Sign Conflicts in LoRA Adapter Merging Are Pervasive, Not Localized

Anonymous authors
Paper under double-blind review

Abstract

LoRA adapters are small additive weight updates that specialize a large language model to a new task without retraining it. Merging several independently trained adapters into a single model is appealing, but it confronts a basic obstacle: at many weights the adapters disagree about whether the value should move up or down, a disagreement known as a *sign conflict*. The standard remedy, TIES-Merging, resolves these conflicts through a single global vote over all weights. Whether such global treatment is necessary is an open question with practical stakes. If the conflicts were concentrated in a few identifiable regions of the network, they could be resolved only there, far more cheaply, and several recent merging methods quietly assume some such concentration. That assumption has never been tested directly. This work presents a systematic empirical analysis of the optimization and structural behaviors that govern parameter-efficient fine-tuning, providing new insights into how sign conflicts and destructive interference manifest across transformer architectures. This work tests it on five language models (Qwen2.5-3B, Llama-3.2-3B, Phi-3.5-mini, Mistral-7B, OLMo-2-1B), each adapted to three sentiment domains with five random seeds, under two measurements: the usual count of how many weights disagree, and a magnitude-weighted measure of how much update the disagreements actually erase in a merge. Both measurements reject the assumption. By count, conflict is pervasive and nearly uniform: roughly 73% of weights disagree, close to the 75% that purely random signs would produce, and two adapters trained on the same data under different seeds disagree as often as adapters trained on different tasks. The conflict therefore originates in the randomness of training rather than in the tasks diverging, and it exposes no pattern a method could exploit. Weighting by magnitude does reveal a genuine pattern: most of the erased update lies in the MLP input projections, which are also the largest adapted tensors, in all five models. Yet even this pattern is too diffuse to act on, since capturing 80% of the damage still requires touching roughly 72% of the weights. The implication for efficient merging is cautionary: the damage is real but thinly spread, so resolving conflicts only in selected regions cannot replace the global vote.

1 Introduction

Low-Rank Adaptation (LoRA) provides an inexpensive route to specializing a large language model to a task, training a small additive weight update, the *task vector*, while leaving the base model frozen (Hu et al., 2022). A natural next step is to compose several such adapters into a single multi-skill model without retraining, by merging their task vectors (Ilharco et al., 2023; Wortsman et al., 2022). Merging, however, degrades quality: independently trained task vectors interfere, and a plain average weakens each skill. The dominant explanation for this damage is *sign conflict*: at some weights the adapters being merged disagree about whether the value should move up or down. The most widely used remedy is TIES-Merging (Yadav et al., 2023), which proceeds in three steps. It first *trims* each task vector, keeping only the largest-magnitude entries and zeroing the rest to suppress low-magnitude noise. It then *elects* a single sign at every weight by summing the signed magnitudes across all adapters and taking the sign of that sum. Finally it averages only the entries whose sign agrees with the elected one, discarding the dissenters. Because the election pools every adapter at once, it is a *global* operation over the entire weight space rather than a local correction, and

this global character is exactly what the present work interrogates. DARE (Yu et al., 2024) likewise acts at the level of individual weights, dropping a large fraction of entries at random and rescaling the remainder.

The localization hypothesis. A global fix is necessary only if conflict is everywhere. This invites a tempting hypothesis: that sign conflicts are *structurally localized*, that is, the conflict concentrates in a small, identifiable set of network regions (combinations of module type and layer depth, termed *cells* here), the same regions across seeds and architectures. Were this true, the global vote would be wasteful: one could resolve conflicts only within the few high-conflict cells, leave the remainder untouched, and recover most of the benefit of TIES while modifying far fewer weights.

Formally, let $\mathcal{C}$ be the set of (module-type $\times$ layer-depth) cells and, for a coverage target $\beta \in (0, 1]$, let $\pi(\beta)$ denote the smallest fraction of parameters whose cells jointly carry a β share of the total destructive cancellation mass (Sec. 3 defines the mass and the budget function π). The strong localization hypothesis asserts a small, seed- and architecture-stable subset $S \subset \mathcal{C}$ for which

$$\pi(\beta) \ll \beta \qquad \text{at} \qquad \beta \approx 0.8, \tag{1}$$

i.e. most of the destructive mass resides in a few cells; its negation $\pi(\beta) \approx \beta$ means the mass is spread in proportion to size, so no cell-level shortcut improves on touching the whole model. The pass/fail bar is fixed in advance: the strong form is *supported* only if a localized budget is genuinely small (concretely $\pi(0.8) \leq 0.4$, most of the destructive mass held by at most 40% of parameters) and *rejected* otherwise. The localization hypothesis is thus more than a description: in its strong form it promises a cheaper merger. This work tests it directly and finds that it fails in precisely that strong form: the measured $\pi(0.8) \approx 0.72$ (Sec. 5.3) sits far above the 0.4 bar, so the hypothesis is rejected.

What this work does. Sign-conflict structure is measured at scale (five instruction-tuned base models, three sentiment domains, five seeds each) and analyzed in two ways: first with the standard *conflict fraction* (how many weights disagree in sign), then with a *magnitude-weighted* measure introduced here, the destructive cancellation mass, which weights each conflict by how much update it erases from a merge. The fraction asks *whether* weights conflict; the cancellation mass asks *how much it costs*, which is what a merge actually pays. Figure 1 summarizes the full pipeline, from adapters to the localization verdict.

Contributions.

- A large, multi-architecture characterization of sign conflicts in LoRA merging (5 base models $\times$ 3 domains $\times$ 5 seeds), showing the raw conflict fraction is pervasive and uniform, ruling out the simple form of localization.
- A *magnitude-weighted* conflict measure (destructive cancellation mass) that reveals a genuine pattern shared by all five architectures: this mass concentrates in MLP `gate`/`up` projections.
- A negative efficiency result with a concrete parameter-budget curve, showing this concentration is too diffuse for localized resolution to match global sign election, a design caution for efficient merging methods.

Findings. These contributions yield three findings, each established across all five base models:

1. **Conflict is pervasive, not localized (by fraction).** The per-cell sign-conflict fraction is ≈0.73 with near-zero spread across modules and depths in every architecture (Sec. 5.1).
2. **Magnitude reveals real, architecture-invariant structure.** Weighting by destructive cancellation mass, it concentrates in MLP input projections (`gate`/`up_proj`, also the largest adapted tensors); the top cancellation-mass cells are 100% MLP `gate`/`up` in every arm, with cross-architecture top-k Jaccard 0.71, though the size-normalized signal is weaker (Sec. 5.2).

3. **The concentration is not actionable.** Capturing 80% of the total cancellation mass requires ≈72% of parameters (Gini 0.39 to 0.55), a saving of only ≈4 to 12 points over uniform that shrinks as coverage grows, so cell-localized resolution cannot replace global TIES (Sec. 5.3).

How each finding survives a measurement designed to overturn it, an independence reference for the fraction and a mass-optimal budget for the concentration, is the substance of the sections that follow.

2 Related Work

Model merging and task arithmetic. Combining independently trained models by operating directly on their weights has become a practical alternative to multi-task training (Yang et al., 2026). Wortsman et al. (2022) average the weights of several fine-tunes of a shared initialization to improve accuracy at no inference cost, and Ilharco et al. (2023) show that the difference between a fine-tuned and a base model, the *task vector*, can be added or subtracted to compose or remove skills. Subsequent methods refine *how* task vectors are combined rather than *whether*: Matena & Raffel (2022) weight each parameter by its estimated Fisher information, Jin et al. (2023) solve a closed-form least-squares objective (RegMean) from the models' inner-product statistics, and Yang et al. (2024) learn per-layer merging coefficients without labels. Across this line, a single difficulty recurs: combining models directly in weight space is effective but loses quality when task vectors interfere. This work studies where that interference resides when the task vectors are LoRA adapters.

Sign-based and pruned merging (TIES, DARE). Yadav et al. (2023) attribute much of the damage to sign disagreement and resolve it (TIES-Merging) by dropping small entries, electing a majority sign per weight, and averaging only the agreeing values, a *global* operation over all weights. Yu et al. (2024) (DARE) drop a large fraction of task-vector entries at random and rescale the rest, exploiting the redundancy in the update; later weight-level variants refine *which* entries to keep, with DELLA (Deep et al., 2024) sampling by magnitude rank and Model Breadcrumbs (Davari & Belilovsky, 2024) masking both low- and high-magnitude outliers. Both TIES and this magnitude-pruning family motivate the localization hypothesis tested here: if sign conflict were concentrated in a few regions, the global drop-and-vote step would be unnecessary work. This work proposes no new merger; it directly tests, across five architectures, the structural assumption such a localized method would require, and finds that it does not hold at the cell level. Relative to DARE in particular, the aim is not a better merger but an explanation of *why* a merge budget is well spent on individual weights rather than on a few structural cells, supplying the cross-architecture evidence DARE's design quietly relies on but does not establish (Sec. 6).

Loss-landscape geometry and interference. How severely merging degrades a model is closely tied to the geometry of the loss surface. Frankle et al. (2020) study linear mode connectivity, whether two trained solutions can be joined by a straight low-loss path, and Ainsworth et al. (2023) show that accounting for the many equivalent orderings of neurons (Git Re-Basin) can place independently trained networks in the same basin, lowering the barrier to merging. Ortiz-Jiménez et al. (2023) trace task-arithmetic interference to entanglement in the weight space and show that editing in the tangent space disentangles tasks, a geometric counterpart to the per-weight conflict studied here. A related line reduces interference in a task-singular-vector basis rather than per weight (Gargiulo et al., 2025); the cell-level view taken here is complementary, asking where conflict concentrates structurally rather than in which singular directions. Sign conflict is itself a per-weight view of the same phenomenon: directions along which two task vectors diverge. This work measures it per structural cell and asks whether the resulting damage is concentrated.

LoRA and parameter-efficient adaptation. LoRA adapts a frozen model with a low-rank additive update $\Delta W = (\alpha/r)BA$ (Hu et al., 2022), which renders the per-task update both cheap to store and a natural unit to merge. Earlier analyses of LoRA structure focus on a single model; this work instead asks whether the *structure of conflict* between LoRA task vectors, by module type and depth, is shared across five architectures, and whether it is concentrated enough to act on.

Localizing what gets merged. A parallel line localizes merging at the level of *which weights carry a task*. Wang et al. (2024) show that task information largely survives merging because tasks occupy non-overlapping weights, and that per-task binary masks recover almost all single-task accuracy from a merged model; He et al. (2025) localize roughly 1% of parameters per task and stitch them back, matching fine-tuned performance while cutting interference. For LoRA specifically, Stoica et al. (2025) trace poor adapter merging to low subspace alignment and realign adapters into a shared SVD basis before merging. These results establish that *task support* is localizable; the question here is different and complementary. Task support is which weights a task uses; sign conflict is where independently trained adapters *destructively cancel*. A weight can matter to a task yet carry little cancellation mass, so a localized task support does not imply a localized *conflict*. This work tests the latter, at the structural-cell level, and finds it does not localize.

Open problems and the question this work takes up. Three problems remain unresolved across this literature. First, the methods above all treat interference as a quantity to suppress (by trimming, sign election, random dropping, or reweighting) yet none establishes *where* in a network the interference actually lives, or whether that location is stable enough to exploit. Second, although prior work shows that the weights a task *uses* can be localized (Wang et al., 2024; He et al., 2025), the distinct question of whether the destructive *sign conflict* between adapters localizes the same way, the redundancy that efficiency-minded methods such as DARE quietly rely on, has not been tested directly. Third, almost all of this evidence comes from a single model family, and although recent work begins to study merging across scales (Yadav et al., 2025), whether any structural claim transfers across architectures remains open. This work takes up the second problem and grounds it in the first and third: it asks whether sign conflict is *structurally localized*, and it does so across five architectures so the answer is not an artifact of one model. We focus on this question because it is the one with an immediate design payoff: a positive answer would license a strictly cheaper merger, while a negative answer tells method designers, with cross-architecture evidence, that the global vote is not wasted effort. The remainder of the work makes the two measures precise and reports which way the evidence falls.

3 Mathematical Foundation

This section defines the formal machinery the rest of the work uses. The experimental apparatus that produces the adapters (base models, domains, seeds, training, and the merge baselines) is described in Sec. 4. Figure 1 previews the full pipeline these definitions formalize. Independently trained LoRA adapters yield task vectors, which are partitioned into (module-type $\times$ depth) cells; sign conflict is then read in two ways, through the standard *fraction* (whether parameters disagree in sign) and through the *cancellation mass* introduced here (how much magnitude a merge destroys). As the results will show, the fraction is pervasive and uniform, whereas the mass concentrates in the MLP input projections; three localization tests, the Gini coefficient, the parameter budget $\pi(\beta)$, and cross-architecture Jaccard overlap, then decide whether that concentration is sharp enough to act on.

Task vectors. For a module with LoRA factors A, B and scaling α/r, the materialized task vector is $\Delta W = (\alpha/r)\, BA$: one ΔW tensor per (module, domain, seed). These tensors are the raw object of study; all structure is read from them.

Granularity and model partitioning. The analysis is defined over four nested granularity levels, and the partition respects module and depth boundaries: a unit never spans two module types or two depth bins:

- **L0 (parameter).** A single scalar entry $d_{i,j}$ of a task vector (domain i, coordinate j); the finest level at which a sign exists.
- **L1 (cell).** All coordinates of ΔW for a fixed (module-type, layer-depth) pair. The *cell* is the atomic unit of every structural statement in the work.
- **L2 (module role).** A canonical grouping of module types that aligns architectures with different naming and fusion: {`q,k,v,qkv`}→ATTN_in, `o`→ATTN_out, {`gate,up,gate_up`}→MLP_in,

Figure 1: **The work's argument in four steps, with a worked example at each stage:** from independently trained LoRA adapters compared weight by weight (1), to what a *sign conflict* is (2), to the two measures and what each one reveals (3), to why a localized shortcut cannot replace the global sign vote (4). Every quantity shown is defined formally in this section.

`down`→MLP_out, with layer depth normalized to relative bins so that, e.g., the 16-layer OLMo arm aligns with the 36-layer Qwen arm.

- **L3 (architecture, arm).** One base model; the level at which invariance is claimed.

Fix a cell and let $d_1, \ldots, d_N$ be its $N = 3$ domain task vectors (one ΔW per domain at a shared seed), with elementwise values $d_{i,j}$.

Sign-conflict definition. A weight j *conflicts* when the domain task vectors disagree in sign there. Formally, with exact zeros left out of the count,

$$\text{conflict}(j) \iff \Big(\min_i d_{i,j} < 0\Big) \wedge \Big(\max_i d_{i,j} > 0\Big), \tag{2}$$

i.e. at least one domain is strictly positive and at least one strictly negative at j. This is precisely the disagreement that TIES sign election targets. The cell's sign-conflict fraction counts how often the condition holds,

$$\text{frac} = \frac{1}{P}\sum_{j=1}^{P} \mathbf{1}[\text{conflict}(j)], \tag{3}$$

where P is the number of coordinates in the cell.

Magnitude-weighted cancellation. Equation (3) weights a $\pm 10^{-4}$ flip identically to a large one, yet only magnitude drives destructive interference in a sum/mean merge. Per coordinate, define

$$\text{abs_sum}_j = \textstyle\sum_i |d_{i,j}| \qquad \text{(total magnitude present)}, \tag{4}$$

$$\text{net}_j = \big|\textstyle\sum_i d_{i,j}\big| \qquad \text{(magnitude surviving a plain merge)}, \tag{5}$$

$$\text{cancel}_j = \text{abs_sum}_j - \text{net}_j \geq 0 \qquad \text{(magnitude destroyed by interference)}. \tag{6}$$

Note $\text{cancel}_j > 0$ exactly when signs disagree at j, so it is the magnitude-aware counterpart of Eq. (2), and precisely the quantity the TIES sign vote aims to recover.

Aggregation across granularity levels. Weight-level quantities roll up to module level by two rules chosen to match what each quantity represents: a fraction (a per-weight rate) is averaged, a mass (a running total) is summed. Within a cell c (coordinate set J_c, $P_c = |J_c|$),

$$\underbrace{\text{frac}_c = \frac{1}{P_c}\sum_{j\in J_c} \mathbf{1}[\text{conflict}(j)],}_{\text{L0}\to\text{L1, mean}} \qquad \underbrace{m_c = \sum_{j\in J_c} \text{cancel}_j,}_{\text{L0}\to\text{L1, sum}} \tag{7}$$

and both are averaged over the five seeds. Cells then roll up to a module role ρ (L1→L2) by summing the mass, $M_\rho = \sum_{c:\,\text{role}(c)=\rho} m_c$, while the size-independent ratio $\text{cancel_fraction} = \sum_j \text{cancel}_j / \sum_j \text{abs_sum}_j$ is used wherever concentration must be separated from tensor size. This weight → cell → module ladder is what allows a conflict at a single coordinate to be reported as a module-level concentration.

Localization metrics. Localization is quantified three ways. (i) **Concentration (Gini).** The Gini coefficient is a standard measure of statistical dispersion (originally introduced in economics to summarize how unequally income is distributed across a population), here applied to ask how unequally the cancellation mass is spread across cells. For cell masses $m_1, \ldots, m_K$ it is the mean absolute pairwise difference normalized by twice the mean,

$$G = \frac{\sum_{a=1}^{K}\sum_{b=1}^{K} |m_a - m_b|}{2K\sum_{a=1}^{K} m_a} \in [0, 1), \tag{8}$$

with $G = 0$ for a perfectly uniform spread and $G \to 1$ when a single cell holds all the mass; the observed 0.39 to 0.55 is therefore moderate, not extreme. The share of mass held by the top-10% of cells is reported

Table 1: The five base models ("arms"). `qkv_proj`/`gate_up_proj` indicate Phi-3.5's fused projections. "Cells" = (module-type × layer-depth) groupings of LoRA-materialized ΔW (Sec. 3, "Granularity and model partitioning").

Arm	HuggingFace repo	Layers	Modules	Cells
qwen (ref)	`Qwen/Qwen2.5-3B-Instruct`	36	7	252
llama	`meta-llama/Llama-3.2-3B-Instruct`	28	7	196
phi	`microsoft/Phi-3.5-mini-instruct`	32	4 (fused)	128
mistral	`mistralai/Mistral-7B-Instruct-v0.3`	32	7	224
olmo	`allenai/OLMo-2-0425-1B-Instruct`	16	7	112

alongside it. (ii) **Parameter efficiency.** Order cells by mass density m_c/P_c and, for a coverage target $\beta \in (0, 1]$, report the parameter budget

$$\pi(\beta) \;=\; \min\Big\{ \frac{\sum_{c\in S} P_c}{\sum_{c\in\mathcal{C}} P_c} \;:\; S \subseteq \mathcal{C},\; \frac{\sum_{c\in S} m_c}{\sum_{c\in\mathcal{C}} m_c} \geq \beta \Big\}, \tag{9}$$

the smallest parameter fraction whose cells jointly capture a β share of total cancellation, evaluated at $\beta \in \{0.5, 0.8, 0.9\}$ ($\mathcal{C}$ the set of cells, m_c the cell mass, P_c its parameter count). The strong localization hypothesis, Eq. (1), is precisely the claim $\pi(\beta) \ll \beta$. (iii) **Cross-architecture stability.** Top-k Jaccard overlap of the highest-conflict cells across arms, in the canonical (module-role × relative-depth) space of L2, against a random-subset baseline.

An independence reference for the fraction. The sign-conflict fraction (3) is read against a parameter-free reference: if the N=3 domain task vectors carried independent, unbiased signs, the expected fraction would be 0.75 (App. B). This 0.75 is the scale on which the measured ≈0.73 is judged in Sec. 5.1: a *pervasive* conflict fraction is one sitting at this independent-sign reference, not merely a numerically large value. These measures are deliberately simple; their force comes from the scale at which they are applied, namely the adapters and architectures described next.

4 Experimental Setup

Base models (robustness arms). Five instruction-tuned base models are studied, spanning four model sizes and two module naming conventions (Table 1). Qwen2.5-3B serves as the reference arm; the remaining four are robustness arms, used to verify that every claim holds across architectures. Phi-3.5 differs structurally in that it fuses the query/key/value projections into a single `qkv_proj` and the gate/up projections into `gate_up_proj`; all other arms keep them separate.

Adapters, domains, seeds. For each arm, LoRA adapters (Hu et al., 2022) are trained on three sentiment source domains (**restaurant**, **electronics**, and **movies**), each over five random seeds $\{42, 123, 456, 789, 1000\}$ (full per-dataset statistics, identifiers, and terms in App. D). Every adapter uses identical hyperparameters: rank $r = 16$, $\alpha = 32$ (so the task-vector scaling is $\alpha/r = 2$), dropout 0.05, with LoRA applied to all seven projection types: the attention projections {`q`,`k`,`v`,`o`} and the MLP projections {`gate`,`up`,`down`}, in every transformer layer (Phi-3.5 uses the equivalent four fused targets). Each single-domain adapter is trained on 10,000 class-balanced (stratified) examples under a 3-class sentiment scheme {negative, neutral, positive}; the domains draw from distinct corpora: restaurant from Yelp polarity, electronics from Amazon-Reviews-2023 (Electronics, with ratings 1 to 2, 3, and 4 to 5⋆ mapped to the three classes), and movies from IMDb. The binary Yelp and IMDb sources contribute only the negative and positive classes. Training runs 3 epochs of cross-entropy with AdamW (learning rate 2×10^{-4}, cosine schedule, 0.1 warmup, weight decay 0.01), effective batch size 16 (1×16 gradient accumulation), maximum sequence length 384, and gradient checkpointing. The $5 \times 3 \times 5 = 75$ adapters used in the conflict analysis are the three source domains over five seeds for each of the five arms. The task-vector and cell definitions these adapters feed are given in Sec. 3.

Baselines and evaluation. The structural analyses (Sec. 3 and Sec. 5) operate directly on the task vectors and require no merge metric; a single merge comparison confirms the efficiency result of Sec. 5.3. The experimental baselines are the standard uniform merges: linear, TIES (Yadav et al., 2023), DARE-linear (Yu et al., 2024), DARE-TIES, and SLERP. On a separate fitting split (a stratified Amazon-Reviews *beauty* corpus, disjoint from both the training domains and the evaluation panel) the best uniform merge among these is selected and the routing thresholds of a cell-gated *localized* merger are fit; plain linear averaging is the best uniform comparator. The localized merger and this linear baseline are then scored on a fixed panel of four held-out tasks (`sst5_short`, `go_emotions_simplified`, `tweet_eval_sentiment`, `tweet_eval_emotion`, up to 2000 examples each) by **macro-F1** with bootstrap confidence intervals over five seeds, paired per example.

Localized merger and gate fitting. The cell-gated *localized* merger is the policy whose efficiency Sec. 5.3 tests: each (module $\times$ depth) cell whose mean sign-conflict fraction exceeds a threshold τ is merged with sign-resolved (TIES/DARE) merging, while every cell below τ is merged by a plain linear average. The threshold τ and the choice of TIES versus DARE for the gated cells are fit on the held-out *beauty* split above (never on the evaluation panel) by grid search maximizing mean macro-F1. This is the only point at which a localized policy is granted a tuning advantage; how it then transfers is reported in Sec. 5.3.

Artifacts. All analysis artifacts and the scripts that produce them are listed in App. A and shipped as supplementary material. With the adapters trained and the two measures defined, the question becomes empirical: applied at this scale, does sign conflict betray any structure a merger could exploit? The next section answers it.

5 Results

5.1 Sign conflict is pervasive and near-uniform

The per-cell sign-conflict fraction (3) sits at $\approx$0.73 with negligible spread in every architecture (Table 3): roughly three in four parameters conflict, almost everywhere, with standard deviation ≤ 0.013 across all (module $\times$ depth) cells. No small high-conflict subset exists. Although the ordering of cells is itself stable across seeds (mean Kendall $\tau = 0.65$ over the ten seed pairs; `results/<arm>/rank_stability.json`), the differences being ordered are minute, so the stable ordering offers nothing a method could exploit. The cross-architecture top-k Jaccard of the most-conflicted cells is only 0.43 (5-arm canonical) and 0.22 (4-arm native), above the random baseline but far from a stable, shared "hot set". **By fraction, sign conflict is not localized.**

Is 0.73 high? A random-sign reference. A simple reference, requiring nothing to be fitted, fixes the scale. If the N=3 domain task vectors carried independent, unbiased signs at each coordinate ($\Pr[+] = \Pr[-] = \frac{1}{2}$, exact zeros aside), a coordinate conflicts unless all three signs agree, so the expected conflict fraction would be $1 - 2\,(\frac{1}{2})^3 = 0.75$ (App. B). The measured median of 0.72 to 0.75 falls within 0.02 to 0.03 of this random-sign value, consistently at or just below it, a small but systematic lean toward agreement. With millions of coordinates per arm this gap is real rather than noise; the reference serves as a scale for the magnitude of the conflict rather than as a statistical test. On that scale sign conflict is *pervasive*: not merely common, but about as common as independent coin flips would render it, leaving no fraction-level pattern for a sign-vote shortcut to exploit.

Task disagreement or training noise? A seed control. The $\approx$0.73 fraction could reflect genuine disagreement between tasks, or merely the fact that retraining under a new random seed lands on different signs. The same cells and the same conflict test settle the question, with conflict measured across three *seeds* of a single domain instead of three domains at a shared seed (averaged over all $\binom{5}{3}$=10 seed-triples and the three domains; Table 2). The within-domain cross-seed fraction is 0.73 to 0.75 in every arm: equal to, and in four of five arms slightly above, the cross-domain value, and likewise sitting at the 0.75 random-sign value. Two adapters for the *same* domain that differ only in random seed therefore disagree in sign as often as adapters for *different* domains. The widespread conflict thus arises from the training process itself, whose

Table 2: Seed control: within-domain cross-seed sign conflict matches the cross-domain fraction. Median over cells of the N=3 conflict, computed across three domains at a shared seed (*cross-domain*) vs. across three seeds of one domain (*cross-seed*, mean over all 10 seed-triples and the 3 domains); both share the 0.75 independence null. The 5-seed column (N=5 null 0.9375) is for reference. (Provenance: App. A.)

Arm	cross-domain (N=3)	cross-seed (N=3)	cross-seed (N=5)
qwen	0.733	0.735	0.925
llama	0.731	0.735	0.924
phi	0.730	0.729	0.919
mistral	0.747	0.749	0.936
olmo	0.724	0.733	0.922
null	0.75		0.9375

Table 3: Native within-arm spread of the conflict fraction (3) vs. the magnitude-weighted `cancel_fraction`. The fraction is uniform (≈0.73, tiny sd); cancellation intensity is lower (≈0.37) and only modestly more varied. (Provenance: App. A.)

	conflict_fraction			cancel_fraction		
Arm	min	med	sd	min	med	sd
qwen	0.635	0.733	0.012	0.254	0.378	0.018
olmo	0.678	0.724	0.013	0.308	0.369	0.016
phi	0.652	0.730	0.013	0.270	0.372	0.018
mistral	0.692	0.747	0.007	0.328	0.408	0.011
llama	0.677	0.731	0.008	0.321	0.384	0.013

random seed largely determines each weight's sign, not from the tasks diverging. This is precisely why no fraction-level pattern exists for a sign-vote shortcut to find.

5.2 Magnitude reveals architecture-invariant structure in MLP projections

Re-weighting by destructive cancellation mass changes the picture, though one complication must be stated at the outset: MLP `gate/up` projections are both the largest LoRA-adapted tensors and the highest cancellation-mass cells, so raw mass partly tracks size. With that caveat, the raw-mass signal is strong and architecture-invariant: in every arm the **top-15 cancellation-mass cells are 100% `gate/up_proj`** across all five architectures, with top-k Jaccard 0.71 (5-arm canonical) and 0.49 (4-arm native), far above the ≈0.18/0.08 random baselines (Table 4). To separate structure from size, cells are re-ranked by the size-normalized `cancel_fraction`: the overlap falls to 0.47 (5-arm canonical) and 0.16 (4-arm native), markedly weaker, though still above the random baseline. The reading is therefore twofold: *where* the destructive mass sits (MLP gate/up) is stable across architectures but largely driven by these projections' size, whereas the per-weight *rate* of cancellation carries a weaker signal that remains present in every architecture. **Destructive mass concentrates in MLP gate/up projections across all five architectures, with size accounting for much, but not all, of that concentration.**

5.3 The concentration is not actionable: a negative efficiency result

Concentration is moderate but *thinly spread*: the Gini of `cancel_mass` across cells is 0.39 to 0.55 and the top-10% of cells hold only 23 to 30% of the mass (Table 5). The decisive quantity for a localized merger is the parameter budget, and it remains close to the uniform baseline (capturing X% of the mass requires X% of the weights) at *every* coverage level: taking cells in order of mass density, capturing 50% of total cancellation requires 38 to 43% of parameters, 80% requires ≈72%, and 90% requires ≈85% (Table 5), a saving of only ≈4 to 12 points over uniform that shrinks as coverage grows. A cell-gated "localized TIES" restricted to

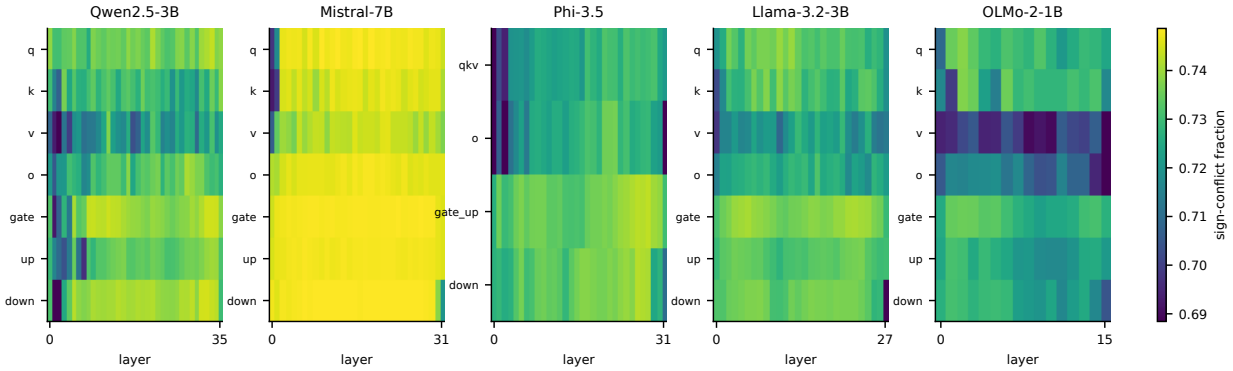

Figure 2: Sign-conflict fraction is pervasive and near-uniform across modules and depths in every architecture. Each panel is one base model (native module type × layer); the colour scale spans only 0.69 to 0.74, i.e. essentially all cells carry a ≈0.73 conflict fraction with no high-conflict hot spots.

Table 4: Cross-architecture top-k Jaccard of highest-conflict cells. Mass-ranked overlap is high and architecture-invariant; fraction- and intensity-ranked overlap are weak. (Provenance: App. A.)

Ranking	Space (cells)	Jaccard	range	random
conflict_fraction	5-arm canonical (20)	0.43	n/a	0.18
conflict_fraction	4-arm native (70)	0.22	n/a	0.08
cancel_mass	5-arm canonical (20)	**0.71**	n/a	0.18
cancel_mass	4-arm native (70)	**0.49**	0.33 to 0.67	0.08
cancel_fraction	5-arm canonical (20)	0.47	0.33 to 0.71	0.18
cancel_fraction	4-arm native (70)	0.16	0.00 to 0.43	0.08

a small high-conflict set would therefore miss most of the destructive interference unless it touched nearly the whole model. Because this ordering ranks cells by cancellation-*mass* density, $\pi(\beta)$ *is* the mass-optimal cell-level gate (the strongest localization signal a cell method could use), so $\pi(0.8) \approx 0.72$ already bounds any mass-gated localized merger analytically, with no merge run required.

A separate check uses the merge itself. The localized merger here selects cells by the conflict *fraction*, which is nearly uniform (Sec. 5.1) and is therefore, by construction, the weaker of the two selection signals; even with that handicap the cell-gated *localized* merger (defined in Sec. 4) fails to beat plain linear merging even on the split its thresholds were tuned on (macro-F1 0.524 vs. 0.525). Per-task panel scores for it and every standard merge appear in Table 7 (App. C). On the four held-out panel tasks the gap only widens: in a paired bootstrap over five seeds (Holm-Bonferroni corrected), the localized policy never improves mean macro-F1 on any architecture. The net effect is negligible on phi, llama, and olmo (mean $|\Delta| < 0.002$, individual tasks scattered around zero) and *significantly negative* on qwen (-0.012) and mistral (-0.047, up to -0.11 on one task). Restricting the fix to a high-conflict subset yields nothing over treating all cells alike.

A caution against reading this in reverse. In the panel (Table 7), TIES, the canonical global sign resolution, does not beat plain linear averaging on any arm; linear is at or near best everywhere, and the sign-electing methods (TIES, DARE-TIES) are the weakest. This does not contradict the localization result, which concerns *where* resolution would have to act, not *whether* resolution helps on these tasks, and two facts hold together. First, the ≈37% cancellation present in *every* cell is genuine and model-wide, so a sign resolution that did help would have to act globally: a cell-localized version misses most of the mass, and the Localized column accordingly ties or loses on every arm. Second, on these near-adjacent sentiment domains there is little for *any* resolution to recover: linear averaging is already near-optimal, and TIES's trim step discards useful magnitude faster than its sign vote returns it, so even the global method falls below linear. **The damage a global vote targets is real and not cell-localizable, yet on tasks this similar**

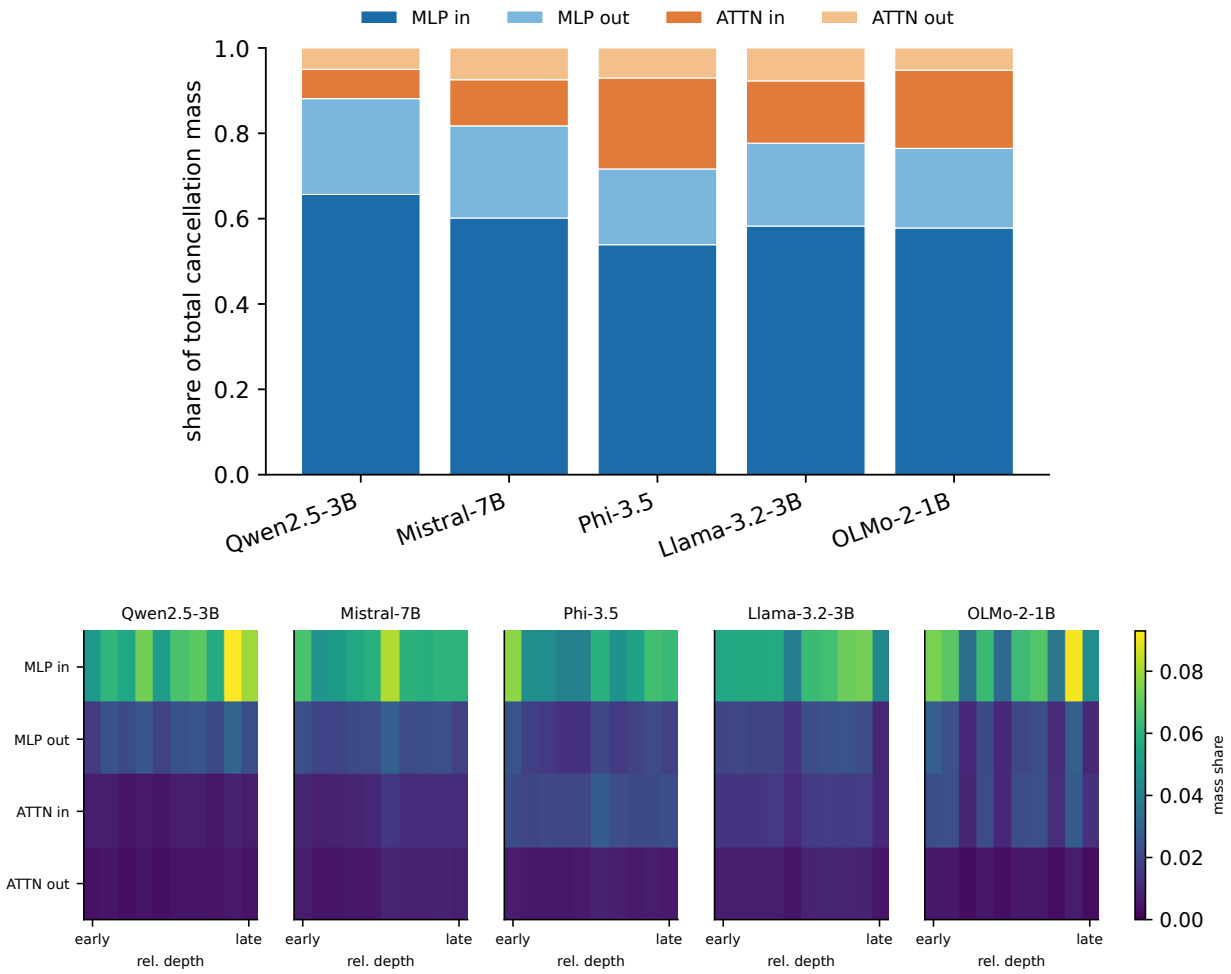

Figure 3: Cancellation mass concentrates in MLP input projections in all five arms. **Top:** share of total cancellation mass by canonical module role. MLP (dark/light blue) carries 72 to 88% in every architecture, MLP_in the single largest role. **Bottom:** per-arm cancellation-mass share over (module role × relative depth); the MLP_in band dominates invariantly across architectures.

Table 5: Localization of cancellation mass. Uniform baseline: Gini ≈0, top-10%→10%, and X% of mass needs X% of params. Observed concentration is modest and the parameter budget to capture most mass stays high. (Provenance: App. A.)

		top-10%	% params to capture cancel-mass		
Arm	Gini	→ mass	50%	80%	90%
qwen	0.551	29.7%	37.8%	70.5%	83.2%
olmo	0.397	23.7%	41.4%	72.3%	85.2%
phi	0.385	24.2%	41.7%	73.4%	85.2%
mistral	0.464	23.2%	43.2%	74.5%	86.1%
llama	0.430	23.0%	40.7%	73.2%	86.2%

even global resolution does not beat plain averaging. Which regime would reward resolution, more divergent tasks where linear leaves real damage on the table, is the open question named in Sec. 6; what the pattern means for merging-method design is taken up next.

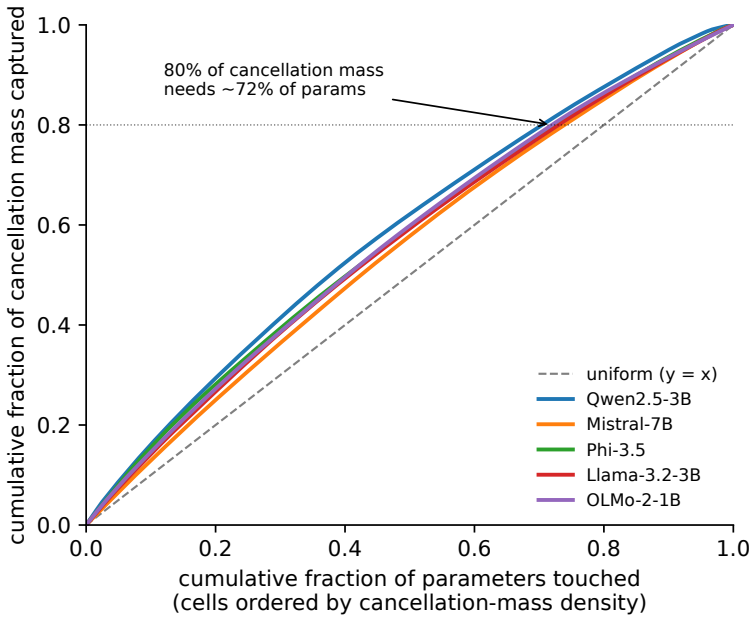

Figure 4: Parameter-efficiency curve. Ordering cells greedily by cancellation-mass density, every architecture hugs the $y = x$ uniform baseline: capturing 80% of the destructive cancellation requires touching ≈72% of parameters. Localized (cell-gated) resolution therefore cannot replace global sign election.

6 Discussion and Limitations

What "localized" should mean. These results separate two questions that the word "localized" usually conflates: *where the destructive mass resides* and *whether one can cheaply act on it*. By the conflict fraction, sign conflict is not localized at all: it is pervasive and near-uniform (Sec. 5.1). Re-weighting by magnitude does yield a clear, interpretable answer to the first question: the destructive mass resides in MLP input projections, invariantly across architectures (Sec. 5.2). The second question, however, is the one that matters for method design, and the answer is negative: the mass is too diffuse for a cell-level budget to capture (Sec. 5.3). Localization as an *interpretation* holds; localization as a *lever* does not.

Implications for merging method design. Three consequences follow. (i) Whatever sign resolution a merge performs cannot be confined to a few cells: because ≈37% of each cell's magnitude is lost to cancellation almost everywhere, the damage a global vote targets is real and spread across the whole model, not gathered in a hot subset. This justifies the global *scope* of the vote, not the vote unconditionally; on the near-adjacent domains studied here even global TIES does not beat plain averaging (Table 7), because there is little task-structured conflict for any resolution to recover (Sec. 6). (ii) A cheaper merger cannot be obtained by restricting the fix to a few structural cells; any genuine saving must be spent on *individual weights*, the territory that DARE-style magnitude pruning (Yu et al., 2024; Deep et al., 2024; Davari & Belilovsky, 2024) already occupies, and which this analysis *justifies* rather than merely re-validates (see "Relation to DARE" below). (iii) For an interference-aware intervention with a structural prior, MLP `gate`/`up` projections are the highest-value targets, consistently across architectures.

Relation to DARE. DARE (Yu et al., 2024) keeps merging cheap by dropping a large fraction of task-vector entries at random and rescaling the rest, quietly assuming that the redundancy which makes this safe is distributed throughout the update rather than confined to a few structural regions. The results here propose no competing merger and make no claim to higher macro-F1; their value relative to DARE is *explanatory*. The widespread ≈73% conflict fraction and the thinly spread parameter budget ($\pi(0.8) \approx 0.72$, close to the uniform diagonal) supply, across five architectures, the evidence for precisely that assumption: the destructive damage is distributed, so a merge budget spent on individual weights (as in DARE) is well

placed, whereas restricting the fix to a few cells (the structural shortcut a cell-level method would take) cannot succeed. In short, this work explains *why* a DARE-style, weight-level treatment is the right design and places a ceiling on what any cell-level alternative could achieve, rather than re-validating DARE on a benchmark.

Limitations: task similarity and the seed-noise reading. The sharpest threat to the central reading is the closeness of the three source domains. Restaurant, electronics, and movie sentiment are all three-class sentiment problems trained under identical hyperparameters, so they diverge only modestly. The headline observation, that two adapters differing only in random seed conflict in sign as often as adapters trained on different domains (Sec. 5.1), is read here as evidence that the conflict is training noise rather than task disagreement; but part of that equality may simply reflect how little these domains diverge in the first place. On genuinely distant tasks (for example math vs. code vs. natural language, the regime DARE and DELLA target), cross-domain conflict could rise above the random-sign reference and carry task structure that cross-seed conflict does not. This would not overturn the fraction being pervasive, but it would qualify the *interpretation* of where the conflict comes from; it is also the same task closeness that leaves little for any sign resolution to recover in the merge panel (Sec. 5.3), so the two readings stand or fall together. A single additional arm with a deliberately divergent task pairing would largely settle this and is the most valuable extension of the present study. By the same token, the single LoRA rank ($r = 16$) is a genuine scope limit, not only a question of magnitudes: because the task vector is $\Delta W = (\alpha/r)BA$, the rank governs how the update is populated, so the per-coordinate density of sign conflict could itself shift with rank.

Limitations: coverage and metric choices. The remaining coverage caveats are narrower. The study uses four held-out evaluation tasks and instruction-tuned base models throughout; the cross-architecture agreement observed here suggests the qualitative picture is not an artifact of a single model. Cancellation is defined on the summed task vectors rather than on the exact weights TIES produces after its drop-and-vote step; this measures the damage a merge must contend with, not what remains after a specific resolution method. Each weight's magnitude is also weighted uniformly; an importance-weighted merge such as Fisher-weighted averaging (Matena & Raffel, 2022) rescales each parameter by its estimated Fisher information, which can alter how costly a given sign disagreement is to the merged model without removing the disagreement itself. Finally, "mass" is dominated partly by the larger MLP tensors, so the cross-architecture stability of *where* mass sits should be read together with the size-adjusted rate, which is weaker (Sec. 5.2).

Limitations: cell- vs. parameter-level scope. The analysis localizes at the (module $\times$ depth) *cell* level. An analysis of individual weights might reveal finer concentration, but acting on it reduces to magnitude pruning; that is, it would re-derive DARE rather than provide a new mechanism. The localization claim is therefore restricted to the cell level, where structural statements that hold across architectures are meaningful and useful to method designers.

Why this is a useful negative result. The strength of the finding is its breadth: the same structure (conflict everywhere, mass concentrated in the MLP, a thinly spread parameter budget) recurs across five architectures spanning four model sizes. The parameter-efficiency curve offers method designers a crisp, actionable takeaway: do not expect cell-localized sign resolution to replace global TIES. Its value is that the claim is informative and correct rather than a new state-of-the-art number. What it leaves open, and what the concluding section takes up, is where a genuinely cheaper merger could come from, if not from structural localization.

7 Conclusion and Future Directions

Across five instruction-tuned architectures, three domains, and five seeds, sign conflicts in LoRA adapter merging are **pervasive**: roughly 73% of weights conflict, almost uniformly. Weighting by magnitude exposes a genuine pattern shared by all five architectures: the destructive cancellation mass concentrates in MLP `gate`/`up` projections, in large part because these are the biggest adapted tensors (Sec. 5.2). Yet this concentration is too diffuse to exploit: capturing most of the cancellation requires touching most of the

weights, so cell-localized resolution cannot replace global TIES, and the cancellation found everywhere is real and not localizable to a hot subset. *Sign conflicts in LoRA merging are pervasive and the destructive mass concentrates in MLP gate/up projections invariantly across five architectures, but it is too distributed for localized resolution to replace global TIES.*

Future directions. Because the cost of merging is distributed rather than concentrated, the most promising direction is to move below the cell level, modeling the damage at the level of individual weights, or to train adapters to be mergeable from the outset by penalizing destructive sign disagreement during fine-tuning, rather than seeking a structural shortcut that, as shown here, does not exist. Alignment methods point to a complementary route: Git Re-Basin (Ainsworth et al., 2023) accounts for the many equivalent orderings of neurons, and KnOTS (Stoica et al., 2025) realigns LoRA adapters into a shared subspace before merging; aligning adapters in weight space this way could reduce precisely the per-weight sign disagreement that no cell-level selection can reach. A further question this analysis leaves open is whether the MLP-input concentration of cancellation mass, invariant across all five architectures, reflects a property of how feedforward layers store task-specific features, a connection that, if established, would turn the present descriptive finding into a predictive one.

Broader Impact Statement

This work is an analysis of *how* independently trained LoRA adapters interfere when their task vectors are merged. It introduces no new model or capability and releases no artifact beyond analysis code and aggregate statistics over public base models. Its practical implication is cautionary: a tempting efficiency shortcut (resolving sign conflict only within a few structural regions) is shown not to work, which should steer practitioners away from localized merges that silently degrade some skills. More reliable merging supports the auditability and reproducibility of composed models; the same techniques could equally compose capabilities without access to the original training data, but we identify no harm specific to this analysis beyond those already general to model-merging research.

Reproducibility Statement

Every numerical claim is supported by an artifact shipped with this submission. The analysis scripts (`scripts/sign_conflict.py`, `scripts/sign_conflict_magweighted.py`, `scripts/magweighted_gate.py`), the supporting `sentiment_lora_merging` package, and the per-arm sign-conflict JSON outputs they produce are provided as supplementary material; running `magweighted_gate.py` from `results/` regenerates Tables 3 to 5 with no external dependencies. Appendix A maps each claim to its file, exact value, and producing script, so every table and figure can be traced to its source without leaving the work; Appendix B derives the independence null of Sec. 5.1, and Appendix C provides the full per-task merge scores. The five base models are public (Table 1); the trained LoRA adapters, and the merge-evaluation outputs derived from them, reconstruct from the configuration and training recipe in Sec. 4.

References

Samuel K. Ainsworth, Jonathan Hayase, and Siddhartha Srinivasa. Git re-basin: Merging models modulo permutation symmetries. In *International Conference on Learning Representations (ICLR)*, 2023.

MohammadReza Davari and Eugene Belilovsky. Model breadcrumbs: Scaling multi-task model merging with sparse masks. In *European Conference on Computer Vision (ECCV)*, 2024.

Pala Tej Deep, Rishabh Bhardwaj, and Soujanya Poria. DELLA-Merging: Reducing interference in model merging through magnitude-based sampling, 2024.

Jonathan Frankle, Gintare Karolina Dziugaite, Daniel M. Roy, and Michael Carbin. Linear mode connectivity and the lottery ticket hypothesis. In *International Conference on Machine Learning (ICML)*, 2020.

Antonio Andrea Gargiulo, Donato Crisostomi, Maria Sofia Bucarelli, Simone Scardapane, Fabrizio Silvestri, and Emanuele Rodolà. Task singular vectors: Reducing task interference in model merging. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2025.

Yifei He, Yuzheng Hu, Yong Lin, Tong Zhang, and Han Zhao. Localize-and-stitch: Efficient model merging via sparse task arithmetic. *Transactions on Machine Learning Research (TMLR)*, 2025.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations (ICLR)*, 2022.

Gabriel Ilharco, Marco Tulio Ribeiro, Mitchell Wortsman, Suchin Gururangan, Ludwig Schmidt, Hannaneh Hajishirzi, and Ali Farhadi. Editing models with task arithmetic. In *International Conference on Learning Representations (ICLR)*, 2023.

Xisen Jin, Xiang Ren, Daniel Preotiuc-Pietro, and Pengxiang Cheng. Dataless knowledge fusion by merging weights of language models. In *International Conference on Learning Representations (ICLR)*, 2023.

Michael Matena and Colin Raffel. Merging models with fisher-weighted averaging. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022.

Guillermo Ortiz-Jiménez, Alessandro Favero, and Pascal Frossard. Task arithmetic in the tangent space: Improved editing of pre-trained models. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023.

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

Ke Wang, Nikolaos Dimitriadis, Guillermo Ortiz-Jiménez, François Fleuret, and Pascal Frossard. Localizing task information for improved model merging and compression. In *International Conference on Machine Learning (ICML)*, 2024.

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

Prateek Yadav, Derek Tam, Leshem Choshen, Colin Raffel, and Mohit Bansal. TIES-Merging: Resolving interference when merging models. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023.

Prateek Yadav, Tu Vu, Jonathan Lai, Alexandra Chronopoulou, Manaal Faruqui, Mohit Bansal, and Tsendsuren Munkhdalai. What matters for model merging at scale? *Transactions on Machine Learning Research (TMLR)*, 2025.

Enneng Yang, Zhenyi Wang, Li Shen, Shiwei Liu, Guibing Guo, Xingwei Wang, and Dacheng Tao. AdaMerging: Adaptive model merging for multi-task learning. In *International Conference on Learning Representations (ICLR)*, 2024.

Enneng Yang, Li Shen, Guibing Guo, Xingwei Wang, Xiaochun Cao, Jie Zhang, and Dacheng Tao. Model merging in LLMs, MLLMs, and beyond: Methods, theories, applications and opportunities. *ACM Computing Surveys*, 2026.

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 *International Conference on Machine Learning (ICML)*, 2024.

A Reproducibility and Claim Provenance

Every claim in the work maps to a specific on-disk artifact, the exact number, and the script that produces it, all included in the supplementary bundle: the cited analysis scripts under `scripts/`, the arm registry at `src/sentiment_lora_merging/models/registry.py`, and the JSON outputs under `results/` (the qwen reference arm at the `results/` root, the other four arms under `results/<arm>/`). Paths below are relative to the bundle root.

Table 6: Each work claim, its evidence artifact, the headline number, and the producing script. "arm" ∈ {(root)=qwen, llama, phi, mistral, olmo}.

Claim	**Evidence (file)**	**Number**	**Script**
C1: conflict fraction pervasive & uniform (≈0.73)	`results/<arm>/sign_conflict_cells.json`; Table 3	med 0.72 to 0.75, sd ≤ 0.013	`scripts/sign_conflict.py`
C1: fraction sits at the independence null (0.75)	App. B; Table 3	$1 - 2^{1-N} = 0.75$ at N=3	n/a
C1: fraction is seed-noise, not task disagreement	Table 2; Sec. 5.1	cross-seed ≈ cross-domain ≈0.73	`scripts/seed_control.py`
C1: not localized by fraction (weak cross-arm overlap)	`scripts/magweighted_gate.py` (A,C)	Jaccard 0.43 / 0.22 vs rand 0.18 / 0.08	`scripts/magweighted_gate.py`
C2: cancellation mass → MLP gate/up, all arms	`results/<arm>/sign_conflict_magweighted.json`; Fig. 3	top-15 cells 100% MLP_in (5/5 arms)	`scripts/magweighted_gate.py` (D)
C2: mass-location architecture-invariant	Table 4	Jaccard 0.71 (canon) / 0.49 (native)	`scripts/magweighted_gate.py` (C)
C3: concentration diffuse (Gini, top-10%)	Table 5	Gini 0.39 to 0.55; top-10%→23 to 30%	`scripts/magweighted_gate.py` (B)
C3: not actionable (param budget, $\pi(0.8)>0.4$)	Fig. 4; Table 5	80% mass needs ≈72% params	`scripts/magweighted_gate.py` (B)
C3: localized merge does not beat linear	Table 7; App. C	panel mean ties or loses, all arms	`results/<arm>/c3_localized.json`
C3: global TIES justified (∼37% cancel everywhere)	`results/<arm>/sign_conflict_magweighted.json`; Table 3	cancel_fraction med ≈0.37	`scripts/magweighted_gate.py` (A)
Setup: 5 arms / cells	`src/sentiment_lora_merging/models/registry.py`; Table 1	36/28/32/32/16 layers; 252/196/128/224/112 cells	n/a

Reproduction. The sign-conflict definitions of Sec. 3 are implemented in `scripts/sign_conflict_magweighted.py` and scored by `scripts/magweighted_gate.py` (bundle commit 51ef83d). The cited scripts and their JSON outputs are bundled, so every number can be both inspected at source and re-run. The localization tables need no external dependencies: running `python ../scripts/magweighted_gate.py` from `results/` reproduces Tables 3 to 5 (and the 100% MLP_in top-cells result) directly from the bundled `results/<arm>/sign_conflict_magweighted.json`. To recompute the per-cell JSON them-

selves, install the package (`pip install -e .`) and run `python scripts/sign_conflict.py` and `python scripts/sign_conflict_magweighted.py`; these additionally require the trained LoRA adapters, which reconstruct from the public base models in Table 1 and the training recipe in Sec. 4. The per-task merge scores of Table 7 are produced by the same pipeline over the held-out evaluation panel.

B Independence Null for the Conflict Fraction

Section 5.1 compares the observed sign-conflict fraction against the value expected if domain task vectors carried independent signs. The reference is elementary. Fix a coordinate and let $s_1, \ldots, s_N \in \{+, -\}$ be the signs of the N domain task vectors there, modeled as independent and unbiased ($\Pr[s_i = +] = \Pr[s_i = -] = \frac{1}{2}$); exact zeros have measure zero under any continuous weight distribution and are excluded by the abstention rule of Eq. (2). By that definition a coordinate conflicts unless all N signs agree, so

$$\Pr[\text{conflict}] \;=\; 1 - \Pr[\text{all }+] - \Pr[\text{all }-] \;=\; 1 - 2\left(\tfrac{1}{2}\right)^N \;=\; 1 - 2^{1-N}. \tag{10}$$

For $N = 3$ this is $1 - 2^{-2} = 0.75$. The measured median conflict fraction across arms is 0.72 to 0.75 (Table 3), so the observed rate is numerically close to, and consistently just below, the independent-sign null, within 0.02 to 0.03: cross-domain signs are close to independent, with a small but systematic bias toward agreement. At the millions-of-coordinates sample size this deviation is statistically real, so the null is used as a *reference scale* for the magnitude of the conflict, not as a statistical test. For calibration, a fraction clearly below 0.75 would indicate systematic sign *agreement* (redundant adapters), and a fraction above 0.75 systematic *disagreement*; the data show neither. This places the headline ≈0.73 on an absolute scale and answers in advance the question of whether it is "high": it is very nearly as high as random sign assignment renders it, and no higher.

The same null serves the within-domain seed control of Sec. 5.1: comparing N=3 seeds of one domain shares the 0.75 reference, while the 5-seed comparison has null $1 - 2^{-4} = 0.9375$. The measured cross-seed fractions (Table 2) sit at these nulls just as the cross-domain fractions do, which is why the conflict is attributed to seed-level sign non-identifiability rather than task disagreement.

C Per-Task Macro-F1 of the Merge Evaluation

Table 7 reports panel macro-F1 for every standard uniform merge and for the cell-gated localized policy of Sec. 5.3, per held-out task and per architecture (mean over the five seeds). These are the per-task numbers summarized as panel means in Sec. 5.3. Two patterns hold throughout: plain *linear* averaging is at or near the best uniform merge on every arm, and the *localized* policy, despite its thresholds being tuned on a held-out split, never exceeds it on the panel mean (it ties or loses on every architecture), consistent with the negative efficiency result. GoEmotions is a 28-class problem far outside the adapters' sentiment training domain, so every policy scores near floor there (< 0.2 macro-F1); for that task the meaningful quantity is the paired per-example delta between policies, not the absolute score. Absolute scores are modest overall because these are zero-shot transfers to held-out label sets, but the comparison is not a floor effect: SST-5, TweetSent, and especially TweetEmo carry clear signal (TweetEmo spans 0.39 to 0.70 across arms and separates the policies by up to 0.14 within an arm), so linear's advantage is decided on tasks with real range rather than on saturated ones.

D Datasets, Statistics, and Terms

All data are public on the HuggingFace Hub under the identifiers in Table 8. Training uses three source domains, each subsampled to 10,000 stratified examples per seed under the 3-class scheme {negative(0), neutral(1), positive(2)}; the two binary sources (Yelp, IMDb) contribute only the negative and positive classes, while the Amazon rating scale is mapped 1 to 2⋆→neg, 3⋆→neutral, 4 to 5⋆→pos. Each arm trains these three domains over five seeds $\{42, 123, 456, 789, 1000\}$. Threshold fitting for the localized merger and the uniform-merge selection use a disjoint Amazon-Reviews *All_Beauty* slice (≈ 1,000 stratified, ≈ 333 per

Table 7: Panel macro-F1 (mean over five seeds) by architecture and held-out task. Columns are the standard uniform merges and the cell-gated localized policy; the **best** entry in each row is bold. "Mean" averages the four tasks. Linear is at or near the best uniform merge everywhere, and the localized policy never beats linear on the panel mean. Source: `results/<arm>/merge_eval/` and `results/<arm>/c3_localized.json`.

Arm	Task	Linear	TIES	DARE-lin	DARE-TIES	SLERP	Localized
qwen	SST-5	**0.469**	0.413	0.468	0.407	0.430	0.442
	GoEmotions	**0.096**	0.058	0.094	0.047	0.049	0.082
	TweetSent	**0.400**	0.390	0.399	0.390	0.384	0.396
	TweetEmo	**0.696**	0.636	0.690	0.611	0.630	0.682
	Mean	**0.415**	0.374	0.413	0.364	0.373	0.401
llama	SST-5	0.300	0.286	0.296	0.280	**0.301**	0.299
	GoEmotions	**0.086**	0.055	0.086	0.049	0.055	0.084
	TweetSent	0.355	0.331	0.359	0.333	0.325	**0.360**
	TweetEmo	**0.603**	0.466	0.601	0.435	0.506	0.598
	Mean	**0.336**	0.285	0.335	0.274	0.297	0.335
phi	SST-5	**0.432**	0.355	0.429	0.339	0.367	0.428
	GoEmotions	0.126	0.057	**0.132**	0.053	0.061	0.124
	TweetSent	**0.394**	0.358	0.393	0.335	0.354	0.394
	TweetEmo	0.684	0.587	**0.686**	0.566	0.591	0.685
	Mean	0.409	0.339	**0.410**	0.323	0.343	0.408
mistral	SST-5	0.436	0.385	**0.438**	0.374	0.409	0.388
	GoEmotions	**0.189**	0.119	0.189	0.117	0.142	0.125
	TweetSent	0.374	0.382	0.371	0.379	0.379	**0.383**
	TweetEmo	**0.678**	0.554	0.667	0.523	0.580	0.557
	Mean	**0.419**	0.360	0.416	0.348	0.377	0.363
olmo	SST-5	0.380	0.367	**0.382**	0.375	0.376	0.381
	GoEmotions	0.075	0.048	**0.076**	0.038	0.044	0.076
	TweetSent	0.389	0.358	**0.390**	0.346	0.357	0.389
	TweetEmo	**0.393**	0.294	0.390	0.257	0.287	0.389
	Mean	0.309	0.267	**0.310**	0.254	0.266	0.309

class, sample seed 42) that is never trained on and never enters the evaluation panel. The panel itself is four held-out classification tasks scored by macro-F1 over up to 2,000 examples each.

Each dataset is used for non-commercial research in accordance with its public release. GoEmotions is distributed under Apache-2.0; the Yelp, Amazon-Reviews-2023, IMDb, and SST sources are released for research use under their respective dataset terms; and the TweetEval splits depend on tweets governed by the platform's terms of service. The loaders that materialize every split are in `src/sentiment_lora_merging/data/`.

Table 8: Datasets used, with HuggingFace identifier, role, label scheme, size, and distribution terms. "train" sizes are per seed; binary sources yield no neutral class. Terms reflect each dataset's public release; readers should consult the linked HuggingFace cards for the authoritative license.

Dataset (HF id)	**Role**	**Labels used**	**Size**	**Terms**
`yelp_polarity`	train: restaurant	neg, pos	10,000	Yelp Dataset terms
`McAuley-Lab/Amazon-Reviews-2023` (Electronics)	train: electronics	neg, neutral, pos	10,000	research use
`imdb`	train: movies	neg, pos	10,000	research use
`McAuley-Lab/Amazon-Reviews-2023` (All_Beauty)	fit split (gate / merge selection)	neg, neutral, pos	$\approx$1,000	research use
`SetFit/sst5` (sst5_short)	eval panel	5: terrible / bad / ok / good / great	$\leq$ 2,000	research use
`go_emotions` (simplified)	eval panel	27 emotions + neutral	$\leq$ 2,000	Apache-2.0
`tweet_eval` (sentiment)	eval panel	neg, neutral, pos	$\leq$ 2,000	Twitter terms
`tweet_eval` (emotion)	eval panel	anger / joy / optimism / sadness	1,421	Twitter terms