

Decouple and Orthogonalize: A Data-Free Framework for LoRA Merging

Shenghe Zheng^{*1,2}, Hongzhi Wang^{†1}, Chenyu Huang⁴, Xiaohui Wang⁴

Jiayuan Fan⁴, Shuyue Hu², Peng Ye^{2,3†}

¹Harbin Institute of Technology, ²Shanghai Artificial Intelligence Laboratory

³The Chinese University of Hong Kong, ⁴Fudan University

shenghez.zheng@gmail.com

Abstract

With the growing availability of open-source task-specific models, model merging has become a practical way to combine multiple models into one. However, existing studies mainly focus on fully fine-tuned models and overlook widely used LoRA modules. We show that full fine-tuning merging methods perform poorly on LoRA due to its larger cross-task magnitude disparities, which distort merged parameters and degrade performance. To address this, we propose **DO-Merging** (Decouple and Orthogonalize), a data-free framework for LoRA merging. Specifically, we decompose model weights into magnitude and direction components and merge them independently, preventing magnitude imbalances from distorting the merged direction. We further introduce a layer-wise gradient descent method with orthogonal constraints that reduces inter-task interference among direction components without requiring any training data. We provide theoretical analysis supporting both components. Extensive experiments across vision, language, and multi-modal domains show that DO-Merging consistently outperforms existing methods, achieving up to 5% gains on vision tasks, with negligible additional cost. Both parts can be integrated into existing merging methods as plug-and-play enhancements.

1 Introduction

Deep learning supports many applications (Devlin et al., 2019; Touvron et al., 2023), yet edge users often lack resources to train models, making ready-to-use ones preferable. With platforms like HuggingFace (Wolf, 2019), such access is now easy. Real-world pipelines usually involve multiple sub-tasks (Yang et al., 2024b; Jin et al., 2022), and deploying separate models increases cost and complexity. Model merging offers a practical alternative by consolidating pretrained models into one

that handles all target tasks (Ilharco et al., 2022; Matena and Raffel, 2022; Yadav et al., 2024), reducing both retraining and deployment effort.

Model merging primarily addresses task interference and is categorized by target models: full fine-tuning or PEFT methods like LoRA (Hu et al., 2022). For full fine-tuning, approaches include: (1) automatic coefficient computation (Yang et al., 2024b; Jin et al., 2022; Matena and Raffel, 2022), (2) optimization-based conflict reduction via task vectors (Yadav et al., 2024; Davari and Belilovsky, 2023; Du et al., 2024; Zheng and Wang, 2024), and (3) task-specific modules (Huang et al., 2024b; Lu et al., 2024b). However, these methods either require architectural changes, limiting usability, or perform poorly in LoRA as shown in Fig. 1(a). Thus, specialized LoRA merging methods are needed. Existing LoRA-specific methods mainly include subspace projection merging (Stoica et al., 2025; Zhao et al., 2024b) and parameter-driven coefficient computation (Chen et al., 2024a; Ouyang et al., 2025; Prabhakar et al., 2025; Huang et al., 2024a). However, without a fundamental analysis of differences between LoRA and full fine-tuning, existing LoRA merging methods often result in limited gains or poor generalization across tasks, underscoring the need for improved strategies.

We find that LoRA modules exhibit wider magnitude disparities across tasks than full fine-tuning (Fig. 1b), and this disparity is closely linked to degraded merging performance (Fig. 1c and Appendix F). We propose that this loss arises because merging should preserve both the scale and direction of parameter distributions to maintain input mapping patterns (Yu et al., 2024a). However, in LoRA, large-magnitude modules dominate the merged distribution, distorting task representations and leading to performance degradation.

To address this, we introduce **DO-Merging** (Decouple and Orthogonalize) as shown in Fig. 2. We decompose each parameter matrix into magnitude

^{*}Work done during the internship at Shanghai AI Lab.

[†]Corresponding Author.

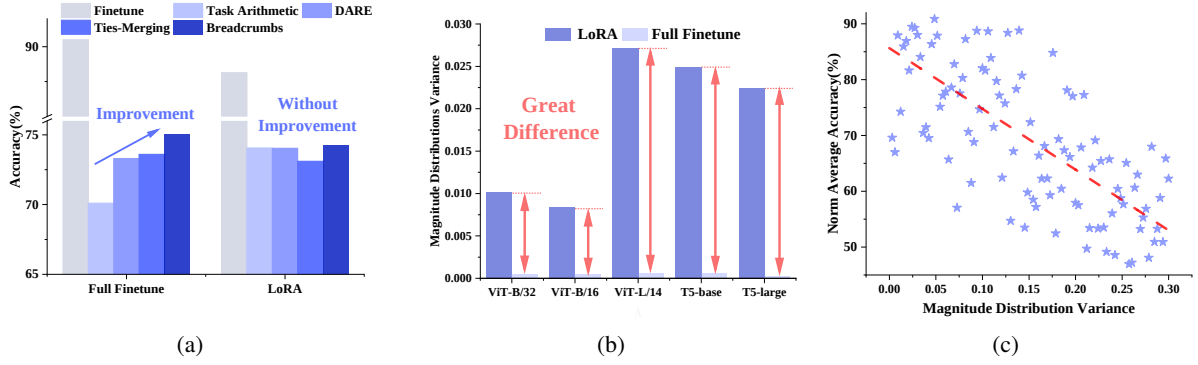


Figure 1: Key Observations on LoRA Merging. (a) Existing methods work well for full fine-tuning but are no better than simple Task Arithmetic on LoRA. (b) LoRA shows larger magnitude disparities than full fine-tuning across tasks. The Magnitude Distribution Variance is calculated as discussed in Appendix C.5. (c) A greater magnitude disparity correlates with worse merging performance.

and direction vectors, merging them separately so that scale differences no longer bias the merged direction. For magnitude vectors, we perform average merging to preserve scale distributions. For direction vectors, we apply data-free, layer-wise gradient descent with orthogonal constraints to reduce task interference. Since LoRA merging operates in the full-rank space (Appendix D.4), we orthogonalize the two LoRA components separately before decoupling for lower cost. Compared to existing methods, DO-Merging imposes no constraints on training (Zhang et al., 2025a) and requires no data, offering a general and efficient alternative.

We provide theoretical analysis supporting DO-Merging. Experiments across vision, language, and multi-modal tasks demonstrate broad applicability, delivering up to 5% gains on vision tasks and consistent improvements on LLMs. Both components plug into existing methods as near-free improvements. Our contributions are:

- We show that existing full fine-tuning merging methods perform poorly on LoRA, closely linked to the greater magnitude disparities in LoRA.
- We propose DO-Merging, the first method to address LoRA merging degradation via magnitude-direction decoupling and data-free orthogonalization, supported by theoretical analysis.
- Comprehensive experiments across vision, language, and multi-modal tasks validate DO-Merging. Both components flexibly combine with existing approaches, bringing consistent gains.

2 Related Work

Model Merging. Model merging combines models trained on different tasks into a single model (Il-

harco et al., 2022; Jin et al., 2022; Yang et al., 2024a; Ye et al., 2025c). The main challenge is resolving task conflicts (Yadav et al., 2024; Qi et al., 2024; Zhou et al., 2024). Existing full fine-tuning merging methods fall into three groups: computing merging coefficients (Yang et al., 2024b; Jin et al., 2022; Matena and Raffel, 2022), reducing parameter conflicts (Ilharco et al., 2022; Yang et al., 2024b; Du et al., 2024; Gargiulo et al., 2024a; Qi et al., 2024; Gargiulo et al., 2024b), and storing auxiliary knowledge (Huang et al., 2024b; Lu et al., 2024b). However, they often require substantial data or computation, task-specific designs, or architectural changes, limiting generality. Details are given in Appendix C.3. Moreover, these methods largely overlook LoRA and degrade noticeably when applied to it, motivating DO-Merging (Fig. 1).

LoRA Merging. LoRA merging methods mainly compute automatic coefficients (Chen et al., 2024a; Shenaj et al., 2024; Huang et al., 2024a; Prabhakar et al., 2025; Zeng et al., 2025; Yang et al., 2024c), often requiring data or heavy computation, or adjust parameters (Zhao et al., 2024b; Stoica et al., 2025), typically relying on task-specific constraints. Due to limited LoRA analysis, these methods often perform poorly. Model swarm-style methods also require data and incur high inference costs. Details are provided in Appendix C.3. We identify large cross-LoRA magnitude disparities as the key challenge and address it with DO-Merging.

Weight Decoupling. Weight decoupling is usually used as a training constraint to improve optimization (Liu et al., 2024a; Hu et al., 2024; Zhang et al., 2025a), which is unsuitable for open-source weights. DoRA (Liu et al., 2024a) reparameterizes LoRA into magnitude and direction during training, while DoTA (Hu et al., 2024) computes

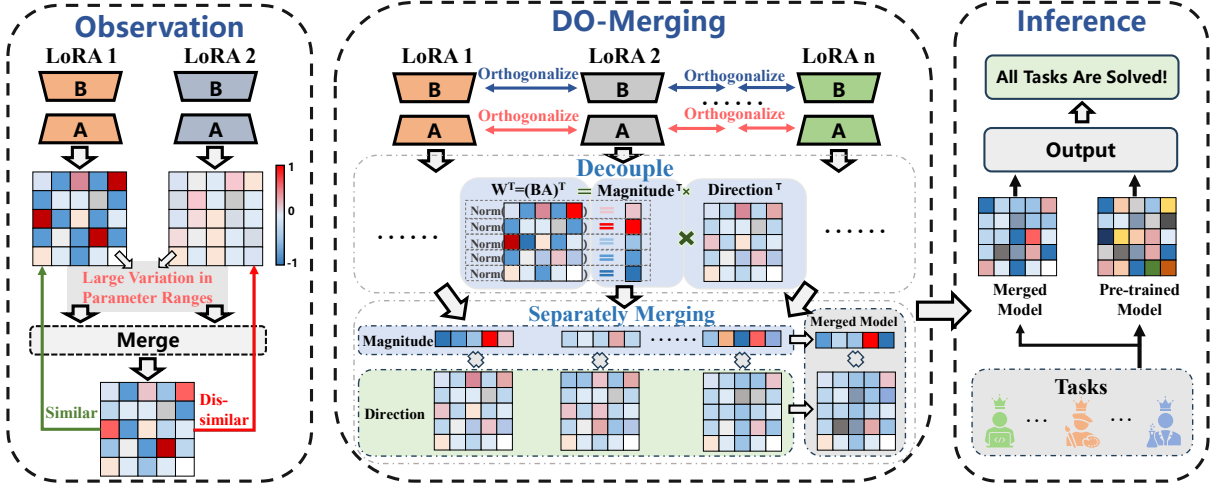


Figure 2: DO-Merging framework. Left: large cross-task LoRA magnitude disparities hurt merging. Middle: orthogonal perturbation, decoupling, and separate merging. Right: one merged model serves multiple tasks.

fusion coefficients from decoupled components, both requiring control over training. Other methods compute fusion coefficients for model merging (Yu et al., 2024a; Xiong et al., 2024), but their limited analysis of LoRA failures leads to poor results. In contrast, DO-Merging applies magnitude-direction decomposition post-hoc to arbitrary pre-trained LoRA modules, requires no retraining or data, and directly merges decoupled components to address LoRA-specific magnitude disparities.

Weight Orthogonality. Prior work shows that orthogonal weights improve model merging (Zhang et al., 2025a; Gargiulo et al., 2024a), but existing methods either require training-time constraints (Zhang et al., 2025a) or sacrifice orthogonality and task performance (Gargiulo et al., 2024a). DO-Merging preserves both for efficient merging.

3 Methodology

3.1 Preliminary

Given a base model θ_{pre} , let $\{\Delta_i\}_{i=1}^n$ denote the LoRA modules for n tasks, each containing K layers. The k -th layer of Δ_i is denoted as a matrix $W_i^k = B_i^k A_i^k$, where B_i^k and A_i^k are the corresponding LoRA parameters. We aim to construct a merged module $\Delta = \lambda \sum_{i=1}^n g(\Delta_i)$ such that the model $\theta_{pre} + \Delta$ can solve all n tasks simultaneously, where g denotes the transformation applied to each module (detailed in Secs. 3.3–3.4).

3.2 Motivation

As shown in Fig. 1(b), LoRA modules exhibit far larger magnitude disparities than full fine-tuning, which strongly correlates with the performance

drop in Fig. 1(c). Ideally, merging should preserve both the magnitude and direction of model weights to maintain consistent input–output mappings. However, when modules have vastly different scales, the merged model becomes dominated by large-magnitude ones, suppressing information from smaller modules and degrading performance. This trend appears across architectures and tasks (Appendix F), suggesting magnitude disparity is a key factor behind unstable merging.

To address this, we decouple magnitude and direction during merging, handling them separately. We merge magnitudes in a distribution-preserving way (Sec. 3.3) and apply efficient orthogonalization to direction vectors (Sec. 3.4). The combined model (Sec. 3.5) mitigates direction loss caused by magnitude imbalance. See Appendix D.1 details.

3.3 Decouple

First, we decouple magnitude by extracting column-wise normalization coefficients as magnitude vectors, leaving direction vectors with uniform magnitudes. For a task vector layer $W = BA \in \mathbb{R}^{m \times d}$, the decoupling process is:

$$W = \overline{W} \text{diag}(\alpha), \quad \alpha_{[j]} = \|W_{[:,j]}\|_2. \quad (1)$$

Here, $\alpha \in \mathbb{R}^{1 \times d}$ is the magnitude vector, and $\overline{W}$ is the direction matrix with unit- ℓ_2 -norm columns. We choose column normalization because for a neural network layer with pre-trained weights W_{pre} and task-specific parameters W_t , the output satisfies $x^{out} = x(W_{pre} + W_t)$, where each output column j depends on all entries of $W_t[:,j]$. This means each column of the task vector contributes jointly to the corresponding output dimension.

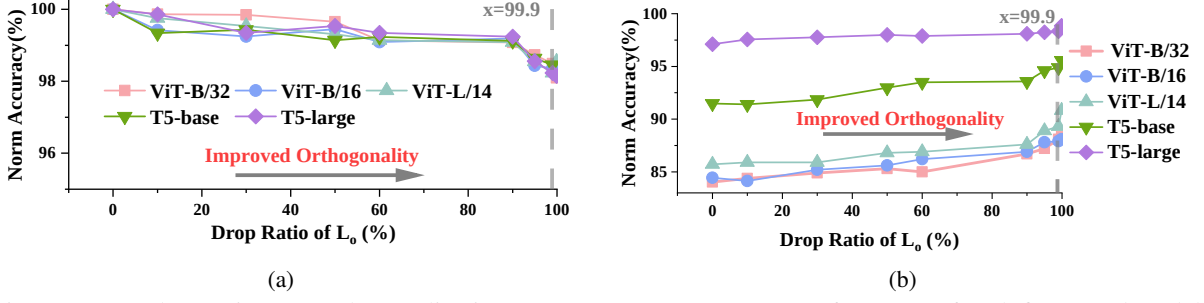


Figure 3: Key observations on orthogonalization (Eq. 4): (a) Average norm performance of each fine-tuned model on its own task (as defined in Eq. 30 in Appendix) remains stable under orthogonal gradient descent; (b) greater orthogonality improves merged norm accuracy.

sion. Thus, aligning column magnitudes across task vectors helps preserve output characteristics.

To preserve amplitude features, we average the magnitude vectors: $\alpha_{\text{merge}} = \frac{1}{n} \sum_{i=1}^n \alpha_i$ (the factor $\frac{1}{n}$ can be absorbed into λ in Eq. 5). The following theorem shows that decoupling alone improves merging. Proof is in Appendix B.2.

Assumption 3.1 Consider matrices $\{W_i\}_{i=1}^n \in \mathbb{R}^{m \times d}$, where $W_i = \bar{W}_i \cdot \text{diag}(\alpha_i)$ with magnitude vector $\alpha_i \in \mathbb{R}^{1 \times d}$ and unit-norm-column direction matrix $\bar{W}_i$. Assume approximately zero-mean entries in $\bar{W}_i$ columns (validated empirically in Appendix B.1) and mutual independence across tasks. The merging quality is measured by:

$$L = \sum_{i=1}^n \frac{\sum_{j=1}^n \|\alpha_j\|_2}{\|\alpha_i\|_2} \|W - W_i\|_F^2. \quad (2)$$

Theorem 3.1 Under Assumption 3.1, let $a = \|\alpha_1\|_2$, $b = \|\alpha_2\|_2$, $S = \sum_j \alpha_{1j} \alpha_{2j}$, and define $W^{(1)} = \frac{1}{2} \sum_{i=1}^2 \bar{W}_i \text{diag}(\alpha_i)$ (non-decoupled) and $W^{(2)} = \frac{1}{2} (\sum_{i=1}^2 \bar{W}_i) \cdot \frac{1}{2} (\sum_{j=1}^2 \text{diag}(\alpha_j))$ (decoupled). The expected loss gap is:

$$\mathbb{E}(L_1) - \mathbb{E}(L_2) = \frac{(a+b)^2}{8ab} [(a-b)^2 - 2(ab-S)]. \quad (3)$$

Defining magnitude ratio $r = \frac{\max(a,b)}{\min(a,b)} \geq 1$ and correlation $\rho = \frac{S}{ab} \in [0, 1]$, decoupling improves merging (i.e., $\mathbb{E}(L_2) \leq \mathbb{E}(L_1)$) whenever $\rho \geq 1 - (r-1)^2/(2r)$. When $r \geq 2 + \sqrt{3}$, this holds for all ρ . The proof is for $n=2$; since the total loss decomposes into pairwise contributions, the result extends to $n > 2$ whenever a sufficient number of task pairs satisfy the condition (see Appendix B.2).

Takeaway 3.1 Large magnitude disparities degrade merging, a common failure mode in LoRA. Decoupling separates magnitude and direction so that scale differences no longer distort the merged

direction. The benefit increases with magnitude disparity under high magnitude alignment (ρ); when $r \geq 2 + \sqrt{3}$, decoupling is unconditionally beneficial (proof in Appendix B.2).

3.4 Orthogonalize

We next optimize direction vectors. Nearly orthogonal task vectors reduce feature overlap and task interference (Gargiulo et al., 2024a), but public LoRA modules rarely satisfy this, unlike training-time orthogonalized LoRA (Zhang et al., 2025a). Leveraging the redundancy of fine-tuned parameters (Yu et al., 2024b; Ye et al., 2025a,b), we impose orthogonality on existing LoRA modules without data. As shown in Fig. 3, this preserves individual accuracy and improves merging. The layer-wise loss is as following:

$$L = \underbrace{\sum_i \sum_{j \neq i} \|(W_i + \delta_i)^T (W_j + \delta_j)\|_F^2}_{L_o} + \underbrace{\eta \sum_i \|\delta_i\|_F^2}_{L_r}, \quad (4)$$

where $\{\delta_i\}_{i=1}^n$ are learnable and η is a regularization hyperparameter (we use $\eta = 1$ by default; sensitivity analysis in Tab. 19 in Appendix). L_o encourages orthogonality between task vectors, while L_r constrains the magnitude of change. In simple terms, we apply orthogonal gradient descent directly to different LoRA modules and use a regularization term to constrain the update magnitude, preserving their performance. Since optimization is layer-wise, computational cost remains minimal.

For LoRA, we apply orthogonality separately on A and B , which approximates orthogonality on their product at lower cost (see Appendix F). The following analysis clarifies why minimizing L_o reduces task interference (details in Appendix B.3).

Remark 3.2 The merged direction $\bar{W}_{\text{merge}} = \sum_j \bar{W}_j$ projected onto task i decomposes as $\|\bar{W}_i\|_F^2 + \mathcal{I}_i$, where the self-contribution $\|\bar{W}_i\|_F^2$

Table 1: Multi-task performance when merging ViT-B/32 on eight vision tasks. Baselines are deterministic given fixed pretrained weights. We report mean $\pm$ std over 3 seeds for DO-Merging.

Method	SUN397	Cars	RESISC45	EuroSAT	SVHN	GTSRB	MNIST	DTD	Avg.
Pre-trained	62.30	59.70	60.70	45.50	31.40	32.60	48.50	43.80	48.06
Finetune	71.81	71.23	94.33	98.85	97.19	98.62	99.63	73.40	88.13
Task Arithmetic (Ilharco et al., 2022)	63.38	60.27	75.46	87.70	84.73	70.74	96.64	53.55	74.06
Ties Merging (Yadav et al., 2024)	63.81	53.95	73.56	86.00	88.27	71.15	97.81	50.32	73.11
Breadcrumbs (Davari and Belilovsky, 2023)	64.02	57.87	74.12	88.73	84.24	74.28	97.29	53.23	74.22
AdaMerging (Yang et al., 2024b)	62.59	57.89	75.94	90.63	78.93	88.82	96.92	52.07	75.47
PCB-Merging (Du et al., 2024)	63.74	58.64	73.58	88.12	85.65	75.12	97.45	54.07	74.54
KNOTS (Stoica et al., 2025)	64.08	59.26	76.11	86.93	86.94	75.54	96.85	57.46	75.39
TSVM (Gargiulo et al., 2024a)	64.81	59.31	78.73	89.63	85.43	76.63	97.02	57.93	76.19
WUDI (Cheng et al., 2025)	63.92	57.81	75.86	89.00	86.85	78.76	97.89	50.96	75.13
CoPA-Merging (Zeng et al., 2025)	63.40	59.56	76.68	88.93	85.54	77.00	97.14	56.56	75.60
DO-Merging(ours)	64.48 $\pm$ 0.24	61.01 $\pm$ 0.33	77.30 $\pm$ 0.25	88.30 $\pm$ 0.27	89.05 $\pm$ 0.23	83.31 $\pm$ 0.16	98.03 $\pm$ 0.12	61.53 $\pm$ 0.21	77.88 $\pm$ 0.22

Algorithm 1 Workflow of DO-Merging

Input: Fine-tuned LoRAs $\{B_i, A_i\}_{i=1}^n$ for each layer, pre-trained model W_{pre} .

- 1: **Step1: Orthogonalize;** $\triangleright$ Gradient descent on Eq. 4
- 2: $\{\hat{B}_i\}_{i=1}^n = Ortho\{B_i\}_{i=1}^n$
- 3: **Step2: Decouple;**
- 4: $\{W_i\}_{i=1}^n = \{\hat{B}_i \hat{A}_i\}_{i=1}^n$
- 5: **for** $i \in [1, n]$ **do**
- 6: Get $\bar{W}_i, \alpha_i$ as Eq. 1.
- 7: **end for**
- 8: **Step3: Merge;**
- 9: $W_{out} = W_{pre} + \lambda \left(\sum_{j=1}^n \bar{W}_j \right) \left(\sum_{i=1}^n \text{diag}(\alpha_i) \right)$

Output: Merged model W_{out} .

preserves task i 's directional information and $\mathcal{I}_i = \sum_{j \neq i} \text{tr}(\bar{W}_i^T \bar{W}_j)$ is inter-task interference. Gradient descent on L_o directly reduces cross-task correlations $\|W_i^T W_j\|_F^2$, decreasing $|\mathcal{I}_i|$; the regularization L_r bounds perturbation magnitude to preserve individual task performance (Fig. 3a).

Takeaway 3.2 Orthogonalizing parameters reduces mutual interference but may cause performance loss, which we mitigate with a penalty term.

3.5 Workflow

Given the merged direction and magnitude vectors, and the direction matrices $\bar{W}_j$ from orthogonalized task vectors, the final result is:

$$W_{out} = W_{pre} + \lambda \left(\sum_{j=1}^n \bar{W}_j \right) \left(\sum_{i=1}^n \text{diag}(\alpha_i) \right). \quad (5)$$

Alg. 1 provides the pseudo-code. We first apply orthogonality constraints to the two low-rank LoRA matrices (lines 2–3) to encourage independent features, then decompose magnitudes and directions

and merge them separately to obtain the final result (lines 5–10). Note that DO-Merging is data-free in that it requires no training data for orthogonalization; only the standard merging coefficient λ is selected via validation, as in all compared methods.

Complementarity. Decoupling reduces loss from magnitude mismatch (Theorem 3.1), while orthogonalization reduces directional interference $\mathcal{I}_i$ (Remark 3.2). They target distinct sources of merging loss and are naturally complementary: decoupling prevents magnitude differences from distorting the merged direction, while orthogonalization further reduces remaining directional conflicts. Empirical results in Tables 1–16 confirm that combining both consistently outperforms either alone.

4 Experiments

We evaluate DO-Merging on vision models, LLMs, and multimodal models, followed by ablations, discussions, and efficiency analysis (Secs. 4.1–4.6). Additional details are in Appendix F. For ViT-B/32, we report mean $\pm$ std over 3 seeds (Tab. 1); since std is consistently below 0.35%, we report single-run results elsewhere for brevity.

4.1 Vision Models

We evaluate ViT-B/32, ViT-B/16, and ViT-L/14 fine-tuned with LoRA on eight visual classification tasks: SUN397, Cars, RESISC45, EuroSAT, SVHN, GTSRB, MNIST, and DTD (Ilharco et al., 2022). Detailed configurations are provided in Appendix C.2. Results are reported in Tabs. 1 and 4, with additional results in Appendix F.

Our method consistently outperforms existing approaches across all three ViT scales, with improvements of 3–5% over Task Arithmetic. Notably, on tasks where models face strong inter-task interference (e.g., SVHN and GTSRB), DO-

Table 2: T5-large merging on eight language tasks. DO-Merging reports mean $\pm$ std over 3 seeds.

Method	COLA	MNLI	MRPC	QNLI	QQP	RTE	SST2	STSBB	Avg.
Pre-trained	73.7	56.6	82.4	91.1	85.5	85.6	94.3	87.5	82.1
Finetune	80.2	88.5	89.2	94.4	87.2	91.7	95.2	90.9	89.6
Task Arithmetic (Ilharco et al., 2022)	76.9	85.4	85.3	93.9	85.8	88.1	95.2	87.8	87.3
Ties Merging (Yadav et al., 2024)	77.1	85.1	86.3	93.9	86.0	87.7	95.1	88.0	87.4
Breadcrumbs (Davari and Belilovsky, 2023)	77.2	85.2	86.1	93.9	86.0	87.4	94.9	88.4	87.4
PCB-Merging (Du et al., 2024)	77.3	85.4	85.8	93.9	86.0	87.9	95.2	88.0	87.4
KNOTS (Stoica et al., 2025)	76.8	87.4	86.4	93.5	86.3	87.3	95.2	88.6	87.7
TSVM (Gargiulo et al., 2024a)	76.0	88.0	84.3	92.9	86.3	87.4	95.2	88.5	87.3
CoPA-Merging (Zeng et al., 2025)	76.0	88.4	86.5	93.0	86.1	87.7	95.3	88.4	87.7
DO-Merging(ours)	78.9 $\pm$ 0.33	88.5 $\pm$ 0.15	88.2 $\pm$ 0.15	93.8 $\pm$ 0.14	86.5 $\pm$ 0.18	87.4 $\pm$ 0.11	96.1 $\pm$ 0.12	89.2 $\pm$ 0.14	88.6 $\pm$ 0.15

Table 3: The merging performance of LLaMA3-8B and Qwen-14B on the corresponding tasks.

Base Model	LLaMA3-8B							Qwen-14B			
Task	SNLI	MNLI	SICK	QNLI	RTE	SCITAIL	Avg.	MMLU	TruthfulQA	BBQ	Avg.
Pre-trained	41.69	34.64	55.63	51.92	42.03	60.72	47.77	69.30	51.27	80.69	67.09
Finetune	92.49	90.30	91.58	94.48	89.85	96.51	92.53	68.35	54.34	93.53	72.07
Task Arithmetic (Ilharco et al., 2022)	86.56	86.05	80.57	64.91	89.85	93.36	83.55	67.62	53.38	78.24	66.41
Ties Merging (Yadav et al., 2024)	86.48	86.15	80.43	65.04	89.85	93.75	83.61	68.27	50.01	84.10	67.46
KNOTS (Stoica et al., 2025)	87.95	86.94	83.03	67.20	89.58	93.58	84.71	68.20	52.48	83.56	68.08
TSVM (Gargiulo et al., 2024a)	88.82	83.96	81.79	74.87	91.30	94.02	85.79	68.10	52.24	82.42	67.58
CoPA-Merging (Zeng et al., 2025)	88.59	87.49	84.59	67.60	89.85	93.67	85.29	68.10	52.11	83.75	67.98
DO-Merging(ours)	89.05	88.05	83.14	77.13	91.30	94.02	87.11	68.45	53.88	84.99	69.10

Table 4: Average multi-task performance on eight tasks.

Method	ViT-B/16	ViT-L/14
Pre-trained	55.20	64.89
Finetune	89.94	93.11
Task Arithmetic (Ilharco et al., 2022)	75.94	79.80
Ties Merging (Yadav et al., 2024)	76.18	79.28
Breadcrumbs (Davari and Belilovsky, 2023)	76.00	79.60
PCB-Merging (Du et al., 2024)	77.19	80.79
KNOTS (Stoica et al., 2025)	76.51	80.95
TSVM (Gargiulo et al., 2024a)	77.19	80.86
WUDI (Cheng et al., 2025)	77.02	80.92
CoPA-Merging (Zeng et al., 2025)	76.79	81.43
DO-Merging(ours)	79.24	84.58

Merging achieves particularly large gains, confirming that reducing task interference via decoupling and orthogonality is especially beneficial when parameter conflicts are severe. Moreover, the improvements are consistent across model scales, with larger models (ViT-L/14) benefiting even more, indicating good scalability of our approach. Note that AdaMerging requires unlabeled test data for coefficient optimization, whereas DO-Merging and all other baselines are strictly data-free.

4.2 Language Models

In this part, we evaluate DO-Merging on language models of different scales. For medium-sized models, we fine-tune T5-base and T5-large (Raffel et al., 2020) on eight GLUE tasks (Tang et al., 2024). Results are in Tab. 2 and Appendix F. DO-Merging improves performance by over 1% on both models.

For large-scale models, we evaluate LLaMA3-8B (Grattafiori et al., 2024) on six NLU tasks (Stoica et al., 2025) and Qwen-14B (Bai et al., 2023) on MMLU (Hendrycks et al., 2021), TruthfulQA (Lin et al., 2022), and BBQ (Parrish et al., 2022). Settings are in Appendix C.2. As shown in Tab. 3, DO-Merging improves over the strongest baseline by over 1% on both models, while other methods offer limited gains over Task Arithmetic, highlighting the need to address magnitude differences.

4.3 Multi-modal Models

We evaluate DO-Merging on multi-modal Qwen2-VL tasks (Wang et al., 2024), including POPE (Li et al., 2023), MMStar (Chen et al., 2024b), MM-Bench (Liu et al., 2024b), RealWorldQA (xAI, 2025), and MathVista (Lu et al., 2024a). These tasks cover hallucination detection, multi-modal reasoning, and mathematical reasoning, with low inter-task correlation making merging challenging. As shown in Tab. 5, DO-Merging performs best across all tasks and even surpasses fine-tuning on MMStar, showing that orthogonalization handles diverse task combinations beyond vision.

4.4 Ablation Study

Decouple and Orthogonalize. Tab. 6 shows ablation results on ViT-B/32, ViT-B/16, and ViT-L/14 across eight visual classification tasks. Adding only orthogonalization yields over 2% average im-

Table 5: Qwen2-VL merging on multi-modal tasks. Improvements in parentheses are relative to Task Arithmetic.

Method	POPE	MMStar	MathVista	RealworldQA	MMBench	Avg.
Pre-trained	85.86	59.33	57.80	69.80	80.32	70.62
Finetune	97.75	63.60	64.00	73.46	97.50	79.26
Task Arithmetic (Ilharco et al., 2022)	91.27	64.60	61.50	69.67	90.98	75.60
Ties Merging (Yadav et al., 2024)	91.27	64.80	61.40	70.42	91.13	75.80(↑ 0.20%)
Breadcrumbs (Davari and Belilovsky, 2023)	92.83	66.40	60.69	69.67	90.98	76.11(↑ 0.51%)
KNOTS (Stoica et al., 2025)	92.42	65.64	61.55	70.46	92.58	76.53(↑ 0.93%)
TSVM (Gargiulo et al., 2024a)	92.74	65.26	61.60	70.21	93.49	76.66(↑ 1.06%)
CoPA-Merging (Zeng et al., 2025)	93.32	65.33	62.50	70.31	93.45	76.98(↑ 1.38%)
DO-Merging(ours)	94.74	66.80	62.80	72.13	96.30	78.50(↑ 2.90%)

Table 6: Ablation of decoupling and orthogonalization. $\times/\checkmark$ denote excluded/included modules.

Decouple	Orthogonalize	ViT-B/32	ViT-B/16	ViT-L/14
$\times$	$\times$	74.06	75.94	79.80
$\times$	$\checkmark$	76.90	78.07	83.02
$\checkmark$	$\times$	74.93	77.05	83.20
$\checkmark$	$\checkmark$	77.87	79.24	84.58

Table 7: Orthogonalization comparison under the same decoupling. TSVM uses SVD whitening, while ours uses regularized gradient descent (Eq. 4).

Orthogonalization	ViT-B/32	ViT-B/16	ViT-L/14
None (Decouple only)	74.93	77.05	83.20
TSVM + Decouple	76.87	78.45	82.66
Ours + Decouple	77.87	79.24	84.58

provement; adding only decoupling yields over 1%. Orthogonalization contributes more, consistent with its role in directly reducing directional interference (Remark 3.2), while decoupling primarily addresses magnitude imbalance. This trend holds consistently across all three ViT models. When both components are used together, the performance gain becomes even more significant, demonstrating their complementary effectiveness.

Magnitude Extraction. Tab. 8 compares different normalization granularities for the magnitude vector. *Matrix Norm* uses normalization over the entire matrix, while *Row Norm* and *Col Norm* use row-wise and column-wise normalization, respectively. Matrix norm performs poorly due to coarse granularity. Column norm outperforms row norm, since each column corresponds to an output dimension jointly, aligning with our analysis.

Orthogonalization Strategy. To isolate the effect of our orthogonalization design, we compare it with TSVM’s SVD-based whitening under the same decoupling setting (Tab. 7). Although both improve over the decouple-only baseline, our regularized gradient-descent approach consistently performs better across all three ViT scales. This advantage

Table 8: Ablation on magnitude vectors: *matrix norm*, *row norm*, and *col norm* denote full-matrix, row-wise, and column-wise normalization.

	ViT-B/32	ViT-B/16	ViT-L/14
Task Arithmetic	74.06	75.94	79.80
Matrix Norm	76.98	76.21	77.45
Row Norm	76.89	76.67	82.52
Col Norm	77.87	79.24	84.58

comes from the explicit regularization term L_r in Eq. 4, which limits perturbation magnitude and better preserves individual task performance. In contrast, TSVM whitening lacks such constraints and may distort task-specific features.

4.5 Discussions

Flexible Combination. Our decoupling and orthogonalization enhance existing merging methods. As shown in Fig. 4(a), adding either to Breadcrumbs and Ties-Merging on ViT-B/32 consistently improves performance, confirming their plug-and-play and complementary roles.

Orthogonal Effect. In Fig. 4(b), gradient descent increases orthogonality, while gradient ascent reduces it. Accuracy drops sharply as orthogonality decreases, highlighting its importance for merging. This supports Remark 3.2: reducing cross-task correlations lowers interference $\mathcal{I}_i$, while the opposite amplifies destructive interference.

The Impact of LoRA Rank. Most experiments use a LoRA rank of 16. Here, we evaluate DO-Merging on ViT-B/32 with ranks 16, 32, 64, and 128. As shown in Fig. 4(c), our method consistently outperforms others at all ranks, confirming robustness across different rank settings.

Transfer to Full-Finetune Merging. As shown in Fig. 5, orthogonalization remains highly effective for full fine-tuning, while decoupling gives smaller gains due to weaker magnitude differences. Nevertheless, DO-Merging still shows its generality.

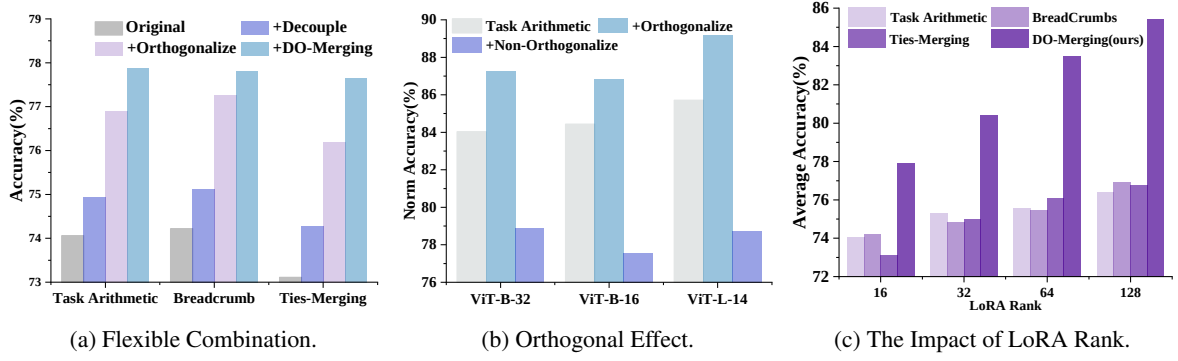


Figure 4: Key properties of DO-Merging. (a) Its two components can be freely combined with other merging methods and consistently improve performance. (b) LoRA orthogonality is crucial, as making direction vectors non-orthogonal degrades performance. (c) DO-Merging performs well across different LoRA ranks.

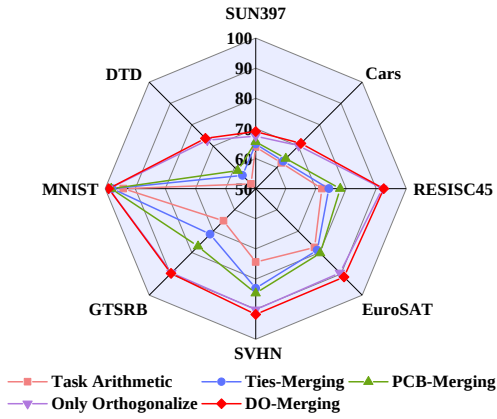


Figure 5: DO-Merging is also effective for merging fully fine-tuned models, as verified on ViT-B/32.

Parameter Sensitivity. We vary λ from 0.2 to 0.7 across three ViT architectures (Tab. 9). All models reach their best performance at $\lambda = 0.4$ and remain competitive within 0.3–0.5, suggesting that the method is relatively insensitive to this hyperparameter. Performance only drops at extreme values ($\lambda \leq 0.2$ or $\lambda \geq 0.7$), where the contribution of pretrained weights and merged task vectors becomes less balanced.

4.6 Efficiency Analysis

We compare DO-Merging with MTL, LoRAMoE (Dou et al., 2023), MoLa (Gao et al., 2024), and MixLoRA (Li et al., 2024) on LLaMA3-8B with rank-16 LoRAs (Stoica et al., 2025). As shown in Tab. 10, DO-Merging requires no training data, finishes within 2 minutes, and uses only 2GB GPU memory. In contrast, MTL and MoE-based methods need over 1M samples, more than 72 hours of training, and 20GB+ memory, causing severe bottlenecks on a single RTX 3090. Thus, DO-Merging offers a practical deployable solution for users with GPUs under 24GB.

Table 9: The sensitivity analysis of the parameter λ .

λ	0.2	0.3	0.4	0.5	0.6	0.7
ViT-B/32	75.43	76.67	77.88	77.42	76.56	70.23
ViT-B/16	76.45	78.54	79.24	78.85	77.45	73.64
ViT-L/14	80.64	82.34	84.58	83.98	82.87	78.56

Table 10: Comparison of resource consumption among different methods using LLaMA3-8B on six NLU tasks.

Method	Training data size	Training Cost (h)	GPU memory
Pre-trained	0	0	0
MTL	1.09M	>72	20GB
LoRAMoE (Dou et al., 2023)	1.09M	>72	>20GB
MoLa (Gao et al., 2024)	1.09M	>72	21GB
MixLoRA (Li et al., 2024)	1.09M	>72	21GB
DO-Merging	0	0.03	2GB

5 Conclusion

In this paper, we study the poor performance of existing model merging methods when applied to LoRA-based fine-tuned models. We identify the large parameter magnitude disparities introduced during LoRA training as a key reason for this degradation, as they can distort the merged update direction and amplify task interference. To address this issue, we propose DO-Merging, a simple yet effective merging framework that decouples magnitude and direction, thereby preserving directional consistency during merging. We further introduce data-free orthogonal perturbations to mitigate interference among task-specific updates. Extensive experiments across vision, language, and multi-modal tasks demonstrate consistent improvements, with gains of up to 5% on vision tasks and over 1% on LLMs and multi-modal models. Since both components are plug-and-play and require neither training data nor architectural modifications, DO-Merging provides a practical and broadly applicable solution for LoRA merging.

Limitations

This paper introduces DO-Merging, a model merging method specifically designed for LoRA, aiming to address the performance degradation caused by the differences between LoRA and full fine-tuning during the adaptation process. Like many existing model merging approaches, its main limitation lies in the need for users to manually find and select the desired model weights, which can be time-consuming and require significant effort in model retrieval.

Acknowledgments

This work is supported by National Natural Science Foundation of China (No. 62571138).

References

- Jinze Bai, Shuai Bai, Yunfei Chu, Zeyu Cui, Kai Dang, Xiaodong Deng, Yang Fan, Wenbin Ge, Yu Han, Fei Huang, and 1 others. 2023. Qwen technical report. *arXiv preprint arXiv:2309.16609*.
- Hongxu Chen, Runshi Li, Bowei Zhu, Zhen Wang, and Long Chen. 2024a. Iteris: Iterative inference-solving alignment for lora merging. *arXiv preprint arXiv:2411.15231*.
- Lin Chen, Jinsong Li, Xiaoyi Dong, Pan Zhang, Yuhang Zang, Zehui Chen, Haodong Duan, Jiaqi Wang, Yu Qiao, Dahua Lin, and 1 others. 2024b. Are we on the right way for evaluating large vision-language models? In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*.
- Runxi Cheng, Feng Xiong, Yongxian Wei, Wanyun Zhu, and Chun Yuan. 2025. Whoever started the interference should end it: Guiding data-free model merging via task vectors. In *Forty-second International Conference on Machine Learning*.
- MohammadReza Davari and Eugene Belilovsky. 2023. Model breadcrumbs: Scaling multi-task model merging with sparse masks. *arXiv preprint arXiv:2312.06795*.
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. [BERT: Pre-training of deep bidirectional transformers for language understanding](#). In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.
- Shihan Dou, Enyu Zhou, Yan Liu, Songyang Gao, Jun Zhao, Wei Shen, Yuhao Zhou, Zhiheng Xi, Xiao Wang, Xiaoran Fan, and 1 others. 2023. Lora-moe: Revolutionizing mixture of experts for maintaining world knowledge in language model alignment. *arXiv preprint arXiv:2312.09979*, 4(7).
- Guodong Du, Junlin Lee, Jing Li, Runhua Jiang, Yifei Guo, Shuyang Yu, Hanting Liu, Sim Kuan Goh, Ho-Kin Tang, Daojing He, and 1 others. 2024. Parameter competition balancing for model merging. *arXiv preprint arXiv:2410.02396*.
- Shangbin Feng, Zifeng Wang, Yike Wang, Sayna Ebrahimi, Hamid Palangi, Lesly Miculicich, Achin Kulshrestha, Nathalie Rauschmayr, Yejin Choi, Yulia Tsvetkov, and 1 others. 2024. Model swarms: Collaborative search to adapt llm experts via swarm intelligence. *arXiv preprint arXiv:2410.11163*.
- Chongyang Gao, Kezhen Chen, Jinmeng Rao, Baochen Sun, Ruibo Liu, Daiyi Peng, Yawen Zhang, Xiaoyuan Guo, Jie Yang, and VS Subrahmanian. 2024. Higher layers need more lora experts. *arXiv preprint arXiv:2402.08562*.
- Antonio Andrea Gargiulo, Donato Crisostomi, Maria Sofia Bucarelli, Simone Scardapane, Fabrizio Silvestri, and Emanuele Rodolà. 2024a. Task singular vectors: Reducing task interference in model merging. *arXiv preprint arXiv:2412.00081*.
- Antonio Andrea Gargiulo, Donato Crisostomi, Maria Sofia Bucarelli, Simone Scardapane, Fabrizio Silvestri, and Emanuele Rodolà. 2024b. Task singular vectors: Reducing task interference in model merging. *arXiv preprint arXiv:2412.00081*.
- Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, and 1 others. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.
- Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2021. Measuring massive multitask language understanding. In *International Conference on Learning Representations*.
- Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yanzhi Li, Shean Wang, Lu Wang, Weizhu Chen, and 1 others. 2022. Lora: Low-rank adaptation of large language models. *ICLR*, 1(2):3.
- Xiaolin Hu, Xiang Cheng, Peiyu Liu, Wei Liu, Jian Luan, Bin Wang, and Yong Liu. 2024. Dota: Weight-decomposed tensor adaptation for large language models. *arXiv preprint arXiv:2412.20891*.
- Chengsong Huang, Qian Liu, Bill Yuchen Lin, Tianyu Pang, Chao Du, and Min Lin. 2024a. Lorahub: Efficient cross-task generalization via dynamic lora composition. In *First Conference on Language Modeling*.
- Chenyu Huang, Peng Ye, Tao Chen, Tong He, Xiangyu Yue, and Wanli Ouyang. 2024b. Emr-merging: Tuning-free high-performance model merging. *Advances in Neural Information Processing Systems*, 37:122741–122769.

- Gabriel Ilharco, Marco Tulio Ribeiro, Mitchell Wortsman, Suchin Gururangan, Ludwig Schmidt, Hananeh Hajishirzi, and Ali Farhadi. 2022. Editing models with task arithmetic. *arXiv preprint arXiv:2212.04089*.
- Xisen Jin, Xiang Ren, Daniel Preotiuc-Pietro, and Pengxiang Cheng. 2022. Dataless knowledge fusion by merging weights of language models. *arXiv preprint arXiv:2212.09849*.
- Dengchun Li, Yingzi Ma, Naizheng Wang, Zhengmao Ye, Zhiyuan Cheng, Yinghao Tang, Yan Zhang, Lei Duan, Jie Zuo, Cal Yang, and 1 others. 2024. Mixlora: Enhancing large language models fine-tuning with lora-based mixture of experts. *arXiv preprint arXiv:2404.15159*.
- Yifan Li, Yifan Du, Kun Zhou, Jinpeng Wang, Xin Zhao, and Ji-Rong Wen. 2023. Evaluating object hallucination in large vision-language models. In *The 2023 Conference on Empirical Methods in Natural Language Processing*.
- Stephanie Lin, Jacob Hilton, and Owain Evans. 2022. Truthfulqa: Measuring how models mimic human falsehoods. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3214–3252.
- Shih-Yang Liu, Chien-Yi Wang, Hongxu Yin, Pavlo Molchanov, Yu-Chiang Frank Wang, Kwang-Ting Cheng, and Min-Hung Chen. 2024a. Dora: Weight-decomposed low-rank adaptation. In *Forty-first International Conference on Machine Learning*.
- Yuan Liu, Haodong Duan, Yuanhan Zhang, Bo Li, Songyang Zhang, Wangbo Zhao, Yike Yuan, Jiaqi Wang, Conghui He, Ziwei Liu, and 1 others. 2024b. Mmbench: Is your multi-modal model an all-around player? In *European conference on computer vision*, pages 216–233. Springer.
- Pan Lu, Hritik Bansal, Tony Xia, Jiacheng Liu, Chunyuan Li, Hannaneh Hajishirzi, Hao Cheng, Kai-Wei Chang, Michel Galley, and Jianfeng Gao. 2024a. Mathvista: Evaluating mathematical reasoning of foundation models in visual contexts. In *The Twelfth International Conference on Learning Representations*.
- Zhenyi Lu, Chenghao Fan, Wei Wei, Xiaoye Qu, Danyang Chen, and Yu Cheng. 2024b. Twin-merging: Dynamic integration of modular expertise in model merging. *arXiv preprint arXiv:2406.15479*.
- Michael S Matena and Colin A Raffel. 2022. Merging models with fisher-weighted averaging. *Advances in Neural Information Processing Systems*, 35:17703–17716.
- Ziheng Ouyang, Zhen Li, and Qibin Hou. 2025. K-lora: Unlocking training-free fusion of any subject and style loras. *arXiv preprint arXiv:2502.18461*.
- Alicia Parrish, Angelica Chen, Nikita Nangia, Vishakh Padmakumar, Jason Phang, Jana Thompson, Phu Mon Htut, and Samuel Bowman. 2022. Bbq: A hand-built bias benchmark for question answering. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 2086–2105.
- Akshara Prabhakar, Yuanzhi Li, Karthik Narasimhan, Sham Kakade, Eran Malach, and Samy Jelassi. 2025. Lora soups: Merging loras for practical skill composition tasks. In *Proceedings of the 31st International Conference on Computational Linguistics: Industry Track*, pages 644–655.
- Biqing Qi, Fangyuan Li, Zhen Wang, Junqi Gao, Dong Li, Peng Ye, and Bowen Zhou. 2024. Less is more: Efficient model merging with binary task switch. *arXiv preprint arXiv:2412.00054*.
- Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of machine learning research*, 21(140):1–67.
- Donald Shenaj, Ondrej Bohdal, Mete Ozay, Pietro Zanuttigh, and Umberto Michieli. 2024. Lora-rar: Learning to merge loras via hypernetworks for subject-style conditioned image generation. *arXiv preprint arXiv:2412.05148*.
- George Stoica, Pratik Ramesh, Boglarka Ecsedi, Leshem Choshen, and Judy Hoffman. 2025. Model merging with svd to tie the knots. In *The Thirteenth International Conference on Learning Representations*.
- Anke Tang, Li Shen, Yong Luo, Han Hu, Bo Du, and Dacheng Tao. 2024. Fusionbench: A comprehensive benchmark of deep model fusion. *arXiv preprint arXiv:2406.03280*.
- Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, and 1 others. 2023. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*.
- Peng Wang, Shuai Bai, Sinan Tan, Shijie Wang, Zhihao Fan, Jinze Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Yang Fan, Kai Dang, Mengfei Du, Xuancheng Ren, Rui Men, Dayiheng Liu, Chang Zhou, Jingren Zhou, and Junyang Lin. 2024. Qwen2-vl: Enhancing vision-language model’s perception of the world at any resolution. *arXiv preprint arXiv:2409.12191*.
- T Wolf. 2019. Huggingface’s transformers: State-of-the-art natural language processing. *arXiv preprint arXiv:1910.03771*.
- Xun Wu, Shaohan Huang, and Furu Wei. 2024. Mixture of lora experts. In *The Twelfth International Conference on Learning Representations*.

- xAI. 2025. [Grok 3: The most powerful AI in the world is here.](#)
- Feng Xiong, Runxi Cheng, Wang Chen, Zhanqiu Zhang, Yiwen Guo, Chun Yuan, and Ruifeng Xu. 2024. Multi-task model merging via adaptive weight disentanglement. *arXiv preprint arXiv:2411.18729*.
- Prateek Yadav, Derek Tam, Leshem Choshen, Colin A Raffel, and Mohit Bansal. 2024. Ties-merging: Resolving interference when merging models. *Advances in Neural Information Processing Systems*, 36.
- Enneng Yang, Li Shen, Guibing Guo, Xingwei Wang, Xiaochun Cao, Jie Zhang, and Dacheng Tao. 2024a. Model merging in llms, mllms, and beyond: Methods, theories, applications and opportunities. *arXiv preprint arXiv:2408.07666*.
- Enneng Yang, Zhenyi Wang, Li Shen, Shiwei Liu, Guibing Guo, Xingwei Wang, and Dacheng Tao. 2024b. Adamerging: Adaptive model merging for multi-task learning. In *The Twelfth International Conference on Learning Representations*.
- Yang Yang, Wen Wang, Liang Peng, Chaotian Song, Yao Chen, Hengjia Li, Xiaolong Yang, Qinglin Lu, Deng Cai, Boxi Wu, and 1 others. 2024c. Lora-composer: Leveraging low-rank adaptation for multi-concept customization in training-free diffusion models. *arXiv preprint arXiv:2403.11627*.
- Peng Ye, Tong He, Baopu Li, Tao Chen, Lei Bai, and Wanli Ouyang. 2025a. beta-darts++: Bi-level regularization for proxy-robust differentiable architecture search. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.
- Peng Ye, Tong He, Shengji Tang, Baopu Li, Tao Chen, Lei Bai, and Wanli Ouyang. 2025b. Stimulative training++: Go beyond the performance limits of residual networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 47(8):6463–6479.
- Peng Ye, Chenyu Huang, Mingzhu Shen, Tao Chen, Yongqi Huang, and Wanli Ouyang. 2025c. Dynamic model merging with mixture of weights. *IEEE Transactions on Circuits and Systems for Video Technology*, 35(8):7925–7939.
- Le Yu, Bowen Yu, Haiyang Yu, Fei Huang, and Yongbin Li. 2024a. Extend model merging from fine-tuned to pre-trained large language models via weight disentanglement. *arXiv preprint arXiv:2408.03092*.
- Le Yu, Bowen Yu, Haiyang Yu, Fei Huang, and Yongbin Li. 2024b. Language models are super mario: Absorbing abilities from homologous models as a free lunch. In *Forty-first International Conference on Machine Learning*.
- Fanhu Zeng, Haiyang Guo, Fei Zhu, Li Shen, and Hao Tang. 2025. Parameter efficient merging for multimodal large language models with complementary parameter adaptation. *arXiv preprint arXiv:2502.17159*.
- Juzheng Zhang, Jiacheng You, Ashwinee Panda, and Tom Goldstein. 2025a. Lori: Reducing cross-task interference in multi-task low-rank adaptation. *arXiv preprint arXiv:2504.07448*.
- Yiqun Zhang, Peng Ye, Xiaocui Yang, Shi Feng, Shufei Zhang, Lei Bai, Wanli Ouyang, and Shuyue Hu. 2025b. Nature-inspired population-based evolution of large language models. *arXiv preprint arXiv:2503.01155*.
- Yuxuan Zhang and Ruizhe Li. 2024. Dlp-lora: Efficient task-specific lora fusion with a dynamic, lightweight plugin for large language models. *arXiv preprint arXiv:2410.01497*.
- Xinyu Zhao, Guoheng Sun, Ruisi Cai, Yukun Zhou, Pingzhi Li, Peihao Wang, Bowen Tan, Yexiao He, Li Chen, Yi Liang, and 1 others. 2024a. Model-glue: Democratized llm scaling for a large model zoo in the wild. *Advances in Neural Information Processing Systems*, 37:13349–13371.
- Ziyu Zhao, Tao Shen, Didi Zhu, Zexi Li, Jing Su, Xuwu Wang, Kun Kuang, and Fei Wu. 2024b. [Merging loras like playing LEGO: pushing the modularity of lora to extremes through rank-wise clustering.](#) *CoRR*, abs/2409.16167.
- Shenghe Zheng and Hongzhi Wang. 2024. Free-merging: Fourier transform for model merging with lightweight experts. *arXiv preprint arXiv:2411.16815*.
- Yu Zhou, Xingyu Wu, Jibin Wu, Liang Feng, and Kay Chen Tan. 2024. Hm3: Hierarchical multi-objective model merging for pretrained models. *arXiv preprint arXiv:2409.18893*.

Appendix

A Notations

Tab. 11 presents the main notations used in this paper and their corresponding meanings. It should be noted that, for simplicity in explanation and proof, the layer index k is sometimes omitted when it does not affect the clarity of the discussion or derivation.

B Proofs and Derivations

In this section, we analyze the validity of the assumptions made in the main paper and provide detailed proofs and derivations.

B.1 Assumption 3.1

In this section, we discuss the validity of Assumption 3.1. Suppose the matrices to be merged are $\{W_i\}_{i=1}^n \in \mathbb{R}^{m \times d}$, and the merged result is W . Let the input data be X . Then for all i , the merging objective is:

$$\begin{aligned} \min \|X^T W - X^T W_i\|_F^2 \\ = \min \text{tr}((W - W_i)^T X X^T (W - W_i)). \end{aligned} \quad (6)$$

In a data-free setting where X is unavailable, we approximate $XX^T \approx I$, which is reasonable because in high-dimensional spaces the empirical covariance of i.i.d. data concentrates around the identity when the sample size is sufficiently large. This yields the simplified proxy objective $\min \|W - W_i\|_F^2$. The overall optimization objective for the merging process is:

$$\min T = \min \sum_{i=1}^n \eta_i \|W - W_i\|_F^2 \quad (7)$$

where η_i are tunable weights that align the overall optimization goal with each individual W_i 's objective. Since modules with smaller magnitudes are more easily distorted during merging, they should receive higher weights. A natural choice is:

$$\eta_i = \frac{\sum_{j=1}^n \|\alpha_j\|_2}{\|\alpha_i\|_2} \quad (8)$$

where $\alpha_i \in \mathbb{R}^{1 \times d}$ is the magnitude vector from the decoupling $W_i = \bar{W}_i \text{diag}(\alpha_i)$, and $\bar{W}_i$ has unit-norm columns. This weighting assigns higher importance to tasks with smaller magnitudes, compensating for their susceptibility to being dominated during merging. This yields the loss function

in Assumption 3.1:

$$L = \sum_{i=1}^n \frac{\sum_{j=1}^n \|\alpha_j\|_2}{\|\alpha_i\|_2} \|W - W_i\|_F^2. \quad (9)$$

Justification for the weighting scheme. Our choice of $\eta_i \propto 1/\|\alpha_i\|_2$ can also be motivated from a relative-error perspective. In the unweighted case ($\eta_i = 1$), the loss is dominated by large-magnitude modules since $\|W - W_i\|_F^2$ scales with $\|\alpha_i\|_2^2$. Our weighting normalizes each term by $\|\alpha_i\|_2$, effectively measuring reconstruction quality relative to each module's own scale. This prevents the optimizer from ignoring small-magnitude modules, which is precisely the failure mode identified in Sec. 3.2. Additionally, the specific form $\eta_i = \sum_j \|\alpha_j\|_2 / \|\alpha_i\|_2$ ensures the loss is permutation-symmetric across tasks (reordering tasks does not change the loss) and reduces to the uniform case $\eta_i = n$ when all magnitudes are equal, providing a natural transition to standard averaging.

Validity of the uncorrelated assumption. We assume that direction columns from different tasks are uncorrelated, i.e., $\mathbb{E}[\bar{W}_i^T \bar{W}_l] = 0$ for $i \neq l$. This is justified by the fact that LoRA modules are fine-tuned independently on different tasks, resulting in distinct parameter subspaces. Moreover, in high-dimensional spaces ($m \gg 1$), randomly oriented unit vectors are nearly orthogonal with high probability, so independent fine-tuning naturally produces approximately uncorrelated direction vectors.

Validity of the zero-mean assumption. We assume that the entries of each direction column $\bar{W}_i$ have zero mean. Since $\bar{W}_i$ is a unit-norm vector in $\mathbb{R}^m$ obtained by normalizing a fine-tuned parameter column, its entries are distributed symmetrically around zero when m is large. Empirically, the mean of entries in LoRA direction columns is negligible compared to their standard deviation ($\sim 1/\sqrt{m}$), supporting this assumption.

B.2 Proof of Theorem 3.1

We prove Theorem 3.1 for the case of two task vectors ($n = 2$); the result extends naturally to multiple tasks.

Setup. Let $W_1 = \bar{W}_1 \text{diag}(\alpha_1)$ and $W_2 = \bar{W}_2 \text{diag}(\alpha_2)$, where $\bar{W}_i \in \mathbb{R}^{m \times d}$ has unit-norm columns and $\alpha_i \in \mathbb{R}^{1 \times d}$ with $\alpha_{ij} \geq 0$. Let $a = \|\alpha_1\|_2$, $b = \|\alpha_2\|_2$, and $S = \sum_{j=1}^d \alpha_{1j} \alpha_{2j}$.

Non-decoupled merge: $W^1 = \frac{1}{2}(W_1 + W_2)$.

Table 11: Notations.

Notations	Descriptions
θ_{pre}	Pre-trained model parameters.
Δ_i	Fine-tuning parameters for task i .
W_{pre}^k	The k -th layer of fine-tuning parameters for task i .
W_i^k	The k -th layer of fine-tuning parameters for task i .
A_i, B_i	The k -th layer of LoRA low-rank parameters for task i .
$\hat{W}_i^k$	The k -th layer of fine-tuning parameters for task i after orthogonalization.
$\overline{W}_i^k$	The k -th layer of fine-tuning direction parameters for task i after decoupling.
α_i^k	The k -th layer of fine-tuning magnitude parameters for task i after decoupling.
W_{ij}	The row- i and col- j of a parameter matrix.

Decoupled merge: $W^2 = \frac{1}{2}(\overline{W}_1 + \overline{W}_2) \cdot \frac{1}{2} \text{diag}(\alpha_1 + \alpha_2)$.

Using unit-norm columns and the uncorrelated assumption:

Step 1: Compute $\mathbb{E}(L_1)$.

For the non-decoupled case:

$$\mathbb{E}[\|W^2 - W_1\|_F^2] = \sum_{j=1}^d [(\frac{\beta_j}{2} - \alpha_{1j})^2 + \frac{\beta_j^2}{4}] \quad (16)$$

$$\begin{aligned} \|W^1 - W_1\|_F^2 &= \|\frac{1}{2}(W_2 - W_1)\|_F^2 \\ &= \frac{1}{4} \sum_{j=1}^d \|\alpha_{2j} \overline{W}_{2[:,j]} - \alpha_{1j} \overline{W}_{1[:,j]}\|_2^2 \end{aligned} \quad (10)$$

$$\begin{aligned} &= \frac{1}{4} \sum_{j=1}^d [\underbrace{\alpha_{2j}^2 \|\overline{W}_{2[:,j]}\|_2^2}_{=1} + \underbrace{\alpha_{1j}^2 \|\overline{W}_{1[:,j]}\|_2^2}_{=1} \\ &\quad - 2\alpha_{1j}\alpha_{2j} \overline{W}_{1[:,j]}^T \overline{W}_{2[:,j]}] \end{aligned} \quad (11)$$

Substituting $\beta_j = \frac{1}{2}(\alpha_{1j} + \alpha_{2j})$ and expanding:

$$\begin{aligned} &= \sum_{j=1}^d \frac{1}{16} [(\alpha_{2j} - 3\alpha_{1j})^2 + (\alpha_{1j} + \alpha_{2j})^2] \\ &= \sum_{j=1}^d \frac{1}{16} [2\alpha_{2j}^2 - 4\alpha_{1j}\alpha_{2j} + 10\alpha_{1j}^2] \end{aligned} \quad (17)$$

By Assumption 3.1, $\mathbb{E}[\overline{W}_{1[:,j]}^T \overline{W}_{2[:,j]}] = 0$ (uncorrelated directions). Thus:

$$\mathbb{E}[\|W^1 - W_1\|_F^2] = \frac{1}{4} \sum_j (\alpha_{1j}^2 + \alpha_{2j}^2) = \frac{1}{4}(a^2 + b^2) \quad (12)$$

By symmetry, $\mathbb{E}[\|W^1 - W_2\|_F^2] = \frac{1}{4}(a^2 + b^2)$. Therefore:

$$\begin{aligned} \mathbb{E}(L_1) &= \frac{a+b}{a} \cdot \frac{a^2+b^2}{4} + \frac{a+b}{b} \cdot \frac{a^2+b^2}{4} \\ &= \frac{(a+b)^2(a^2+b^2)}{4ab} \end{aligned} \quad (13)$$

Step 2: Compute $\mathbb{E}(L_2)$.

For the decoupled case, define $\beta_j = \frac{1}{2}(\alpha_{1j} + \alpha_{2j})$:

$$\begin{aligned} W^2 - W_1 &= \beta_j(\overline{W}_{1[:,j]} + \overline{W}_{2[:,j]})/2 \\ &\quad - \alpha_{1j} \overline{W}_{1[:,j]} \quad (\text{column } j) \end{aligned} \quad (14)$$

$$= (\frac{\beta_j}{2} - \alpha_{1j}) \overline{W}_{1[:,j]} + \frac{\beta_j}{2} \overline{W}_{2[:,j]} \quad (15)$$

Similarly:

$$\begin{aligned} \mathbb{E}[\|W^2 - W_2\|_F^2] &= \sum_{j=1}^d \frac{1}{16} [10\alpha_{2j}^2 \\ &\quad - 4\alpha_{1j}\alpha_{2j} + 2\alpha_{1j}^2] \end{aligned} \quad (18)$$

Substituting $\sum_j \alpha_{1j}^2 = a^2$, $\sum_j \alpha_{2j}^2 = b^2$, and $\sum_j \alpha_{1j}\alpha_{2j} = S$:

$$\mathbb{E}[\|W^2 - W_1\|_F^2] = \frac{1}{16}(10a^2 - 4S + 2b^2) \quad (19)$$

$$\mathbb{E}[\|W^2 - W_2\|_F^2] = \frac{1}{16}(2a^2 - 4S + 10b^2) \quad (20)$$

Therefore:

$$\begin{aligned} \mathbb{E}(L_2) &= \frac{a+b}{a} \cdot \frac{10a^2 - 4S + 2b^2}{16} \\ &\quad + \frac{a+b}{b} \cdot \frac{2a^2 - 4S + 10b^2}{16} \\ &= \frac{(a+b)}{16ab} [b(10a^2 - 4S + 2b^2) \\ &\quad + a(2a^2 - 4S + 10b^2)] \\ &= \frac{(a+b)^2}{16ab} [2(a^2 - ab + b^2) + 10ab - 4S] \\ &= \frac{(a+b)^2(a^2 + 4ab + b^2 - 2S)}{8ab} \end{aligned} \quad (21)$$

where the third equality uses $a^3 + b^3 = (a+b)(a^2 - ab + b^2)$.

Step 3: Compare $\mathbb{E}(L_1)$ and $\mathbb{E}(L_2)$.

Define $f = \mathbb{E}(L_1) - \mathbb{E}(L_2)$. Substituting the results from Steps 1 and 2:

$$\begin{aligned} f &= \frac{(a+b)^2(a^2+b^2)}{4ab} - \frac{(a+b)^2(a^2+4ab+b^2-2S)}{8ab} \\ &= \frac{(a+b)^2}{8ab} [2(a^2+b^2) - (a^2+4ab+b^2-2S)] \\ &= \frac{(a+b)^2}{8ab} [(a-b)^2 - 2(ab-S)] \quad (22) \end{aligned}$$

which proves Eq. 3. $\square$

Corollary (Sufficient condition). Define the magnitude ratio $r = \frac{\max(a,b)}{\min(a,b)} \geq 1$ and the magnitude correlation $\rho = \frac{S}{ab} \in [0, 1]$ (bounded by Cauchy-Schwarz). Rewriting f using ρ :

$$f = \frac{(a+b)^2}{8ab} [(a-b)^2 - 2ab(1-\rho)] \quad (23)$$

so $f \geq 0$ if and only if $\rho \geq 1 - \frac{(a-b)^2}{2ab} = 1 - \frac{(r-1)^2}{2r}$ (using $r = \max(a,b)/\min(a,b)$).

When $r \geq 2 + \sqrt{3}$, the right-hand side is non-positive, so $f \geq 0$ holds for all $\rho \geq 0$. To verify: the condition $1 - \frac{(r-1)^2}{2r} \leq 0$ requires $(r-1)^2 \geq 2r$, i.e., $r^2 - 4r + 1 \geq 0$, whose positive root is $r = 2 + \sqrt{3} \approx 3.73$.

Special case: proportional magnitudes. When $\alpha_2 \propto \alpha_1$ (i.e., $\alpha_{2j} = c\alpha_{1j}$ for some $c > 0$), we have $S = c \sum_j \alpha_{1j}^2 = ca^2 = ab$, so $\rho = 1$ and the loss gap reduces to $f = \frac{(a+b)^2(a-b)^2}{8ab}$, which is strictly positive whenever $a \neq b$.

Extension to $n > 2$. The proof above is stated for two tasks ($n = 2$). For $n > 2$, the loss L in Eq. 2 decomposes into contributions from each task W_i versus the merged model W . The same pairwise analysis applies: for each pair (i, j) , the decoupled merge reduces the expected loss by f_{ij} . Since the total loss is a sum of such pairwise contributions, decoupling improves overall merging quality when the aggregate gap $\sum_{i < j} f_{ij} > 0$ —a property we verify empirically in Tab. 20.

B.3 Remark on Orthogonalization

We elaborate on the two claims in Remark 3.2.

Part 1: Projection decomposition.

The projection of the merged direction $\bar{W}_{\text{merge}} = \sum_{j=1}^n \bar{W}_j$ onto task i 's direction matrix is:

$$\begin{aligned} \text{tr}(\bar{W}_i^T \bar{W}_{\text{merge}}) &= \text{tr}(\bar{W}_i^T \sum_{j=1}^n \bar{W}_j) \\ &= \text{tr}(\bar{W}_i^T \bar{W}_i) + \sum_{j \neq i} \text{tr}(\bar{W}_i^T \bar{W}_j) \quad (24) \end{aligned}$$

$$= \|\bar{W}_i\|_F^2 + \mathcal{I}_i \quad (25)$$

where $\mathcal{I}_i = \sum_{j \neq i} \text{tr}(\bar{W}_i^T \bar{W}_j)$ is the inter-task interference for task i . The term $\|\bar{W}_i\|_F^2$ represents the desired self-contribution that faithfully preserves task i 's directional information. The interference $\mathcal{I}_i$ can be positive (over-amplification of shared directions) or negative (destructive cancellation of task-specific features), both distorting the merged representation. When $\bar{W}_i^T \bar{W}_j = 0$ for all $j \neq i$, we have $\mathcal{I}_i = 0$, and the merged direction perfectly preserves each task's contribution. $\square$

Part 2: L_o reduces interference while L_r bounds perturbation.

We show that minimizing $L_o = \sum_{i < j} \|(W_i + \delta_i)^T (W_j + \delta_j)\|_F^2$ directly reduces the cross-task correlations that constitute the interference terms. We demonstrate the mechanism for the case $n = 2$; the general case follows by linearity of the gradient over all pairs.

For $n = 2$, writing

$$\begin{aligned} L_o &= \|(W_1 + \delta_1)^T (W_2 + \delta_2)\|_F^2 \\ &= \text{tr}((W_2 + \delta_2)^T (W_1 + \delta_1) \\ &\quad (W_1 + \delta_1)^T (W_2 + \delta_2)), \quad (26) \end{aligned}$$

the gradient with respect to δ_2 at $\delta_1 = \delta_2 = 0$ is:

$$\nabla_{\delta_2} L_o|_{\delta=0} = 2W_1 W_1^T W_2 \quad (27)$$

Gradient descent yields the update $\delta_2 = -\eta_r \cdot 2W_1 W_1^T W_2$, and:

$$\begin{aligned} L_o(\delta_2) &= L_o(0) - 4\eta_r \|W_1 W_1^T W_2\|_F^2 \\ &\quad + O(\eta_r^2) \quad (28) \end{aligned}$$

Since $\|W_1 W_1^T W_2\|_F^2 \geq 0$, the loss strictly decreases whenever $W_1^T W_2 \neq 0$, directly reducing the cross-task correlation $\|W_1^T W_2\|_F^2$ and therefore the interference $\mathcal{I}_i$.

The regularization term $L_r = \eta \sum_i \|\delta_i\|_F^2$ ensures that each perturbation remains bounded. The balance between L_o and L_r is controlled by η :

larger η produces smaller perturbations (less impact on individual tasks) at the cost of less orthogonalization, while smaller η allows more aggressive orthogonalization. In practice, the high redundancy of fine-tuned parameters (Yu et al., 2024b) ensures that moderate perturbations have negligible impact on task performance, as confirmed empirically in Fig. 3(a). $\square$

C Reproducibility

C.1 Code

The code is provided in an anonymous format.¹

C.2 Datasets

Merging 8 ViTs. We use ViT-B/32 and ViT-L/14 as pre-trained models, and fine-tune them on 8 image classification datasets (SUN397, Cars, RESISC45, EuroSAT, SVHN, GTSRB, MNIST, and DTD), then merge them and test their performance. Configuration details follow (Huang et al., 2024b).

Merging Medium-sized Language Models. We use T5 as the pre-trained model, fine-tune it on 8 classification task datasets from GLUE benchmark for merging, including CoLA, SST-2, MRPC, STS-B, QQP, MNLI, QNLI, and RTE. CoLA is evaluated with the Matthews correlation coefficient, STS-B with the average of the Pearson and Spearman correlation coefficients, and the others by accuracy. Details follow (Huang et al., 2024b).

Merging Large Language Models. In this section, we conduct two experiments. The first uses LLaMA3-8B as the pre-trained model and LoRA as the PEFT method, fine-tuned and merged on SNLI, MNLI, SICK, QNLI, RTE, SCITAIL. Details can be found in (Stoica et al., 2025). The second part uses Qwen-14B as the pre-trained model and LoRA as the PEFT method. We fine-tune and merge on three generative tasks: MMLU, TruthfulQA, and BBQ. Configuration details can be found in (Lu et al., 2024b).

Merging Multi-Modal Models. In this part, we select Qwen2-VL as the base model and evaluate it on five multimodal tasks: POPE, MMStar, MM-Bench, MathVista, and RealWorldQA. We also use open-source weights for model merging. The open-source weights are available at provided link². It is

important to note that these weights are not released by this work, but are sourced from existing models on HuggingFace, in compliance with double-blind review requirements.

C.3 Baselines

Finetune. Solve each task with its fine-tuned model, but requires storing separate models for each task, leading to significant storage overhead.

Task Arithmetic (Ilharco et al., 2022). Define task vectors as the merging target. For task k , the task vector is defined as $v_k = \theta_k - \theta_{pre}$, where θ_{pre} is the pre-trained model parameters, and θ_k is the fine-tuned parameters for task k . The merging process can be represented as $\theta_m = \theta_{pre} + \lambda \sum_{i=1}^K v_i$, where λ is the merging coefficient. This method suffers significant performance degradation due to unaddressed task conflicts. For LoRA, the task vector refers to the matrix constructed by LoRA.

Ties-Merging (Yadav et al., 2024). Attempts to resolve parameter conflicts during merging by eliminating redundancy and sign conflicts. However, resolving parameter conflicts is insufficient to address task conflicts, resulting in performance loss.

Breadcrumbs (Davari and Belilovsky, 2023). Discards parameters with the largest and smallest absolute values as redundant, negatively impacting model merging. This approach is simple, but it shows a great performance decline on certain tasks.

PCB-Merging (Du et al., 2024). Uses internal balancing to measure parameter importance within tasks and mutual balancing to assess parameter similarity across tasks, discarding redundant parameters and adjusting merging coefficients. It requires considerable computational resources.

AdaMerging (Yang et al., 2024b). Uses an unsupervised approach to learn the merging coefficient for each task vector or layer. AdaMerging++ additionally applies Ties-Merging before calculating the merging coefficient. This method is limited to classification tasks and requires certain training resources, making it unsuitable for edge deployment.

DARE (Yu et al., 2024b). DARE randomly discards a large portion of task vector parameters, potentially reducing parameter interference among models. This method is simple, but due to the lack

¹<https://anonymous.4open.science/t/DO-Merging-review-F06D>

²POPE: <https://huggingface.co/jpark677/qwen2-vl-7b-instruct-pope-lora-ep-3-waa-f>;
MMStar: <https://huggingface.co/jpark677/qwen2-vl-7b-instruct-mmstar-lora-unfreeze-vision-ep-3-waa-f>;

MMBench: <https://huggingface.co/jpark677/qwen2-vl-7b-instruct-mmbench-dev-lora-ep-3-waa-f>;
MathVista: <https://huggingface.co/jpark677/qwen2-vl-7b-instruct-mathvista-lora-ep-3-waa-f> ;
RealWorldQA: <https://huggingface.co/jpark677/qwen2-vl-7b-instruct-realworldqa-lora-ep-3-waa-f>

of further optimization in the merging, it suffers from significant performance degradation.

TSVM (Gargiulo et al., 2024a). The motivation of this method lies in the effectiveness of applying whitening constraints on the SVD components of the task vector before merging. In essence, this serves as a form of orthogonal constraint. However, unlike our approach, this constraint does not guarantee the preservation of the original capabilities under all circumstances. It may affect the model’s initial performance and lead to a decline in merging quality. In contrast, the orthogonal method we propose offers stronger generalization ability.

KNOTS (Stoica et al., 2025). This is a method specifically designed for LoRA merging. The key observation is that, compared to full fine-tuning, LoRA produces parameter features with greater discrepancies. To address this, the paper proposes merging LoRA models in a shared SVD space, which helps bring their feature outputs closer together. However, this approach does not account for the fact that the issue of uneven distribution in the original space still exists in the shared space. As a result, it fails to address the performance loss caused by the parameter magnitude distribution issues we identify, leading to only limited improvements in merging performance.

COPA-Merging (Zeng et al., 2025). This is another method designed specifically for LoRA merging. Its main idea is based on the observation that the parameter distributions of LoRA’s A and B matrices exhibit different characteristics. Accordingly, it automatically computes normalization coefficients during merging based on these differences and incorporates them into the computation of merged parameters. While this approach identifies the distributional differences between the A and B matrices in LoRA, it does not further address the more fundamental issue that the distribution gap between LoRA and full fine-tuning which leads to suboptimal performance in LoRA merging. In contrast, our work tackles the root cause of the poor performance in LoRA merging.

C.4 Experiment Details

In this section, we discuss the details of the experiments. All experiments are conducted on a single NVIDIA A800 GPU.

Orthogonalization Optimization. For the orthogonalization step (Eq. 4), we optimize the perturbations $\{\delta_i\}$ using Adam with a learning rate of 1×10^{-3} for 200 gradient descent steps per layer.

The regularization coefficient is set to $\eta = 1$ by default. The optimization is performed independently for each layer and is highly efficient: it takes less than 2 minutes in total for an 8B-parameter model on a single GPU, as the layer-wise formulation keeps memory and compute costs low.

C.5 Computational Details

Magnitude Distribution Variance. Assume there are n fine-tuned models $\{\theta_i\}_{i=1}^n$, each containing K layers. The k -th layer of the i -th model is denoted as W_i^k . The Magnitude Distribution Variance v is then defined as:

$$v = \sum_{k=1}^K \text{var}(|W_i^k|_{i=1}^n), \quad (29)$$

where $\text{var}(\cdot)$ denotes the variance operator.

Norm Average Accuracy. Assume we have n fine-tuned models corresponding to n tasks $\{t_i\}_{i=1}^n$, and the performance of these fine-tuned models on their respective tasks is $\{a_i\}_{i=1}^n$. Let the performance of the merged model on these n tasks be $\{\hat{a}_i\}_{i=1}^n$. Then, the normalized average performance (acc_n) of the merged model is defined as:

$$acc_n = \frac{\sum_{i=1}^n \hat{a}_i}{\sum_{i=1}^n a_i}. \quad (30)$$

D More Discussions

In this part, we provide a more detailed explanation of our method. In Sec. D.1, we introduce the key innovations of our approach in greater detail. In Sec. D.2 and Sec. D.3, we analyze why we choose model merging instead of other techniques to enhance LoRA’s multi-task capabilities. In Sec. D.4, we explain why we merge the product matrix of LoRA’s two low-rank matrices instead of merging the matrices directly.

D.1 Novelty of DO-Merging

In this section, we analyze the key innovations of our proposed DO-Merging method, which are summarized in three aspects.

Observation. As shown in Fig. 1, we observe that LoRA modules trained on different tasks exhibit significantly larger variations in parameter magnitude compared to full fine-tuning. We further establish a connection between this observation and the suboptimal performance of existing merging methods on LoRA, both empirically and theoretically. **To the best of our knowledge, this is the**

first work that analyzes the parameter difference between LoRA and full fine-tuning from this perspective and proposes targeted improvements for LoRA merging.

Decouple. Based on the phenomenon, we demonstrate the correlation between parameter distribution and merging performance. **We argue that effective model merging should preserve both the shape and scale of parameter distributions.** When the scale differences are large, parameters with larger magnitudes tend to dominate the merged result, leading to information loss from smaller-scale parameters and thus performance degradation. To address this, we propose a decoupling strategy. **Unlike DoRA (Liu et al., 2024a), which reparameterizes magnitude and direction during training, and DoTA (Hu et al., 2024), which uses decoupled components to compute merging coefficients, our approach applies decomposition post-hoc to existing LoRA weights without imposing constraints on the training process.** This allows us to make full use of publicly available LoRA weights and improves usability. Compared to coefficient-based approaches, we explore how weight decoupling can be effectively applied during the merging process, supported by theoretical analysis and empirical results.

Orthogonalize. Our orthogonalization strategy aims to reduce task interference during the merging of direction vectors. Prior works have shown that enforcing orthogonality among different LoRA modules during training can improve merging performance (Zhang et al., 2025a). However, since most publicly available LoRA weights do not satisfy such constraints, **we propose an orthogonalization method applicable to all pre-trained weights.** To meet the efficiency needs of diverse users, we introduce a **data-free and layer-wise** orthogonalization approach. Unlike prior methods, we further incorporate constraints on parameter variation ranges to ensure that the orthogonalization process does not harm the model’s original task performance or erase task-specific features. Meanwhile, unlike other orthogonalization methods applied to pretrained weights (Gargiulo et al., 2024a), DO-Merging offers much stronger generalization: **it consistently preserves single-task performance while improving orthogonality across all settings**, which is crucial for reliable model merging. We also provide rigorous theoretical guarantees and comprehensive experimental validation for the proposed method.

In summary, we begin by identifying the parameter distribution discrepancy between LoRA and full fine-tuning, and establish its link to the poor performance of LoRA merging through both theory and experiments. To address this issue, we innovatively propose a method that decouples the direction and magnitude of parameter updates. Furthermore, to reduce task interference, we introduce a data-free orthogonalization scheme. For each component of our method, we provide solid theoretical analysis and extensive experimental verification. Therefore, we believe that our DO-Merging method offers a promising and innovative solution for model merging in the LoRA domain.

D.2 Why not LoRA-MoE?

LoRA-MoE (Wu et al., 2024; Dou et al., 2023; Zhang and Li, 2024; Zhao et al., 2024a) is a method that combines multiple LoRA weights using a router, which directs incoming inputs to the corresponding LoRA based on the task. The key advantage of this approach is that, as long as the router performs well, each task-specific LoRA can retain its performance on the corresponding task. However, compared to the model merging approach studied in this paper, LoRA-MoE suffers from two major issues that hinder its practical deployment:

Usability. LoRA-MoE inevitably requires modifications to the model architecture. At a minimum, adding a router and supporting multiple LoRA modules. This poses a significant barrier for users without deep learning expertise who still wish to apply deep learning techniques to their problems. Even for those who are capable of modifying the model, distributing or further fine-tuning the model becomes much more complex. In contrast, the model merging methods we focus on do not alter the model structure and result in a single, standard model. This makes it significantly easier for users to deploy, distribute, and continue training. In today’s context, usability plays a critical role in determining the practical value of a method.

Cost. LoRA-MoE requires training the router, which involves a substantial amount of data and computational resources for both forward and backward passes. When dealing with large models such as 32B or 70B variants, this training becomes practically infeasible for average users due to resource constraints, limiting the widespread adoption of LoRA-MoE.

In summary, LoRA-MoE is better suited for high-performance-demanding scenarios, but its usability

and training cost issues limit its broader applicability. On the other hand, model merging is more user-friendly, making it more accessible to a wider audience. Both approaches are valuable, and there is also potential for them to be used together to further enhance performance.

D.3 Why not Model Swarm?

Model Swarm (Feng et al., 2024; Zhang et al., 2025b) refers to a family of methods that use evolutionary algorithms or other techniques to collaboratively combine existing model weights into stronger models. The advantage of this approach lies in its ability to achieve strong performance, sometimes even outperforming individually fine-tuned models. However, compared to the model merging approach studied in this paper, Model Swarm faces two major issues that limit its practical deployment:

Cost. Model Swarm requires a certain amount of data and substantial computation to determine the direction of collaborative evolution. At the very least, it involves repeated evaluations of each evolved model’s performance to guide further optimization. This process demands a significant amount of inference resources, making it impractical for average users due to high costs. In contrast, the model merging methods we focus on do not require extensive inference to construct a unified weight set, resulting in lower overhead and broader applicability.

Usability. Constructing stronger weights using Model Swarm also requires a certain level of deep learning expertise. As more and more fields adopt deep learning techniques, this requirement becomes a barrier for a large number of non-expert users, limiting the method’s real-world usage.

In summary, Model Swarm suffers from high computational cost and limited usability. It is better suited for performance-critical scenarios, whereas model merging is more appropriate for resource-constrained settings commonly faced by general users. Both approaches are valuable, and there is also potential for them to be combined for further performance gains.

D.4 Why not Merge A and B Separately?

Here, we discuss why our merging method is performed on the product matrix $W = BA$, rather than separately on the individual low-rank matrices A and B . Suppose we have two LoRA modules, and we select a certain layer for merging. Let their

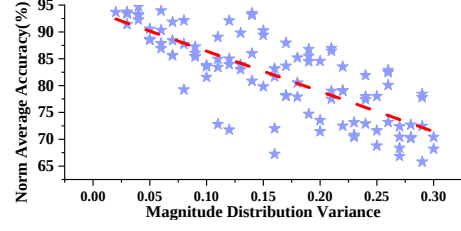


Figure 6: A greater magnitude disparity between models correlates with worse merging performance on T5-base.

parameters be denoted as A_1, B_1 and A_2, B_2 , respectively. Our current merging strategy is formulated as:

$$W_{\text{merge}} = \lambda(B_1 A_1 + B_2 A_2). \quad (31)$$

Alternatively, if merging is done separately on A and B matrices and then the full matrix is constructed, it can be expressed as:

$$\begin{aligned} W_{\text{merge}} &= \beta(B_1 + B_2)(A_1 + A_2) \\ &= \beta(B_1 A_1 + B_2 A_2) + \underbrace{\beta(B_1 A_2 + B_2 A_1)}_{\text{Cross term}}. \end{aligned} \quad (32)$$

It can be observed that this formulation introduces cross terms between the low-rank matrices of different LoRA modules. Intuitively, due to the distinct features learned during training, these cross terms act as noise and degrade the merging performance.

We validate this observation through experiments on ViT-B/32, ViT-B/16, and ViT-L/14 across eight vision tasks. The detailed results are shown in Tab. 12, Tab. 13, and Tab. 14. The results show that, for the same merging method across models, merging the A and B matrices separately typically leads to an average performance drop of over 10% compared to first computing their matrix product and then merging. This performance degradation validates the significant negative impact of the cross terms on merging effectiveness. Therefore, in practice, we perform merging in the $W = BA$ space, which leads to better performance.

E Limitations and Future Works

This paper introduces DO-Merging, a model merging method specifically designed for LoRA, aiming to address the performance degradation caused by the differences between LoRA and full fine-tuning during the adaptation process. Like many existing model merging approaches, its main limitation

Table 12: Merging results on eight vision tasks using ViT-B/32. *Separate* denotes merging the low-rank matrices individually, while *Concat* denotes merging the product of the low-rank matrices.

Method		SUN397	Cars	RESISC45	EuroSAT	SVHN	GTSRB	MNIST	DTD	Avg.
Task Arithmetic	Separate	61.37	51.08	63.63	70.33	68.73	51.77	85.73	46.60	62.40
	Concat	63.38	60.27	75.46	87.70	84.73	70.74	96.64	53.55	74.06
Ties-Merging	Separate	61.37	51.23	63.56	69.43	68.41	51.97	85.68	46.10	62.21
	Concat	63.81	53.95	73.56	86.00	88.27	71.15	97.81	50.32	73.10
Breadcrumbs	Separate	62.46	52.69	62.51	63.56	64.80	47.36	84.49	45.85	60.46
	Concat	64.02	57.87	74.12	88.73	84.24	74.28	97.29	53.23	74.22

Table 13: Merging results on eight vision tasks using ViT-B/16. *Separate* denotes merging the low-rank matrices individually, while *Concat* denotes merging the product of the low-rank matrices.

Method		SUN397	Cars	RESISC45	EuroSAT	SVHN	GTSRB	MNIST	DTD	Avg.
Task Arithmetic	Separate	64.18	57.32	69.03	63.63	76.73	58.84	94.54	46.38	66.33
	Concat	66.82	63.34	76.32	83.96	91.59	76.90	97.76	50.85	75.94
Ties-Merging	Separate	64.34	57.12	69.64	63.57	76.96	58.46	94.42	46.40	66.36
	Concat	66.41	63.41	77.07	84.06	91.97	77.50	98.02	51.06	76.19
Breadcrumbs	Separate	65.25	59.59	70.16	68.81	73.68	55.26	93.18	46.91	66.61
	Concat	65.38	64.12	77.49	83.75	91.32	77.20	97.87	50.92	76.00

lies in the need for users to manually find and select the desired model weights, which can be time-consuming and require significant effort in model retrieval.

Future works can focus on developing a fully user-friendly, end-to-end model merging pipeline. Such a system could automatically handle model discovery, selection, merging method adaptation, and result generation based on user requirements. This would greatly enhance the practical usability of model merging techniques.

F Additional Experimental Results

F.1 Comparative experiment.

Tab. 15 shows the test results of fine-tuning and merging ViT-B/16 on eight vision tasks, and Tab. 16 shows the corresponding results for ViT-L/14. Tab. 17 shows the corresponding results for T5-base. It can be observed that our DO-Merging method demonstrates consistently improved performance across both architectures. These results serve as supplementary validation for the superior performance of DO-Merging.

F.2 Visualization results.

Fig. 6 shows the relationship between magnitude disparity and performance on T5-base. Consistent with Theorem 3.1 and the analysis on ViT, the two exhibit a clear negative correlation.

F.3 Orthogonal Operations on B and A .

Here we look at orthogonal operations on B and A . Suppose $B_1, B_2 \in \mathbb{R}^{m \times k}$ and $A_1, A_2 \in \mathbb{R}^{k \times n}$, and define $W_i = B_i A_i$ for $i = 1, 2$. If we assume that B_1 and B_2 , as well as A_1 and A_2 , are orthogonal, then W_1 and W_2 would also be orthogonal. In practice, full orthogonality is hard to achieve, but experiments on visual tasks as shown in Tab. 18 show that orthogonalizing B and A separately or directly orthogonalizing W results in almost the same average performance, making the separate approach a simple and effective approximation.

F.4 Sensitivity of η .

We use $\eta = 1$ as the default in Eq. 4, which keeps the orthogonality loss L_o and regularization loss L_r at similar scales, reducing parameter changes while enforcing orthogonality. Tab. 19 shows ablation results across three ViT architectures, averaged over eight visual classification tasks. Performance remains stable across η values from 0.8 to 1.2, with $\eta = 1.0$ achieving the best or near-best results on all three models. Smaller η (e.g., 0.6) allows more aggressive orthogonalization but may perturb task vectors excessively, while larger η (e.g., 1.4) over-constrains the perturbation and limits the orthogonalization effect. If needed, task-specific adjustment can be easily done at the start to balance the two losses. The consistent optimality of $\eta = 1$

Table 14: Merging results on eight vision tasks using ViT-L/14. *Separate* denotes merging the low-rank matrices individually, while *Concat* denotes merging the product of the low-rank matrices.

Method		SUN397	Cars	RESISC45	EuroSAT	SVHN	GTSRB	MNIST	DTD	Avg.
Task Arithmetic	Separate	69.11	78.31	79.06	74.93	85.17	66.41	96.48	56.38	75.73
	Concat	70.19	80.09	82.62	79.41	90.18	77.93	98.12	59.89	79.80
Ties-Merging	Separate	69.54	78.24	79.12	74.24	85.56	66.34	96.65	56.14	75.73
	Concat	70.32	79.26	81.95	76.93	91.58	76.70	98.20	59.36	79.29
Breadcrumbs	Separate	67.62	71.15	77.14	66.37	88.86	67.89	96.71	54.73	73.81
	Concat	70.34	80.84	82.18	77.95	90.59	77.49	98.20	59.22	79.60

Table 15: Multi-task performance when merging ViT-B/16 on eight vision tasks.

Method	SUN397	Cars	RESISC45	EuroSAT	SVHN	GTSRB	MNIST	DTD	Avg.
Pre-trained	63.80	64.66	66.36	54.59	52.01	43.49	51.70	45.00	55.20
Finetune	75.10	99.14	98.64	99.58	95.42	78.62	75.40	97.69	89.94
Task Arithmetic (Ilharco et al., 2022)	66.82	63.34	76.32	83.96	91.59	76.90	97.76	50.85	75.94
Ties Merging (Yadav et al., 2024)	66.41	63.41	77.07	84.06	91.97	77.50	98.02	51.06	76.19
Breadcrumbs (Davari and Belilovsky, 2023)	65.38	64.12	77.49	83.75	91.32	77.20	97.87	50.92	76.00
AdaMerging (Yang et al., 2024b)	64.67	61.63	78.76	92.74	86.41	91.38	97.21	50.69	77.94
PCB-Merging (Du et al., 2024)	64.12	61.54	79.21	92.89	87.12	80.60	97.82	54.23	77.19
KNOTS (Stoica et al., 2025)	65.30	59.20	78.29	90.38	88.29	80.51	97.84	52.30	76.51
TSVM (Gargiulo et al., 2024a)	65.60	58.89	78.05	90.81	91.86	82.77	98.10	51.51	77.19
WUDI (Cheng et al., 2025)	65.30	59.30	78.40	90.45	90.49	80.64	98.32	53.34	77.02
CoPA-Merging (Zeng et al., 2025)	65.60	59.20	78.25	90.54	89.58	80.49	98.20	52.49	76.79
DO-Merging(ours)	65.70	66.61	80.78	93.07	89.27	82.90	97.81	57.79	79.24

Table 16: Multi-task performance when merging ViT-L/14 on eight vision tasks.

Method	SUN397	Cars	RESISC45	EuroSAT	SVHN	GTSRB	MNIST	DTD	Avg.
Pre-trained	66.80	77.90	71.37	62.15	58.42	50.54	76.35	55.58	64.89
Finetune	79.64	91.32	96.63	99.03	97.76	99.30	99.72	81.49	93.11
Task Arithmetic (Ilharco et al., 2022)	70.19	80.09	82.62	79.41	90.18	77.93	98.12	59.89	79.80
Ties Merging (Yadav et al., 2024)	70.32	79.26	81.95	76.93	91.58	76.70	98.20	59.36	79.29
Breadcrumbs (Davari and Belilovsky, 2023)	70.34	80.84	82.18	77.95	90.59	77.49	98.20	59.22	79.60
AdaMerging (Yang et al., 2024b)	72.42	83.34	83.56	83.53	93.57	84.53	98.20	63.56	82.83
PCB-Merging (Du et al., 2024)	70.13	81.40	81.56	81.03	92.35	81.53	98.26	60.12	80.79
KNOTS (Stoica et al., 2025)	70.49	80.30	82.21	80.56	92.65	80.92	98.26	62.27	80.95
TSVM (Gargiulo et al., 2024a)	70.77	79.34	83.35	81.63	92.51	79.94	98.36	61.01	80.86
WUDI (Cheng et al., 2025)	70.21	79.45	84.32	82.56	92.10	80.42	97.58	60.75	80.92
CoPA-Merging (Zeng et al., 2025)	70.54	80.68	83.49	83.62	92.48	80.28	97.84	62.56	81.43
DO-Merging(ours)	70.22	85.35	87.90	84.92	94.74	87.13	97.78	68.62	84.58

Table 17: Multi-task performance when merging T5-base on eight language tasks.

Method	COLA	MNLI	MRPC	QNLI	QQP	RTE	SST2	STSB	Avg.
Pre-trained	69.1	56.5	76.2	88.4	82.1	80.1	91.2	62.2	75.7
Finetune	69.1	82.7	85.5	90.9	84.0	84.5	92.9	87.4	84.6
Task Arithmetic (Ilharco et al., 2022)	68.8	55.2	78.7	89.8	83.7	79.1	91.5	72.4	77.4
Ties Merging (Yadav et al., 2024)	68.3	56.3	79.4	89.8	83.7	79.4	91.6	71.2	77.5
Breadcrumbs (Davari and Belilovsky, 2023)	68.3	57.3	78.5	89.8	83.6	79.5	91.8	72.4	77.6
PCB-Merging (Du et al., 2024)	68.3	57.4	79.3	89.8	83.6	79.2	91.5	72.6	77.7
KNOTS (Stoica et al., 2025)	69.4	63.6	78.5	89.8	83.6	80.2	91.5	70.6	78.4
TSVM (Gargiulo et al., 2024a)	69.1	67.8	79.9	89.7	83.6	80.9	91.6	69.4	79.0
CoPA-Merging (Zeng et al., 2025)	69.4	59.0	79.2	89.9	83.6	80.1	91.4	75.4	78.5
DO-Merging(ours)	69.3	75.8	79.7	90.5	84.0	80.1	93.1	74.4	80.9

Table 18: Comparison between performing orthogonalization on the A and B of LoRA individually and on the combined product matrix.

	ViT-B/32	ViT-B/16	ViT-L/14
Seperate	77.88	79.24	84.58
Combine	77.82	79.43	84.47

Table 19: Sensitivity analysis of the regularization hyperparameter η in Eq. 4.

η	0.6	0.8	1.0	1.2	1.4
ViT-B/32	75.86	76.59	77.88	77.53	77.12
ViT-B/16	78.54	78.76	79.24	79.10	78.84
ViT-L/14	83.87	84.43	84.58	84.62	84.46

Table 20: Pairwise loss gap analysis. For each model, we partition all $\binom{n}{2}$ task pairs by the sign of f_{ij} and report the count, mean magnitude ratio $\bar{r}$, and subtotal $\sum f$. The net aggregate $\sum_{i < j} f_{ij}$ is strongly positive in both cases.

Model	Pairs	$f_{ij} > 0$		$f_{ij} \leq 0$		Net $\sum f_{ij}$
		Count	$\bar{r}$	Count	$\bar{r}$	
ViT-B/32	28	14	3.67	14	1.41	+80.0
ViT-L/14	28	12	2.76	16	1.17	+149.9

across model scales suggests it is a reliable default without per-model tuning.

F.5 Empirical Verification of Theorem Conditions

We evaluate whether decoupling yields a net benefit by computing the pairwise loss gap f_{ij} from Theorem 3.1 for all $\binom{n}{2}$ task pairs:

$$f_{ij} = \frac{(a_i + a_j)^2}{8a_i a_j} \left[(a_i - a_j)^2 - 2a_i a_j (1 - \rho_{ij}) \right], \quad (33)$$

where $a_i = \|\alpha_i\|_2$ and $\rho_{ij} = \sum_k \alpha_{ik} \alpha_{jk} / (a_i a_j)$. A positive f_{ij} means decoupling reduces the expected loss for pair (i, j) . Following the $n > 2$ extension (Appendix B.2), the total loss decomposes into pairwise terms, so the aggregate $\sum_{i < j} f_{ij}$ determines whether decoupling is beneficial overall.

Two structural patterns emerge from Tab. 20:

(1) Large $r \Rightarrow f_{ij} > 0$ with large magnitude. Pairs with substantial magnitude disparity consistently yield positive f_{ij} , as the benefit term $(a_i - a_j)^2$ dominates. These are precisely the cases where decoupling is needed: without it, the high-magnitude task would dominate the merged direction.

(2) $r \approx 1 \Rightarrow f_{ij}$ can be negative, but $|f_{ij}|$ is inherently small. When $a_i \approx a_j$, the benefit term

$(a_i - a_j)^2 \rightarrow 0$ and the prefactor $\frac{(a_i + a_j)^2}{8a_i a_j} \rightarrow \frac{1}{2}$, so $|f_{ij}| \rightarrow 0$ regardless of ρ . Decoupling is neither needed nor harmful when magnitudes are already balanced.

This asymmetry, large positive f_{ij} when r is large versus small negative f_{ij} when $r \approx 1$, ensures that the aggregate loss gap is strongly positive. Positive contributions outweigh negative ones by $3.3 \times$ (ViT-B/32) and $2.8 \times$ (ViT-L/14), confirming that decoupling yields a substantial net improvement in merging quality.