EXPLOITING SYMMETRY IN LOW-RANK DECOMPOSITION FOR LORA FUSION

Anonymous authors
Paper under double-blind review

ABSTRACT

Low-Rank Adaptation (LoRA) is widely used to fine-tune Large Language Models across diverse downstream tasks, motivating model fusion as a way to integrate multiple task-specific LoRA adapters into a single model without access to the original training data. Existing LoRA fusion methods, ranging from naive averaging to interference-resolution approaches, overlook a key geometric property of the LoRA parameter space: LoRA adapters admit multiple reparameterizations that are functionally equivalent, a property we refer to as low-rank decomposition (LRD) symmetry. Ignoring this symmetry, however, causes direct fusion of independently trained adapters to suffer from parameter-space misalignment, leading to degraded representations and performance. To address this issue, we propose a **symmetry-aware parameter alignment** framework that maps LoRA adapters into a shared latent basis prior to fusion, first through a closed-form rotational alignment and then through a gradient-based refinement that relaxes orthogonality to the wider space of invertible reparameterizations. Our method is a plug-and-play enhancement compatible with existing LoRA fusion techniques. Extensive experiments on real-world benchmarks show broad improvements across diverse fusion methods. Our implementation is available on `https://anonymous.4open.science/r/GRA`.

1 INTRODUCTION

Parameter-efficient fine-tuning (PEFT) (Han et al., 2024) has emerged as a cornerstone paradigm for adapting Large Language Models (LLMs) to diverse downstream tasks. As a representative PEFT method, Low-Rank Adaptation (LoRA) (Hu et al., 2022) decomposes high-dimensional weight updates into the product of two low-rank matrices, $\Delta W = BA$. This structure drastically reduces the number of trainable parameters while enabling modular task adaptation, where the base model is fine-tuned on different tasks using different LoRA adapters. However, as the ecosystem of task-specific LoRA adapters grows, maintaining a large collection of adapters introduces increasing memory overhead and deployment complexity.

Model fusion, which combines multiple models into a single unified model without access to the original training data, has therefore attracted significant attention as a mechanism for integrating LoRA adapters. Existing LoRA fusion approaches range from simple arithmetic averaging (Wortsman et al., 2022) to more sophisticated preprocessing techniques such as TIES (Yadav et al., 2023) and DARE (Yu et al., 2024), which mitigate parameter interference through trimming and sign resolution (TIES) and random drop-and-rescale (DARE), respectively. Many of these approaches operate directly on the low-rank factors (A, B) of LoRA adapters and implicitly assume that independently trained adapters share a comparable latent coordinate system. In other words, dimensions at the same position in different adapters are treated as semantically aligned and therefore safe to combine in an element-wise manner.

However, this assumption is problematic due to the inherent non-identifiability of low-rank decompositions. For any LoRA adapter (A, B), there exist infinitely many equivalent factorizations related by symmetry transformations such as $(A, B) \sim (R^{-1}A, BR)$ that produce the same weight update $\Delta W = BA$. Although these factorizations are functionally identical, their latent rank spaces may be expressed in different coordinate systems. Consequently, independently trained adapters can exhibit substantial subspace misalignment even when their induced weight updates are comparable. Directly

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

merging the factors of such adapters mixes parameters defined in incompatible bases, leading to destructive interference during fusion.

To address this issue, we propose a **symmetry-aware parameter alignment framework** that acts as a plug-and-play module for LoRA fusion algorithms. Our key insight is that LoRA's low-rank parameterization introduces intrinsic symmetry, making the factor representation non-unique. Although equivalent factorizations yield identical model behavior individually, their parameters may not be directly comparable across adapters. Unlike prior works that focus on *how* to fuse adapters (e.g., pruning or resolving sign conflicts), our work focuses on *where* fusion should occur in parameter space. As illustrated in Figure 1, we transform LoRA adapters into a shared geometric basis *before* fusion, thereby resolving symmetry-induced misalignment. In practice, we adopt one adapter as a reference and align all other adapters to its coordinate system. We emphasize that this reference defines a shared *gauge* for factor-space fusion rather than a globally canonical basis for all adapters: our goal is to reduce the measurable factor-space mismatch caused by low-rank non-identifiability, not to eliminate all coordinate ambiguity.

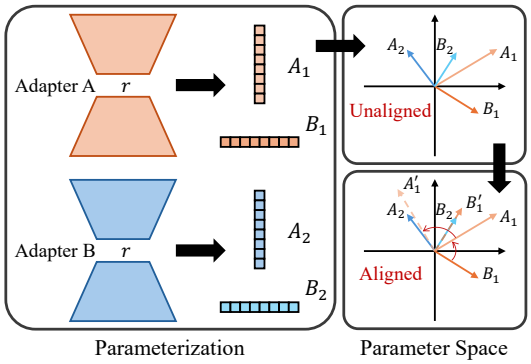

Figure 1: Illustration of the subspace misalignment problem in LoRA fusion. While two adapters may induce identical weight updates ($\Delta W_1 = \Delta W_2$), their latent rank spaces can differ by symmetry transformations. Direct averaging in mismatched coordinate systems causes destructive interference, whereas our pre-fusion alignment restores geometric correspondence.

Our framework consists of two complementary stages that progressively reduce the discrepancy between the target and the reference adapters under the symmetry of low-rank decomposition: (i) *Geometric Alignment (rotation-based)*. We first exploit the rotational symmetry of low-rank factorization to obtain an orthogonal alignment between adapters. This step admits a stable closed-form solution and establishes a shared geometric coordinate system for independently trained adapters. (ii) *Refined Alignment (optimization-based)*. While rotation corrects differences in latent bases, it cannot fully capture all transformations allowed by low-rank symmetry. Starting from the rotation solution, we further perform gradient-based refinement that expands the search space beyond orthogonal transformations, allowing subtle non-orthogonal adjustments that better align the functional behavior of adapters.

Furthermore, we introduce three structural variants of our method: **A-only**, **B-only**, and **Joint** alignment, which respectively align the input projection (A), the output projection (B), or both components. These variants are not a nested component ablation; rather, they are three distinct alignment objectives over the same LRD-preserving reparameterization, enabling flexible control over where alignment pressure is applied within the LoRA factors. Our contributions are summarized as follows:

- **Symmetry-Induced Misalignment Analysis**. We identify subspace misalignment arising from low-rank decomposition invariance as an overlooked source of error in LoRA fusion. Our analysis highlights that independently trained adapters may reside in incompatible coordinate systems even when their functional updates are comparable.
- **A Fusion-Agnostic Alignment Framework**. We propose a two-stage alignment framework combining closed-form geometric initialization with gradient-based refinement. The method is compatible with a wide range of existing LoRA fusion algorithms and can be applied as a lightweight preprocessing step.
- **Empirical Evaluation**. Experiments across multiple benchmarks demonstrate broad average improvements of our method for diverse fusion baselines while preserving the behavior of individual adapters.

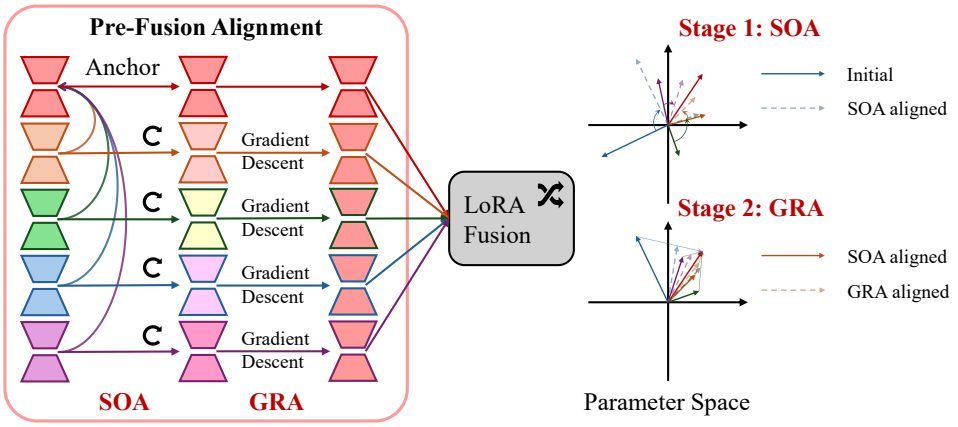

Figure 2: Overview of our symmetry-aware alignment framework. We first choose an anchor adapter from the independently trained LoRA adapters. For the first stage (SOA), we find the optimal rotation that makes each remaining adapter closest to the anchor by solving the Orthogonal Procrustes problem in Equation (6). In the second stage (GRA), we aim to minimize the distance to ensure the linear transformations align each remaining adapter to the anchor adapter.

2 PRELIMINARIES

Low Rank Adaptation for Language Models. Low-Rank Adaptation (LoRA) (Hu et al., 2022) hypothesizes that the change in weights during model adaptation has a low intrinsic rank. For a pre-trained weight matrix $W_0 \in \mathbb{R}^{d\times k}$, LoRA constrains its update ΔW by decomposing it into the product of two low-rank matrices $B \in \mathbb{R}^{d\times r}$ and $A \in \mathbb{R}^{r\times k}$, where rank $r \ll \min(d, k)$. The forward pass is modified as:

$$h = W_0 x + \Delta W x = W_0 x + BAx, \tag{1}$$

where x denotes the input activation to the weight matrix W_0. LoRA enables indirect training of each dense layer in a neural network by injecting and optimizing low-rank decomposition matrices ($\Delta W = BA$) that represent the layer's weight updates, while keeping the original weight matrix W_0 frozen (Hu et al., 2022).

LoRA Fusion for Multi-Task Learning. We consider a set of N task-specific LoRA adapters $\{\theta_0, \ldots, \theta_{N-1}\}$ trained from the same base model W_0. Each module $\theta_i = \{A_i, B_i\}$ encodes the learned knowledge from task i. The goal of model fusion is to combine these modules into a single unified module θ_f that performs well across tasks. A general fusion framework can be expressed as:

$$\theta_f = f(\{\theta_0, \theta_1, \ldots, \theta_{N-1}\}), \tag{2}$$

where f represents the fusion function. Existing methods like Fisher (Matena & Raffel, 2022) and TIES (Yadav et al., 2023) perform such aggregation element-wise on the parameters A, B. A consolidated summary of the symbols used throughout the paper is provided in Section B.

3 METHODOLOGY

In this section, we present a framework for aligning LoRA adapters prior to fusion. Our method targets factor-space LoRA fusion, where adapters are merged directly through their low-rank factors. The key idea is to exploit the intrinsic symmetry of low-rank decomposition to map independently trained adapters into a shared latent basis before fusion. We first introduce the theoretical foundation of low-rank decomposition (LRD) symmetry in LoRA, and then describe our two-stage alignment procedure: SVD-based Orthogonal Alignment (SOA) and Gradient-based Refined Alignment (GRA). An overview of the framework is shown in Figure 2, with the full procedure given in Algorithm 1.

3.1 LOW-RANK DECOMPOSITION SYMMETRY

LoRA's low-rank formulation introduces an inherent reparameterization freedom: different pairs of (B, A) can yield the same update $\Delta W = BA$. We characterize this symmetry under linear transformations of the latent rank-r space and use it as the foundation of our alignment procedure.

Proposition 1 (Low-Rank Decomposition Symmetry)**.** *Let $\theta = \{A, B\}$ be the weight update of a LoRA adapter, where $B \in \mathbb{R}^{d\times r}$ and $A \in \mathbb{R}^{r\times k}$. For any invertible matrix $R \in \mathbb{R}^{r\times r}$ (i.e., $rank(R) = r$), the reparameterization*

$$\tilde{B} = BR, \qquad \tilde{A} = R^{-1}A \tag{3}$$

preserves the LoRA update, namely $\tilde{B}\tilde{A} = BA$ (equivalently, $\Delta\tilde{W} = \Delta W$).

We leave the proof in Section G. This symmetry implies that the learned factors A and B are not unique: a LoRA adapter is identifiable only up to an equivalence class of invertible basis transformations in the latent rank-r space. Consequently, independently trained adapters may represent functionally valid solutions in different latent bases. For fusion methods that operate directly on the LoRA factors, such non-identifiability can cause subspace misalignment: coordinates at the same position across adapters need not correspond to the same latent direction. Directly combining these factors therefore mixes incompatible bases and can degrade fusion performance. Our alignment can thus be viewed as a *reference-gauge* step: it selects a more compatible factor representative within each adapter's LRD-equivalent class before factor-space fusion, without altering the induced update ΔW of any individual adapter.

3.2 SVD-BASED ORTHOGONAL ALIGNMENT (SOA)

We first seek a coarse but robust alignment by restricting the transformation R to the orthogonal group, i.e., $R^{-1} = R^\top$. This isolates the dominant basis mismatch while preserving norms and numerical stability, and also yields a closed-form optimum that serves as a principled initialization for later refinement. To address the misalignment issue, we designate one adapter (denoted by θ_0) as the anchor adapter, and align all other adapters θ_i ($0 < i < N$) toward it. We consider three alignment objectives: aligning only A, aligning only B, and aligning both jointly.

Anchor selection. Rather than choosing an arbitrary task adapter as the reference, we adopt a reproducible, data-free *medoid* rule: we select the adapter with the smallest mean Frobenius distance to the remaining adapters in the pool. This makes the reference a fixed gauge choice rather than a task preference. Aligned main-table results use this medoid anchor. We quantify anchor sensitivity in Section F.1.

Optimization Objective (A-only / B-only). For each non-anchor adapter $\theta_i = \{A_i, B_i\}$, we first study the simpler setting that aligns only one factor to its anchor counterpart. The other factor is updated according to Equation (3) to preserve functional equivalence. Denote the orthogonal group by $\mathcal{O}(r) = \{R \in \mathbb{R}^{r\times r} \mid R^\top R = I\}$. The objectives are

$$\begin{cases} \min\limits_{R\in\mathcal{O}(r)} & \|B_i R - B_0\|_F^2, \ \textit{(B-only)}, \\ \min\limits_{R\in\mathcal{O}(r)} & \|R^\top A_i - A_0\|_F^2, \ \textit{(A-only)}. \end{cases} \tag{4}$$

The following theorem gives the optimal solution. We leave the proof in Section G.

Theorem 1. *For either objective in Equation (4), define the correlation matrix as*

$$M = \begin{cases} B_i^\top B_0, & \textit{(B-only)}, \\ A_i A_0^\top, & \textit{(A-only)}. \end{cases} \tag{5}$$

Let $U\Sigma V^\top$ be the SVD of M. Then the optimal solution is $R_i^* = UV^\top \in \mathcal{O}(r)$.

Optimization Objective (Joint). We next consider a unified formulation that jointly aligns both LoRA factors. For each non-anchor adapter $\theta_i = \{A_i, B_i\}$, we seek a single orthogonal transformation that simultaneously matches it to the anchor adapter $\theta_0 = \{A_0, B_0\}$:

$$\min_{R\in\mathcal{O}(r)} \|B_i R - B_0\|_F^2 + \|R^\top A_i - A_0\|_F^2. \tag{6}$$

Theorem 2. *Define*

$$M_{\text{joint}} = B_i^\top B_0 + A_i A_0^\top. \tag{7}$$

Let $M_{\text{joint}} = U\Sigma V^\top$ be its singular value decomposition. Then the optimal orthogonal transformation for Equation (6) is

$$R_i^* = UV^\top \in \mathcal{O}(r). \tag{8}$$

We leave the proof in Section G. This analytical solution provides a fast and globally optimal solution within the orthogonal group, without iterative optimization. Importantly, these variants are not nested: the A-only and B-only objectives prioritize one factor, whereas the joint objective aligns both factors simultaneously. In all three cases, after solving the selected objective, both factors are transformed as $(B_i R, R^{-1} A_i)$ so that the induced update $B_i A_i$ is preserved before fusion; the variants differ only in where the alignment pressure is placed, not in whether functional equivalence is maintained.

3.3 Gradient-based Refined Alignment (GRA)

While SOA provides a robust geometric initialization, the transformation between independently trained adapters may not be strictly orthogonal. To allow more flexible alignment, we introduce a gradient-based refinement stage that relaxes the orthogonality constraint. Importantly, GRA uses the same alignment objectives as SOA, but optimizes them in the larger space of general invertible transformations. We initialize R_i with the SOA solution and then refine it by gradient descent.

Refinement Loss (A-only / B-only). For each non-anchor adapter $\theta_i = \{A_i, B_i\}$, we first refine the alignment by updating R_i with respect to a single factor:

$$\begin{cases} \mathcal{L}_{\text{GRA}}^{\text{B}}(R_i) = \|B_i R_i - B_0\|_F^2, \\ \mathcal{L}_{\text{GRA}}^{\text{A}}(R_i) = \|R_i^{-1} A_i - A_0\|_F^2. \end{cases} \tag{9}$$

After optimization, both A and B are updated according to Equation (3) to preserve the induced update ΔW.

Refinement Loss (Joint). We further consider a joint refinement that simultaneously updates both factors under a shared transformation:

$$\mathcal{L}_{\text{GRA}}^{\text{J}}(R_i) = \|B_i R_i - B_0\|_F^2 + \|R_i^{-1} A_i - A_0\|_F^2. \tag{10}$$

Optimization. We perform T steps of optimization using Adam to update R_i. Since unconstrained refinement may introduce excessive distortion, we regularize R_i with $\mathcal{L}_{\mathcal{O}} = \|R_i R_i^\top - I\|_F^2$ to keep it near-orthogonal and maintain a well-conditioned transformation. Let η denote the learning rate:

$$R_i^{(t+1)} \leftarrow R_i^{(t)} - \eta \nabla_{R_i} (\mathcal{L}_{\text{GRA}} + \lambda \mathcal{L}_{\mathcal{O}}), \tag{11}$$

where λ is a loss weight factor (default 1e-3). After refinement, the final aligned parameters $\tilde{\theta}_i = \{R_i^{-1} A_i, B_i R_i\}$ are used for fusion. In this way, the two-stage design combines the stability of closed-form orthogonal alignment with the flexibility of non-orthogonal refinement. Additional variants, including direct least-squares alignment and removing the orthogonality regularizer, are discussed in Section E, where we also validate the SOA initialization against direct refinement from random initialization (Section E.3).

4 Experiments

In this section, we aim to answer the following three research questions: **RQ1:** How broadly do the proposed alignment algorithms improve the performance of diverse LoRA fusion baselines? **RQ2:** Can our method effectively reduce symmetry-induced misalignment in LoRA factor space, and does this reduction correlate with downstream performance gains? **RQ3:** Is our method robust across different evaluation settings, including unseen tasks, layer choices, and computational overhead?

4.1 Empirical Settings

Models and Benchmarks. To evaluate the versatility and effectiveness of our approach, we employ two distinct transformer architectures covering both decoder-only and encoder-decoder paradigms.

Specifically, we adopt **Llama-3.1-8B** (Grattafiori et al., 2024) as our primary backbone for natural language understanding, and evaluate it on the GLUE benchmark (Wang et al., 2018) as well as a suite of natural language inference (NLI) benchmarks following (Stoica et al., 2025), which comprise multiple NLI tasks. We include details of the benchmarks in Section D.2. In addition, following prior work (Huang et al., 2024), we employ **Flan-T5-Large** (Chung et al., 2022) to assess the composability of adapters on complex reasoning tasks from the Big-Bench Hard (BBH) benchmark (Suzgun et al., 2022). For GLUE and NLI, we train one LoRA adapter per subtask and fuse the resulting pool, so these are *in-family* adapter-fusion settings; BBH instead follows the LoRAHub cross-task protocol and serves as an *out-of-domain* reasoning-transfer stress test. Table 1 summarizes the adapter pools, training source, and evaluation for each benchmark family, and we verify the method on a newer backbone (`Qwen3-0.6B`) in Section F.2.

Benchmark	Adapter pool	Training source	Evaluation	Domain
GLUE	one per GLUE subtask	GLUE train split	GLUE validation	in-family
NLI	one per NLI subtask	NLI train loaders	NLI test loaders	in-family
BBH	LoRAHub pool	LoRAHub few-shot	BBH zero-shot/test	out-of-domain

Table 1: Experimental setup. GLUE and NLI use one trained adapter per subtask (six adapters each) on `Llama-3.1-8B`; BBH follows the LoRAHub protocol on `Flan-T5-Large` as an out-of-domain reasoning-transfer test.

Baselines. We compare against a diverse set of LoRA fusion baselines. ***Data-agnostic methods*** include **Simple** (Wortsman et al., 2022) and **Magnitude-Prune**. ***Data-driven methods*** include **Fisher** (Matena & Raffel, 2022), **RegMean** (Jin et al., 2023), **TIES** (Yadav et al., 2023), **DARE** (Yu et al., 2024), **LoRAHub** (Huang et al., 2024), and **Adaptive LoRA Merge (ALM)** (Miyano & Arase, 2025). In addition, we include **Git-Rebasin** (Ainsworth et al., 2023) as a permutation-based alignment baseline. We provide detailed descriptions of all baselines in Section D.3. When a baseline does not specify merge weights, we assign equal weights to all adapters for fairness. LoRAHub is a gradient-free method that estimates its composition weights from only a few target-task examples rather than by training on the full target dataset. Because its search still repeatedly evaluates merged models, we primarily report it on benchmarks with relatively small evaluation sets, and provide a protocol-matched leave-target-out LoRAHub comparison on GLUE/NLI in Section F.3.

	Baseline Acc.	+Git-Rebasin	+SOA(A)	+SOA(B)	+SOA(J)	+GRA(A)	+GRA(B)	+GRA(J)	Avg Impr.
Data-Agnostic Methods									
Simple	53.57	53.61	53.64	53.88	53.99	54.22	**54.59**	54.41	+0.55
Magnitude-Prune	57.95	60.77	65.82	64.42	64.21	**66.02**	64.71	65.86	+7.22
Data-Driven Methods									
TIES	53.07	53.18	53.23	53.51	53.73	53.99	54.16	**54.63**	+0.80
Fisher	58.98	60.12	59.63	60.79	59.37	**64.96**	64.44	63.67	+3.16
RegMean	54.69	54.82	54.95	55.16	55.38	56.97	**57.22**	57.19	+1.46
DARE-Linear	53.18	53.71	55.05	54.15	54.80	57.54	57.38	**58.52**	+3.06
DARE-TIES	52.96	52.99	53.41	**53.70**	53.62	52.98	53.01	53.33	+0.38
ALM	53.68	53.81	55.30	54.48	54.72	**56.41**	55.34	55.67	+1.64
Avg Impr.	/	+0.62	+1.62	+1.50	+1.47	+3.13	+2.85	**+3.15**	+2.29

Table 2: Performance across the GLUE benchmark. Scores are unweighted means over the six tasks, and aligned scores use the medoid anchor. SOA represents SVD-based Orthogonal Alignment; GRA represents Gradient-based Refined Alignment. (A)/(B)/(J) denote the A-only, B-only, and Joint objectives. Git-Rebasin is a permutation-based alignment baseline. Row-wise Avg Impr. averages the six SOA/GRA gains over Baseline Acc.; the bottom row averages each column over all eight listed fusion methods.

4.2 In-Domain LoRA Fusion Results (RQ1)

To answer RQ1, we evaluate our alignment method on the GLUE and NLI benchmarks, each of which comprises multiple subtasks. For each subtask, we independently fine-tune a LoRA adapter and subsequently fuse all trained adapters into a single one. We then evaluate the fused adapter on each subtask within the corresponding benchmark.

	Baseline Acc.	+Git-Rebasin	+SOA(A)	+SOA(B)	+SOA(J)	+GRA(A)	+GRA(B)	+GRA(J)	Avg Impr.
Data-Agnostic Methods									
Simple	47.09	47.21	48.47	48.78	48.61	48.71	48.81	**49.01**	+1.64
Magnitude-Prune	51.02	51.05	56.37	54.85	55.50	**56.41**	55.31	56.06	+4.73
Data-Driven Methods									
TIES	47.16	47.24	47.39	47.32	47.51	47.68	**47.69**	47.64	+0.38
Fisher	50.76	51.57	54.99	54.00	54.75	**57.00**	55.31	56.82	+4.72
RegMean	47.17	47.30	49.54	50.00	49.41	49.75	**50.29**	49.85	+2.64
DARE-Linear	47.56	49.64	51.44	51.69	51.48	51.24	**52.32**	51.71	+4.09
DARE-TIES	47.13	47.23	**47.75**	47.12	47.04	47.32	47.27	47.38	+0.18
ALM	47.11	47.33	48.37	48.49	48.41	48.76	48.80	**49.01**	+1.53
Avg Impr.	/	+0.45	+2.42	+2.16	+2.21	+2.73	+2.60	**+2.81**	+2.49

Table 3: Performance across the NLI benchmark. Scores are unweighted means over the six tasks, and aligned scores use the medoid anchor. Row-wise Avg Impr. averages the six SOA/GRA gains over Baseline Acc.; the bottom row averages each column over all eight listed fusion methods.

Table 2 and Table 3 report the average accuracy of the fused adapter across all subtasks for both benchmarks. In addition to standard fusion baselines, we also include **Adaptive LoRA Merge (ALM)** as an additional fusion baseline and **Git-Rebasin** as a permutation-based alignment baseline.

From the results, we draw the following observations: ***(i)*** Overall, both SOA and GRA improve average performance across a broad range of LoRA fusion baselines on GLUE and NLI, covering both data-agnostic and data-driven settings, with a small number of method-specific exceptions. This indicates that symmetry-induced misalignment affects multiple factor-space LoRA fusion families rather than a single baseline. ***(ii)*** Among all variants, **GRA(J)** achieves the best average performance overall on both benchmarks, showing that refining both A and B in a shared transformation space is the most effective configuration on average. ***(iii)*** The gains are not uniform across baselines. Larger improvements are observed for value-based baselines such as **Magnitude-Prune** and **Fisher**, while sign-based methods such as **TIES** typically benefit less. This suggests that alignment is especially important when fusion relies directly on parameter magnitudes or weighted averaging, where mismatched latent bases can more easily cause destructive interference. ***(iv)*** In aggregate, **Git-Rebasin** provides smaller gains than SOA and GRA across the two benchmarks, indicating that permutation-based alignment alone does not capture the full improvement available in LoRA factor space. ***(v)*** In general, GRA outperforms SOA on average, validating the benefit of combining closed-form orthogonal initialization with subsequent optimization-based refinement. In addition, Table 7 reports individual NLI/Simple anchor runs, and the full six-anchor mean $\pm$ standard-deviation sweep in Section F.1 shows that all reported all-anchor means remain above their unaligned baselines, although anchor sensitivity is non-negligible in some settings.

On the large Magnitude-Prune gain. Across the six SOA/GRA variants, **Magnitude-Prune** improves by an average of $+7.22$ percentage points on GLUE. This summary differs from the GRA(J) setting analyzed in Section F.6, whose aggregate gain is $+7.91$ points (65.86 versus 57.95). Magnitude-Prune makes discrete keep-or-drop decisions based on the absolute values of entries in the LoRA factors, so it is especially sensitive to how the *same* update is written: two adapters can share an identical ΔW while using differently scaled or rotated factor matrices, and a small change in representation can flip which near-threshold entries are retained. The task-level GRA(J) deltas are highly heterogeneous, with large gains on some tasks and decreases on others; this pattern is compatible with, but does not by itself prove, the proposed gauge-sensitivity explanation.

4.3 Misalignment Reduction Analysis (RQ2)

To answer RQ2, we directly measure the degree of factor-space misalignment before and after alignment. Specifically, for each benchmark, we compute the average Frobenius distance between each non-anchor adapter and the anchor adapter, and report the reduction achieved by different alignment objectives. Figure 3 visualizes the absolute distances before and after alignment on GLUE and NLI, and annotates the corresponding reduction percentages. We observe that all alignment variants reduce the discrepancy between adapters in factor space, confirming that independently trained LoRA adapters indeed exhibit measurable misalignment. Moreover, the **Joint** variant achieves

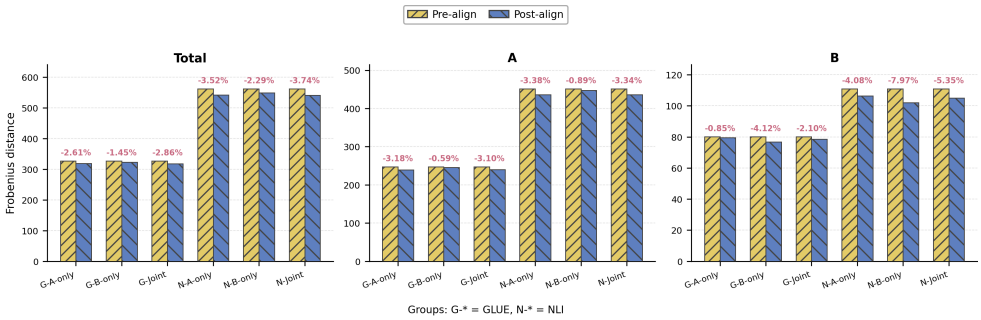

Figure 3: Compact one-row view of average Frobenius distances before and after alignment. Subplots correspond to *Total*, A, and B. Within each subplot, groups 'G-*' and 'N-*' denote GLUE and NLI objectives, respectively; pink labels indicate percentage reduction.

the largest overall distance reduction on both benchmarks, which is consistent with its strong downstream performance in RQ1.

These results provide direct evidence for our core motivation: although different LoRA factorizations may induce comparable weight updates, their low-rank factors can still reside in different latent bases. Our alignment method effectively reduces this discrepancy before fusion. We further note that the magnitude of distance reduction is positively correlated with the downstream performance gains in most cases, supporting the view that mitigating misalignment is the key for observed improvements.

4.4 Robustness and Efficiency Analysis (RQ3)

Out-of-Domain Results. To evaluate whether the proposed alignment remains useful when none of the source adapters is trained on the evaluation task, we test on the BBH benchmark. The results are reported in Table 4.

	Baseline	+GRA(A)	+GRA(B)	+GRA(J)
Data-Agnostic Methods				
Simple	20.65	**23.74**	22.82	20.29
MP	21.61	21.37	**22.68**	21.38
Data-Driven Methods				
TIES	23.24	**23.27**	23.24	23.24
Fisher	23.89	23.89	**23.99**	23.93
LoRAHub	24.36	**24.73**	24.09	24.69
RegMean	23.42	**24.62**	24.09	**24.62**
DARE-Linear	24.95	**25.16**	24.37	**25.16**
DARE-TIES	23.51	**23.74**	23.51	23.70
Avg Impr.	/	+0.61	+0.40	+0.17

Table 4: Performance across the BBH benchmark. MP represents Magnitude-Prune.

Overall, our alignment method still improves most fusion baselines in this out-of-domain setting, although the gains are smaller than those on GLUE and NLI. This is expected because BBH is a stress test rather than our primary target setting: it evaluates out-of-domain compositional reasoning, so the bottleneck is often source-adapter capability and domain transfer rather than factor-space mismatch alone. GRA can make the existing factor-space arithmetic more coherent, but it cannot create reasoning skills that are absent from the source adapters. We also note that the recommended variant *flips* out-of-domain: **GRA(J)** is strongest in-domain (GLUE/NLI) whereas **GRA(A)** is strongest on BBH. This follows from the functional asymmetry of the two factors in $\Delta W = BA$: A defines the input-side low-rank feature projection, whereas B maps features to task-specific output directions. For related in-domain adapters, task-output directions remain correlated, so aligning both factors (Joint) exploits shared feature *and* output structure; for out-of-domain transfer, aligning the shared feature side A is more useful, while forcing the more task-specific B into a shared gauge is overly restrictive. Nevertheless, the consistent improvements suggest that factor-space misalignment remains relevant beyond in-domain fusion.

Partial Layer Alignment. To study whether alignment is equally important across model depth, we conduct three partial-alignment experiments under the magnitude-prune baseline: head-layer alignment, tail-layer alignment, and disjoint layer-slice alignment. Figure 4 reports the results. We find that aligning deeper layers generally leads to larger gains than aligning shallow layers, suggesting that high-level task-specific representations are more sensitive to factor-space misalignment. At the same time, the layer-slice analysis shows that the contribution of alignment is highly non-uniform across layers.

Time Overhead. Finally, we report the average runtime of different stages in the fusion pipeline

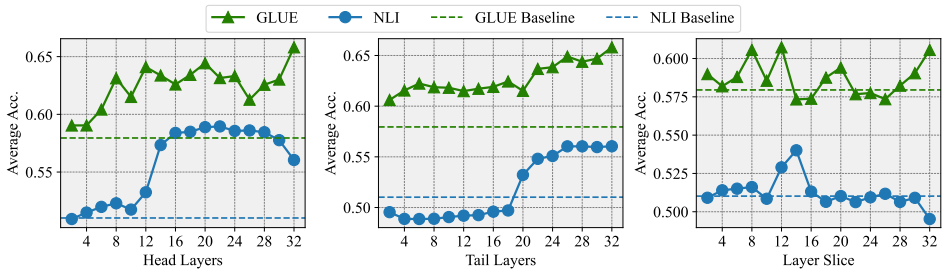

Figure 4: Partial layer alignment analysis on the GLUE and NLI benchmarks. Alignment is applied to specific subsets of Transformer layers while keeping the fusion pipeline unchanged, including head layers (left), tail layers (middle), and disjoint two-layer slices (right).

in Table 5, including alignment, fusion, and downstream evaluation. The results show that both SOA and GRA introduce only minor additional cost compared with the overall evaluation time. Alignment requires only low-rank singular value decompositions and a short refinement over rank-r transforms, so its cost scales with the LoRA rank rather than the full model dimension. Because alignment is performed once before deployment, the resulting factors can be cached and reused across downstream fusion queries, amortizing the added work while leaving inference latency unchanged. As Table 5 shows, both alignment stages remain inexpensive to downstream evaluation.

	GLUE	NLI	BBH
SOA	0.31	0.38	2.79
GRA	0.53	0.85	5.72
Simple	0.02	0.01	0.05
Magnitude-Prune	2.24	2.35	27.38
DARE-Linear	1.80	2.36	16.86
TIES	6.41	6.46	22.15
DARE-Ties	2.48	2.02	19.95
RegMean	0.25	0.43	1.22
Fisher	0.18	0.33	0.89
Evaluation	1472.05	1800.49	868.84

Table 5: Avg runtime in seconds.

5 Related Work

LoRA Fusion. Prior work reduces interference between independently trained adapters along three lines: reweighting and sparsification (Yadav et al., 2023; Miyano & Arase, 2025; Xiao et al., 2024), subspace factorization (Zheng et al., 2025; Zhang & Zhou, 2025; Stoica et al., 2025), and modular composition of rank-1 units (Huang et al., 2024; Su et al., 2025; Zhao et al., 2025b). These decide *how* to combine factors; we instead fix *where* the combination happens, so our alignment is a preprocessing step compatible with any factor-space fusion rule. Because the reparameterization preserves each source update ΔW_i exactly, the alignment can precede reweighting, sparsification, subspace factorization, or rank-wise selection without changing the underlying fusion rule. It does not target training-time orthogonality (Zhang et al., 2025b) or fusion that sees only the materialized update ΔW (Stoica et al., 2025); see Section A and Section C.

Parameter Space Symmetry. Permutation invariance (Entezari et al., 2022; Ainsworth et al., 2023) motivates weight-matching procedures that place models in a common gauge before averaging (Singh & Jaggi, 2020; Imfeld et al., 2024), which recent work extends from discrete matchings to continuous rotations (Zhang et al., 2025a; Zhao et al., 2025a). Unlike these whole-model alignments, LoRA admits an internal change-of-basis symmetry: for any invertible R, $BA = (BR)(R^{-1}A)$. We exploit this exact function-preserving freedom to align only the rank-r latent bases of already-trained adapters before factor-space fusion; Section C gives the extended discussion.

6 Conclusions

In this work, we identify latent subspace misalignment arising from low-rank decomposition symmetry as a fundamental challenge in LoRA fusion. To address this issue, we introduce a symmetry-aware pre-fusion alignment framework that aligns independently trained LoRA adapters into a shared geometric basis before fusion. The proposed approach is lightweight, fusion-agnostic, and preserves adapter functionality. Empirical results across diverse benchmarks demonstrate broad average improvements from enforcing geometric correspondence before fusion. More broadly, treating low-rank non-identifiability as an explicit gauge-selection problem offers a general lens on when and why factor-space fusion succeeds, and may guide future alignment-aware merging methods. We discuss the scope and limitations of our framework in Section A.

Ethics Statement

This paper studies symmetry-aware parameter alignment for improving the fusion of independently trained LoRA adapters. The proposed framework operates purely at the parameter level and does not introduce new training data, modify model outputs, or alter the inference behavior of individual adapters. While we acknowledge the importance of responsible use of large language models, we do not foresee that the techniques presented in this work pose significant ethical concerns or negative societal impacts.

AI Use Statement

In this work, we did not use generative AI tools for any research task that requires disclosure. Specifically, we have not used generative AI tools to develop theoretical models or conceptual frameworks, to formulate mathematical claims, to provide critical ingredients for proving mathematical claims or to assist in the writing of proofs, to propose or refine hypotheses, to design or provide feedback on our research methodology or experiments, to interpret experimental results, or to implement the proposed SOA/GRA methods; generating synthetic data sets, cleaning and reformatting data sets, assisting with translation, and supporting qualitative and thematic data analysis are not applicable to this work. Our use of generative AI tools (ChatGPT and Claude) was confined to preparing the manuscript: identifying and summarizing related literature, creating and modifying figures, drafting and editing parts of the manuscript for readability, and suggesting the structure of the paper. We have reviewed all AI-assisted work: all literature identified with AI assistance was read and verified by the authors; every AI-edited passage was checked against the intended technical content; and all experimental numbers, tables, and figures reported in this paper were produced by our own implementation and verified by the authors. The full proofs of our theoretical results, given in Section G, were derived and written by the authors. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

Reproducibility Statement

We provide the implementation of our alignment framework at the anonymized repository linked in the abstract. Implementation details, including the backbone models, LoRA hyperparameters, and alignment/refinement settings, are given in Section D. The benchmark subtasks and adapter pools are described in Section D.2 and summarized in Table 1, and all baselines are detailed in Section D.3. Full proofs of Proposition 1, Theorem 1, and Theorem 2 are provided in Section G.

References

Samuel K. Ainsworth, Jonathan Hayase, and Siddharth Srinivasa. Git re-basin: Merging models modulo permutation symmetries. In *Proceedings of the International Conference on Learning Representations*, 2023.

Vijay Badrinarayanan, Bamdev Mishra, and Roberto Cipolla. Symmetry-invariant optimization in deep networks. *arXiv preprint arXiv:1511.01754*, 2015.

Johanni Brea, Berfin Simsek, Bernd Illing, and Wulfram Gerstner. Weight-space symmetry in deep networks gives rise to permutation saddles, connected by equal-loss valleys across the loss landscape. *arXiv preprint arXiv:1907.02911*, 2019.

Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Yunxuan Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, Albert Webson, Shixiang Shane Gu, Zhuyun Dai, Mirac Suzgun, Xinyun Chen, Aakanksha Chowdhery, Alex Castro-Ros, Marie Pellat, Kevin Robinson, Dasha Valter, Sharan Narang, Gaurav Mishra, Adams Yu, Vincent Zhao, Yanping Huang, Andrew Dai, Hongkun Yu, Slav Petrov, Ed H. Chi, Jeff Dean, Jacob Devlin, Adam Roberts, Denny Zhou, Quoc V. Le, and Jason Wei. Scaling instruction-finetuned language models, 2022. URL https://arxiv.org/abs/2210.11416.

Simon S Du, Wei Hu, and Jason D Lee. Algorithmic regularization in learning deep homogeneous models: Layers are automatically balanced. *Advances in neural information processing systems*, 31, 2018.

Rahim Entezari, Hanie Sedghi, Olga Saukh, and Behnam Neyshabur. The role of permutation invariance in linear mode connectivity of neural networks. In *International Conference on Learning Representations*, 2022.

John C Gower and Garmt B Dijksterhuis. *Procrustes problems*, volume 30. OUP Oxford, 2004.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, Arun Rao, Aston Zhang, Aurelien Rodriguez, Austen Gregerson, Ava Spataru, Baptiste Roziere, Bethany Biron, Binh Tang, Bobbie Chern, Charlotte Caucheteux, Chaya Nayak, Chloe Bi, Chris Marra, Chris McConnell, Christian Keller, Christophe Touret, Chunyang Wu, Corinne Wong, Cristian Canton Ferrer, Cyrus Nikolaidis, Damien Allonsius, Daniel Song, Danielle Pintz, Danny Livshits, Danny Wyatt, David Esiobu, Dhruv Choudhary, Dhruv Mahajan, Diego Garcia-Olano, Diego Perino, Dieuwke Hupkes, Egor Lakomkin, Ehab AlBadawy, Elina Lobanova, Emily Dinan, Eric Michael Smith, Filip Radenovic, Francisco Guzmán, Frank Zhang, Gabriel Synnaeve, Gabrielle Lee, Georgia Lewis Anderson, Govind Thattai, Graeme Nail, Gregoire Mialon, Guan Pang, Guillem Cucurell, Hailey Nguyen, Hannah Korevaar, Hu Xu, Hugo Touvron, Iliyan Zarov, Imanol Arrieta Ibarra, Isabel Kloumann, Ishan Misra, Ivan Evtimov, Jack Zhang, Jade Copet, Jaewon Lee, Jan Geffert, Jana Vranes, Jason Park, Jay Mahadeokar, Jeet Shah, Jelmer van der Linde, Jennifer Billock, Jenny Hong, Jenya Lee, Jeremy Fu, Jianfeng Chi, Jianyu Huang, Jiawen Liu, Jie Wang, Jiecao Yu, Joanna Bitton, Joe Spisak, Jongsoo Park, Joseph Rocca, Joshua Johnstun, Joshua Saxe, Junteng Jia, Kalyan Vasuden Alwala, Karthik Prasad, Kartikeya Upasani, Kate Plawiak, Ke Li, Kenneth Heafield, Kevin Stone, Khalid El-Arini, Krithika Iyer, Kshitiz Malik, Kuenley Chiu, Kunal Bhalla, Kushal Lakhotia, Lauren Rantala-Yeary, Laurens van der Maaten, Lawrence Chen, Liang Tan, Liz Jenkins, Louis Martin, Lovish Madaan, Lubo Malo, Lukas Blecher, Lukas Landzaat, Luke de Oliveira, Madeline Muzzi, Mahesh Pasupuleti, Mannat Singh, Manohar Paluri, Marcin Kardas, Maria Tsimpoukelli, Mathew Oldham, Mathieu Rita, Maya Pavlova, Melanie Kambadur, Mike Lewis, Min Si, Mitesh Kumar Singh, Mona Hassan, Naman Goyal, Narjes Torabi, Nikolay Bashlykov, Nikolay Bogoychev, Niladri Chatterji, Ning Zhang, Olivier Duchenne, Onur Çelebi, Patrick Alrassy, Pengchuan Zhang, Pengwei Li, Petar Vasic, Peter Weng, Prajjwal Bhargava, Pratik Dubal, Praveen Krishnan, Punit Singh Koura, Puxin Xu, Qing He, Qingxiao Dong, Ragavan Srinivasan, Raj Ganapathy, Ramon Calderer, Ricardo Silveira Cabral, Robert Stojnic, Roberta Raileanu, Rohan Maheswari, Rohit Girdhar, Rohit Patel, Romain Sauvestre, Ronnie Polidoro, Roshan Sumbaly, Ross Taylor, Ruan Silva, Rui Hou, Rui Wang, Saghar Hosseini, Sahana Chennabasappa, Sanjay Singh, Sean Bell, Seohyun Sonia Kim, Sergey Edunov, Shaoliang Nie, Sharan Narang, Sharath Raparthy, Sheng Shen, Shengye Wan, Shruti Bhosale, Shun Zhang, Simon Vandenhende, Soumya Batra, Spencer Whitman, Sten Sootla, Stephane Collot, Suchin Gururangan, Sydney Borodinsky, Tamar Herman, Tara Fowler, Tarek Sheasha, Thomas Georgiou, Thomas Scialom, Tobias Speckbacher, Todor Mihaylov, Tong Xiao, Ujjwal Karn, Vedanuj Goswami, Vibhor Gupta, Vignesh Ramanathan, Viktor Kerkez, Vincent Gonguet, Virginie Do, Vish Vogeti, Vítor Albiero, Vladan Petrovic, Weiwei Chu, Wenhan Xiong, Wenyin Fu, Whitney Meers, Xavier Martinet, Xiaodong Wang, Xiaofang Wang, Xiaoqing Ellen Tan, Xide Xia, Xinfeng Xie, Xuchao Jia, Xuewei Wang, Yaelle Goldschlag, Yashesh Gaur, Yasmine Babaei, Yi Wen, Yiwen Song, Yuchen Zhang, Yue Li, Yuning Mao, Zacharie Delpierre Coudert, Zheng Yan, Zhengxing Chen, Zoe Papakipos, Aaditya Singh, Aayushi Srivastava, Abha Jain, Adam Kelsey, Adam Shajnfeld, Adithya Gangidi, Adolfo Victoria, Ahuva Goldstand, Ajay Menon, Ajay Sharma, Alex Boesenberg, Alexei Baevski, Allie Feinstein, Amanda Kallet, Amit Sangani, Amos Teo, Anam Yunus, Andrei Lupu, Andres Alvarado, Andrew Caples, Andrew Gu, Andrew Ho, Andrew Poulton, Andrew Ryan, Ankit Ramchandani, Annie Dong, Annie Franco, Anuj Goyal, Aparajita Saraf, Arkabandhu Chowdhury, Ashley Gabriel, Ashwin Bharambe, Assaf Eisenman, Azadeh Yazdan, Beau James, Ben Maurer, Benjamin Leonhardi, Bernie Huang, Beth Loyd, Beto De Paola, Bhargavi Paranjape, Bing Liu, Bo Wu, Boyu Ni, Braden Hancock, Bram Wasti, Brandon Spence, Brani Stojkovic, Brian Gamido, Britt Montalvo, Carl Parker, Carly Burton, Catalina Mejia, Ce Liu, Changhan Wang, Changkyu Kim, Chao Zhou, Chester Hu, Ching-Hsiang Chu, Chris Cai, Chris Tindal, Christoph Feichtenhofer, Cynthia Gao, Damon Civin, Dana Beaty, Daniel Kreymer, Daniel Li, David Adkins, David Xu,

Davide Testuggine, Delia David, Devi Parikh, Diana Liskovich, Didem Foss, Dingkang Wang, Duc Le, Dustin Holland, Edward Dowling, Eissa Jamil, Elaine Montgomery, Eleonora Presani, Emily Hahn, Emily Wood, Eric-Tuan Le, Erik Brinkman, Esteban Arcaute, Evan Dunbar, Evan Smothers, Fei Sun, Felix Kreuk, Feng Tian, Filippos Kokkinos, Firat Ozgenel, Francesco Caggioni, Frank Kanayet, Frank Seide, Gabriela Medina Florez, Gabriella Schwarz, Gada Badeer, Georgia Swee, Gil Halpern, Grant Herman, Grigory Sizov, Guangyi, Zhang, Guna Lakshminarayanan, Hakan Inan, Hamid Shojanazeri, Han Zou, Hannah Wang, Hanwen Zha, Haroun Habeeb, Harrison Rudolph, Helen Suk, Henry Aspegren, Hunter Goldman, Hongyuan Zhan, Ibrahim Damlaj, Igor Molybog, Igor Tufanov, Ilias Leontiadis, Irina-Elena Veliche, Itai Gat, Jake Weissman, James Geboski, James Kohli, Janice Lam, Japhet Asher, Jean-Baptiste Gaya, Jeff Marcus, Jeff Tang, Jennifer Chan, Jenny Zhen, Jeremy Reizenstein, Jeremy Teboul, Jessica Zhong, Jian Jin, Jingyi Yang, Joe Cummings, Jon Carvill, Jon Shepard, Jonathan McPhie, Jonathan Torres, Josh Ginsburg, Junjie Wang, Kai Wu, Kam Hou U, Karan Saxena, Kartikay Khandelwal, Katayoun Zand, Kathy Matosich, Kaushik Veeraraghavan, Kelly Michelena, Keqian Li, Kiran Jagadeesh, Kun Huang, Kunal Chawla, Kyle Huang, Lailin Chen, Lakshya Garg, Lavender A, Leandro Silva, Lee Bell, Lei Zhang, Liangpeng Guo, Licheng Yu, Liron Moshkovich, Luca Wehrstedt, Madian Khabsa, Manav Avalani, Manish Bhatt, Martynas Mankus, Matan Hasson, Matthew Lennie, Matthias Reso, Maxim Groshev, Maxim Naumov, Maya Lathi, Meghan Keneally, Miao Liu, Michael L. Seltzer, Michal Valko, Michelle Restrepo, Mihir Patel, Mik Vyatskov, Mikayel Samvelyan, Mike Clark, Mike Macey, Mike Wang, Miquel Jubert Hermoso, Mo Metanat, Mohammad Rastegari, Munish Bansal, Nandhini Santhanam, Natascha Parks, Natasha White, Navyata Bawa, Nayan Singhal, Nick Egebo, Nicolas Usunier, Nikhil Mehta, Nikolay Pavlovich Laptev, Ning Dong, Norman Cheng, Oleg Chernoguz, Olivia Hart, Omkar Salpekar, Ozlem Kalinli, Parkin Kent, Parth Parekh, Paul Saab, Pavan Balaji, Pedro Rittner, Philip Bontrager, Pierre Roux, Piotr Dollar, Polina Zvyagina, Prashant Ratanchandani, Pritish Yuvraj, Qian Liang, Rachad Alao, Rachel Rodriguez, Rafi Ayub, Raghotham Murthy, Raghu Nayani, Rahul Mitra, Rangaprabhu Parthasarathy, Raymond Li, Rebekkah Hogan, Robin Battey, Rocky Wang, Russ Howes, Ruty Rinott, Sachin Mehta, Sachin Siby, Sai Jayesh Bondu, Samyak Datta, Sara Chugh, Sara Hunt, Sargun Dhillon, Sasha Sidorov, Satadru Pan, Saurabh Mahajan, Saurabh Verma, Seiji Yamamoto, Sharadh Ramaswamy, Shaun Lindsay, Shaun Lindsay, Sheng Feng, Shenghao Lin, Shengxin Cindy Zha, Shishir Patil, Shiva Shankar, Shuqiang Zhang, Shuqiang Zhang, Sinong Wang, Sneha Agarwal, Soji Sajuyigbe, Soumith Chintala, Stephanie Max, Stephen Chen, Steve Kehoe, Steve Satterfield, Sudarshan Govindaprasad, Sumit Gupta, Summer Deng, Sungmin Cho, Sunny Virk, Suraj Subramanian, Sy Choudhury, Sydney Goldman, Tal Remez, Tamar Glaser, Tamara Best, Thilo Koehler, Thomas Robinson, Tianhe Li, Tianjun Zhang, Tim Matthews, Timothy Chou, Tzook Shaked, Varun Vontimitta, Victoria Ajayi, Victoria Montanez, Vijai Mohan, Vinay Satish Kumar, Vishal Mangla, Vlad Ionescu, Vlad Poenaru, Vlad Tiberiu Mihailescu, Vladimir Ivanov, Wei Li, Wenchen Wang, Wenwen Jiang, Wes Bouaziz, Will Constable, Xiaocheng Tang, Xiaojian Wu, Xiaolan Wang, Xilun Wu, Xinbo Gao, Yaniv Kleinman, Yanjun Chen, Ye Hu, Ye Jia, Ye Qi, Yenda Li, Yilin Zhang, Ying Zhang, Yossi Adi, Youngjin Nam, Yu, Wang, Yu Zhao, Yuchen Hao, Yundi Qian, Yunlu Li, Yuzi He, Zach Rait, Zachary DeVito, Zef Rosnbrick, Zhaoduo Wen, Zhenyu Yang, Zhiwei Zhao, and Zhiyu Ma. The llama 3 herd of models, 2024. URL https://arxiv.org/abs/2407.21783.

Zeyu Han, Chao Gao, Jinyang Liu, Jeff Zhang, and Sai Qian Zhang. Parameter-efficient fine-tuning for large models: A comprehensive survey, 2024. URL https://arxiv.org/abs/2403.14608.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. In *Proceedings of the International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=nZeVKeeFYf9.

Chengsong Huang, Qian Liu, Bill Yuchen Lin, Tianyu Pang, Chao Du, and Min Lin. Lorahub: Efficient cross-task generalization via dynamic lora composition. In *First Conference on Language Modeling*, 2024.

Moritz Imfeld, Jacopo Graldi, Marco Giordano, Thomas Hofmann, Sotiris Anagnostidis, and Sidak Pal Singh. Transformer fusion with optimal transport. In *International Conference on Learning Representations*, 2024.

Joel Jang, Seungone Kim, Bill Yuchen Lin, Yizhong Wang, Jack Hessel, Luke Zettlemoyer, Hannaneh Hajishirzi, Yejin Choi, and Prithviraj Ammanabrolu. Personalized soups: Personalized large

language model alignment via post-hoc parameter merging. *arXiv preprint arXiv:2310.11564*, 2023.

Xisen Jin, Xiang Ren, Daniel Preotiuc-Pietro, and Pengxiang Cheng. Dataless knowledge fusion by merging weights of language models. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=FCnohuR6AnM.

Daniel Kunin, Javier Sagastuy-Brena, Surya Ganguli, Daniel LK Yamins, and Hidenori Tanaka. Neural mechanics: Symmetry and broken conservation laws in deep learning dynamics. In *International Conference on Learning Representations*, 2021.

Seungeon Lee, Soumi Das, Manish Gupta, and Krishna P. Gummadi. LoRA on the go: Instance-level dynamic LoRA selection and merging. In Maria Liakata, Viviane P. Moreira, Jiajun Zhang, and David Jurgens (eds.), *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 39583–39601, San Diego, California, United States, July 2026. Association for Computational Linguistics. ISBN 979-8-89176-390-6. doi: 10.18653/v1/2026.acl-long.1837. URL https://aclanthology.org/2026.acl-long.1837/.

Jian Liang, Wenke Huang, Xianda Guo, Guancheng Wan, Bo Du, and Mang Ye. Thanora: Task heterogeneity-aware multi-task low-rank adaptation. *arXiv preprint arXiv:2505.18640*, 2025.

Derek Lim, Haggai Maron, Marc T Law, Jonathan Lorraine, and James Lucas. Graph metanetworks for processing diverse neural architectures. In *International Conference on Learning Representations*, 2024.

Michael Matena and Colin Raffel. Merging models with fisher-weighted averaging. In *Advances in Neural Information Processing Systems*, volume 35, pp. 17703–17716, 2022.

Qi Meng, Shuxin Zheng, Huishuai Zhang, Wei Chen, Qiwei Ye, Zhi-Ming Ma, Nenghai Yu, and Tie-Yan Liu. G-sgd: Optimizing relu neural networks in its positively scale-invariant space. In *International Conference on Learning Representations*, 2019.

Ryota Miyano and Yuki Arase. Adaptive lora merge with parameter pruning for low-resource generation. *arXiv preprint arXiv:2505.24174*, 2025.

Aviv Navon, Aviv Shamsian, Ethan Fetaya, Gal Chechik, Nadav Dym, and Haggai Maron. Equivariant deep weight space alignment. In *International Conference on Machine Learning*, pp. 37376–37395, 2024.

Behnam Neyshabur, Russ R Salakhutdinov, and Nati Srebro. Path-sgd: Path-normalized optimization in deep neural networks. *Advances in neural information processing systems*, 28, 2015.

Peter H Schönemann. A generalized solution of the orthogonal procrustes problem. *Psychometrika*, 31(1):1–10, 1966.

Yihua Shao, Xiaofeng Lin, Xinwei Long, Siyu Chen, Minxi Yan, Yang Liu, Ziyang Yan, Ao Ma, Hao Tang, and Jingcai Guo. Icm-fusion: In-context meta-optimized lora fusion for multi-task adaptation. *arXiv preprint arXiv:2508.04153*, 2025.

Berfin Simsek, François Ged, Arthur Jacot, Francesco Spadaro, Cl'ement Hongler, Wulfram Gerstner, and Johanni Brea. Geometry of the loss landscape in overparameterized neural networks: Symmetries and invariances. In *International Conference on Machine Learning*, pp. 9722–9732, 2021.

Sidak Pal Singh and Martin Jaggi. Model fusion via optimal transport. *Advances in Neural Information Processing Systems*, 33:22045–22055, 2020.

George Stoica, Pratik Ramesh, Boglarka Ecsedi, Leshem Choshen, and Judy Hoffman. Model merging with svd to tie the knots. In *The Thirteenth International Conference on Learning Representations*, 2025.

Zhan Su, Fengran Mo, Guojun Liang, Jinghan Zhang, Bingbing Wen, Prayag Tiwari, and Jian-Yun Nie. Tensorized clustered lora merging for multi-task interference. *arXiv preprint arXiv:2508.03999*, 2025.

Mirac Suzgun, Nathan Scales, Nathanael Schärli, Sebastian Gehrmann, Yi Tay, Hyung Won Chung, Aakanksha Chowdhery, Quoc V. Le, Ed H. Chi, Denny Zhou, and Jason Wei. Challenging big-bench tasks and whether chain-of-thought can solve them, 2022. URL https://arxiv.org/abs/2210.09261.

Pengwei Tang, Xiaolin Hu, Yong Liu, Lizhong Ding, Dongjie Zhang, Xing Wu, and Debing Zhang. Put the space of lora initialization to the extreme to preserve pre-trained knowledge. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 33232–33240, 2026.

Hoang V. Tran, Thieu Vo, An Nguyen The, Tho Tran Huu, Minh-Khoi Nguyen-Nhat, Thanh Tran, Duy-Tung Pham, and Tan Minh Nguyen. Equivariant neural functional networks for transformers. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=uBai0ukstY.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R Bowman. Glue: A multi-task benchmark and analysis platform for natural language understanding. In *Proceedings of the 2018 EMNLP workshop BlackboxNLP: Analyzing and interpreting neural networks for NLP*, pp. 353–355, 2018.

Hongyi Wang, Mikhail Yurochkin, Yuekai Sun, Dimitris Papailiopoulos, and Yasaman Khazaeni. Federated learning with matched averaging. In *International Conference on Learning Representations*, 2020.

Mitchell Wortsman, Gabriel Ilharco, Samir Yitzhak Gadre, Rebecca Roelofs, Raphael Gontijo-Lopes, Ari S Morcos, Hongseok Namkoong, Ali Farhadi, Yair Carmon, Simon Kornblith, and Ludwig Schmidt. Model soups: averaging weights of multiple fine-tuned models improves accuracy without increasing inference time. In *International Conference on Machine Learning*, pp. 23965–23998. PMLR, 2022.

Xun Wu, Shaohan Huang, and Furu Wei. Mixture of loRA experts. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=uWvKBCYh4S.

Shitao Xiao, Zheng Liu, Peitian Zhang, and Xingrun Xing. Lm-cocktail: Resilient tuning of language models via model merging. In *Findings of the Association for Computational Linguistics ACL 2024*, pp. 2474–2488, 2024.

Prateek Yadav, Derek Tam, Leshem Choshen, Colin A Raffel, and Mohit Bansal. Ties-merging: Resolving interference when merging models. *Advances in Neural Information Processing Systems*, 36:7093–7115, 2023.

Le Yu, Bowen Yu, Haiyang Yu, Fei Huang, and Yongbin Li. Language models are super mario: Absorbing abilities from homologous models as a free lunch. In *Forty-first International Conference on Machine Learning*, 2024.

Binchi Zhang, Zaiyi Zheng, Zhengzhang Chen, and Jundong Li. Beyond the permutation symmetry of transformers: The role of rotation for model fusion. In *Forty-second International Conference on Machine Learning*, 2025a.

Haobo Zhang and Jiayu Zhou. Unraveling lora interference: Orthogonal subspaces for robust model merging. *arXiv preprint arXiv:2505.22934*, 2025.

Juzheng Zhang, Jiacheng You, Ashwinee Panda, and Tom Goldstein. LoRI: Reducing cross-task interference in multi-task low-rank adaptation. In *Second Conference on Language Modeling*, 2025b. URL https://openreview.net/forum?id=b8cW86QcOD.

Bo Zhao, Nima Dehmamy, Robin Walters, and Rose Yu. Symmetry teleportation for accelerated optimization. *Advances in neural information processing systems*, 35:16679–16690, 2022.

Bo Zhao, Iordan Ganev, Robin Walters, Rose Yu, and Nima Dehmamy. Symmetries, flat minima, and the conserved quantities of gradient flow. In *The Eleventh International Conference on Learning Representations*, 2023.

Bo Zhao, Robert M Gower, Robin Walters, and Rose Yu. Improving convergence and generalization using parameter symmetries. In *The Twelfth International Conference on Learning Representations*, 2024a.

Bo Zhao, Robin Walters, and Rose Yu. Symmetry in neural network parameter spaces. *arXiv preprint arXiv:2506.13018*, 2025a.

Ziyu Zhao, Leilei Gan, Guoyin Wang, Wangchunshu Zhou, Hongxia Yang, Kun Kuang, and Fei Wu. LoraRetriever: Input-aware LoRA retrieval and composition for mixed tasks in the wild. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar (eds.), *Findings of the Association for Computational Linguistics: ACL 2024*, pp. 4447–4462, Bangkok, Thailand, August 2024b. Association for Computational Linguistics. doi: 10.18653/v1/2024.findings-acl.263. URL https://aclanthology.org/2024.findings-acl.263/.

Ziyu Zhao, Tao Shen, Didi Zhu, Zexi Li, Jing Su, Xuwu Wang, and Fei Wu. Merging loRAs like playing LEGO: Pushing the modularity of loRA to extremes through rank-wise clustering. In *The Thirteenth International Conference on Learning Representations*, 2025b. URL https://openreview.net/forum?id=j6fsbpAllN.

Ziyu Zhao, Yixiao Zhou, Zhi Zhang, Didi Zhu, Tao Shen, Zexi Li, Jinluan Yang, Xuwu Wang, Jing Su, Kun Kuang, et al. Each rank could be an expert: Single-ranked mixture of experts lora for multi-task learning. *arXiv preprint arXiv:2501.15103*, 2025c.

Shenghe Zheng, Hongzhi Wang, Chenyu Huang, Xiaohui Wang, Tao Chen, Jiayuan Fan, Shuyue Hu, and Peng Ye. Decouple and orthogonalize: A data-free framework for lora merging. *arXiv preprint arXiv:2505.15875*, 2025.

Allan Zhou, Kaien Yang, Kaylee Burns, Adriano Cardace, Yiding Jiang, Samuel Sokota, J Zico Kolter, and Chelsea Finn. Permutation equivariant neural functionals. *Advances in neural information processing systems*, 36:24966–24992, 2023a.

Allan Zhou, Kaien Yang, Yiding Jiang, Kaylee Burns, Winnie Xu, Samuel Sokota, J Zico Kolter, and Chelsea Finn. Neural functional transformers. *Advances in neural information processing systems*, 36:77485–77502, 2023b.

Liu Ziyin, Mingze Wang, Hongchao Li, and Lei Wu. Parameter symmetry and noise equilibrium of stochastic gradient descent. *arXiv preprint arXiv:2402.07193*, 2024.

APPENDIX

A Limitations **17**

B Notation and Symbols **17**

C Extended Related Work **17**

D Experimental Details. **18**

D.1 Implementation Details 18

D.2 Benchmarks 19

D.3 Baselines 20

E Additional Experiments. **21**

E.1 Least-Square Alignment (LSA) 21

E.2 Ablation Study 22

E.3 Two-Stage Design Validation: SOA Initialization vs. Random Initialization 22

F Extended Analyses **23**

F.1 Anchor Sensitivity with the Medoid Rule 24

F.2 Results on Qwen 24

F.3 Protocol-Matched Leave-Target-Out LoRAHub 25

F.4 Leave-Target-Out LoraRetriever Fusion 25

F.5 Negative Control (Anchor-Repulsion) and LoRI-D Comparison 26

F.6 Per-Task Breakdown of Magnitude-Prune Gains 27

G Theoretical Proof **27**

A LIMITATIONS

Our method has two main limitations, which also delineate its intended scope. First, our alignment requires the target and anchor factors to have matching shapes, which precludes its direct use for fusing LoRA adapters of different ranks. Second, and more fundamentally, our framework is designed for *factor-space* LoRA fusion: it requires explicit access to the low-rank factors A and B before fusion, and applies the LRD-preserving reparameterization $(B_i R_i, R_i^{-1} A_i)$ to select a shared reference gauge. Consequently, it is not directly applicable to fusion pipelines that operate solely on the merged update ΔW; once a pipeline collapses an adapter into ΔW and no longer exposes A/B, the corresponding gauge-alignment problem must be reformulated in update space, which we leave to future work. We also emphasize that the reference adapter defines a shared gauge for fusion rather than a globally canonical basis for all adapters: the optimization reduces, but does not necessarily eliminate, the factor-space mismatch caused by low-rank non-identifiability.

B NOTATION AND SYMBOLS

We summarize the notation used throughout the paper in Table 6.

Symbol	Domain	Meaning
W_0	$\mathbb{R}^{d\times k}$	Frozen pre-trained weight matrix of a target layer.
ΔW	$\mathbb{R}^{d\times k}$	Low-rank weight update induced by a LoRA adapter, $\Delta W = BA$.
$B,\ A$	$\mathbb{R}^{d\times r},\ \mathbb{R}^{r\times k}$	The two low-rank LoRA factors: A is the input-side down projection and B the output-side up projection.
r	integer	LoRA rank, with $r \ll \min(d, k)$.
$d,\ k$	integer	Output and input dimensions of the target layer.
$x,\ h$	vector	Input activation to the layer and the resulting output.
$\theta_i = \{A_i, B_i\}$	adapter	The i-th task-specific LoRA adapter (module).
N	integer	Number of task-specific adapters to be fused.
θ_0	adapter	Anchor (reference) adapter selected by the medoid rule.
θ_f	adapter	Fused adapter produced by the fusion function.
$f(\cdot)$	mapping	Fusion function combining a set of adapters into one.
R	$\mathbb{R}^{r\times r}$	Invertible reparameterization / alignment transform on the rank-r space.
$R^{-1},\ R^\top$	$\mathbb{R}^{r\times r}$	Inverse and transpose of R (equal when R is orthogonal).
$\tilde{B},\ \tilde{A}$	matrix	Reparameterized factors $\tilde{B} = BR$, $\tilde{A} = R^{-1}A$, preserving ΔW.
$\mathcal{O}(r)$	set	Orthogonal group $\{R \in \mathbb{R}^{r\times r} \mid R^\top R = I\}$.
I	$\mathbb{R}^{r\times r}$	Identity matrix.
$M,\ M_{\text{joint}}$	$\mathbb{R}^{r\times r}$	Correlation matrices for the A/B-only and joint SOA objectives.
$U\Sigma V^\top$	factorization	Singular value decomposition of the correlation matrix.
$R_i^* = UV^\top$	$\mathcal{O}(r)$	Closed-form optimal orthogonal alignment for adapter i.
$\|\cdot\|_F$	scalar	Frobenius norm.
$\mathcal{L}_{\text{GRA}}^{\text{A}},\ \mathcal{L}_{\text{GRA}}^{\text{B}},\ \mathcal{L}_{\text{GRA}}^{\text{J}}$	scalar	GRA refinement losses (A-only, B-only, and joint).
$\mathcal{L}_{\mathcal{O}} = \|RR^\top - I\|_F^2$	scalar	Orthogonality regularizer keeping R near-orthogonal.
λ	scalar	Weight of the orthogonality regularizer (default 10^{-3}).
η	scalar	Learning rate used in GRA gradient descent.
T	integer	Number of GRA refinement steps.
$\tilde{\theta}_i = \{R_i^{-1}A_i,\ B_iR_i\}$	adapter	Final aligned adapter used for fusion.

Table 6: Summary of frequently used notation.

C EXTENDED RELATED WORK

LoRA Fusion Fusing multiple LoRA adapters enables a model to combine skills from different fine-tuned experts, but a key challenge is interference between their parameters. Directly summing LoRA updates trained on diverse tasks can cause performance degradation, especially when their weight magnitudes differ widely (Zhang & Zhou, 2025). Recent work has proposed various strategies

to mitigate such conflicts: (1) Reweighting and sparsification (Yadav et al., 2023; Jang et al., 2023; Miyano & Arase, 2025; Liang et al., 2025; Xiao et al., 2024). These methods compute the importance of LoRA parameters and prune conflicting directions to address interference when combining the adapters. (2) Subspace factorization (Zheng et al., 2025; Zhang & Zhou, 2025; Tang et al., 2026; Stoica et al., 2025). Approaches based on subspace factorization explicitly enforce null-space projections so that task information occupies minimally overlapping subspaces before fusion. (3) Modular composition (Huang et al., 2024; Su et al., 2025; Zhao et al., 2025c;b). This type of technique treats rank-1 components as composable units, clustering, retrieving, or routing them to assemble a merged adapter with selective rank-wise activation. Complementary strategies such as ICM-Fusion (Shao et al., 2025) leverage in-context meta-optimization to balance conflicting LoRA update directions across tasks. Together, these lines of work highlight three pillars for LoRA fusion: scale calibration (to avoid domination), geometric disentanglement (to reduce overlap), and modularity (to enable selective composition). Our method is orthogonal to and compatible with these directions: it operates in the LoRA factor space with explicit access to A and B, and can be applied as a preprocessing step for any factor-space fusion rule. In contrast, methods that alter adapter construction at training time (e.g., LoRI (Zhang et al., 2025b), which trains task adapters toward approximately orthogonal update subspaces) or fuse only through the materialized update ΔW (e.g., KnOTS (Stoica et al., 2025)) target a different geometric object; because our reparameterization preserves each ΔW_i exactly, it neither conflicts with such training-time orthogonality nor addresses pure ΔW-space fusion (see Section A and Section F.5). A related distinction concerns retrieval- and routing-based composition. Pipelines that first select a subset of adapters and then combine them through their low-rank factors, such as LoraRetriever (Zhao et al., 2024b), still expose A and B at fusion time and therefore fall within our scope; we verify this in Section F.4. By contrast, approaches that keep adapters separate and combine only their outputs through learned gating, such as Mixture of LoRA Experts (Wu et al., 2024) and LoRA on the Go (Lee et al., 2026), never perform factor-space arithmetic, so they lie outside the applicable scope rather than acting as competing baselines.

Parameter Space Symmetry Parameter space symmetry refers to specific transformations of neural network weights that preserve the model's functionality. Beyond classical invariances such as rescaling (Neyshabur et al., 2015; Badrinarayanan et al., 2015; Du et al., 2018; Meng et al., 2019; Kunin et al., 2021) and translation (Kunin et al., 2021), which illuminate optimization geometry and have been used to stabilize training dynamics (Zhao et al., 2022; 2023; 2024a; Ziyin et al., 2024), a critical form of symmetry for merging is permutation invariance (Brea et al., 2019; Simsek et al., 2021; Entezari et al., 2022; Ainsworth et al., 2023), where neurons are reindexed without changing the realized function. This observation motivates weight-matching procedures that reorder units or features to place models in a common gauge before averaging (Wang et al., 2020; Singh & Jaggi, 2020; Imfeld et al., 2024; Navon et al., 2024). Recent work extends from discrete to continuous alignments: rotation-based methods (Zhang et al., 2025a; Zhao et al., 2025a) learn orthogonal changes of basis for attention layers in Transformers. Optimal-transport alignments cast neuron correspondence as a transport problem that yields soft or hard matchings prior to fusion (Imfeld et al., 2024). In parallel, neural functionals and metanetworks develop permutation equivariant operators that act directly on weights to compose models across architectures (Navon et al., 2024; Zhou et al., 2023b;a; Lim et al., 2024; Tran et al., 2025). In this paper, we perform symmetry-aware matching by aligning LoRA bases in the symmetry class before composition, thereby reducing interference in merging.

D Experimental Details

D.1 Implementation Details

Platform. Our implementation is based on Python 3.11 and Pytorch 2.7.1. All fine-tuning, alignment, and LoRA fusion processes are conducted on a cluster equipped with 4 Nvidia A100 80GB GPUs.

Fine-tuning and Alignment. For each downstream task, we train task-specific LoRA adapters (Hu et al., 2022) with rank $r = 16$, $\alpha = 16$, and dropout 0.05. Fine-tuning is conducted for up to 3 epochs (or equivalent steps depending on the dataset size) with a learning rate of $1e{-4}$ and a batch size of 16. After training, we apply the proposed pre-fusion alignment framework. We align all remaining adapters toward the anchor using SOA followed by GRA for 100 steps with a learning rate of $1e{-5}$. Aligned main-paper results use the medoid anchor (the adapter with the smallest mean Frobenius distance to the rest) as a fixed, data-free gauge. For the sensitivity analysis in Section F.1,

Algorithm 1 Pre-fusion Alignment for LoRA Fusion

Require: N LoRA adapters $\{\theta_i\}_{i=0}^{N-1}$ where $\theta_i = \{A_i, B_i\}$.
Require: Anchor index k (default: the medoid adapter, i.e. $\arg\min_k \frac{1}{N-1}\sum_{j\neq k}\|\theta_j - \theta_k\|_F$), Alignment Mode (Joint/A-only/B-only).
Require: Refinement steps T (default: 100), Learning rate η (default: 1e-5), Loss weight λ (default: 1e-3).
Ensure: Aligned set of LoRA adapters $\{\theta_k\} \cup \{\tilde{\theta}_i\}_{i\neq k}$.

Switch anchor parameters $\theta_0, \theta_k \leftarrow \theta_k, \theta_0$.
for $i = 1$ to $N-1$ **do**
  // Stage 1: Geometric Alignment (SOA)
  if mode is Joint **then**
    $M \leftarrow B_i^\top B_0 + A_i A_0^\top$
  else if mode is A-only **then**
    $M \leftarrow A_i A_0^\top$
  else if mode is B-only **then**
    $M \leftarrow B_i^\top B_0$
  end if
  Compute SVD: $U, \Sigma, V^\top \leftarrow \text{SVD}(M)$
  Initialize rotation: $R_i \leftarrow UV^\top$
  // Stage 2: Refined Alignment (GRA)
  for $t = 1$ to T **do**
    if mode is Joint **then**
      $\mathcal{L} \leftarrow \|B_i R_i - B_0\|_F^2 + \|R_i^{-1} A_i - A_0\|_F^2$
    else if mode is A-only **then**
      $\mathcal{L} \leftarrow \|R_i^{-1} A_i - A_0\|_F^2$
    else if mode is B-only **then**
      $\mathcal{L} \leftarrow \|B_i R_i - B_0\|_F^2$
    end if
    Update $R_i \leftarrow R_i - \eta \nabla_{R_i}(\mathcal{L} + \lambda \|R_i R_i^\top - I\|_F^2)$
  end for
  // Apply Transformation
  Update $\tilde{B}_i \leftarrow B_i R_i$, $\tilde{A}_i \leftarrow R_i^{-1} A_i$
end for
Switch anchor parameters $\theta_0, \theta_k \leftarrow \theta_k, \theta_0$

we repeat the same alignment for every anchor choice and report the mean $\pm$ standard deviation over all six anchors. To validate the role of initialization, we additionally compare against direct gradient-based refinement from random initialization. Besides downstream task performance, we also analyze factor-space misalignment before and after alignment, anchor sensitivity, layer-wise effects, and computational overhead.

Full Procedure. Algorithm 1 states the complete pre-fusion alignment procedure, covering anchor selection, the closed-form SOA stage, the GRA refinement loop, and the final LRD-preserving reparameterization applied to each non-anchor adapter.

D.2 Benchmarks

D.2.1 GLUE

We select the following six subtasks from the GLUE benchmark.[1]

CoLA (Corpus of Linguistic Acceptability). A binary classification task that evaluates whether a given sentence is grammatically acceptable in English.

MNLI (Multi-Genre Natural Language Inference). A natural language inference task that determines whether a premise entails, contradicts, or is neutral with respect to a hypothesis across multiple text genres.

[1] https://huggingface.co/datasets/nyu-mll/glue

MRPC (Microsoft Research Paraphrase Corpus). A paraphrase identification task that judges whether two sentences are semantically equivalent.

QNLI (Question-answering Natural Language Inference). A sentence-pair classification task that determines whether a given sentence contains the answer to a corresponding question.

RTE (Recognizing Textual Entailment). A binary entailment task that predicts whether a hypothesis can be inferred from a premise.

SST-2 (Stanford Sentiment Treebank). A sentiment analysis task that classifies sentences into positive or negative sentiment.

D.2.2 NLI

In addition to the following three subtasks, we also incorporate MNLI, QNLI, and RTE tasks from the GLUE benchmark.

SciTail (Science Textual Entailment).[2] A natural language inference task derived from science question–answer pairs, where the model predicts whether a premise entails a hypothesis in a scientific context.

SICK (Sentences Involving Compositional Knowledge).[3] An NLI dataset that evaluates entailment relations between sentence pairs, with a focus on compositional semantics and logical reasoning.

SNLI (Stanford Natural Language Inference).[4] A large-scale natural language inference benchmark that classifies sentence pairs into entailment, contradiction, or neutral.

D.3 BASELINES

Simple (Arithmetic Average). A simple merging baseline that directly averages the LoRA weight updates across adapters.

Magnitude-Prune (Magnitude-based Pruning). A data-agnostic method that prunes low-magnitude LoRA updates and merges the remaining parameters.

TIES (TrIm, Elect Sign & Merge). A merging algorithm that trims small updates, resolves sign conflicts, and merges aligned parameters.

DARE (Drop And Rescale). A method that randomly drops a fraction of updates and rescales the rest to reduce interference before merging.

Fisher (Fisher-Weighted Averaging). An importance-weighted fusion approach that averages parameters using Fisher information as weights.

RegMean (Regression Mean). A fusion method that solves for layer-wise merge coefficients via regression to best match model outputs.

LoRAHub. A dynamic composition method that learns task-specific weights to combine multiple LoRA adapters from a few examples.

ALM (Adaptive LoRA Merge). A method that tunes the per-adapter merge weights on a small amount of target-task data and prunes low-contribution parameters, so that the merged adapter adapts to the target distribution rather than using fixed uniform coefficients.

Git-Rebasin. A permutation-based alignment baseline that searches for a permutation of hidden units bringing one model into correspondence with another before averaging. We apply it to the LoRA factors so that it plays the same pre-fusion alignment role as SOA and GRA, which lets us isolate the gain from permutation matching alone.

[2] https://huggingface.co/datasets/allenai/scitail
[3] https://huggingface.co/datasets/RobZamp/sick
[4] https://huggingface.co/datasets/stanfordnlp/snli

E Additional Experiments

E.1 Least-Square Alignment (LSA)

E.1.1 Method

In Section 3.3, we employ gradient descent to explicitly reduce the discrepancy between two LoRA adapters in the parameter space. In this section, we attempt to derive a closed-form solution to the corresponding optimization problem. Specifically, for the joint objective, the loss function in Equation (10) is transferred into:

$$\min_{R}(||R^{-1}A - A_0||_F^2 + ||BR - B_0||_F^2), \tag{12}$$

$$\text{s.t. } R \in \mathbb{R}^{r\times r}, \det(R) \neq 0. \tag{13}$$

Where $\theta_0 = \{A_0, B_0\}$ represents the anchor LoRA adapter. Unfortunately, the above objective does not admit a closed-form solution in the general case. Consequently, following the design philosophy of the A-only and B-only variants in the main text, we decompose it into two sub-objectives (LSA(A) and LSA(B)) and analyze them individually. In such cases, the optimal solution to LSA(A) is obtained directly via a least-squares formulation:

$$R^* = \left(A_0 A^\top (AA^\top)^{-1}\right)^{-1}. \tag{14}$$

Similarly, we obtain the optimal R^* for LSA(B) as follows:

$$R^* = (B^\top B)^{-1} B^\top B_0. \tag{15}$$

Rescaling. Meanwhile, since we remove the orthogonality constraint on R, the Frobenius norms $\|\tilde{A}\|_F$ and $\|\tilde{B}\|_F$ are no longer guaranteed to equal $\|A\|_F$ and $\|B\|_F$, respectively. To prevent the transformation from introducing imbalanced scaling effects on matrices A and B, we further incorporate an explicit rescaling step. Specifically, we multiply R and divide R^{-1} by a common scaling factor to ensure that their rescaled norms are equal, i.e.,

$$\tilde{R} = \alpha R, \qquad \tilde{R}^{-1} = \alpha^{-1} R^{-1}, \tag{16}$$

where the scaling factor α is chosen such that

$$\|\tilde{R}\|_F = \|\tilde{R}^{-1}\|_F, \tag{17}$$

which yields

$$\alpha = \sqrt{\frac{\|R^{-1}\|_F}{\|R\|_F}}. \tag{18}$$

We denote this variant as Least-Squares Alignment with Rescaling (LSAR).

E.1.2 Empirical Results

Figure 5 reports the results of extending SOA to the least-squares alignment (LSA) and its rescaled variant (LSAR). Each cell retains the absolute accuracy while color encodes the change from the corresponding unaligned baseline. ***(i)* Non-orthogonality alone is unstable.** Compared with the generally more reliable SOA results, directly relaxing the orthogonality constraint (LSA) frequently causes substantial performance degradation across both benchmarks. This indicates that the extra degrees of freedom in R tend to introduce uncontrolled scaling and conditioning, which disrupts the balance of the low-rank factors and harms fusion. ***(ii)* Rescaling is essential to unlock LSA.** We observe that the *A-only* variant yields improvements only on a subset of baselines, while the *B-only* variant exhibits severe and inconsistent degradation. This asymmetry indicates that LSA alone fails to address a fundamental issue. Incorporating rescaling (LSAR) largely mitigates these failures and often recovers or surpasses the gains of SOA, suggesting that the main bottleneck of LSA is the scale imbalance induced by unconstrained R and R^{-1}. ***(iii)* A-side alignment remains the dominant contributor.** Across GLUE and NLI, the A-only variants are generally more effective than the B-only ones under non-orthogonal transformations, implying that correcting misalignment in the input-side latent subspace is the primary driver of fusion quality, whereas adjusting output projections alone is comparatively insufficient.

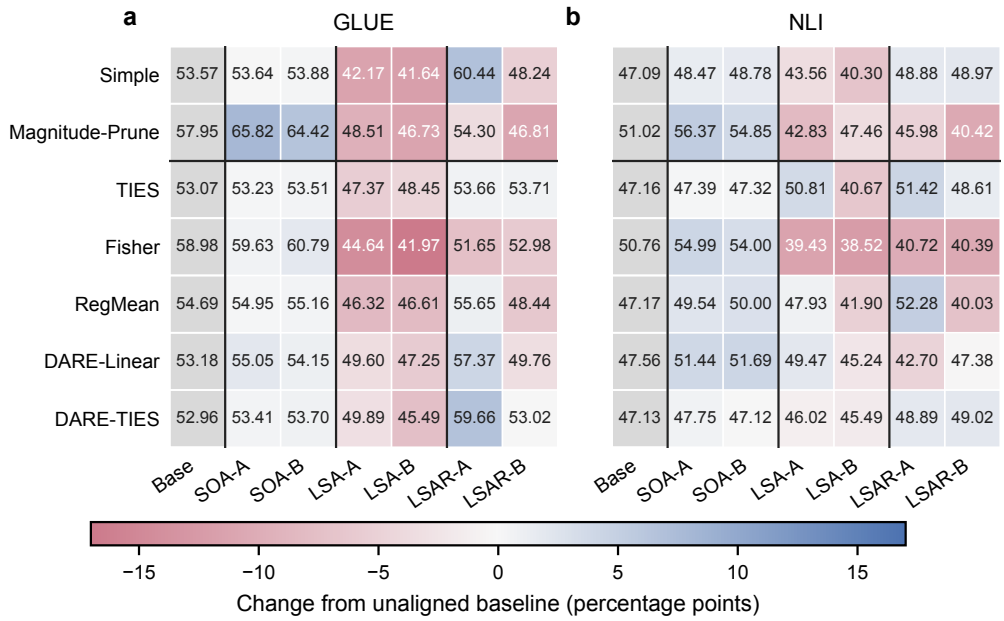

Figure 5: SOA, unconstrained LSA, and rescaled LSAR on GLUE and NLI. Cell text gives absolute accuracy; the gray Base column gives the unaligned score, and color gives each aligned cell's percentage-point change from it. SOA is generally stable, whereas unconstrained LSA often degrades performance and LSAR recovers only a subset of settings.

E.2 Ablation Study

In this section, we examine the contribution of the orthogonality loss term $\mathcal{L}_{\mathcal{O}} = \|R_i R_i^\top - I\|_F^2$. Specifically, we conduct an ablation study by directly optimizing the transformation R_i with:

$$R_i^{(t+1)} \leftarrow R_i^{(t)} - \eta \nabla_{R_i} \mathcal{L}_{\text{GRA}}, \tag{19}$$

where η represents the learning rate. Figure 6 reports the ablation performance on the GLUE benchmark as gains over each unaligned fusion baseline. From the results, we make the following observations: ***(i)*** In most cases, incorporating $\mathcal{L}_{\mathcal{O}}$ leads to performance improvements, underscoring the importance of orthogonality regularization. ***(ii)*** Even without $\mathcal{L}_{\mathcal{O}}$, all reported variants outperform their unaligned baselines and most outperform the corresponding SOA result in Figure 5, providing descriptive evidence for the $\mathcal{L}_{\text{GRA}}$ loss design in eq. (9) and eq. (10).

E.3 Two-Stage Design Validation: SOA Initialization vs. Random Initialization

We investigate the benefit of the proposed two-stage design. In particular, we compare our default **GRA** initialization strategy (starting from the SOA solution) against a variant that directly performs gradient-based refinement from random initialization, denoted as $\mathbf{GRA_{RI}}$.

Figure 7 reports the comparison on GLUE and NLI. Direct optimization from random initialization can already improve some fusion baselines, indicating that non-orthogonal refinement itself can be beneficial. SOA-initialized GRA gives a higher reported score in 39 of 42 method–objective comparisons, ties once, and is lower twice; its average gain is larger for every benchmark and objective. These results support the practical benefit of resolving the dominant orthogonal mismatch in closed form before refinement, without claiming that SOA initialization wins every individual setting.

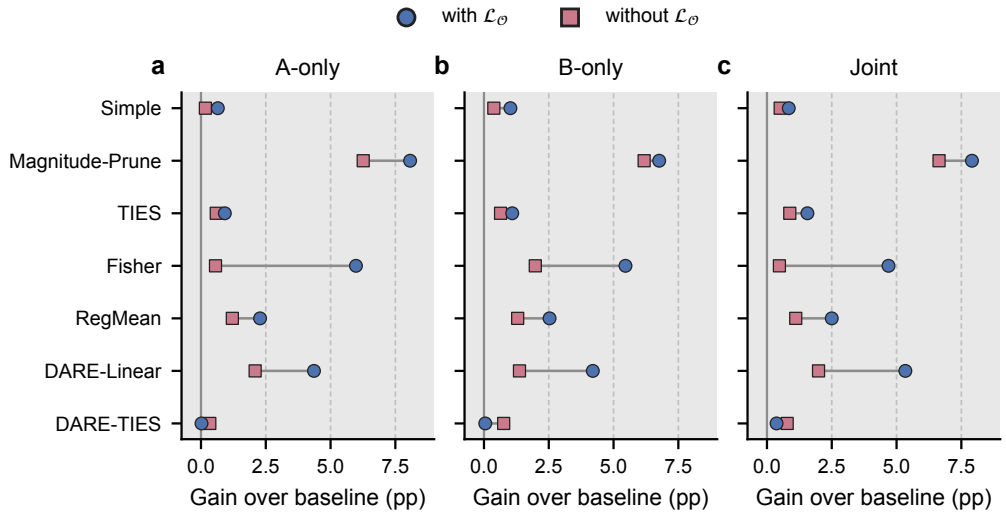

Figure 6: Effect of the orthogonality regularizer $\mathcal{L}_{\mathcal{O}}$ in GRA. Each row compares gains over the same unaligned GLUE baseline with and without the regularizer. The regularized objective is better in 18 of 21 method–objective settings; the three exceptions are all DARE-TIES variants.

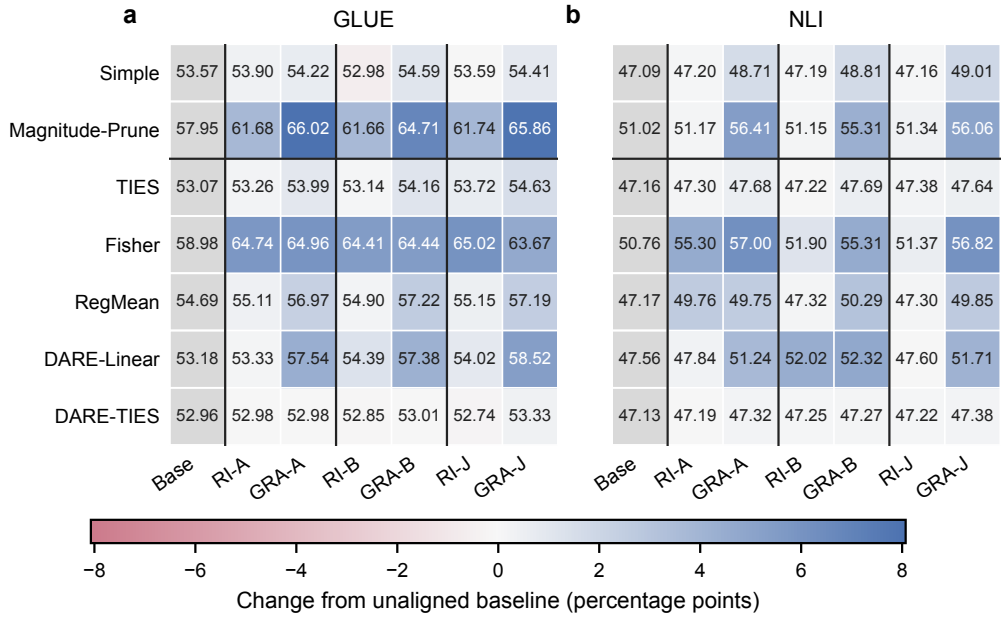

Figure 7: Randomly initialized refinement (RI) versus SOA-initialized GRA on GLUE and NLI. Cell text gives absolute accuracy; the gray Base column gives the unaligned score, and color gives each refinement cell's percentage-point change from it. Across the 42 reported cells, SOA initialization is higher in 39, tied in one, and lower in two.

F Extended Analyses

This appendix collects additional analyses and experiments that further support the main claims: anchor sensitivity with the medoid rule (Section F.1), a newer backbone (Section F.2), protocol-matched leave-target-out comparisons for LoRAHub (Section F.3) and LoraRetriever (Section F.4), a negative-control Anchor-Repulsion study and a LoRI-D training-time comparison (Section F.5), and task-level Magnitude-Prune deltas (Section F.6).

F.1 Anchor Sensitivity with the Medoid Rule

The main tables report a single *medoid* anchor, i.e., the adapter with the smallest mean Frobenius distance to the rest, which we adopt as a fixed, data-free gauge; for the NLI pool this is the MNLI adapter, whose row in Table 7 supplies the NLI/Simple GRA entries of Table 3. For a fixed anchor, each entry first averages accuracy over the six benchmark tasks. Figure 8 then reports the mean and sample standard deviation of these six-task scores across all six anchor choices for three representative fusion methods. These standard deviations measure anchor sensitivity, not training-seed uncertainty or confidence intervals. All all-anchor means remain above their unaligned baselines, although the variation is non-negligible in several GLUE settings.

Anchor task	GRA(A)	GRA(B)	GRA(J)	Avg.
MNLI	48.71	48.81	49.01	48.84
QNLI	48.76	48.72	48.94	48.81
RTE	48.96	49.11	49.04	49.04
SciTail	48.34	48.82	48.73	48.63
SICK	48.59	48.86	48.83	48.76
SNLI	48.63	49.06	48.89	48.86
Mean ± SD	48.67±0.21	48.90±0.15	48.91±0.12	48.82±0.13

Table 7: Individual NLI/Simple anchor runs. Each body row is the six-task accuracy mean for one anchor; the final row reports mean ± sample standard deviation across all six anchors. MNLI is the medoid anchor, so its row gives the NLI/Simple GRA entries of Table 3.

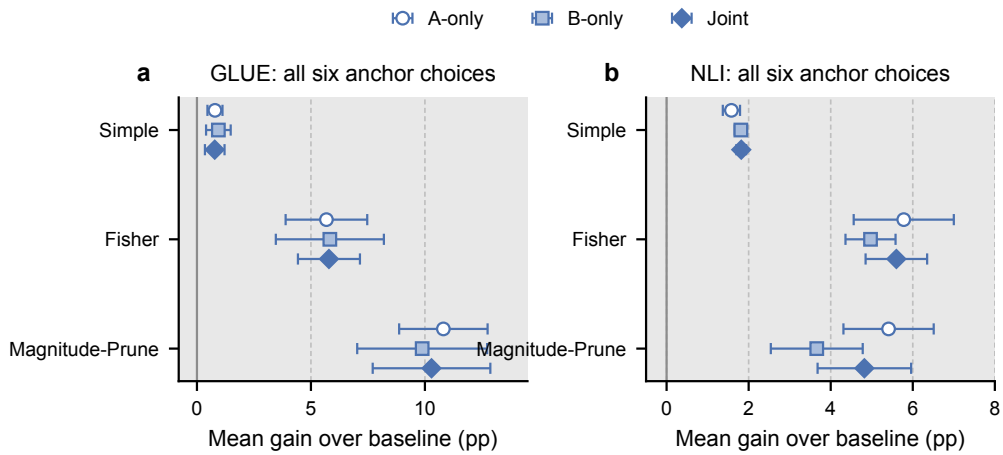

Figure 8: All-anchor sensitivity on GLUE and NLI. Points show the mean gain over the unaligned main-table baseline; error bars show ±1 sample standard deviation across six anchor choices. They do not represent training-seed uncertainty or confidence intervals. Marker shape distinguishes A-only, B-only, and Joint objectives.

F.2 Results on Qwen

GLUE and NLI are used here as controlled LoRA-merging testbeds that provide task-specific adapters and standardized metrics; recent LoRA-merging and alignment work continues to use them as controlled fusion testbeds (Zhao et al., 2025b; Zhang & Zhou, 2025; Stoica et al., 2025). To verify that our conclusions are not tied to the original `Llama-3.1-8B` setting, we additionally run the full SOA/GRA grid on `Qwen3-0.6B` (Figure 9). The Qwen3-0.6B runs show broad positive average gains with method-specific exceptions, indicating that the alignment effect also transfers to a newer backbone.

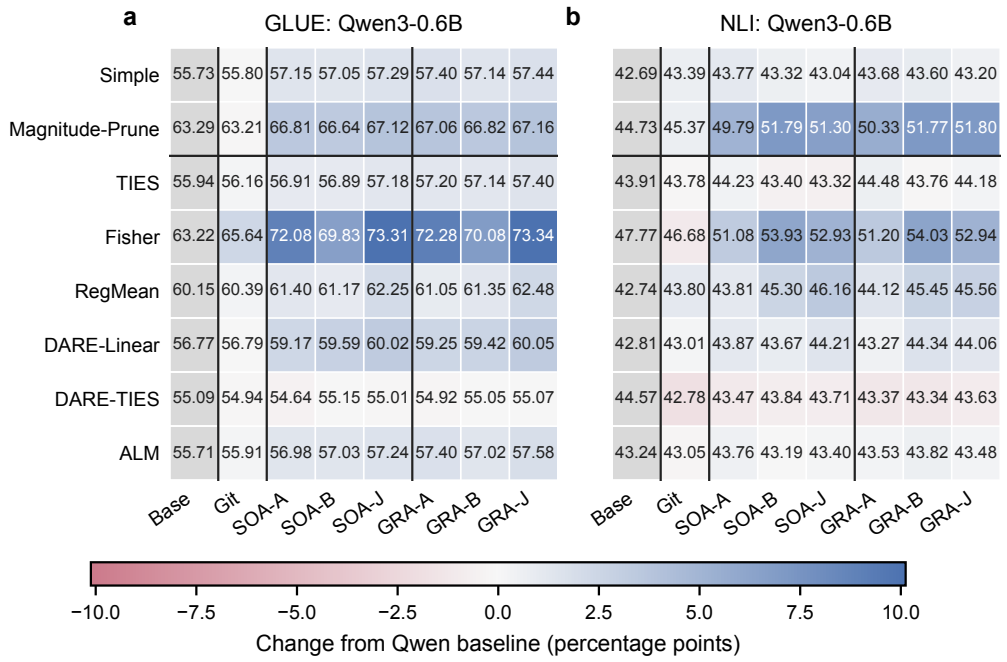

Figure 9: Full alignment grid on `Qwen3-0.6B`. Cell text gives absolute six-task accuracy; the gray Base column gives the unaligned score, and color gives each aligned cell's percentage-point change from it. Positive average gains transfer to the newer backbone, while the visible negative cells make the method-specific exceptions explicit.

F.3 Protocol-Matched Leave-Target-Out LoRAHub

LoRAHub is gradient-free and estimates merge weights from a few target-task examples, so a 5-shot search is consistent with its protocol. However, LoRAHub is designed for cross-task composition on unseen target tasks, whereas our GLUE/NLI setup contains one trained adapter per evaluation subtask; directly including the target task's own adapter would make the comparison less faithful. We therefore add a protocol-matched leave-target-out setting: for each target subtask, we use 5 target-task examples only for LoRAHub weight search, exclude the target task's own adapter from the candidate pool, merge only the remaining adapters, and evaluate on the target task (Table 8). Even for this weight-search-based method, post-hoc factor alignment remains applicable and improves the resulting cross-task composition.

F.4 Leave-Target-Out LoRARetriever Fusion

The relevant distinction for our method is whether a pipeline fuses adapters through the materialized update $\Delta W = BA$ or through the LoRA factors A and B. Because GRA preserves ΔW, it is neutral for pipelines that operate only on the materialized update, and only affects procedures that still compose the factors. This makes LoraRetriever a suitable check: following the same leave-target-out principle, for each target subtask we exclude the target task's own adapter from the retrieval pool, retrieve multiple non-target adapters, and compose the top-k ($k = 3$) retrieved adapters with factor-space fusion (Table 8). Post-hoc factor alignment improves the resulting factor-space fusion over the corresponding LoraRetriever baseline. By contrast, methods such as Mixture of LoRA Experts and LoRA on the Go do not meet the factor-space fusion condition and are therefore outside the applicable scope rather than conflicting baselines.

Pipeline	Benchmark	Baseline	GRA(A)	GRA(B)	GRA(J)
LoRAHub	GLUE	51.01	54.75	54.41	54.62
LoRAHub	NLI	49.00	50.03	50.34	50.64
LoraRetriever	GLUE	53.47	54.79	54.09	54.63
LoraRetriever	NLI	49.39	52.48	52.44	52.63

Table 8: Protocol-matched leave-target-out composition. LoRAHub uses five target-task examples for weight search; LoraRetriever composes the top three retrieved non-target adapters. In both cases, the target task's adapter is excluded from the candidate pool.

F.5 Negative Control (Anchor-Repulsion) and LoRI-D Comparison

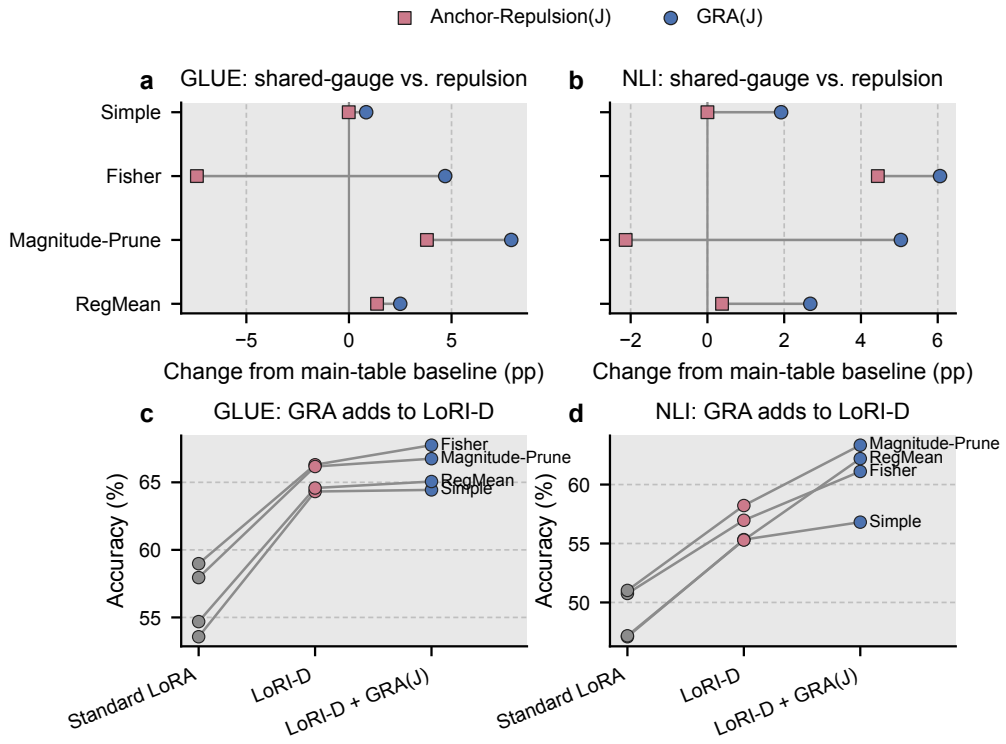

Figure 10: Negative-control and compatibility analyses. (a–b) Changes relative to the unaligned main-table baselines for GRA(J) and Anchor-Repulsion(J). Anchor-Repulsion differs in both objective direction and initialization. (c–d) Standard-LoRA unaligned fusion, separately trained LoRI-D adapters, and GRA(J) applied to the same LoRI-D checkpoints. Only the LoRI-D-to-LoRI-D+GRA transition isolates post-hoc alignment.

Anchor-Repulsion negative control. To test whether the *direction* of alignment matters, we run Anchor-Repulsion on the same standard-LoRA checkpoints, fusion rules, and evaluation splits used in the main tables. As shown in Figure 10a–b, it pushes the joint factors away from the anchor and starts from a random rotator, whereas GRA(J) pulls toward the shared gauge from the SOA solution. The comparison therefore changes both objective direction and initialization and is treated as a negative control rather than a one-variable ablation. The unaligned and GRA(J) values are copied from the main tables. Anchor-Repulsion is below GRA(J) in all eight listed settings, although it can remain above the unaligned baseline in some cases.

LoRI-D training-time comparison. LoRI is a training-time method that learns task adapters toward approximately orthogonal update subspaces, operating on the induced update $\Delta W_i = B_i A_i$; our method instead performs a post-hoc factor-space gauge alignment that preserves each ΔW_i

exactly. The two are therefore complementary rather than conflicting. Figure 10c–d separates two comparisons: the Standard-LoRA points reproduce the unaligned fusion baselines from the main tables and use a different adapter pool from LoRI-D, whereas the within-LoRI-D transition compares the same LoRI-D checkpoints before and after GRA(J). Every within-LoRI-D transition is positive, showing that post-hoc alignment can add to the training-time construction when the fusion rule remains coordinate-sensitive.

F.6 Per-Task Breakdown of Magnitude-Prune Gains

To expose where the Magnitude-Prune gains come from, Figure 11 reports task-level changes under $\mathrm{GRA}(J)$. The aggregate gains are $65.86 - 57.95 = +7.91$ points on GLUE and $56.06 - 51.02 = +5.04$ points on NLI. The per-task distribution is highly heterogeneous, with large gains on some tasks and decreases on others, a pattern compatible with the threshold sensitivity of a hard pruning rule.

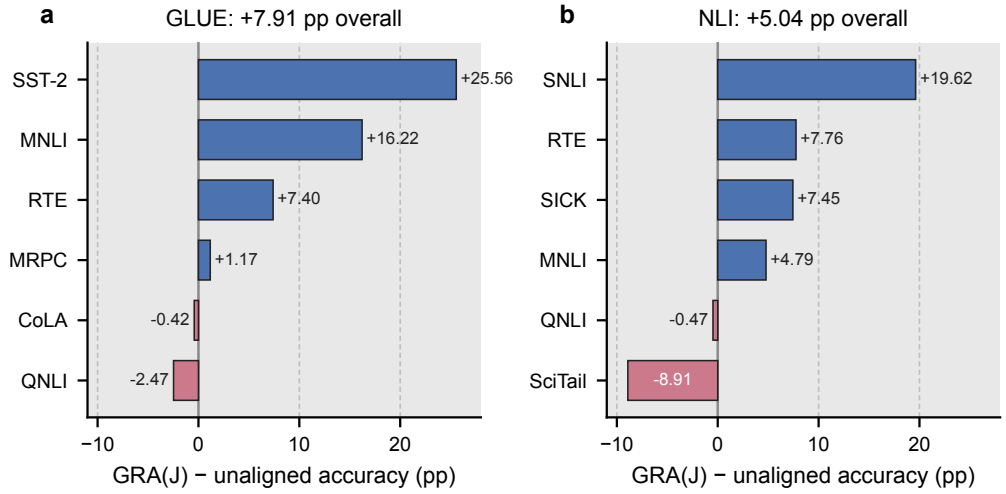

Figure 11: Per-task percentage-point deltas for Magnitude-Prune under GRA(J), computed as aligned minus unaligned accuracy. Blue and pink bars denote positive and negative changes.

G Theoretical Proof

Proposition 1. Let $\theta = \{A, B\}$ be the weight update of a LoRA adapter, where $B \in \mathbb{R}^{d\times r}$ and $A \in \mathbb{R}^{r\times k}$. For any invertible matrix $R \in \mathbb{R}^{r\times r}$ (i.e., $rank(R) = r$), the reparameterization

$$\tilde{B} = BR, \qquad \tilde{A} = R^{-1}A \tag{20}$$

preserves the LoRA update, namely $\tilde{B}\tilde{A} = BA$ (equivalently, $\Delta\tilde{W} = \Delta W$).

Proof. Let $R \in \mathbb{R}^{r\times r}$ be invertible and define $\tilde{B} = BR$ and $\tilde{A} = R^{-1}A$. Then

$$\tilde{B}\tilde{A} = (BR)(R^{-1}A) = B(RR^{-1})A = BA.$$

Hence, the induced LoRA update is unchanged: $\Delta\tilde{W} = \tilde{B}\tilde{A} = BA = \Delta W$. □

Theorem 1. For either objective in Equation (4), we define the correlation matrix as:

$$M = \begin{cases} B_i^\top B_0, & \text{(B-only)}, \\ A_i A_0^\top, & \text{(A-only)}, \end{cases} \tag{21}$$

where $\{A_0, B_0\}$ is the anchor adapter. Let $M = U\Sigma V^\top$ be its singular value decomposition. Then the optimal orthogonal alignment is

$$R^* = UV^\top \in \mathcal{O}(r). \tag{22}$$

Proof. We prove the B-only case; the A-only case follows identically. We consider:

$$\begin{aligned}
&\min_{R\in\mathcal{O}(r)} \|B_iR - B_0\|_F^2 \\
=&\min_{R\in\mathcal{O}(r)} \langle B_iR - B_0,\ B_iR - B_0\rangle_F \\
=&\|B_i\|_F^2 + \|B_0\|_F^2 - 2\max_{R\in\mathcal{O}(r)} \langle B_iR,\ B_0\rangle_F.
\end{aligned}$$

Hence, the above minimization is equivalent to

$$\begin{aligned}
&\max_{R\in\mathcal{O}(r)} \langle B_iR,\ B_0\rangle_F \\
=&\max_{R\in\mathcal{O}(r)} \mathrm{tr}\big((B_iR)^\top B_0\big) \\
=&\max_{R\in\mathcal{O}(r)} \mathrm{tr}\big(R^\top B_i^\top B_0\big) \\
=&\max_{R\in\mathcal{O}(r)} \langle R,\ B_i^\top B_0\rangle_F \\
=&\max_{R\in\mathcal{O}(r)} \langle R,\ M\rangle_F,
\end{aligned}$$

which is an Orthogonal Procrustes problem (Schönemann, 1966; Gower & Dijksterhuis, 2004). Let $M = U\Sigma V^\top$ be the SVD, we have:

$$\begin{aligned}
\max_{R\in\mathcal{O}(r)} \langle R,\ M\rangle_F &= \max_{R\in\mathcal{O}(r)} \langle R,\ U\Sigma V^\top\rangle_F \\
&= \max_{R\in\mathcal{O}(r)} \mathrm{tr}\big(\Sigma U^\top RV\big).
\end{aligned}$$

Since $U^\top RV \in \mathcal{O}(r)$ and Σ is diagonal with nonnegative entries, $\mathrm{tr}(\Sigma Q) \le \mathrm{tr}(\Sigma)$ for any $Q \in \mathcal{O}(r)$, and the equality is achieved when $Q = I$. Therefore the maximizer satisfies $U^\top RV = I$, i.e., $R^* = UV^\top$. □

Theorem 2. We define:

$$M_{\text{joint}} = B_i^\top B_0 + A_iA_0^\top. \tag{23}$$

Let $M_{\text{joint}} = U\Sigma V^\top$ be its singular value decomposition. Then the optimal orthogonal transformation for the objective in Equation (6) is

$$R_i^* = UV^\top \in \mathcal{O}(r). \tag{24}$$

Proof. By Theorem 1, each squared alignment term admits an equivalent Orthogonal Procrustes form:

$$\begin{aligned}
\min_{R\in\mathcal{O}(r)} \|B_iR - B_0\|_F^2 &\iff \max_{R\in\mathcal{O}(r)} \langle R,\ B_i^\top B_0\rangle_F, \\
\min_{R\in\mathcal{O}(r)} \|R^\top A_i - A_0\|_F^2 &\iff \max_{R\in\mathcal{O}(r)} \langle R,\ A_iA_0^\top\rangle_F.
\end{aligned}$$

Therefore, Equation (6) is equivalent to a single Procrustes objective

$$\max_{R\in\mathcal{O}(r)} \big\langle R,\ B_i^\top B_0 + A_iA_0^\top\big\rangle_F = \max_{R\in\mathcal{O}(r)} \langle R,\ M_{\text{joint}}\rangle_F. \tag{25}$$

Let $M_{\text{joint}} = U\Sigma V^\top$ be the SVD. Then

$$\begin{aligned}
\max_{R\in\mathcal{O}(r)} \langle R,\ M_{\text{joint}}\rangle_F &= \max_{R\in\mathcal{O}(r)} \langle R,\ U\Sigma V^\top\rangle_F \\
&= \max_{R\in\mathcal{O}(r)} \mathrm{tr}\big(\Sigma U^\top RV\big) \\
&= \mathrm{tr}(\Sigma),
\end{aligned}$$

where the maximum is attained when $U^\top RV = I$, i.e., $R_i^* = UV^\top$. Similar results are provided in (Zhang et al., 2025a) as well. □