CONFLICT-GUIDED DATA-FREE MODEL MERGING

Anonymous authors
Paper under double-blind review

ABSTRACT

Data-free model merging fuses several fine-tuned models into a single network. Recently proposed data-free methods estimate task statistics from model weights and apply regularization to reduce the interference between the fine-tuned specialized models. However, these approaches typically apply uniform regularization strength across layers, even though expert disagreement varies across the network. We introduce *conflict-guided model merging*, a data-free model merging scheme that assigns regularization strength to each layer based on a conflict score derived from the task vectors. This score quantifies how well a single shared update can accommodate competing experts. The conflict score guides both Ridge regularization and an orthogonalized iterative solver, while maintaining the existing task-statistics estimator. Experimental results show that regularization alone improves merging accuracy, and further gains come from adapting regularization strength to layer-specific conflict. Conflict-guided Ridge improves accuracy over ACTMat on CLIP ViT-B/32, with statistically significant gains when merging 14 and 20 tasks. For input-adaptive merging, selective storage has accuracy close to ours MASS baseline with 50% of its per-task factor storage.

1 INTRODUCTION

Model merging combines multiple fine-tuned networks into a single model without additional retraining. In static merging, identical merged weights are applied to all inputs, while dynamic merging selects or combines parameter subsets based on the input. A central challenge is interference among the fine-tuned models, as weight modifications that enhance performance on one task may conflict with those required for another. Therefore, model merging requires careful balancing of the experts' contributions, which becomes especially difficult when task data are unavailable, and merging must rely exclusively on model weights.

RegMean (Jin et al., 2023) addresses interference by identifying merged weights whose layer outputs closely approximate those of each expert. This approach requires input covariances estimated from task data. ACTMat (Gamal Abdel Hameed et al., 2026) removes this dependency by estimating covariances from task vectors, which represent the weight changes introduced during fine-tuning. However, minimizing a loss function based on estimated statistics does not always yield the most accurate merged model. Recent methods employ regularization or iterative refinement to improve the merged solution (Cheng et al., 2025; Xu et al., 2026). Typically, these techniques apply a uniform regularization strength across all layers, despite varying levels of expert disagreement throughout the network. Alternative strategies adjust layer scales by depth or parameter norms (Wang et al., 2025; Si et al., 2025), but do not directly assess how well a shared update accommodates competing experts.

This study introduces *conflict-guided model merging*, a data-free merging scheme that allocates regularization strength to each layer based on a conflict score derived from the task vectors. This score quantifies the disagreement that remains after an unregularized merge. Specifically, we divide the minimum ACTMat loss by the loss obtained when retaining the pretrained weights, with both quantities computed solely from task vectors. A low score indicates that a shared update can closely approximate the experts under the estimated covariances, while a high score suggests that significant disagreement persists even at the optimal fit. We use this score to determine whether layers with greater conflict benefit from increased regularization. We evaluate two alternatives independently: incorporating a Ridge penalty into the merging loss and running an orthogonalized iterative solver

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.

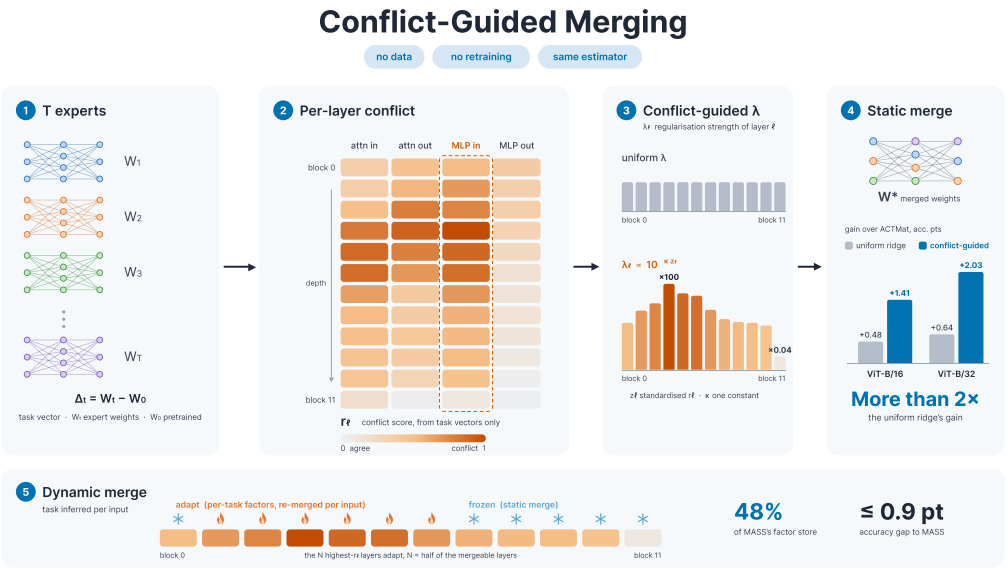


Figure 1: **Overview of conflict-guided model merging.** Task vectors (1) are used to compute a conflict score for each layer (2), with darker cells indicating greater conflict. Layers with higher scores receive stronger Ridge penalties (3), while the covariance estimator of ACTMat (Gamal Abdel Hameed et al., 2026) remains unchanged. The bars (4) show accuracy gains over ACTMat when merging eight tasks. For input-adaptive merging (5), we re-merge the layers with the highest conflict for each input and keep the remaining layers fixed. This uses 46–48% of the per-task factor storage of our MASS baseline (Crisostomi et al., 2026), with a normalized accuracy gap of at most 0.9 points.

for a fixed step budget. Both approaches retain ACTMat's covariance estimator. Figure 1 presents an overview of the approach and its principal results.

The experiments separately evaluate the contributions of regularization and conflict adaptation. Replacing ACTMat's closed-form solution with a regularized solver increases accuracy by +0.48 and +0.64 percentage points on ViT-B/16 and ViT-B/32, respectively, when merging eight tasks. Adjusting the regularization strength to each layer's conflict score yields additional gains of +0.94 and +1.39 percentage points. These further improvements constitute the primary focus of this study. Conflict-guided Ridge also surpasses ACTMat's Adam solver on both backbones, indicating that the improvements extend beyond comparison with an unregularized closed-form solution (Section 5.2).

The study also examines whether the conflict score can reduce task-specific storage in routed merging (Crisostomi et al., 2026) by identifying which expert layers to retain. Selecting a whole layer type matches the conflict score, but with more storage (62% vs. 48%). Under more restrictive budgets, the conflict ranking informs which individual layers to retain (Section 5.4).

The experiments also delineate the limitations of conflict adaptation. The results demonstrate its effectiveness for fully fine-tuned CLIP ViT-B encoders, whereas improvements on ViT-L/14 are minimal. For T5, step-size scaling and grouping layers for conflict-score normalization haven performance close to that of uniform regularization, but does not establish a consistent advantage for conflict adaptation (Section D). We release the code, experimental records, and table-generation scripts to ensure each reported result can be traced to its source run.

2 Background

Merging as layer-wise interference minimization. For a given linear layer, let W_0 and W_t denote the pretrained and task-t weight matrices. We define the layer's task update as $\Delta_t = W_t - W_0$. RegMean (Jin et al., 2023) poses merging as a regression: for one linear layer with input distribution

$\mathcal{D}_t$ under task t,

$$\Delta^* \in \arg\min_{\Delta} \sum_{t=1}^{T} \mathbb{E}_{z\sim\mathcal{D}_t}\left[\|\Delta z - \Delta_t z\|_2^2\right], \tag{1}$$

with closed form $\Delta^* = (\sum_t W_t C_t)(\sum_t C_t)^+$ and $C_t = \mathbb{E}_{z\sim\mathcal{D}_t}[zz^\top]$ the second moment of the layer's inputs. The objective is convex, separable across layers, and bounds negative transfer (Sun et al., 2025). Because every expert shares the base, minimising over W is the same as minimising over the update $\Delta = W - W_0$. Its drawback is that it requires C_t, which is estimated from each task's training data.

The ACTMat estimator. ACTMat (Gamal Abdel Hameed et al., 2026) proposes a data-free estimation of $C_t = \mathbb{E}_{z\sim\mathcal{D}_t}[zz^\top]$. If the layer's activations and output-gradient norms are uncorrelated and the covariances approximately stationary, the difference matrix carries the second-order statistics, and ACTMat uses $\Delta_t^\top \Delta_t$ as a proxy for C_t. The merged update then becomes data-free,

$$\Delta^\star \approx \Big(\sum_{t=1}^{T} \Delta_t\, C_t\Big)\Big(\sum_{t=1}^{T} C_t\Big)^+, \qquad C_t = \Delta_t^\top \Delta_t. \tag{2}$$

Following ACTMat, we apply Eq. (2) to parameters stored as matrices and average all other parameters. We call these matrices the *mergeable layers*.

Orthogonalized optimization. Muon (Jordan et al., 2024) forms a momentum matrix from recent gradients and orthogonalizes it before updating the weights. Conceptually, this means taking its singular value decomposition (SVD) and replacing the nonzero singular values with one, while keeping the singular vectors. This gives the corresponding directions equal weight in the update. In practice, Muon approximates this operation efficiently using Newton–Schulz iterations, without computing the SVD. We use this update rule to optimize the layer-wise merging objective.

3 Related work

Data-free merging. Most data-free methods act directly on the geometry of the task vectors. Task Arithmetic (Ilharco et al., 2023) sums them; TIES (Yadav et al., 2023) trims and resolves sign disagreements; TSV (Gargiulo et al., 2025) and Iso-C (Marczak et al., 2025) exploit the low-rank structure of the per-layer matrices, respectively, orthogonalizing dominant singular directions across tasks and flattening the spectrum of the summed update; KnOTS (Stoica et al., 2025) aligns LoRA updates on a shared basis first. CAT Merging (Sun et al., 2025) is closest in motivation but needs a forward pass on unlabeled exemplars per task, projects a subspace of conflicting directions rather than reading a per-layer scalar, and trims task vectors in the Task Arithmetic lineage rather than damping the interference solve; thus, its gains are not a data-free baseline for ours. The interference-minimization line (Jin et al., 2023; Gamal Abdel Hameed et al., 2026) instead defines an objective and derives the update from it, which is what makes the solver a separate design choice. We keep that objective and its estimator intact.

Solving the interference system. We are not the first to note that the closed form is not the only way to obtain the merge, and two recent methods make the point directly. WUDI-Merging (Cheng et al., 2025) minimises $\sum_t \|(W - \Delta_t)\Delta_t^\top\|_F^2 / \|\Delta_t\|_F^2$ by Adam from $W^{(0)} = \sum_t \Delta_t$; since $\|(W - \Delta_t)\Delta_t^\top\|_F^2 = \|(W - \Delta_t)C_t^{1/2}\|_F^2$, that is a task-weighted Eq. (1) under the same data-free proxy, solved iteratively and stopped early rather than in closed form. ACE-Merging (Xu et al., 2026) instead keeps a closed form and damps it with a Tikhonov term, choosing between two normalizations of its covariance estimate by a data-free heterogeneity coefficient read from the task-vector norms. What neither varies is the axis we do: their damping is one constant per model family, while ours is a continuous per-layer quantity read off the merge residual, which is what Table 2 tests. In our experiments ACTMat outperformed ACE-Merging, therefore we didn't compared against ACE-Merging. MaTS (Tam et al., 2024), Fisher-weighted averaging (Matena & Raffel, 2022) and Daheim et al. (2024) solve or damp the same family of systems, but all take their weighting from data, so its strength is tied to an estimate whose quality is known; data-free, no such estimate exists, which makes *how much to damp, and where* an open question rather than a

calibration exercise. Two further methods set a per-layer scale without any solve: LiNeS (Wang et al., 2025) by depth and NAN (Si et al., 2025) by the inverse task-vector norm; we run both as the controls our score has to beat. AdaMerging (Yang et al., 2024) also adapts per layer but learns its coefficients by test-time entropy minimization, where ours is closed-form and consumes no data.

Routing at inference. MoErging methods retain per-task structure for inference: WEMoE (Tang et al., 2024b) and SMILE (Tang et al., 2024a) insert learned or constructed gates, Twin-Merging (Lu et al., 2024) trains a router on per-task data, and EMR-Merging (Huang et al., 2024) and TALL masks (Wang et al., 2024) store per-task corrections. MASS (Crisostomi et al., 2026), our main point of comparison, stores the top $1/T$ singular directions of each task vector, routes each input according to how well each task subspace reconstructs an intermediate activation, and re-merges the selected subspaces with TSV. It selects the output head by the highest logit, so no task identity is assumed. This requires a second forward pass and additional storage for per-task factors. We differ from MASS in which layers retain these factors and which operator re-merges them, holding the router fixed to isolate the effects of these two choices.

4 METHOD

We propose *conflict-guided model merging*: use the disagreement remaining after an unregularized ACTMat merge to set each layer's regularization strength. The proposed ingredient is the assignment of strength across layers from task vectors alone. We evaluate it in two independent variants, conflict-guided Ridge and conflict-guided Muon.

4.1 OBJECTIVE AND CONFLICT MEASUREMENT

Fix a mergeable layer ℓ, with task updates $\Delta_t = W_t - W_0 \in \mathbb{R}^{d_{\text{out}} \times d_{\text{in}}}$ for $t = 1, \dots, T$. ACTMat uses the covariance proxy $C_t = \Delta_t^\top \Delta_t$. Using this proxy, we write Eq. (1) in terms of a candidate merged update Δ:

$$\mathcal{L}_\ell(\Delta) = \sum_t \left\| (\Delta - \Delta_t) C_t^{1/2} \right\|_F^2.$$

The merged layer weights are $W_0 + \Delta$. Let $\Delta^\star$ denote the minimum-norm, unregularized minimizer given by Eq. (2). We compute this reference once per layer, independently of the regularized variant used afterward.

We measure the optimal residual disagreement relative to the null merge $\Delta = 0$, which retains the pretrained weights:

$$r_\ell = \frac{\sum_t \left\| (\Delta^\star - \Delta_t) C_t^{1/2} \right\|_F^2}{\sum_t \left\| \Delta_t C_t^{1/2} \right\|_F^2} \in [0, 1]. \tag{3}$$

If the denominator is zero we set $r_\ell = 0$. Otherwise the score lies in $[0, 1]$: $\Delta^\star$ minimizes the numerator, while the feasible candidate $\Delta = 0$ makes the numerator exactly equal to the denominator. Therefore, the optimum cannot exceed the denominator. A score of zero means that one update matches all tasks under the proxy geometry; a score near one means that even the best fit offers little improvement over the null merge. The ratio is invariant to a common rescaling of the task vectors.

Next, we normalize each layer's conflict score against the score of other layers in a normalization pool $\mathcal{P}$. In our ViT experiments, it contains all mergeable layers, including the attention and MLP projections across transformer blocks. Let $\mu_\mathcal{P}$ and $\sigma_\mathcal{P}$ denote the mean and standard deviation of their conflict scores. We define

$$z_\ell = \frac{r_\ell - \mu_\mathcal{P}}{\sigma_\mathcal{P}}. \tag{4}$$

Thus r_ℓ measures the absolute disagreement, while z_ℓ measures the relative disagreement. The standardized score z_ℓ directly sets the Ridge penalty or Muon step size in the two variants below.

4.2 Two Independent Conflict-Guided Variants

We hypothesize that layers with greater unresolved conflict benefit from stronger regularization. Each variant implements this hypothesis through its own regularization parameter.

Variant 1: Conflict-guided Ridge. Ridge adds an explicit penalty $\lambda_\ell \|\Delta\|_F^2$ to the fitting loss. Let $\alpha > 0$ be the shared base strength and $\kappa_{\mathrm{R}} > 0$ the conflict sensitivity, with $\sigma_{\max}$ denoting the largest singular value. We set the penalty and solve the resulting regularized objective:

$$\Delta_{\mathrm{Ridge}} = \arg\min_{\Delta} \mathcal{L}_\ell(\Delta) + \lambda_\ell \|\Delta\|_F^2, \quad \lambda_\ell = \alpha\, \sigma_{\max}\Big(\sum_t C_t\Big) 10^{\kappa_{\mathrm{R}} z_\ell} \tag{5a}$$

Which has the closed form solution $\Delta_{\mathrm{Ridge}} = \Big(\sum_t \Delta_t C_t\Big)\Big(\sum_t C_t + \lambda_\ell I\Big)^{-1}$. Because Δ is an update, the penalty pulls the merged weights toward the pretrained model. Scaling by $\sigma_{\max}$ expresses α relative to the layer's spectrum; the factor $10^{\kappa_{\mathrm{R}} z_\ell}$ increases the penalty where conflict is higher. Setting $\kappa_{\mathrm{R}} = 0$ recovers uniform Ridge: one relative α shared by all layers, so λ_ℓ still follows each layer's spectrum.

Variant 2: Conflict-guided Muon. We minimizes the same ACTMat loss with Muon-style orthogonalized steps. Let $\Delta^{(k)}$ be the candidate update at iteration k, starting from the mean update $\Delta^{(0)} = \frac{1}{T}\sum_t \Delta_t$, with gradient $g^{(k)} = 2\sum_t (\Delta^{(k)} - \Delta_t) C_t$. $\mathrm{NS}(\cdot)$ approximately orthogonalizes the momentum buffer with a fixed number of Newton–Schulz iterations (Jordan et al., 2024).

Let K be the step budget, c_k a cosine decay with $c_0 = 1$, $\eta_0 > 0$ the base step, μ the momentum, and $\kappa_{\mathrm{M}} > 0$ the conflict sensitivity. For $k = 0, \ldots, K-1$,

$$\begin{aligned} \eta_\ell &= \eta_0\, 10^{-\kappa_{\mathrm{M}} z_\ell}, \qquad m^{(k)} = \mu\, m^{(k-1)} + g^{(k)}, \quad m^{(-1)} = 0, \\ \Delta^{(k+1)} &= \Delta^{(k)} - \eta_\ell c_k\, \mathrm{NS}\big(m^{(k)}\big). \end{aligned} \tag{5b}$$

We return $\Delta^{(K)}$. Regularization is implicit: η_ℓ sets how far the K steps move from the mean initialization, so higher conflict gives a shorter path. Setting $\kappa_{\mathrm{M}} = 0$ recovers uniform Muon. The finite budget matters: run to convergence, a smaller step would not restrict the solution.

Algorithm: Conflict-Guided Merging

Input: Task vectors, normalization pools, one variant chosen for the entire run, and its fixed hyperparameters.

1. Compute each layer's covariance proxies and closed-form reference $\Delta^\star$.
2. Compute r_ℓ using Eq. (3), then z_ℓ within each pool using Eq. (4).
3. For each layer, execute the chosen branch:
   If Ridge: Set λ_ℓ and compute Δ_{Ridge} using Eq. (5a).
   Otherwise, Muon: Initialize at the mean task update, set η_ℓ, and take K steps using Eq. (5b).

Output: Add the resulting merged updates to the pretrained weights.

4.3 Selective Per-Task Storage

We also show the utility of conflict-guidance for input-adaptive merging. Using MASS's factorization and routing procedure (Crisostomi et al., 2026), we use the conflict score to select the leading layers for dynamic merging. At these layers, we re-merge the routed expert subset with Eq. (5a); all other layers use the static merge. On the dynamically merged layers, the conflict score also sets the Ridge penalty of the re-merge; the solve is done in factor space, see Appendix J.

5 Results

5.1 Experimental Setup

We evaluate merging on the 8, 14, and 20 task suites using publicly released CLIP (Radford et al., 2021) ViT-B/32, ViT-B/16, and ViT-L/14 expert checkpoints. All static comparisons use the same

Table 1: Vision full fine-tuning, all nine settings, test split. ACTMat is absolute macro accuracy and every other column is the difference from it in the same setting. WUDI (task-weighted) and the ACTMat repository's Adam solver (lr 10^{-5}, 300 steps) are existing solvers of the same objective; U and C are one solver run at a single uniform strength and with the strength set per layer by the conflict score, so the two $C-U$ columns give the gain from the conflict score. Bold marks the best of the six comparable solver columns. The full eight-task table, with the spectrum-based and data-dependent baselines, is Table 4.

Setting	ACTMat	WUDI	Adam	Ridge U	Ridge C	Muon U	Muon C	Ridge C-U	Muon C-U
ViT-B/16, $T = 8$	89.16	-0.03	+0.85	+0.48	+1.41	+1.13	**+1.51**	+0.94	+0.38
ViT-B/16, $T = 14$	84.92	-3.67	+0.84	+0.48	**+1.37**	+1.12	+1.11	+0.89	-0.01
ViT-B/16, $T = 20$	81.33	-16.01	+0.29	+0.24	+1.00	+0.79	**+1.21**	+0.76	+0.42
ViT-B/32, $T = 8$	85.67	+0.41	+1.30	+0.64	**+2.03**	+1.17	+1.71	+1.39	+0.54
ViT-B/32, $T = 14$	79.80	-3.54	+1.67	+0.70	**+2.22**	+1.64	+2.07	+1.52	+0.44
ViT-B/32, $T = 20$	76.78	-19.51	+1.17	+0.64	+1.86	+1.46	**+2.11**	+1.23	+0.65
ViT-L/14, $T = 8$	94.15	-0.03	-0.09	-0.09	+0.05	**+0.11**	-0.02	+0.14	-0.12
ViT-L/14, $T = 14$	90.05	+0.01	+0.00	+0.02	+0.14	**+0.43**	+0.09	+0.11	-0.33
ViT-L/14, $T = 20$	88.51	-4.92	-0.04	+0.02	+0.28	**+0.48**	+0.32	+0.25	-0.16

expert weights, heads, and loaders, and report macro top-1 test accuracy. Alongside published baselines, we evaluate the ACTMat repository's absolute-λ Ridge and Adam variants. Adam is run in two configurations: lr 10^{-5} for 300 steps (Tables 1 and 4) and lr 10^{-3} for 100 steps (Tables 5 and 19). Complete static results appear in Appendix A, with configuration sensitivity examined in Appendices F and F.2.

For routed merging, we compare against our MASS reimplementation using the same checkpoints, heads, splits, routing layer, threshold, top-k, and head-selection rule. We also report normalized accuracy, computed by dividing each task's merged accuracy by its expert's accuracy and averaging across tasks. Our version keeps the per-task factors at half the mergeable layers and relies on a static merged backbone, while the MASS port uses its own truncation-based backbone. To isolate the effect of the layer selection, we rerun both methods on a shared backbone. Additionally, while each method initially uses its own routing score, we also provide results where both methods utilize the same score. Details in Appendix I.

We select all numerical hyperparameters on the validation split of the eight-task ViT-B/16 setting, then keep them fixed across backbones, task counts, and modalities. On ViT, conflict scores are normalized over all mergeable layers. Appendix F gives the selected values and search grids, and Appendix F.1 describes the implementation details. All vision results use the absolute Muon step of Eq. (5b); the relative step of Appendix D is used only on T5.

5.2 Static Merging

Table 1 separates the two effects. Regularizing the solver at all, with no conflict signal anywhere in it, already improves +0.48 and +0.64 over ACTMat on ViT-B/16 and ViT-B/32 at eight tasks for the Ridge, and +1.13 and +1.17 for the Muon solver. Setting that strength per layer from the conflict score adds +0.94 and +1.39 to the Ridge and +0.38 and +0.54 to the Muon solver. For the Ridge regularization, adding the conflict-adaptive score consistently has a positive additional gain, for different number of tasks and architectures. However, for Muon, the gain can be smaller and disappears at 14 tasks on ViT-B/16 (see Figure 2). On ViT-L/14, neither added gain surpasses our 0.3-point reporting threshold (refer to Section 6). This trend is also evident in the independently fine-tuned experts (check Appendix G).

ACTMat's Adam variant provides a stronger comparison, as it also improves on the closed-form solution on the ViT-B backbones. Even against its strongest observed configuration in each setting, our best conflict-guided variant improves accuracy by +0.53 to +0.67 points.

5.3 Ablations and Analysis

We assess the value of the conflict score by replacing it with readily available layer descriptors while holding the objective, proxy, solver, damping law, base strength, and normalization pool fixed

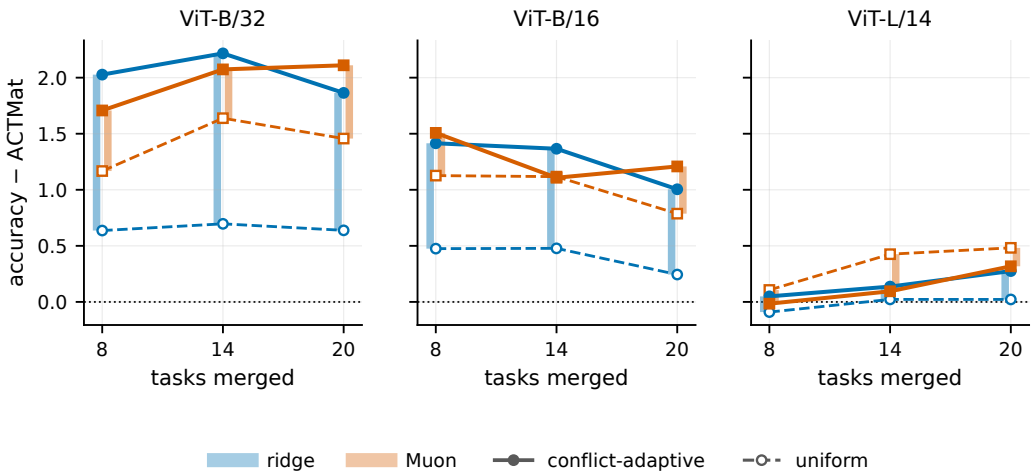


Figure 2: Conflict-guided vs uniform regularization, test split. Both comparison are drawn: dashed is the solver run at a single uniform strength, solid is the same solver with the strength set per layer by the conflict score, one colour per solver, and the vertical bar at each task count is the gain from the conflict score. Accuracy is measured against the ACTMat merge of the same cell. On the two ViT-B backbones the bars are consistently positive. On ViT-L/14 the bars collapse and the Muon pair inverts (Section 6).

Table 2: Only the per-layer score entering z_ℓ changes between rows; objective, proxy, solver, damping law, base strength and standardization pool are fixed. Differences are against the ACTMat merge of the same setting. Block depth and task-vector norm are published per-layer schedules. Layer type (one bit) gives MLP input projections one strength and all other layers another. The reassignment row keeps the measured conflict values but shuffles them across layers (mean, standard deviation, number of draws). Layer width and the uniform-strength sweep are in Table 9.

Per-layer score	B/16, 8	B/16, 14	B/32, 8	B/32, 14
No score (uniform damping)	+0.48	+0.48	+0.64	+0.70
Layer type (one bit)	+0.87	+0.81	+1.22	+1.63
Block depth	+0.69	+0.83	+1.05	+1.31
Task-vector norm	+0.72	+0.92	+1.61	+1.81
Conflict values, randomly reassigned	-0.09 $\pm$ 0.39 (20)	+0.04 $\pm$ 0.33 (20)	+0.26 $\pm$ 0.29 (20)	+0.55 $\pm$ 0.38 (20)
Conflict (ours)	**+1.41**	**+1.37**	**+2.03**	**+2.22**

(Table 2). The alternatives include uniform weight baseline, layer type (additional weight on MLP layers), random shuffling of our conflict values, block depth based scores (Wang et al., 2025) and task-vector norm (Si et al., 2025) based scores. Our approach outperforms the strongest alternative in all four settings by +0.41 to +0.55 points. These margins are close to our 0.3-point reporting threshold, so the results support conflict as the best score tested without establishing it as uniquely effective. Even when experts are fine-tuned from a different seed, conflict still outperforms the best alternative score by +0.42. Meanwhile, the alternatives can vary by as much as 2.70 and their order can change (see Appendix G).

The gains depend on how regularization is assigned across layers. To show this, we keep the measured conflict values but permute them across layers. This has a strong negative effect that reduces accuracy by +1.33 to +1.77 points. These controls indicate that the benefit comes from the adaptive nature of our regularization that weights each layer appropriately.

This benefit is concentrated in the MLP input projections. Layer-type controls show no measurable contribution from attention projections, and probes that apply the correction to one layer at a time locate the gains mainly in early-to-middle blocks. Selecting layers in descending order of conflict recovers 90% of the final gain with 36% of the layers for Ridge and 48% for Muon (Figure 3). A ridge is typically required when the sum of the condition numbers, $\sum_t C_t$, is nearly singular. One might think that the conflict score would serve as a method to identify these layers, but that's

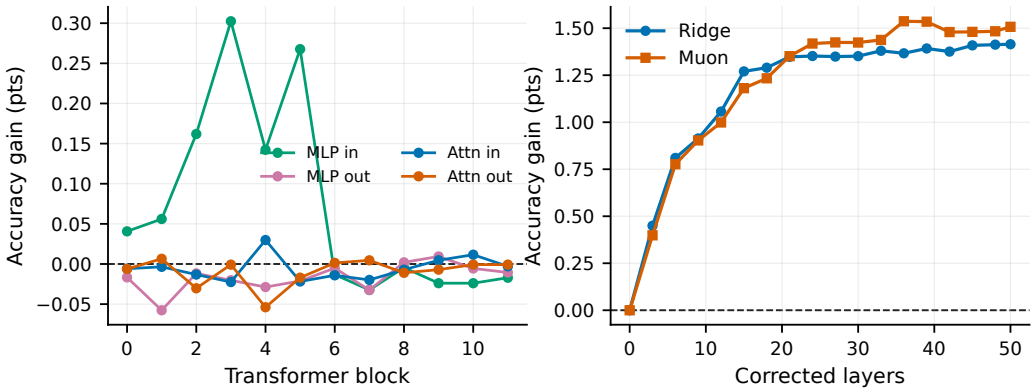


Figure 3: Where the damping pays, on ViT-B/16 at 8 tasks with the conflict-guided Ridge, test split. Left: the gain from correcting one layer and leaving the rest at the closed form, by block and layer type; the MLP input projections carry it and the attention projections do not. Right: the gain against the number of layers corrected, taken in conflict-rank order, for both solvers.

Table 3: Input-adaptive merging on the test split. Entries report macro accuracy, with normalized accuracy in parentheses. Both routed columns are ours, on identical expert checkpoints, heads, splits and router settings. Store is per-task factor storage as a percentage of what the port keeps.

Backbone	T	Static merge	MASS (our port)	Ours, top-N	Store %
ViT-B/32	8	85.7	89.2 (96.1)	88.8 (95.8)	48
	14	79.8	86.1 (94.8)	85.3 (93.9)	48
	20	76.8	77.5 (85.6)	77.9 (85.8)	48
ViT-B/16	8	89.2	92.3 (97.5)	91.7 (96.8)	48
	14	84.9	90.1 (97.2)	89.5 (96.5)	47
	20	81.3	77.1 (83.5)	78.5 (84.9)	46
ViT-L/14	8	94.2	91.6 (95.7)	92.0 (96.1)	48
	14	90.1	90.0 (95.6)	90.0 (95.6)	46
	20	88.5	86.6 (91.6)	87.3 (92.4)	48

not the case. In fact, the layers that show the most conflict are often the best-conditioned ones. Interestingly, using the condition number as a scoring method doesn't outperform a uniform ridge (see Appendix H for more details).

Since merging is deterministic, we examine whether the average gains are shared across tasks and persist with different experts. Compared task by task with the closed-form solution, the conflict-guided solvers improve 6 of 8 to 18 of 20 tasks on the ViT-B backbones, and 3 to 5 of 8 tasks on ViT-L/14. We emphasize these counts because the eight-task suite is a fixed, heterogeneous collection rather than a random sample from a population. None of the eight-task ViT-B comparisons remains statistically significant after Holm correction, so we place greater weight on the evidence from the 14- and 20-task suites (Appendix A). The choice of experts remains an important source of uncertainty. On the single independent expert set we tested, the ordering is preserved, although all gains are 0.2–0.4 points smaller (Appendix G).

On the eight-task ViT-L/14 benchmark, our variants stay within ± 0.2 points of ACTMat. The closed-form merge is already only 1.65 points below the individual experts, compared with 5.48 on ViT-B/16, leaving less room for improvement. However, this gap alone does not predict a benefit: the vision LoRA settings have two to six points of headroom but show no accuracy gain. In these settings, the per-layer residual is zero on most layers, and the adaptive schedule becomes effectively uniform (Appendix B). We examine these limits further in Section D.

5.4 Input-Adaptive Merging

By storing per-task factors only at the highest-conflict layers, we manage to cut the memory usage of routed merging by about half, with only a slight dip in accuracy (see Table 3). Our approach retains around 46–48% of the per-task factors that our MASS port uses. Across all nine settings we tested, the normalized accuracy only drops by a maximum of 0.9 points. In the six scenarios involving eight and fourteen tasks, the median difference is -0.53, while the largest drop is -0.85. Interestingly, on the ViT-L/14 model, the half-size storage option performs just as well, if not better, than the full-store version across all task counts. We take a careful look at the results from the twenty-task suite: it includes datasets that are almost identical, which can lead to confusion for the router. As a result, the accuracy we see is more about routing mistakes than the actual quality of the merging process (see Appendix I).

6 Limitations and Conclusion

Limitations. Merging is deterministic, so there is no run-to-run variance; what our numbers do not capture is sensitivity to the choice of experts. On an independently fine-tuned set the picture held but every gain was 0.2 to 0.4 points smaller and our Ridge and Muon variants swapped rank (Appendix G), so we use 0.3 points as a practical reporting threshold and decline to rank our variants against each other. What we claim, we claim for fully fine-tuned CLIP ViT-B encoders: the conflict schedule is supported there and nowhere else we looked. The evidence base is two model families from one checkpoint source, CLIP ViT at three sizes and T5 at two, so the transfer findings come from a single architecture change; on text the solve ties the strongest existing solver, the schedule reaches only parity, and at twelve text tasks every variant is behind the closed form. The input-adaptive results are vision-only and rest on a port of MASS that omits a redundancy filter and two checkpoint conventions, affecting both of our columns identically (Appendix I). The LoRA and text experts are ours, the originals never having been released.

Conclusion. The layer-wise interference objective has received more attention than the question of how to solve it, and where the solver has been varied the damping has stayed a single constant for the whole model. Conflict-guided damping makes it per-layer, from a measure of how irreconcilable the experts are that the task vectors already contain. On fully fine-tuned ViT-B models that residual carries information about where to damp which layer type, depth, width and task-vector norm do not carry, and which a permutation of its own values across layers destroys. Used instead to choose which layers keep per-task factors, it beats a random choice, but a one-bit layer-type rule matches it using more storage, and the ranking only starts to matter below a type-sized budget (Section 5.4). That it does not survive LoRA experts or a T5 is the other half of the result: the signal is not universal, and a step size and a standardization pool are part of a method's specification rather than details of its implementation.

Beyond regularization, this analysis could help identify which tasks are poorly represented at each layer and where additional task-specific capacity would be useful. Future work should test when the residual can reliably guide these decisions across architectures and fine-tuning methods.

References

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. In *Proceedings of the 42nd International Conference on Machine Learning*, pp. 10121–10143, 2025. WUDI-Merging; arXiv:2503.08099. Minimises $\sum_t \|(W - \Delta_t)\Delta_t^\top\|_F^2 / \|\Delta_t\|_F^2$ by Adam from $W^{(0)} = \sum_t \Delta_t$, which is the ACTMat objective with a per-task weighting. Code: github.com/nathanielyvo/WUDI-Merging.

Donato Crisostomi, Alessandro Zirilli, Antonio Andrea Gargiulo, Maria Sofia Bucarelli, Simone Scardapane, Fabrizio Silvestri, Iacopo Masi, and Emanuele Rodolà. MASS: Moerging through adaptive subspace selection. In *International Conference on Learning Representations (ICLR)*, 2026. arXiv:2504.05342.

Nico Daheim, Thomas Möllenhoff, Edoardo Maria Ponti, Iryna Gurevych, and Mohammad Emtiyaz Khan. Model merging by uncertainty-based gradient matching. In *International Conference on Learning Representations (ICLR)*, 2024. arXiv:2310.12808.

Marawan Gamal Abdel Hameed, Derek Tam, Pascal Jr Tikeng Notsawo, Colin Raffel, and Guillaume Rabusseau. Model merging via data-free covariance estimation. *arXiv preprint arXiv:2604.01329*, 2026.

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)*, pp. 18695–18705, 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.

Chenyu Huang, Peng Ye, Tao Chen, Tong He, Xiangyu Yue, and Wanli Ouyang. EMR-merging: Tuning-free high-performance model merging. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024.

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.

Keller Jordan, Yuchen Jin, Vlado Boza, Jiacheng You, Franz Cesista, Laker Newhouse, and Jeremy Bernstein. Muon: An optimizer for hidden layers in neural networks. https://kellerjordan.github.io/posts/Muon/, 2024.

Zhenyi Lu, Chenghao Fan, Wei Wei, Xiaoye Qu, Dangyang Chen, and Yu Cheng. Twin-merging: Dynamic integration of modular expertise in model merging. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024.

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

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

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 *International Conference on Machine Learning (ICML)*, pp. 8748–8763, 2021.

Chongjie Si, Kangtao Lv, Jingjing Jiang, Yadao Wang, Yongwei Wang, Xiaokang Yang, Wenbo Su, Bo Zheng, and Wei Shen. Nan: A training-free solution to coefficient estimation in model merging. *arXiv preprint arXiv:2505.16148*, 2025.

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.

Wenju Sun, Qingyong Li, Yangliao Geng, and Boyang Li. Cat merging: A training-free approach for resolving conflicts in model merging. In *International Conference on Machine Learning (ICML)*, volume 267, 2025.

Derek Tam, Mohit Bansal, and Colin Raffel. Merging by matching models in task parameter subspaces. *Transactions on Machine Learning Research (TMLR)*, 2024. arXiv:2312.04339.

Anke Tang, Li Shen, Yong Luo, Shuai Xie, Han Hu, Lefei Zhang, Bo Du, and Dacheng Tao. SMILE: Zero-shot sparse mixture of low-rank experts construction from pre-trained foundation models. *arXiv preprint arXiv:2408.10174*, 2024a.

Anke Tang, Li Shen, Yong Luo, Nan Yin, Lefei Zhang, and Dacheng Tao. Merging multi-task models via weight-ensembling mixture of experts. In *International Conference on Machine Learning (ICML)*, pp. 47778–47799, 2024b.

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

Ke Wang, Nikolaos Dimitriadis, Alessandro Favero, Guillermo Ortiz-Jimenez, François Fleuret, and Pascal Frossard. Lines: Post-training layer scaling prevents forgetting and enhances model merging. In *International Conference on Learning Representations (ICLR)*, 2025. arXiv:2410.17146.

Bo Xu, Haotian Wu, Hehai Lin, Weiquan Huang, Beier Zhu, Yao Shu, and Chengwei Qin. ACE-merging: Data-free model merging with adaptive covariance estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2026. arXiv:2603.02945.

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

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.

REPRODUCIBILITY STATEMENT

Model weights are handled as follows. The vision experts are the publicly released checkpoints used by prior work and are fetched by a script in the repository. The LoRA and text experts are ours, trained with the original authors' scripts and hyperparameters because those checkpoints were never released; the release documents their training cost rather than shipping them. Datasets are obtained through their standard loaders, with one exception documented in the release: one dataset requires a specific canonical split, which the setup script installs.

All experiments ran on a single machine with four RTX 6000 Ada GPUs. The complete campaign, including the experts we trained ourselves, is recorded in the manifest with per-job wall-clock time.

A STATIC MERGING: COMPLETE TABLES

Table 5 adds to Table 1 the baselines that do not fit there, at 14 and 20 tasks. What is stable across task counts is less than an ordering: the uniform Ridge is the lowest of our four variants in all nine cells, while the other three change places. The Ridge and the Muon solve swap rank between task counts on every backbone, and at 14 tasks on ViT-B/16 the uniform Muon solve edges the conflict-guided one by +0.01. The largest gap between our two conflict-guided variants is +0.32, near our 0.3-point threshold (Section 6), which is why we decline to rank our own variants against each other. The margin over the closed form is not monotone in T either (Figure 2). All of them stay well clear of the uniform Ridge on both ViT-B backbones at every task count.

Table 6 restricts the conflict-guided correction to one layer type at a time. Both solvers agree: the MLP input projections carry the correction, and the attention projections are indistinguishable from applying no correction at all.

Table 7 reports the agreement between the data-free proxy and the input covariance measured on held-out data, which the method never uses. The proxy is far closer to the measured covariance than an isotropic prior is, which supports the estimator's premise; the weak rank correlation in the last row suggests that the conflict score isn't just a straightforward measure of proxy error.

Table 4: Vision full fine-tuning at eight tasks, test split, with the baselines Table 1 omits. Parentheses give accuracy differences from ACTMat. Individual experts and data-dependent RegMean are reference points, not competitors. Rows marked *authors' code* are variants from the ACTMat repository.

Method	ViT-B/16	ViT-B/32	ViT-L/14
Individual experts (ceiling)	94.65 (+5.48)	92.85 (+7.18)	95.80 (+1.65)
RegMean (uses data)	85.26 (-3.90)	82.00 (-3.67)	90.19 (-3.96)
TIES	79.37 (-9.79)	73.87 (-11.80)	86.48 (-7.68)
TSV	88.51 (-0.65)	85.26 (-0.41)	92.79 (-1.36)
Iso-C	89.00 (-0.16)	84.76 (-0.91)	93.39 (-0.76)
WUDI-Merging	89.13 (-0.03)	86.08 (+0.41)	94.12 (-0.03)
ACTMat (reference)	**89.16**	**85.67**	**94.15**
+ absolute Ridge (authors' code)	86.20 (-2.96)	83.19 (-2.48)	91.34 (-2.81)
+ Adam solver (authors' code, lr 10^{-5}, 300 steps)	90.01 (+0.85)	86.97 (+1.30)	94.07 (-0.09)
Ours: uniform Ridge	89.64 (+0.48)	86.31 (+0.64)	94.06 (-0.09)
Ours: uniform Muon	90.29 (+1.13)	86.84 (+1.17)	94.26 (+0.11)
Ours: conflict-guided Ridge	90.58 (+1.41)	87.70 (+2.03)	94.20 (+0.05)
Ours: conflict-guided Muon	90.67 (+1.51)	87.38 (+1.71)	94.14 (-0.02)
Zero-shot (no merging)	55.55 (-33.61)	48.23 (-37.44)	65.20 (-28.95)

Table 5: Vision full fine-tuning at 14 and 20 tasks, test split. Differences are against the ACTMat merge of the same cell.

Method	B/16 T=14	B/16 T=20	B/32 T=14	B/32 T=20	L/14 T=14	L/14 T=20
ACTMat (reference)	**84.92**	**81.33**	**79.80**	**76.78**	**90.05**	**88.51**
+ Adam solver (authors' code, lr 10^{-3}, 100 steps)	85.16 (+0.24)	82.00 (+0.67)	80.98 (+1.18)	78.23 (+1.45)	89.88 (-0.18)	88.77 (+0.26)
TSV	84.36 (-0.55)	80.43 (-0.91)	79.67 (-0.13)	76.57 (-0.21)	89.08 (-0.98)	87.70 (-0.81)
Ours: conflict-guided Ridge	86.28 (+1.37)	82.34 (+1.00)	82.02 (+2.22)	78.65 (+1.86)	90.19 (+0.14)	88.78 (+0.28)
Ours: conflict-guided Muon	86.03 (+1.11)	82.54 (+1.21)	81.88 (+2.07)	78.89 (+2.11)	90.15 (+0.09)	88.83 (+0.32)
TIES	71.93 (-12.99)	66.27 (-15.06)	64.95 (-14.85)	58.50 (-18.28)	79.32 (-10.74)	74.47 (-14.04)
Task arithmetic	49.04 (-35.88)	30.53 (-50.80)	41.05 (-38.75)	27.02 (-49.77)	68.52 (-21.54)	46.43 (-42.08)
Weight averaging	69.43 (-15.49)	65.38 (-15.96)	63.94 (-15.86)	60.87 (-15.91)	76.76 (-13.30)	71.66 (-16.85)

Table 6: Restricting the correction to one layer type, ViT-B/16. Differences are against the ACTMat merge of the same setting, i.e. the same merge with no correction applied.

Layer set	Ridge, T=8	Ridge, T=14	Muon, T=8	Muon, T=14
MLP input projection only	90.41 (+1.25)	86.11 (+1.20)	90.47 (+1.31)	86.07 (+1.15)
MLP output projection only	89.17 (+0.01)	84.96 (+0.04)	88.96 (-0.20)	84.82 (-0.10)
Attention input projection only	89.20 (+0.03)	85.05 (+0.13)	89.35 (+0.18)	85.06 (+0.14)
Attention output projection only	89.11 (-0.06)	84.87 (-0.05)	89.07 (-0.09)	84.87 (-0.05)
All matrices	90.58 (+1.41)	86.28 (+1.37)	90.67 (+1.51)	86.03 (+1.11)

Paired per-task comparison. Merging is deterministic, so there is no run-to-run variance to report; Table 8 instead asks whether a mean difference of one to two points is carried by the task set or by a few tasks. Each method is paired against the ACTMat merge task by task, over all nine vision cells. Confidence intervals are Student's t at the stated degrees of freedom, and the p values are Holm-corrected across every test in the table rather than within a cell, which is the conservative choice and the one that makes the 8-task cells look as marginal as they are. Read that way, every variant clears the five-percent level in 2 of the three ViT-B/16 cells and 1 of the three ViT-B/32 cells, in both cases the largest task counts, and in 0 ViT-L/14 cells. The pattern is that resolution improves with the number of merged tasks: with eight paired observations (seven degrees of freedom) the interval is wide even where the mean difference is two points, whereas at 19 degrees of freedom a one-point difference is comfortably resolved.

The permutation control read as a test. The permutation row of Table 9 keeps the measured conflict values and destroys only their assignment to layers, so it is a null distribution for the question the schedule is supposed to answer: does it matter *which* layer gets which damping? Reported as a

Table 7: Proxy versus measured input covariance, averaged over mergeable layers and tasks, 8 tasks.

Statistic	ViT-B/16	ViT-B/32	ViT-L/14
cosine(proxy, measured covariance)	0.313	0.327	0.335
cosine(identity, measured covariance)	0.123	0.128	0.110
Spearman(conflict, cosine)	0.200	0.121	0.296

Table 8: Paired per-task comparison against the ACTMat merge, test split. Every ViT-B setting improves the large majority of its tasks; ViT-L/14 does not separate from zero. CIs are unadjusted 95% intervals; p values are Holm-adjusted.

Setting	Method	mean Δ	95% CI	paired t	df	p (Holm)	tasks improved
ViT-B/16, 8	conflict-guided Ridge	+1.41	[+0.66, +2.17]	+4.42	7	0.052	7/8
	conflict-guided Muon	+1.51	[+0.68, +2.34]	+4.29	7	0.054	7/8
ViT-B/16, 14	conflict-guided Ridge	+1.37	[+0.90, +1.83]	+6.32	13	$<$0.001	13/14
	conflict-guided Muon	+1.11	[+0.57, +1.64]	+4.48	13	0.012	12/14
ViT-B/16, 20	conflict-guided Ridge	+1.00	[+0.58, +1.43]	+4.99	19	0.002	17/20
	conflict-guided Muon	+1.21	[+0.65, +1.77]	+4.49	19	0.005	17/20
ViT-B/32, 8	conflict-guided Ridge	+2.03	[+0.21, +3.84]	+2.64	7	0.336	7/8
	conflict-guided Muon	+1.71	[+0.03, +3.38]	+2.41	7	0.419	6/8
ViT-B/32, 14	conflict-guided Ridge	+2.22	[+0.69, +3.75]	+3.13	13	0.111	13/14
	conflict-guided Muon	+2.07	[+0.34, +3.81]	+2.58	13	0.250	11/14
ViT-B/32, 20	conflict-guided Ridge	+1.86	[+0.96, +2.77]	+4.32	19	0.007	18/20
	conflict-guided Muon	+2.11	[+1.14, +3.08]	+4.55	19	0.005	18/20
ViT-L/14, 8	conflict-guided Ridge	+0.05	[-0.24, +0.34]	+0.40	7	1.000	4/8
	conflict-guided Muon	-0.02	[-0.34, +0.31]	-0.11	7	1.000	3/8
ViT-L/14, 14	conflict-guided Ridge	+0.14	[-0.05, +0.33]	+1.55	13	0.724	9/14
	conflict-guided Muon	+0.09	[-0.19, +0.38]	+0.71	13	1.000	9/14
ViT-L/14, 20	conflict-guided Ridge	+0.28	[-0.01, +0.56]	+1.99	19	0.427	14/20
	conflict-guided Muon	+0.32	[-0.09, +0.72]	+1.64	19	0.710	11/20

Table 9: Table 2 in full: every alternative score and the uniform-strength sweep. Objective, proxy, solver, damping law, strength and standardization pool are identical in every row and only the per-layer score entering z_ℓ changes. Differences are against the ACTMat merge. The last control keeps the true conflict values and destroys only their assignment to layers.

Per-layer score	B/16, 8	B/16, 14	B/32, 8	B/32, 14
No score (uniform damping)	+0.48	+0.48	+0.64	+0.70
at 4× strength	+0.30	+0.35	+0.87	+1.13
at 10× strength	-0.39	-0.34	+0.26	+0.68
at 20× strength	-1.46	-1.33	-0.91	-0.29
Layer type (one bit)	+0.87	+0.81	+1.22	+1.63
Block depth	+0.69	+0.83	+1.05	+1.31
Layer width	-2.36	-2.20	-2.42	-1.77
Task-vector norm	+0.72	+0.92	+1.61	+1.81
Conflict values, randomly reassigned	-0.09 $\pm$ 0.39 (20)	+0.04 $\pm$ 0.33 (20)	+0.26 $\pm$ 0.29 (20)	+0.55 $\pm$ 0.38 (20)
Conflict (ours)	**+1.41**	**+1.37**	**+2.03**	**+2.22**

mean and a spread over three draws it is a sanity check; drawn more deeply it becomes a permutation test.

On ViT-B/16 at 8 tasks we have 20 draws. Their mean is -0.09 against +1.41 for the true assignment, which sits 3.8 standard deviations above the null, and 0 of the draws reach it. As a Monte Carlo permutation test with plus-one correction that is $p = 0.048$, the smallest value 20 draws can produce being $1/(20 + 1)$. We report the draw count alongside the p value because the two are not independent: a permutation test with few draws cannot produce a small p however large the effect, and quoting one without the other invites the reader to mistake a bound for a measurement. Twenty

is therefore not an arbitrary stopping point but the fewest draws that can reach the five-percent level at all.

We ran the control deeply in all four cells of Table 9, at the full 20 draws each. At 14 tasks on ViT-B/16, where the vision claim rests, the null mean is +0.04 against +1.37, 4.0 standard deviations above it, with 0 of 20 draws reaching the true assignment and $p = 0.048$. On ViT-B/32 it is 0 of 20 at 8 tasks ($p = 0.048$) and 0 of 20 at 14 ($p = 0.048$). In no cell does a single permutation of the conflict values reach the accuracy of the measured assignment: the Monte Carlo p is 0.048 in every one of the four, the smallest a twenty-draw test can attain. What the measurement contributes is the assignment of damping to layers, and it is not one backbone's or one task count's artefact.

B LoRA EXPERTS

The original authors' LoRA checkpoints were never released, so we retrained them with their script and their hyperparameters: rank-16 LoRA adapters (Hu et al., 2022) on the packed attention input projection, their learning rate, and their per-dataset epoch schedule. Table 10 is the result. The closed form and the spectrum-based baselines land within 0.6 points of their published values, which is the check that our pipeline merges the same objects theirs does. Two rows differ by more: our experts are 1.35 below theirs and our KnOTS-TSV is 1.45 above, differences we attribute to one dataset whose training split we could not match exactly. We note them rather than describing the reproduction as uniform. In this regime no solver improves on the closed form: the Ridge variants tie it, except conflict Ridge on ViT-L/14 (-0.91), and the iterative solvers trail slightly. We report this as a null result.

Table 10: Vision LoRA in the original setting, 8 tasks, test split.

Method	ViT-B/16	ViT-B/32	ViT-L/14
Individual experts (ceiling)	82.75 (+4.71)	79.83 (+5.72)	88.15 (+2.63)
TSV	78.29 (+0.25)	73.05 (-1.07)	85.18 (-0.35)
KnOTS-TSV	76.25 (-1.79)	71.49 (-2.63)	84.12 (-1.41)
ACTMat (reference)	**78.04**	**74.11**	**85.53**
Ours: uniform Ridge	78.06 (+0.02)	74.03 (-0.08)	85.49 (-0.04)
Ours: conflict-guided Ridge	78.03 (-0.01)	74.09 (-0.03)	84.61 (-0.91)
Ours: uniform Muon	77.68 (-0.36)	73.15 (-0.97)	84.90 (-0.63)

Table 11 is the same recipe with a single adapter initialization shared across tasks instead of an independent one per task. Which of the two is the intended setting is not obvious: the released training script seeds each run identically, so taken literally it produces the shared-initialization regime, while only independent initializations reproduce the published numbers. We report both and draw no conclusion about what was run originally. The rank-16 input subspaces then overlap, the tail of $\sum_t C_t$ falls to near machine precision, and the pseudo-inverse amplifies it: the merged update grows to between 1.7 and 2.6 times the norm of the plain sum of task vectors. Accuracy collapses accordingly. Damping helps in proportion to its strength rather than uniformly: our frozen α is far too weak here, and the truncation baseline, which discards the offending directions outright, is the best method on ViT-B (on ViT-L/14 the authors' absolute Ridge is 0.2 ahead). The training precision is not responsible; we verified this by repeating the fine-tunes in full precision and observing the same collapse.

A third regime, and why we separate it. Before matching the original recipe we trained a LoRA set of our own with adapters on the MLP projections as well as the attention projection and a ten times larger learning rate. The experts are four to seven points stronger, and their updates are correspondingly larger: the plain sum of the eight task vectors already destroys the network. In that regime every operator that preserves per-task magnitude fails, the closed form worst of all, and only shrinking operators survive (Table 12). We report it because it is a real data point about the objective on rank-constrained updates, and we keep it separate because it is not the setting the original work evaluated; it should not be read as a comparison against their published numbers.

Table 11: The same LoRA recipe with a shared adapter initialization across tasks.

Method	ViT-B/16	ViT-B/32	ViT-L/14
Individual experts (ceiling)	82.77 (+37.67)	79.64 (+31.29)	88.16 (+17.21)
ACTMat (reference)	**45.09**	**48.35**	**70.95**
+ absolute Ridge (authors' code)	72.38 (+27.29)	72.61 (+24.26)	85.48 (+14.53)
Ours: uniform Ridge	54.50 (+9.41)	61.48 (+13.13)	80.90 (+9.95)
Ours: uniform Muon	76.71 (+31.61)	73.62 (+25.27)	84.84 (+13.90)
TSV	79.39 (+34.29)	73.65 (+25.30)	85.28 (+14.33)
Ours: conflict-guided Ridge	72.81 (+27.72)	73.09 (+24.75)	84.08 (+13.14)
Ours: conflict-guided Muon	75.01 (+29.92)	67.87 (+19.52)	84.61 (+13.66)

Table 12: The strong-LoRA regime: our own recipe, not the original one. Differences are against the ACTMat merge of the same column.

Method	ViT-B/16	ViT-B/32	ViT-L/14
Individual experts (ceiling)	90.48 (+84.14)	88.07 (+67.64)	92.75 (+20.28)
ACTMat (reference)	**6.34**	**20.43**	**72.47**
+ Adam solver (authors' code)	6.93 (+0.59)	23.71 (+3.28)	80.76 (+8.29)
TSV	45.97 (+39.63)	31.99 (+11.56)	75.81 (+3.34)
Ours: uniform Ridge	7.22 (+0.88)	26.07 (+5.64)	78.65 (+6.18)
Ours: uniform Muon	20.15 (+13.81)	37.82 (+17.39)	82.71 (+10.24)
Weight averaging	68.21 (+61.87)	61.73 (+41.30)	75.25 (+2.77)
Task arithmetic	64.08 (+57.74)	61.61 (+41.18)	81.65 (+9.18)

Mechanism of the shared-initialization collapse. With a shared adapter initialization the rank-16 input subspaces of the experts are correlated, so the tail of $\sum_t C_t$ falls to near machine precision and the pseudo-inverse amplifies it: the merged update grows to between 1.7 and 2.6 times the norm of the plain sum. Damping helps in proportion to its strength. Our frozen α, chosen on full fine-tuning, is far too weak; the authors' much stronger absolute damping recovers more; and the truncation baseline, which discards the offending directions outright, is best on ViT-B. This is the one regime in which their absolute-λ choice is better calibrated than ours.

C Input-adaptive merging: operator and selection isolated

Table 13 holds the router and the routed layer set fixed and varies only what happens on the selected subset, which is what separates the contribution of the re-merge operator from the contribution of the layer selection. Reading down a column: the first row is our port of MASS at full store; the second keeps our layer selection but the original truncation operator; the third and fourth swap in the ACTMat operator with a uniform and then a conflict-guided damping.

Table 13: Input-adaptive merging with the router held fixed, routed head. Differences are against the static ACTMat merge of the same setting.

Method	B/16 T=8	B/16 T=14	B/32 T=8	B/32 T=14	Store %
MASS (our port, full store)	92.29 (+3.12)	90.12 (+5.20)	89.21 (+3.54)	86.11 (+6.31)	100
Top-N conflict layers, TSV re-merge	90.89 (+1.73)	88.13 (+3.22)	87.40 (+1.74)	83.59 (+3.79)	48
+ ACTMat re-merge	90.67 (+1.51)	88.39 (+3.47)	89.91 (+4.24)	85.01 (+5.21)	48
+ conflict-guided Ridge	90.58 (+1.42)	88.37 (+3.45)	89.87 (+4.20)	85.00 (+5.20)	48

Two readings follow. First, the ACTMat operator beats truncation clearly on ViT-B/32 but not on ViT-B/16, and on a shared backbone its effect is -0.05 to +0.23, within our threshold. Second, the conflict-guided damping inside the re-merge does not help: at matched store it is below the uniform damping in all nine cells, by 0.011 to 0.115 points. This is why Section 5.4 claims the conflict signal as a selector in this setting and not as a schedule.

Table 14 repeats the comparison with the router held equal on both sides. The single-block router is the one the original method specifies and is what Table 3 uses; the ensemble router is ours. It helps both methods at eight tasks but lowers both at 14 tasks on ViT-B. On ViT-L/14 we ran it only for the full-store method.

Table 14: Router held equal on both sides. Absolute macro accuracy with normalized accuracy in parentheses.

Backbone	T	Router	MASS (our port)	Ours, top-N
ViT-B/32	8	single block (theirs)	89.2 (96.1)	88.8 (95.8)
		ensemble (ours)	90.0 (96.7)	90.0 (96.7)
ViT-B/32	14	single block (theirs)	86.1 (94.8)	85.3 (93.9)
		ensemble (ours)	81.7 (90.0)	82.9 (91.2)
ViT-B/32	20	single block (theirs)	77.5 (85.6)	77.9 (85.8)
		ensemble (ours)	–	69.7 (77.5)
ViT-B/16	8	single block (theirs)	92.3 (97.5)	91.7 (96.8)
		ensemble (ours)	92.8 (97.8)	92.9 (97.9)
ViT-B/16	14	single block (theirs)	90.1 (97.2)	89.5 (96.5)
		ensemble (ours)	89.2 (96.0)	89.0 (95.8)
ViT-B/16	20	single block (theirs)	77.1 (83.5)	78.5 (84.9)
		ensemble (ours)	–	77.2 (83.6)
ViT-L/14	8	single block (theirs)	91.6 (95.7)	92.0 (96.1)
		ensemble (ours)	95.1 (99.2)	–
ViT-L/14	14	single block (theirs)	90.0 (95.6)	90.0 (95.6)
		ensemble (ours)	92.6 (98.2)	–
ViT-L/14	20	single block (theirs)	86.6 (91.6)	87.3 (92.4)
		ensemble (ours)	83.5 (88.6)	–

The accuracy–store frontier. Figure 4 sweeps the number of stored layers for each configuration. It is *not* an operator comparison: the curves use different static backbones (see legend), so the separation between curves is the joint effect of the operator and the backbone. On a shared backbone the operator is a tie (-0.05 to +0.23), which is why we read this figure for the shape of each curve rather than for the distance between them. The shape is a diminishing return that has not yet saturated: between half store and the largest store measured the curves still gain 0.36 to 1.81 points, the larger figures at 14 tasks, so the half-size store of Section 5.4 buys its memory saving at a real if small cost rather than at none.

Table 15: Which layers to store, with the re-merge operator, static backbone and router held fixed. The first three rows are matched in layer count (24, the number of MLP projections on a ViT-B encoder), not in storage. The last three gate to the 12 MLP input projections and choose inside that gate, where the store falls by a further factor of three. Random selections are three draws, reported as mean and standard deviation.

Selection rule	Store (% of MASS)	ViT-B/16, 8 tasks	ViT-B/32, 8 tasks	ViT-B/16, 14 tasks
Conflict ranking (ours)	47.7	90.58	89.87	88.37
MLP projections (one-bit rule)	62	90.68	90.02	88.83
Random subset	48.7	90.28 $\pm$ 0.06	89.21 $\pm$ 0.11	88.09 $\pm$ 0.08
All 12 MLP input projections	31	90.11	88.98	87.70
Highest-conflict 6 MLP inputs (ours)	15.5	89.19	88.15	86.20
Random 6 MLP inputs	15.5	88.46 $\pm$ 0.31	86.76 $\pm$ 0.47	85.36 $\pm$ 0.43

Below a type-sized store, the ranking is the selector. The verdict above is measured at a store of 24 layers, where a rule naming the MLP projections already fills the budget, so the two rules are barely competing. Below it they separate. Gating to the 12 MLP input projections and ranking by conflict inside the gate, the six highest-conflict projections use 15.5% of our MASS port's per-task store, a third of the row above, and beat six random projections of the same type by +0.73 to +1.39 in all three ViT-B settings, above every individual draw. The gate is also worth something against ranking by conflict across all layer types, though here we can say less than we would like: the two

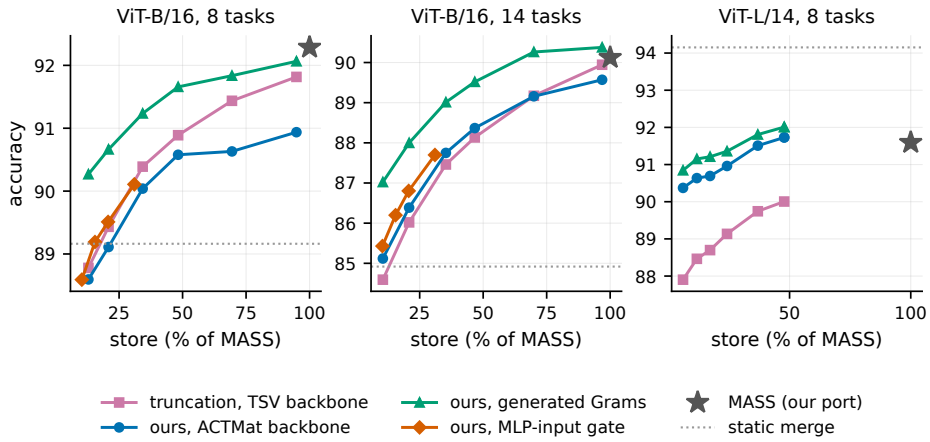


Figure 4: Accuracy against store size, test split. Each curve sweeps the number of stored layers for one configuration; the dotted line is the static merge and the star is our MASS port at full store. The first three curves vary the re-merge operator and the static backbone together, so the gap between them is not attributable to the operator, which is a tie once the backbone is matched; we therefore read them for the shape of each curve rather than for the distance between them. The fourth, *MLP-input gate*, ranks by conflict inside the MLP input projections instead of across all mergeable layers, and is what reaches the small-store end of the axis at all. Every curve is still climbing at the largest store measured. The ViT-L/14 sweep stops at 48 percent of MASS's store, so its curves make no claim beyond that point.

rules were run at a common budget in only one place, around a fifth of the store, where the gate is ahead by +0.40 and +0.42 on the two ViT-B/16 settings and level on ViT-B/32 (-0.05, against a store three points larger). The six blocks chosen are the same in all three settings, the first six after the input block, which is where Figure 3 puts the static gain, so the dynamic selector and the static anatomy agree on where the action is. The cut is not free: six layers sit -3.9 to -1.1 below the port and the gap grows with the task count, while all 12 input projections (31% of the store, and no measurement needed to pick them) close it to -2.4 to -0.2. Two limits. These are single runs, and the random draws spread 0.31 to 0.47, so the margin over random is several times its own spread while the individual entries are not resolved to that precision. And this is a ViT-B result: on ViT-L/14 the per-type conflict ordering inverts, the attention input projection carrying more than the MLP input, so the gate was not run there and which type to gate on is open.

The routing score was an uncontrolled variable, and we measured it. Each method ran with the routing score its own authors specify: residual reconstruction for MASS, subspace energy for ours. That makes the router an uncontrolled variable in a cross-method comparison, so we re-ran both methods under each score on a shared TSV static backbone (Table 16). Residual reconstruction is the better score for *both* methods, by up to 0.84 points, but it helps them almost equally: the difference between them moves by at most 0.07 in every cell. The comparison in Section 5.4 therefore does not depend on which score is used. We have since made the resolved score part of every result record, rather than only the command-line flag, so the choice cannot go unrecorded again.

On this shared backbone, half the store costs 0.38 to 2.39 points.

D TRANSFER TO A DIFFERENT ARCHITECTURE

Transferring the ViT recipe to T5 gives different results for regularization and conflict adaptation (Table 17). With α fixed at its ViT-selected value, uniform Ridge improves over the closed form in three of the four text settings on each metric, although the successful settings differ between metrics. The authors' Adam solver does the same, and the two remain within 0.88 points of each other. Regularization therefore remains useful on T5, but our solver offers no clear advantage over the existing alternative. The conflict schedule fares worse: applying it unchanged provides no gain, and its Muon variant loses 7 to 16 points, depending on setting and metric. We trace these failures to the scaling of optimization steps and the normalization of conflict scores.

Table 16: Both methods under both routing scores, test split, on a shared TSV static backbone so that only the router and the stored-layer set differ. Each method's own default is marked. Switching the score moves either method's absolute accuracy by up to 0.84 points, but the difference between them moves by at most 0.07.

Backbone, tasks	Router	MASS port	Ours, half store	Difference	Gap moves by
ViT-B-16, 8	residual (MASS default)	92.29	91.40	-0.88	0.03
	energy (our default)	91.80	90.95	-0.85	
ViT-B-16, 14	residual (MASS default)	90.12	88.23	-1.90	0.01
	energy (our default)	90.02	88.13	-1.89	
ViT-B-16, 20	residual (MASS default)	77.08	75.45	-1.64	0.02
	energy (our default)	76.81	75.19	-1.62	
ViT-B-32, 8	residual (MASS default)	89.21	88.02	-1.19	0.07
	energy (our default)	88.72	87.46	-1.26	
ViT-B-32, 14	residual (MASS default)	86.11	83.85	-2.26	0.02
	energy (our default)	85.92	83.64	-2.29	
ViT-B-32, 20	residual (MASS default)	77.54	75.15	-2.39	0.00
	energy (our default)	77.32	74.93	-2.39	
ViT-L-14, 8	residual (MASS default)	91.59	91.20	-0.39	0.01
	energy (our default)	90.75	90.37	-0.38	
ViT-L-14, 14	residual (MASS default)	90.05	89.04	-1.01	0.01
	energy (our default)	89.84	88.82	-1.01	
ViT-L-14, 20	residual (MASS default)	86.57	85.09	-1.48	0.01
	energy (our default)	86.45	84.96	-1.49	

Table 17: What holds where. *Regularized solve* is any of our solvers against ACTMat's closed form; *conflict schedule* is that solver with the strength set per layer against the same solver at one uniform strength. The vision entries are the eight-task cells, with all nine in Table 1; differences under the 0.3 points of Section 6 are read as zero.

Regime	Regularized solve	Conflict schedule	Reading
ViT-B/16, B/32 full tuning	+0.48 to +1.17	+0.38 to +1.39	both supported
ViT-L/14, full tuning	-0.09	+0.14	no measurable gain
Vision LoRA	within resolution	residual degenerate	unsupported
T5, recipe unchanged	mixed	-9.24 on Muon	does not transfer
T5, after corrections	ties the best solver	within 0.39	solve transfers, not score

A fixed absolute step in Eq. (5b) is very different relative to the merged update across architectures. We therefore also use a relative step, $\eta_\ell = \rho \, \|\Delta^\star\|_F / \|\mathrm{NS}(g^{(0)})\|_F \, 10^{-\kappa_M z_\ell}$, where ρ is the fraction of $\|\Delta^\star\|_F$ covered by the first step. The vision step corresponds to $\rho = 0.0157$ on ViT-B/16 but $\rho = 0.00092$ on T5, about 17 times smaller, so optimization makes too little progress within the budget. Using the relative step with the vision ρ and removing conflict adaptation changes the accuracy difference relative to ACTMat from -9.24 to +0.64 points on T5-base full fine-tuning, without text data or further tuning.

Conflict normalization introduces a separate problem. ViT has four main repeating types of mergeable layers, and its globally standardized scores have $\max |z_\ell| = 2.00$. T5 combines modules with different score distributions, including an encoder–decoder shift and extreme values from relative-attention-bias tables. Pooling them together gives $\max |z_\ell| = 6.19$. Because the score enters an exponent in Eq. (5), the same schedule spans damping multipliers of 10^6 on text, compared with 10^2 on vision. Standardizing within module families reduces this distortion and brings conflict Ridge to gains over ACTMat of +0.56 on full fine-tuning and +0.46 on LoRA (Appendix E).

Both scaling problems can be identified before measuring text accuracy. Normalizing within module families addresses the differences between modules more effectively than clipping z_ℓ, which recovers less performance. The appropriate pool still depends on the architecture: family-wise normalization costs half a point on the vision setting used to select the constants (Appendix E).

These corrections restore performance but do not establish a benefit from conflict adaptation. After correcting the pool, conflict Ridge remains within 0.39 points of uniform Ridge at the same damping in all four text settings on both metrics. Our pre-registered criterion required a gain of 1.0 point over uniform Ridge on the WSC-excluded metric. Conflict achieves only a tenth of that margin, and every

alternative score performs worse, so none meets the criterion. Increasing the number of tasks does not resolve this limitation. At twelve text tasks, the individual experts reach 70.23 against ACTMat's 66.36, yet every tested variant performs worse than ACTMat (Appendices E and H).

An exploratory sweep of α on text reaches +1.77, but we do not use this result to support our performance claims. The upstream T5 validation split contains only 32 examples, which we considered insufficient for tuning, so the sweep was conducted on the test set.

The transfer experiments make step-size scaling and normalization pools essential parts of the method's specification. Correcting them makes regularization competitive with the existing solver, but does not make the conflict schedule consistently useful on text. A related sensitivity appears in a published solver of a task-weighted version of the objective: rescaling its initialization improves accuracy by +16.26 points at 20 vision tasks, and the failure can likewise be predicted from a data-free measurement.

E TEXT: FULL TABLES AND THE TASK-COUNT STUDY

Two measurement caveats apply throughout this appendix. The upstream harness's validation split is the first 32 shuffled examples per task, so per-task accuracy moves in steps of about three points and cannot resolve the effects we are measuring; every text number we report is therefore on the test split. Separately, the WSC test split has 72 examples, so a single example moves it by 1.4 points. We report macro accuracy both with and without that task, and we treat the WSC-excluded metric as the one that carries weight.

Table 18 is the task-count study. We fine-tuned five further T5-base experts to test the hypothesis that the method fails on text only because seven tasks leave no headroom. The headroom does open, and the conclusion is nonetheless negative: every variant is worse than the closed form.

Table 18: Text at 12 tasks, T5-base full fine-tuning, test split.

Method	T5-base FFT, 12 tasks
Individual experts (ceiling)	70.23 (+3.88)
ACTMat (reference)	**66.36**
Uniform Ridge	65.83 (-0.53)
Conflict-guided Ridge	65.63 (-0.73)
Relative-step Muon	65.55 (-0.81)
Relative-step Muon + conflict	65.22 (-1.13)

Table 19: Transferring the vision recipe to T5, test split. Each text cell reports macro accuracy and macro accuracy excluding WSC, whose test split has 72 examples and moves in 1.4-point steps. Differences are against the ACTMat merge of the same cell. The ViT-B/16 column shows what each correction costs on the modality the constants were chosen on.

Method	ViT-B/16 T=8	T5-base FFT		T5-base LoRA		T5-large FFT		T5-large LoRA	
	macro	macro	no-WSC	macro	no-WSC	macro	no-WSC	macro	no-WSC
ACTMat (reference)	89.16	79.33	84.22	77.39	82.42	85.39	87.12	82.06	85.78
+ Adam solver (authors' code, lr 10^{-3}, 100 steps)	90.10 (+0.93)	79.07 (-0.26)	83.92 (-0.30)	78.24 (+0.86)	82.02 (-0.40)	85.54 (+0.16)	87.30 (+0.18)	83.34 (+1.29)	86.58 (+0.80)
Uniform Ridge (no conflict signal)	89.64 (+0.48)	79.53 (+0.20)	83.98 (-0.23)	77.80 (+0.41)	82.90 (+0.48)	85.17 (-0.21)	87.33 (+0.22)	83.31 (+1.26)	86.55 (+0.77)
Conflict Ridge, global pooling (published recipe)	90.58 (+1.41)	79.24 (-0.09)	83.88 (-0.33)	77.06 (-0.33)	82.72 (+0.30)	84.59 (-0.80)	87.12 (+0.00)	83.04 (+0.99)	86.70 (+0.92)
Conflict Muon, absolute step (published recipe)	90.67 (+1.51)	70.09 (-9.24)	73.90 (-10.32)	65.96 (-11.43)	68.15 (-14.27)	77.26 (-8.13)	79.95 (-7.17)	65.80 (-16.26)	70.75 (-15.03)
Uniform Muon, relative step ρ (M1)	90.12 (+0.96)	79.97 (+0.64)	84.27 (+0.05)	77.01 (-0.37)	81.98 (-0.43)	85.79 (+0.40)	87.35 (+0.23)	81.81 (-0.24)	85.27 (-0.52)
Conflict Ridge, family-wise pooling (M2)	90.07 (+0.90)	79.89 (+0.56)	84.17 (-0.05)	77.84 (+0.46)	82.72 (+0.30)	84.79 (-0.60)	87.12 (+0.00)	83.01 (+0.96)	86.43 (+0.65)

F CONSTANT SELECTION

Every constant was selected once, on the validation split of ViT-B/16 with 8 tasks, and then frozen for all backbones, task counts and modalities. The selection rule was fixed in advance: keep the default unless the validation grid improves on it by more than a fixed margin, and break ties toward the default. Table 20 lists every value and the grid it was chosen from. It is generated from the same dictionary the runs read, which is also the dictionary the constants hash is computed over,

so the table cannot disagree with what was run. Each result record includes a hash of the frozen configuration (6a0007ea81).

Table 20: Frozen constants, their values, and the validation grids they were selected against.

Symbol	Where	Value	Selected against
α	Ridge damping, Eq. (5a)	0.0005	5e-5, 1e-4, 5e-4, 1e-3
κ_{R}	Ridge exponent, Eq. (5a)	1	-1, 0, 0.5, 1, 1.5, 2
η_0	Muon base step, Eq. (5b)	0.0001	1e-5, 1e-4, 1e-3
κ_{M}	Muon exponent, Eq. (5b)	0.5	-1, -0.5, 0, 0.5, 1
K	Muon iterations	500	50, 100, 250, 500, 1000
μ	Muon momentum	0.95	kept at the optimiser's default
NS steps	Newton–Schulz iterations in $\mathrm{NS}(\cdot)$	5	kept at the optimiser's default
$\Delta^{(0)}$	Muon initialization	$mean$	mean, sum
N	stored layers, Section 4.3	24	12, 24, 36
τ	router threshold	0.2	fixed to the original method's value
top-k	router candidates	3	fixed to the original method's value
λ	authors' absolute Ridge (control)	0.001	their default; swept in Appendix F
lr	authors' Adam solver (control)	0.001	their default and 1e-5
iters	authors' Adam solver (control)	100	100, 300, 500

The baselines did not receive comparable treatment: they were run at their published defaults, while the constants above were chosen on a grid. Appendix F reports the one place where that matters most, the authors' absolute-damping control, at their constant and swept.

Three properties of that grid are worth recording. The exponent κ is unimodal at the frozen value for both solvers, and flipping its sign is destructive, which is the evidence that the signal is being used in the direction the method claims. The number of Muon iterations peaks at the frozen value rather than plateauing: the neighbouring grid points give 92.16 at 250 and 91.97 at 1000 against 92.35 at 500 on the validation split, so the drop at 1000 is larger than our 0.3-point threshold and K is a genuinely tuned constant, not an insensitive one.

We also ran the authors' absolute-damping control at the effective strength their constant implies on this backbone, and recovered their reported number to within 0.2 points, confirming that their variant and ours differ in the choice of scale rather than in the family of solvers.

That raises a question the single control cannot answer: is their -2.96 a bad *parameterization* or simply a bad *constant*? Their $\lambda = 10^{-3}$ corresponds to a relative strength tens of times our frozen α on this backbone, so the comparison as published confounds the two. We swept the absolute λ over five decades to separate them. Two results settle it. At $\lambda = 10^{-5}$ the same absolute variant *gains* +0.24 rather than losing three points, so the published number reflects a constant chosen too large. And running *our* relative Ridge at the effective strength their constant implies, forty times our frozen α, it is -0.96 to -0.19 *behind* their absolute variant on the three backbones.

So at matched strength the relative parameterization carries no accuracy advantage; if anything it is marginally worse. The case for scaling by $\sigma_{\max}$ is therefore *portability alone*: one α serves layers whose spectra differ by orders of magnitude and transfers across backbones and to T5 without re-tuning, which is what Section D exercises. We state this plainly because the natural reading of the -2.96 row is that absolute damping is broken, and it is not.

F.1 What a reimplementation needs that the method section does not say

The equations in Section 4 fix the method but leave several choices implicit. They are all in the released code; we state them here so the paper is self-contained.

The orthogonalization. $\mathrm{NS}(\cdot)$ is the quintic Newton–Schulz iteration with coefficients $(a, b, c) = (3.4445, -4.7750, 2.0315)$, applied for five steps to $M/\|M\|_F$, transposing first when the matrix is taller than it is wide. Five steps do not converge to the exact polar factor and are not meant

to: the iteration is the same one the optimiser we borrow from uses. Its output is scaled by $0.2\sqrt{\max(d_{\text{out}}, d_{\text{in}})}$.

The step schedule. The decay in Eq. (5b) is $c_k = \frac{1}{2}\big(1 + \cos(\pi k/K)\big)$ over $k = 0, \ldots, K-1$, so $c_0 = 1$ and the last step is almost zero. Because c_k depends on K, changing the iteration count changes the whole trajectory and not just its length, which is why K is a selected constant and not a stopping criterion. The iteration carries momentum μ and no weight decay (Table 20).

Module families. On a CLIP vision tower the families are the four repeating matrix types, attention input projection, attention output projection, and the two MLP projections, with everything else pooled as a remainder. On T5 they are the encoder and decoder self-attention, the cross-attention, the two feed-forward towers, and the relative-attention-bias tables as a family of their own. That last family is the point: the bias tables are $32 \times n_{\text{heads}}$ and sit six standard deviations out under a global pool, which is what turns the global z_ℓ into an encoder indicator rather than a per-layer signal. A family with one member receives $z_\ell = 0$.

Everything else. z_ℓ is not clipped; clipping was tried and recovers less than family-wise pooling (Appendix H). $C_t^{1/2}$ is never formed: the objective and the residual are evaluated through C_t directly. Ties in the top-N conflict ranking are broken by the layer's position in the state dictionary, which is deterministic. In the input-adaptive setting only two-dimensional parameters are routed; one-dimensional parameters and biases take their static merged value and are never stored per task, so they contribute nothing to the store percentages we report.

F.2 Where an iterative solver starts matters more than which one it is

WUDI-Merging (Cheng et al., 2025) minimises a task-weighted version of the ACTMat objective. Writing $\|(W - \Delta_t)\Delta_t^\top\|_F^2 = \|(W - \Delta_t)C_t^{1/2}\|_F^2$ with $C_t = \Delta_t^\top \Delta_t$, its loss is Eq. (1) under the same data-free proxy, weighted per task by $1/\|\Delta_t\|_F^2$ and solved by Adam from $W^{(0)} = \sum_t \Delta_t$ rather than in closed form. The ACTMat repository ships an Adam variant that differs in exactly two respects: it starts at the mean and it does not weight. Holding the learning rate and the step budget fixed at the published values isolates those two choices (Table 21).

Table 21: Two iterative solvers of the (task-weighted) ACTMat objective on ViT-B/16, both Adam at 10^{-5} for 300 steps, differing only in where they start and whether each task is weighted. Differences are against the ACTMat closed form of the same cell.

Configuration	8 tasks	14 tasks	20 tasks
Start at $\sum_t \Delta_t$, weighted (published)	89.13 (-0.03)	81.24 (-3.67)	65.33 (-16.01)
Start at $\sum_t \Delta_t$, unweighted	89.18 (+0.02)	81.02 (-3.90)	64.06 (-17.27)
Start at the mean, weighted	89.74 (+0.58)	85.33 (+0.41)	81.59 (+0.25)
Start at the mean, unweighted	**90.01 (+0.85)**	**85.76 (+0.84)**	**81.63 (+0.29)**

The starting point accounts for essentially the whole difference, and it is the term that scales: moving from the sum to the mean is worth +0.61 at 8 tasks, +4.09 at 14 and +16.26 at 20. The weighting never exceeds 0.43 at any task count and is not monotone, so it sits at or below our 0.3-point threshold (Section 6). At 20 tasks the published configuration falls 16 points below the closed form while the same algorithm from the mean is slightly above it.

This is predictable without measuring any accuracy, which is why it belongs in this paper rather than in a footnote about someone else's hyperparameters. The norm of $\sum_t \Delta_t$ grows with the task count while the mean's does not, so the sum starts progressively further from the optimum: 1.02, 1.19 and 1.40 times $\|\Delta^\star\|_F$ at 8, 14 and 20 tasks, against a flat 0.98 for the mean (`scripts/init_distance.py`, averaged over 24 mergeable layers). A quantity computable from the task vectors alone orders a sixteen-point accuracy collapse.

Two limitations. This is our port of the published recipe onto our checkpoints, heads and layer set, and that work reports stronger numbers on its own pipeline, so the finding is about configuration transfer rather than a refutation. And the mechanism is not established: Adam's step is roughly the

learning rate per coordinate regardless of gradient scale, so its furthest travel is about $\mathrm{lr} \cdot K \cdot \sqrt{d}$, which exceeds the extra distance in every cell here. The attribution rests on Table 21, not on a budget argument. What we take from it is the same lesson as Section D: a solver's starting point and step size have to be stated relative to the scale of the thing being solved for.

G REPLICATION ON INDEPENDENTLY FINE-TUNED EXPERTS

All results in the main text merge one public set of checkpoints. To check that the ordering of methods is a property of the method rather than of that particular set, we fine-tuned an independent set of ViT-B/16 experts on the same eight tasks with a different seed and repeated the headline comparison with the constants unchanged. The qualitative picture reproduced: both conflict-guided solvers improved on the closed form, and the truncation baseline remained below the closed form. The input-adaptive comparison also reproduced, with the half-store gate the same short distance below our MASS port as on the public checkpoints. Two things did not reproduce exactly, and we record them because they calibrate how finely these results can be read. Every gain is smaller on the second set, whose experts are themselves weaker: the conflict Ridge gains +1.23 there against +1.41 on the first, and the conflict Muon +1.20 against +1.51. And the Ridge and Muon variants swap rank, the conflict Muon being ahead by 0.09 on the first set and behind by -0.03 on the second. Differences of that size between our own variants are therefore not resolvable, which is why Section 6 declines to rank them. This is a replication across expert sets, not an estimate of run-to-run variance: merging is deterministic, and the paired analysis in Table 8 is what speaks to whether a gain is carried by the benchmark.

The null-score control on the second expert set. The control of Table 9 was repeated in full on the independently fine-tuned experts, since a margin under half a point measured on one set of checkpoints is exactly the kind of result that should not be trusted until it is repeated.

The margin replicates: conflict beats the best free descriptor by +0.42 there against +0.55 on the published experts. More informative is what happens to the descriptors themselves. Conflict's gain moves 0.19 between the two expert sets, from +1.41 to +1.23. The four descriptors move by up to 2.70, and their ordering changes: layer type is the strongest alternative on the published experts and block depth on the second set, while task-vector norm, the best alternative on three of the four original cells, falls to -0.38.

We had drafted a practical aside saying that a practitioner unwilling to compute a residual could take the task-vector norm and keep most of the gain. The replication contradicts it, and it has been removed. That is the concrete value of the second expert set: not that it confirmed the headline, but that it caught a recommendation which would have been wrong.

H CONDITIONING IS NOT THE SIGNAL

The T5 sweeps in this appendix are exploratory: the upstream validation split has only 32 examples, so they run on test, and they never reselect a frozen constant. Section 5.3 states that the conditioning of $\sum_t C_t$ does not explain where regularization helps. This appendix records the evidence, because the hypothesis is a natural one and the negative result saves the reader from repeating it.

We instrumented every mergeable layer with effective rank, numerical rank, condition number and the fraction of the spectrum falling below the damping the method applies, and compared these against the per-layer conflict. Conflict *anti*-correlates with degeneracy on both modalities, and the evidence is the sign rather than the size: the rank correlation between conflict and normalised effective rank is positive in all 11 cells we instrumented, ranging from +0.30 to +0.97, and from +0.49 to +0.88 on the full fine-tuning cells. The highest-conflict layers are therefore the best-conditioned ones. The layers that actually pay in the leave-one-layer-in probe have no near-null spectrum at all.

We then replaced the conflict score in Eq. (5) by each conditioning statistic in turn, holding the damping law, its strength and the standardization pool fixed, and evaluated on the strongest text cell. None beat a uniform Ridge: the best conditioning-derived score trails the uniform baseline, and a variant that damps only the most degenerate quarter of each family is the worst of all.

The same table disposes of our own score on this cell, and we report it in the same breath. The margin we pre-registered for declaring any per-layer score useful here was a 1.0-point improvement over the uniform Ridge on the WSC-excluded metric. The conflict score achieved +0.13, and on macro accuracy -0.09 against the uniform Ridge. The gate therefore failed for the hypothesis under test and for our own, and no claim about text in this paper rests on a per-layer score of either kind.

I Comparison protocol against the routed baseline

Our port of MASS matches the original on the axes that define the method: the same backbones, the same task suites, the same test split, the same first-pass merge, the same rank fraction per task, the same projection router at the same layer with the same threshold and top-k, the same orthogonalised re-merge, and the same head-selection rule. We audited each of these against the published description before running any comparison.

Three differences remain, and all three affect our port and our own method identically, which is why the comparison within our columns is exact even though the absolute numbers differ from the published ones. First, the original work re-fine-tuned one dataset's expert to correct an image-orientation mismatch with a near-duplicate dataset; we use the released expert. Second, it merges the label spaces of two overlapping datasets into one classification head; we keep them separate. Both choices matter only when the router must disambiguate near-duplicate tasks, which is why their effect is confined to the 20-task suite and why we draw no method conclusions there. Third, the original applies a redundancy filter that drops near-duplicate task matrices before the first-pass merge; our port omits it, and the original's own ablation puts its value at a few tenths of a point.

A fourth difference is not ours to control: the published expert ceiling is one to three points below what we measure on the same checkpoint source. Normalized accuracy divides by that ceiling, but this does not make it comparable across papers, so every comparison we draw is against our own port.

J The routed re-merge in factor space

Section 4.3 re-merges a routed subset with the Ridge of Eq. (5a). Doing so densely would cost a d_{in}^3 solve per layer per input, which is not affordable at inference. Since each stored update is already a truncated factorization $\Delta_t = U_t S_t V_t^\top$ of rank r, the solve can be pushed through the factors. With $C_t = V_t S_t^2 V_t^\top$, concatenate horizontally over the routed tasks Ω: $B = [\, V_t S_t \,]_{t\in\Omega} \in \mathbb{R}^{d_{\text{in}} \times |\Omega| r}$ and $Y = [\, U_t S_t^2 \,]_{t\in\Omega} \in \mathbb{R}^{d_{\text{out}} \times |\Omega| r}$, so that $\sum_{t\in\Omega} C_t = BB^\top$, $\sum_{t\in\Omega} \Delta_t C_t = YB^\top$, and

$$\Delta_\Omega \;=\; Y\big(B^\top B + \lambda_\ell I\big)^{-1} B^\top. \tag{6}$$

This is exact for the retained, truncated factors: the inverse is taken on an $|\Omega| r \times |\Omega| r$ matrix rather than on $\sum_t C_t$. Forming $B^\top B$ costs $O(|\Omega|^2 r^2 d_{\text{in}})$ and factorizing it $O(|\Omega|^3 r^3)$; applying Δ_Ω to an input without forming it costs $O(|\Omega| r (d_{\text{in}} + d_{\text{out}}))$. For the top-$k$ routing of Section 5.4 this is two orders of magnitude below the dense solve. We verify the identity numerically against a dense float64 solve in the released tests, agreeing to within 2×10^{-6}.

K Provenance of external code and data

Baseline operators are ported from the original authors' repositories rather than reimplemented, at pinned commits recorded in the release. This includes the closed-form estimator we use as the reference, the two unregularized variants we report as controls, the truncation and spectrum-flattening baselines, the regression-based merge that uses data, the LoRA-alignment baselines, and the routing components of the input-adaptive baseline. Dataset loaders and the zero-shot heads follow the conventions of prior work on this benchmark so that accuracies are comparable to published numbers. The release records, for each external component, its source, its commit and the file in our library that vendors it.