Higher-order Inter-task Interference Modeling for Flexible Model Merging

Anonymous authors
Paper under double-blind review

Abstract

Model merging combines multiple task-specific fine-tuned models into a single multi-task model without additional training. However, it faces a fundamental dilemma: merging all N models into one degrades accuracy due to inter-task interference, whereas maintaining N independent experts alongside a router incurs an N-fold increase in storage costs. We argue that this dilemma can be resolved by utilizing $K < N$ experts, where tasks with minimal interference share the same expert, and K serves as a user-defined storage budget. This flexible strategy adapts seamlessly to various application scenarios with diverse memory constraints. The core of this approach lies in estimating inter-task compatibility to group tasks with low interference. Our empirical study reveals that inter-task interference *does not accumulate consistently*: two tasks that exhibit only mild interference in pairwise merges may suffer severe degradation when a third one is introduced, whereas three tasks with strong pairwise conflicts can exhibit a stabilizing effect. To address this, we introduce ***Interference-Aware Flexible Merging* (IAFM)**, a framework that constructs a task interference graph whose edge weights aggregate information from both *pairwise* and *triple* task merging experience, thereby capturing higher-order interactions overlooked by purely pairwise analysis. We evaluate IAFM on 8, 14, and 20 vision tasks using OpenCLIP ViT-B/32, and 6 NLP tasks using Llama3-8B models, benchmarking it against the state-of-the-art model merging methods. IAFM achieves a favorable trade-off between accuracy and storage overhead.

1 Introduction

The success of large pre-trained models has led to a proliferation of task-specific fine-tuned models. A single base model can be adapted to hundreds or thousands of downstream tasks via fine-tuning, but storing and serving each model independently incurs prohibitive costs. Model merging (Ilharco et al., 2023) offers a compelling alternative: by combining the parameters of multiple fine-tuned models into a single model through simple operations in weight space, we obtain a multi-task model without additional training. This paradigm has been extensively studied, with methods ranging from task arithmetic (Ilharco et al., 2023) and sign-consensus merging (Yadav et al., 2023) to sparsified merging (Yu et al., 2024), layer-wise alignment (Bjorner et al., 2024), and representation analysis (Chen et al., 2025).

Despite these advances, model merging faces a fundamental tension: merging all N models into a single entity inevitably induces ***inter-task interference***. When weight updates for different tasks conflict in the parameter space, the merged model's performance on individual tasks degrades. Recent work has characterized this interference through various lenses, ranging from gradient analysis in multi-task learning (Yu et al., 2020) to singular vector overlap in task matrices (Gargiulo et al., 2025). However, empirical evidence reveals that inherent inter-task interference cannot be fully eliminated, making performance degradation unavoidable in purely monolithic full merging. To circumvent this limitation, router-based approaches such as Twin-Merging (Lu et al., 2024) have been proposed. These methods allocate dedicated experts to individual tasks, thereby significantly improving merging performance.

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.

Existing approaches to managing this interference occupy two extremes. On one end, *full merging* (Ilharco et al., 2023; Yadav et al., 2023; Yu et al., 2024; Gargiulo et al., 2025) consolidates all N experts into a single model, optimizing storage at the cost of accuracy. On the other end, *router-based approaches* (Lu et al., 2024) maintain dedicated parameters for each task, achieving near-per-task-expert performance at $N\times$ storage cost. Since neither extreme is universally optimal, we propose a flexible middle ground that utilizes $K < N$ experts, where tasks with minimal interference share the same expert. By partitioning N tasks into K groups of compatible tasks, our approach *flexibly* enables a user-controllable trade-off between average merging accuracy and storage overhead.

The cornerstone of our approach is accurate estimation of inter-task interference. *Pairwise interference* can be measured by merging two tasks and observing the resulting accuracy drop on each. A straightforward approach to optimal task grouping is to greedily aggregate pairwise estimates. This strategy implicitly assumes *consistent interference accumulation* as more tasks are merged: for instance, if $\text{Interf}(A, B) < \text{Interf}(A, C)$, one would expect this ordering to be preserved when introducing a new task D, i.e., $\text{Interf}(A, B, D) < \text{Interf}(A, C, D)$. However, our empirical results reveal that: *task interference does not accumulate consistently* with the number of merged tasks. Specifically, we observe that two tasks which interfere only mildly with each other can nevertheless *collapse* when a third task is added to the merge (i.e., $\text{Interf}(A, B) < \text{Interf}(A, C) \Rightarrow \text{Interf}(A, B, D) > \text{Interf}(A, C, D)$); conversely, three tasks with severe pairwise conflicts can sometimes "rescue" one another and yield an acceptable merged model (i.e., $\text{Interf}(A, B) > \text{Interf}(A, C) \Rightarrow \text{Interf}(A, B, D) < \text{Interf}(A, C, D)$). In other words, the interference among three tasks is not the aggregation (or any linear combination) of the three pairwise interference scores. Capturing multi-task higher-order interactions is therefore essential for accurately estimating global task interference.

In this paper, we propose **Interference-Aware Flexible Merging (IAFM)**, a framework that explicitly models higher-order task interference. Central to our method is the construction of a *task interference graph*, where nodes represent tasks and edges encode interference scores. Crucially, we go beyond simple pairwise analysis by introducing *higher-order (triple) merging*: for each task pair (A, B), we evaluate their joint interference in the presence of a third task C, ranging over all remaining task set $\mathcal{T} \setminus \{A, B\}$. We then aggregate both pairwise and triple interference scores into unified edge weights. The resulting adjacency matrix thus captures how task compatibility manifests *within the broader context of the full task set*, directly addressing the limitations of purely pairwise approaches identified earlier. Given this graph, we apply spectral clustering to group N tasks into K groups with minimal intra-group interference. Each group is then merged into a single expert using any base merging algorithm (e.g., TA).

We conduct extensive experiments on merging 8, 14, and 20 vision tasks based on OpenCLIP ViT-B/32. We follow the protocol of Gargiulo et al. (2025) and show that IAFM consistently recovers a large fraction of per-task expert accuracy across all three scales, significantly outperforming random grouping and similarity-based baselines at every storage budget K. Furthermore, we evaluate IAFM on 6 NLP tasks based on Llama3-8B under the KnOTS protocol (Stoica et al., 2025), demonstrating that it achieves a favorable trade-off between accuracy and storage overhead in this distinct modality.

Notably, to the best of our knowledge, we are the first to group tasks based on their interference relations. Although prior work has proposed task interference metrics such as the Singular Task Interference (STI) score (Gargiulo et al., 2025), these approaches do not adequately capture true inter-task interference. For instance, STI essentially measures the correlation/orthogonality among tasks; consequently, a non-zero STI score cannot distinguish between interfering and compatible task relationships. In contrast, our approach explicitly models interference and compatibility through Eq. (11).

2 Related Work

Full Merging Methods. Training-free model merging combines fine-tuned task vectors without additional training (Frankle et al., 2020; Entezari et al., 2022). Task arithmetic (Ilharco et al., 2023) established the basic recipe; TIES-Merging (Yadav et al., 2023),

DARE (Yu et al., 2024), TSV (Gargiulo et al., 2025), and Core-Space (Panariello et al., 2025) progressively reduce interference via sparsification, sign consensus, and SVD-based projection.

Router-Based Methods. To avoid interference entirely, router-based approaches keep all N task vectors and route at inference (Lu et al., 2024; Chen et al., 2025; Muqeeth et al., 2024), paying $N\times$ storage.

Task Interference. Multi-task learning has long studied interference via gradient conflicts (Yu et al., 2020; Standley et al., 2020); in the merging setting, the closest signal is the Singular Task Interference score (Fifty et al., 2021; Gargiulo et al., 2025). However, STI essentially measures the correlation/orthogonality among tasks, instead of task interference.

3 Method

3.1 Preliminaries

Given a pre-trained base model with parameters θ_{base}, fine-tuning on task t produces task-specific parameters θ_t. The *task vector* τ_t is defined as:

$$\tau_t = \theta_t - \theta_{\text{base}}. \tag{1}$$

For a set of task vectors $\{\tau_1, \ldots, \tau_N\}$, a *full* merged model θ_{merged} is obtained via:

$$\theta_{\text{merged}} = \theta_{\text{base}} + \lambda \cdot \mathcal{M}(\tau_1, \ldots, \tau_N), \tag{2}$$

where $\mathcal{M}$ is a merging operator (e.g., task arithmetic averaging, TIES sign consensus) and λ is a scaling factor.

3.2 Overview of Our Approach

We build upon the router-based model merging paradigm, which avoids merging all task vectors into a single one. Instead, it adopts a Mixture-of-Experts (MoE) strategy that treats each task vector as an independent expert, employing a router to select one expert per input sample. While this approach effectively eliminates inter-task interference by keeping experts completely separated, the performance gains come at the expense of storage, as all N experts must be retained in memory during inference.

Unlike prior methods that maintain all N experts, IAFM clusters the N tasks into K groups and assigns a single expert to each group. This design is motivated by our empirical observation that task compatibility varies significantly: while some tasks exhibit strong conflicts, others are highly compatible. By grouping compatible tasks and merging their task vectors into a shared expert, IAFM achieves a favorable trade-off between mitigating performance degradation and reducing storage overhead.

We now formalize our flexible merging problem. Let $\mathcal{T} = \{1, \ldots, N\}$ denote the index set of N tasks, and let $a_t(\theta) \in [0, 1]$ denote the val-set accuracy of model θ on task $t \in \mathcal{T}$. Given a partition $\mathcal{G} = \{G_1, \ldots, G_K\}$ of $\mathcal{T}$ into K disjoint groups and a base merging operator $\mathcal{M}$, the k-th expert is obtained by merging the task vectors within group G_k:

$$\theta^k_{\text{expert}} \;=\; \theta_{\text{base}} + \lambda_k \cdot \mathcal{M}(\{\tau_t : t \in G_k\}), \quad k = 1, \ldots, K. \tag{3}$$

Each task $t \in G_k$ is served by expert k, contributing per-task accuracy $a_t(\theta^k_{\text{expert}})$.

Define the average per-group accuracy under partition $\mathcal{C}$ as

$$\bar{a}(\mathcal{C}) \;\triangleq\; \frac{1}{N} \sum_{k=1}^{K} \sum_{t \in G_k} a_t\big(\theta^k_{\text{expert}}\big). \tag{4}$$

The number of groups equals the user-supplied storage budget $K \leq N$, with K=1 recovering the full-merge regime (single model for all N tasks) and K=N recovering the router-based regime (N independent experts). Equivalently, the total parameter footprint of the K experts is K/N of the N-expert baseline.

The flexible merging problem is

$$\begin{aligned} \mathcal{C}^\star &= \arg\max_{\mathcal{C}:|\mathcal{C}|=K} \bar{a}(\mathcal{C}) \\ &= \arg\max_{\mathcal{C}:|\mathcal{C}|=K} \frac{1}{N}\sum_{k=1}^{K}\sum_{t\in G_k} a_t\big(\theta^k_{\text{expert}}\big)\,. \end{aligned} \tag{5}$$

The combinatorial search over groupings of N tasks into K groups has a search space of size equal to the Stirling number $S(N,K)$ (i.e. the number of surjections $\mathcal{T}\to\{1,\dots,K\}$ up to label permutation), which already exceeds 10^8 for $(N,K)=(20,5)$. Exhaustive search is therefore infeasible at the scales we care about. A straightforward approach is to greedily aggregate pairwise estimates, decomposing the objective (5) into a sum of pairwise compatibility terms. However, this greedy strategy proves ineffective due to the *inconsistent interference accumulation* phenomenon revealed by our empirical results. For instance, even if $\text{Interf}(A,B) < \text{Interf}(A,C)$, this ordering is not guaranteed to be preserved when introducing a new task D; one may observe $\text{Interf}(A,B,D) > \text{Interf}(A,C,D)$. Conversely, a pair with higher pairwise interference can exhibit lower joint interference upon the addition of D, i.e., $\text{Interf}(A,B) > \text{Interf}(A,C)$ yet $\text{Interf}(A,B,D) < \text{Interf}(A,C,D)$. These reversals demonstrate that pairwise rankings are insufficient predictors of multi-task compatibility. As a consequence, exhaustive search is computationally infeasible; greedy and pairwise heuristics can be suboptimal. To address this challenge, IAFM aggregates pairwise and *higher-order (triple)* interference scores, which enables it to account for inconsistent accumulation.

3.3 Interference-Aware Flexible Merging (IAFM)

Our approach consists of three stages: (1) Task Interference Analysis. We measure interference between pairs of tasks through both pairwise and triple-merging experiments. (2) Graph Construction and Clustering. We build a directed task interference graph and apply spectral clustering to group tasks into K groups. (3) Grouped Merging. Task vectors within each group are merged into a single expert using any base merging algorithm.

3.3.1 Task Interference Analysis

Pairwise Interference. For any two tasks i and j, we measure their interference by merging their models and evaluating on task i's val set. Let τ_i and τ_j be the task vectors for tasks i and j. The pairwise merged model on task i's val set is:

$$\theta^{(i)}_{i\oplus j} = \theta_{\text{base}} + \mathcal{M}(\tau_i,\tau_j), \tag{6}$$

and the pairwise accuracy is $\text{Acc}(\theta^{(i)}_{i\oplus j}, \mathcal{D}^{\text{val}}_i)$.

The pairwise interference score from task j to i is defined as:

$$\text{Interf}(i \mid i,j) = 1 - \text{Acc}(\theta^{(i)}_{i\oplus j}, \mathcal{D}^{\text{val}}_i). \tag{7}$$

This captures how well task i fares when merged with task j alone. A high score means high interference between i and j.

Higher-Order Analysis via Triple Merging. A key insight of this paper is that pairwise interference alone can miss higher-order interaction patterns. Two tasks may exhibit low pairwise interference but suffer when merged together in the presence of a third task, or conversely, may show high pairwise interference but "rescue" each other in a higher-order merge.

To capture this, we introduce *triple merging analysis*. For tasks i, j, and a third task k (where $k \neq i,j$), we merge all three and evaluate on task i:

$$\theta^{(i)}_{i\oplus j\oplus k} = \theta_{\text{base}} + \mathcal{M}(\tau_i,\tau_j,\tau_k), \tag{8}$$

$$\text{Interf}(i \mid i,j,k) = 1 - \text{Acc}(\theta^{(i)}_{i\oplus j\oplus k}, \mathcal{D}^{\text{val}}_i). \tag{9}$$

The triple information enriches the pairwise score. For each source task s and target task x, we build a matrix $\mathbf{M}^{(s)}$ of size $(N-2)\times(N-2)$, where rows index the third task k and columns index the target task x:

$$\mathbf{M}^{(s)}_{k,x} = \mathrm{Acc}_{tri}(s \mid s, x, k). \tag{10}$$

The final edge weight from task s to task x is computed as:

$$\mathbf{A}_{s,x} = \alpha \cdot \underbrace{\mathbf{M}^{(s)}_{x,x}}_{\text{pairwise}} + \beta \cdot \frac{1}{N-2} \underbrace{\sum_{k\neq x} \mathbf{M}^{(s)}_{k,x}}_{\text{triple-enhanced}}, \tag{11}$$

where α and β balance the two terms. The diagonal term $\mathbf{M}^{(s)}_{x,x}$ corresponds to the 2-task merge (s, x) evaluated on s, while the off-diagonal sum aggregates information from all 3-task merges involving s, x, and a third task k.

We can further enhance our higher-order analysis by extending triple merging to quartet merging, which accounts for interference among four tasks. However, our experimental results indicate that the marginal performance gains do not justify the substantial computational overhead, as quartet merging is highly time-consuming.

3.3.2 Task Interference Graph and Clustering

Graph Construction. We interpret $\mathbf{A}$ as the weighted adjacency matrix of a directed graph $G = (V, E, \mathbf{A})$, where V is the set of N tasks, and the directed edge weight $\mathbf{A}_{s,x}$ represents the compatibility of source task s when merged with target task x. To obtain a symmetric affinity for clustering, we symmetrize $\mathbf{A}$.

Spectral Clustering. Given K, the number of experts allowed by the storage budget, we perform spectral clustering on the graph Laplacian of $\mathbf{A}$. We compute the K smallest eigenvectors of Laplacian matrix, form the spectral embedding, and apply K-means to obtain group assignments $\{C_1, \ldots, C_K\}$.

3.3.3 Grouped Model Merging

After clustering, each group $\mathcal{C}_t$ contains tasks with minimal interference. We then merge the task vectors within each group using any standard merging algorithm $\mathcal{M}$. Our framework is agnostic to the specific choice of $\mathcal{M}$. In our experiments, we incorporate several state-of-the-art merging algorithms into our framework, including TA, TIES-Merging, DARE, TSV and Core-Space Merging.

At inference time, each input is routed to the expert corresponding to its task. For multi-task inference where the task identity is unknown, a lightweight task classifier can be used, similar to router-based approaches such as Twin-Merging (Lu et al., 2024).

4 Experiments

We evaluate IAFM on both vision and natural language understanding tasks. For vision, we follow the protocol of Gargiulo *et al.* (Gargiulo et al., 2025) on three vision-task suites of increasing scale ($N = 8, 14, 20$), using OpenCLIP ViT-B/32 with full fine-tuned task vectors. For language, we adopt the KnOTS protocol (Stoica et al., 2025) with six NLI tasks ($N = 6$) based on Meta-Llama-3-8B with LoRA rank-16 adapters.

4.1 Experimental Setup

Vision Tasks. For vision, we adopt the three suites of Gargiulo *et al.* (Gargiulo et al., 2025), which build incrementally on the original Task Arithmetic benchmark (Ilharco et al., 2023): (1) *8 tasks*: Stanford Cars, DTD, EuroSAT, GTSRB, MNIST, RESISC45, SUN397, SVHN; (2) *14 tasks* = 8 + {STL10, OxfordIIITPet, Flowers102, CIFAR100, PCAM,

FER2013}; (3) *20 tasks* = 14 + {CIFAR10, Food101, FashionMNIST, RenderedSST2, EMNIST, KMNIST}.

NLP Tasks. For natural language, we follow the KnOTS protocol (Stoica et al., 2025) and evaluate on six natural language inference (NLI) tasks (N=6): SNLI, MNLI, SICK, QNLI, RTE, and SciTail.

Models and checkpoints. For vision, the base model is OpenCLIP ViT-B/32. For each task we use a *full* fine-tuned visual encoder, downloaded from the public checkpoint release of (Gargiulo et al., 2025). For NLP, the base model is `Meta-Llama-3-8B`. For each NLI task we fine-tune a LoRA adapter (Hu et al., 2022) (rank r=16) on the attention projections using the KnOTS training recipe (Stoica et al., 2025).

Metrics. For every method we report two numbers per suite. (1) *Absolute accuracy*: average top-1 accuracy across the N task test sets. (2) *Normalized accuracy*: per-task accuracy divided by the corresponding fine-tuned expert accuracy.

Baselines. We compare against the full set of training-free merging methods evaluated in (Gargiulo et al., 2025), plus two storage-aware grouping baselines. All baselines use the same per-task fine-tuned checkpoints and the same α-selection protocol: (1) *Task Arithmetic (TA)* (Ilharco et al., 2023): $\theta_{\text{merged}} = \theta_{\text{base}} + \alpha \frac{1}{N} \sum_i \tau_i$. (2) *TIES-Merging* (Yadav et al., 2023): trim–elect–sign, then disjoint mean. (3) *DARE* (Yu et al., 2024): drop-and-rescale at p=0.9, then task arithmetic. (4) *Consensus TA* (Wang et al., 2024): TALL masks + sign consensus. (5) *TSV* (Gargiulo et al., 2025): rank-1/T SVD + Procrustes orthogonalization. (6) *Core-Space* (Panariello et al., 2025): align task-specific delta subspaces to a shared SVD reference basis, then merge in the shared core space (lossless for LoRA; adapted to full FT via SVD on τ_i). (7) *Zeroshot (lower bound)*: base model for all N tasks. (8) *Single-Task Experts*: N independent fine-tuned experts.

For storage-aware comparisons ($K > 1$) we additionally include: (1) *Random Grouping*: K groups drawn uniformly at random. (2) *Similarity Grouping*: groups built by single-linkage clustering on cosine similarity of flattened task vectors τ_i. All grouping baselines share the same per-group TSV merging pipeline as IAFM; only the grouping function differs.

Implementation. The task interference matrix $\mathbf{A}$ is computed via $N(N-1)$ pairwise merges and $N(N-1)(N-2)$ triple merges; for N=20 this is ∼6.8k triple merges plus 380 pairwise merges (7,220 ordered evaluations), taking ∼1–2 GPU-hours on a single A100. The scaling $\alpha \in \{0.5, 0.75, 1.0, 1.25, 1.5\}$ is searched on the held-out validation split; spectral clustering uses the `scikit-learn` default with K supplied by the storage budget; we set α=β=1 throughout unless ablated.

4.2 Vision Tasks Merging Results

Tables 1 report absolute and normalized accuracy of every baseline on the 8-, 14-, and 20-task suites. We instantiate IAFM with two storage budgets, K=$N/2$ and K=$\lceil N/4 \rceil$, to illustrate the trade-off. We use α=β=1 for IAFM (default) and α=1.0 for all merging baselines unless a different value was empirically better. All storage-aware rows in Table 1 (Random, Similarity, and IAFM) and Table 3 share the same per-group TSV merging operator; the only axis that varies is the grouping function.

Four observations follow from Table 1. *First*, on every benchmark suite the methods cluster into two regimes: full-merge baselines (TA, TIES, DARE, Conse. TA, TSV, Core-Space) use only $1/N$ of the storage but lose 16–42 normalized points from the per-task expert accuracy (e.g. on $N = 14$, TSV reaches 84.30 while Experts sits at 100.00); router-based Twin-Merging nearly closes the gap (98.50 normalized at $N = 20$) but pays the full $N\times$ storage cost. IAFM populates the previously empty middle of this trade-off. *Second*, IAFM at K=$N/2$ beats *both* random and similarity-based grouping on every (N, K) pair reported.

Table 1: Main merging results across all three vision-task suites ($N = 8$, 14, 20). We report *absolute* (Abs.) and *normalized* (Norm.) accuracy (%).

	$N=8$		$N=14$		$N=20$	
Method	**Abs.**	**Norm.**	**Abs.**	**Norm.**	**Abs.**	**Norm.**
Experts	86.53	100.00	90.32	100.00	90.91	100.00
TA	50.47	62.89	60.79	67.28	58.09	64.71
TIES	50.98	63.52	60.52	66.46	52.68	58.03
DARE	49.76	62.22	60.88	67.37	57.65	64.06
Conse. TA	49.80	62.20	60.25	66.71	57.32	63.01
TSV	53.05	65.73	76.62	84.30	73.39	80.72
Core-Space	53.78	64.44	76.29	83.95	73.25	80.45
Twin	73.12	89.21	88.51	98.80	88.09	98.50
Storage-aware ($K=N/2$):						
Random	89.30	96.29	87.16	96.20	87.94	96.55
Similarity	89.28	96.48	86.34	95.71	88.85	97.56
IAFM	**90.89**	**97.47**	**88.97**	**98.30**	**89.55**	**98.36**
Storage-aware ($K=\lceil N/4 \rceil$):						
Random	86.87	93.85	85.17	94.04	85.39	93.73
Similarity	85.30	92.34	83.38	92.13	85.22	93.46
IAFM	**87.36**	**94.09**	**86.87**	**95.86**	**86.85**	**95.69**
Zeroshot	48.26	53.59	57.21	63.69	56.10	62.41

The normalized margins over the stronger baseline at each N are +0.99 ($N = 8$, IAFM 97.47 vs Similarity 96.48), +2.10 ($N=14$, IAFM 98.30 vs Random 96.20), and +0.80 ($N=20$, IAFM 98.36 vs Similarity 97.56); the same ordering holds in absolute accuracy on every suite. *Third*, at $K=\lceil N/4 \rceil$ the margin over the best baseline is +0.24 ($N = 8$, IAFM 94.09 vs Random 93.85), +1.82 ($N = 14$, IAFM 95.86 vs Random 94.04), and +1.96 ($N=20$, IAFM 95.69 vs Random 93.73); the gap widens on the 14-/20-task suites where grouping decisions are harder, but on the 8-task suite the margin is small because even a coarse partition leaves little room to beat. *Fourth*, the gap between IAFM and Twin-Merging narrows with N but reverses direction: on $N = 8$ IAFM *exceeds* Twin at $K=N/2$ by +8.26 normalized points (97.47 vs 89.21), while on $N = 14$ Twin leads by +0.50 (98.80 vs 98.30) and on $N = 20$ by +0.14 (98.50 vs 98.36). Storage-aware grouping is therefore most valuable at moderate N where it can match or exceed a router with substantially fewer experts.

4.3 NLP Tasks Merging Results

To verify that IAFM generalizes beyond vision, we conduct experiments on a suite of six natural language inference (NLI) tasks following the KnOTS protocol (Stoica et al., 2025). We evaluate per-task accuracy under the same merging operators as the vision experiments (TA, TSV, Core-Space). The base model is the frozen Llama-3-8B; only the LoRA adapter weights are merged. Metrics are reported as absolute test accuracy and normalized accuracy (relative to each task's single-adapter performance).

Table 2: NLP results on the 6-task NLI suite (Llama-3-8B LoRA). Abs. and norm. accuracy averaged across six datasets. All storage-aware rows use per-group TSV operator.

Method	**Abs.**	**Norm.**
Experts	92.53	100.00
Task Arithmetic	75.08	81.14
TSV	79.50	85.89
Core-Space	80.92	87.39
Storage-aware ($K=3$):		
Random	88.66	95.74
Similarity	89.08	96.19
IAFM	**90.01**	**97.18**
Storage-aware ($K=2$):		
Random	87.34	94.32
Similarity	85.50	92.35
IAFM	**89.10**	**95.11**

Table 2 reports results at two storage budgets (K=3 and K=2). At K=1 (full merge of all six adapters), task interference is severe: even the strongest single-operator baseline (Core-Space) drops by 12.61 normalized points (87.39). IAFM with K=2 groups the six tasks into two experts, recovering most of the lost accuracy (95.11 normalized) and consistently outperforming both Random (94.32, +0.79 margin) and Similarity (92.35, +2.76 margin); at K=3 IAFM reaches 97.18 normalized, beating Random (95.74, +1.44) and Similarity (96.19, +0.99). Closing the gap to the per-task expert accuracy (92.53 absolute / 100.00 normalized) the remaining distance is 2.82 normalized points at K=3 and 4.89 at K=2. The relative gains mirror those observed on the vision suites, confirming that higher-order interference patterns are equally informative for NLP task grouping.

4.4 Accuracy-Storage Trade-off

A central claim of this paper is that the storage budget K should be a user-controlled knob, not a fixed binary choice. Figure 1 plots normalized accuracy as we sweep K from 2 to $N-1$ on the 8-task suites, comparing IAFM against random and similarity grouping (both using TSV per group).

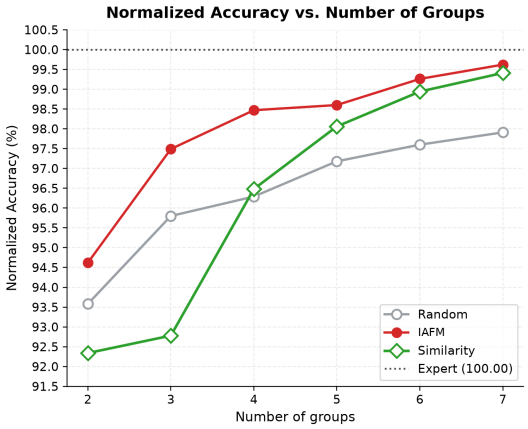


Figure 1: Flexible accuracy–storage trade-off on the 8-task suite: IAFM strictly dominates random and similarity grouping at every K.

Two trends are worth highlighting: (1) *IAFM strictly dominates random / similarity grouping at every K.* The improvement is largest in the low-storage regime ($K/N \leq 0.6$) where each grouping decision is amplified; at K=7 the gap narrows because even random groupings contain a few good groups. (2) *IAFM performs more stably than the similarity-based grouping scheme.* Increasing K from 2 to 4 yields rapid improvements for the similarity-based grouping scheme. However, as K increases further from 4 to 7, its performance improves slowly, sometimes falling below that of random grouping. In contrast, IAFM remains significantly more stable across all values of K.

4.5 Higher-Order Task Interference Modeling

The primary contribution of IAFM, extending beyond the pairwise interference baseline, is the *triple* term in Eq. (11). We isolate this contribution by re-running IAFM with β=0 (pairwise only) and comparing to α=β=1 (pairwise + triple). Both variants use the same downstream spectral-clustering pipeline.

Table 3: Effect of the triple-merging term (β) on normalized accuracy (%). Pairwise (β=0) uses only pairwise edge weights; full IAFM (β=1) uses Eq. (11).

	$N=14$		$N=20$	
Ablation	$K=7$	$K=4$	$K=10$	$K=5$
β=0	96.20	94.04	97.67	94.98
β=1 (full IAFM)	98.30	95.86	98.36	95.69
Δ (triple)	2.10	1.82	0.69	0.71

Findings. (i) Triple merging improves accuracy at *every* (N, K) configuration tested, with gains ranging from +0.69 (at $(N$=20, K=10)) to +2.10 (at $(N$=14, K=7)). (ii) The improvement is substantially larger at N=14 ($\sim$+2 points) than at N=20 ($\sim$+0.7 points), suggesting that the pairwise-only baseline leaves more headroom for higher-order correction at moderate N. Within each N, the gain is comparable across the two K values, indicating that the triple-merging benefit is robust to the storage budget. (iii) The improvement pattern is consistent with the empirical observation in §4.6: as more tasks are added, more potential rescue/harm interactions become visible, so the third-order signal becomes *more* informative even when its per-triple magnitude shrinks.

4.6 Phenomenon of Inconsistent Interference Accumulation

The ablation in §4.5 shows that adding the triple term to IAFM's edge weights improves accuracy. We now ask the deeper question: *why* does pairwise information alone fall short? The answer, which we substantiate below, is that task interference *does not accumulate consistently* as more tasks are merged—an empirical fact that pairwise analysis fundamentally cannot capture and that directly motivates the higher-order design of IAFM.

For each ordered triple (s, x, k) of distinct tasks in $\mathcal{T}$, we measure three quantities (all in absolute accuracy on task s's val set, at the per-suite optimal scaling λ=1.0): (1) *Pairwise accuracy*: $a^{(2)}_{s,x} = a_s\big(\theta_{\text{base}} + \frac{1}{2}(\tau_s + \tau_x)\big)$. (2) *Triple accuracy*: $a^{(3)}_{s,x,k} = a_s\big(\theta_{\text{base}} + \frac{1}{3}(\tau_s +$

Table 4: Distribution of the accumulation gap $\Delta_{s;x,k}$ over all 336 ordered triples on the 8-task suite.

Statistic	**Value**
Mean Δ	-0.033
Std of Δ	0.127
Min Δ	-0.360
Max Δ	$+0.251$
Collapse ($\Delta < -0.1$)	17.3%
Rescue ($\Delta > +0.1$)	8.6%
Corr($\hat{a}^{(2)}, a^{(3)}$)	0.659

Table 5: Concrete counter-examples of pairwise reasoning. *Case A*: pairwise predicts compatible, triple collapses. *Case B*: pairwise predicts interference, triple rescues. Numbers are absolute accuracy points on task s.

Case	s	x	k	$\hat{a}^{(2)}_{s;x,k}$	$a^{(3)}_{s,x,k}$	Δ
Case A (collapse)						
A1	EuroSAT	DTD	SUN397	0.500	0.235	-0.265
A2	MNIST	GTSRB	SUN397	0.772	0.515	-0.257
Case B (rescue)						
B1	SUN397	Cars	RESISC45	0.470	0.550	$+0.080$
B2	SUN397	Cars	DTD	0.447	0.520	$+0.073$

$\tau_x + \tau_k)\big)$. (3) *Pairwise prediction* (the natural baseline that uses only pairwise information): $\hat{a}^{(2)}_{s;x,k} = \frac{1}{2}\big(a^{(2)}_{s,x} + a^{(2)}_{s,k}\big)$, i.e. the average of the two relevant pairwise merges.

We define the *accumulation gap* as

$$\Delta_{s;x,k} \triangleq a^{(3)}_{s,x,k} - \hat{a}^{(2)}_{s;x,k}. \tag{12}$$

If interference composed linearly, $\Delta_{s;x,k}$ would be near zero for every triple. The experiments below show that this is decisively *not* the case.

Table 4 summarises the distribution of Δ over all $N(N-1)(N-2)$=336 ordered triples in the 8-task suite. Three observations follow from these statistics. (i) The correlation between the pairwise prediction and the triple outcome is only 0.659—far below 1, confirming that pairwise information is a poor predictor of higher-order merging accuracy. (ii) Both directions occur: pairwise *over*-estimates (collapse, $\Delta \ll 0$) and pairwise *under*-estimates (rescue, $\Delta \gg 0$). This rules out any explanation of the gap as a constant bias. (iii) The magnitudes are substantial: the Table 4 distribution extremes show a worst collapse of 36 percentage points and a strongest rescue of 25 percentage points relative to the pairwise baseline.

Concrete counter-examples. Aggregate statistics can be hand-waved away; concrete triples cannot. Table 5 lists four representative ordered triples from the 14-task suite that exhibit the two qualitative failure modes of pairwise reasoning. All numbers are absolute accuracy on the source task s. Each row of Table 5 is a counter-example to the implicit assumption underlying any pairwise similarity grouping baseline: that two tasks exhibiting low pairwise interference will maintain minimal interference when a third task is merged. Case A shows this assumption fails when a third "interfering" task is present; Case B shows that pairwise harm does *not* imply consistent harm—a third task can sometimes rescue the merge. Both failure modes invalidate pairwise reasoning as a guide to grouping. They also validate our design choice: IAFM constructs edge weights by aggregating *triple* merging outcomes, so the resulting interference graph explicitly encodes how each candidate edge behaves *in the context of the full task set* rather than in isolation. The downstream gain reported in §4.5 is a direct consequence.

5 Conclusion

We presented IAFM, a flexible framework for model merging that exploits task interference patterns to determine grouping strategies. By combining pairwise and higher-order merging analyses, our method constructs a rich task interference graph that captures complex interaction patterns. Our approach represents a principled middle ground between full merging and router-based merging, offering a continuous performance–storage trade-off. The framework is agnostic to the underlying merging algorithm and can be integrated with any existing method. To our knowledge, this is the first study to comprehensively model task interference.

6 AI DISCLOSURE

In this work, we used generative AI tools for "design or provide feedback on research experiments, implement methods". We have not used generative AI tools for "Generate synthetic data sets, help develop theoretical models or conceptual frameworks, formulate mathematical claims, provide critical ingredients for proving mathematical claims, assist in the writing of proofs, propose or refine hypotheses, assist with translation, clean and reformat dataset, support qualitative and thematic data analysis, interpret results". Additionally, we used generative AI tools for "suggest experimental parameters, create or edit software code, draft parts of a research paper, edit a research paper to improve readability". We have reviewed all AI-assisted work. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

REFERENCES

J. Bjorner, G. Stoica, P. Ramesh, D. Bolya, T. Hearn, and J. Hoffman. ZipIt! merging models from different tasks without training. In *International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=LEYUkvdUhq.

Z. Chen, Z. Zhou, B. Zhang, W. Zhang, X. Sun, and J. Yan. SE-Merging: A self-enhanced approach for dynamic model merging. In *2025 International Joint Conference on Neural Networks*, pages 1–8, 2025. doi: 10.1109/IJCNN64981.2025.11229001. URL https://doi.org/10.1109/IJCNN64981.2025.11229001.

R. Entezari, H. Sedghi, O. Saukh, and B. Neyshabur. The role of permutation invariance in linear mode connectivity of neural networks. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=dNigytemkL.

C. Fifty, E. Amid, Z. Zhao, T. Yu, R. Anil, and C. Finn. Efficiently identifying task groupings for multi-task learning. In *Advances in Neural Information Processing Systems*, volume 34, pages 27503–27516, 2021. URL https://proceedings.neurips.cc/paper/2021/hash/e77910ebb93b511588557806310f78f1-Abstract.html.

J. Frankle, G. K. Dziugaite, D. Roy, and M. Carbin. Linear mode connectivity and the lottery ticket hypothesis. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 3259–3269. PMLR, 2020. URL https://proceedings.mlr.press/v119/frankle20a.html.

A. A. Gargiulo, D. Crisostomi, M. S. Bucarelli, S. Scardapane, F. Silvestri, and E. Rodolà. Task singular vectors: Reducing task interference in model merging. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18695–18705, 2025. URL https://openaccess.thecvf.com/content/CVPR2025/html/Gargiulo_Task_Singular_Vectors_Reducing_Task_Interference_in_Model_Merging_CVPR_2025_paper.html.

E. J. Hu, Y. Shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, and W. Chen. Lora: Low-rank adaptation of large language models. In *The Tenth International Conference on Learning Representations (ICLR)*, 2022.

G. Ilharco, M. T. Ribeiro, M. Wortsman, S. Gururangan, L. Schmidt, H. Hajishirzi, and A. Farhadi. Editing models with task arithmetic. In *The Eleventh International Conference on Learning Representations (ICLR)*, 2023.

Z. Lu, C. Fan, W. Wei, X. Qu, D. Chen, and Y. Cheng. Twin-Merging: Dynamic integration of modular expertise in model merging. 37, 2024. URL https://proceedings.neurips.cc/paper_files/paper/2024/hash/8fcd17eb91bae20d9826786d7d6be799-Abstract-Conference.html.

M. Muqeeth, H. Liu, and C. Raffel. Soft merging of experts with adaptive routing. 2024. URL https://openreview.net/forum?id=7I1991c54z.

A. Panariello, D. Marczak, S. Magistri, A. Porrello, B. Twardowski, A. D. Bagdanov, S. Calderara, and J. van de Weijer. Accurate and efficient low-rank model merging in core space. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025.

T. Standley, A. Zamir, D. Chen, L. Guibas, J. Malik, and S. Savarese. Which tasks should be learned together in multi-task learning? In *International Conference on Machine Learning (ICML)*, 2020.

G. Stoica, P. Ramesh, B. Ecsedi, L. Choshen, and J. Hoffman. Model merging with SVD to tie the KnOTS. In *International Conference on Learning Representations*, 2025. URL `https://openreview.net/forum?id=67X93aZHII`.

K. Wang, N. Dimitriadis, G. Ortiz-Jimenez, F. Fleuret, and P. Frossard. Localizing task information for improved model merging and compression. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pages 50268–50287. PMLR, 2024. URL `https://proceedings.mlr.press/v235/wang24k.html`.

P. Yadav, D. Tam, L. Choshen, C. Raffel, and M. Bansal. Ties-merging: Resolving interference when merging models. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 36, 2023.

L. Yu, B. Yu, H. Yu, F. Huang, and Y. Li. Language models are super mario: Absorbing abilities from homologous models as a free lunch. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pages 57755–57775. PMLR, 2024. URL `https://proceedings.mlr.press/v235/yu24p.html`.

T. Yu, S. Kumar, A. Gupta, S. Levine, K. Hausman, and C. Finn. Gradient surgery for multi-task learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.