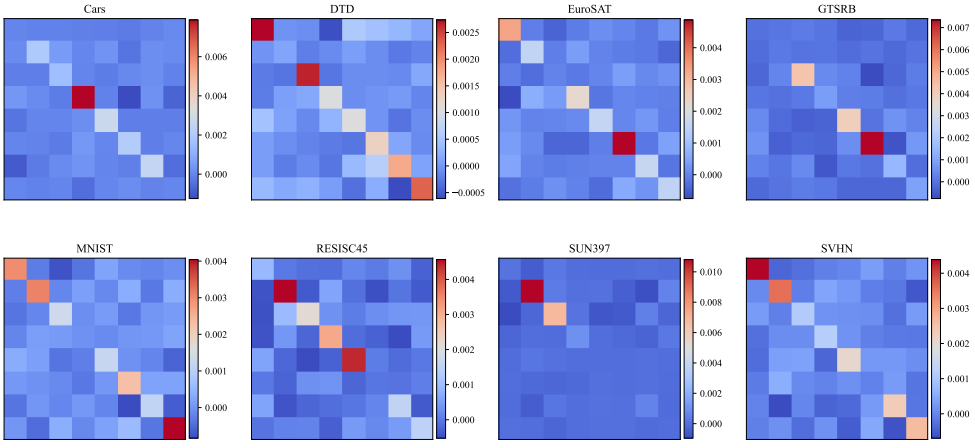
Figure 14. Visualization of task representations in the shared cover space. We apply block-diagonal masks to eliminate cross-task directional interference (off-diagonal blocks) when projecting the aggregated task representations back to parameter space, where the mask size implicitly balances directional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



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

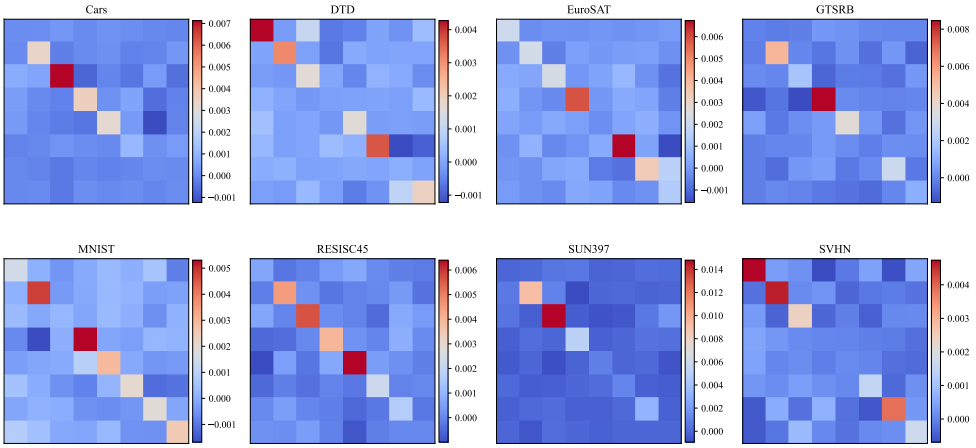


(e) encoder.layers.5.self_attn.k_proj.weight

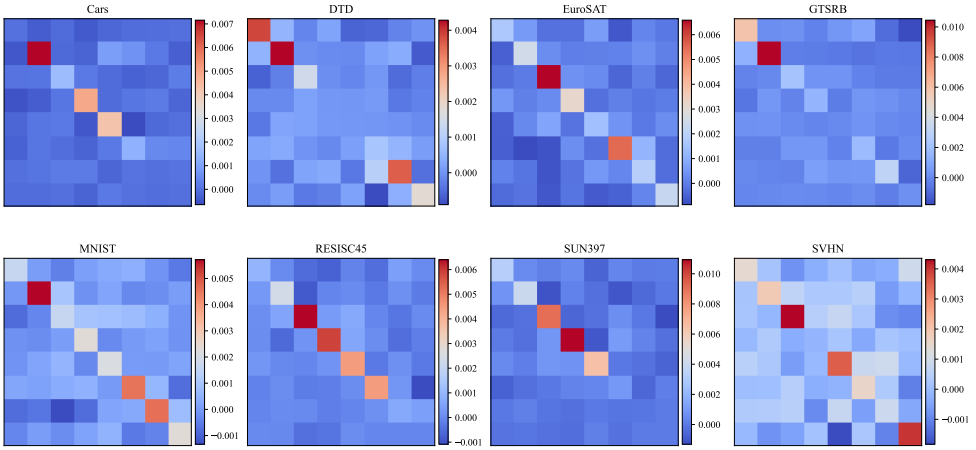


(f) encoder.layers.5.self_attn.v_proj.weight

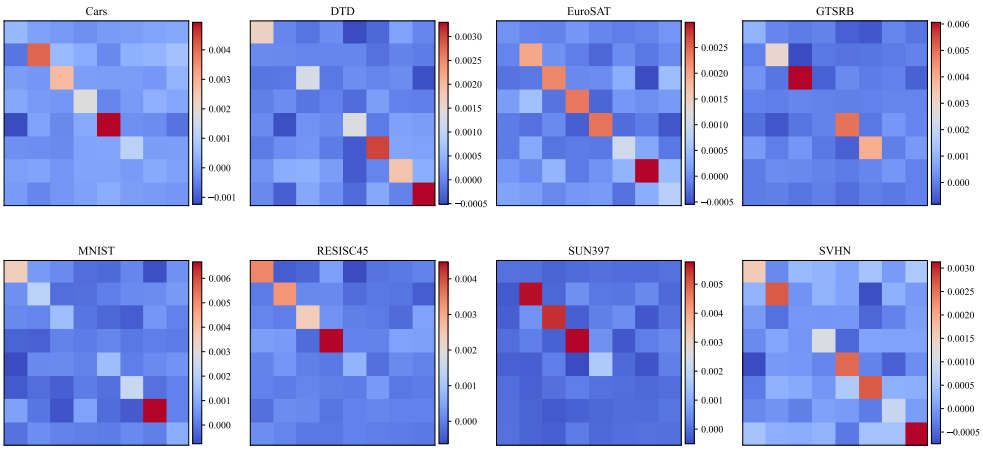
Figure 14. Visualization of task representations in the shared cover space. We apply block-diagonal masks to eliminate cross-task directional interference (off-diagonal blocks) when projecting the aggregated task representations back to parameter space, where the mask size implicitly balances directional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