What Changes in Hidden States When Model Merging Hurts Task Performance?

Canasai Kruengkrai[1] **Koichiro Yoshino**[1,2]
[1]Guardian Robot Project, RIKEN
[2]The University of Osaka
canasai.kruengkrai@riken.jp, yoshino.koichiro.es@osaka-u.ac.jp

Abstract

Model merging is a practical way to combine several fine-tuned models, but merged models can lose performance relative to their task-specific counterparts. Task accuracy shows the degradation, but not how merging changes the model's internal representations. We study this question through *hidden-state distortion*, which measures the difference between a merged model's hidden states and those of the corresponding task-specific reference model on the same examples. Across two model families, six tasks, and three merge methods, we find that the distortion is highly anisotropic: much of the merged-reference change is concentrated in a small number of directions. Removing distortion components along dominant directions often improves accuracy more than removing those along unrelated directions. When we jointly fit a single distortion basis across tasks for each model family, this removal still improves accuracy, suggesting that the tasks share some useful distortion structure. In addition, the distortion score provides a label-free criterion for choosing among candidate task-arithmetic merge scales, matching supervised selection for both model families.

1 Introduction

Model merging combines multiple fine-tuned models into a single model (Wortsman et al., 2022; Matena and Raffel, 2022; Ilharco et al., 2023; Jin et al., 2023). It can reuse existing checkpoints and reduce the need for joint multitask training (Zhou et al., 2024; Yang et al., 2026). However, the merged model can perform worse than the task-specific models used to construct it. Task accuracy tells us how much performance is lost, but not what changes inside the model.

Prior work relates merge degradation to interference among task-specific updates and to structure in the parameters being combined (Yadav et al., 2023; Daheim et al., 2024; Cao et al., 2026; Lee et al., 2025; Cheng et al., 2025). Other work uses representation information to construct or correct merged models (Yang et al., 2024; Wei et al., 2025; Chen et al., 2025; Zhang et al., 2026). These studies help explain or reduce merge conflict, but they do not directly show what changes in the model's internal computation after merging.

In this paper, we study how merging changes the model's hidden states, which provide an intermediate view between the merged parameters and the model's predictions. For each task, we use its task-specific source model as the reference and compare it with the merged model on the same examples. At each analyzed position, we take the difference between the merged and reference hidden states. We call this difference *hidden-state distortion*. It gives us a direct representation of the hidden-state change introduced by merging for that task.

We first ask whether post-merge distortion is spread across many hidden-state directions or concentrated in a small number of directions. We pool the position-level distortions across examples and use singular value decomposition (SVD) to identify their dominant directions. If much of the distortion lies along a few directions, the next question is whether those directions matter for prediction. We test this by removing their contribution from the merged hidden states and comparing the result with control interventions. We also ask whether a basis learned across tasks captures useful shared distortion structure. Finally, we examine whether the magnitude of the distortion can help choose among task-arithmetic merge scales without gold labels.

We evaluate two model families, six tasks, and three merge methods. Under task arithmetic (Ilharco et al., 2023), the top 10 distortion directions explain 74.1% of the distortion variance for T5-large and 72.3% for Phi-3.5, while the matched-norm and shuffled controls are much more diffuse. Removing the distortion components along these directions improves average accuracy by 7.8 points

Proceedings of the 15th International Joint Conference on Natural Language Processing and the 5th Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics, pages 4021–4034
November 6-10, 2026 ©2026 Association for Computational Linguistics

for T5-large and 3.3 points for Phi-3.5, whereas removing components along a random subspace of the same size changes accuracy by only 1.0 and –0.1 points. When we fit a single distortion basis across all six tasks, removing distortion along its dominant directions still improves accuracy, suggesting that some useful distortion structure is shared across tasks. For task arithmetic, using the distortion score to choose the merge scale gives the same choice as supervised accuracy for both model families.

Our contributions are as follows:

1. We study post-merge degradation through merged-reference hidden-state distortion, which measures the change between a merged model and its task-specific reference model.[1]

2. We find that post-merge distortion is strongly anisotropic and that some useful distortion structure is shared across tasks.

3. We show through controlled hidden-state interventions that removing dominant distortion components often improves task accuracy more than removing unrelated components.

4. We show that merged-reference distortion can serve as a label-free criterion for choosing among task-arithmetic merge scales.

2 Related work

Merge degradation. A recurring problem in model merging is interference among task-specific updates. Yadav et al. (2023) reduce conflicts in task-vector coordinates before aggregation. Later work studies degradation through gradient mismatch (Daheim et al., 2024), parameter interference addressed through sharpness-aware fine-tuning (Lee et al., 2025) or data-free task-vector adjustment (Cheng et al., 2025), and task-level model-merging collapse (Cao et al., 2026).

Recent work also asks why some source models or tasks are more affected by merging than others. Rahamim et al. (2026) define mergeability and find that models fine-tuned on knowledge already handled well by the base model tend to retain that knowledge better after merging. Ramesh et al. (2026) characterize cross-task interference through differences in behavior between the merged model and its constituent models on their respective tasks, and adapt the constituent models to reduce this interference before merging.

Structure-guided merging. Several methods use structure in task updates or representations to reduce interference during merging. One line of work focuses on the task updates themselves, using tangent-space linearization (Ortiz-Jimenez et al., 2023), task-parameter subspaces (Tam et al., 2024), singular-vector structure (Gargiulo et al., 2025), and common and task-specific subspaces (Marczak et al., 2025). Zhang and Zhou (2025) use orthogonal subspaces to reduce interference among LoRA updates, while Li et al. (2026) restrict merging to a low-dimensional subspace and Fang et al. (2025) decompose neuron-level task updates into complementary subspaces. Other work uses representation information more directly. Yang et al. (2024) modify post-merge representations through representation surgery, while Wei et al. (2025) extend this approach by modeling the representation bias between merged and individual models with a probabilistic module. Chen et al. (2025) use representation bias to guide weight-unit matching and layer-wise merging coefficients.

Zhang et al. (2026) use unlabeled calibration data to learn layer-wise merge coefficients by aligning the merged model's hidden states and logits with those of the corresponding experts. The merged–expert representation difference therefore serves directly as a signal for constructing the merge. Our analysis starts from an already-constructed merge and asks whether this hidden-state difference is concentrated in particular directions and whether those directions are related to model behavior.

Representation geometry. Prior work compares neural representations through both functional alignment and geometric structure. Bansal et al. (2021) use model stitching to test whether representations from different networks are functionally compatible. Valeriani et al. (2023) characterize the geometry of hidden representations in large transformers, while Doimo et al. (2024) examine how that geometry changes with few-shot learning and fine-tuning. Huh et al. (2024) study shared representational structure across models, and Park et al. (2024) develop a geometric framework for linear representations of concepts in language models. Our work instead studies the geometry of the representation change induced by model merg-

[1] Code is available at https://github.com/riken-grp/merged-reference-distortion.

ing rather than the geometry of the representations themselves.

Activation-space intervention. Other work tests the behavioral role of representation structure by intervening directly on model activations. Li et al. (2023) identify attention-head directions associated with truthfulness and shift activations along those directions during inference. Rimsky et al. (2024) derive steering directions from contrastive activation differences and use them to modify model behavior. Wu et al. (2024) instead learn interventions directly in representation space through representation fine-tuning. Our intervention targets directions defined by the hidden-state distortion between the merged and task-specific reference models and measures how removing the distortion components along those directions changes prediction behavior.

3 Preliminaries

3.1 Reference and merged models

We consider a set of tasks $\mathcal{T}$, with one task-specific model for each task $t \in \mathcal{T}$, all obtained by fine-tuning the same pretrained model. A merge method combines these models into a single model. We represent the merge operation abstractly as:

$$\boldsymbol{\theta}_{\text{merge}} = \text{Merge}(\boldsymbol{\theta}_0, \{\boldsymbol{\theta}_t\}_{t\in\mathcal{T}}), \tag{1}$$

where $\boldsymbol{\theta}_0$ denotes the pretrained base-model parameters and $\boldsymbol{\theta}_t$ the task-specific parameters used for task t. We keep the merge operator abstract so that the same representation analysis applies across merge methods. Appendix A gives the specific merge definitions.

To study what changes for a particular task t after merging, we use its task-specific source model as the reference. This model represents the task before its parameters are combined with those of the other tasks. We compare its hidden states with those of the merged model on examples from task t.

3.2 Position-level hidden-state distortion

In our evaluation setup, each input is paired with a set of candidate strings. For each input–candidate pair, we run the task-specific reference model and the merged model and compare their hidden states at the same candidate-scoring positions. Let $\mathbf{H}_{\text{ref}} \in \mathbb{R}^{L\times d}$ and $\mathbf{H}_{\text{merge}} \in \mathbb{R}^{L\times d}$ denote the resulting sequences of hidden states, where L is the number of analyzed positions and d is the hidden dimension.

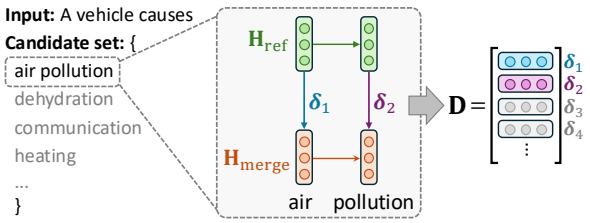


Figure 1: Construction of the merged-reference distortion matrix. For one input, we compare the task-specific reference and merged hidden states for each candidate string. Their differences at the analyzed positions give position-level distortion vectors. Pooling these vectors across inputs and candidate strings forms the matrix $\mathbf{D}$.

At one analyzed position, let $\mathbf{h}_{\text{ref}} \in \mathbb{R}^d$ and $\mathbf{h}_{\text{merge}} \in \mathbb{R}^d$ denote the corresponding hidden states from the two models. We define the position-level distortion as:

$$\boldsymbol{\delta} = \mathbf{h}_{\text{merge}} - \mathbf{h}_{\text{ref}} \in \mathbb{R}^d. \tag{2}$$

This vector measures how the representation at that position changes after merging. We work at the position level because merging can have different effects across candidate-scoring positions.

3.3 Distortion matrix and principal directions

Each analyzed position gives one distortion vector $\boldsymbol{\delta}$. To examine whether these local changes share common structure, we pool them across examples and candidate strings into the merged-reference distortion matrix $\mathbf{D}$. Figure 1 illustrates this construction. We define:

$$\mathbf{D} = \begin{bmatrix} \boldsymbol{\delta}_1^\top \\ \vdots \\ \boldsymbol{\delta}_N^\top \end{bmatrix} \in \mathbb{R}^{N\times d}, \tag{3}$$

where N is the total number of pooled positions. Each row of $\mathbf{D}$ is therefore a change in hidden state rather than a hidden state itself.

The distortion vectors may contain a common overall shift. We want the principal directions to describe how the distortions vary around this shift, so we first compute the mean distortion:

$$\boldsymbol{\mu} = \frac{1}{N}\sum_{i=1}^{N} \boldsymbol{\delta}_i \in \mathbb{R}^d, \tag{4}$$

and center the matrix:

$$\mathbf{D}_c = \mathbf{D} - \mathbf{1}\boldsymbol{\mu}^\top \in \mathbb{R}^{N\times d}, \tag{5}$$

where $\mathbf{1} \in \mathbb{R}^N$ is the all-ones vector.

We apply SVD (Golub and Van Loan, 2013) to the centered matrix:

$$\mathbf{D}_c = \mathbf{L}\mathbf{\Sigma}\mathbf{R}^\top, \tag{6}$$

where $\mathbf{L} \in \mathbb{R}^{N\times r}$, $\mathbf{\Sigma} \in \mathbb{R}^{r\times r}$, and $\mathbf{R} \in \mathbb{R}^{d\times r}$, with $r = \mathrm{rank}(\mathbf{D}_c)$. The diagonal entries of $\mathbf{\Sigma}$ are the singular values $\sigma_1, \ldots, \sigma_r$, and the columns of $\mathbf{R}$ give orthonormal directions in hidden-state space. Let $\mathbf{r}_j \in \mathbb{R}^d$ denote the j-th column of $\mathbf{R}$. We collect the first k directions into:

$$\mathbf{U}_k = \begin{bmatrix}\mathbf{r}_1 & \cdots & \mathbf{r}_k\end{bmatrix} \in \mathbb{R}^{d\times k}. \tag{7}$$

The columns of $\mathbf{U}_k$ form an orthonormal basis for the top-k distortion subspace.

4 Distortion subspace analysis

4.1 Measuring anisotropy

The principal directions tell us where the distortion varies most, but not how concentrated that variation is. We ask whether the distortion is approximately isotropic, with variation spread across many directions, or concentrated in only a few. We refer to the latter concentration as *anisotropy* (Ethayarajh, 2019).

For the j-th singular value σ_j of the centered distortion matrix $\mathbf{D}_c$, we define the variance associated with that direction as:

$$\nu_j = \frac{\sigma_j^2}{N-1}. \tag{8}$$

We normalize these values to obtain the explained variance ratio (Jolliffe and Cadima, 2016):

$$\rho_j = \frac{\nu_j}{\sum_{\ell=1}^{r}\nu_\ell}. \tag{9}$$

Here, ρ_j gives the fraction of total distortion variance associated with the j-th principal direction.

To measure how quickly the variance accumulates in the leading directions, we define the cumulative explained variance of the top k directions as:

$$c_k = \sum_{j=1}^{k}\rho_j. \tag{10}$$

A large c_k for small k means that a small number of directions accounts for much of the distortion variance. We also use the participation ratio (Gao et al., 2017):

$$\mathrm{PR} = \frac{\left(\sum_{j=1}^{r}\nu_j\right)^2}{\sum_{j=1}^{r}\nu_j^2}. \tag{11}$$

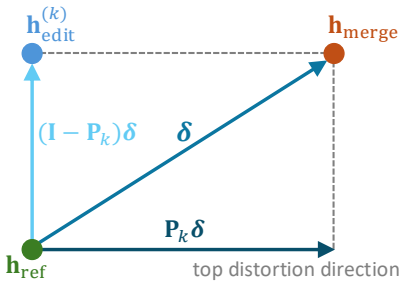


Figure 2: Hidden-state edit at one analyzed position (schematic). The distortion vector $\boldsymbol{\delta}$ is decomposed into its projection $\mathbf{P}_k\boldsymbol{\delta}$ onto the top-k distortion subspace and the residual $(\mathbf{I}-\mathbf{P}_k)\boldsymbol{\delta}$. The edited state $\mathbf{h}_{\text{edit}}^{(k)}$ is formed by adding this residual to $\mathbf{h}_{\text{ref}}$.

While c_k measures how much variance is captured by a chosen number of leading directions, PR summarizes the effective number of directions contributing to the distortion. Lower PR corresponds to stronger anisotropy.

4.2 Removing distortion along dominant directions

If the distortion is strongly anisotropic, a further question is whether its dominant directions affect the model's predictions. We test this by removing the component of the merged-reference difference that lies in the top-k distortion subspace. Using the basis $\mathbf{U}_k$, we define the projection matrix:

$$\mathbf{P}_k = \mathbf{U}_k\mathbf{U}_k^\top \in \mathbb{R}^{d\times d}. \tag{12}$$

At one analyzed position, $\mathbf{P}_k\boldsymbol{\delta}$ is the projection of the distortion onto the top-k subspace, and $(\mathbf{I}-\mathbf{P}_k)\boldsymbol{\delta}$ is the remaining component. We construct the edited hidden state as:

$$\mathbf{h}_{\text{edit}}^{(k)} = \mathbf{h}_{\text{ref}} + (\mathbf{I}-\mathbf{P}_k)\boldsymbol{\delta} \in \mathbb{R}^d. \tag{13}$$

The edit retains the part of the merged-reference change outside the dominant distortion subspace. Figure 2 illustrates this construction at a single position.

During evaluation, we replace the merged hidden states at the analyzed positions with their edited counterparts and recompute the model's predictions. If removing the dominant distortion component changes task accuracy, this provides evidence that these directions matter for model behavior.

The analysis above estimates the distortion basis separately for each task. To test whether distortion structure is shared across tasks, we also construct a single basis from pooled training-split distortions.

We combine the distortion vectors from all tasks, center the pooled matrix, and compute its principal directions with the same SVD procedure. We then apply the same top-k shared basis to every task using the intervention above.

4.3 Functional and structural controls

A change after top-k removal could occur simply because the intervention removes the distortion component lying in a k-dimensional subspace. As a functional control, we compare against a random k-dimensional subspace. Let $\widetilde{\mathbf{U}}_k \in \mathbb{R}^{d\times k}$ denote a random orthonormal basis, and define:

$$\widetilde{\mathbf{P}}_k = \widetilde{\mathbf{U}}_k \widetilde{\mathbf{U}}_k^\top. \tag{14}$$

We use the same hidden-state edit as in Section 4.2, replacing $\mathbf{P}_k$ with $\widetilde{\mathbf{P}}_k$. This comparison tests whether removing distortion projected onto a random subspace of the same dimensionality produces a similar change in model behavior to removing its projection onto the dominant distortion subspace.

For a structural control, we replace each pooled distortion vector with a random vector of the same norm. This preserves the distortion magnitude at each position while removing consistent directional alignment across the pooled vectors. If strong anisotropy remains under this control, it could be explained by the distribution of distortion magnitudes rather than shared directions.

We also construct a shuffled control by independently permuting the coordinates within each distortion vector. This preserves the coordinate values of each vector but disrupts their alignment across positions. Comparing the resulting spectrum with the original distortion matrix tests whether the observed anisotropy depends on consistent directional structure across the pooled distortions.

4.4 Label-free merge selection with distortion

So far, we have used merged-reference distortion to analyze and intervene on a chosen merged model. We next ask whether the same signal can also help choose among candidate merge scales. For task t and merge scale λ, let $\mathbf{D}_t^{(\lambda)}$ denote the merged-reference distortion matrix computed on a small selection subset, and let $\mathbf{H}_{\mathrm{ref},t}$ denote the corresponding pooled reference hidden states. The rows of $\mathbf{H}_{\mathrm{ref},t}$ are aligned with those of $\mathbf{D}_t^{(\lambda)}$. We define the task-normalized distortion score as:

$$s_t(\lambda) = \frac{\left\|\mathbf{D}_t^{(\lambda)}\right\|_F^2}{\|\mathbf{H}_{\mathrm{ref},t}\|_F^2}. \tag{15}$$

The numerator measures the size of the merged-reference hidden-state change, while the denominator normalizes by the scale of the task-specific reference representations. A smaller value means that the candidate merged model stays closer to the reference representations for that task.

We average these task-level scores across tasks and select the merge scale with the lowest value:

$$\lambda^\star = \arg\min_\lambda \frac{1}{|\mathcal{T}|}\sum_{t\in\mathcal{T}} s_t(\lambda). \tag{16}$$

The selection procedure uses task inputs together with hidden states from the candidate merged models and the corresponding task-specific reference models. It does not use gold output labels or task accuracy. The reference models are needed only during scale selection. Once a scale is selected, the resulting merged model is used normally, without reference models at inference time.

5 Experimental setup

Tasks, splits, and evaluation. Following Yadav et al. (2023), we use six tasks from their NLP model-merging setup: PAWS (Zhang et al., 2019), QASC (Khot et al., 2020), QuaRTz (Tafjord et al., 2019), WikiQA (Yang et al., 2015), WinoGrande (Sakaguchi et al., 2021), and WSC (Levesque et al., 2012). For each input, the evaluation protocol provides a finite set of candidate output strings. The model scores each candidate by its conditional log probability and predicts the candidate with the highest score. Because the official test splits of QASC, WinoGrande, and WSC do not provide public gold labels, we use the validation splits for final evaluation across all six tasks and report accuracy. Merge hyperparameters are selected using fixed subsets of 200 examples per task sampled from the training split.

Models and merge methods. We use two model families that differ in both architecture and how the task-specific models are adapted. T5-large (Raffel et al., 2020) is a 770M-parameter encoder–decoder model. We fine-tune one full model for each task and merge the resulting model weights. Phi-3.5-mini-instruct (Abdin et al., 2024) is a 3.8B-parameter causal language model. We fine-tune one LoRA adapter (Hu et al., 2022) for each task and merge the adapter weights. We refer to this model as Phi-3.5. For both model families, we evaluate simple averaging (**AVG**; Choshen et al.,

Model	Method	PAWS	QASC	QuaRTz	WikiQA	WinoGrande	WSC	Avg.	Drop
T5-large	Task-specific	95.2	99.8	89.3	96.0	66.4	76.0	87.1	–
	AVG	44.2	96.7	75.3	95.2	49.9	60.6	70.3	16.8
	TA	70.5	98.2	83.9	90.3	62.5	64.4	**78.3**	8.8
	TIES	70.9	95.9	78.4	87.6	64.8	68.3	77.6	9.5
Phi-3.5	Task-specific	94.8	83.6	92.2	96.6	82.6	80.8	88.4	–
	AVG	82.5	78.7	92.5	86.8	73.2	62.5	79.4	9.1
	TA	88.2	79.8	92.5	91.3	75.8	68.3	**82.6**	5.8
	TIES	88.6	79.9	89.1	91.1	73.6	63.5	81.0	7.5

Table 1: Validation accuracy across six tasks. Task-specific denotes the separately fine-tuned reference model for each task. Avg. is the mean accuracy across tasks. Drop is the difference between the task-specific and merged average accuracy. Bold marks the best average accuracy among the three merged models for each model family.

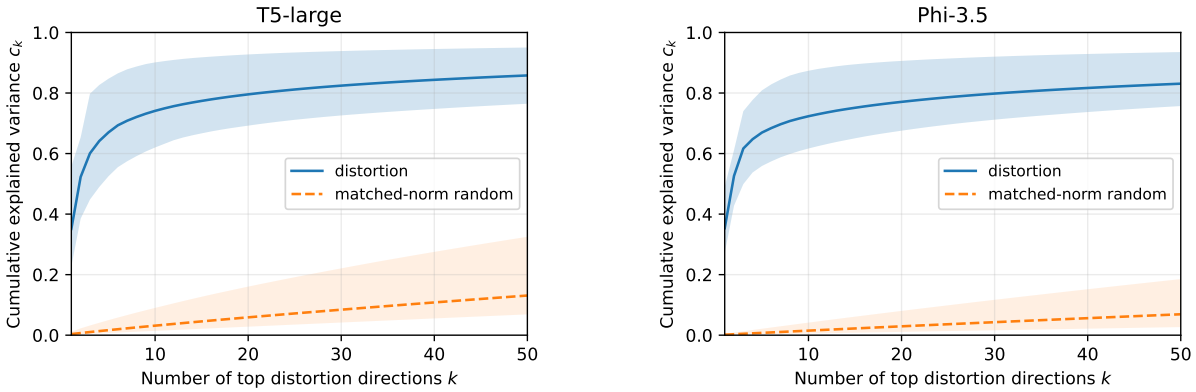


Figure 3: Cumulative explained variance c_k of the merged-reference distortion under TA on the validation split, averaged over six tasks. Shaded regions show the minimum and maximum across tasks. The matched-norm random control replaces each distortion vector with a random vector of the same norm.

2022; Wortsman et al., 2022), task arithmetic (**TA**; Ilharco et al., 2023), and TRIM, ELECT SIGN & MERGE (**TIES**; Yadav et al., 2023). Appendix A describes the merge methods in detail.

Representation extraction. For each input, we pair it with every candidate string and run a forward pass on the resulting input–candidate pair. We collect hidden states at the positions where candidate tokens are scored. For T5-large, these are final-layer decoder states at the output-token positions of the candidate string. For Phi-3.5, they are final-layer hidden states used to predict the candidate tokens. We extract states for all candidate strings, rather than only the gold candidate, so that the representation analysis follows the same candidate-scoring process used for evaluation.

Merge-scale selection. To test whether distortion can guide merge-scale selection, we compare it with prediction confidence, reference agreement, and supervised accuracy. All four criteria use the same candidate TA scales and the same fixed 200-example training subset for each task. Distortion uses the normalized merged-reference distortion score defined in Section 4.4. Confidence is the mean maximum candidate probability after normalizing the merged model's candidate scores, averaged over examples and tasks. Reference agreement is the fraction of examples on which the merged and task-specific reference models select the same candidate, averaged across tasks.

Supervised selection uses average task accuracy on the same examples. We select the scale with the lowest distortion or the highest confidence, agreement, or supervised accuracy. Distortion, confidence, and agreement do not use gold labels.

Appendix B provides additional fine-tuning, merge-hyperparameter, control construction, selection, and compute details.

6 Results

6.1 How much does merging hurt task performance?

Before analyzing the hidden states, we first quantify how much task performance is lost after merging. Table 1 compares each merged model with

Model	Top-1	Top-5	Top-10	Matched-norm-10	Shuffled-10	PR
T5-large	35.1 (22.7–55.7)	67.0 (52.5–84.7)	74.1 (62.0–90.1)	3.1 (1.4–9.0)	3.1 (1.4–8.9)	7.0 (3.1–11.3)
Phi-3.5	35.4 (27.1–49.1)	66.9 (55.9–80.9)	72.3 (61.7–87.4)	1.5 (0.5–4.2)	1.5 (0.5–4.1)	6.3 (3.9–9.4)

Table 2: Anisotropy of the merged-reference distortion under TA on the validation split. Top-1, Top-5, and Top-10 report cumulative explained variance, averaged over six tasks. Matched-norm-10 and Shuffled-10 report the corresponding Top-10 values for the two structural controls. Parentheses show the minimum and maximum across tasks. PR denotes participation ratio.

Task	Top-10 ΔAcc.	Random-10 ΔAcc.
T5-large		
PAWS	+24.0	+2.6
QASC	+0.5	+0.2
QuaRTz	+3.6	+0.3
WikiQA	+5.6	+1.0
WinoGrande	−0.2	+0.1
WSC	+13.5	+1.9
Avg.	**+7.8**	**+1.0**
Phi-3.5		
PAWS	+6.6	+0.4
QASC	+2.4	−0.3
QuaRTz	+0.5	+0.3
WikiQA	+4.4	+0.1
WinoGrande	−0.1	+0.1
WSC	+5.8	−1.0
Avg.	**+3.3**	**−0.1**

Table 3: Hidden-state intervention results for TA on the validation split. ΔAcc. is the change in accuracy relative to the unedited merged model. Top-10 removes the distortion component in the top-10 subspace, while Random-10 removes the component in a random 10-dimensional subspace.

Task	ΔAcc.	ΔGold	ΔInc.	ΔMargin
T5-large				
PAWS	+24.0	+1.922	−2.292	+4.214
WinoGrande	−0.2	+0.094	+0.093	+0.001
Phi-3.5				
PAWS	+6.6	+0.199	−1.790	+1.989
WinoGrande	−0.1	+0.624	+0.629	−0.005

Table 4: Score decomposition for PAWS and WinoGrande under the TA Top-10 intervention. ΔAcc. is the change in accuracy relative to the unedited merged model. ΔGold and ΔInc. are average changes in the gold-candidate and strongest incorrect-candidate log-likelihoods, and ΔMargin is the average change in their difference.

its task-specific reference. Merging lowers average validation accuracy for both model families, with larger drops for T5-large. The size of the drop also depends on the merge method. Among the three methods, TA gives the best average accuracy for both model families, reaching 78.3 for T5-large and 82.6 for Phi-3.5. Therefore, we use TA as the primary setting for the hidden-state analysis because it is the strongest merge method while still showing a clear performance drop.

6.2 Is post-merge distortion concentrated in a few directions?

Having established a performance drop, we next ask whether the hidden-state distortion is spread broadly or concentrated in a small number of directions. To measure this, we examine how much of the distortion variance is captured by the top singular directions. Figure 3 shows that the cumulative explained variance rises quickly for both model families. A small number of directions account for much of the distortion, indicating strong anisotropy.

We then compare the real distortion with two structural controls: matched-norm random and shuffled. The matched-norm control preserves each distortion vector's norm while randomizing its direction, whereas the shuffled control preserves its coordinate values but independently permutes their positions. Table 2 shows that both controls are much more diffuse than the real distortion. At $k = 10$, the top directions capture 74.1% of the variance for T5-large and 72.3% for Phi-3.5, compared with much smaller values for the controls. The participation ratios are also low, at 7.0 and 6.3, indicating that the distortion has a small effective dimensionality. These results suggest that many distortion vectors vary along the same few directions. The same pattern appears across the six tasks and both model families. Appendix C.1 reports the task-level results. Appendix C.2 shows similar concentration for AVG and TIES.

6.3 Do dominant distortion directions affect predictions?

The concentration in a few directions suggests that merging changes representations in a structured way. We test whether this structure matters for prediction by removing the distortion component in the top-10 subspace and comparing the result with a functional control that removes the component in a random 10-dimensional subspace. Table 3 shows a clear difference between the two interventions. Top-10 improves average accuracy by 7.8 points for T5-large and 3.3 points for Phi-3.5, whereas Random-10 changes accuracy by only 1.0 and –0.1 points, respectively. The effect also varies substantially across tasks. PAWS and WSC improve considerably, whereas WinoGrande changes little for either model family. Appendix C.3 reports results for additional values of k. This variation motivates a closer look at why the same intervention can produce very different changes in accuracy.

We focus on PAWS and WinoGrande as contrasting cases from Table 3. For an input x, let y^* denote the gold candidate. We measure how strongly the gold candidate is preferred over the highest-scoring incorrect candidate using the gold-candidate margin:

$$\text{Margin}(x) = \log p(y^* \mid x) - \max_{y \neq y^*} \log p(y \mid x). \tag{17}$$

A larger margin means that the gold candidate is scored further above the strongest incorrect candidate. In Table 4, ΔMargin is the average change in the gold-candidate margin under Top-10 relative to the unedited merged model, computed over the validation examples. We also report ΔGold and ΔInc., the corresponding average changes in the gold-candidate and strongest incorrect-candidate log-likelihood scores, to show how each contributes to the margin change.

On PAWS, Top-10 raises the gold-candidate score while lowering the strongest incorrect-candidate score for both model families, producing a large positive ΔMargin. On WinoGrande, the gold and strongest incorrect scores instead increase by nearly the same amount, leaving ΔMargin close to zero. This contrast helps explain why Top-10 changes accuracy substantially on PAWS but barely changes it on WinoGrande. On PAWS, the intervention improves the gold candidate relative to its strongest competitor, whereas on WinoGrande it largely moves the two scores together. Thus, the behavioral effect depends not only on how much the hidden state changes, but on how that change redistributes scores between competing candidates.

Task	Global Top-10 ΔAcc.	Random-10 ΔAcc.
T5-large		
PAWS	+16.4	+2.6
QASC	+0.5	+0.2
QuaRTz	+2.3	+0.3
WikiQA	+5.2	+1.0
WinoGrande	–0.3	+0.1
WSC	+9.6	+1.9
Avg.	**+5.6**	**+1.0**
Phi-3.5		
PAWS	+3.2	+0.4
QASC	–0.1	–0.3
QuaRTz	+0.3	+0.3
WikiQA	+4.6	+0.1
WinoGrande	0.0	+0.1
WSC	+6.7	−1.0
Avg.	**+2.4**	**–0.1**

Table 5: Cross-task hidden-state intervention results for TA on the validation split. Global Top-10 uses the shared SVD basis fit from up to 1,000 training examples per task across all six tasks. Random-10 reuses the same seeded random-subspace control as in Table 3.

6.4 Is some distortion structure shared across tasks?

The task-specific results show that dominant distortion directions can change predictions within individual tasks. We next ask whether one basis learned jointly across tasks can still affect those tasks. For each model family, we pool distortions from up to 1,000 training examples per task across all six tasks and fit a single SVD basis. We then apply this shared basis to the validation examples of every task, removing the distortion component in its top-10 subspace. Table 5 shows average accuracy gains of 5.6 points for T5-large and 2.4 points for Phi-3.5. The gains vary across tasks, with the largest improvements on PAWS and WSC for T5-large, and on PAWS, WikiQA, and WSC for Phi-3.5.

For comparison, we reuse the Random-10 functional control from Table 3. It changes average accuracy only slightly for either model family. The larger effect of Global Top-10 suggests that the pooled distortions contain structure shared across tasks.

λ	Dist. ↓	Conf. ↑	Agree. ↑	Sup. ↑
T5-large				
0.1	0.206	0.8582	0.690	68.8
0.3	0.094	0.8887	0.808	79.4
0.5	**0.076**	0.9433	**0.899**	**88.3**
0.7	0.097	0.9610	0.841	83.0
1.0	0.169	**0.9614**	0.768	76.2
Phi-3.5				
0.1	0.664	**0.9254**	0.757	76.4
0.3	**0.366**	0.8851	**0.855**	**85.4**
0.5	0.493	0.7760	0.743	73.2
0.7	0.722	0.7162	0.495	48.5
1.0	0.880	0.7569	0.512	50.6

Table 6: Merge-scale selection for TA on the training subset. Dist. is the normalized merged-reference distortion score. Conf. is prediction confidence, and Agree. is prediction agreement with the task-specific reference models. Sup. is supervised selection accuracy (%). Arrows indicate the preferred direction for each criterion, and bold marks the scale selected by each criterion.

6.5 Can distortion guide label-free merge selection?

Finally, we examine whether distortion can help choose among candidate merge scales without gold labels. Table 6 compares distortion, prediction confidence, reference agreement, and supervised accuracy using the same candidate scales and the same training examples. Distortion selects $\lambda = 0.5$ for T5-large and $\lambda = 0.3$ for Phi-3.5. These are the same scales selected by supervised accuracy.

The other label-free criteria help clarify where the useful selection signal comes from. Confidence depends only on the merged model's own outputs, whereas reference agreement compares the merged model's prediction with that of the task-specific reference model. Confidence selects different scales for both model families, while agreement selects the same scales as distortion and supervised accuracy. This contrast suggests that the task-specific references provide useful information for choosing the merge scale. Distortion uses that information at the representation level, while agreement uses it at the prediction level.

7 Conclusion

A drop in task performance tells us that merging has damaged a model, but it does not show how that damage appears during computation. By comparing a merged model with the task-specific models from which it was built, we find that the resulting hidden-state change has a clear structure. Much of the distortion lies in a small number of directions, and removing components along those directions can alter the model's predictions. Our results provide a representation-level view of merge degradation that complements parameter-space analyses and task-level evaluation. Tracing these distortion patterns back to the underlying parameter updates is an important direction for future work and may help explain where these representation changes come from and how parameter-level conflict produces them.

Limitations

Our functional control removes the component of the distortion that lies in a random k-dimensional subspace. Projecting out the component along the dominant top-k distortion subspace changes model behavior much more than the random-k control, but it also removes a larger fraction of the observed distortion. The control is therefore dimensionality-matched but not energy-matched. Because subspace identity and the amount of distortion removed vary together in this comparison, we cannot determine whether the larger behavioral effect comes from the specific dominant subspace or from removing more distortion.

Distortion-based merge selection does not use gold task labels, but it requires task-specific reference models during scale selection. Computing the criterion requires forward passes through both the reference models and candidate merged models to extract hidden states. Once these states are available, computing the distortion score is inexpensive.

Our analysis measures final-layer hidden states at candidate-scoring positions. This controlled setting lets us compare merged and reference representations directly, but it leaves open whether the same distortion structure appears at other layers or during free-form generation.

Acknowledgments

This work was supported by JSPS KAKENHI Grant Number JP26K14954. Part of this research was conducted by AIST and its subcontractor, The University of Osaka, under FRONTia, a Japanese national program led by the Ministry of Economy, Trade and Industry (METI) and the New Energy and Industrial Technology Development Organization (NEDO), aimed at developing a domestic multimodal foundation model for AI robots and physical AI.

References

Marah Abdin, Jyoti Aneja, Hany Awadalla, et al. 2024. Phi-3 technical report: A highly capable language model locally on your phone. *Preprint*, arXiv:2404.14219.

Yamini Bansal, Preetum Nakkiran, and Boaz Barak. 2021. Revisiting model stitching to compare neural representations. In *Advances in Neural Information Processing Systems*, volume 34, pages 225–236. Curran Associates, Inc.

Yuan Cao, Dezhi Ran, Yuzhe Guo, Mengzhou Wu, Simin Chen, Linyi Li, Wei Yang, and Tao Xie. 2026. An empirical study and theoretical explanation on task-level model-merging collapse. *Preprint*, arXiv:2603.09463.

Luming Chen, Ziwei Xiang, Kai Lei, and Xu-Yao Zhang. 2025. Multi-task model fusion via adaptive merging. In *ICASSP 2025 - 2025 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1–5.

Runxi Cheng, Feng Xiong, Yongxian Wei, Wanyun Zhu, and Chun Yuan. 2025. 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*, volume 267 of *Proceedings of Machine Learning Research*, pages 10121–10143. PMLR.

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

Nico Daheim, Thomas Möllenhoff, Edoardo Ponti, Iryna Gurevych, and Mohammad Emtiyaz Khan. 2024. Model merging by uncertainty-based gradient matching. In *The Twelfth International Conference on Learning Representations*.

Diego Doimo, Alessandro Pietro Serra, Alessio ansuini, and Alberto Cazzaniga. 2024. The representation landscape of few-shot learning and fine-tuning in large language models. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*.

Kawin Ethayarajh. 2019. How contextual are contextualized word representations? Comparing the geometry of BERT, ELMo, and GPT-2 embeddings. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 55–65, Hong Kong, China. Association for Computational Linguistics.

Zitao Fang, Guodong Du, Shuyang Yu, Yifei Guo, Yiwei Zhang, Yiyao Cao, Jing Li, Ho-Kin Tang, and Sim Kuan Goh. 2025. To see a world in a spark of neuron: Disentangling multi-task interference for training-free model merging. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 15720–15740, Suzhou, China. Association for Computational Linguistics.

Peiran Gao, Eric Trautmann, Byron Yu, Gopal Santhanam, Stephen Ryu, Krishna Shenoy, and Surya Ganguli. 2017. A theory of multineuronal dimensionality, dynamics and measurement. *bioRxiv*.

Antonio Andrea Gargiulo, Donato Crisostomi, Maria Sofia Bucarelli, Simone Scardapane, Fabrizio Silvestri, and Emanuele Rodolà. 2025. Task singular vectors: Reducing task interference in model merging. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 18695–18705.

Gene H. Golub and Charles F. Van Loan. 2013. *Matrix Computations - 4th Edition*. Johns Hopkins University Press, Philadelphia, PA.

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

Minyoung Huh, Brian Cheung, Tongzhou Wang, and Phillip Isola. 2024. Position: The platonic representation hypothesis. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pages 20617–20642. PMLR.

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

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

Ian T. Jolliffe and Jorge Cadima. 2016. Principal component analysis: a review and recent developments. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 374(2065):20150202.

Tushar Khot, Peter Clark, Michal Guerquin, Peter Jansen, and Ashish Sabharwal. 2020. QASC: A dataset for question answering via sentence composition. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(05):8082–8090.

Yeoreum Lee, Jinwook Jung, and Sungyong Baik. 2025. Mitigating parameter interference in model merging via sharpness-aware fine-tuning. In *The Thirteenth International Conference on Learning Representations*.

Hector J. Levesque, Ernest Davis, and Leora Morgenstern. 2012. The winograd schema challenge. In

Proceedings of the Thirteenth International Conference on Principles of Knowledge Representation and Reasoning, pages 552–561.

Kenneth Li, Oam Patel, Fernanda Viégas, Hanspeter Pfister, and Martin Wattenberg. 2023. Inference-time intervention: Eliciting truthful answers from a language model. In *Thirty-seventh Conference on Neural Information Processing Systems*.

Longhua Li, Lei Qi, Qi Tian, and Xin Geng. 2026. Model merging in the essential subspace. In *Conference on Computer Vision and Pattern Recognition 2026*.

Sourab Mangrulkar, Sylvain Gugger, Lysandre Debut, Younes Belkada, Sayak Paul, and Benjamin Bossan. 2022. PEFT: State-of-the-art parameter-efficient fine-tuning methods. `https://github.com/huggingface/peft`.

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

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

Guillermo Ortiz-Jimenez, Alessandro Favero, and Pascal Frossard. 2023. Task arithmetic in the tangent space: Improved editing of pre-trained models. In *Thirty-seventh Conference on Neural Information Processing Systems*.

Kiho Park, Yo Joong Choe, and Victor Veitch. 2024. The linear representation hypothesis and the geometry of large language models. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pages 39643–39666. PMLR.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67.

Adir Rahamim, Asaf Yehudai, Boaz Carmeli, Leshem Choshen, Yosi Mass, and Yonatan Belinkov. 2026. Will it merge? on the causes of model mergeability. In *Findings of the Association for Computational Linguistics: ACL 2026*, pages 26551–26570, San Diego, California, United States. Association for Computational Linguistics.

Pratik Ramesh, George Stoica, Arun Iyer, Leshem Choshen, and Judy Hoffman. 2026. Resolving interference (ri): Disentangling models for improved model merging. *Preprint*, arXiv:2603.13467.

Nina Rimsky, Nick Gabrieli, Julian Schulz, Meg Tong, Evan Hubinger, and Alexander Turner. 2024. Steering llama 2 via contrastive activation addition. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 15504–15522, Bangkok, Thailand. Association for Computational Linguistics.

Keisuke Sakaguchi, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. 2021. WinoGrande: an adversarial winograd schema challenge at scale. *Commun. ACM*, 64(9):99–106.

Rishi Sharma, James Allen, Omid Bakhshandeh, and Nasrin Mostafazadeh. 2018. Tackling the story ending biases in the story cloze test. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 752–757, Melbourne, Australia. Association for Computational Linguistics.

Oyvind Tafjord, Matt Gardner, Kevin Lin, and Peter Clark. 2019. QuaRTz: An open-domain dataset of qualitative relationship questions. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 5941–5946, Hong Kong, China. Association for Computational Linguistics.

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

Lucrezia Valeriani, Diego Doimo, Francesca Cuturello, Alessandro Laio, Alessio ansuini, and Alberto Cazzaniga. 2023. The geometry of hidden representations of large transformer models. In *Thirty-seventh Conference on Neural Information Processing Systems*.

Qi Wei, Shuo He, Enneng Yang, Tingcong Liu, Haobo Wang, Lei Feng, and Bo An. 2025. Representation surgery in model merging with probabilistic modeling. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pages 66015–66032. PMLR.

Mitchell Wortsman, Gabriel Ilharco, Samir Ya Gadre, Rebecca Roelofs, Raphael Gontijo-Lopes, Ari S Morcos, Hongseok Namkoong, Ali Farhadi, Yair Carmon, Simon Kornblith, and Ludwig Schmidt. 2022. Model soups: averaging weights of multiple fine-tuned models improves accuracy without increasing inference time. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pages 23965–23998. PMLR.

Zhengxuan Wu, Aryaman Arora, Zheng Wang, Atticus Geiger, Dan Jurafsky, Christopher D Manning, and Christopher Potts. 2024. ReFT: Representation finetuning for language models. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*.

Prateek Yadav, Derek Tam, Leshem Choshen, Colin Raffel, and Mohit Bansal. 2023. TIES-merging: Resolving interference when merging models. In *Thirty-seventh Conference on Neural Information Processing Systems*.

Enneng Yang, Li Shen, Guibing Guo, Xingwei Wang, Xiaochun Cao, Jie Zhang, and Dacheng Tao. 2026. Model merging in llms, mllms, and beyond: Methods, theories, applications, and opportunities. *ACM Computing Surveys*, 58(8):1–41.

Enneng Yang, Li Shen, Zhenyi Wang, Guibing Guo, Xiaojun Chen, Xingwei Wang, and Dacheng Tao. 2024. Representation surgery for multi-task model merging. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pages 56332–56356. PMLR.

Yi Yang, Wen-tau Yih, and Christopher Meek. 2015. WikiQA: A challenge dataset for open-domain question answering. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 2013–2018, Lisbon, Portugal. Association for Computational Linguistics.

Dengming Zhang, Xiaowen Ma, Zhen-Liang Ni, Zhenkai Wu, Han Shu, Xin Jiang, and Xinghao Chen. 2026. Expert merging: Model merging with unsupervised expert alignment and importance-guided layer chunking. In *The Fourteenth International Conference on Learning Representations*.

Haobo Zhang and Jiayu Zhou. 2025. Unraveling LoRA interference: Orthogonal subspaces for robust model merging. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 26459–26472, Vienna, Austria. Association for Computational Linguistics.

Yuan Zhang, Jason Baldridge, and Luheng He. 2019. PAWS: Paraphrase adversaries from word scrambling. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 1298–1308, Minneapolis, Minnesota. Association for Computational Linguistics.

Yuyan Zhou, Liang Song, Bingning Wang, and Weipeng Chen. 2024. MetaGPT: Merging large language models using model exclusive task arithmetic. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 1711–1724, Miami, Florida, USA. Association for Computational Linguistics.

A Merge methods

For T5-large, the merge methods operate on full-model parameters. Let $\boldsymbol{\theta}_0$ denote the pretrained parameters and $\boldsymbol{\theta}_t$ the task-specific parameters for task t. For methods that operate on parameter updates, we define the task vector as:

$$\boldsymbol{\tau}_t = \boldsymbol{\theta}_t - \boldsymbol{\theta}_0.$$

For Phi-3.5, the merge methods instead operate directly on the LoRA adapter parameters. Let $\boldsymbol{\phi}_t$ denote the adapter parameters for task t.

AVG (Choshen et al., 2022; Wortsman et al., 2022) averages the task-specific parameters. For T5-large, the merged parameters are:

$$\boldsymbol{\theta}_{\mathrm{AVG}} = \frac{1}{|\mathcal{T}|} \sum_{t \in \mathcal{T}} \boldsymbol{\theta}_t.$$

For Phi-3.5, the same averaging is applied to $\boldsymbol{\phi}_t$.

TA (Ilharco et al., 2023) sums the task vectors for T5-large:

$$\boldsymbol{\theta}_{\mathrm{TA}} = \boldsymbol{\theta}_0 + \lambda \sum_{t \in \mathcal{T}} \boldsymbol{\tau}_t,$$

where λ controls the scale of the combined update. For Phi-3.5, TA instead sums and scales the LoRA adapter parameters.

TIES (Yadav et al., 2023) reduces conflicts before aggregation. It removes small-magnitude entries, determines an aggregate sign for each parameter coordinate, and retains entries whose signs agree with that aggregate sign. For T5-large, this procedure is applied to the task vectors. For Phi-3.5, it is applied to the LoRA adapter parameters.

B Additional details

Tasks. The six tasks cover question answering, paraphrase identification, and coreference resolution. We exclude Story Cloze (Sharma et al., 2018) because it requires a separate access request rather than being directly available through the same dataset pipeline. For WSC, we use the SuperGLUE `wsc.fixed` configuration.

Fine-tuning. We fine-tune one task-specific model for each task. For T5-large, we use full-model fine-tuning for 5,000 batches with batch size 32, maximum sequence length 128, learning rate 1×10^{-5}, and weight decay 0.0. For Phi-3.5, we train LoRA adapters using the PEFT library (Mangrulkar et al., 2022), targeting the fused attention

input projection `qkv_proj` and the attention output projection `o_proj`, with rank 16, LoRA alpha 32, and dropout 0.05. We train each adapter for 3,000 batches with batch size 4, gradient accumulation 4, maximum sequence length 512, learning rate 1×10^{-4}, and bfloat16 precision.

Merge hyperparameters. AVG has no tuned parameter. For TA, we evaluate $\lambda \in \{0.1, 0.3, 0.5, 0.7, 1.0\}$. For TIES, we retain the top 10%, 20%, and 30% of parameter entries by magnitude, use mass-based sign resolution and disjoint-sum merging, and evaluate the same λ grid as TA. We use the same candidate settings for T5-large and Phi-3.5.

Control construction. For the random-k intervention, we draw a Gaussian random matrix in $\mathbb{R}^{d \times k}$ and orthonormalize its columns with QR decomposition to obtain the random basis. We then use the same projection-removal edit as for the top-k distortion basis. For the matched-norm structural control, we independently draw a Gaussian vector for each distortion vector, normalize it, and rescale it to match the original distortion norm. For the shuffled control, we independently permute the coordinates within each distortion vector. The matched-norm and shuffled matrices are then centered and analyzed with the same SVD procedure as the original distortion matrix.

Merge selection. For distortion-based selection, we evaluate the same TA scale grid above. For each candidate scale, we construct the corresponding merged model and compute the normalized distortion score on a fixed 200-example training subset for each task. The confidence, reference-agreement, and supervised criteria use the same candidate scales and exactly the same sampled training examples. Thus, the comparison changes the selection criterion while holding the candidate models and selection examples fixed.

Compute. We run the experiments on NVIDIA GeForce RTX 3090 GPUs. Fine-tuning one task-specific model takes about 28–34 minutes per task for T5-large and 28–47 minutes per task for Phi-3.5 LoRA. Standard evaluation takes about 7–199 seconds per task for T5-large and 10–411 seconds per task for Phi-3.5, depending on dataset size and candidate length.

Hidden-state extraction requires forward passes through both the task-specific reference model and the merged model. Across merge methods, extracting both sets of representations takes about 14–198 seconds per task for T5-large and 21–656 seconds per task for Phi-3.5. Once the hidden states are saved, the anisotropy analysis is CPU-only. Computing the SVD and summary statistics takes about 0.4–31 seconds per task for T5-large and 1.6–312 seconds per task for Phi-3.5.

C Supplementary results

C.1 Task-level anisotropy

Table 7 breaks the anisotropy results down by task. Across all six tasks and both model families, the top distortion directions capture much more variance than the matched-norm random and shuffled controls. The degree of concentration still varies across tasks.

C.2 Anisotropy for additional merge methods

Tables 8 and 9 use the same metrics as Table 2 for AVG and TIES. Both methods show the same qualitative behavior as TA: distortion is concentrated in the top directions, while the matched-norm and shuffled controls remain much more diffuse. This shows that the anisotropy is not specific to TA.

C.3 Sensitivity to the number of intervention directions

Figure 4 shows that the intervention effect is not tied to $k = 10$. Several tasks improve over a range of k. WinoGrande remains close to the merged baseline across the tested values of k.

Task	Top-1	Top-5	Top-10	Matched-norm-10	Shuffled-10	PR
T5-large						
PAWS	36.8	84.7	90.1	1.4	1.4	4.2
QASC	26.2	54.7	62.5	1.4	1.4	10.4
QuaRTz	22.7	52.5	62.0	3.1	3.1	11.3
WikiQA	55.7	75.1	80.3	1.8	1.8	3.1
WinoGrande	29.4	61.0	69.1	2.0	2.0	8.3
WSC	39.9	74.0	80.7	9.0	8.9	4.7
Phi-3.5						
PAWS	41.8	80.9	87.4	0.5	0.5	4.3
QASC	32.0	56.9	62.0	0.7	0.7	7.6
QuaRTz	27.1	55.9	61.7	1.7	1.7	9.4
WikiQA	33.2	70.5	75.7	0.8	0.8	5.9
WinoGrande	49.1	67.7	71.4	1.1	1.1	3.9
WSC	29.0	69.7	75.8	4.2	4.1	6.5

Table 7: Task-level anisotropy of the merged-reference distortion under TA on the validation split. Top-1, Top-5, and Top-10 report cumulative explained variance for each task. Matched-norm-10 and Shuffled-10 report the corresponding Top-10 values for the two structural controls. PR denotes participation ratio.

Model	Top-1	Top-5	Top-10	Matched-norm-10	Shuffled-10	PR
T5-large	53.9 (34.5–68.2)	78.1 (59.8–92.2)	83.1 (67.1–95.1)	3.0 (1.5–7.8)	3.0 (1.5–7.7)	3.8 (2.1–7.0)
Phi-3.5	38.9 (30.2–50.7)	71.2 (62.8–84.3)	75.9 (66.3–90.1)	1.4 (0.5–4.0)	1.4 (0.6–4.0)	5.1 (3.2–6.3)

Table 8: Anisotropy of the merged-reference distortion under AVG on the validation split, averaged over six tasks.

Model	Top-1	Top-5	Top-10	Matched-norm-10	Shuffled-10	PR
T5-large	36.3 (18.3–51.3)	65.4 (43.6–84.2)	73.0 (56.0–89.9)	2.9 (1.4–7.8)	2.9 (1.4–7.8)	8.2 (3.5–17.8)
Phi-3.5	40.8 (28.2–53.7)	69.6 (55.7–84.8)	74.1 (59.5–89.8)	1.4 (0.6–3.9)	1.4 (0.6–3.9)	5.2 (3.1–8.3)

Table 9: Anisotropy of the merged-reference distortion under TIES on the validation split, averaged over six tasks.

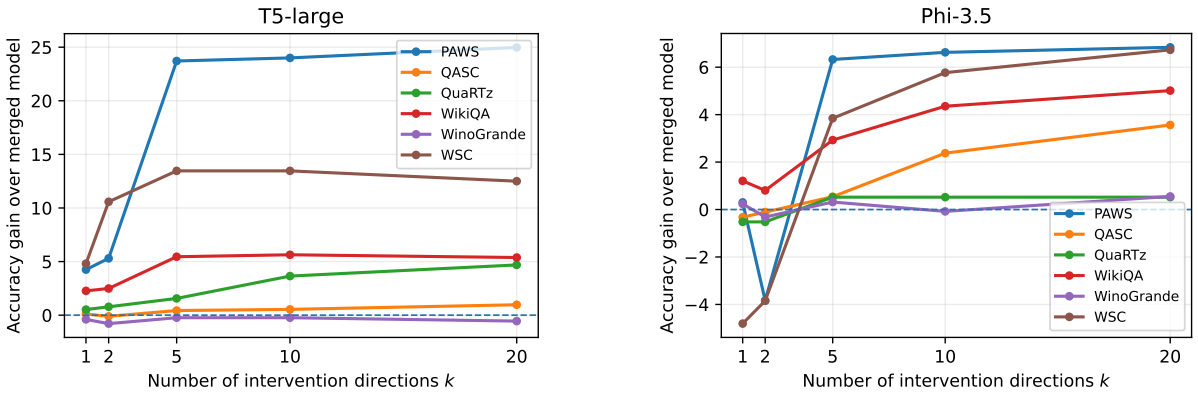


Figure 4: Sensitivity of the top-k hidden-state intervention under TA. The x-axis is the number of intervention directions k. The y-axis is the accuracy gain over the unedited merged model on the validation split. Each curve corresponds to one task.