arXiv:2508.21421v3 [cs.LG] 25 Feb 2026

Rethinking Layer-wise Model Merging through Chain of Merges

Pietro Buzzega [1] **Riccardo Salami** [1] **Angelo Porrello** [1] **Simone Calderara** [1]

Abstract

Fine-tuning pretrained models has become a standard pathway to achieve state-of-the-art performance across a wide range of domains, leading to a proliferation of task-specific model variants. As the number of such specialized models increases, merging them into a unified model without retraining has become a critical challenge. Existing merging techniques operate at the level of individual layers, thereby overlooking the inter-layer dependencies inherent in deep networks. We show that this simplification induces distributional mismatches in intermediate activations during merging, as changes applied to early layers fail to propagate to downstream ones. We identify these mismatches as a form of *internal covariate shift*, comparable to the phenomenon encountered in the initial phases of neural networks training. To address this, we propose *Chain of Merges* (CoM), a layer-wise merging procedure that sequentially merges weights across layers while sequentially updating activation statistics. By explicitly accounting for inter-layer interactions, CoM mitigates covariate shift and produces a coherent merged model through a series of conditionally optimal updates. Experiments on standard benchmarks demonstrate that CoM achieves state-of-the-art performance across both vision and language tasks. Codebase is available in the supplementary material.

1. Introduction

The availability of large-scale pretrained models has reshaped machine learning (Radford et al., 2021; Touvron et al., 2023), with fine-tuning emerging as the most accessible path to obtaining state-of-the-art performance across diverse domains (Raffel et al., 2020; Wang et al., 2019). As these foundation models are increasingly adapted to specialized tasks and datasets, a natural question arises: *how can we combine task-specific checkpoints without retraining?* This challenge, broadly referred to as model merging (Ilharco et al., 2023; Matena & Raffel, 2022; Wortsman et al., 2022a), has recently proven effective for achieving modularity, knowledge reuse, and efficient deployment.

Since specialized modules are typically trained independently, there is no guarantee that their weights can be seamlessly combined (Yadav et al., 2023; Stoica et al., 2023). In practice, naive strategies such as weight averaging (McMahan et al., 2017; Wortsman et al., 2022a) often lead to strong performance degradation when combining heterogeneous models (Tang et al., 2024; Daheim et al., 2024; Tam et al., 2024). To tackle this challenge, the literature has proposed a wide range of heuristics, spanning techniques that mitigate interference (Yadav et al., 2023; Yu et al., 2024), align parameters via permutation-based matching (Ainsworth et al., 2023; Singh & Jaggi, 2020), preserve important weights (Matena & Raffel, 2022; Lee et al., 2025), and perform interpolation within orthonormal or task-adaptive subspaces (Marczak et al., 2025; Gargiulo et al., 2025; Tam et al., 2024). While these approaches achieve decent performance, they typically rely on problem-specific assumptions and extended hyperparameter search, lacking a unifying theoretical foundation.

A different line of work focuses on aligning model activations at the layer level (Stoica et al., 2023; Jin et al., 2023; Tatro et al., 2020; Jordan et al., 2022), typically by permuting or modifying parameters to facilitate compatibility. While these approaches lay a foundation for more principled model composition, they overlook a key challenge: layers in deep networks are not independent, but conditioned on the outputs of preceding computation. Merging them independently can introduce inconsistencies across the network. In fact, modifying early-layer parameters through merging can shift the distribution of their output activations, resulting in unexpected inputs for downstream layers. This triggers a butterfly effect, where even small discrepancies accumulate as they propagate through the network, leading to escalating mismatches and consequent performance degradation.

We identify this issue as a form of *internal covariate shift* (ICS) (Ioffe & Szegedy, 2015), a well-known problem in training dynamics where rapidly shifting early-layer activations produce unstable output distributions that hinder

[1]AImageLab, Department of Engineering, University of Modena and Reggio Emilia, Italy.

downstream learning (Arpit et al., 2016). Analogously, we refer to its manifestation in model merging as *merging covariate shift* (MCS), which occurs when an early layer is altered through merging, causing abrupt shifts in the inputs to subsequent layers. While merging covariate shift may be tolerable for methods operating entirely in parameter space – since they adjust weights directly without depending on intermediate activations – it becomes a critical issue for methods that rely on activation statistics. These approaches rely on layer inputs, which inevitably change when preceding layers are merged. Yet, they merge all layers simultaneously using activations computed before merging, failing to account for the resulting distributional shifts and consequently undermining performance.

To address this challenge, we propose **Chain of Merges (CoM)**: a recursive merging approach that begins at the input layer and iteratively updates parameters until reaching the last one. Specifically, we propose updating activation statistics after each merging step, replacing the original task-specific activations with those produced by the partially merged model. This process explicitly captures inter-layer dependencies and ensures global consistency, providing a framework applicable to any activation-based merging methodology. Building on this, we follow (Jin et al., 2023) and cast parameter merging as a layer-wise distillation problem, where the merged weights are optimized to replicate the activation distributions of the original task-specific modules. This problem admits a closed-form solution for linear layers, which constitute a substantial portion of transformer-based architectures and are typically the only layers optimized during fine-tuning (e.g., LoRA-style adaptation keeps all other weights fixed (Hu et al., 2022)).

Our main contributions can be summarized as follows:

- We identify and analyze the presence of internal covariate shift in model merging, empirically showing that activation mismatches accumulate across layers.
- We introduce Chain of Merges (CoM), which progressively distills parameters by updating activation statistics, ensuring consistency across the network.
- We evaluate CoM on standard model merging benchmarks across language and vision settings, showing it outperforms existing methods by a large margin on LoRA fine-tuning, while matching state of the art on traditional full-rank checkpoints.

2. Background

Model merging aims to combine a collection of N models, all sharing an identical architecture, independently trained on distinct input datasets. Each model comprises L linear layers, which are the target of the merging procedure. For a given layer $l \in \{1, 2, \ldots, L\}$, the set of corresponding weight matrices is denoted as $\{\boldsymbol{W}_i^l\}_{i=1}^N$, all having the same dimensions. Notably, our study focuses on Transformer-based architectures, where linear projections constitute the vast majority of the parameter count (approximately 95%).

RegMean (Jin et al., 2023) proposes to find a single linear transformation, $\boldsymbol{W}_M^l$, that best approximates the behavior of the original layers when applied to their respective inputs $\{\boldsymbol{X}_i^l\}_{i=1}^N$. This is accomplished by minimizing the following objective function:

$$\text{minimize } \Omega^l = \sum_{i=1}^{N} \left\| \boldsymbol{W}_M^l \boldsymbol{X}_i^l - \boldsymbol{W}_i^l \boldsymbol{X}_i^l \right\|_2^2. \quad (1)$$

By differentiating Ω^l with respect to $\boldsymbol{W}_M^l$ and setting the result to zero, we can obtain a closed-form solution for the optimal merged layer:

$$\boldsymbol{W}_M^l = \left(\sum_{i=1}^{N} \boldsymbol{W}_i^l \boldsymbol{X}_i^l \boldsymbol{X}_i^{l\top} \right) \left(\sum_{i=1}^{N} \boldsymbol{X}_i^l \boldsymbol{X}_i^{l\top} \right)^{-1}. \quad (2)$$

Here, each $\mathbf{X}_i^l \in \mathbb{R}^{d \times \text{samples}}$ is the input data to the l^{th} layer of the i^{th} model, and the corresponding Gram matrix $\boldsymbol{X}_i^l \boldsymbol{X}_i^{l\top} \in \mathbb{R}^{d \times d}$ captures the pairwise correlations between individual examples.

3. Methodology

Merging Covariate Shift. When using Eq. 2, the resulting $\boldsymbol{W}_M^l$ closely matches the outputs of the original task-specific layers in isolation. However, merging all layers simultaneously based on the initial inputs overlooks the dependencies between successive layers. Specifically, the inputs $\boldsymbol{X}_i^l$ to the l^{th} layer in Eq. 2 correspond to the activations of the $(l-1)^{\text{th}}$ layer. Once the original parameters $\boldsymbol{W}_i^{l-1}$ are replaced with their merged counterparts $\boldsymbol{W}_M^{l-1}$, these activations shift accordingly. As a result, the statistics used to merge layer l no longer align with the distribution actually produced after layer $(l-1)$ has been merged. This mismatch induces a shift in activation statistics, analogous to internal covariate shift, which we refer to as merging covariate shift (MCS).

3.1. Chain of Merges

Recursive Dependence. To address MCS, we revise the closed-form solution presented in Eq. 2. Instead of relying on the inputs $\boldsymbol{X}_i^l$ – that is, the activations produced by the preceding **unmerged** layer – we employ the activations produced by the preceding **merged** layer, $\hat{\boldsymbol{X}}_i^l$. These represent the actual inputs received by layer l during inference, once all preceding layers are merged. Formally, we define the

pre- and post-merging inputs to layer l of model i as:

$$\begin{aligned} \boldsymbol{X}_i^l &= \boldsymbol{\sigma}^{l-1}\left(\boldsymbol{W}_i^{l-1}\boldsymbol{X}_i^{l-1}\right), \\ \hat{\boldsymbol{X}}_i^l &= \boldsymbol{\sigma}^{l-1}\left(\boldsymbol{W}_M^{l-1}\hat{\boldsymbol{X}}_i^{l-1}\right), \end{aligned} \tag{3}$$

where $\boldsymbol{\sigma}^l$ denotes the (possibly composite) activation function connecting layers l and $(l+1)$. Substituting $\boldsymbol{X}_i^l$ for $\hat{\boldsymbol{X}}_i^l$ yields the revised expression for the merged weights:

$$\boldsymbol{W}_M^l = \left(\sum_{i=1}^N \boldsymbol{W}_i^l \hat{\boldsymbol{X}}_i^l \hat{\boldsymbol{X}}_i^{l\top}\right)\left(\sum_{i=1}^N \hat{\boldsymbol{X}}_i^l \hat{\boldsymbol{X}}_i^{l\top}\right)^{-1}. \tag{4}$$

This substitution induces a *recursive* dependence: the inputs to layer l now depend on the outputs of layer $(l-1)$, which are themselves computed using the merged weights $\boldsymbol{W}_M^{l-1}$ and the inputs to layer $(l-1)$. In turn, these depend on the merged outputs of layer $(l-2)$, and so forth. Hence, at every preceding layer all unmerged activations and parameters must be replaced by their merged counterparts, propagating the correction backward through the entire network.

Initial step. The recursive chain starts at the point where either the weights or the inputs are fixed. Such a base case occurs at the first layer, where the inputs correspond to raw data and are unaffected by prior merging. Thus, the merged weights can be directly computed using Eq. 2 as:

$$\boldsymbol{W}_M^1 = \Big(\sum_{i=1}^N \boldsymbol{W}_i^1 \boldsymbol{X}_i^1 \boldsymbol{X}_i^{1\top}\Big)\Big(\sum_{i=1}^N \boldsymbol{X}_i^1 \boldsymbol{X}_i^{1\top}\Big)^{-1}. \tag{5}$$

Recursive step. The merged weights from the initial step are used to propagate consistent activations forward through the network. For subsequent layers $l = 2, \ldots, L$, the algorithm proceeds recursively, alternating between computing the activations $\hat{\boldsymbol{X}}_i^l$ and updating the merged weights $\boldsymbol{W}_M^l$ according to Eq. 4. By ensuring that the computed statistics at each stage reflect the accumulated effect of all preceding merges, this auto-regressive scheme *fully mitigates* merging covariate shift throughout the network. Importantly, this recursive formulation incurs no extra cost beyond the computation required for Eq. 4, as discussed in Sec. C.

3.2. Correlation-based importance

Although the merged weight matrix produced by our strategy is *optimal* with respect to the regression objective, it must be regarded as an approximation rather than an exact reconstruction of the original task-specific weights. This limitation comes from the structural compression of multiple linear transformations into a single one of fixed dimensionality, which necessarily discards some representational capacity. As a result, a residual regression error is *inevitable*, introducing perturbations into the activations of the merged model across all tasks.

Task Importance. The objective of model merging is to retain the performance across all tasks; however, the relative importance of each task is not uniform. Tasks that are semantically similar to the pretraining naturally benefit from the representations of the base model, as their data distribution aligns with that seen during pretraining. In contrast, those that are semantically distant cannot rely on pretrained features and face a higher risk of performance degradation after merging. Preserving these tasks is more critical, as they stand to lose the most if their task-specific weights are not adequately incorporated into the merged model. To account for this asymmetry, we assign each task–layer pair an importance weight ω_i^l, which should reflect how strongly the merged weights have to be biased toward preserving task i when merging layer l.

Feature correlation as a proxy. To quantify each task's semantic distance from pretraining, we use the overall correlation of each layer's pretrained input features $\boldsymbol{X}_i^l$, which reflects redundancy in task-specific representations. Indeed, highly correlated features align in similar directions, deviating from the pretraining thus indicating greater task importance — as the task is harder to preserve. In contrast, weakly correlated features suggest a distribution closer to pretraining, making the task easier to preserve. This is consistent with prior work (Cogswell et al., 2015; Morcos et al., 2018), demonstrating that decorrelated features enhance generalization, and empirically validated in Sec. 5.

To reduce computation, we leverage the features of the merged model rather than those of the pretrained one, as our merging procedure (Eq. 4) naturally computes the Gram matrix of each layer's inputs. Specifically, we define task importance as the sum of the absolute values of the off-

Algorithm 1 – Chain of Merges

Require: Model weights $\{\boldsymbol{W}_i^l\}_{i=1,2,\ldots,N}^{l=1,2,\ldots,L}$, first layer inputs $\{\boldsymbol{X}_i^1\}_{i=1}^N$, depth L

for $l = 1$ to L **do**
  for $i = 1$ to N **do**
    if $l = 1$ **then**
      $\hat{\boldsymbol{X}}_i^1 \leftarrow \boldsymbol{X}_i^1$
    else
      $\hat{\boldsymbol{X}}_i^l \leftarrow \sigma^{l-1}(\boldsymbol{W}_M^{l-1}\hat{\boldsymbol{X}}_i^{l-1})$
    $\tilde{\boldsymbol{X}}_i^l \leftarrow \hat{\boldsymbol{X}}_i^l / \|\hat{\boldsymbol{X}}_i^l\|_2$
    $\boldsymbol{G}_i^l \leftarrow \tilde{\boldsymbol{X}}_i^l \tilde{\boldsymbol{X}}_i^{l\top}$
    $\omega_i^l \leftarrow \sum_{p<q} |(\boldsymbol{G}_i^l)_{pq}|$
  $\boldsymbol{W}_M^l \leftarrow \Big(\sum_{i=1}^N \omega_i^l \boldsymbol{W}_i^l \boldsymbol{G}_i^l\Big)\Big(\sum_{i=1}^N \omega_i^l \boldsymbol{G}_i^l\Big)^{-1}$
return $\{\boldsymbol{W}_M^l\}_{l=1}^L$

diagonal entries in the Gram matrix $\boldsymbol{G}_i^l$:

$$\omega_i^l = \sum_{p<q} |(\boldsymbol{G}_i^l)_{pq}|, \quad \text{with} \quad \boldsymbol{G}_i^l = \tilde{\boldsymbol{X}}_i^l \tilde{\boldsymbol{X}}_i^{l\top} \tag{6}$$

This metric measures the overall correlation between intermediate activations. Larger ω_i^l indicates stronger correlations, meaning the task has diverged more during fine-tuning and is more critical to retain, while smaller ω_i^l reflects near-orthogonal features, suggesting the task is semantically close to the pretrained model.

Activation Normalization. Our approach requires the inversion of $\sum_{i=1}^N \hat{\boldsymbol{X}}_i^l \hat{\boldsymbol{X}}_i^{l\top}$, whose conditioning critically affects the numerical stability of the solution. In Transformer architectures, activation Gram matrices are frequently ill-conditioned due to the inherent low-dimensionality of the underlying token representations (Barbero et al., 2024; Arefin et al., 2024), especially as fine-tuning typically occurs within a low-dimensional subspace (Aghajanyan et al., 2021; Kumar et al., 2022). Drawing on prior work showing that layer normalization (Ba et al., 2016) mitigates representational collapse (Wu et al., 2024), we replace the gram matrix $\hat{\boldsymbol{X}}_i^l \hat{\boldsymbol{X}}_i^{l\top}$ with the correlation matrix $\boldsymbol{G}_i^l$, which is computed from normalized features and therefore better conditioned. The complete procedure of the proposed methodology is summarized in Alg. 1.

Importance-weighted merging. To incorporate task- and layer-specific importance, we extend our objective Ω with the weighting factor ω_i^l, biasing the model toward more sensitive tasks. The resulting merging rule for $\boldsymbol{W}_M^l$ becomes:

$$\begin{aligned} \arg\min_{\boldsymbol{W}_M^l} \sum_{i=1}^N \omega_i^l \left\| \boldsymbol{W}_M^l \tilde{\boldsymbol{X}}_i^l - \boldsymbol{W}_i^l \tilde{\boldsymbol{X}}_i^l \right\|_2^2 = \\ = \left(\sum_{i=1}^N \omega_i^l \, \boldsymbol{W}_i^l \, \boldsymbol{G}_i^l \right) \left(\sum_{i=1}^N \omega_i^l \, \boldsymbol{G}_i^l \right)^{-1}. \end{aligned} \tag{7}$$

The complete procedure of the proposed method is summarized in Alg. 1, with the full derivation of Eq. 7 provided in Sec. A.

4. Experimental Study

4.1. Evaluation settings

Vision experiments. We evaluate our approach in the vision domain using the benchmark of (Ilharco et al., 2023), which involves merging checkpoints from eight classification datasets: Stanford Cars (Krause et al., 2013), DTD (Cimpoi et al., 2014), EuroSAT (Helber et al., 2019), GTSRB (Stallkamp et al., 2011), MNIST (LeCun et al., 2002), RESISC45 (Cheng et al., 2017), SUN397 (Xiao et al., 2016), and SVHN (Netzer et al., 2011).

Language experiments. Following (Stoica et al., 2025; Panariello et al., 2025), we assess model generalization to the language domain on six datasets: SNLI (Bowman et al., 2015), MultiNLI (Williams et al., 2017), SICK (Marelli et al., 2014), SciTail (Khot et al., 2018), RTE (Wang et al., 2019), and QNLI (Wang et al., 2019). In SNLI, MultiNLI, and SICK, the task is to classify the relationship between a *premise* and a *hypothesis* as entailment, contradiction, or neutral. SciTail, RTE, and QNLI only involve two labels, so the outputs space is restricted accordingly.

Evaluated approaches. We compare our method with 15 leading techniques in the model-merging domain. *Task-Arithmetic (TA)* (Ilharco et al., 2023), *TIES* (Yadav et al., 2023), and *DARE* (Yu et al., 2024) operate by directly merging task vectors in weight space. Improving on them, *Consensus TA* (Wang et al., 2024) prunes task-specific checkpoints to retain only shared parameters before merging, while *LiNeS* (Wang et al., 2025) scales parameter updates by layer depth to preserve both general features and task-specific representations. Among SVD-based techniques, *Iso-C* (Marczak et al., 2025) perform isotropization, decomposing the weights via SVD and reconstructing them with equal singular values, while *KnOTS* (Stoica et al., 2025) aligns weights to improve merging, enhancing existing methods like TIES and DARE. Improving on a similar framework, Core_{TSV} (Panariello et al., 2025) performs merging within compact weight spaces to reduce computational overhead. Its best performing variant adopts *TSV* (Gargiulo et al., 2025), which further compresses the weights and estimates task interference to guide the merging process. *RegMean* (Jin et al., 2023) aligns model parameters by solving a closed-form regression problem across all linear layers, while *FisherAVG* (Matena & Raffel, 2022) combines models using Fisher Information as importance weights. *MaTS* (Tam et al., 2024) extends these two by leveraging conjugate gradient optimization to align models within their respective task-parameter subspaces. On a different line, *AdaMerging++* (Yang et al., 2024) and *ProDistill* (Xu et al., 2025) learn layer-wise scalar coefficients via gradient descent; the former minimizes the entropy of the final predictions and the latter reduces the ℓ_2 norm between fine-tuned and merged layer activations. Finally, *Localize and Stitch* (He et al., 2024) leverages validation data to find and keep just 1% of the model's parameters, minimizing conflicts during merging. *Zero-shot* denotes CLIP's zero-shot performance, while *Individual FT* denotes the performance of each fine-tuned model when evaluated on its own.

Evaluation Protocol. All our experiments follow a *static merging* protocol as in (Stoica et al., 2025; Ilharco et al., 2022; Gargiulo et al., 2025; Panariello et al., 2025): each methodology outputs a single merged backbone used for all tasks at inference time. No task identifiers, routing mecha-

nisms, or input-dependent adapters are allowed at test time. Performance is evaluated using task-specific heads: CLIP zero-shot heads for vision and fine-tuned heads for language.

Implementation Details. Following (Stoica et al., 2025; Gargiulo et al., 2025; Ilharco et al., 2023), we use ViT-B/32 and ViT-L/14 (Dosovitskiy et al., 2021) CLIP encoders as the vision-task backbones for all examined methods. For natural language tasks, we utilize Llama 3-8B (Grattafiori et al., 2024). Each model is fine-tuned using LoRA (Hu et al., 2022) or traditional fine-tuning. While the former is applied solely to attention modules (*i.e.*, query, key, value, and output projection layers) with rank $r = 16$, the latter modifies all weights; for all non-linear layers, we employ simple averaging. To ensure both reproducibility and fair comparison, we employ the LoRA fine-tuned checkpoints provided by (Stoica et al., 2025), and the full fine-tuning checkpoint from (Ilharco et al., 2023). We the use bfloat16 data type for NLP tasks – as it was shown to generally outperform float16 (Kalamkar et al., 2019) – except during Gram-matrix inversion, where float32 is used to ensure numerical stability. Following the original benchmark, we report the average normalized accuracy of the merged model across all datasets. For constructing the Gram matrices, we draw balanced examples across tasks and classes; if the number of classes exceeds the examples, classes are randomly subsampled. To mitigate conditioning issues (see Sec. 3), we use the Moore–Penrose pseudoinverse together with Tikhonov regularization (Hoerl & Kennard, 1970).

Hyperparameters: 500 and 300 samples for vision and language respectively, and a Tikhonov coefficient of 0.95.

4.2. Results

Vision tasks — LoRA. On the smaller ViT-B/32 model, simple parameter-space methods yield limited performance: TA, TIES, DARE, Consensus TA, and LiNeS reach normalized accuracies in the mid-60s, and absolute ones in the mid 50%. Instead, more advanced baselines show mixed results. While RegMean, TSV, and $\text{KnOTS}_{\text{TIES}}$ yield marginal gains, FisherAVG and MaTS, and Iso-C offer more substantial improvements. Core_{TSV} emerges as the strongest baseline, consistently securing the second-best performance, while CoM significantly outperforms all existing methodologies, surpassing Core_{TSV} by more than 16 percentage points.

Moving to the larger ViT-L/14 backbone lifts the performance of nearly all methods and closes the gap between simple parameter-space baselines and more advanced approaches, placing the median normalized accuracy around 75%. TSV and KnOTS gain a couple of points thanks to their SVD-based merging, while Iso-C and Core_{TSV} secure the third- and second-best results with a solid margin. In contrast, RegMean becomes an outlier, underperforming with respect to all other baselines, suggesting that activation matching is less effective than straightforward averaging on this larger architecture. Even here, CoM delivers state of the art performance with a clear gap, reaching 91.06% versus 86.21% for the second-best method. Notably, the sharp contrast between RegMean's performance and that of CoM suggests that Merging Covariate Shift is more pronounced in the ViT-L model than in its base variant.

Table 1. Normalized and absolute average accuracies (%) of merged models with ViT-B/32, ViT-L/14, and Llama-3 8B, LoRA fine-tuned. Best results in bold, second-best underlined.

	ViT-B/32		ViT-L/14		Llama-3 8B	
Method	**Norm**	**Abs**	**Norm**	**Abs**	**Norm**	**Abs**
Zero-shot	57.49	48.32	70.11	64.69	51.09	47.42
Indiv. FT	100.0	84.05	100.0	92.27	100.0	92.54
TA	63.78	53.61	74.79	69.01	90.38	83.64
TIES	63.70	53.54	75.51	69.67	91.08	84.29
$\text{DARE}_{\text{TIES}}$	63.65	53.50	75.53	69.69	89.44	82.77
Cons. TA	64.72	54.40	76.70	70.77	90.79	84.02
LiNeS	63.63	53.48	74.65	68.88	90.84	84.06
FisherAVG	70.04	54.87	75.32	69.50	—	—
RegMean	66.02	55.49	69.85	64.45	87.58	81.05
MaTS	70.01	58.84	75.97	70.10	—	—
TSV	66.66	56.03	77.99	71.96	92.55	85.65
Iso-C	70.66	59.39	83.70	77.23	57.08	52.82
$\text{KnOTS}_{\text{TIES}}$	67.73	56.93	78.99	72.88	92.53	85.63
CORE_{TSV}	<u>76.43</u>	<u>64.24</u>	<u>86.21</u>	<u>79.55</u>	<u>94.16</u>	<u>87.14</u>
CoM (ours)	**92.40**	**77.85**	**91.06**	**84.02**	**99.50**	**92.07**

Language tasks – LoRA. Results on the six natural language benchmarks with LlaMA3-8B (Tab. 1) show that merging is generally less destructive in this domain, as weight-space baselines such as Task Arithmetic already retain very high normalized accuracy (90.38% on average). More sophisticated parameter-based techniques provide small gains w.r.t. TA, while other approaches yield mixed results: $\text{KnOTS}_{\text{TIES}}$ and TSV perform well, whereas RegMean drops lower than TA, following a similar trend shown with the ViT-L architecture. A clear outlier is Iso-C, which drops to 57.08%, likely because rescaling singular values interacts poorly with language model fine-tuning. Finally, while Core_{TSV} secures the second-highest rank, whereas CoM delivers near-perfect merged performance, surpassing the strongest baseline by more than 5 points. Taken alongside our results in the vision domain, these findings demonstrate that CoM is an exceptionally effective strategy for merging low-rank modules, consistently preserving the performance of specialized models. Notably, we omit FisherAVG and MaTS from this comparison, as computing the Fisher Information Matrix for Llama3-8B is prohibitively expensive.

Table 2. Average normalized accuracy of CoM (LoRA fine-tuning) varying the number of examples used for each task.

# of samples	2	5	10	50	100	200	300	400	500
ViT-B/32	75.35	83.86	87.26	90.61	91.86	92.20	92.21	92.38	92.40
ViT-L/14	81.77	87.21	88.95	90.57	90.99	90.97	90.99	91.00	91.06
Llama3-8B	96.15	97.03	98.50	98.62	98.95	99.04	99.50	99.48	99.33

Table 3. Normalized and absolute average accuracies (%) of merged models with ViT-B/32 and ViT-L/14, full fine-tuning. Best results in bold, second-best underlined.

	ViT-B/32		**ViT-L/14**	
Method	**Norm**	**Abs**	**Norm**	**Abs**
Zero-shot	52.2	48.3	67.6	64.7
Individual FT	100.0	92.5	100.0	95.7
TA	76.5	70.8	88.7	84.9
TIES	81.2	75.1	90.8	86.9
Consensus TA	81.4	75.0	90.2	86.3
LiNeS	80.1	74.1	90.3	86.4
FisherAVG	73.8	68.3	85.9	82.2
RegMean	77.6	71.8	87.5	83.7
Loc-and-Stitch	86.3	79.9	90.4	86.5
AdaMerging++	87.6	81.1	95.1	91.0
ProDistill	92.9	86.0	96.1	91.9
TSV	92.8	85.9	97.2	93.0
ISO-C	<u>93.2</u>	<u>86.3</u>	**98.4**	**94.2**
CoM (ours)	**94.8**	**87.7**	<u>97.8</u>	<u>93.6</u>

Vision tasks – Full fine-tuning. With full fine-tuning, performance improves substantially across all methods compared to LoRA. On ViT-B/32, simple baselines such as TA and TIES already achieve competitive results. However, more advanced methods make a strong difference: Localize-and-Stitch and AdaMerging++ deliver strong results around the mid-to-high 80s, while ProDistill and TSV lead the baselines with normalized accuracies above 92%. CoM further advances the state of the art, achieving $94.8\%/87.7\%$ normalized and absolute accuracy. Scaling up to the ViT-L/14 backbone strengthens the same trend. Most methods see consistent gains, with AdaMerging++, ProDistill, and TSV all surpassing 95% normalized accuracy. CoM once again achieves the top results on par with Iso-C. These results confirm that CoM remains highly effective even under full fine-tuning, producing merged models that mostly preserve the performance of their specialized counterparts. However, the smaller performance gap between CoM and competing methods suggests that closed-form activation matching is less effective when merging full-rank checkpoints. More accuracy results for each dataset are provided in Sec. E.

Table 4. Impact of individual components of CoM. *MCS* denotes solving merging covariate shift, *Norm* refers to activation normalization, and *FC* indicates feature correlation weighting.

Components			Architecture		
MCS	Norm	FC	ViT-B/32	ViT-L/14	Llama3-8B
✗	✗	✗	66.02	69.85	87.58
✓	✗	✗	82.53	83.55	99.35
✓	✓	✗	83.51	86.80	99.48
✓	✓	✓	92.62	91.06	99.50

5. Model Analysis

CoM – Number of examples. Since CoM uses input data to estimate task-specific Gram matrices, we perform an ablation study to determine how many samples are required for reliable performance. Consistent with existing merging methods that use validation data for tuning, we sample from the validation set to estimate the Gram matrices. As can be seen in Tab. 2, a sufficiently large sample size is essential to ensure numerical stability of the matrices, which allows the inversion in Eq. 2 to produce meaningful results. Empirically, we observe that stability can be maintained with as few as 2 samples per task. As shown in Tab. 2, CoM already surpasses the current state of the art with only 5 samples and approaches near-maximal performance with 100 samples, highlighting its data efficiency and robustness.

Impact of Individual Components. While CoM is primarily designed to mitigate merging covariate shift during composition, it also incorporates additional components that contribute to overall performance. We quantify the contribution of each component via an ablation study, presented in Tab. 4, systematically removing them to assess their individual impact on performance (we report the Average column). Our results indicate that addressing merging covariate shift alone is sufficient to achieve state-of-the-art performance. However, other components also play a significant role. In particular, weighting by the off-diagonal norm leads to substantial improvements in vision domains, while proving less critical for language tasks. We attribute this discrepancy to the inherent differences between the two settings: language tasks involve highly similar distributions, sharing the *same label space* across all datasets, whereas vision tasks

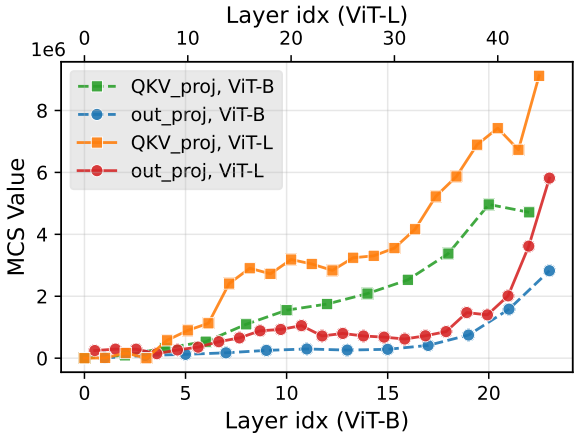


Figure 1. Merging covariate shift across layers.

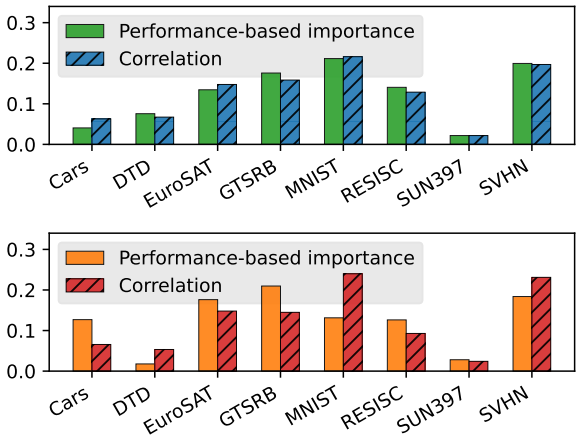


Figure 2. Task importance *vs.* feature correlation.

correspond to classification problems defined on entirely different classes and domains. A more detailed discussion is provided in Sec. B. Finally, although activation normalization has a comparatively smaller impact, it consistently enhances performance across all benchmarks by ensuring numerical stability of the solution.

Measuring MCS. Covariate shift refers to changes in data distributions. Therefore, it is necessary to define a suitable distribution over the network activations in order to quantify this phenomenon. Following Huang & Yu (2020), we model the distribution of activation outputs as a multivariate Gaussian and evaluate MCS using the Earth Mover's (EM) Distance (Villani, 2008), also referred to as the squared 2-Wasserstein distance or the Fréchet distance (Dowson & Landau, 1982). This choice is convenient, as the EM distance admits a closed-form solution for Gaussian distributions. Merging covariate shift for the l^{th} layer of all models can be measured as:

$$\text{MCS}^l = \sum_{i=1}^{N} \|\mu_{i,l} - \hat{\mu}_{i,l}\|_2^2 + \text{Tr}\left[\Sigma_{i,l} + \right. \\ \left. + \hat{\Sigma}_{i,l} - 2\left(\hat{\Sigma}_{i,l}^{1/2}\,\Sigma_{i,l}\,\hat{\Sigma}_{i,l}^{1/2}\right)^{1/2}\right], \tag{8}$$

where $\mu_{i,l}$ and $\hat{\mu}_{i,l}$ denote the empirical means, while $\Sigma_{i,l}$ and $\hat{\Sigma}_{i,l}$ represent the empirical covariance matrices of the inputs $\boldsymbol{X}_i^l$ and $\hat{\boldsymbol{X}}_i^l$ (Eq. 3), respectively. To investigate the presence of MCS during model merging, we measure it using Eq. 8 and report the results in Fig. 1. The results indicate that MCS is present across all layers and tends to increase with depth, as earlier layers influence subsequent ones and the mismatch accumulates. We analyze the two projections before and after attention (the only fine-tuned layers) separately, as they show slightly different behaviors.

Feature correlation. To motivate correlation-based weighting beyond its empirical effectiveness, we investigate whether the semantic distance between specialized tasks and pretraining varies and whether it correlates with our weighting factor ω_i^l. We estimate such semantic distance using the performance gap between each task-specific model (evaluated in isolation) and the zero-shot performance of the base model, which reflects how much fine-tuning can improve the considered task. In Fig. 2, we compare this accuracy gap with our correlation-based weighting factor, averaged across layers to produce a single value per task. The two measures, normalized for visual comparison, exhibit a consistent correlation across datasets and architectures. Results for the language setting are provided in Sec. B.

Complexity. We assess computational efficiency (FLOPs) and memory overhead for ViT-B/32 in Fig. 3, distinguishing between `CoM_fast` (5 samples, SOTA performance) and `CoM_best` (500 samples for vision and 300 for NLP). CoM maintains a resource profile comparable to other SOTA methods while exceeding RegMean in computational efficiency by requiring fewer examples. Indeed, the primary distinction between these two lies in the update rule, which utilizes merged activations rather than task-specific ones. This shift improves accuracy without increasing the number of forward passes (one per task), as activations are cached and the forward pass is effectively paused and resumed at each layer. We refer the reader to Sec. C for results on additional architectures.

6. Comparison with Related Work

Pioneering model merging techniques rely on linear interpolation of model parameters. FedAvg (McMahan et al., 2017) introduces averaging in the context of Federated Learning, assuming a shared initialization across models. Building on

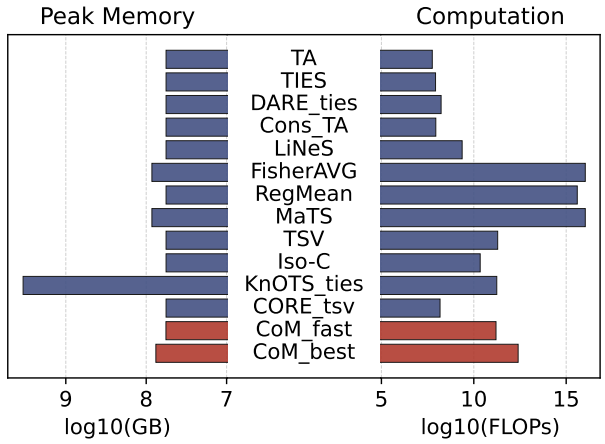


Figure 3. Computational cost and memory overhead (ViT-B/32).

this idea, Model Soups (Wortsman et al., 2022a) propose a greedy strategy that incrementally incorporates models into the mixture only if they improve validation performance; WiSE-FT (Wortsman et al., 2022b) enhances fine-tuning by weighting model updates to boost generalization and robustness, and Task Arithmetic (Ilharco et al., 2023) enables personalized model editing, allowing control over individual contributions. These methods are simple and generally effective, but can suffer performance degradation due to parameters interference (Tang et al., 2024; Tam et al., 2024).

To address this problem, more recent approaches seek to reduce interference by applying heuristics prior to merging: TIES (Yadav et al., 2023) addresses parameter redundancy by pruning and aligning parameter signs; DARE (Yu et al., 2024) applies stochastic dropping and rescaling; Consensus Merging (Wang et al., 2024) learns task-specific pruning masks; LiNeS (Wang et al., 2025) adjusts parameter magnitudes based on their layer depth within the network; and Git Rebasin (Ainsworth et al., 2023) leverages permutations to encourage Linear Mode Connectivity.

In parallel, another line of research argues that model ensembling should be performed within subspaces that maximize alignment between parameter vectors. Within this framework, TSV (Gargiulo et al., 2025) compresses network weights into a low-rank structure and approximates whitening by solving the Procrustes problem. KnOTS (Stoica et al., 2025) performs merging in an aligned parameter space using singular value decomposition. CORE (Panariello et al., 2025) improves KnOTS' efficiency by merging within a lower-rank subspace, while Iso-C (Marczak et al., 2025) enforces layer-wise isotropic matrices by rescaling singular values, producing vectors of equal magnitude. These methodologies rely on heuristics applied directly to parameters, while CoM aligns activations, enabling more precise merging that optimally preserves task-specific features.

A complementary research direction focuses on aligning model features. To do this, Neuron Alignment (Tatro et al., 2020) employs layer-wise regression to optimize bipartite neuron matching, while ZIPIt! (Stoica et al., 2023) introduces permutations based on intermediate features. Alternatively, Optimal Transport Fusion (Singh & Jaggi, 2020) treats alignment as an optimal transport problem, calculating soft matchings between activation distributions. Fisher-weighted averaging (Matena & Raffel, 2022) directly averages parameters weighted by the Fisher Information Matrix (Fisher, 1922), while Regmean (Jin et al., 2023) derives a closed-form solution to match activations layer-wise. MaTS (Tam et al., 2024) unifies the two preceding approaches under a common linear system and solves it via conjugate gradient. Similarly to RegMean, ProDistill (Xu et al., 2025) minimizes the ℓ_2 norm between fine-tuned and merged models activations, but learns layer-wise scalar coefficients via gradient descent. AdaMerging (Yang et al., 2024) also uses gradient descent but minimizes the entropy of the final predictions by modifying all layers jointly. In contrast, our Chain of Merges updates activation statistics sequentially, thereby preserving network consistency.

An orthogonal line of work replaces a single, task-agnostic parameter vector with input- or task-conditioned inference. EMR-Merging leverages a task oracle to merge lightweight task-specific masks and rescalers for each example at test time (Huang et al., 2024). Similarly, Twin-Merging compresses knowledge into exclusive components, dynamically integrating them using a router module (Lu et al., 2024). We-MoE, instead, merges part of the modules statically, leveraging a Mixture-of-Experts on the MLP layers only, with a routing mechanism selecting experts at inference (Tang et al., 2024). These approaches operate under a dynamic merging protocol with task- or input-dependent routing, which differs from the static setting adopted in this work.

7. Conclusions

In this work, we identify and address Merging Covariate Shift (MCS), a form of internal covariate shift that emerges when merging layers independently in methodologies that rely on activation statistics. To mitigate the adversarial effect of this phenomenon, we propose Chain of Merges (CoM), a framework that updates activation statistics autoregressively, capturing inter-layer dependencies and fully eliminating MCS. Empirical results on standard vision and language benchmarks demonstrate that CoM consistently outperforms existing methods across diverse architectures and domains.

Limitations. Our approach *relies on validation data samples* and focuses just on *activation matching*: even though all other methods require data for hyperparameters tuning, we aim to explore generative solutions to remove data dependency, and extend CoM to other merging objectives.

Impact statement

This work studies model merging: combining multiple task-specialized versions of a pretrained model into a single model without retraining. If broadly adopted, effective merging methods like Chain of Merges (CoM) could reduce the need to store, serve, and repeatedly fine-tune separate checkpoints for every downstream task. This can lower computational cost and energy use for deployment and maintenance, enabling more modular reuse of foundation models in resource-constrained settings (e.g., smaller labs, edge/cloud cost-limited applications). It may also improve accessibility by allowing practitioners to integrate new skills into an existing model library more efficiently.

At the same time, merging can propagate or compound undesirable behaviors present in any constituent model. For example, if one task-specific checkpoint encodes harmful biases, privacy leakage, or unsafe generation patterns, merging may transfer these properties into the unified model – potentially making them harder to attribute and mitigate. In addition, our method currently requires representative data samples to estimate activation statistics, which may raise privacy, licensing, or governance concerns in settings where task data cannot be shared or centrally processed. Finally, more capable merged models can lower the barrier to misuse (*e.g.*, scaling the breadth of a model's capabilities for malicious applications).

To mitigate risks, we recommend: *i)* applying dataset and checkpoint governance (licensing, provenance tracking) before merging; *ii)* performing safety and bias evaluations on merged models, not only on individual task models; and *iii)* when data sharing is constrained, exploring privacy-preserving variants (*e.g.*, secure aggregation of statistics, differential privacy, or synthetic/teacher-generated statistics) before real-world deployment.

References

Aghajanyan, A., Gupta, S., and Zettlemoyer, L. Intrinsic dimensionality explains the effectiveness of language model fine-tuning. In Zong, C., Xia, F., Li, W., and Navigli, R. (eds.), *Association for Computational Linguistics*, pp. 7319–7328, Online, August 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.acl-long.568. URL https://aclanthology.org/2021.acl-long.568/.

Ainsworth, S., Hayase, J., and Srinivasa, S. Git re-basin: Merging models modulo permutation symmetries. In *ICLR*, 2023.

Arefin, M. R., Subbaraj, G., Gontier, N., LeCun, Y., Rish, I., Shwartz-Ziv, R., and Pal, C. Seq-vcr: Preventing collapse in intermediate transformer representations for enhanced reasoning. *International Conference on Learning Representations*, 2024.

Arpit, D., Zhou, Y., Kota, B., and Govindaraju, V. Normalization propagation: A parametric technique for removing internal covariate shift in deep networks. In *International Conference on Machine Learning*, pp. 1168–1176. PMLR, 2016.

Ba, J. L., Kiros, J. R., and Hinton, G. E. Layer normalization. *arXiv preprint arXiv:1607.06450*, 2016.

Barbero, F., Banino, A., Kapturowski, S., Kumaran, D., Araújo, J. G., Vitvitskyi, A., Pascanu, R., and Velickovic, P. Transformers need glasses! information over-squashing in language tasks. *URL https://arxiv.org/abs/2406.04267*, 2024.

Bowman, S. R., Angeli, G., Potts, C., and Manning, C. D. The snli corpus. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2015.

Cheng, G., Han, J., and Lu, X. Remote sensing image scene classification: Benchmark and state of the art. *Proceedings of the IEEE*, 105(10):1865–1883, 2017.

Cimpoi, M., Maji, S., Kokkinos, I., Mohamed, S., and Vedaldi, A. Describing textures in the wild. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 3606–3613, 2014.

Cogswell, M., Ahmed, F., Girshick, R., Zitnick, L., and Batra, D. Reducing overfitting in deep networks by decorrelating representations. *arXiv preprint arXiv:1511.06068*, 2015.

Daheim, N., Möllenhoff, T., Ponti, E. M., Gurevych, I., and Khan, M. E. Model merging by uncertainty-based gradient matching. *International Conference on Learning Representations*, 2024.

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., et al. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*, 2021.

Dowson, D. and Landau, B. The fréchet distance between multivariate normal distributions. *Journal of multivariate analysis*, 12(3):450–455, 1982.

Fisher, R. A. On the mathematical foundations of theoretical statistics. *Philosophical transactions of the Royal Society of London. Series A, containing papers of a mathematical or physical character*, 222(594-604):309–368, 1922.

Gargiulo, A. A., Crisostomi, D., Bucarelli, M. S., Scardapane, S., Silvestri, F., and Rodola, E. Task singular vectors: Reducing task interference in model merging. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 18695–18705, 2025.

Grattafiori, A., Dubey, A., Jauhri, A., Pandey, A., Kadian, A., Al-Dahle, A., Letman, A., Mathur, A., Schelten, A., Vaughan, A., et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

He, Y., Hu, Y., Lin, Y., Zhang, T., and Zhao, H. Localize-and-stitch: Efficient model merging via sparse task arithmetic. *Transaction on Machine Learning research*, 2024.

Helber, P., Bischke, B., Dengel, A., and Borth, D. Eurosat: A novel dataset and deep learning benchmark for land use and land cover classification. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 12(7):2217–2226, 2019.

Hoerl, A. E. and Kennard, R. W. Ridge regression: Biased estimation for nonorthogonal problems. *Technometrics*, 12(1):55–67, 1970.

Hu, E. J., yelong shen, Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., Wang, L., and Chen, W. LoRA: Low-rank adaptation of large language models. In *ICLR*, 2022.

Huang, C., Ye, P., Chen, T., He, T., Yue, X., and Ouyang, W. Emr-merging: Tuning-free high-performance model merging. *Advances in Neural Information Processing Systems*, 37:122741–122769, 2024.

Huang, Y. and Yu, Y. An internal covariate shift bounding algorithm for deep neural networks by unitizing layers' outputs. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8465–8473, 2020.

Ilharco, G., Wortsman, M., Gadre, S. Y., Song, S., Hajishirzi, H., Kornblith, S., Farhadi, A., and Schmidt, L. Patching open-vocabulary models by interpolating weights. *NeurIPS*, 2022.

Ilharco, G., Ribeiro, M. T., Wortsman, M., Schmidt, L., Hajishirzi, H., and Farhadi, A. Editing models with task arithmetic. In *ICLR*, 2023.

Ioffe, S. and Szegedy, C. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International conference on machine learning*, pp. 448–456. pmlr, 2015.

Jin, X., Ren, X., Preotiuc-Pietro, D., and Cheng, P. Dataless knowledge fusion by merging weights of language models. In *ICLR*, 2023.

Jordan, K., Sedghi, H., Saukh, O., Entezari, R., and Neyshabur, B. Repair: Renormalizing permuted activations for interpolation repair. *arXiv preprint arXiv:2211.08403*, 2022.

Kalamkar, D., Mudigere, D., Mellempudi, N., Das, D., Banerjee, K., Avancha, S., Vooturi, D. T., Jammalamadaka, N., Huang, J., Yuen, H., et al. A study of bfloat16 for deep learning training. *arXiv preprint arXiv:1905.12322*, 2019.

Khot, T., Sabharwal, A., and Clark, P. Scitail: A textual entailment dataset from science question answering. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32, 2018.

Krause, J., Stark, M., Deng, J., and Fei-Fei, L. 3d object representations for fine-grained categorization. In *Proceedings of the IEEE international conference on computer vision workshops*, pp. 554–561, 2013.

Kumar, A., Raghunathan, A., Jones, R., Ma, T., and Liang, P. Fine-tuning can distort pretrained features and underperform out-of-distribution. *International Conference on Learning Representations*, 2022.

LeCun, Y., Bottou, L., Bengio, Y., and Haffner, P. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 2002.

Lee, S., Liu, J., Wang, Q., Wang, J., Cai, X., and Wu, Y. Dynamic fisher-weighted model merging via Bayesian optimization. In Chiruzzo, L., Ritter, A., and Wang, L. (eds.), *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics*, pp. 4923–4935, Albuquerque, New Mexico, April 2025. Association for Computational Linguistics. ISBN 979-8-89176-189-6. doi: 10.18653/v1/2025.naacl-long.254. URL https://aclanthology.org/2025.naacl-long.254/.

Lu, Z., Fan, C., Wei, W., Qu, X., Chen, D., and Cheng, Y. Twin-merging: Dynamic integration of modular expertise in model merging. *Advances in Neural Information Processing Systems*, 37:78905–78935, 2024.

Marczak, D., Magistri, S., Cygert, S., Twardowski, B., Bagdanov, A. D., and van de Weijer, J. No task left behind: Isotropic model merging with common and task-specific subspaces. *arXiv preprint arXiv:2502.04959*, 2025.

Marelli, M., Bentivogli, L., Baroni, M., Bernardi, R., Menini, S., and Zamparelli, R. Semeval-2014 task 1: Evaluation of compositional distributional semantic models on full sentences through semantic relatedness and textual entailment. In *Proceedings of the 8th international workshop on semantic evaluation (SemEval 2014)*, pp. 1–8, 2014.

Matena, M. S. and Raffel, C. A. Merging models with fisher-weighted averaging. *Advances in Neural Information Processing Systems*, 2022.

McMahan, B., Moore, E., Ramage, D., Hampson, S., and y Arcas, B. A. Communication-efficient learning of deep networks from decentralized data. In *Artificial intelligence and statistics*. PMLR, 2017.

Morcos, A. S., Barrett, D. G., Rabinowitz, N. C., and Botvinick, M. On the importance of single directions for generalization. *arXiv preprint arXiv:1803.06959*, 2018.

Netzer, Y., Wang, T., Coates, A., Bissacco, A., Wu, B., Ng, A. Y., et al. Reading digits in natural images with unsupervised feature learning. In *NIPS workshop on deep learning and unsupervised feature learning*, volume 2011, pp. 7. Granada, 2011.

Panariello, A., Marczak, D., Magistri, S., Porrello, A., Twardowski, B., Bagdanov, A. D., Calderara, S., and van de Weijer, J. Accurate and efficient low-rank model merging in core space. *NeurIPS*, 2025.

Radford, A., Kim, J. W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., et al. Learning transferable visual models from natural language supervision. In *ICML*, 2021.

Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W., and Liu, P. J. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of machine learning research*, 21 (140):1–67, 2020.

Singh, S. P. and Jaggi, M. Model fusion via optimal transport. *Advances in Neural Information Processing Systems*, 33:22045–22055, 2020.

Stallkamp, J., Schlipsing, M., Salmen, J., and Igel, C. The german traffic sign recognition benchmark: a multi-class classification competition. In *The 2011 international joint conference on neural networks*, pp. 1453–1460. IEEE, 2011.

Stoica, G., Bolya, D., Bjorner, J., Ramesh, P., Hearn, T., and Hoffman, J. Zipit! merging models from different tasks without training. *arXiv preprint arXiv:2305.03053*, 2023.

Stoica, G., Ramesh, P., Ecsedi, B., Choshen, L., and Hoffman, J. Model merging with svd to tie the knots. *International Conference on Learning Representations*, 2025.

Tam, D., Bansal, M., and Raffel, C. Merging by matching models in task parameter subspaces. *Transaction on Machine Learning research*, 2024.

Tang, A., Shen, L., Luo, Y., Yin, N., Zhang, L., and Tao, D. Merging multi-task models via weight-ensembling mixture of experts. *International Conference on Machine Learning*, 2024.

Tatro, N., Chen, P.-Y., Das, P., Melnyk, I., Sattigeri, P., and Lai, R. Optimizing mode connectivity via neuron alignment. *Advances in Neural Information Processing Systems*, 33:15300–15311, 2020.

Touvron, H., Lavril, T., Izacard, G., Martinet, X., Lachaux, M.-A., Lacroix, T., Rozière, B., Goyal, N., Hambro, E., Azhar, F., et al. Llama: Open and efficient foundation language models. *arXiv:2302.13971*, 2023.

Villani, C. *Optimal transport: old and new*, volume 338. Springer, 2008.

Wang, A., Singh, A., Michael, J., Hill, F., Levy, O., and Bowman, S. Glue: A multi-task benchmark and analysis platform for natural language understanding. In *ICLR*, 2019.

Wang, K., Dimitriadis, N., Ortiz-Jimenez, G., Fleuret, F., and Frossard, P. Localizing task information for improved model merging and compression. *International Conference on Machine Learning*, 2024.

Wang, K., Dimitriadis, N., Favero, A., Ortiz-Jimenez, G., Fleuret, F., and Frossard, P. Lines: Post-training layer scaling prevents forgetting and enhances model merging. *International Conference on Learning Representations*, 2025.

Williams, A., Nangia, N., and Bowman, S. R. A broad-coverage challenge corpus for sentence understanding through inference. *arXiv preprint arXiv:1704.05426*, 2017.

Wortsman, M., Ilharco, G., Gadre, S. Y., Roelofs, R., Gontijo-Lopes, R., Morcos, A. S., Namkoong, H., Farhadi, A., Carmon, Y., Kornblith, S., et al. Model soups: averaging weights of multiple fine-tuned models improves accuracy without increasing inference time. In *International Conference on Machine Learning*, 2022a.

Wortsman, M., Ilharco, G., Kim, J. W., Li, M., Kornblith, S., Roelofs, R., Lopes, R. G., Hajishirzi, H., Farhadi, A., Namkoong, H., et al. Robust fine-tuning of zero-shot models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 7959–7971, 2022b.

Wu, X., Ajorlou, A., Wang, Y., Jegelka, S., and Jadbabaie, A. On the role of attention masks and layernorm in transformers. *Advances in Neural Information Processing Systems*, 37:14774–14809, 2024.

Xiao, J., Ehinger, K. A., Hays, J., Torralba, A., and Oliva, A. Sun database: Exploring a large collection of scene categories. *International Journal of Computer Vision*, 119(1):3–22, 2016.

Xu, J., Li, J., and Zhang, J. Scalable model merging with progressive layer-wise distillation. *International Conference on Machine Learning*, 2025.

Yadav, P., Tam, D., Choshen, L., Raffel, C. A., and Bansal, M. Ties-merging: Resolving interference when merging models. *Advances in Neural Information Processing Systems*, 36:7093–7115, 2023.

Yang, E., Wang, Z., Shen, L., Liu, S., Guo, G., Wang, X., and Tao, D. Adamerging: Adaptive model merging for multi-task learning. *International Conference on Learning Representations*, 2024.

Yu, L., Yu, B., Yu, H., Huang, F., and Li, Y. Language models are super mario: Absorbing abilities from homologous models as a free lunch. In *Forty-first International Conference on Machine Learning*, 2024.

Appendix

A. Derivation of the correlation-weighted merging solution

We want to minimize the following objective with respect to the matrix $\boldsymbol{W}_M^l$:

$$\tilde{\Omega}_l = \sum_{i=1}^{N} \omega_i^l \, \|\boldsymbol{W}_M^l \tilde{\boldsymbol{X}}_i^l - \boldsymbol{W}_i^l \tilde{\boldsymbol{X}}_i^l\|_2^2.$$

To simplify the notation, define $\boldsymbol{G}_i^l = \tilde{\boldsymbol{X}}_i^l (\tilde{\boldsymbol{X}}_i^l)^\top$. Expanding the norm and applying standard rules of matrix differentiation, the gradient of the objective with respect to $\boldsymbol{W}_M^l$ is:

$$\nabla_{\boldsymbol{W}_M^l} \tilde{\Omega}_l = 2\left[\boldsymbol{W}_M^l \Big(\sum_{i=1}^{N} \omega_i^l \boldsymbol{G}_i^l\Big) - \Big(\sum_{i=1}^{N} \omega_i^l \boldsymbol{W}_i^l \boldsymbol{G}_i^l\Big)\right].$$

The minimizer is obtained by setting the derivative equal to zero as:

$$\boldsymbol{W}_M^l \Big(\sum_{i=1}^{N} \omega_i^l \boldsymbol{G}_i^l\Big) = \sum_{i=1}^{N} \omega_i^l \boldsymbol{W}_i^l \boldsymbol{G}_i^l.$$

Finally, we can solve for $\boldsymbol{W}_M^l$ by multiplying on the right hand side with the inverse of $\sum_{i=1}^{N} \omega_i^l \boldsymbol{G}_i^l$:

$$\boldsymbol{W}_M^l = \left(\sum_{i=1}^{N} \omega_i^l \boldsymbol{W}_i^l \boldsymbol{G}_i^l\right)\left(\sum_{i=1}^{N} \omega_i^l \boldsymbol{G}_i^l\right)^{-1}.$$

If the latter matrix is singular, the Moore–Penrose pseudoinverse should be used instead.

B. Supplementary Ablations for Textual Datasets

Correlation-based importance. In Fig. 4, we report the correlation-based importance analysis for textual tasks (SNLI, MNLI, SICK, QNLI, RTE, and SciTail). Following the approach used for vision datasets, we compute the performance gap between task-specific fine-tuned models and the zero-shot performance of the base Llama3-8B model, and compare it to the correlation-based weighting factor ω_i^l averaged across layers.

In contrast to vision tasks, textual datasets exhibit a relatively uniform importance distribution: both performance gaps and correlation weights vary little across tasks. As a result, the correlation between the two measures is weaker, and correlation-weighted merging provides limited benefit. These findings highlight that the effectiveness of correlation-based weighting depends on task heterogeneity, as greater diversity amplifies its impact on the merged model.

Measuring MCS. In Fig. 5, we extend the analysis from Sec. 5 to the Llama3-8B architecture, computing the MCS for each layer using the same methodology. It can be seen that the amount of merging covariate shift observed in textual datasets is broadly comparable to that identified in vision models. This consistency suggests that the underlying dynamics of parameter interference and shift remain stable across different model scales and modalities.

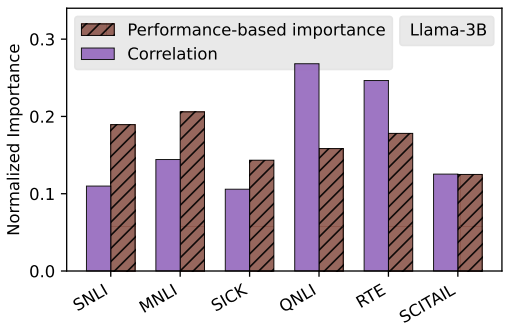


Figure 4. Task importance *vs.* correlation.

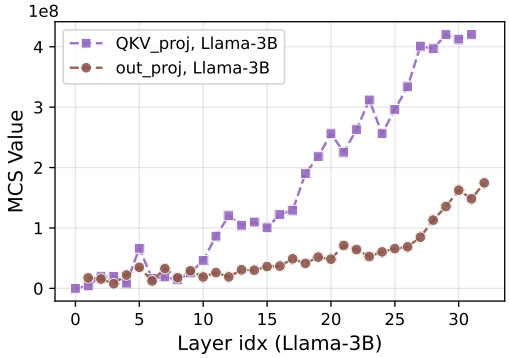
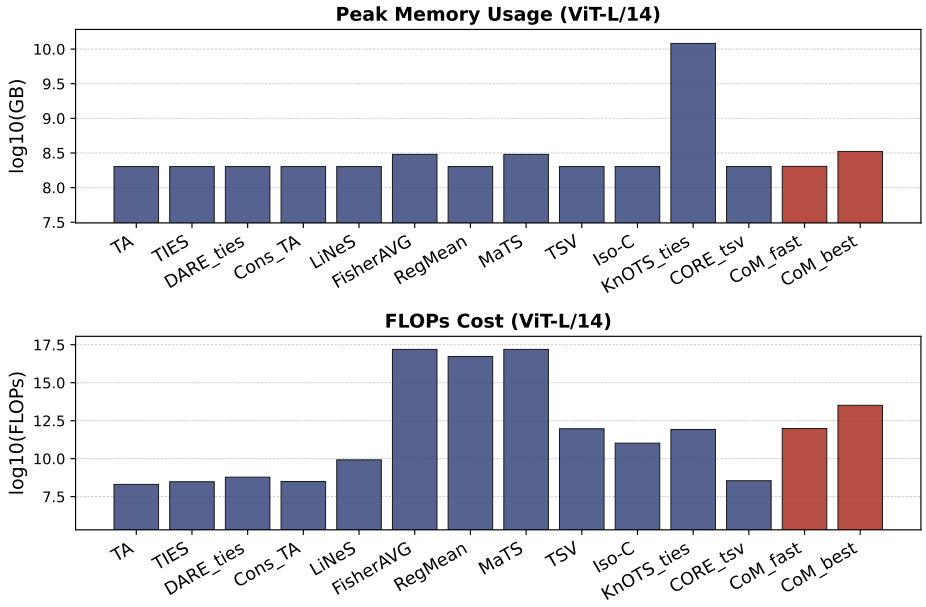
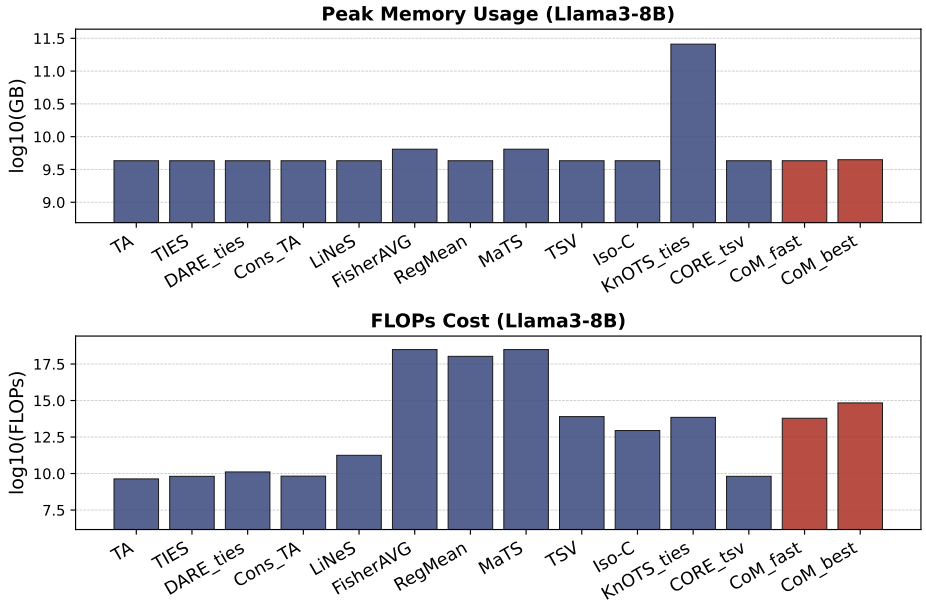


Figure 5. Merging covariate shift across layers.



(a) Peak memory usage and merging computational cost for ViT-L/14.



(b) Peak memory usage and merging computational cost for Llama3-8B.

Figure 6. Comparison of merging methods across computational and memory cost.

C. Computational and Memory Cost

We evaluate the efficiency of the proposed methodology in terms of computational complexity and memory usage, comparing it against established techniques. For the architectures used[1] Figs. 6a and 6b (log-scale) report the number of FLOPs as a measure of computational cost, and the theoretical memory overhead in GB. We denote our methodology with the minimum number of samples achieving state-of-the-art performance (5) as `CoM_fast`, and the variant using the optimal number of samples for maximum performance as `CoM_best`. The results show that CoM achieves performance comparable to

[1]We refer the reader to Fig. 3 for ViT-B/32.

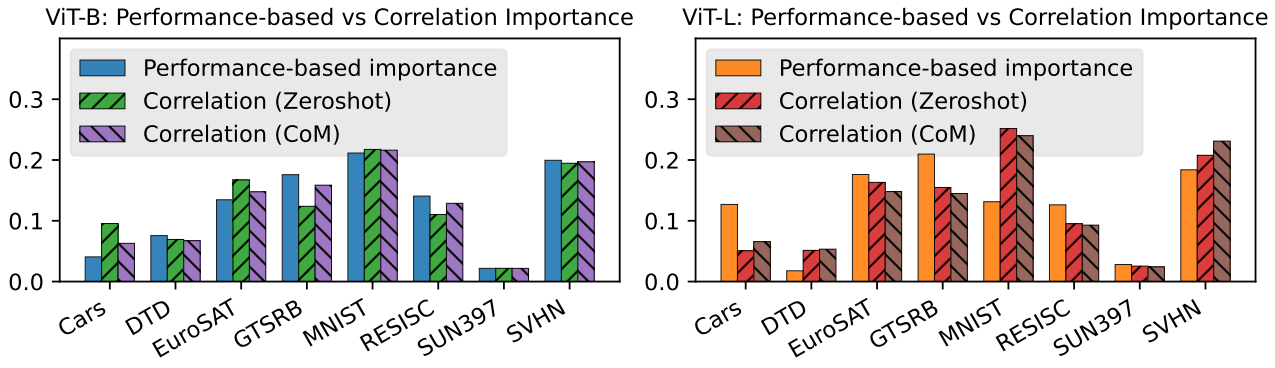


Figure 7. **Ablation of Task Importance Coefficients (ViT-B and ViT-L).** We compare the importance scores derived from our proposed CoM method (which uses the merged model as a reference) against those derived from the Zero-shot pretrained model and a performance-based oracle. The results demonstrate that CoM coefficients strongly correlate with the Zero-shot reference, justifying the use of the merged model as an efficient proxy.

alternative merging strategies. For clarity, methods such as Task Arithmetic, TIES, $\text{DARE}_{\text{TIES}}$, Consensus TA, and LiNeS are omitted from the computational plot, since their complexity is negligible.

A notable comparison is with RegMean. Although our approach shares a similar formulation, it differs in the update rule by using the merged activations instead of the task-specific ones. This key distinction improves accuracy while keeping the number of forward passes unchanged: RegMean requires one forward pass per task-specific model to compute the Gram matrices, whereas CoM performs the same number of passes on the merged model. Finally, CoM requires fewer examples in practice, resulting in consistently lower computational cost, as shown in Tab. 2.

These findings highlight that the proposed recursive scheme provides an effective balance between efficiency and performance, ensuring state-of-the-art accuracy while keeping both computation and memory overhead limited.

D. Analysis of Task Importance Coefficients

In this section, we provide a deeper analysis of the task importance coefficients used in our CoM method. We first elaborate on the theoretical connection between feature correlation and generalization, and subsequently validate our choice of using the merged model as a reference for computing these statistics.

D.1. Theoretical Motivation: Orthogonality and Generalization

Our approach quantifies inter-feature correlation to estimate task importance. We argue that for in-distribution inputs, representations from large-scale pretrained models are approximately decorrelated, as they capture broad, general-purpose structures rather than task-specific patterns.

Consequently, high off-diagonal correlations in the input Gram matrix G indicate that a specific task concentrates on a narrow subspace of the original data distribution, diverging significantly from the pretraining initialization. Intuitively, when a task relies heavily on a limited set of feature directions, those features exhibit higher correlation, revealing that the model is focusing on a restricted region of the representation space. This perspective aligns with established literature demonstrating that feature decorrelation is linked to improved generalization (Cogswell et al., 2015; Morcos et al., 2018). Our method leverages this principle: we treat significant deviations from the decorrelated pretrained state (high inter-feature correlation) as a signal of task specificity.

D.2. Ablation on Reference Models

To maintain comparability across tasks and layers, correlation coefficients must be computed using a single, task-agnostic reference model. Using task-specific fine-tuned models as references would be improper, as this would quantify how well a model fits its own specific task rather than providing a standardized measure of distributional shift.

Table 5. Normalized accuracies of merged models on vision tasks with ViT-B/32, LoRA fine-tuning. Best results in bold, second-best underlined.

ViT-B/32	Cars	DTD	ESAT	GTSRB	MNIST	RESISC	SUN	SVHN	**Average**
TA	81.97	73.72	48.97	42.24	53.12	71.50	97.46	41.25	63.78
TIES	82.37	72.72	49.91	36.62	57.16	69.38	96.92	44.56	63.70
$\text{DARE}_{\text{TIES}}$	82.14	73.72	49.35	37.78	56.63	70.14	97.35	42.12	63.65
Consensus TA	81.21	75.45	52.53	40.33	56.65	71.72	98.41	41.46	64.72
LiNeS	81.82	73.99	49.53	41.08	53.01	71.39	97.63	48.55	63.63
FisherAVG	80.27	73.36	66.82	38.89	71.92	69.67	95.63	63.73	70.04
RegMean	79.89	71.07	37.56	41.82	62.71	71.23	95.73	<u>68.17</u>	66.02
MaTS	80.08	74.09	<u>79.24</u>	39.02	<u>73.10</u>	69.38	95.04	50.16	70.01
TSV	83.44	75.55	50.99	45.03	59.31	73.33	96.40	49.23	66.66
Iso-C	80.16	83.03	51.44	74.76	70.72	79.98	<u>98.96</u>	48.12	70.66
$\text{KnOTS}_{\text{TIES}}$	<u>83.75</u>	74.45	50.36	47.31	67.01	71.79	96.51	50.62	67.73
Core_{TSV}	82.98	<u>85.12</u>	50.95	<u>84.25</u>	71.14	<u>84.39</u>	**99.06**	53.53	<u>76.43</u>
CoM (ours)	**90.17**	**87.11**	**90.33**	**90.50**	**99.22**	**94.36**	89.24	**98.09**	**92.40**

Ideally, the zero-shot (pretrained) model serves as the ground truth reference. However, for computational efficiency within our pipeline, CoM computes correlations using the merged model. We posit that the merged model is a valid proxy because it remains approximately task-agnostic and provides a balanced representation across all tasks.

To validate this approximation, we conducted an ablation study comparing the task importance scores derived from three different sources:

1. **Performance-based importance (Oracle):** importance scores calculated based on actual evaluation performance.
2. **Correlation (Zero-shot):** coefficients computed using the original pretrained checkpoint (the ideal reference).
3. **Correlation (CoM):** coefficients computed using our proposed method with the merged model (the efficient proxy).

The results, illustrated in Fig. 7 for both ViT-B and ViT-L architectures, show a high degree of alignment between the coefficients computed via CoM (purple/brown hatched bars) and those computed via the Zero-shot model (green/red hatched bars). This confirms that using the merged model to compute the Gram matrices does not introduce significant deviation from the ideal pretrained reference, validating the efficiency of our protocol. Furthermore, both correlation-based metrics track the general trends of the performance-based oracle, particularly in identifying high-importance tasks such as EuroSAT and SVHN.

E. Accuracies on each dataset

Following (Stoica et al., 2025; Panariello et al., 2025), we report the normalized accuracies on each dataset in Tab. 5 for ViT-B/32, in Tab. 6 for ViT-L/14 and in Tab. 7 for Llama3-8B with LoRA fine-tuning.

Table 6. Normalized accuracies of merged models on vision tasks with ViT-L/14, LoRA fine-tuning. Best results in bold, second-best underlined.

ViT-L/14	Cars	DTD	ESAT	GTSRB	MNIST	RESISC	SUN	SVHN	**Average**
TA	80.01	79.50	65.59	59.98	82.20	79.55	86.71	64.74	74.79
TIES	79.65	78.28	64.43	61.10	83.82	79.42	87.45	69.94	75.51
$\text{DARE}_{\text{TIES}}$	79.70	78.82	64.99	60.63	83.92	79.32	87.07	69.84	75.53
Consensus TA	81.88	81.32	68.60	64.55	85.04	81.56	86.90	63.72	76.70
LiNeS	80.89	79.88	65.25	59.74	81.86	79.37	86.69	64.37	74.65
FisherAVG	76.61	77.07	48.72	48.45	89.93	78.39	86.95	<u>96.45</u>	75.32
RegMean	77.81	77.98	53.08	53.91	59.75	78.87	86.91	70.53	69.85
MaTS	77.80	78.44	57.89	55.19	85.15	79.52	86.35	87.44	75.97
TSV	82.38	80.11	66.12	68.18	85.46	83.02	87.89	70.76	77.99
Iso-C	86.83	86.94	<u>80.65</u>	78.36	<u>92.09</u>	87.88	88.50	68.69	83.70
$\text{KnOTS}_{\text{TIES}}$	82.47	80.26	64.65	68.85	88.48	82.37	<u>88.18</u>	76.63	78.99
Core_{TSV}	**91.54**	**91.34**	80.24	<u>86.79</u>	87.39	**91.51**	**89.59**	71.30	<u>86.21</u>
CoM (ours)	<u>88.72</u>	<u>87.85</u>	**87.27**	**92.77**	**99.37**	<u>90.96</u>	83.71	**97.81**	**91.06**

Table 7. Normalized accuracies of merged models on language with Llama3-8B. Best results are highlighted in bold, second-best underlined.

Method	SNLI	MNLI	SICK	QNLI	RTE	SCITAIL	**Average**
TA	93.57	95.28	87.96	68.71	100.00	96.73	90.38
TIES	95.17	96.19	84.18	74.18	100.00	96.78	91.08
$\text{DARE}_{\text{TIES}}$	94.76	<u>96.80</u>	78.39	72.08	98.39	96.20	89.44
Consensus TA	93.62	93.58	91.43	66.82	101.61	97.66	90.79
LiNeS	93.31	95.25	88.74	71.13	100.00	96.59	90.84
RegMean	<u>97.67</u>	96.32	79.79	65.17	96.78	89.77	87.58
TSV	95.38	95.12	88.83	76.80	101.60	97.56	92.55
Iso-C	55.00	39.04	76.54	55.90	46.77	69.25	57.08
$\text{KnOTS}_{\text{TIES}}$	91.82	94.19	<u>92.97</u>	78.57	100.00	97.61	92.53
Core_{TSV}	95.86	95.70	89.25	<u>83.89</u>	**102.42**	<u>97.86</u>	<u>94.16</u>
CoM (ours)	**99.00**	**97.34**	**99.43**	**100.12**	<u>101.61</u>	**99.46**	**99.50**