# SWIN: SEED WEIGHT INTEGRATION FOR LLM FINE-TUNING

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

## ABSTRACT

Fine-tuning large language models exhibits high variance across random seeds, forcing repeated runs or expensive ensembles to reach top performance. We introduce SWIN (**S**eed **W**eight **IN**tegration), a training-free, geometry-aware fusion method that integrates models fine-tuned with different seeds without increasing inference cost, motivated by the largely low-rank nature of fine-tuning updates and the emergence of complementary, task-relevant directions across seeds. SWIN first computes per-layer low-rank SVDs and forms basis-invariant projection operators to estimate a shared left/right consensus subspace. In that shared coordinate system it decouples each seed's core update into orientation and scale via polar decomposition, aggregates orientations with a Procrustes projection and averages symmetric scales, and reconstructs a single low-rank update. By preserving matrix geometry and avoiding brittle element-wise averaging, SWIN effectively merges complementary, task-relevant directions while suppressing redundant spectral tails. Across mathematical reasoning, commonsense reasoning, code generation, and RL benchmarks, SWIN consistently matches or exceeds the best individual seed and outperforms element-wise baselines (e.g., improving MetaMath by up to +4.0 points and the three-task RL average by +7.6 points) with no extra training or inference overhead.

## 1 INTRODUCTION

Fine-tuning large language models (LLMs) is computationally expensive and exhibits high variance across random seeds (Biderman et al., 2024). Even with identical data and hyperparameters, different initialization and minibatch ordering can lead to substantially different results, often requiring multiple training runs to identify the best checkpoint. While ensembles of these models reduce variance and improve performance (Izmailov et al., 2018), they proportionally increase inference costs—a critical limitation for deployment scenarios with strict latency or resource constraints.

Post-hoc weight-space merging offers an appealing alternative: combining multiple trained checkpoints into a single model that maintains the inference cost of an individual model (Wortsman et al., 2022). However, existing approaches face two fundamental challenges. First, most merging methods apply element-wise operations (averaging, masking, or magnitude-based selection) (Yadav et al., 2023; Yu et al., 2024; Ilharco et al., 2022) that treat weight matrices as flat vectors, ignoring their inherent geometric structure. This simplification can cause destructive interference between parameters, leading to suboptimal fusion (Ainsworth et al., 2022). Second, while most research focuses on multi-task merging—combining models trained on different datasets (Stoica et al., 2024; Wang et al., 2024)—the single-task setting, where models are fine-tuned on the same data with different random seeds, remains underexplored despite offering better initial alignment and unique opportunities for effective fusion.

In this paper, we investigate this single-task, multi-seed setting and uncover two key insights. First, layer-wise fine-tuning updates are highly redundant: truncating each update to its top 50% singular directions preserves over 99% of accuracy (Section 3.3), revealing substantial low-utility capacity in the spectral tail. Second, different seeds develop complementary capabilities, excelling at distinct sub-tasks within the same domain—with no single seed dominating across all categories (Section 3.4).

We propose SWIN, a training-free fusion method that addresses these challenges through structure-preserving aggregation. SWIN operates in two stages: (1) estimate a shared low-rank subspace in



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.

a basis-invariant way via a consensus objective over projection operators, admitting a closed-form eigen-decomposition; (2) fuse seed updates in shared coordinates by decoupling *orientation* and *scale*—aggregating orientations via an orthogonality-constrained Procrustes projection and averaging symmetric scale components. Unlike element-wise rules, SWIN respects the geometry of weight updates, leading to more effective knowledge consolidation.

We evaluate *SWIN* across mathematical reasoning, commonsense reasoning, code generation and RL training tasks. Our method consistently outperforms element-wise baselines such as Model Soup (Wortsman et al., 2022), TIES (Yadav et al., 2023), and DARE (Yu et al., 2024) while matching or exceeding the best individual seed. For instance, when fine-tuning LLaMA-2-7B (Touvron et al., 2023) on MetaMathQA (Yu et al., 2023), *SWIN* achieves 71.0% on GSM8K (Cobbe et al., 2021) and 22.1% on MATH (Hendrycks et al., 2021), compared to 67.2% and 20.4% for the best seed—relative improvements of 5.7% and 8.3% respectively. These gains are achieved without additional training or data, demonstrating the effectiveness of structure-preserving fusion over element-wise approaches.

The main contributions of this paper are:

- We show that leading singular directions of layer-wise updates retain most accuracy, and that different seeds learn complementary sub-skills on the same task.
- We propose SWIN, a two-stage fusion method that (i) estimates a shared subspace via a closed-form consensus objective on projection operators, and (ii) performs structure-preserving aggregation via polar decomposition and an orthogonality-constrained projection, adding no inference-time cost.
- SWIN consistently matches or exceeds the best individual seed and outperforms element-wise merging baselines across math, commonsense, and code benchmarks, improving MetaMathQA by up to 4.0 points and the three-task RL average by 7.6 points.

## 2 Preliminaries

### 2.1 Model Merge

Model merging combines knowledge from multiple trained models into a single model without additional training (Wortsman et al., 2022; Yadav et al., 2023; Yu et al., 2024; Ilharco et al., 2022). The dominant line of work addresses *multi-task merging*—combining models trained on different datasets—using interference-reduction strategies such as TIES (Yadav et al., 2023), DARE (Yu et al., 2024), Task Arithmetic (Ilharco et al., 2022), KnOTS (Stoica et al., 2024), TSV (Gargiulo et al., 2025), and Iso-CTS (Marczak et al., 2025). These methods are designed for interference reduction across heterogeneous tasks and do not fit the multi-seed regime, where models share the same basin and the dominant challenge is basis misalignment rather than task conflict (see Appendix E).

The *single-task, multi-seed* setting—combining models fine-tuned on the same data with different random seeds—has received less attention, with prior work (Model Soup (Wortsman et al., 2022), SWA (Izmailov et al., 2018), Git Re-Basin (Ainsworth et al., 2022)) focusing on computer vision. SWIN is, to our knowledge, the first geometry-aware, subspace-level fusion method for this regime in LLMs, covering both full fine-tuning and RL post-training.

### 2.2 Singular Value Decomposition (SVD)

For any matrix $W \in \mathbb{R}^{m\times n}$, the singular value decomposition (SVD) writes

$$W = U\,\Sigma\,V^{\top}, \tag{1}$$

where $U \in \mathbb{R}^{m\times m}$ and $V \in \mathbb{R}^{n\times n}$ are orthogonal, and $\Sigma = \mathrm{diag}(\sigma_1, \ldots, \sigma_r, 0, \ldots)$ has nonnegative entries with $\sigma_1 \geq \cdots \geq \sigma_r > 0$. The vectors in $U$ and $V$ form orthonormal directions, and each $\sigma_i$ shows how much $W$ scales the corresponding direction.

## 3 Empirical Observation on Single-Task Model Fusion

We motivate SWIN with four empirical observations: same-basin mode connectivity, low-rank redundancy, complementary seed specialization, and weak vector-wise basis correspondence across seeds.

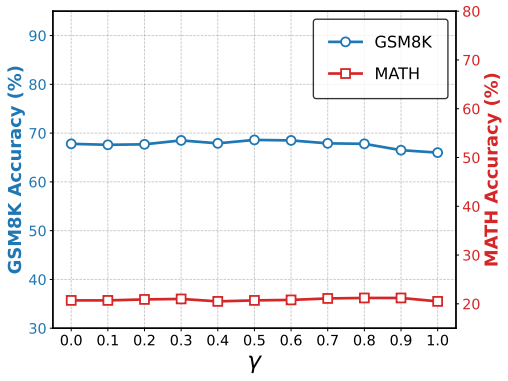



Figure 1: Basin analysis: linear interpolation between two seed-specific fine-tuning updates. Accuracy on GSM8K and MATH remains smooth across $\gamma \in [0, 1]$, with no performance barrier, confirming same-task seeds share a loss basin.

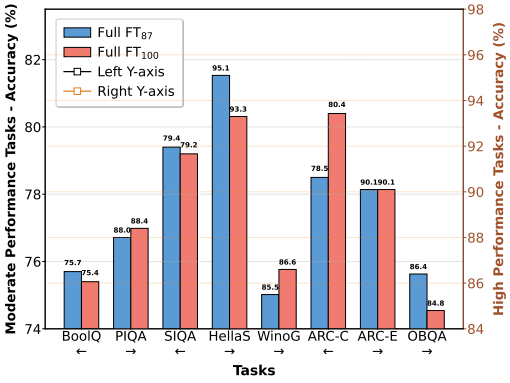



Figure 2: Performance Comparison of LLaMA3-8B Models across sub-tasks. No single seed dominates, indicating complementary specialization.

### 3.1 Same-Task Seed Models Share a Basin

We linearly interpolate between two seed-specific fine-tuning updates,

$$\Delta\theta_m(\gamma) = \gamma\Delta\theta_1 + (1-\gamma)\Delta\theta_2, \quad \gamma \in [0, 1], \tag{2}$$

and evaluate $\theta_m(\gamma) = \theta_{\text{pre}} + \Delta\theta_m(\gamma)$ on downstream tasks. Figure 1 shows that accuracy remains stable across $\gamma$, with no abrupt drop at intermediate points: **same-task seed models are largely mode-connected, providing a favorable starting point for post-hoc fusion.** The non-identical endpoints leave room for consolidation, motivating the subspace-based merging in Section 4.

### 3.2 Multi-Seed vs. Multi-Task: Different Geometric Regimes

Multi-task models trained on different objectives occupy different loss basins: linear interpolation between a Math-specialized and a Code-specialized model causes severe accuracy collapse (GSM8K: 67.2 $\rightarrow$ 4.6 at $\gamma$=0.25; full results in Appendix G). Multi-seed models, by contrast, share the same basin (Fig. 1), with pairwise task-vector cosine similarity $\sim$0.6–0.8 vs. $\sim$0.0 for multi-task pairs. This structural difference means the dominant challenge in the multi-seed regime is not *task interference* between distant solutions, but *basis misalignment* between nearby solutions that express the same shared subspace in different coordinate systems—a distinction that existing interference-reduction methods (TSV, Iso-CTS) are not designed to handle (Appendix E). SWIN targets precisely this regime via consensus subspace estimation and orientation–scale decoupling.

### 3.3 Redundancy in LLM Fine-Tuning

Related work suggests that LLM fine-tuning updates are sparse and redundant (Yu et al., 2024), and that models fine-tuned across diverse tasks converge to a shared, low-dimensional weight subspace (Kaushik et al., 2025)—the *Universal Weight Subspace Hypothesis*. Building on this, we ask a more structured question: how much of a fully fine-tuned model can be recovered if each *layer-wise* update is truncated to its leading singular directions? A positive answer would indicate low-utility capacity in the spectral tail that could be repurposed during fusion.

| **Task** | **10%** | **30%** | **50%** | **80%** | **100%** |
|---|---|---|---|---|---|
| GSM8K | 65.0 | 65.4 | 66.5 | 67.0 | 67.0 |
| MATH | 14.9 | 19.7 | 20.1 | 20.3 | 20.3 |
| **Average** | 40.0 | 42.6 | 43.3 | 43.7 | 43.7 |

Table 1: Accuracy Recovery of LLaMA2-7B under Layer-wise Low-Rank Truncation (MetaMathQA Fine-tuning).

Concretely, for each layer's update $\Delta\theta \in \mathbb{R}^{m\times n}$ of the fine-tuned model $\theta = \theta_{\text{pre}} + \Delta\theta$, we keep only the top-$r$ SVD components to form $\tilde{\theta}(r) = \theta_{\text{pre}} + \widehat{\Delta\theta}(r)$ and evaluate it under the same decoding protocol.

Table 1 shows rapid accuracy recovery at low ranks, saturating at 50%: informative content concentrates in the leading spectrum, while the residual tail contributes little and represents repurposable capacity.

**Implication for Fusion.** The low-utility spectral tail suggests repurposable capacity. In Sec. 4.2, we exploit it by estimating a shared subspace across seeds and using it to incorporate complementary directions, effectively reallocating tail capacity to more informative structure.

### 3.4 Diverse Ability among Trained Models with Different Random Seeds

Building on Sec. 3.3, we hypothesize that stochastic variability from different random seeds can supply complementary information. Even under identical data and hyperparameters, optimization noise (e.g., minibatch ordering and initialization) can steer training toward distinct yet aligned representation subspaces.

Figure 2 supports this: no single seed dominates across sub-tasks; the best-performing seed varies by sub-task with substantial gaps across categories. On LLaMA3-8B commonsense reasoning, one seed leads on *HellaSwag* while another leads on ARC-C/WinoG, suggesting seed-wise updates encode partially complementary task-relevant directions that SWIN can consolidate.

### 3.5 Row-wise Max Basis Overlap Across Seeds

Even within the same task, the leading singular directions extracted from different seed models need not align at the *basis-vector* level. To quantify this effect, we take two seed models and compute the rank-$r$ truncated SVD bases from their layer-wise update matrices $\Delta W$. Let $U_1 = [u_{1,1}, \ldots, u_{1,r}]$ and $U_2 = [u_{2,1}, \ldots, u_{2,r}]$ denote the top-$r$ left singular bases. Rather than comparing $U_1^\top U_2$ directly, we summarize the best possible correspondence for each $u_{1,i}$ by the row-wise maximum similarity

$$s_i = \max_{j\in\{1,\ldots,r\}} \left|u_{1,i}^\top u_{2,j}\right|, \qquad i = 1, \ldots, r. \tag{3}$$

We visualize $\{s_i\}_{i=1}^r$ across layers as a heatmap (Fig. 3), where the $x$-axis indexes basis vectors in $U_1$ and the color encodes their best-match similarity in $U_2$.

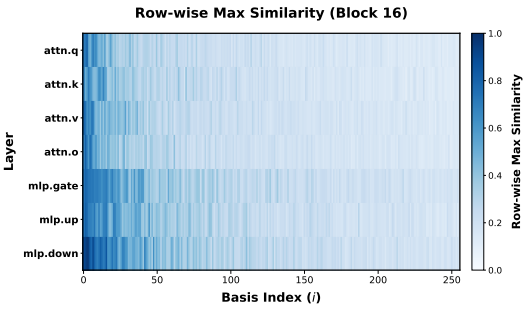



Figure 3: **Row-wise max basis overlap across seeds.** We compute the top-$r$ SVD bases $U_1, U_2$ from the layer-wise update matrices $\Delta W$ of two seed models, and visualize the row-wise maximum similarity $s_i = \max_j |u_{1,i}^\top u_{2,j}|$. The prevalence of near-zero $s_i$ indicates weak basis correspondence across seeds.

Many directions in $U_1$ only achieve small similarity even after maximizing over all vectors in $U_2$ (often 0.1–0.2), indicating substantial rotational ambiguity. This makes brittle vector-wise alignment ill-posed and motivates the basis-invariant, subspace-level approach of SWIN (Sec. 4).

## 4 Our Method

The four properties established in Sec. 3 motivate a two-stage strategy: (1) estimate a shared low-rank subspace in a basis-invariant manner, and (2) perform structure-preserving aggregation in shared coordinates, separating orientation from scale before aggregation.

### 4.1 Problem Definition

Let $\theta_{\text{pre}}$ be a pretrained model and $\theta_i = \theta_{\text{pre}} + \Delta\theta_i$ denote the fully fine-tuned model obtained with seed $s_i$ on the same dataset under identical hyperparameters, for $i = 1, \ldots, N$. We operate layer-wise on $\{\Delta\theta_i\}$ and seek a merged update

$$\Delta\theta_M = \text{Merge}(\Delta\theta_1, \ldots, \Delta\theta_N), \quad \theta_M = \theta_{\text{pre}} + \alpha\, \Delta\theta_M, \tag{4}$$

where $\alpha$ is a global scaling factor. Our goal is to merge same-task seed updates in a geometry-aware manner, operating in a shared subspace while preserving orthogonality.

### 4.2 Overview of SWIN

SWIN fuses each layer-wise weight update in two stages. **Stage 1** estimates shared left/right low-rank subspaces across seeds and expresses all updates in common coordinates. **Stage 2** performs structure-preserving aggregation in these shared coordinates and reconstructs a single low-rank update. Unlike element-wise rules, SWIN operates in a shared subspace and preserves orthogonal structure in the fused representation to avoid reintroducing redundancy.

Since basis-vector correspondence across seeds is weak (Section 3.5), SWIN relies on basis-invariant projection operators rather than matching individual SVD vectors. Concretely, for each seed update we compute a rank-$r$ truncated SVD and form its left/right projectors. We estimate shared subspaces via a closed-form consensus objective on these projectors, yielding comparable core matrices for all seeds. Finally, we fuse the cores by decoupling orientation and scale with polar decomposition: orientations are aggregated via an orthogonality-constrained projection (Procrustes), while PSD scales are averaged, and the merged core is lifted back to weight space.

### 4.3 Stage 1: Shared Subspace Estimation

A key challenge in multi-seed fusion is that seed-specific SVD bases are often not reliably aligned. As shown in Fig. 3, even after taking a row-wise maximum over all basis vectors in $U_2$, many vectors in $U_1$ only achieve small similarity (e.g., $\max_j |u_{1,i}^\top u_{2,j}| \approx 0.1$–$0.2$), indicating substantial rotational/permutation ambiguity. Therefore, instead of performing brittle vector-wise matching, we estimate a basis-invariant shared subspace at the subspace level.

For each seed update, we compute a rank-$r$ truncated SVD,

$$\Delta\theta_i \approx U_i \Sigma_i V_i^\top, \qquad U_i^\top U_i = I, \quad V_i^\top V_i = I, \tag{5}$$

where $r \ll \min(m, n)$. We set $r = 1/2$ of full rank uniformly across all layers; a sensitivity ablation over $r \in \{1/4, 1/2, 3/4, 1\}$ is provided in Appendix C (Table 12). We represent the left/right subspaces by their projection operators,

$$P_i = U_i U_i^\top, \qquad Q_i = V_i V_i^\top.$$

**Consensus subspace via an optimization principle.** Rather than concatenating updates and performing a single large SVD, we treat each $\Delta\theta_k$ as a sample from the same basin and define the shared subspace by minimizing the average squared chordal distance between projection operators. Concretely, we seek a rank-$r$ projector $P = UU^\top$ that is closest to all seed projectors:

$$P^* = \arg \min_{P:\, P^2 = P,\, \text{tr}(P) = r} \frac{1}{N} \sum_{i=1}^{N} \|P - P_i\|_F^2. \tag{6}$$

An analogous definition is used for the right subspace $\{Q_i\}$.

**Closed-form solution.** Problem (6) corresponds to computing the extrinsic Fréchet mean on the Grassmann manifold under the chordal distance: let

$$\bar{P} = \frac{1}{N} \sum_{i=1}^{N} P_i, \qquad \bar{Q} = \frac{1}{N} \sum_{i=1}^{N} Q_i, \tag{7}$$

then the consensus subspaces are given by the top-$r$ eigenspaces,

$$U^* = \mathrm{eig}_r(\bar{P}), \qquad V^* = \mathrm{eig}_r(\bar{Q}), \tag{8}$$

and $P^* = U^*U^{*\top}$, $Q^* = V^*V^{*\top}$. Intuitively, directions that are consistently present across seeds tend to accumulate larger eigenvalues in $\bar{P}$ and $\bar{Q}$, while seed-specific or noisy directions are attenuated.

With $(U^*, V^*)$ fixed, each seed update can be expressed in shared coordinates as

$$S_i = (U^*)^\top \Delta\theta_i V^* \in \mathbb{R}^{r\times r}. \tag{9}$$

These matrices $\{S_i\}$ are directly comparable across seeds, as they are expressed in the same consensus coordinate system.

### 4.4 Stage 2: Structure-Preserving Fusion in the Shared Subspace

A naive approach would average the cores $\{S_i\}$ directly. However, even within the same shared subspace, seed cores may differ by orthogonal reparameterizations; averaging in misaligned coordinates can cause cancellation and wash out informative directions. Accordingly, this design assumes that, after projection into the shared subspace, seed-specific differences are dominated by orthogonal reparameterizations rather than arbitrary linear distortions—an assumption empirically supported by the weak vector-wise SVD correspondence observed in Section 3.5. To address this, we separate *orientation* from *scale* using polar decomposition.

Each core admits the decomposition

$$S_i = O_i H_i, \qquad O_i^\top O_i = I, \quad H_i \succeq 0, \tag{10}$$

where $O_i$ captures the orthogonal orientation and $H_i$ captures the symmetric positive semidefinite scale. If $S_i = A_i D_i B_i^\top$ is the SVD, then $O_i = A_i B_i^\top$ and $H_i = B_i D_i B_i^\top$.

**Orientation aggregation (Procrustes mean).** We define the fused orientation as the orthogonal matrix that is closest to all seed orientations:

$$O^* = \arg\min_{O\in\mathcal{O}(r)} \frac{1}{N}\sum_{i=1}^{N} \|O - O_i\|_F^2. \tag{11}$$

Equivalently, this projects the Euclidean mean $\bar{O} = \frac{1}{N}\sum_{i=1}^{N} O_i$ back to the orthogonal group. Let $\bar{O} = PDQ^\top$ be the SVD; the closed-form solution is

$$O^* = PQ^\top. \tag{12}$$

The scale components are fused by the arithmetic mean,

$$H^* = \frac{1}{N}\sum_{i=1}^{N} H_i, \tag{13}$$

which preserves symmetry and positive semidefiniteness. The merged core is then $S_M = O^*H^*$, and the final update is reconstructed as

$$\Delta\theta_M = U^* S_M (V^*)^\top, \qquad \theta_M = \theta_{\text{pre}} + \alpha\,\Delta\theta_M. \tag{14}$$

$\alpha$ is selected via a lightweight grid search over $\{1.0, 1.1, 1.2, 1.3\}$ on a 500-sample held-out validation set; performance varies by less than 0.5% across this range (Appendix C, Table 13).

Overall, SWIN performs fusion entirely within the shared low-rank consensus subspace: it aligns core orientations via an orthogonality-constrained projection and averages PSD scales, thereby consolidating complementary information across seeds while avoiding destructive interference from coordinate mismatch.

## 5 Experiments

**Setup.** We fine-tune LLaMA-2-7B (Touvron et al., 2023), LLaMA-3-8B, and Gemma-2-9B on MetaMathQA (Yu et al., 2023) (math), Commonsense-170k (commonsense), and Code-Feedback (code) with three seeds (42, 87, 100) using identical hyperparameters; for RL we use DAPO on DeepSeek-R1-Distill-Qwen-1.5B. Baselines are Model Soup (Wortsman et al., 2022), TIES (Yadav et al., 2023), DARE (Yu et al., 2024), and KnOTS (Stoica et al., 2024). Full details in Appendix B.

| Model | Method | HumanEval | HumanEval+ | MBPP | MBPP+ | Average |
|---|---|---|---|---|---|---|
| LLaMA2-7B | Full $FT_{42}$ | 31.7 | 29.9 | 41.5 | 33.3 | 34.1 |
| | Full $FT_{87}$ | 34.8 | 32.3 | 40.2 | 32.8 | 35.0 |
| | Full $FT_{100}$ | 32.3 | 31.1 | 40.7 | 34.1 | 34.6 |
| | Model Soup | 31.7 | 30.5 | 41.5 | 34.1 | 34.5 |
| | TIES | 36.6 | 33.5 | 39.2 | 32.8 | 35.5 |
| | DARE | 34.1 | 32.3 | 40.5 | 33.3 | 35.1 |
| | KnOTS | 34.8 | 33.5 | 37.0 | 31.2 | 34.1 |
| | **SWIN** | **38.4** | **35.4** | **39.4** | **33.6** | **36.7** ($\uparrow$ 1.7) |
| LLaMA3-8B | Full $FT_{42}$ | 53.0 | 49.4 | 64.8 | 55.3 | 55.6 |
| | Full $FT_{87}$ | 57.3 | 52.4 | 64.6 | 56.1 | 57.6 |
| | Full $FT_{100}$ | 56.7 | 52.4 | 64.3 | 55.8 | 57.3 |
| | Model Soup | 56.7 | 53.0 | 65.1 | 56.1 | 57.7 |
| | TIES | 56.1 | 51.8 | 65.9 | 56.9 | 57.7 |
| | DARE | 57.9 | 53.0 | 65.3 | 56.3 | 58.1 |
| | KnOTS | 57.9 | 53.7 | 64.6 | 55.6 | 58.0 |
| | **SWIN** | **57.9** | **54.3** | **66.4** | **56.6** | **58.8** ($\uparrow$ 1.2) |

Table 2: Accuracy of Fine-Tuning LLaMA2-7b and LLaMA3-8B on CodeFeedback.

| Model | Task | Individual seeds | | | Merged models | | | | Gain |
|---|---|---|---|---|---|---|---|---|---|
| | | 42 | 87 | 100 | Soup | TIES | DARE | SWIN | |
| LLaMA2-7B | GSM8K | 66.9 | 67.2 | 66.6 | 59.7 | 68.1 | 68.3 | **71.2** | $\uparrow$ 4.0 |
| | MATH | 19.0 | 19.4 | 20.4 | 16.9 | 19.8 | 20.1 | **22.3** | $\uparrow$ 1.9 |
| LLaMA3-8B | GSM8K | 80.6 | 80.3 | 81.3 | 81.8 | 81.3 | 81.4 | **82.1** | $\uparrow$ 0.8 |
| | MATH | 31.8 | 31.5 | 31.6 | 32.8 | 32.0 | 32.1 | **33.5** | $\uparrow$ 1.7 |
| Gemma2-9B | GSM8K | 81.0 | 82.2 | 81.5 | 82.9 | 81.9 | 82.2 | **85.0** | $\uparrow$ 2.8 |
| | MATH | 36.4 | 36.1 | 37.3 | 39.9 | 38.1 | 38.3 | **41.3** | $\uparrow$ 4.0 |

Table 3: Accuracy on MetaMathQA. Gain is measured against the best individual seed.

### 5.1 Mathematical Reasoning

Table 3 shows SWIN outperforming all reported baselines across architectures. On LLaMA-2-7B, it achieves 71.0/22.1 on GSM8K/MATH versus 67.2/20.4 for the best individual seeds ($+3.8$/$+1.7$ points); Gemma-2-9B gains $+4.0$ MATH points. Mathematical reasoning requires precise logical steps, making interference between conflicting solutions particularly undesirable.

Model Soup improves on larger models such as Gemma-2-9B but degrades on LLaMA-2-7B (59.7 vs. 67.2 on GSM8K), consistent with destructive interference from averaging geometrically misaligned updates. TIES is more stable but achieves smaller gains, suggesting that element-wise interference resolution alone does not fully exploit complementary seed knowledge.

### 5.2 Commonsense Reasoning

Table 4 reports LLaMA3-8B results; subscripts identify training seeds. SWIN improves the eight-task average from 85.6 for the best seed to 86.6, with gains on WinoGrande (86.6$\rightarrow$87.7) and ARC-C (80.4$\rightarrow$82.2). Different seeds specialize in different reasoning types, motivating orthogonalization to preserve and balance their complementary directions during fusion. Section 5.8 examines this complementarity in detail.

| Model | Method | BoolQ | PIQA | SIQA | HellaS | WinoG | ARC-C | ARC-E | OBQA | Avg. |
|---|---|---|---|---|---|---|---|---|---|---|
| LLaMA3-8B | Full $FT_{42}$ | **75.7** | **89.1** | 79.8 | 94.5 | 86.5 | 80.1 | 91.0 | 88.2 | 85.6 |
| | Full $FT_{87}$ | **75.7** | 88.0 | 79.4 | 95.1 | 85.5 | 78.5 | 90.1 | 86.4 | 84.8 |
| | Full $FT_{100}$ | 75.4 | 88.4 | 79.2 | 93.3 | 86.6 | 80.4 | 90.1 | 84.8 | 84.8 |
| | Model Soup | 75.3 | **89.1** | 81.0 | 96.0 | 84.1 | 81.9 | 91.9 | 88.4 | 86.0 |
| | TIES | 75.3 | 89.0 | 79.8 | 95.9 | **88.0** | 81.6 | 91.8 | 86.4 | 86.0 |
| | DARE | 75.6 | 89.0 | 79.8 | 95.1 | 86.6 | 81.6 | 91.0 | **88.8** | 85.9 |
| | KnOTS | 75.6 | 88.4 | 79.8 | 95.3 | 86.5 | 80.4 | 91.0 | 86.4 | 85.4 |
| | **SWIN** | **75.7** | **89.1** | **81.3** | **96.3** | 87.7 | **82.2** | **92.1** | **88.8** | **86.6** ($\uparrow$ 1.0) |

Table 4: Accuracy of Fine-Tuning LLaMA3-8B on Commonsense-170k.

### 5.3 Code Generation

Tables 2 and 15 report full code-generation results. SWIN achieves the highest average on both architectures. On LLaMA-3-8B, its average is 58.8 versus 57.6 for the best seed and 57.7 for Model Soup. HumanEval+ improves from 52.4 to 54.3 and MBPP+ from 56.1 to 56.6 relative to the best seeds. These stricter tests probe the precise syntax and logic that structure-preserving fusion aims to retain.

On LLaMA-2-7B, average accuracy rises from 35.0 to 36.7, and HumanEval from 34.8 to 38.4 (10.3% relative), suggesting particular benefits for consolidating complementary coding strategies at smaller scale.

### 5.4 Reinforcement Learning

We perform full-parameter DAPO training of DeepSeek-R1-Distill-Qwen-1.5B with three seeds, then compare SWIN with the individual models, Model Soup, TIES, DARE, and KnOTS. In Table 5, SWIN reaches 43.3/33.3 on AIME24/AIME25, the highest scores among all compared methods. Its three-task average improves from the best seed's 42.8 to 50.5 ($+7.6$ points), supporting consolidation of complementary DAPO behaviors. MATH500 improves from 72.0 to 74.8, providing additional evidence on a larger test set; further analysis appears in Appendix J.

| Model | Method | AIME24 | AIME25 | MATH500 | Avg. |
|---|---|---|---|---|---|
| DeepSeek-R1-Distill-Qwen-1.5B | Full $FT_{42}$ | 33.3 | 20.0 | 72.0 | 41.8 |
| | Full $FT_{87}$ | 26.7 | 26.7 | 71.5 | 41.6 |
| | Full $FT_{100}$ | 26.7 | 30.0 | 71.8 | 42.8 |
| | Model Soup | 30.0 | 30.0 | 72.4 | 44.1 |
| | TIES | 26.7 | 30.0 | 72.1 | 42.9 |
| | DARE | 36.7 | 26.7 | 72.3 | 45.2 |
| | KnOTS | 36.7 | 23.3 | 72.0 | 44.0 |
| | **SWIN** | **43.3** | **33.3** | **74.8** | **50.5** ($\uparrow$ 7.6) |

Table 5: RL results on DeepSeek-R1-Distill-Qwen-1.5B. Avg. is the unweighted mean of AIME24, AIME25, and MATH500. Gain over the best individual seed is computed before rounding.

### 5.5 Training Efficiency Analysis

On LLaMA2-7B/MetaMathQA, SWIN (3 seeds $\times$ 3 epochs) and a 9-epoch single-seed run use the same aggregate training epochs. Table 6 shows that extending one run from 3 to 6 and 9 epochs reduces GSM8K accuracy from 67.2 to 65.6 and 64.4, and MATH from 19.4 to 18.2 and 16.9. SWIN instead reaches 71.0/22.1: $+3.8$/$+2.7$ relative to that 3-epoch baseline. Seed diversity therefore yields gains that longer training does not recover in this comparison (Appendix J).

| Model | Task | Epoch 3 | Epoch 6 | Epoch 9 |
|---|---|---|---|---|
| LLaMA2-7B | GSM8K | **67.2** | 65.6 | 64.4 |
| | MATH | **19.4** | 18.2 | 16.9 |

Table 6: Extended single-seed training degrades due to overfitting, while SWIN with the same total budget achieves 71.0/22.1.

### 5.6 Generalization Analysis

| Model | Dataset | Method | Train loss | Test accuracy |
|---|---|---|---|---|
| LLaMA2-7B | MetaMathQA | Adam | 0.0603 | 43.3 |
| | | **SWIN** | 0.0655 | **44.7** |

Table 7: Training loss and test accuracy on LLaMA2-7B fine-tuned on MetaMathQA.

In Table 7, SWIN achieves higher test accuracy despite a higher training loss than Adam, consistent with reduced overfitting. This comparison alone does not establish flatter minima; Appendix J provides further analysis.

Beyond full fine-tuning, SWIN applies directly to LoRA deltas ($+5.3$/$+2.9$ on GSM8K/MATH) and improves multi-task merging by $+3.3$ average over TIES. Appendix H reports these results and the larger-model experiment.

### 5.7 Ablation Studies

Table 9 varies the number of seeds from 1 to 6. Average accuracy peaks at $N = 3$ (46.6), while GSM8K alone peaks at $N = 5$ (71.9), indicating diminishing aggregate gains from additional seeds. Early seeds can supply complementary directions; later seeds increasingly overlap. Appendix C also reports layer-wise contributions (FFN Up contributes most), Stage 2, rank, and scaling ablations.

### 5.8 Combining Complementary Capabilities

No individual seed dominates all commonsense tasks (Table 4). SWIN matches or exceeds the best individual seed on seven of eight tasks and achieves the highest overall average, consistent with inheriting multiple seeds' strengths. It does not dominate every method on every task: BoolQ remains below the best seed (75.6 vs. 75.7), and WinoGrande below TIES (87.7 vs. 88.0).

## 6 Conclusion

We introduce SWIN, a training-free, geometry-aware fusion method that merges same-task seed models by estimating a shared low-rank subspace and applying orientation–scale-decoupled aggregation to avoid destructive interference. Motivated by the low-rank redundancy and complementary specialization of seed updates, SWIN consistently matches or exceeds the best individual seed and outperforms element-wise baselines across math, commonsense, code, and RL benchmarks—with no extra inference cost. While SWIN requires $N$ independent training runs, these are fully parallelizable and the best average uses $N{=}3$ (Table 9), making the overall budget comparable to a few extended single-seed runs. We hope this work encourages further exploration of geometry-aware fusion for LLM post-training.

### AI USE STATEMENT

We used OpenAI Codex to assist with language polishing and wording revisions to improve the clarity and readability of the manuscript. The authors take responsibility for the final content, references, and reported results.

### REPRODUCIBILITY STATEMENT

Section 4 describes the SWIN algorithm, including shared subspace estimation, polar decomposition, orientation and scale aggregation, and reconstruction of the merged model. Appendix B reports training hyperparameters, while Appendix J describes the protocols for the empirical analyses. Appendix C examines the effects of the number of seeds, layer selection, fusion stages, truncation rank, and scaling factor. Table 6 reports the comparison with extended single-seed training.

### ETHICS STATEMENT

SWIN studies weight-space merging of language models fine-tuned with different random seeds using existing datasets and benchmarks. Merged models may inherit or alter biases, harmful behaviors, factual inaccuracies, and privacy risks present in the underlying models. Our experiments assess task performance and merging behavior rather than safety, fairness, or privacy guarantees. Improved benchmark performance should therefore not be interpreted as evidence of safer deployment. Downstream use should include appropriate safety evaluation and safeguards and comply with the applicable model and dataset licenses and usage terms.

## REFERENCES

Samuel K Ainsworth, Jonathan Hayase, and Siddhartha Srinivasa. Git re-basin: Merging models modulo permutation symmetries. *arXiv preprint arXiv:2209.04836*, 2022.

Dan Biderman, Jacob Portes, Jose Javier Gonzalez Ortiz, Mansheej Paul, Philip Greengard, Connor Jennings, Daniel King, Sam Havens, Vitaliy Chiley, Jonathan Frankle, Cody Blakeney, and John Patrick Cunningham. LoRA learns less and forgets less. *Transactions on Machine Learning Research*, 2024. ISSN 2835-8856. URL https://openreview.net/forum?id=aloEru2qCG. Featured Certification.

Yonatan Bisk, Rowan Zellers, Jianfeng Gao, Yejin Choi, et al. Piqa: Reasoning about physical commonsense in natural language. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pp. 7432–7439, 2020.

Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde De Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, et al. Evaluating large language models trained on code. *arXiv preprint arXiv:2107.03374*, 2021.

Runxi Cheng, Feng Xiong, Yongxian Wei, Wanyun Zhu, and Chun Yuan. Whoever started the interference should end it: Guiding data-free model merging via task vectors. *arXiv preprint arXiv:2503.08099*, 2025.

Christopher Clark, Kenton Lee, Ming-Wei Chang, Tom Kwiatkowski, Michael Collins, and Kristina Toutanova. Boolq: Exploring the surprising difficulty of natural yes/no questions. *arXiv preprint arXiv:1905.10044*, 2019.

Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. Think you have solved question answering? try arc, the ai2 reasoning challenge. *arXiv preprint arXiv:1803.05457*, 2018.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, et al. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*, 2021.

Guodong Du, Junlin Lee, Jing Li, Runhua Jiang, Yifei Guo, Shuyang Yu, Hanting Liu, Sim K Goh, Ho-Kin Tang, Daojing He, et al. Parameter competition balancing for model merging. *Advances in Neural Information Processing Systems*, 37:84746–84776, 2024.

Antonio Andrea Gargiulo, Donato Crisostomi, Maria Sofia Bucarelli, Simone Scardapane, Fabrizio Silvestri, and Emanuele Rodola. Task singular vectors: Reducing task interference in model merging. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 18695–18705, 2025.

Yifei He, Siqi Zeng, Yuzheng Hu, Rui Yang, Tong Zhang, and Han Zhao. Mergebench: A benchmark for merging domain-specialized llms. In *Advances in Neural Information Processing Systems*, 2025.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. Measuring mathematical problem solving with the math dataset. *arXiv preprint arXiv:2103.03874*, 2021.

Gabriel Ilharco, Marco Tulio Ribeiro, Mitchell Wortsman, Suchin Gururangan, Ludwig Schmidt, Hannaneh Hajishirzi, and Ali Farhadi. Editing models with task arithmetic. *arXiv preprint arXiv:2212.04089*, 2022.

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

Prakhar Kaushik, Shravan Chaudhari, Ankit Vaidya, Rama Chellappa, and Alan Yuille. The universal weight subspace hypothesis. *arXiv preprint arXiv:2512.05117*, 2025.

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph E. Gonzalez, Hao Zhang, and Ion Stoica. Efficient memory management for large language model serving with pagedattention. In *Proceedings of the ACM SIGOPS 29th Symposium on Operating Systems Principles*, 2023.

Jiawei Liu, Chunqiu Steven Xia, Yuyao Wang, and Lingming Zhang. Is your code generated by chatgpt really correct? rigorous evaluation of large language models for code generation. *Advances in Neural Information Processing Systems*, 36, 2024.

Daniel Marczak, Simone Magistri, Sebastian Cygert, Bartłomiej Twardowski, Andrew D Bagdanov, and Joost van de Weijer. No task left behind: Isotropic model merging with common and task-specific subspaces. In *International Conference on Machine Learning*, 2025.

Todor Mihaylov, Peter Clark, Tushar Khot, and Ashish Sabharwal. Can a suit of armor conduct electricity? a new dataset for open book question answering. *arXiv preprint arXiv:1809.02789*, 2018.

Aniello Panariello, Daniel Marczak, Simone Magistri, Angelo Porrello, Bartłomiej Twardowski, Andrew D Bagdanov, Simone Calderara, and Joost van de Weijer. Accurate and efficient low-rank model merging in core space. *arXiv preprint arXiv:2509.17786*, 2025.

Keisuke Sakaguchi, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. Winogrande: An adversarial winograd schema challenge at scale. *Communications of the ACM*, 64(9):99–106, 2021.

Maarten Sap, Hannah Rashkin, Derek Chen, Ronan LeBras, and Yejin Choi. Socialiqa: Commonsense reasoning about social interactions. *arXiv preprint arXiv:1904.09728*, 2019.

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

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023.

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

Mitchell Wortsman, Gabriel Ilharco, Samir Ya Gadre, Rebecca Roelofs, Raphael Gontijo-Lopes, Ari S Morcos, Hongseok Namkoong, Ali Farhadi, Yair Carmon, Simon Kornblith, et al. Model soups: averaging weights of multiple fine-tuned models improves accuracy without increasing inference time. In *International conference on machine learning*, pp. 23965–23998. PMLR, 2022.

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

Zongzhen Yang, Binhang Qi, Hailong Sun, Wenrui Long, Ruobing Zhao, and Xiang Gao. CABS: Conflict-aware and balanced sparsification for enhancing model merging. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 70973–70999. PMLR, 2025. URL https://proceedings.mlr.press/v267/yang25x.html.

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. URL https://openreview.net/forum?id=fq0NaiU8Ex.

Longhui Yu, Weisen Jiang, Han Shi, Jincheng Yu, Zhengying Liu, Yu Zhang, James T Kwok, Zhenguo Li, Adrian Weller, and Weiyang Liu. Metamath: Bootstrap your own mathematical questions for large language models. *arXiv preprint arXiv:2309.12284*, 2023.

Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. Hellaswag: Can a machine really finish your sentence? *arXiv preprint arXiv:1905.07830*, 2019.

Tianyu Zheng, Ge Zhang, Tianhao Shen, Xueling Liu, Bill Yuchen Lin, Jie Fu, Wenhu Chen, and Xiang Yue. Opencodeinterpreter: Integrating code generation with execution and refinement. *arXiv preprint arXiv:2402.14658*, 2024.

## A Experimental Setup

**Tasks and Datasets.** For mathematical reasoning, we fine-tune on MetaMathQA (Yu et al., 2023) and evaluate on GSM8K (Cobbe et al., 2021) and MATH (Hendrycks et al., 2021). For commonsense reasoning, we use Commonsense-170k for training and evaluate on eight benchmarks: BoolQ (Clark et al., 2019), PIQA (Bisk et al., 2020), SIQA (Sap et al., 2019), HellaSwag (Zellers et al., 2019), WinoGrande (Sakaguchi et al., 2021), ARC-C/E (Clark et al., 2018), and OBQA (Mihaylov et al., 2018). For code generation, we train on Code-Feedback (Zheng et al., 2024) and evaluate on HumanEval (Chen et al., 2021) and MBPP from EvalPlus (Liu et al., 2024).

**Models and Training.** We use LLaMA-2-7B (Touvron et al., 2023), LLaMA-3-8B, and Gemma-2-9B as base models. For each task, we perform full-parameter fine-tuning with three different random seeds (42, 87, 100) using identical hyperparameters and schedules. Training uses BFloat16 precision with cosine learning rate scheduling and 3% linear warmup over 3 epochs.

**Baselines.** We compare against element-wise fusion methods: Model Soup (Wortsman et al., 2022) (uniform averaging), TIES (Yadav et al., 2023) (interference resolution), and DARE (Yu et al., 2024) (parameter dropping with rescaling). We also report individual seed performance to establish upper bounds.

## B Implementation Details Setting

We set the weight decay to 0 and employ a cosine learning rate scheduler with a linear warmup ratio of 0.03. For evaluation, we utilize vLLM (Kwon et al., 2023) to ensure efficient and scalable inference during testing.

| Dataset | LR | Scheduler | Warmup | Epochs | Batch |
|---|---|---|---|---|---|
| *LLaMA2-7B* | | | | | |
| MetaMathQA | 1e-5 | cosine | 300 | 3 | 128 |
| commonsense-170k | 1e-5 | cosine | 300 | 3 | 32 |
| Code-Feedback | 1e-5 | cosine | 300 | 3 | 32 |
| *LLaMA3-8B* | | | | | |
| MetaMathQA | 5e-6 | cosine | 300 | 3 | 128 |
| commonsense-170k | 5e-6 | cosine | 300 | 3 | 32 |
| Code-Feedback | 5e-6 | cosine | 300 | 3 | 32 |

Table 8: Implementation Details of the experiments on MetaMathQA, commonsense-170k and Code-Feedback.

## C Ablation Study

We ablate key design choices in SWIN, focusing on the number of seeds and the layer types that drive the gains.

**Effect of Number of Seeds.** Table 9 evaluates $N = 1$–$6$ on LLaMA2-7B. Average accuracy peaks at $N = 3$ (46.6), with diminishing returns thereafter (46.5, 46.4, and 46.1 for $N = 4, 5, 6$). The individual-task optima differ: GSM8K peaks at $N = 5$ (71.9), while MATH peaks at $N = 3$ (22.1). Thus, additional seeds may benefit one task without improving the aggregate score, consistent with increasing overlap among seed updates.

**Layer-wise Contribution.** We apply fusion to one layer type at a time. Table 10 shows that FFN layers contribute most, especially the Up projection (69.5% GSM8K, 20.1% MATH). Gate projection also helps, while attention projections (Q/K/V/O) yield smaller gains, consistent with FFNs being a main carrier of task-specific information.

**Effect of Stage 2 (Polar Decomposition + Procrustes).** To quantify the contribution of Stage 2, we replace it with vanilla averaging in the shared subspace (i.e., Stage 1 only + mean of projected

| Task | N = 1 | N = 2 | N = 3 | N = 4 | N = 5 | N = 6 |
|---|---|---|---|---|---|---|
| *LLaMA2-7B* | | | | | | |
| GSM8K | 66.6 | 70.5 | 71.0 | 71.6 | 71.9 | 71.0 |
| MATH | 20.6 | 20.5 | 22.1 | 21.4 | 20.8 | 21.1 |
| **Average** | 43.6 | 45.5 | **46.6** | 46.5 | 46.4 | 46.1 |

Table 9: Model Merging Performance with Varying Number of Models (N=1 to N=6).

| Task | Q | K | V | O | Gate | Up | Down | Full FT |
|---|---|---|---|---|---|---|---|---|
| GSM8K | 67.2 | 66.6 | 67.2 | 65.8 | 67.7 | **69.5** | 66.6 | 67.2 |
| MATH | 19.4 | 19.7 | 19.3 | 19.7 | 19.7 | **20.1** | 19.3 | 19.4 |

Table 10: Layer-wise ablation study of SWIN on LLaMA2-7B. Each column shows the result of applying SWIN to specific layer types, with Full FT as the baseline.

updates) and compare to the full SWIN pipeline. Table 11 shows that Stage 1 alone already provides a large gain over element-wise baselines (Model Soup: 59.7%), confirming the value of the shared consensus subspace. Stage 2 contributes an additional $+2.1/+1.1$ points by properly handling orientation misalignment via polar-Procrustes aggregation. Both stages are necessary for SWIN's full performance.

| Variant | GSM8K | MATH |
|---|---|---|
| Stage 1 + Vanilla Avg | 68.9 | 21.0 |
| Stage 1 + Stage 2 (SWIN) | **71.0** | **22.1** |

Table 11: Stage 2 ablation on LLaMA2-7B / MetaMathQA. Replacing Stage 2 (polar decomposition + Procrustes) with vanilla averaging degrades performance, showing both stages are necessary.

**Sensitivity to rank $r$.** We ablate the SVD truncation ratio on LLaMA2-7B / MetaMathQA. Table 12 shows that $r{=}1/2$ of full rank works best; full rank slightly degrades performance, consistent with truncation acting as a beneficial regularizer that discards noisy spectral tails. Performance is robust across all choices, with less than 1-point variation.

| Task | $r{=}1/4$ | $r{=}1/2$ | $r{=}3/4$ | $r{=}1$ |
|---|---|---|---|---|
| GSM8K | 70.1 | **71.0** | 70.6 | 70.5 |
| MATH | 21.3 | **22.1** | 21.4 | 21.3 |

Table 12: Rank sensitivity ablation (LLaMA2-7B / MetaMathQA). $r{=}1/2$ full rank is used in all main experiments; even $r{=}1/4$ substantially outperforms the best individual seed (67.2 / 20.4).

**Sensitivity to scaling factor $\alpha$.** $\alpha$ (Eq. 14) is selected by a lightweight grid search over $\{1.0, 1.1, 1.2, 1.3\}$ using a 500-sample held-out validation set. Table 13 shows that performance varies by less than 0.5% across the search range, confirming that SWIN does not require careful tuning of $\alpha$. All baselines (TIES, DARE) are tuned with comparable effort on the same validation set.

| Task | $\alpha{=}1.0$ | $\alpha{=}1.1$ | $\alpha{=}1.2$ | $\alpha{=}1.3$ |
|---|---|---|---|---|
| GSM8K | 71.0 | 70.8 | **71.3** | 71.0 |
| MATH | **22.1** | 21.8 | 22.3 | 21.8 |

Table 13: Sensitivity of $\alpha$ on LLaMA2-7B / MetaMathQA. Performance is stable across the search range ($<$0.5% variation), and $\alpha{=}1.0$ is used as the default.

## D BASELINE METHODS

**Model Soup (Wortsman et al., 2022)**: Model Soup shows that instead of picking one "best" fine-tuned checkpoint, we can average the weights of several runs (all starting from the same pretrained model) and often get higher accuracy and better robustness, with no extra inference cost.

**Task Arithmetic (Ilharco et al., 2022)**: Task vectors are created by subtracting the weights of a pre-trained model from the fine-tuned version. These vectors can be manipulated through simple arithmetic operations (e.g.negation, addition) to steer model behavior, create multi-task models, or improve performance without additional data.

**TIES (Yadav et al., 2023)**: TIES-MERGING reduces interference by addressing redundant parameter values and sign disagreements across models through a three-step process. This method consistently outperforms other merging techniques across various domains.

**DARE (Yu et al., 2024)** : DARE drops a large portion of update parameters and rescales the remaining ones, maintaining performance while reducing redundancy in fine-tuning.

**Additional model-merging context.** Model merging also encompasses task-singular-vector methods (Gargiulo et al., 2025), low-rank fusion in core space (Panariello et al., 2025), task-vector-guided approaches (Cheng et al., 2025), and benchmarks for merging domain-specialized LLMs (He et al., 2025). Other multi-task approaches include TALL-Masks (Wang et al., 2024), PCB-Merging (Du et al., 2024), and conflict-aware balanced sparsification (CABS) (Yang et al., 2025). These complement the single-task line of work represented by Model Soup (Wortsman et al., 2022), SWA (Izmailov et al., 2018), and Git Re-Basin (Ainsworth et al., 2022), which has largely been studied in computer vision.

## E MULTI-TASK SUBSPACE BASELINES ON MULTI-SEED BENCHMARKS

Multi-task subspace methods are designed for interference reduction across heterogeneous tasks and do not fit the multi-seed geometry, where the challenge is basis alignment rather than task conflict. Here we empirically verify this by applying two representative subspace baselines—TSV (Gargiulo et al., 2025) and Iso-C (Marczak et al., 2025)—directly to our multi-seed setup (LLaMA2-7B, MetaMathQA, 3 seeds).

| Method | GSM8K | MATH |
|---|---|---|
| Best Seed | 67.2 | 20.4 |
| TIES (Yadav et al., 2023) | 68.1 | 19.8 |
| TSV (Gargiulo et al., 2025) | 68.1 | 19.8 |
| Iso-C (Marczak et al., 2025) | 67.6 | 20.1 |
| **SWIN** (ours) | **71.0** ($\uparrow$ 3.8) | **22.1** ($\uparrow$ 1.7) |

Table 14: Multi-task subspace methods (TSV, Iso-C) applied to the multi-seed setting. Both underperform SWIN significantly, confirming that their design assumptions (spectral flattening, task-specific subspace decomposition) do not fit the multi-seed geometry where the challenge is basis alignment, not task interference.

Both TSV and Iso-C underperform SWIN by a large margin. The three assumptions underlying Iso-CTS all break down in the multi-seed setting: (1) *Spectral flattening*: multi-seed spectra are already similar (same objective), so flattening destroys informative magnitude structure. (2) *Common vs. task-specific decomposition*: all meaningful directions are "common" in multi-seed, making the task-specific component noise-dominated. (3) *Alignment maximization*: misalignment in multi-seed stems from basis rotation ambiguity, not different objectives. SWIN's consensus subspace and polar-Procrustes aggregation are specifically designed for case (3), explaining its advantage.

## F THE EXPERIMENTAL RESULTS ON CODE GENERATION TASK

In this section, we provide the full results of fine-tuning LLaMA2-7B and LLaMA3-8B on code generation task. In Table 15, we also provide the results of each seed model.

| Method | HumanEval | HumanEval+ | MBPP | MBPP+ | Average |
|---|---|---|---|---|---|
| *LLaMA2-7B* | | | | | |
| Full $FT_{42}$ | 31.7 | 29.9 | 41.5 | 33.3 | 34.1 |
| Full $FT_{87}$ | 34.8 | 32.3 | 40.2 | 32.8 | 35.0 |
| Full $FT_{100}$ | 32.3 | 31.1 | 40.7 | 34.1 | 34.6 |
| Model Soup | 31.7 | 30.5 | 41.5 | 34.1 | 34.5 |
| TIES | 36.6 | 33.5 | 39.2 | 32.8 | 35.5 |
| DARE | 34.1 | 32.3 | 40.5 | 33.3 | 35.1 |
| **SWIN** | 38.4 | 35.4 | 39.4 | 33.6 | 36.7 |
| *LLaMA3-8B* | | | | | |
| Full $FT_{42}$ | 53.0 | 49.4 | 64.8 | 55.3 | 55.6 |
| Full $FT_{87}$ | 57.3 | 52.4 | 64.6 | 56.1 | 57.6 |
| Full $FT_{100}$ | 56.7 | 52.4 | 64.3 | 55.8 | 57.3 |
| Model Soup | 56.7 | 53.0 | 65.1 | 56.1 | 57.7 |
| TIES | 56.1 | 51.8 | 65.9 | 56.9 | 57.7 |
| DARE | 57.9 | 53.0 | 65.3 | 56.3 | 58.1 |
| **SWIN** | 57.9 | 54.3 | 66.4 | 56.6 | 58.8 |

Table 15: Accuracy of Fine-Tuning LLaMA2-7b and LLaMA3-8B on CodeFeedback.

## G BASIN GAP: MULTI-SEED VS. MULTI-TASK INTERPOLATION

Table 16 reports the full multi-task interpolation experiment. We linearly interpolate between a Math-specialized and a Code-specialized LLaMA2-7B model and evaluate at five $\gamma$ values. The severe accuracy collapse at intermediate points (GSM8K: 67.2 $\rightarrow$ 4.6 at $\gamma$=0.25) contrasts sharply with the smooth multi-seed interpolation in Fig. 1, confirming that multi-task models occupy different basins while multi-seed models share one.

| $\gamma$ (Math, Code) | GSM8K | MATH | HumanEval(+) | MBPP(+) |
|---|---|---|---|---|
| (0, 1) Code only | 0.0 | 0.0 | 31.7 / 29.9 | 41.5 / 33.3 |
| (0.25, 0.75) | 4.6 | 0.3 | 26.8 / 22.6 | 41.5 / 33.6 |
| (0.5, 0.5) | 54.0 | 12.9 | 17.7 / 15.2 | 30.2 / 24.6 |
| (0.75, 0.25) | 59.7 | 17.3 | 8.5 / 7.9 | 18.3 / 14.3 |
| (1, 0) Math only | **67.2** | **20.4** | 3.7 / 2.4 | 4.5 / 4.0 |

Table 16: Multi-task (Math $\leftrightarrow$ Code) interpolation on LLaMA2-7B. Severe collapse at intermediate $\gamma$ confirms models reside in different basins.

## H GENERALIZATION RESULTS

**LoRA multi-seed merging.** We fine-tune LLaMA2-7B on MetaMathQA with LoRA using three seeds and apply SWIN to the LoRA delta matrices directly.

**Multi-task merging.** We fine-tune LLaMA2-7B separately on MetaMathQA, CodeFeedback, and Commonsense-170k, then merge the three task-specific models.

**Scalability to 14B.** We fine-tune LLaMA2-14B on MetaMathQA with 3 seeds. SWIN achieves consistent gains at this scale.

| Model | Task | seed=42 | seed=100 | seed=202 | SWIN |
|---|---|---|---|---|---|
| LLaMA2-7B | GSM8K | 64.0 | 63.8 | 64.1 | **69.4** (↑ 5.3) |
| | MATH | 15.3 | 15.3 | 14.9 | **18.2** (↑ 2.9) |

Table 17: SWIN on LoRA fine-tuning (LLaMA2-7B / MetaMathQA, 3 seeds). No pipeline modification needed.

| Method | GSM8K | MATH | HumanEval | MBPP | Commonsense | Avg |
|---|---|---|---|---|---|---|
| TIES | 66.3 | 18.4 | 27.6 | 36.6 | 80.3 | 45.8 |
| **SWIN** | **70.8** | **21.3** | **30.5** | **39.2** | **83.5** | **49.1** |

Table 18: Multi-task merging (LLaMA2-7B). SWIN outperforms TIES by $+3.3$ average across three domains.

# I Case Study

## I.1 SWIN Consolidates Complementary Reasoning Across Seeds

We present a representative example where all the individual seed models produce incorrect final answers under the same prompt and decoding protocol, while SWIN yields the correct decision and profit by consolidating useful intermediate steps from different seeds. For example, the answer is 125 and only SWIN can output the correct answer.

**Question and Answer**

**Question**: A merchant wants to make a choice of purchase between 2 purchase plans: jewelry worth $ 5,000 or electronic gadgets worth $ 8,000. His financial advisor speculates that the jewelry market will go up 2.5% while the electronic gadgets market will rise 1.2% within the same month. If the merchant is looking to maximize profit at the end of this month by making a choice, how much profit would this be?
**Answer**: 125

**Response: seed42**

**seed42**: If the jewelry market goes up 2.5%, the value of the jewelry will increase by 2.5/100 * $5,000 = $125. If the electronic gadgets market goes up 1.2%, the value of the gadgets will increase by 1.2/100 * $8,000 = $96. So, the total increase in value for the jewelry is $125 and for the gadgets is $96. If the merchant chooses the jewelry, the total value of the purchase will be $5,000 + $125 = $5,125. If the merchant chooses the gadgets, the total value of the purchase will be $8,000 + $96 = $8,996. To maximize profit, the merchant should choose the option with the higher value, which is $8,996. The profit would be the difference between the two values, $8,996 - $5,125 = $3,871.
#### 3871

| Method | GSM8K | MATH |
|---|---|---|
| Best Seed (14B) | 72.6 | 22.8 |
| **SWIN (14B)** | **73.9** (↑ 1.3) | **23.6** (↑ 0.8) |

Table 19: SWIN on LLaMA2-14B / MetaMathQA (3 seeds).

**Response: seed87**

**seed87**: If the jewelry market goes up 2.5%, the value of the jewelry will increase by 2.5% of $5,000, which is 0.025 * $5,000 = $125. If the electronic gadgets market goes up 1.2%, the value of the gadgets will increase by 1.2% of $8,000, which is 0.012 * $8,000 = $96. So, the total increase in value for the jewelry is $125 and for the gadgets is $96. If the merchant chooses the jewelry, the profit would be the increase in value minus the original cost, which is $125 - $5,000 = -$4,875. If the merchant chooses the gadgets, the profit would be the increase in value minus the original cost, which is $96 - $8,000 = -$7,044. Since the merchant wants to maximize profit, the choice would be the option with the larger profit, which is the jewelry with a profit of -$4,875.
#### 4875

**Response: seed100**

**seed100**: If the jewelry market goes up 2.5%, the value of the jewelry will increase by 2.5% of $5,000, which is 0.025 * $5,000 = $125. If the electronic gadgets market goes up 1.2%, the value of the gadgets will increase by 1.2% of $8,000, which is 0.012 * $8,000 = $96. So, the total increase in value for the jewelry is $125 and for the gadgets is $96. To maximize profit, the merchant should choose the option with the larger increase in value. Therefore, the profit would be $96 - $125 = -$29.
#### 29

**Response: SWIN**

**SWIN**: First, let's calculate the increase in value for each purchase plan. For the jewelry purchase plan, the increase in value is 2.5% of $5,000, which is 0.025 * $5,000 = $125. For the electronic gadgets purchase plan, the increase in value is 1.2% of $8,000, which is 0.012 * $8,000 = $96.
Next, let's calculate the total value of each purchase plan after the increase. For the jewelry purchase plan, the total value is $5,000 + $125 = $5,125. For the electronic gadgets purchase plan, the total value is $8,000 + $96 = $8,964.
Finally, let's calculate the profit by subtracting the original cost from the total value. For the jewelry purchase plan, the profit is $5,125 - $5,000 = $125. For the electronic gadgets purchase plan, the profit is $8,964 - $8,000 = $964.
Comparing the profits, we can see that the merchant would maximize their profit by choosing the jewelry purchase plan, with a profit of $125.
#### 125

# J Additional Motivation and Empirical Analysis

## J.1 From Empirical Observations to the Fusion Design

The observations in Section 3 motivate SWIN jointly. Same-task seed models are largely mode-connected, providing a favorable local region for post-hoc fusion. Their layer-wise updates admit low-rank approximations, with diminishing returns in the spectral tail. At the same time, different seeds specialize in complementary sub-skills, while their leading singular bases need not correspond vector by vector. These findings suggest using the low-utility capacity in the spectral tail to consolidate complementary directions from other seeds. Directly stitching seed updates together is brittle when their low-rank bases do not match, and element-wise averaging does not explicitly preserve the orthogonal structure of those bases. SWIN therefore estimates a shared subspace with basis-invariant projectors and then separates orientation from scale during aggregation in the common coordinates.

## J.2 Details of the Empirical Protocols

**Weight-space connectivity.** For the interpolation experiment in Section 3.1, the intermediate models are evaluated under the same decoding protocol as the two endpoint models. A path between distinct basins can cross a high-loss region and exhibit an intermediate degradation in downstream accuracy. The smooth GSM8K and MATH accuracy curves in Figure 1 provide evidence for a favorable connection between same-task seed solutions. The non-identical endpoints nevertheless leave room to consolidate complementary information.

**Low-rank recovery.** For a fully fine-tuned model $\theta = \theta_{\text{pre}} + \Delta\theta$ on dataset $\mathcal{D}$, we compute the SVD of each layer-wise update and retain its leading $r$ components to construct $\widehat{\Delta\theta}(r)$. The reconstructed model $\tilde{\theta}(r) = \theta_{\text{pre}} + \widehat{\Delta\theta}(r)$ uses the same decoding protocol as the full model. Table 1 shows rapid recovery at low ranks followed by early saturation: 50% rank closely matches the full model, and additional rank provides only marginal gains. This motivates consolidating seed updates in a shared low-rank subspace without reintroducing low-utility tail directions.

**Seed-wise sub-task evaluation.** For seeds $\{s_i\}_{i=1}^{N}$, we obtain $\theta_i = \theta_{\text{pre}} + \Delta\theta_i$ by full-parameter fine-tuning on the same dataset with identical hyperparameters and training schedules. Under a fixed decoding protocol, we evaluate a suite of sub-tasks $\mathcal{T} = \{t_1, \ldots, t_M\}$ from the same domain and compute $\mathrm{Acc}(\theta_i; t)$ for each seed and sub-task. The commonsense example in Figure 2 uses LLaMA3-8B with seeds $\{42, 87, 100\}$. The identity of the strongest seed varies across sub-tasks, motivating fusion of complementary task-relevant directions rather than selection of a single seed.

**Interpreting basis overlap.** If two seeds admitted a stable one-to-one basis correspondence, most row-wise maximum similarities in Section 3.5 would be close to one. The observed values, often around $0.1$–$0.2$, instead indicate substantial rotational or permutation ambiguity. Weak vector-level correspondence does not mean that the learned updates are unrelated: different orthonormal bases can describe related subspaces. Projection operators provide a representation invariant to within-subspace rotations, motivating the subspace-level comparison used by SWIN.

## J.3 Training Budget and Seed Complementarity

The training-budget comparison evaluates three seeds trained for three epochs each against a single seed trained for six or nine epochs on LLaMA2-7B/MetaMathQA. Table 6 reports GSM8K accuracy of 67.2, 65.6, and 64.4, and MATH accuracy of 19.4, 18.2, and 16.9, for three, six, and nine epochs respectively. SWIN reaches 71.0 on GSM8K and 22.1 on MATH with nine aggregate training epochs across three seeds. Thus, extending a single training trajectory does not recover the benefit of combining distinct seed trajectories in this comparison. The seed-count ablation in Table 9 offers a complementary view: early seed additions contribute useful directions, while later additions increasingly overlap with existing representations. This explains why additional seeds can have diminishing benefits and motivates examining both the individual task scores and their average when selecting the number of models to merge.

### J.4 Training Loss and Test Generalization

Weight-space merging is often associated with flatter solutions and improved generalization (Wortsman et al., 2022). Table 7 examines the relationship between training loss and test accuracy in an additional LLaMA2-7B/MetaMathQA comparison. The Adam baseline has lower training loss, whereas SWIN attains higher test accuracy despite a higher training loss. This is consistent with reduced overfitting and a benefit from combining seed-specific information; the comparison alone does not establish the shape of the loss landscape.

### J.5 Additional Code-Generation and RL Analysis

On LLaMA3-8B, SWIN achieves an average code-generation score of 58.8, compared with 57.6 for the best individual seed and 57.7 for Model Soup. On the stricter benchmarks, HumanEval+ improves from 52.4 to 54.3 and MBPP+ from 56.1 to 56.6 relative to the best individual seeds. These benchmarks use more rigorous test cases, making them useful for assessing whether fusion preserves parameter relationships needed for precise syntax and logic. On LLaMA2-7B, the average rises from 35.0 for the best seed to 36.7, while HumanEval increases from 34.8 to 38.4, a 10.3% relative improvement (Tables 2 and 15).

In the RL setting, the three seed models are obtained through full-parameter DAPO training of DeepSeek-R1-Distill-Qwen-1.5B before merging. Table 5 compares SWIN with the individual seeds, Model Soup, TIES, DARE, and KnOTS. SWIN attains 43.3 on AIME24 and 33.3 on AIME25, with an unweighted three-task average of 50.5 versus 42.8 for the best individual seed (a gain of 7.6 points computed before rounding). The improvements on both AIME years support consolidation of complementary behaviors across DAPO runs; the table additionally reports MATH500 in the current evaluation.