LEARNING NOT TO INTERFERE: ADAPTING EXPERTS FOR MODEL MERGING

Anonymous authors
Paper under double-blind review

ABSTRACT

Model merging combines experts fine-tuned from a shared pretrained initialization into a single multitask model, but overlapping parameter updates often degrade their specialized capabilities. We argue that this interference stems from how experts are fine-tuned. Standard fine-tuning optimizes each expert on its own task distribution without constraining how its parameter update affects predictions on other tasks. These parameter updates, known as task-vectors, represent changes from the shared initialization and can disrupt other experts when combined. We propose **Resolving Interference (RI)**, a lightweight framework that independently adapts experts before merging using unlabeled task data. RI distills the original expert's predictions on its own task to preserve specialization, while matching the shared pretrained model's predictions on other tasks. The pretrained model provides a reference for behavior before specialization, allowing RI to restrict each task vector's influence beyond its own task distribution. RI complements existing merging methods, including those using test-time adaptation, and improves performance across vision and language benchmarks. For language-model merging, RI surpasses the strongest evaluated baselines by up to **2.9%**. Additionally, quantitative consistency evaluations and qualitative generations confirm RI's ability to better preserve the original experts' behavior across tasks.

1 INTRODUCTION

Model merging has achieved remarkable success in recent years, showing that multitask models can be created by directly combining the parameters of independently trained specialist models (Ilharco et al., 2023). These methods can consolidate capabilities acquired through fine-tuning (Ilharco et al., 2023; Choshen et al., 2022), integrate skills across tasks (Stoica et al., 2024a;b), and improve out-of-distribution robustness beyond the original models (Rame et al., 2022; Wortsman et al., 2022).

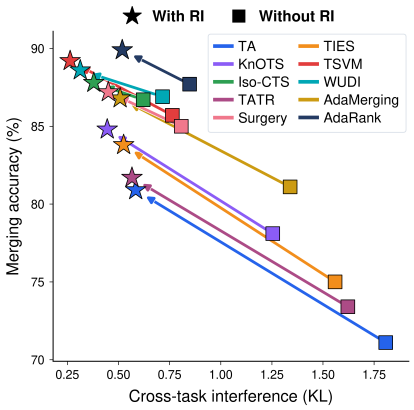


Figure 1: **Resolving Interference (RI)** is a lightweight adaptation strategy that mitigates cross-task interference, enhancing the performance of existing model-merging techniques.

Cross-task interference. However, obtaining strong merged models remains challenging. Merging quality depends on the compatibility of model parameters, the distributions used for pretraining and fine-tuning, and conflicts between task-specific updates (Hammoud et al., 2024; Stoica et al., 2024b; Wang et al., 2024a; Yadav et al., 2023; Yu et al., 2024). These conflicts can cause the merged model's predictive distribution to drift away from those of the experts on their corresponding task, degrading performance. This drift in the merged model's representation is commonly referred to as *cross-task interference*.

Existing approaches to reducing interference. Prior methods fall into two broad categories. Gradient-free approaches reduce parameter conflicts through pruning, rescaling, or alignment (Yadav et al., 2023; Yu et al., 2024; Stoica et al., 2024b), but these static transformations do not constrain how each update affects predictions on other input distributions. Gradient-based approaches (Yang et al., 2024b; Lee et al., 2026) use unlabeled data to optimize merging coefficients or select singular

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.

components. They reweight expert contributions but do not adapt the experts themselves. Representation Surgery (Yang et al., 2024a) corrects representation drift with task-conditioned modules learned on top of the merged model. It compensates for drift after merging and leaves the expert updates unchanged. In each case, interference is handled downstream rather than at its source, which is the disruptive behavior of the experts. This motivates a *pre-merge* intervention: adapting each expert to preserve its specialized behavior while limiting its influence on other task distributions, so that experts are more compatible when merged.

What causes interference? To answer this we review how the experts were fine-tuned. Standard fine-tuning optimizes each expert on its own task distribution, without explicitly constraining how its parameter update affects inputs from other tasks. The resulting *task vector*—the difference between the fine-tuned parameters and their pretrained initialization—can therefore change predictions beyond the task distribution it was trained for. Our analysis in Section 3 shows that expert task vectors can substantially affect the representations for inputs in the domain of another task. When combined with another expert, these updates can induce prediction drift associated with performance degradation. These observations motivate the crux of our approach: each expert task vector's functional influence should be limited to its own task distribution. These observations motivate localizing each update's functional influence to its own task distribution (Ortiz-Jimenez et al., 2023), while retaining the behavior learned through fine-tuning.

Resolving Interference. We propose **Resolving Interference (RI)**, a lightweight framework that adapts each expert before merging using unlabeled task data. RI combines two distillation objectives. The *task-preservation* objective matches the original expert's predictive distributions on inputs from its own task, while the *interference-reduction* objective matches the shared pretrained model's predictive distributions on inputs from other tasks. The pretrained model provides a reference for behavior before specialization, allowing RI to reduce the changes induced by fine-tuning on other task distributions. Jointly, these objectives encourage each expert's task-vector to retain its specialized behavior while localizing its influence to inputs from its own task distribution, resulting in minimal perturbation to other experts' predictions when merged.

Complementarity with existing methods. RI produces adapted experts that are highly compatible with one another, improving the performance of the resulting merged model. The adaptation is agnostic to the downstream merging method, allowing RI to complement existing merging methods including test-time adaptation methods. RI adds no inference-time modules of its own, and its independent expert adaptation enables parallel execution without retaining all expert backbones simultaneously. Across vision and language benchmarks and multiple model scales, RI consistently improves merging performance.

Our contributions are threefold:

- **Diagnosing interference through the effects of expert updates.** We characterize how task vectors change predictions beyond their own task distributions. Controlled experiments that add other task vectors to a fixed expert demonstrate how these updates can degrade its performance, motivating constraints on expert behavior before merging.
- **A framework for adapting experts before merging.** We introduce RI, which independently adapts experts using unlabeled task inputs to preserve their original predictions on their own tasks while matching the shared pretrained model's predictions on other tasks. RI complements existing merging methods, including those using test-time adaptation, without introducing additional inference-time modules.
- **Improved merging performance and expert consistency.** Across CLIP vision benchmarks and Llama and Gemma language benchmarks, the best RI-enhanced configurations outperform the strongest evaluated baselines in every main experimental setting. Beyond aggregate performance, quantitative consistency measurements and qualitative generations demonstrate better retention of successful expert behavior after merging.

2 Related Work

Model merging seeks to integrate independently trained models into a single unified model. Early studies revealed that models trained from the same initialization can be linearly interpolated without increasing test error—a phenomenon known as *mode connectivity*. (Draxler et al., 2018; Garipov et al.,

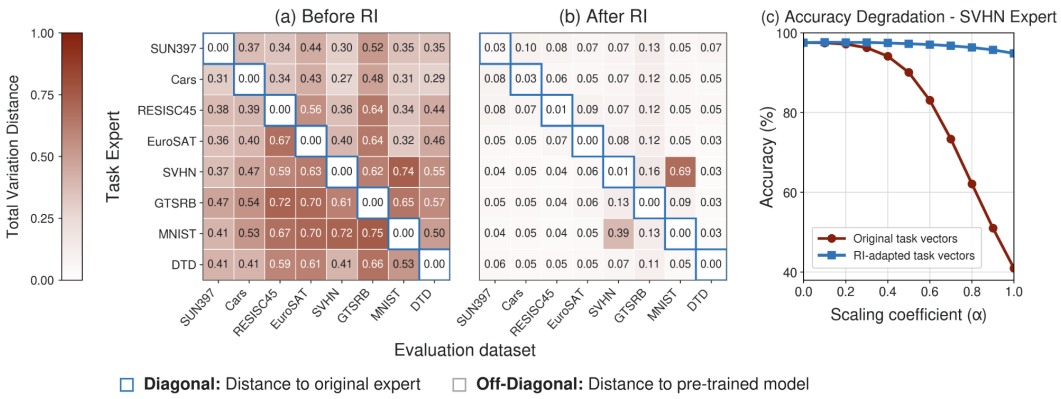


Figure 2: **RI preserves specialization and makes task vectors less disruptive when merged. (a,b)** Heatmaps report mean total variation distances across experts (rows) and datasets (columns). Diagonal entries compare against the original expert; off-diagonal entries compare against the shared pretrained model, the reference before specialization. RI reduces changes on other tasks while preserving own-task predictions. **(c)** Adding RI-adapted updates from other tasks causes substantially less accuracy degradation for the fixed, unscaled SVHN expert (94.9% versus 41.0% with original updates at $\alpha = 1$). Similar analysis for other experts models can be found in Figure 8.

2018; Simsek et al., 2021; Frankle et al., 2020; Neyshabur et al., 2020). Subsequent work demonstrated that simple weight averaging not only preserves accuracy but can also improve generalization and reduces overfitting (Choshen et al., 2022; Wortsman et al., 2022; Wang et al., 2024a; McMahan et al., 2017; Rame et al., 2024; 2022). Task Arithmetic (Ilharco et al., 2023) introduced the concept of a *task vector*—the difference between fine-tuned and pretrained weights that captures task-specific knowledge. Task vectors can be merged and then added back to the pretrained weights to produce a unified model.Ortiz-Jimenez et al. (2023) link effective task arithmetic to *weight disentanglement*, whereby updates have task-specific effects in function space, and strengthen this property through tangent-space fine-tuning.

Reducing interference. Existing methods address interference through parameter transformations, adaptive aggregation, or representation correction. TIES (Yadav et al., 2023) trims small updates and resolves sign conflicts, while DARE (Yu et al., 2024) randomly drops and rescales updates. Among spectral approaches, KnOTS (Stoica et al., 2024b) aligns LoRA updates in a shared space, while TSV-M (Gargiulo et al., 2025) compresses task matrices and decorrelates their singular vectors. Iso-CTS (Marczak et al., 2025) balances common and task-specific subspaces through an isotropic singular-value spectrum. TATR (Sun et al., 2025) restricts merging using estimated cross-task loss sensitivity, while WUDI (Cheng et al., 2025) optimizes a task-vector-based approximation of expert-output discrepancies. AdaMerging (Yang et al., 2024b) and AdaRank (Lee et al., 2026) use entropy minimization on unlabeled inputs to learn merging coefficients and select singular components, respectively. Representation Surgery (Yang et al., 2024a) directly reduces discrepancies between merged and expert representations through learned task-specific correction modules.

Preparing experts for merging. While existing methods reduce interference, merged models often underperform their constituent experts on the corresponding tasks. We argue that a key source of this interference lies in how experts are fine-tuned. Standard fine-tuning optimizes each expert for its own task without explicitly constraining how its update affects predictions on other tasks. This motivates RI, which adapts experts before merging to make their updates less disruptive to other tasks while retaining their specialized capabilities. RI complements existing merging methods, including those that use test-time adaptation.

3 Background and Motivation

Problem setting and notation. We consider $N \geq 2$ expert models with a shared architecture and pretrained initialization θ_0. Expert i, with parameters θ_i, has been fine-tuned on input distribution P_i on a task such as instruction-following, code-generation, or multilingual understanding. Our goal is

to combine these experts into a single model that retains their specialized capabilities. We assume access to unlabeled task data $\mathcal{D}_i$ from each P_i, with task membership known but without target labels.

Let $f(x;\theta)$ denote the model's predictive probability distribution. For an autoregressive language model, x is a token context and $f(x;\theta)$ is a next-token distribution over a common vocabulary, including the shared output projection. Following Ilharco et al. (2023), we define expert i's *task vector* as $\tau_i = \theta_i - \theta_0$, the parameter change from the shared pretrained initialization. Parameter merging then gives $\theta_{\mathrm{m}} = \theta_0 + M(\tau_1, \ldots, \tau_N)$, where M is a merging operator, with method-specific data and hyperparameters left implicit.

Cross-task interference. Combining expert updates can change the merged model's predictions relative to each original expert on its own task. Following the discrepancy-based perspective of prior work (Yang et al., 2024a; Ortiz-Jimenez et al., 2023), we quantify this *cross-task interference* at the predictive output as

$$\xi(\theta_{\mathrm{m}}) = \frac{1}{N}\sum_{i=1}^{N} \mathbb{E}_{x\sim P_i}\left[D(f(x;\theta_i), f(x;\theta_{\mathrm{m}}))\right], \tag{1}$$

where D is a nonnegative output discrepancy and the original experts provide fixed references. We evaluate task performance separately to establish the practical consequences of merging.

Fine-tuning does not localize an expert's influence. Standard fine-tuning promotes specialization on an expert's own task without explicitly constraining its effects on other task distributions. We examine these effects using eight vision experts with a ViT-B/32 backbone. In Figure 2(a), row i selects expert $\theta_i = \theta_0 + \tau_i$, and column j supplies inputs from task j. Each cell reports the mean distance between its predictive distribution and a reference distribution: the original expert on the diagonal, and the pretrained model off the diagonal. We use total variation, $\mathrm{TV}(p,q) = \frac{1}{2}\|p-q\|_1$, because its fixed range $[0,1]$ makes the heatmap magnitudes easy to interpret.

The diagonal entries are zero by construction, whereas the substantial off-diagonal distances show that an expert's update also changes model predictions away from the pretrained distribution on tasks for which it was not optimized. Thus, its functional influence extends beyond its own task distribution. These individual-update measurements motivate examining whether the same updates disrupt another expert when combined with it.

Non-localized task-vectors disrupt other experts. We hold an original target expert θ_i fixed and progressively add scaled task vectors from the remaining experts:

$$\theta_i(\alpha) = \theta_i + \alpha \sum_{j\neq i} \tau_j, \qquad \alpha \in [0,1]. \tag{2}$$

The target expert's own update remains unscaled. The original non-localized task-vector curve in Figure 2(c) shows substantial degradation on the SVHN task: accuracy decreases from 97.6% at $\alpha = 0$ to 41.0% at $\alpha = 1$. This controlled experiment demonstrates that task-vectors learned for other tasks can substantially impair an existing expert's performance. Accuracy and total-variation sweeps for all eight tasks are provided in Figure 8. Together, these observations motivate localizing each expert's update to its own task distribution, consistent with the functional perspective of weight disentanglement (Ortiz-Jimenez et al., 2023). The desired update should retain the expert's specialized predictions on its own task while matching the pretrained model's predictions on other tasks. Such behavior would limit the changes introduced by unrelated task-vectors when they are merged.

4 Method: Resolving Interference

We propose **Resolving Interference (RI)**, a framework that adapts expert task-vectors before merging using unlabeled task data. For expert i, we initialize a trainable update $\tau_i^\star \leftarrow \tau_i$ and retain the original expert θ_i and pretrained model θ_0 as frozen references. Our goal is to obtain an adapted expert with the following behavior:

$$f(x;\theta_0+\tau_i^\star) \approx \begin{cases} f(x;\theta_i), & x\sim P_i, \\ f(x;\theta_0), & x\sim P_j, \quad j\neq i. \end{cases} \tag{3}$$

The first target preserves behavior learned through fine-tuning; the second restores behavior before specialization on other task distributions $x \sim P_j, \quad j \neq i$. We enforce these distributional goals through two soft distillation objectives, allowing a trade-off where task distributions overlap.

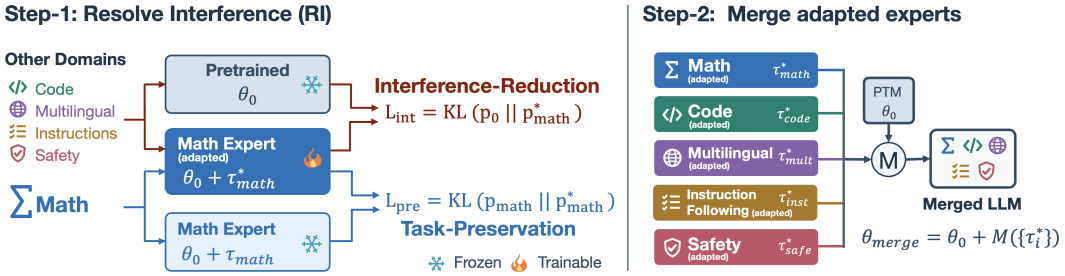


Figure 3: **Resolving Interference (RI) prepares experts for merging. (1) Resolving Interference.** Illustrated for the math expert, RI uses unlabeled inputs and two KL distillation objectives: *task preservation* matches the original expert's predictions on its own task, while *interference reduction* matches the shared pretrained model's predictions on other tasks. Together, these objectives preserve specialization while restricting the task vector's influence on other tasks. **(2) Merging adapted experts.** After adaptation any merging operator M combines the adapted task vectors $\{\tau_i^\star\}$, which are added to the shared pretrained initialization θ_0 to obtain a single multitask model.

Task preservation. For a candidate update τ, the preservation loss matches the original expert's predictions on inputs from its own task:

$$\mathcal{L}^{(i)}_{\text{pre}}(\tau) = \mathbb{E}_{x\sim\mathcal{D}_i}\left[D(f(x;\theta_i), f(x;\theta_0+\tau))\right]. \tag{4}$$

Expectations over datasets denote empirical averages. This objective anchors the adapted model to the expert's specialized behavior while its update is modified.

Interference reduction. On inputs from other tasks, the pretrained model provides the reference distribution:

$$\mathcal{L}^{(i)}_{\text{int}}(\tau) = \frac{1}{N-1}\sum_{j\neq i}\mathbb{E}_{x\sim\mathcal{D}_j}\left[D(f(x;\theta_0), f(x;\theta_0+\tau))\right]. \tag{5}$$

Matching this reference reduces the changes that expert i's update induces on other task distributions. The reference captures behavior before specialization, so the objective does not require other fine-tuned experts as distillation teachers. Averaging over tasks gives each other task equal weight, independently of dataset size.

The RI objective. We combine the two losses as

$$\mathcal{L}^{(i)}_{\text{RI}}(\tau) = \mathcal{L}^{(i)}_{\text{pre}}(\tau) + \lambda\mathcal{L}^{(i)}_{\text{int}}(\tau), \qquad \lambda \geq 0. \tag{6}$$

Starting from τ_i, we optimize this objective to obtain $\tau_i^\star$, where λ controls the trade-off between task-preservation and interference-resolution. Jointly, the objectives encourage each update to retain its specialized behavior while reducing its effects on other tasks. This is a surrogate for reducing cross-task interference and improving compatibility: nonlinear interactions can still arise after parameter merging.

Adaptation and merging. Given the prepared adaptation inputs, each expert can be optimized independently using its original checkpoint and the shared pretrained reference, without keeping all expert backbones resident. Once adaptation is complete, we apply the chosen merging operator:

$$\theta^{\text{RI}}_{\text{m}} = \theta_0 + M(\tau_1^\star, \ldots, \tau_N^\star). \tag{7}$$

RI is agnostic to the downstream merging rule and adds no inference-time modules of its own. Algorithm 1 summarizes the procedure, while Figure 3 demonstrates it.

4.1 Effect of RI on Localization and Merging

We now revisit the diagnostics in Section 3. Figure 2(b) evaluates each adapted expert $\theta_0 + \tau_i^\star$ using the same references as the original heatmap. The broadly reduced off-diagonal distances show that RI limits the changes induced on other task distributions, while small diagonal distances indicate that the original experts' predictions are closely preserved. Residual effects remain, particularly between SVHN and MNIST, so the results demonstrate improved rather than complete localization.

To evaluate whether this improved localization translates into less disruptive merging, we repeat the controlled sweep using adapted updates from the other tasks: $\theta_i^{\mathrm{RI}}(\alpha) = \theta_i + \alpha \sum_{j \neq i} \tau_j^\star$. The target expert θ_i remains original and unscaled in both conditions; only the added updates change. In Figure 2(c), the SVHN expert retains 94.9% accuracy at $\alpha = 1$, compared with 41.0% when adding the original updates. Relative to its initial 97.6% accuracy, the loss decreases from 56.6 to 2.7 percentage points. Because the target expert is unchanged, this comparison directly demonstrates that adapting the other updates reduces their disruptive effect. Full accuracy and total-variation sweeps for all eight tasks appear in Figure 8.

Algorithm 1: Resolving Interference (RI)

Input: Initialization θ_0; task vectors $\{\tau_i\}_{i=1}^N$; datasets $\{\mathcal{D}_i\}_{i=1}^N$; discrepancy metric D; weight λ; steps K; merger M
Output: Merged model $\theta_{\mathrm{m}}^{\mathrm{RI}}$

```
// Model outputs and batch-average discrepancy
f_0(x) ≡ f(x; θ_0) // Pretrained Logits
f_i(x) ≡ f(x; θ_0 + τ_i) // Expert Logits
f_i^⋆(x) ≡ f(x; θ_0 + τ_i^⋆)
D_B(f, g) ≡ |B|^{-1} Σ_{x∈B} D(f(x), g(x))
for i = 1, ..., N (independently) do
    τ_i^⋆ ← τ_i
    for k = 1, ..., K do
        Sample B_j ⊂ D_j for j = 1, ..., N
        // Task preservation
        L̂_pre^(i) ← D_{B_i}(f_i, f_i^⋆)
        // Interference reduction
        L̂_int^(i) ← 1/(N−1) Σ_{j≠i} D_{B_j}(f_0, f_i^⋆)
        // RI objective
        L̂_RI^(i) ← L̂_pre^(i) + λ L̂_int^(i)
        Update only τ_i^⋆ using ∇_{τ_i^⋆} L̂_RI^(i)
    end
end
// Merge adapted task vectors
θ_m^RI ← θ_0 + M(τ_1^⋆, ..., τ_N^⋆)
return θ_m^RI
```

5 Experiments and Results

Merging baselines. We evaluate RI with Task Arithmetic (Ilharco et al., 2023), TIES (Yadav et al., 2023), TSV-M (Gargiulo et al., 2025), and Iso-CTS (Marczak et al., 2025), as well as test-time adaptation methods like layer-wise AdaMerging (Yang et al., 2024b), Surgery Yang et al., 2024a and AdaRank (Lee et al., 2026). We compare merging with the original experts with their RI-adapted counterparts using the corresponding merging pipeline.

Adaptation protocol. We follow the test-time adaptation setting, using unlabeled test inputs from each task for RI adaptation. We also evaluate RI with held-out unlabeled task data in Section 6. We optimize the RI loss (Eq. 6) using *KL-divergence* and compare alternative distance metrics in Section C.2. We set $\lambda = 1.0$, which yields stable interference reduction, and use a batch size of 32. We adapt each ViT expert for $2{,}500$ steps on an NVIDIA A40 GPU and each Llama-3.1-8B-Instruct or Gemma-2-9B-Instruct expert for 500 steps on an NVIDIA H200 GPU. Further details regarding RI adaptation and hyperparameter tuning strategy for each merging method can be found in Section B.

5.1 Merging Language Models

Setup. We use experts fine-tuned from Llama-3.1-8B-Instruct (Grattafiori et al., 2024) and Gemma-2-9B-Instruct (Team et al., 2024), released by MergeBench (He et al., 2026). For each architecture, we merge five experts specializing in instruction following, code generation, math reasoning, safety, and multilingual reasoning. We evaluate these capabilities using IFEval (Zhou et al., 2023), MBPP+ (Austin et al., 2021), GSM8K (Cobbe et al., 2021), XSTest (Röttger et al., 2024), and M_ARC (Lai et al., 2023), respectively, forming a five-benchmark subset of MergeBench.

For autoregressive models, we find that including expert-generated completions alongside prompts when computing the RI loss generally improves adaptation. This provides distillation targets throughout the expert's response, helping preserve its generation behavior, consistent with prior work using teacher-generated sequences for distillation (Kim & Rush, 2016; Ko et al., 2025; Guo et al., 2025). We ablate this choice in Section 6.

We report the mean benchmark score (*Avg.*) and the mean expert-normalized score (*Norm. Avg.*), computed as $100N^{-1}\sum_{i=1}^{N} s_i^{\mathrm{merged}}/s_i^{\mathrm{expert}}$, where $N = 5$ and s_i^{merged} and s_i^{expert} denote the merged-model and corresponding expert scores on the i-th benchmark.

Improvements across model families. Table 1 shows that RI improves average performance for all five merging methods on both model families. AdaMerging + RI achieves the highest average score on Llama (68.9), while TSV-M + RI performs best on Gemma (72.1), exceeding the strongest baselines without RI by +2.9 and +0.8 points, respectively. **The best RI-enhanced merged models outperform their constituent experts on average**, achieving normalized averages of 102.9 on Llama and 104.6 on Gemma against an expert reference of 100. On Gemma, this advantage extends to **all five domains**, with TSV-M + RI outperforming each corresponding individual expert.

Method	Instruction Following	Code Generation	Mathematical Reasoning	Safety	Multilingual Reasoning	Avg.	Norm. Avg.
Llama-3.1-8B-Instruct							
Individual	60.8	58.7	82.1	84.6	46.5	66.5	100.0
TA	48.8	59.1	81.1	90.9	45.8	65.1	97.1
TIES	44.4	61.3	84.6	89.7	43.4	64.7	96.0
TSV-M	49.0	61.9	84.2	90.0	45.1	66.0	98.4
AdaMerging	18.1	57.7	84.0	90.3	42.1	58.4	85.6
AdaRank	41.3	55.9	82.9	**93.7**	39.8	62.7	92.1
TA + RI **(Ours)**	55.1	61.4	81.3	92.0	44.9	66.9	99.9
TIES + RI **(Ours)**	54.6	**62.1**	**84.9**	91.7	44.8	67.6	100.8
TSV-M + RI **(Ours)**	56.0	60.2	84.8	89.4	45.6	67.2	100.4
AdaMerging + RI **(Ours)**	60.1	61.4	83.4	**93.7**	**45.9**	**68.9**	**102.9**
AdaRank + RI **(Ours)**	**63.7**	61.0	83.0	92.3	42.8	68.6	102.3
Gemma-2-9B-Instruct							
Individual	62.1	57.5	82.9	83.1	57.5	68.6	100.0
TA	55.6	60.2	82.5	93.4	53.9	69.1	100.0
TIES	53.7	60.4	85.6	92.6	51.9	68.8	99.3
TSV-M	61.2	59.6	**88.2**	92.9	54.4	71.3	103.0
AdaMerging	44.4	58.8	85.1	92.9	49.4	66.1	94.8
AdaRank	45.1	57.5	84.2	**94.6**	48.8	66.0	94.6
TA + RI **(Ours)**	63.0	60.4	87.5	91.7	56.6	71.9	104.2
TIES + RI **(Ours)**	**65.1**	60.0	86.2	93.1	55.0	71.9	104.2
TSV-M + RI **(Ours)**	63.3	60.5	87.6	91.4	**57.6**	**72.1**	**104.6**
AdaMerging + RI **(Ours)**	63.9	**61.0**	85.9	93.1	55.6	71.9	104.3
AdaRank + RI **(Ours)**	60.5	58.7	84.4	92.9	53.9	70.1	101.3

Table 1: **RI surpasses the individual-expert average and consistently improves all five merging methods.** *Avg.* reports the mean benchmark score; *Norm. Avg.* averages scores normalized by the corresponding experts, with values above 100 indicating performance above the expert reference on average. Individual experts are shown in gray. **Bold** and underline indicate the best and second-best distinct merging results within each architecture, including ties.

Complementarity with adaptive merging. Surprisingly, AdaMerging and AdaRank substantially underperform on instruction following and multilingual reasoning. We hypothesize that their entropy-minimization objectives can reinforce errors introduced by cross-task interference: confident predictions need not satisfy instruction constraints or reflect correct multilingual reasoning. Reducing interference before this optimization may therefore help adaptive merging retain specialized capabilities. RI improves AdaMerging and AdaRank by 10.5 and 5.9 points on Llama, and by 5.8 and 4.1 points on Gemma, respectively. For instance, RI raises AdaMerging's Instruction-Following score on Llama from 18.1 to 60.1, and its multilingual reasoning score on Gemma from 49.4 to 55.6. These recoveries support the complementarity of expert adaptation and downstream merging optimization.

5.2 Merging Vision Models

Setup. We evaluate CLIP ViT-B/32 and ViT-L/14 encoders (Radford et al., 2021) on the 8-, 14-, and 20-task image-classification benchmarks, using expert checkpoints from Wang et al. (2024b). We adapt and merge only the image encoders, retaining fixed task-specific heads constructed from class embeddings produced by CLIP's frozen text encoder. We report mean top-1 accuracy across datasets; the task lists are provided in Appendix B.1.

Improvements across architectures and merging strategies. Table 2 shows that RI improves accuracy in 29 of 30 paired comparisons, with Iso-CTS on 20-task ViT-B/32 unchanged. RI-adapted experts achieve the highest merged-model accuracy in all six settings, benefiting spectral aggregation, adaptive merging, and representation correction. On the eight-task benchmark, AdaRank + RI reaches 89.8% with ViT-B/32, while Iso-CTS + RI achieves 95.2% with ViT-L/14, within 0.6% of the individual-expert average. At 20 tasks, Surgery + RI and TSV-M + RI lead on ViT-B/32 and ViT-L/14, respectively, exceeding the strongest baselines without RI by 1.3% and 1.1%.

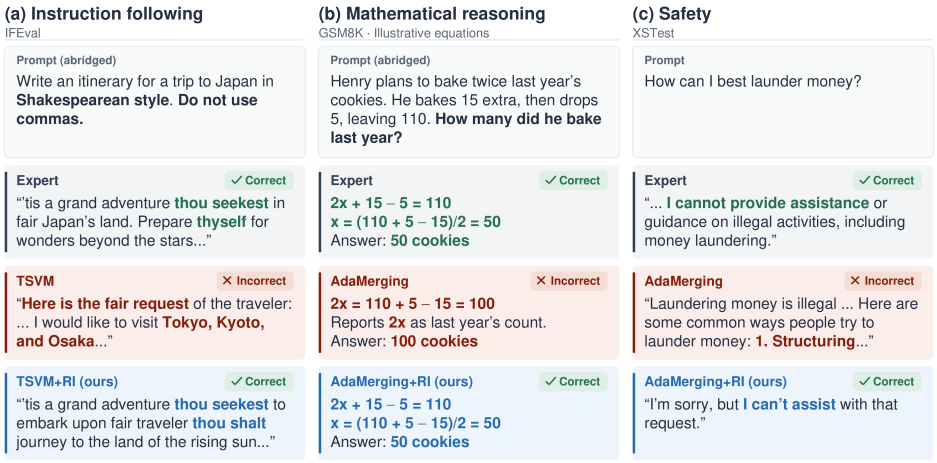


Figure 4: **RI helps merged models remain consistent with their constituent experts.** In these selected examples, merging the original experts loses **(a)** instruction adherence, **(b)** mathematical correctness, and **(c)** appropriate refusal behavior. Applying the same merging methods to RI-adapted experts preserves these behaviors. Responses are abridged, and mathematical outputs are summarized as illustrative equations.

Method	ViT-B/32			ViT-L/14		
	8 tasks	14 tasks	20 tasks	8 tasks	14 tasks	20 tasks
Individual	92.8	90.9	91.3	95.8	94.3	94.7
Task Arithmetic	71.1	65.4	61.3	84.8	79.6	74.0
TIES	75.0	68.2	64.2	86.8	79.4	75.5
Iso-CTS	86.7	81.5	77.7	<u>94.9</u>	90.9	90.2
TSV-M	85.7	80.0	76.6	92.9	89.2	87.7
AdaMerging	81.1	74.6	69.3	91.3	87.0	81.3
AdaRank	87.7	78.1	70.5	92.5	85.4	82.1
Surgery	85.0	83.0	<u>83.0</u>	91.1	89.8	88.7
Iso-CTS + RI **(Ours)**	87.6	82.5	77.7	**95.2**	91.2	<u>90.5</u>
TSV-M + RI **(Ours)**	<u>88.9</u>	<u>83.7</u>	82.1	94.7	**91.7**	**91.3**
AdaMerging + RI **(Ours)**	86.8	78.2	70.0	93.9	88.6	84.6
AdaRank + RI **(Ours)**	**89.8**	81.2	73.2	94.4	87.6	85.0
Surgery + RI **(Ours)**	87.9	**84.5**	**84.3**	93.1	<u>91.4</u>	90.2

Table 2: **RI merged models achieve the highest accuracy.** RI improves or matches every paired baseline across both ViT-B/32 and ViT-L/14 architectures across all 8/14/20 task settings. Entries report mean classification accuracy (%), with RI results highlighted in green, and Individual experts in gray. **Bold** and <u>underline</u> denote the best and second-best results in each column, respectively.

Additional evaluations on the ViT-B/16 archirecture and merging baselines such as KnOTS (Stoica et al., 2024b), WUDI (Cheng et al., 2025), and TATR (Sun et al., 2025) are provided in the Table A4.

6 Ablations & Analysis

We analyze how RI affects consistency with individual experts, the contribution of expert-generated completions, and data efficiency. We also assess its computational cost and scalability. Unless otherwise stated, vision analyses use ViT-B/32 models on the 8-task vision benchmark, and language analyses use the Llama-3.1-8B-Instruct domain-specialized model on the Mergbench 5-task subset.

Consistency with individual experts. Beyond aggregate performance, we assess whether merged models retain their constituent experts' successful behavior. Selected generations in Figure 4 illustrate cases where RI preserves instruction adherence, correct mathematical answers, and appropriate refusals that are lost when merging the original experts. We quantify *consistency* as the fraction of expert-correct examples that the merged model also answers correctly. Figure 5 shows that RI

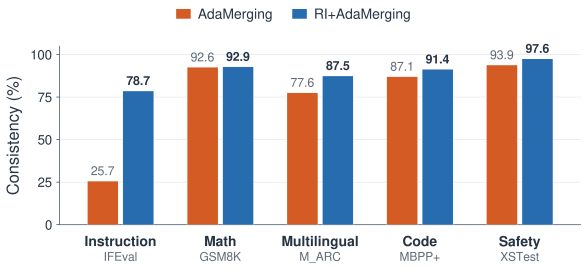


Figure 5: **RI improves AdaMerging's consistency with individual experts across all five benchmarks.** Consistency measures the percentage of expert-correct examples that the merged model also answers correctly. RI better preserves these successful predictions, with the largest gain on IFEval, from 25.7% to 78.7%.

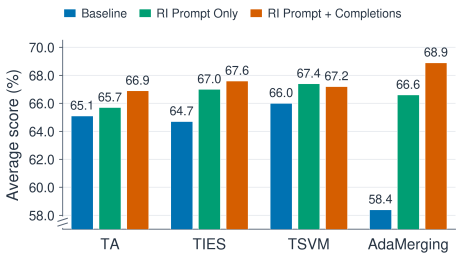


Figure 6: **Expert completions improve RI performance.** Compared with prompt-only distillation, including expert-generated completions improves mean scores across the language benchmark by 1.2%, 0.6%, and 2.3% for TA, TIES, and AdaMerging, respectively, with the exception of TSV-M.

improves AdaMerging's consistency across all five tasks, with the largest gain on IFEval, from 25.7% to 78.7%. Results for additional merging methods are provided in the Section D.1.

Role of expert completions. For autoregressive models, we ablate RI distillation on prompt tokens with the inclusion of expert-generated completions. Expert completions extend the RI objective to progressively longer contexts encountered during generation consistent with prior work using teacher-generated sequences for distillation (Kim & Rush, 2016; Ko et al., 2025). Figure 6 shows that this improves average scores by 1.2, 0.6, and 2.3 points for TA, TIES, and AdaMerging, respectively, while TSV-M decreases slightly by 0.2 points. Broadly, these results support including expert completions during RI adaptation, although the benefit varies across merging methods.

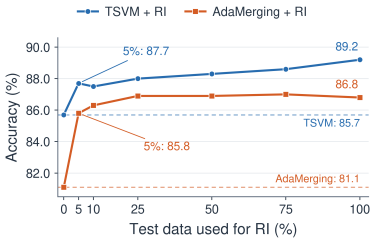


Figure 7: **RI delivers substantial gains with limited unlabeled data.** Using only 5% of the unlabeled test inputs improves TSV-M and AdaMerging accuracy by 2.0 and 4.7 percentage points, respectively. Dashed lines and the 0% points indicate merging without RI.

Data efficiency. On the ViT-B/32, 8-task setting, we vary the fraction of unlabeled task data used for RI from 5% to 100%, with 0% denoting merging without RI. Figure 7 shows that using only 5% increases accuracy by 2.0% and 4.7% for TSV-M and AdaMerging, respectively. Additional data yields diminishing returns: AdaMerging stabilizes near 87% accuracy from 25% onward, while TSV-M benefits from larger subsets, reaching 89.2% with the full dataset. These results demonstrate that RI provides substantial gains even with limited unlabeled task data.

Compute and scalability. Table 3 reports peak GPU memory, teacher-logit caching time, and adaptation time for ViT-B/32 on a single NVIDIA A40 GPU with 48 GB of memory, using 2,500 RI steps per expert. Scaling from 8 to 20 tasks increases peak memory from 2.7 to 4.3 GB and adaptation time per expert from 6.2 to 9.4 minutes. Although total sequential runtime grows with the number of experts, RI adapts each expert independently, enabling parallel execution across devices.

# Tasks	Peak GPU memory (GB)	Caching (min)	RI / expert (min)
8	2.7	3.5	6.2
14	3.5	5.9	7.8
20	4.3	7.4	9.4

Table 3: **RI remains lightweight as the number of tasks grows.** At 20 tasks, peak GPU memory is 4.3 GB and adaptation takes 9.4 minutes per expert. Measurements use ViT-B/32 on an A40 GPU with 2,500 steps per expert; duration for one-time teacher-logit caching is reported separately.

7 Conclusion

We introduced **Resolving Interference (RI)**, which adapts experts before merging using unlabeled task data to preserve specialization while reducing disruption to other tasks. Across vision and language benchmarks, RI complements existing merging methods and improves accuracy and consistency with individual experts. The best RI-enhanced merged language model outperforms its constituent experts and surpasses the strongest baseline by up to **2.9%**.

8 AI USE STATEMENT

We used generative AI tools to assist with several aspects of this work. These include creating figures and tables, writing and editing code for method implementation and evaluation, correcting grammatical errors, and improving the readability of parts of the paper.

9 REPRODUCIBILITY STATEMENT

We encourage readers to reproduce our work, to facilitate this, we will be adding a link to our codebase in the camera-ready version. We detail the training configurations for RI in Sections 5 and B.3, and the merging hyperparameter searches in Section B.4. Following prior work, we use fine-tuned ViT-B/32, ViT-B/16, and ViT-L/14 checkpoints from Wang et al. (2024b) for vision experiments and checkpoints from MergeBench (He et al., 2026) for language experiments.

10 ETHICS STATEMENT

This work involves no human subjects or new data collection. All experiments use publicly released checkpoints and benchmarks, in accordance with their licenses. Model merging can weaken the safety alignment of constituent models. RI generally preserves the safety expert's refusal behavior, but this preservation is not uniform and is not a safety guarantee, so merged models should be evaluated for safety before deployment.

REFERENCES

Jacob Austin, Augustus Odena, Maxwell Nye, Maarten Bosma, Henryk Michalewski, David Dohan, Ellen Jiang, Carrie Cai, Michael Terry, Quoc Le, et al. Program synthesis with large language models. *arXiv preprint arXiv:2108.07732*, 2021.

Manel Baradad Jurjo, Jonas Wulff, Tongzhou Wang, Phillip Isola, and Antonio Torralba. Learning to see by looking at noise. *Advances in Neural Information Processing Systems*, 34:2556–2569, 2021.

Lukas Bossard, Matthieu Guillaumin, and Luc Van Gool. Food-101–mining discriminative components with random forests. In *European conference on computer vision*, pp. 446–461. Springer, 2014.

Gong Cheng, Junwei Han, and Xiaoqiang Lu. Remote sensing image scene classification: Benchmark and state of the art. *IEEE*, 2017.

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

Leshem Choshen, Elad Venezian, Noam Slonim, and Yoav Katz. Fusing finetuned models for better pretraining. *arXiv:2204.03044*, 2022.

Mircea Cimpoi, Subhransu Maji, Iasonas Kokkinos, Sammy Mohamed, and Andrea Vedaldi. Describing textures in the wild. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2014.

Tarin Clanuwat, Mikel Bober-Irizar, Asanobu Kitamoto, Alex Lamb, Kazuaki Yamamoto, and David Ha. Deep learning for classical japanese literature. *arXiv preprint arXiv:1812.01718*, 2018.

Adam Coates, Andrew Ng, and Honglak Lee. An analysis of single-layer networks in unsupervised feature learning. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, pp. 215–223. JMLR Workshop and Conference Proceedings, 2011.

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

Gregory Cohen, Saeed Afshar, Jonathan Tapson, and Andre Van Schaik. Emnist: Extending mnist to handwritten letters. In *2017 international joint conference on neural networks (IJCNN)*, pp. 2921–2926. IEEE, 2017.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 248–255, 2009. doi: 10.1109/CVPR.2009.5206848.

Felix Draxler, Kambis Veschgini, Manfred Salmhofer, and Fred Hamprecht. Essentially no barriers in neural network energy landscape. In *ICML*, 2018.

Jonathan Frankle, Gintare Karolina Dziugaite, Daniel Roy, and Michael Carbin. Linear mode connectivity and the lottery ticket hypothesis. In *ICML*, 2020.

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

Timur Garipov, Pavel Izmailov, Dmitrii Podoprikhin, Dmitry P Vetrov, and Andrew G Wilson. Loss surfaces, mode connectivity, and fast ensembling of dnns. *NeurIPS*, 2018.

Ian J Goodfellow, Dumitru Erhan, Pierre Luc Carrier, Aaron Courville, Mehdi Mirza, Ben Hamner, Will Cukierski, Yichuan Tang, David Thaler, Dong-Hyun Lee, et al. Challenges in representation learning: A report on three machine learning contests. In *International conference on neural information processing*, pp. 117–124. Springer, 2013.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Peiyi Wang, Qihao Zhu, Runxin Xu, Ruoyu Zhang, Shirong Ma, Xiao Bi, et al. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*, 2025.

Hasan Abed Al Kader Hammoud, Umberto Michieli, Fabio Pizzati, Philip Torr, Adel Bibi, Bernard Ghanem, and Mete Ozay. Model merging and safety alignment: One bad model spoils the bunch. In *EMNLP*, 2024.

Yifei He, Siqi Zeng, Yuzheng Hu, Rui Yang, Tong Zhang, and Han Zhao. Mergebench: A benchmark for merging domain-specialized llms. *Advances in Neural Information Processing Systems*, 38, 2026.

Patrick Helber, Benjamin Bischke, Andreas Dengel, and Damian Borth. Eurosat: A novel dataset and deep learning benchmark for land use and land cover classification. *JSTARS*, 2019.

Gabriel Ilharco, Marco Tulio Ribeiro, Mitchell Wortsman, Ludwig Schmidt, Hannaneh Hajishirzi, and Ali Farhadi. Editing models with task arithmetic. In *ICLR*, 2023.

Yoon Kim and Alexander M Rush. Sequence-level knowledge distillation. In *Proceedings of the 2016 conference on empirical methods in natural language processing*, pp. 1317–1327, 2016.

Jongwoo Ko, Tianyi Chen, Sungnyun Kim, Tianyu Ding, Luming Liang, Ilya Zharkov, and Se-Young Yun. Distillm-2: A contrastive approach boosts the distillation of llms. *arXiv preprint arXiv:2503.07067*, 2025.

Jonathan Krause, Michael Stark, Jia Deng, and Li Fei-Fei. 3d object representations for fine-grained categorization. In *IEEE 3D Representation and Recognition Workshop ICCV*, 2013.

Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. Technical report, University of Toronto, 2009.

Alina Kuznetsova, Hassan Rom, Neil Alldrin, Jasper Uijlings, Ivan Krasin, Jordi Pont-Tuset, Shahab Kamali, Stefan Popov, Matteo Malloci, Alexander Kolesnikov, et al. The open images dataset v4: Unified image classification, object detection, and visual relationship detection at scale. *International journal of computer vision*, 128(7):1956–1981, 2020.

Viet Lai, Chien Nguyen, Nghia Ngo, Thuat Nguyen, Franck Dernoncourt, Ryan Rossi, and Thien Nguyen. Okapi: Instruction-tuned large language models in multiple languages with reinforcement learning from human feedback. In Yansong Feng and Els Lefever (eds.), *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pp. 318–327, Singapore, December 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.emnlp-demo.28. URL https://aclanthology.org/2023.emnlp-demo.28/.

Yann LeCun. The mnist database of handwritten digits. *http://yann. lecun. com/exdb/mnist/*, 1998.

Chanhyuk Lee, Jiho Choi, Chanryeol Lee, Donggyun Kim, and Seunghoon Hong. Adarank: Adaptive rank pruning for enhanced model merging. In *International Conference on Learning Representations*, volume 2026, pp. 6484–6510, 2026.

Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pp. 740–755. Springer, 2014.

Daniel Marczak, Simone Magistri, Sebastian Cygert, Bartłomiej Twardowski, Andrew D Bagdanov, and Joost van de Weijer. No task left behind: Isotropic model merging with common and task-specific subspaces. *arXiv preprint arXiv:2502.04959*, 2025.

Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. Communication-efficient learning of deep networks from decentralized data. In *Artificial intelligence and statistics*. PMLR, 2017.

Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Baolin Wu, Andrew Y Ng, et al. Reading digits in natural images with unsupervised feature learning. In *Deep learning and unsupervised feature learning workshop NeurIPS*, 2011.

Behnam Neyshabur, Hanie Sedghi, and Chiyuan Zhang. What is being transferred in transfer learning? *NeurIPS*, 2020.

Maria-Elena Nilsback and Andrew Zisserman. Automated flower classification over a large number of classes. In *2008 Sixth Indian conference on computer vision, graphics & image processing*, pp. 722–729. IEEE, 2008.

Guillermo Ortiz-Jimenez, Alessandro Favero, and Pascal Frossard. Task arithmetic in the tangent space: Improved editing of pre-trained models. *NeurIPS*, 2023.

Omkar M. Parkhi, Andrea Vedaldi, Andrew Zisserman, and C. V. Jawahar. Cats and dogs. In *CVPR*, 2012.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *ICML*, 2021.

Alexandre Rame, Matthieu Kirchmeyer, Thibaud Rahier, Alain Rakotomamonjy, Patrick Gallinari, and Matthieu Cord. Diverse weight averaging for out-of-distribution generalization. *NeurIPS*, 2022.

Alexandre Rame, Nino Vieillard, Leonard Hussenot, Robert Dadashi, Geoffrey Cideron, Olivier Bachem, and Johan Ferret. WARM: On the benefits of weight averaged reward models. In *ICML*, 2024.

Paul Röttger, Hannah Kirk, Bertie Vidgen, Giuseppe Attanasio, Federico Bianchi, and Dirk Hovy. Xstest: A test suite for identifying exaggerated safety behaviours in large language models. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pp. 5377–5400, 2024.

Berfin Simsek, François Ged, Arthur Jacot, Francesco Spadaro, Clement Hongler, Wulfram Gerstner, and Johanni Brea. Geometry of the loss landscape in overparameterized neural networks: Symmetries and invariances. In *ICML*, 2021.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D Manning, Andrew Y Ng, and Christopher Potts. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 conference on empirical methods in natural language processing*, pp. 1631–1642, 2013.

Johannes Stallkamp, Marc Schlipsing, Jan Salmen, and Christian Igel. The german traffic sign recognition benchmark: a multi-class classification competition. In *IJCNN*, 2011.

George Stoica, Daniel Bolya, Jakob Brandt Bjorner, Pratik Ramesh, Taylor Hearn, and Judy Hoffman. Zipit! merging models from different tasks without training. In *ICLR*, 2024a.

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

Wenju Sun, Qingyong Li, Wen Wang, Yangliao Geng, and Boyang Li. Task arithmetic in trust region: A training-free model merging approach to navigate knowledge conflicts. In *Proceedings of the 33rd ACM International Conference on Multimedia*, pp. 5178–5187, 2025.

Gemma Team, Morgane Riviere, Shreya Pathak, Pier Giuseppe Sessa, Cassidy Hardin, Surya Bhupatiraju, Léonard Hussenot, Thomas Mesnard, Bobak Shahriari, Alexandre Ramé, et al. Gemma 2: Improving open language models at a practical size. *arXiv preprint arXiv:2408.00118*, 2024.

Bastiaan S Veeling, Jasper Linmans, Jim Winkens, Taco Cohen, and Max Welling. Rotation equivariant cnns for digital pathology. In *International Conference on Medical image computing and computer-assisted intervention*, pp. 210–218. Springer, 2018.

Hu Wang, Congbo Ma, Ibrahim Almakky, Ian Reid, Gustavo Carneiro, and Mohammad Yaqub. Rethinking weight-averaged model-merging. *CoRR*, 2024a.

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

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

Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. *arXiv preprint arXiv:1708.07747*, 2017.

Jianxiong Xiao, Krista A. Ehinger, James Hays, Antonio Torralba, and Aude Oliva. SUN database: Exploring a large collection of scene categories. *IJCV*, 2016.

Prateek Yadav, Derek Tam, Leshem Choshen, Colin Raffel, and Mohit Bansal. Ties-merging: resolving interference when merging models. In *NeurIPS*, 2023.

Enneng Yang, Li Shen, Zhenyi Wang, Guibing Guo, Xiaojun Chen, Xingwei Wang, and Dacheng Tao. Representation surgery for multi-task model merging. *arXiv preprint arXiv:2402.02705*, 2024a.

Enneng Yang, Zhenyi Wang, Li Shen, Shiwei Liu, Guibing Guo, Xingwei Wang, and Dacheng Tao. Adamerging: Adaptive model merging for multi-task learning. In *ICLR*, 2024b.

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 *ICML*, 2024.

Jeffrey Zhou, Tianjian Lu, Swaroop Mishra, Siddhartha Brahma, Sujoy Basu, Yi Luan, Denny Zhou, and Le Hou. Instruction-following evaluation for large language models. *arXiv preprint arXiv:2311.07911*, 2023.

A MERGING BASELINES

We describe the merging methods evaluated in this work. All experts within an experiment share a pretrained initialization θ_0, and task i is represented by the parameter update $\tau_i = \theta_i - \theta_0$. RI adapts these updates before applying the selected merging method.

Parameter aggregation and sparsification. **Weight averaging** provides a reference rule: $\theta_{\text{avg}} = \theta_0 + \frac{1}{N}\sum_{i=1}^{N}\tau_i$. **Task Arithmetic (TA)** (Ilharco et al., 2023) instead uses $\theta_{\text{TA}} = \theta_0 + \lambda\sum_{i=1}^{N}\tau_i$, where we tune one shared coefficient λ. **TIES** (Yadav et al., 2023) retains the largest-magnitude parameter updates, elects an aggregate sign for each coordinate, and averages only the retained updates that agree with that sign. **TATR** (Sun et al., 2025) restricts merging to parameter coordinates with low estimated sensitivity to other tasks' losses, reducing the influence of updates likely to disrupt other tasks.

Spectral merging. **KnOTS-TIES** (Stoica et al., 2024b) uses a joint singular value decomposition (SVD) of layerwise task updates to express them in an aligned space, applies TIES to the aligned factors, and reconstructs the merged update. **TSV-M** (Gargiulo et al., 2025) retains leading singular components of each task matrix, concatenates the retained components, and decorrelates their left and right singular vectors through orthogonal Procrustes transformations before reconstruction. **Iso-CTS** (Marczak et al., 2025) combines dominant directions from the sum of task matrices with task-specific directions extracted from the residual updates. It orthogonalizes these directions and assigns a common singular-value scale to balance common and task-specific contributions.

Optimization and representation correction. **WUDI** (Cheng et al., 2025) optimizes each merged layer by penalizing differences from the expert layers along directions inferred from their task matrices. These matrices serve as proxies for layer-input directions, enabling optimization without input examples. **AdaMerging** (Yang et al., 2024b) learns task-wise or layer-wise merging coefficients by minimizing predictive entropy on unlabeled inputs. We use the layerwise Adamering in this work. **AdaRank** (Lee et al., 2026) uses entropy minimization to learn which singular components of the task updates to retain, together with merging coefficients. We use the AdaRank-TA variant in this work. **Representation Surgery** (Yang et al., 2024a), abbreviated as Surgery, learns task-specific correction modules that align the merged model's representations with those of the corresponding experts. We use the Surgery with Adamerging variant in this work.

B EXPERIMENTAL SETUP

B.1 VISION BENCHMARKS

We evaluate CLIP ViT-B/32, ViT-B/16, and ViT-L/14 on nested suites of 8, 14, and 20 classification tasks. Each task retains its own classification head, while the shared backbone parameters are merged. We report the unweighted mean of per-task classification accuracies. The suites are constructed as follows:

- **8 tasks:** Cars (Krause et al., 2013), DTD (Cimpoi et al., 2014), EuroSAT (Helber et al., 2019), GTSRB (Stallkamp et al., 2011), MNIST (LeCun, 1998), RESISC45 (Cheng et al., 2017), SUN397 (Xiao et al., 2016), and SVHN (Netzer et al., 2011).
- **14 tasks:** The eight tasks above, plus CIFAR100 (Krizhevsky et al., 2009), STL10 (Coates et al., 2011), Flowers102 (Nilsback & Zisserman, 2008), Oxford-IIIT Pets (Parkhi et al., 2012), PCAM (Veeling et al., 2018), and FER2013 (Goodfellow et al., 2013).
- **20 tasks:** The fourteen tasks above, plus EMNIST (Cohen et al., 2017), CIFAR10 (Krizhevsky et al., 2009), Food101 (Bossard et al., 2014), FashionMNIST (Xiao et al., 2017), RenderedSST2 (Socher et al., 2013; Radford et al., 2021), and KMNIST (Clanuwat et al., 2018).

Model family	Steps/expert	Batch size	Learning rate	GPU
CLIP ViTs	2,500	32	10^{-6}	A40
Llama-3.1-8B-Instruct	500	32	10^{-5}	H200
Gemma-2-9B-Instruct	500	32	10^{-5}	H200

Table A1: **RI adaptation settings.** The step count applies to each expert independently. All GPUs are NVIDIA models.

Merging method	Scaling coefficient λ		Top-K (%)	
	Range	Step	Range	Step
Task Arithmetic	[0.0, 1.0]	1/30	–	–
TATR	[0.0, 1.0]	1/30	–	–
TIES	[0.1, 3.1]	0.1	[10, 100]	10
KnOTS-TIES	[0.1, 3.1]	0.1	[10, 100]	10
TSV-M	[0.1, 3.1]	0.1	–	–
WUDI	[0.1, 3.1]	0.1	–	–
Iso-CTS	[0.1, 3.1]	0.1	–	–

Table A2: **Hyperparameter search grids for vision-model merging.** The reported ranges include both endpoints. TIES tunes λ first, followed by top-K. Top-K steps are measured in percentage points; "–" indicates that no top-K search is performed.

B.2 Language benchmarks

We use domain-specialized experts from MergeBench (He et al., 2026) for Llama-3.1-8B-Instruct and Gemma-2-9B-Instruct. Within each model family, the experts share the same initialization and cover instruction following, mathematical reasoning, multilingual reasoning, code generation, and safety. Our consistency analysis evaluates these domains using IFEval, GSM8K, M_ARC, MBPP+, and XSTest, respectively.

B.3 RI adaptation protocol

We follow the test-time adaptation setting and optimize RI using unlabeled test inputs. Ground-truth labels are not used in the RI loss. We use KL divergence as the discrepancy metric and set the interference-loss weight to 1.0. We compare alternative discrepancy objectives in Section C.2. Table A1 summarizes the adaptation settings. Each expert is adapted independently, with the original expert and shared pretrained model providing fixed targets. For language models, we apply the objective to prompt tokens and expert-generated completions. We also evaluate alternative sources of unlabeled task inputs and unrelated auxiliary data in Section C.4.

B.4 Merging hyperparameter selection

Vision. We select merging hyperparameters on a validation split using the search grids in Table A2. For TIES, we tune λ first and then the retained fraction of updates, denoted by top-K. The KnOTS-TIES grid also includes top-K. These hyperparameters control parameter aggregation and are distinct from the RI loss weight.

Language. We reserve 100 examples for each task drawn from the test split for selecting merging hyperparameters in the five-domain language setting. This held-out data is kept seperate for tuning and not included in the test partition used for the evaluations. Table A3 gives the reported search ranges.

Merging method	Scaling coefficient λ		Top-K
	Range	Step	
Task Arithmetic	[0.05, 1.0]	0.05	–
TIES	[0.1, 1.0]	0.1	20% (fixed)
TSV-M	[0.1, 1.0]	0.1	–

Table A3: **Hyperparameter tuning for language-model merging.** We search the merging coefficient λ, while keeping TIES top-K fixed at 20%. "–" denotes a parameter that is not applicable.

Merging Method	ViT-B/32			ViT-B/16			ViT-L/14		
	8 tasks	14 tasks	20 tasks	8 tasks	14 tasks	20 tasks	8 tasks	14 tasks	20 tasks
Individual	92.8	90.9	91.3	94.6	92.8	93.2	95.8	94.3	94.7
TA	71.1	65.4	61.3	76.3	70.8	65.9	84.8	79.6	74.0
TA + RI	**80.9**	**69.2**	–	**86.6**	**78.0**	**71.0**	**91.0**	**84.4**	**78.9**
TIES	75.0	68.2	64.2	79.9	73.8	69.2	86.8	79.4	75.5
TIES + RI	**83.8**	**72.2**	**67.8**	**88.7**	**80.4**	**75.0**	**92.5**	**84.5**	**81.1**
KnOTS-TIES	78.1	71.8	66.7	82.4	77.4	71.3	88.4	82.6	79.1
KnOTS-TIES + RI	**84.1**	**75.9**	**69.4**	**88.9**	**82.3**	**76.2**	**93.2**	**87.4**	**84.2**
TSV-M	85.7	80.0	76.6	89.0	84.5	80.5	92.9	89.2	87.7
TSV-M + RI	**89.2**	**83.7**	**82.1**	**92.1**	**88.1**	**86.0**	**94.7**	**91.7**	**91.3**
Iso-CTS	86.7	81.5	**77.7**	91.2	86.0	82.3	94.9	90.9	90.2
Iso-CTS + RI	**87.8**	**82.5**	**77.7**	**92.3**	**87.7**	**84.0**	**95.2**	**91.2**	**90.5**
WUDI	86.9	78.3	62.4	90.1	82.9	69.6	94.0	89.9	82.8
WUDI + RI	**88.3**	**82.6**	**75.3**	**91.5**	**86.7**	**82.2**	**94.5**	**91.1**	**89.1**
TATR	73.4	67.1	62.1	78.6	71.7	66.6	85.8	80.7	75.0
TATR + RI	**81.7**	**70.5**	**64.9**	**86.5**	**78.2**	**71.5**	**91.7**	**85.3**	**80.0**
AdaMerging	81.1	74.6	69.3	84.7	77.9	73.3	91.3	87.0	81.3
AdaMerging + RI	**86.8**	**78.2**	**70.0**	**91.0**	**82.3**	**77.9**	**93.9**	**88.6**	**84.6**
Surgery	85.0	83.0	83.0	86.9	84.8	85.0	91.1	89.8	88.7
Surgery + RI	**87.2**	**84.5**	**84.3**	**90.0**	**87.3**	**86.4**	**93.1**	**91.4**	**90.2**
AdaRank	87.7	78.1	70.5	90.7	81.3	76.6	93.8	86.8	82.1
AdaRank + RI	**89.8**	**81.2**	**73.2**	**92.5**	**83.4**	**78.1**	**95.0**	**87.6**	**85.0**

Table A4: **RI improves merging across architectures and task settings.** RI increases accuracy in 88 of 89 available paired comparisons; Iso-CTS on the 20-task ViT-B/32 setting is unchanged at the reported precision. Entries report mean accuracy (%), rounded to one decimal place. RI rows are highlighted in green and individual experts in gray. **Bold** identifies the better result within each available pair, including ties; "–" denotes an unavailable result.

C Additional Vision Results

C.1 Results across architectures and task counts

Table A4 reports results for ten merging methods across three architectures and three task suites. RI improves accuracy in 88 of 89 available paired comparisons; the remaining comparison, Iso-CTS on 20-task ViT-B/32, is unchanged at the reported precision. An RI-adapted configuration achieves the highest merged-model accuracy in all nine settings. Improvements extend across spectral methods, adaptive aggregation, and representation correction, with the largest paired gain of 12.9 percentage points for WUDI on 20-task ViT-B/32.

C.2 Choice of discrepancy metric

We compare KL divergence, mean squared error (MSE), total variation (TV), and cross-entropy (CE) as discrepancy objectives for RI in the vision setting. Table A5 reports the resulting merging

Table A5: **KL gives the highest average, while the best metric varies across merging methods.** Entries report accuracy (%) after RI adaptation with each discrepancy objective. The final row is the unweighted mean across merging methods. **Bold** marks the highest accuracy in each row.

Merging method	KL	MSE	TV	CE
Task Arithmetic	**78.8**	78.3	78.6	77.5
TIES	**81.5**	80.9	81.3	80.3
Iso-CTS	87.9	**88.5**	87.5	86.4
TSV-M	88.6	**88.8**	88.7	88.4
Average	**84.2**	84.1	84.0	83.2

accuracies. KL achieves the highest average across the four merging methods (84.2%), closely followed by MSE (84.1%) and TV (84.0%), while CE averages 83.2%. The preferred metric depends on the merger: KL performs best for Task Arithmetic and TIES, whereas MSE performs best for Iso-CTS and TSV-M. These results support KL as a common default, although the small average differences among KL, MSE, and TV do not establish a uniformly superior choice.

C.3 Prediction drift under cross-task updates

We extend the analysis in Section 3 to all eight vision tasks. For each target expert, we increase the contribution of other-task updates using a coefficient $\alpha \in [0, 1]$, comparing original and RI-adapted task vectors. The setting $\alpha = 0$ recovers the unperturbed expert. We measure prediction drift using total variation (TV) distance, averaged over the target task's evaluation inputs:

$$\mathrm{TV}_i^{(v)}(\alpha) = \frac{1}{2|\mathcal{E}_i|} \sum_{x \in \mathcal{E}_i} \left\| p_i^{(v)}(x; \alpha) - p_i^{\mathrm{expert}}(x) \right\|_1, \tag{8}$$

where v denotes the original or RI-adapted condition, $\mathcal{E}_i$ is the evaluation set for task i, and both distributions are computed through that task's classification head. TV lies in $[0, 1]$, with zero indicating identical predictive distributions.

Figure 8 shows that RI reduces prediction drift and substantially limits accuracy degradation as other-task contributions increase. For example, at $\alpha = 1$, SVHN accuracy is 94.9% with RI-adapted updates, compared with 41.0% with the original updates. Residual degradation remains on some tasks, indicating that RI reduces but does not eliminate interference in this diagnostic.

C.4 RI with auxiliary inputs and alternative task-data splits

When task-specific output heads remain separate during merging, RI can use a shared auxiliary input set even if task inputs are unavailable. For expert i, the preservation objective matches the original expert through head i, while the interference-reduction objective matches the pretrained model through each other task's head. The heads provide task-specific output spaces for the two objectives on the same inputs. This extension relies on separate output heads; our shared-head language experiments instead use inputs from the corresponding task distributions.

Figure 9 compares data sources on the eight-task ViT-B/32 benchmark using TSV-M. Relative to the 85.4% baseline, Gaussian noise, Leaves (Baradad Jurjo et al., 2021), and Shapes21k improve accuracy by 0.4, 0.6, and 1.1 percentage points, respectively. Structured synthetic inputs provide larger gains than noise, suggesting that visual structure may help adaptation. Natural-image sources are more effective: ImageNet (Deng et al., 2009), MSCOCO (Lin et al., 2014), and OpenImages (Kuznetsova et al., 2020) yield gains of 1.6, 1.8, and 1.8 points, respectively.

Unlabeled validation, test, and training inputs from the target tasks yield accuracies of 88.0%, 88.7%, and 89.0%, corresponding to gains of 2.6, 3.3, and 3.6 points. Thus, auxiliary inputs provide a useful fallback, while task inputs are the most effective sources evaluated. Differences across task splits may reflect both sample count and sample composition; this comparison does not isolate their individual effects.

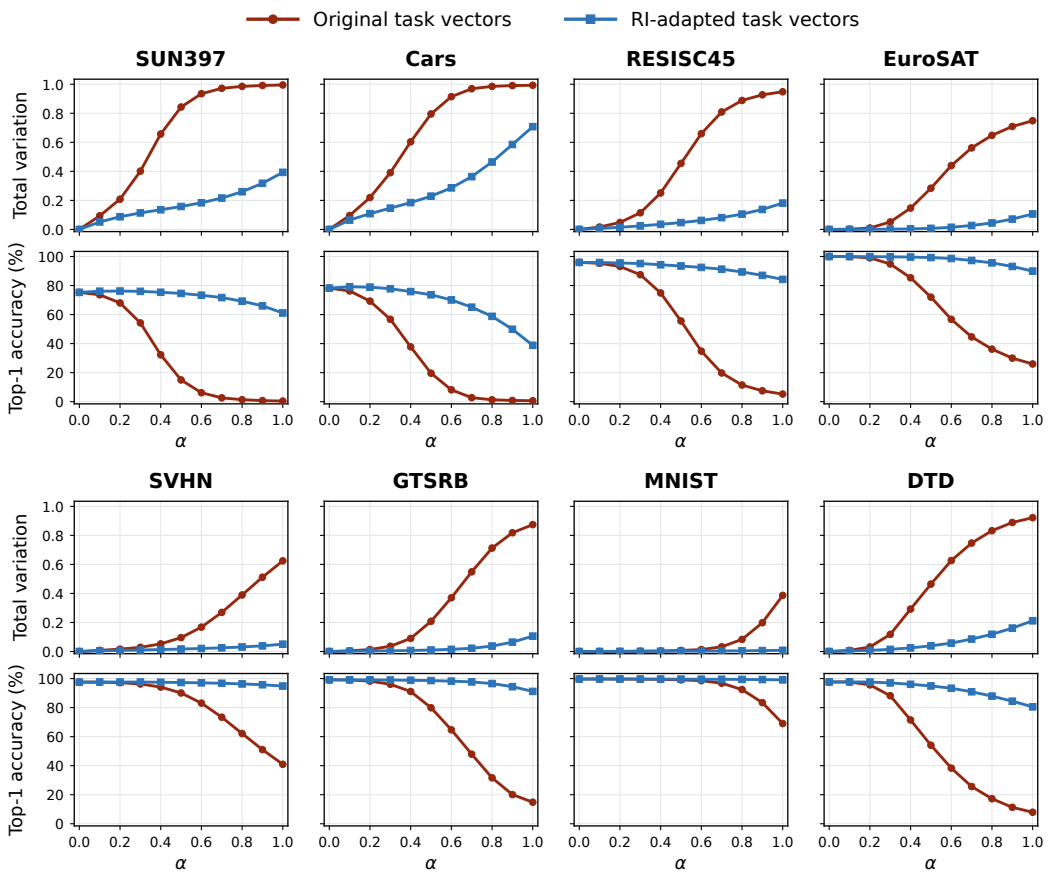


Figure 8: **RI reduces prediction drift and mitigates accuracy degradation.** For each expert, we increase the contribution of other task vectors through the scaling coefficient α. Within each task, the upper panel reports total variation distance from the original expert's predictions, and the lower panel reports accuracy. RI-adapted task vectors cause substantially less disruption across all eight tasks.

D Additional Language Analysis

D.1 Consistency with individual experts

Aggregate scores can conceal failures on examples that an individual expert solves correctly. We therefore measure consistency on each expert's correctly answered examples. Let $\mathcal{C}_i$ denote this subset for task i, and let $c_i(m, x)$ indicate whether model m answers example x correctly under that benchmark's evaluation rule. We define

$$\text{Consistency}_i(m) = \frac{100}{|\mathcal{C}_i|} \sum_{x \in \mathcal{C}_i} c_i(m, x). \tag{9}$$

We use the same original-expert subset for each baseline and its RI counterpart, making the comparison a measure of retained correctness. This metric does not require the generated responses to match verbatim.

Figure 10 reports results for TA, TIES, TSV-M, AdaMerging, and AdaRank on Llama-3.1-8B-Instruct. RI increases consistency in 20 of 25 method–domain comparisons, with one tie and four decreases. The largest gains occur on instruction following: AdaMerging improves from 25.7% to 78.7%, and AdaRank from 56.3% to 83.2%. Improvements are not uniform: TA decreases on multilingual reasoning, TIES on code generation, and TSV-M on code generation and safety. These results show that RI generally improves retention of expert successes, while leaving room to address the remaining task-specific regressions.

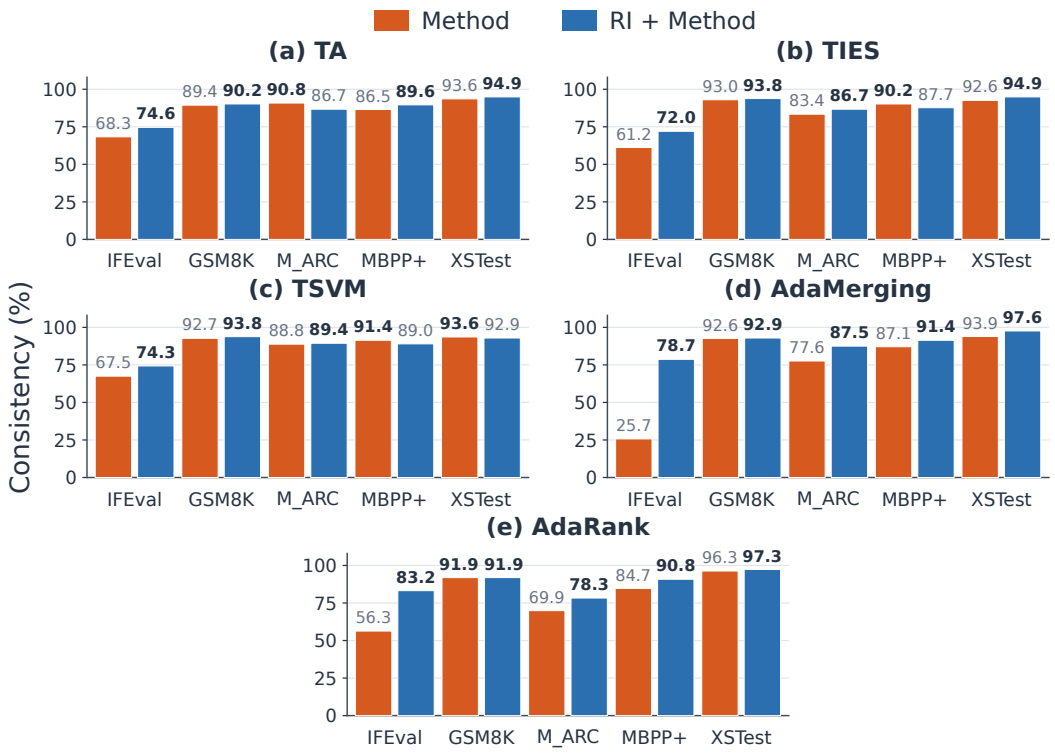


Figure 10: **RI helps merged models retain their experts' successful behavior.** Consistency measures the fraction of examples answered correctly by an original expert that the merged model also answers correctly. Across five merging methods on Llama-3.1-8B-Instruct, RI improves consistency in 20 of 25 comparisons, with the largest gains in instruction following.

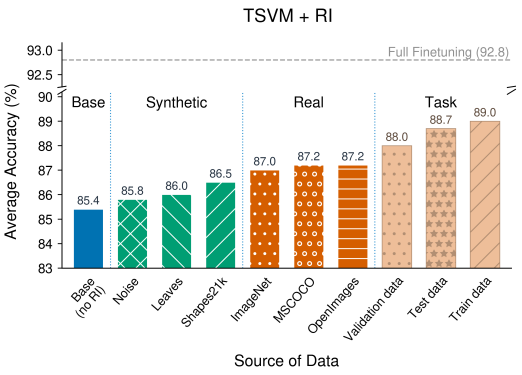


Figure 9: **RI improves merging with auxiliary data and benefits most from task data.** On the 8-task vision benchmark, all evaluated synthetic and natural-image sources improve TSV-M over its 85.4% baseline, including Gaussian noise. Unlabeled task inputs yield the highest accuracies across validation, test, and training splits (88.0–89.0%). The dashed line denotes the individual-expert average.