Rethinking Layer-wise Model Merging through Chain of Merges

Pietro Buzzega **Riccardo Salami** **Angelo Porrello** **Simone Calderara**

AImageLab - University of Modena and Reggio Emilia, Modena, Italy
name.surname@unimore.it

Abstract

Model merging has emerged as a simple yet effective method to fuse models without re-training. Existing 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 *covariate shift* that compounds across the network. To address this issue, we propose *Chain of Merges* (CoM), a novel procedure that sequentially merges weights across layers while accounting for inter-layer interactions. In particular, CoM mitigates covariate shift through a series of regression problems, where input activations are recomputed at each step to reflect the updated representations. Additionally, we introduce a dynamic weighting mechanism that prioritizes task-layer pairs most susceptible to performance degradation, improving robustness during merging. Experiments on standard benchmarks demonstrate that CoM achieves state-of-the-art performance across heterogeneous tasks. github.com/aimagelab/chain-of-merges

1 Introduction

The availability of large-scale pretrained models has reshaped machine learning [44, 56], with fine-tuning offering an accessible path to state-of-the-art performance across domains [45, 58]. As these models are adapted to specialized tasks, a natural question arises: *how can we combine task-specific checkpoints without retraining?* This challenge, known as model merging [23, 36, 62], enables modularity, knowledge reuse, and efficient deployment.

Since specialized modules are trained independently, there is no guarantee that their weights can be seamlessly combined [67, 51]. Naive strategies like weight averaging [37, 62] often cause performance degradation [54, 11, 53]. To address this, current literature relies on a variety of heuristics, *e.g.* interference mitigation [67, 70], permutation-based parameter alignment [2, 49], critical weight preservation [36, 32], and interpolation within orthonormal or task-adaptive subspaces [34, 14, 53].

While these approaches achieve competitive performance, they still present notable limitations. They often rely on problem-specific design choices and extensive hyperparameter tuning, while lacking a unifying theoretical foundation that provides strong guarantees in real-world deployment settings. Motivated by these shortcomings, a complementary line of work focuses on **aligning model activations** [51, 25, 55, 26], among which RegMean [25] is the closest to our work. By operating in activation space, these methods cast model merging as an interpretable, layer-wise optimization problem aimed at preserving the functional behavior of each layer after merging. In many cases, these are simple regression problems that admit closed-form solutions, leading to more stable behavior. This perspective generally provides stronger theoretical grounding, albeit at the cost of requiring additional data to estimate activation statistics.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

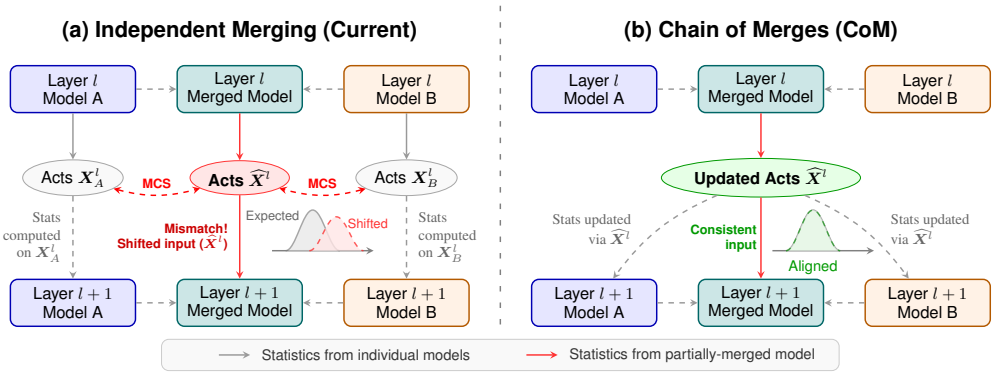


Figure 1: (a) Independent merging induces a distribution shift as Layer $l+1$ receives mismatched activations. (b) CoM recomputes statistics using updated activations, realigning the distribution.

While these activation-based approaches lay the foundation for more principled, interpretable, and trustworthy model merging, they overlook a **simple but crucial** factor: each layer is conditioned on the outputs of preceding computations. Indeed, these approaches merge layers **independently**, proceeding sequentially from one layer to the next. As this process unfolds, the output distribution of each layer shifts due to the merging operations applied to preceding layers. Current methods, starting from RegMean [25], fail to account for this shift and instead rely on the original activation distributions – namely, those computed from the individual models prior to merging.

As a principal contribution, we identify this issue as a form of *covariate shift* [24], a well-known phenomenon in training dynamics where rapidly changing early-layer activations induce unstable output distributions that hinder downstream learning [4]. By analogy, we refer to its manifestation in model merging as ***Merging Covariate Shift*** (MCS), which arises when early layers are modified through merging, causing shifts in the inputs to subsequent layers. In this respect, we show that merging layers in isolation inherently alters the distribution of their output activations, resulting in misaligned inputs for downstream layers. This effect compounds across the network: even small discrepancies propagate and amplify, causing progressively larger mismatches and consequent performance degradation. This naturally leads to the following research question:

How can we design a layer-wise merging procedure that accounts for inter-layer interactions and prevents the propagation of distributional mismatches across layers?

Building on this crucial yet overlooked limitation, we propose **Chain of Merges (CoM)**, illustrated in Fig. 1. CoM is a recursive, layer-wise merging method that updates activation statistics after each merging step, replacing the original task-specific activations with those produced by the partially merged model. In doing so, CoM explicitly accounts for inter-layer dependencies and the MCS issue. As a second result, we show that this simple modification can dramatically improve performance, enabling our approach to surpass the state of the art by a significant margin.

As a third contribution, we emphasize that not all tasks are equally important or equally difficult. An activation-based approach should account for this heterogeneity by prioritizing certain tasks over others during optimization. In particular, we hypothesize that models which deviate more strongly from the base model are also more susceptible to performance degradation. We introduce a **dynamic weighting mechanism** into our framework, designed to better preserve these more vulnerable representations. We further show that a proxy for such susceptibility – and the resulting performance degradation – can be directly estimated from the feature correlation matrix, which is the same input we leverage to solve the optimization problem underlying CoM.

To sum up: **(i)** we identify and analyze covariate shift in model merging, showing that activation mismatches accumulate in deep networks, and propose Chain of Merges (CoM), a method that progressively merges parameters while updating activation statistics to ensure consistency across the merged network; **(ii)** we introduce an importance-weighting mechanism that biases the solution toward task-layer pairs most divergent from the base model, thereby preserving their features; and **(iii)** we evaluate CoM on standard model merging benchmarks across language and vision settings, showing that it outperforms existing methods by a large margin on LoRA fine-tuning.

2 Towards Understanding *Merging Covariate Shift*

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 $l \in \{1, 2, \dots, L\}$, 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%).

Among activation-based approaches, RegMean [25] 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 $\{\boldsymbol{X}_i^l\}_{i=1}^N$, with $\boldsymbol{X}_i^l \in \mathbb{R}^{d \times T_i}$ having one column per input token. This is accomplished by minimizing the following objective:

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

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:

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

Here, the Gram matrix $\boldsymbol{X}_i^l {\boldsymbol{X}_i^l}^\top \in \mathbb{R}^{d \times d}$ captures the pairwise correlations between input features.

2.1 Merging Covariate Shift

To solve the problem in Eq. 1, RegMean [25] and related methods compute layer-wise statistics of the network, using as reference the model trained individually on each task. In particular, denoting by $\boldsymbol{X}_i^l$ the activations of model i at layer l, these are defined recursively as

$$\boldsymbol{X}_i^{l+1} = \boldsymbol{\sigma}\left(\boldsymbol{W}_i^l \boldsymbol{X}_i^l\right), \quad l = 1, \dots, L-1, \tag{3}$$

starting from the input $\boldsymbol{X}_i^1$. For simplicity of notation, we assume that the activation function $\boldsymbol{\sigma}$ is the same across all layers. Consequently, up to layer $l+1$, for each model i we obtain the following **target activations** for the RegMean problem:

$$\boldsymbol{X}_i^2 = \boldsymbol{\sigma}\left(\boldsymbol{W}_i^1 \boldsymbol{X}_i^1\right), \dots, \boldsymbol{X}_i^l = \boldsymbol{\sigma}\left(\boldsymbol{W}_i^{l-1} \boldsymbol{X}_i^{l-1}\right), \boldsymbol{X}_i^{l+1} = \boldsymbol{\sigma}\left(\boldsymbol{W}_i^l \boldsymbol{X}_i^l\right). \tag{4}$$

At inference time, the original parameters $\boldsymbol{W}_i^{l-1}$ are replaced with their merged counterparts $\boldsymbol{W}_M^{l-1}$, obtained by solving Eq. 2. Consequently, the activations shift accordingly and become:

$$\widehat{\boldsymbol{X}}_i^2 = \boldsymbol{\sigma}\left(\boldsymbol{W}_M^1 \boldsymbol{X}_i^1\right), \dots, \widehat{\boldsymbol{X}}_i^l = \boldsymbol{\sigma}\left(\boldsymbol{W}_M^{l-1} \widehat{\boldsymbol{X}}_i^{l-1}\right), \widehat{\boldsymbol{X}}_i^{l+1} = \boldsymbol{\sigma}\left(\boldsymbol{W}_M^l \widehat{\boldsymbol{X}}_i^l\right). \tag{5}$$

Problem. The statistics used to merge layer $l+1$ are computed using the original parameters $\boldsymbol{W}_i^1, \dots, \boldsymbol{W}_i^l$ prior to merging. However, at inference time, the preceding layers have already been replaced by their merged counterparts $\boldsymbol{W}_M^1, \dots, \boldsymbol{W}_M^l$. As a result, the input activations to layer $l+1$ differ from those used during the estimation of the merging statistics. This mismatch induces a shift in the activation distributions, analogous to internal covariate shift, which we refer to as *Merging Covariate Shift* (MCS).

To make MCS measurable, we follow the common view of covariate shift as a distributional discrepancy [21], and model the activation inputs of each layer as multivariate Gaussians. In particular, we consider the pre-merging activations $\boldsymbol{X}_i^l$, computed using the original parameters $\boldsymbol{W}_i^{1:l-1}$, and the post-merging activations $\widehat{\boldsymbol{X}}_i^l$, obtained with the merged counterparts $\boldsymbol{W}_M^{1:l-1}$. We hence quantify the distance between these two distributions using the squared 2-Wasserstein distance (also known as Fréchet distance) [57, 13], which admits a closed form for Gaussians:

$$\text{MCS}^l = \sum_{i=1}^{N} \left[\left\| \mu_i^l - \widehat{\mu}_i^l \right\|_2^2 + \text{Tr}\left[\Sigma_i^l + \widehat{\Sigma}_i^l - 2 \left(\left(\widehat{\Sigma}_i^l\right)^{1/2} \Sigma_i^l \left(\widehat{\Sigma}_i^l\right)^{1/2} \right)^{1/2} \right] \right], \tag{6}$$

where μ_i^l, Σ_i^l are the mean and covariance of $\boldsymbol{X}_i^l$, and $\widehat{\mu}_i^l, \widehat{\Sigma}_i^l$ are those of $\widehat{\boldsymbol{X}}_i^l$.

As shown in Fig. 2, evaluated on 8-Vision [22], MCS is present across all layers and increases sharply with depth, suggesting that mismatches compound as they propagate through the network. Notably, this effect is significantly more pronounced in larger models (ViT-L), suggesting that model scale exacerbates the accumulation of distributional mismatches. We also analyze the two projections before and after attention separately, as they exhibit different magnitudes: QKV projections show substantially higher MCS than output projections, indicating that certain components of the attention mechanism are more sensitive to merging-induced perturbations.

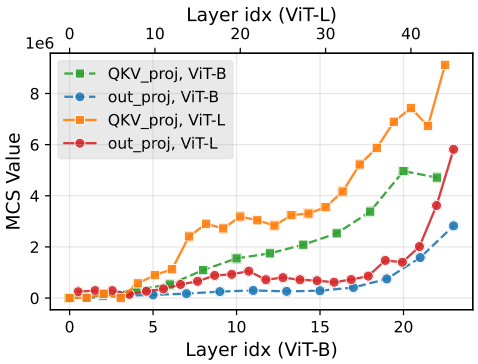


Figure 2: MCS across layers.

3 Chain of Merges

Recursive Dependence. To address MCS, we revise the closed-form solution presented in Eq. 2. Instead of relying on the stale inputs $\boldsymbol{X}_i^l$, computed using the original parameters $\boldsymbol{W}_i^1, \ldots, \boldsymbol{W}_i^{l-1}$, we employ the activations $\widehat{\boldsymbol{X}}_i^l$ produced by the preceding **merged** layers, i.e., using $\boldsymbol{W}_M^1, \ldots, \boldsymbol{W}_M^{l-1}$. These correspond to the actual inputs received by layer l at inference time, once all preceding layers have been merged. Substituting $\boldsymbol{X}_i^l$ with $\widehat{\boldsymbol{X}}_i^l$ yields the revised expression for the merged weights:

$$\boldsymbol{W}_M^l = \left(\sum_{i=1}^{N} \boldsymbol{W}_i^l \widehat{\boldsymbol{X}}_i^l \widehat{\boldsymbol{X}}_i^{l\top} \right) \left(\sum_{i=1}^{N} \widehat{\boldsymbol{X}}_i^l \widehat{\boldsymbol{X}}_i^{l\top} \right)^{-1}. \tag{7}$$

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 through the entire network. By ensuring that the statistics computed at each stage reflect the accumulated effect of all preceding merges, this autoregressive scheme mitigates Merging Covariate Shift by design.

We can summarize the CoM procedure as follows:

Initial step. The recursive chain starts at the first layer, where the inputs correspond to raw data and are unaffected by prior merging. Thus, the merged weights can be computed using Eq. 2.

Recursive step. The merged weights obtained at the initial step are used to propagate consistent activations throughout the network. For subsequent layers $l = 2, \ldots, L$, the algorithm proceeds iteratively, alternating between computing the activations $\widehat{\boldsymbol{X}}_i^l$ – which account for merging in preceding layers – and updating the merged weights $\boldsymbol{W}_M^l$ according to Eq. 7.

Fast implementation. Importantly, our recursive formulation incurs a computational cost that is **linear** in the number of merged models. The overall forward computation requires exactly one forward pass per model – matching the cost of RegMean – despite the sequential nature of the merging procedure and the fact that activation statistics are updated at every layer. To achieve this efficiency, we implement CoM by interleaving merging and propagation at the layer level. Specifically, we leverage forward pre-hooks attached to each fine-tuned linear projection and maintain, for each task i, a single activation buffer that stores the input to the **next** target layer in the partially merged model.

3.1 Correlation-based importance

Although our method addresses the Merging Covariate Shift, the merged layer still cannot exactly reproduce the outputs of every task-specific layer. A single matrix $\boldsymbol{W}_M^l$ must match the activations $\boldsymbol{W}_i^l \widehat{\boldsymbol{X}}_i^l$ of N distinct layers on their respective inputs, and, in general, no single linear map achieves this for all tasks at once. The least-squares solution of Eq. 7 therefore leaves a residual error, which perturbs the activations of the merged model on every task.

Given that an error during merging is therefore unavoidable, the remaining question is how to distribute this error across the different tasks. *Are there tasks that are more important than others?*

Task Importance. The goal of model merging is to retain performance across all tasks; however, their importance is not uniform. In this regard, we distinguish two categories of tasks:

- `Category (a)` Tasks that are semantically **similar** to the pretraining distribution, and therefore naturally benefit from the representations of the base model.
- `Category (b)` Tasks that are semantically **distant** from pretraining, which cannot rely on pretrained features and are thus more prone to performance degradation after merging.

We argue that preserving `Category (b)` tasks is more critical. Indeed, as they deviate more from the base model, they are also more likely to suffer larger losses 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, reflecting the categorization introduced above: *the more* distant a task is from the base model, *the more* important it is to preserve it.

Feature correlation as a proxy. We require a task-layer importance score ω_i^l that can be computed during merging, without labels. We construct this score based on the **redundancy** of input activations:

> If, for a given task, the activations at a certain layer are highly redundant, we interpret this as a signal that the model **has collapsed** its representations to accommodate the task, and thus that the task **has significantly deviated** from the base model.

For an empirical assessment of this intuition, in Sec. 4.3 (Fig. 6) we show that weight scores based on activation redundancy are *linearly correlated* with a performance-based notion of task importance.

More precisely, we measure how strongly different activation directions co-vary, and use this as a signal of how much task i has specialized away from the pretrained features at layer l. Large-scale pretraining encourages representations that are broad and approximately decorrelated, as they must support many downstream uses [10, 38]. When a model is fine-tuned, this generality is partially sacrificed: activations collapse onto a narrower subspace, and previously independent feature directions begin to align. The amount of off-diagonal mass in the activation correlation matrix thus provides a direct measure of how far the task has pushed the layer away from the pretrained, decorrelated state. Tasks that are semantically distant from pretraining rely most on this specialization, and are therefore assigned greater weight during optimization.

CoM already produces the inputs $\widehat{\boldsymbol{X}}_i^l$ required by Eq. 5. We reuse them after normalizing each token (column of $\widehat{\boldsymbol{X}}_i^l$) to unit ℓ_2 norm (see App. A.2 for more details) to obtain $\tilde{\boldsymbol{X}}_i^l$, and form the correlation matrix $\boldsymbol{G}_i^l = \frac{1}{T_i}\tilde{\boldsymbol{X}}_i^l \tilde{\boldsymbol{X}}_i^{l\top}$. The importance score is the off-diagonal ℓ_1 mass of $\boldsymbol{G}_i^l$, which captures feature coupling:

$$\omega_i^l = \sum_{p<q} |(\boldsymbol{G}_i^l)_{pq}|. \tag{8}$$

A large ω_i^l flags a layer that has narrowed toward a task-specific subspace and is therefore fragile under merging, while a small value indicates near-orthogonal features and low sensitivity. In App. B.2, we further relate the decorrelation of pretrained features to generalization, and show that computing ω_i^l with the merged model instead of the pretrained one yields closely matching scores.

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:

$$\underset{\boldsymbol{W}_M^l}{\arg\min} \sum_{i=1}^{N} \frac{\omega_i^l}{T_i} \left\| \boldsymbol{W}_M^l \tilde{\boldsymbol{X}}_i^l - \boldsymbol{W}_i^l \tilde{\boldsymbol{X}}_i^l \right\|_F^2 = \Big(\sum_{i=1}^{N} \omega_i^l \, \boldsymbol{W}_i^l \, \boldsymbol{G}_i^l\Big)\Big(\sum_{i=1}^{N} \omega_i^l \, \boldsymbol{G}_i^l\Big)^{-1}. \tag{9}$$

The complete procedure is summarized in Alg. 1, with further implementation details in App. A.2. The full derivation of Eq. 9 is provided in App. A.1.1, while the special case of normalization layers is addressed in App. A.1.2.

4 Experiments

Experimental Setup. We evaluate our approach across both **vision** and **language** domains. For vision, we use the benchmark of [23], which involves merging checkpoints from eight classification datasets: Stanford Cars [29], DTD [9], EuroSAT [17], GTSRB [50], MNIST [31], RESISC45 [8], SUN397 [65], and SVHN [39]. For language, following [52, 41], we evaluate on six datasets: SNLI [7], MultiNLI [61], SICK [35], SciTail [28], RTE [58], and QNLI [58]. SNLI, MultiNLI, and SICK are three-way classification tasks (entailment, contradiction, neutral), while SciTail, RTE, and QNLI are binary classification tasks.

Competitors. We compare against methods from the main model merging families: task-vector and interference-reduction methods (TA [23], TIES [67], DARE [70], Consensus TA [59], LiNeS [60]), Fisher or regression-based activation matching (FisherAVG [36], RegMean [25], MaTS [53]), SVD/subspace methods (TSV [14], Iso-C [34], KnOTS [52], Core_{TSV} [41], Core_{ISO} [41]), and data-driven layer-wise optimization methods (AdaMerging++ [68], ProDistill [66], Localize-and-Stitch [16]). We refer the reader to Sec. 5 for a detailed comparison with these approaches. Importantly, we compare our approach against merging techniques involving **both full fine-tuning and LoRA fine-tuning** [19], evaluating them in two distinct phases.

Evaluation protocol. All our experiments follow a *static merging* protocol as in [52, 22, 14, 41]: each method outputs a single merged backbone used for all tasks at inference time. No task identifiers, routing mechanisms, 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. We report both the average normalized and absolute accuracies of the merged model across all datasets. *Zero-shot* denotes CLIP's zero-shot performance, while *Individual FT* denotes the performance of each fine-tuned model.

Implementation details. Following [52, 14, 23], we use ViT-B/32 and ViT-L/14 [12] CLIP encoders as the vision-task backbones. For natural language tasks, we utilize Llama-3-8B [15]. Each model is fine-tuned using either LoRA [19] or full 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 reproducibility and fair comparison, we employ the LoRA fine-tuned checkpoints provided by [52], and the full fine-tuning checkpoints from [23]. We use `bfloat16` for NLP tasks – as it was shown to generally outperform `float16` [27] – except during Gram-matrix inversion, where `float32` is used for numerical stability. To mitigate conditioning issues in language experiments (see App. A.2), we use the Moore–Penrose pseudoinverse together with Tikhonov regularization [18] applied as in RegMean [25] ($\alpha = 0.95$ multiplying off-diagonal terms).

Calibration and hyperparameter selection. For constructing the Gram matrices, we draw balanced examples across tasks and classes; if the number of classes exceeds the number of examples, classes are randomly subsampled. Calibration examples are drawn from the validation set for CoM (500 examples per task; training set on Llama-3-8B, as RTE and SICK have fewer validation examples), and their labels are only used to balance classes. CoM results are averaged over three random draws of the calibration set, whereas the other baselines are run once as they are deterministic (either not using data or using the entire training set). We use the entire training set to estimate RegMean's Gram matrices following the original work [25], and discuss the sensitivity of CoM to the construction of the calibration set in App. C.1. Following previous works [67, 52, 14], all hyperparameters are tuned on the entire validation set for each method.

4.1 Comparison with the State-of-the-Art

Vision tasks – LoRA-based Model Merging. We refer the reader to Tab. 1 for a comprehensive comparison. On the smaller ViT-B/32 model, parameter-space baselines retain only moderate performance, with TA, TIES, DARE, Consensus TA, and LiNeS remaining in the mid-60s in normalized accuracy. Stronger baselines such as FisherAVG, MaTS, Iso-C, and Core_{ISO} improve over these methods, but CoM achieves a much larger gain, surpassing the best baseline by more than 16 points (Tab. 1). On ViT-L/14, most methods improve and the gap between simple and advanced baselines narrows; nevertheless, CoM remains the best method, reaching 91.35% normalized accuracy versus 86.08% for Core_{ISO}. CoM improves over RegMean by 19.4 and 8.2 points on ViT-B/32 and ViT-L/14, respectively, isolating the benefit of correcting MCS within the same activation-matching objective.

Table 1: Normalized and Absolute Average accuracies (%) of merged LoRA models.

	ViT-B/32		ViT-L/14		Llama-3	
Method	**Norm**	**Abs**	**Norm**	**Abs**	**Norm**	**Abs**
Zero-shot	60.6	48.1	71.4	65.3	51.1	47.4
Indiv. FT	100.0	84.0	100.0	92.3	100.0	92.5
TA	63.9	51.4	74.8	68.6	90.3	83.5
TIES	64.1	51.6	75.5	69.4	90.3	83.5
$\text{DARE}_{\text{TIES}}$	64.1	51.7	75.0	68.8	90.0	83.2
Cons. TA	65.3	52.7	76.7	70.4	90.4	83.6
LiNeS	63.5	51.0	74.3	68.1	90.8	84.0
FisherAVG	68.1	55.4	77.5	71.2	—	—
RegMean	73.3	61.4	83.1	76.7	94.8	87.7
MaTS	68.3	55.6	77.5	71.1	—	—
TSV	66.8	54.1	77.8	71.5	92.2	85.2
Iso-C	73.4	59.9	84.7	77.9	70.8	65.6
$\text{KnOTS}_{\text{TIES}}$	68.3	55.7	79.1	72.7	92.9	85.9
CORE_{TSV}	66.6	53.8	78.2	71.9	94.2	87.1
CORE_{ISO}	76.1	62.3	86.1	79.2	90.4	83.6
CoM	**92.7**	**78.4**	**91.4**	**84.6**	**99.8**	**92.3**

Table 2: Normalized and Absolute Average accuracies (%) of merged full fine-tuned models.

	ViT-B/32		ViT-L/14	
Method	**Norm**	**Abs**	**Norm**	**Abs**
Zero-shot	54.8	48.1	69.8	65.3
Indiv. FT	100.0	90.5	100.0	94.1
TA	76.7	69.4	88.9	83.9
TIES	81.1	73.4	90.8	85.6
Cons. TA	81.6	74.1	90.6	85.5
LiNeS	81.0	73.6	91.0	85.9
FisherAVG	69.7	61.9	81.7	76.7
RegMean	87.5	79.8	93.7	88.6
Loc.-Stitch	86.3	79.9	90.4	86.5
AdaMerge++	87.6†	81.1†	95.1†	91.0†
ProDistill	92.9†	86.0†	96.1†	91.9†
TSV	92.2	83.7	96.9	91.3
ISO-C	92.4	83.8	**98.2**	**92.4**
CoM	**94.8**	**86.1**	97.6	92.0

Language tasks – LoRA-based Model Merging. Results on the six natural language benchmarks with Llama-3-8B (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 (90.27% on average). More sophisticated parameter-based techniques provide small gains w.r.t. TA, while RegMean becomes the strongest baseline. Iso-C is a clear outlier, likely because singular-value rescaling interacts poorly with language-model fine-tuning. CoM delivers near-perfect merged performance, surpassing the strongest baseline (RegMean) by 5 points. We omit FisherAVG and MaTS from this comparison because computing the Fisher Information Matrix or the corresponding linear systems for Llama-3-8B is prohibitively expensive.

Vision tasks – Full Fine-Tuning Model Merging. With full fine-tuning, performance improves substantially across all methods[1] compared to LoRA (Tab. 2). On ViT-B/32, simple baselines already become competitive, while ProDistill, TSV, and Iso-C are the strongest baselines, all above 92% normalized accuracy. CoM further advances the state of the art, achieving $94.8\%/86.1\%$ normalized and absolute accuracy. Scaling to ViT-L/14 strengthens most baselines, with AdaMerging++, ProDistill, TSV, and Iso-C all surpassing 95% normalized accuracy. CoM remains on par with the best-performing method, confirming that the proposed recursive activation matching is also effective beyond low-rank adaptation. This overlap occurs solely on ViT-L/14, where performance is essentially saturated (Iso-C: 98.2% *vs.* CoM: 97.6% normalized accuracy).

Overall. CoM consistently achieves the best or on-par performance across all settings. The gap with respect to RegMean highlights the importance of mitigating Merging Covariate Shift for effective activation-based merging. Moreover, we highlight that the gains are more pronounced in the LoRA regime than with full fine-tuning. This is consistent with prior findings that LoRA checkpoints are harder to merge than fully fine-tuned ones [52, 71]. We hypothesize that LoRA's restriction of adaptation to a few directions encourages different tasks to encode their knowledge in overlapping dominant directions of the pretrained model. The resulting interference may amplify errors introduced in early layers as they propagate through this restricted adaptation space, making the correction of Merging Covariate Shift particularly important. With full fine-tuning, updates can span broader subspaces, potentially reducing interference; moreover, on ViT-L/14 the benchmark is essentially saturated, leaving little room for improvement (see App. B.4 for the full discussion).

Additional results. App. D.2 reports full fine-tuning results on the language tasks, and App. D.1 reports the accuracy on each individual dataset for the LoRA-merged models.

[1] Core_{TSV} and Core_{ISO} are not applicable to full fine-tuning, as they require factorized (LoRA) updates.

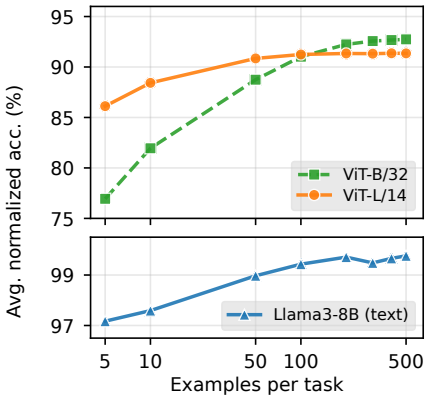


Figure 3: **Sample efficiency.** Accuracy of CoM (LoRA) *vs.* calibration examples per task, on vision (top) and text (bottom).

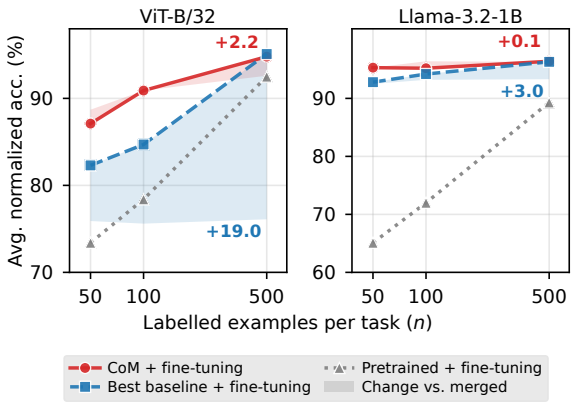


Figure 4: Fine-tuning after merging on the same n examples per task; shading and numbers: change *vs.* the merged model (best baseline picked for each n).

4.2 Data efficiency of model merging

Model merging methods differ in how they use data: weight-space approaches are data-free, apart from the validation data used to tune their hyperparameters, whereas activation-space methods such as RegMean and CoM estimate activation statistics on a small calibration set. At first glance, this reliance on data is the **main limitation** of our method. We herein assess how critical this dependency is in practice, by varying how much data is available, where it comes from, and what it is used for.

How much data do we need? Fig. 3 varies the number of calibration examples per task. CoM is highly data-efficient: already competitive with a handful of samples, it approaches its maximum at roughly 100 examples, after which gains saturate. A minimum number of examples is thus required to stabilize the Gram matrices of the closed-form solution.

If we have labeled data, why not fine-tune? To assess how efficiently merging exploits calibration data, we use the same n examples per task, drawn from all tasks, to fine-tune three starting points: the pretrained model (the naive choice), the CoM merge, and the best competing merge (Fig. 4; details in App. C.3). Initializing from a merged model is consistently more advantageous than starting from the pretrained one, which suggests that the prior of the specialists is preserved and useful. The choice of the initialization matters most in the low-data regime, whereas with more data it matters less, as the data end up compensating for it. Moreover, fine-tuning brings CoM only marginal gains ($+2.2$ and $+0.1$ points at $n = 500$ on ViT-B/32 and Llama-3.2-1B) and, with few examples, strongly hurts it, whereas the best competing merge gains much more ($+19.0$ and $+3.0$), showing that CoM already reaches a good optimum, extracting most of the knowledge that the available data can provide.

Can we use generated data for calibration? We further test whether calibration requires real data from the true task distribution, or whether it can be replaced by surrogate data: we sample synthetic images from a generative model and use them for calibration. For each vision task, we generate 500 images with Stable Diffusion v1.5 [47] using one class-conditioned prompt per image, and use them in place of both the calibration and the validation data; the hyperparameters of the data-free baselines are selected on the same synthetic images. On the LoRA checkpoints (Fig. 5; details in App. C.2), CoM remains the best-performing method on both backbones, ahead of the strongest baseline (Core_{ISO}) by 7.4 points on ViT-B/32 and 4.1 points on ViT-L/14. Although the margin is smaller than with real data, this suggests that CoM is fairly robust to the choice of the calibration samples.

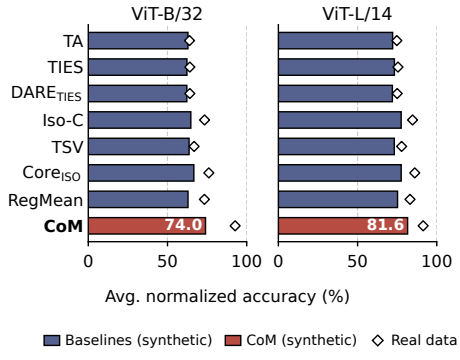


Figure 5: Baselines and CoM (LoRA) calibrated on 500 **synthetic images** per task (bars); diamonds: methods with real data.

Table 3: Impact of CoM components on LoRA models (avg. normalized accuracy): *MCS* (covariate shift correction) and *FC* (feature correlation weighting). All rows use activation normalization.

Components		Architecture		
MCS	FC	ViT-B/32	ViT-L/14	Llama-3-8B
✗	✗	74.32	83.05	97.52
✓	✗	85.83	87.38	99.76
✗	✓	84.02	88.27	97.52
✓	✓	92.74	91.35	99.63

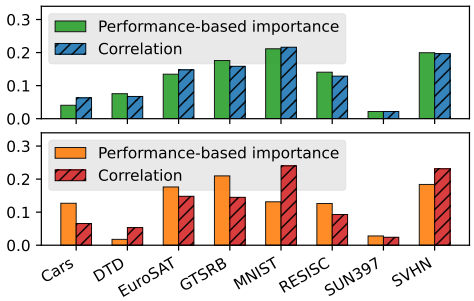


Figure 6: Task importance *vs.* feature correlation for ViT-B (above) and ViT-L (below).

4.3 Impact of MCS and feature correlation on merging

We quantify the impact of the two main contributions of CoM (correcting Merging Covariate Shift via recursive updates, and feature-correlation weighting) in the most challenging LoRA setting (Tab. 3). On both vision backbones, either addressing Merging Covariate Shift or feature-correlation weighting is sufficient to surpass all competing approaches (Tab. 1), and the two components are complementary, as their combination yields the best results. On language, feature-correlation weighting brings no gain. We conjecture that the negligible impact of weighting tasks on language is due to these tasks being more similar and equally complex, with a shared label space, whereas vision tasks are very diverse and heterogeneous, e.g., in their out-of-distribution distance from the pretrained model. A more detailed discussion is in App. B.1 and App. B.3, and the full ablation including activation normalization is in Tab. 4 of App. A.2.

Feature correlation. To further motivate correlation-based weighting, we test whether our weighting factor ω_i^l correlates with a notion of "*semantic distance*" between each fine-tuned model and the base model, which we intuitively quantify as the **performance gap** between the two, i.e., how much fine-tuning improves the task. Fig. 6 compares this gap with our correlation-based weighting factor, averaged across layers to obtain a single value per task[2]. The two measures exhibit a **clear correlation** across vision datasets and architectures, corroborating our claims (Sec. 3.1). Using the pretrained model instead of the merged one to compute the correlation scores yields closely aligned results (see App. B.2). On language, instead, tasks are more homogeneous and the correlation becomes weaker (App. B.3), consistently with the absence of gains from FC weighting on Llama-3-8B (Tab. 3).

4.4 Computational costs

We assess computational efficiency (FLOPs) and memory overhead for ViT-B/32 in Fig. 7, distinguishing between `CoM_fast` (5 samples per task, on par with state-of-the-art methods whose hyperparameters are tuned on the whole validation set) and `CoM_best` (500 samples per task). CoM maintains a resource profile **comparable to other SOTA methods** while being more efficient than RegMean, which requires the entire dataset for calibration. 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 App. A.3 for experimental results on additional architectures.

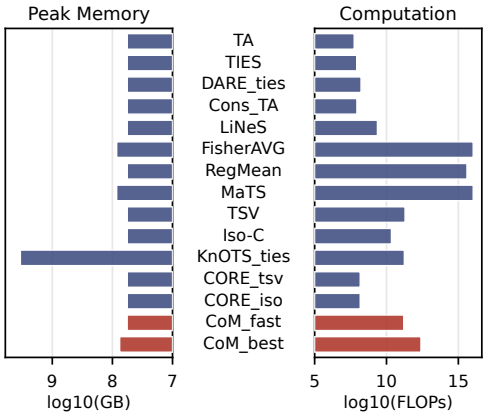


Figure 7: Peak memory (left), FLOPs (right) on ViT-B/32.

[2]Both quantities are min-max scaled and ℓ_1-normalized for visualization; since this is a positive affine transformation, correlations are identical on raw and normalized scores.

5 Comparison with Related Work

Early model merging methods average checkpoints, as in FedAvg [37], Model Soups [62], and WiSE-FT [63], or add task vectors to the pretrained weights, as in Task Arithmetic [23], showing that independently fine-tuned models can often be combined without retraining. However, heterogeneous task updates interfere with each other [54, 53]. TIES [67] mitigates this by trimming small updates and resolving sign conflicts, whereas DARE [70] randomly drops and rescales weights. Consensus [59] keeps only weights relevant to multiple tasks, and LiNeS [60] scales the updates with layer depth.

A second family merges in aligned parameter subspaces: TSV [14], KnOTS [52], Core_{TSV} and Core_{ISO} [41], and Iso-C [34] use SVD, low-rank compression, whitening, or isotropic rescaling to align task updates. These methods are effective and often efficient, but their alignment remains defined in parameter space. CoM is complementary: it matches layer behavior directly on activations, updating them as the merged network is built.

A complementary line of merging-aware training instead biases training toward solutions that are mergeable by design. Tangent-space fine-tuning [40] linearizes the model around the pretrained weights, so that a linear combination of parameters corresponds to a linear combination of model outputs; τJp [69] promotes weight disentanglement by penalizing representation drift, and TAK [43] does so without access to other tasks' data, through a dataless regularizer built on a Kronecker-factorized approximation of the Generalized Gauss-Newton matrix. Unlike these approaches, CoM leaves training untouched and operates post hoc on independently fine-tuned models.

Feature- and activation-based methods are closest to our work: Neuron Alignment [55], ZipIt! [51], Optimal Transport Fusion [49], Fisher-weighted averaging [36], RegMean [25], LoRM [48], MaTS [53], ProDistill [66], and AdaMerging [68] use importance, activation matching, or data-driven optimization to improve compatibility. However, they typically estimate layer-wise statistics from the original task-specific networks, or optimize layer coefficients, without correcting the distribution shift induced by previously merged layers. CoM instead recomputes activation statistics sequentially, making layer-wise matching conditionally consistent.

Another line of work aligns and recalibrates networks before combining them. Permutation-based re-basin methods [2, 42], including TransFusion [46] for task vectors, align the units of independently trained models into the same loss basin, and REPAIR [26] corrects the internal activation statistics of the interpolated network, which collapse when two models' weights are averaged. These methods couple adjacent layers, as the alignment of a layer is constrained by its neighbors, but typically align one model at a time to a single target. The dependency modeled by CoM is instead inherently multi-task: once an earlier layer is merged, the inputs of the following layers change differently for each task, so statistics collected from the original task-specific networks become stale, and each merged layer must accommodate several shifted input distributions at once.

Finally, dynamic merging approaches adapt the merged weights to the task or input at inference time. EMR-Merging [20] applies lightweight task-specific masks and rescaling factors to a unified task vector, given the task identity; Twin-Merging [33] and WeMoE [54] instead use a router to combine task-specific modules or task vectors for each input. These methods gain accuracy at the price of additional storage, task information, or routing at inference. We instead follow the standard static inference protocol: a single merged backbone serves all tasks, without task identifiers or routing.

6 Discussion

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. CoM requires calibration examples (albeit only a few) to estimate activation statistics, and its gains are less pronounced with full fine-tuning on vision tasks, where merging is inherently easier and benchmark performance is nearing saturation.

Acknowledgments and Disclosure of Funding

We acknowledge the CINECA award under the ISCRA initiative, for the availability of high-performance computing resources and support. This work is supported by the Horizon Europe Chips Joint Undertaking under the NexTArc project (HORIZON-JU-Chips-2024-2-RIA). NexTArc – Next Generation Open Innovations in Trustworthy Embedded AI Architectures for Smart Cities, Mobility and Logistics (Grant Agreement ID: 101194287, DOI: 10.3030/101194287). Additionally, the research activities of Angelo Porrello have been partially supported by the Department of Engineering "Enzo Ferrari" through the program FAR2025DIP (CUP E93C25000370005). We also gratefully acknowledge Tetra Pak for funding the PhD position of Riccardo Salami.

References

[1] Armen Aghajanyan, Sonal Gupta, and Luke Zettlemoyer. Intrinsic dimensionality explains the effectiveness of language model fine-tuning. In Chengqing Zong, Fei Xia, Wenjie Li, and Roberto Navigli, editors, *Association for Computational Linguistics*, pages 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/`.

[2] Samuel Ainsworth, Jonathan Hayase, and Siddhartha Srinivasa. Git re-basin: Merging models modulo permutation symmetries. In *International Conference on Learning Representations*, 2023.

[3] Md Rifat Arefin, Gopeshh Subbaraj, Nicolas Gontier, Yann LeCun, Irina Rish, Ravid Shwartz-Ziv, and Christopher Pal. Seq-vcr: Preventing collapse in intermediate transformer representations for enhanced reasoning. *International Conference on Learning Representations*, 2024.

[4] Devansh Arpit, Yingbo Zhou, Bhargava Kota, and Venu Govindaraju. Normalization propagation: A parametric technique for removing internal covariate shift in deep networks. In *International Conference on Machine Learning*, pages 1168–1176. PMLR, 2016.

[5] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. Layer normalization. *arXiv preprint arXiv:1607.06450*, 2016.

[6] Federico Barbero, Andrea Banino, Steven Kapturowski, Dharshan Kumaran, Joao GM Araújo, Alex Vitvitskyi, Razvan Pascanu, and Petar Velickovic. Transformers need glasses! information over-squashing in language tasks. In *Advances in Neural Information Processing Systems*, 2024.

[7] Samuel R Bowman, Gabor Angeli, Christopher Potts, and Christopher D Manning. A large annotated corpus for learning natural language inference. In *Conference on Empirical Methods in Natural Language Processing*, 2015.

[8] Gong Cheng, Junwei Han, and Xiaoqiang Lu. Remote sensing image scene classification: Benchmark and state of the art. *Proceedings of the IEEE*, 105(10):1865–1883, 2017.

[9] Mircea Cimpoi, Subhransu Maji, Iasonas Kokkinos, Sammy Mohamed, and Andrea Vedaldi. Describing textures in the wild. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3606–3613, 2014.

[10] Michael Cogswell, Faruk Ahmed, Ross Girshick, Larry Zitnick, and Dhruv Batra. Reducing overfitting in deep networks by decorrelating representations. In *International Conference on Learning Representations*, 2016.

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

[12] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*, 2021.

[13] DC Dowson and BV Landau. The fréchet distance between multivariate normal distributions. *Journal of multivariate analysis*, 12(3):450–455, 1982.

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

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

[16] Yifei He, Yuzheng Hu, Yong Lin, Tong Zhang, and Han Zhao. Localize-and-stitch: Efficient model merging via sparse task arithmetic. *Transactions on Machine Learning Research*, 2024.

[17] Patrick Helber, Benjamin Bischke, Andreas Dengel, and Damian Borth. 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.

[18] Arthur E Hoerl and Robert W Kennard. Ridge regression: Biased estimation for nonorthogonal problems. *Technometrics*, 12(1):55–67, 1970.

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

[20] Chenyu Huang, Peng Ye, Tao Chen, Tong He, Xiangyu Yue, and Wanli Ouyang. Emr-merging: Tuning-free high-performance model merging. *Advances in Neural Information Processing Systems*, 37:122741–122769, 2024.

[21] You Huang and Yuanlong Yu. An internal covariate shift bounding algorithm for deep neural networks by unitizing layers' outputs. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8465–8473, 2020.

[22] Gabriel Ilharco, Mitchell Wortsman, Samir Yitzhak Gadre, Shuran Song, Hannaneh Hajishirzi, Simon Kornblith, Ali Farhadi, and Ludwig Schmidt. Patching open-vocabulary models by interpolating weights. *Advances in Neural Information Processing Systems*, 2022.

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

[24] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International Conference on Machine Learning*, pages 448–456. pmlr, 2015.

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

[26] Keller Jordan, Hanie Sedghi, Olga Saukh, Rahim Entezari, and Behnam Neyshabur. REPAIR: Renormalizing permuted activations for interpolation repair. In *International Conference on Learning Representations*, 2023.

[27] Dhiraj Kalamkar, Dheevatsa Mudigere, Naveen Mellempudi, Dipankar Das, Kunal Banerjee, Sasikanth Avancha, Dharma Teja Vooturi, Nataraj Jammalamadaka, Jianyu Huang, Hector Yuen, et al. A study of bfloat16 for deep learning training. *arXiv preprint arXiv:1905.12322*, 2019.

[28] Tushar Khot, Ashish Sabharwal, and Peter Clark. Scitail: A textual entailment dataset from science question answering. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, 2018.

[29] Jonathan Krause, Michael Stark, Jia Deng, and Li Fei-Fei. 3d object representations for fine-grained categorization. In *Proceedings of the IEEE international conference on computer vision workshops*, pages 554–561, 2013.

[30] Ananya Kumar, Aditi Raghunathan, Robbie Jones, Tengyu Ma, and Percy Liang. Fine-tuning can distort pretrained features and underperform out-of-distribution. *International Conference on Learning Representations*, 2022.

[31] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 2002.

[32] Sanwoo Lee, Jiahao Liu, Qifan Wang, Jingang Wang, Xunliang Cai, and Yunfang Wu. Dynamic fisher-weighted model merging via Bayesian optimization. In Luis Chiruzzo, Alan Ritter, and Lu Wang, editors, *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics*, pages 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/.

[33] Zhenyi Lu, Chenghao Fan, Wei Wei, Xiaoye Qu, Dangyang Chen, and Yu Cheng. Twin-merging: Dynamic integration of modular expertise in model merging. *Advances in Neural Information Processing Systems*, 37:78905–78935, 2024.

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

[35] Marco Marelli, Luisa Bentivogli, Marco Baroni, Raffaella Bernardi, Stefano Menini, and Roberto Zamparelli. 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)*, pages 1–8, 2014.

[36] Michael S Matena and Colin A Raffel. Merging models with fisher-weighted averaging. *Advances in Neural Information Processing Systems*, 2022.

[37] Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. Communication-efficient learning of deep networks from decentralized data. In *International Conference on Artificial Intelligence and Statistics*. PMLR, 2017.

[38] Ari S Morcos, David GT Barrett, Neil C Rabinowitz, and Matthew Botvinick. On the importance of single directions for generalization. In *International Conference on Learning Representations*, 2018.

[39] Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Baolin Wu, Andrew Y Ng, et al. Reading digits in natural images with unsupervised feature learning. In *NIPS workshop on deep learning and unsupervised feature learning*, volume 2011, page 7. Granada, 2011.

[40] Guillermo Ortiz-Jimenez, Alessandro Favero, and Pascal Frossard. Task arithmetic in the tangent space: Improved editing of pre-trained models. In *Advances in Neural Information Processing Systems*, pages 66727–66754, 2023.

[41] Aniello Panariello, Daniel Marczak, Simone Magistri, Angelo Porrello, Bartłomiej Twardowski, Andrew D Bagdanov, Simone Calderara, and Joost van de Weijer. Accurate and efficient low-rank model merging in core space. *Advances in Neural Information Processing Systems*, 2025.

[42] Fidel A Guerrero Peña, Heitor Rapela Medeiros, Thomas Dubail, Masih Aminbeidokhti, Eric Granger, and Marco Pedersoli. Re-basin via implicit Sinkhorn differentiation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2023.

[43] Angelo Porrello, Pietro Buzzega, Felix Dangel, Thomas Sommariva, Riccardo Salami, Lorenzo Bonicelli, and Simone Calderara. Dataless weight disentanglement in task arithmetic via kronecker-factored approximate curvature. In *International Conference on Learning Representations*, 2026.

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

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

[46] Filippo Rinaldi, Giacomo Capitani, Lorenzo Bonicelli, Donato Crisostomi, Federico Bolelli, Elisa Ficarra, Emanuele Rodolà, Simone Calderara, and Angelo Porrello. Update your transformer to the latest release: Re-basin of task vectors. In *International Conference on Machine Learning*, 2025.

[47] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 10684–10695, June 2022.

[48] Riccardo Salami, Pietro Buzzega, Matteo Mosconi, Jacopo Bonato, Luigi Sabetta, and Simone Calderara. Closed-form merging of parameter-efficient modules for federated continual learning. *International Conference on Learning Representations*, 2025.

[49] Sidak Pal Singh and Martin Jaggi. Model fusion via optimal transport. *Advances in Neural Information Processing Systems*, 33:22045–22055, 2020.

[50] Johannes Stallkamp, Marc Schlipsing, Jan Salmen, and Christian Igel. The german traffic sign recognition benchmark: a multi-class classification competition. In *The 2011 international joint conference on neural networks*, pages 1453–1460. IEEE, 2011.

[51] George Stoica, Daniel Bolya, Jakob Bjorner, Pratik Ramesh, Taylor Hearn, and Judy Hoffman. ZipIt! merging models from different tasks without training. In *International Conference on Learning Representations*, 2024.

[52] George Stoica, Pratik Ramesh, Boglarka Ecsedi, Leshem Choshen, and Judy Hoffman. Model merging with svd to tie the knots. *International Conference on Learning Representations*, 2025.

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

[54] Anke Tang, Li Shen, Yong Luo, Nan Yin, Lefei Zhang, and Dacheng Tao. Merging multi-task models via weight-ensembling mixture of experts. *International Conference on Machine Learning*, 2024.

[55] Norman Tatro, Pin-Yu Chen, Payel Das, Igor Melnyk, Prasanna Sattigeri, and Rongjie Lai. Optimizing mode connectivity via neuron alignment. *Advances in Neural Information Processing Systems*, 33:15300–15311, 2020.

[56] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. Llama: Open and efficient foundation language models. *arXiv:2302.13971*, 2023.

[57] Cédric Villani. *Optimal transport: old and new*, volume 338. Springer, 2008.

[58] Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. Glue: A multi-task benchmark and analysis platform for natural language understanding. In *International Conference on Learning Representations*, 2019.

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

[60] Ke Wang, Nikolaos Dimitriadis, Alessandro Favero, Guillermo Ortiz-Jimenez, Francois Fleuret, and Pascal Frossard. Lines: Post-training layer scaling prevents forgetting and enhances model merging. *International Conference on Learning Representations*, 2025.

[61] Adina Williams, Nikita Nangia, and Samuel R Bowman. A broad-coverage challenge corpus for sentence understanding through inference. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics*, 2018.

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

[63] Mitchell Wortsman, Gabriel Ilharco, Jong Wook Kim, Mike Li, Simon Kornblith, Rebecca Roelofs, Raphael Gontijo Lopes, Hannaneh Hajishirzi, Ali Farhadi, Hongseok Namkoong, et al. Robust fine-tuning of zero-shot models. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7959–7971, 2022.

[64] Xinyi Wu, Amir Ajorlou, Yifei Wang, Stefanie Jegelka, and Ali Jadbabaie. On the role of attention masks and layernorm in transformers. *Advances in Neural Information Processing Systems*, 37:14774–14809, 2024.

[65] Jianxiong Xiao, Krista A Ehinger, James Hays, Antonio Torralba, and Aude Oliva. Sun database: Exploring a large collection of scene categories. *International Journal of Computer Vision*, 119 (1):3–22, 2016.

[66] Jing Xu, Jiazheng Li, and Jingzhao Zhang. Scalable model merging with progressive layer-wise distillation. *International Conference on Machine Learning*, 2025.

[67] Prateek Yadav, Derek Tam, Leshem Choshen, Colin A Raffel, and Mohit Bansal. Ties-merging: Resolving interference when merging models. *Advances in Neural Information Processing Systems*, 36:7093–7115, 2023.

[68] Enneng Yang, Zhenyi Wang, Li Shen, Shiwei Liu, Guibing Guo, Xingwei Wang, and Dacheng Tao. Adamerging: Adaptive model merging for multi-task learning. *International Conference on Learning Representations*, 2024.

[69] Kotaro Yoshida, Yuji Naraki, Takafumi Horie, Ryosuke Yamaki, Ryotaro Shimizu, Yuki Saito, Julian McAuley, and Hiroki Naganuma. Mastering task arithmetic: τjp as a key indicator for weight disentanglement. In *International Conference on Learning Representations*, 2025.

[70] Le Yu, Bowen Yu, Haiyang Yu, Fei Huang, and Yongbin Li. Language models are super mario: Absorbing abilities from homologous models as a free lunch. In *International Conference on Machine Learning*, 2024.

[71] Fanhu Zeng, Haiyang Guo, Fei Zhu, Li Shen, and Hao Tang. RobustMerge: Parameter-efficient model merging for MLLMs with direction robustness. In *Advances in Neural Information Processing Systems*, 2025.

Appendix

The appendix is organized as follows:

- App. A collects the details of the method:
  - App. A.1 provides the closed-form derivations: the correlation-weighted merging solution (App. A.1.1) and the merging of normalization layers (App. A.1.2).
  - App. A.2 gives further implementation details: the activation normalization that improves the conditioning of the Gram matrices, and the complete algorithm.
  - App. A.3 compares the computational (FLOPs) and memory cost of CoM against existing methods.
- App. B collects further analyses of CoM:
  - App. B.1 complements the ablation of the main paper with a further analysis of the components of CoM (MCS correction, feature-correlation weighting, and activation normalization).
  - App. B.2 analyzes the task importance coefficients, covering their theoretical motivation (App. B.2.1) and an ablation on the reference model used to compute them (App. B.2.2).
  - App. B.3 reports supplementary ablations on the textual datasets, namely the correlation-based importance and the MCS analysis on Llama-3-8B.
  - App. B.4 discusses why the gains of CoM are larger with LoRA than with full fine-tuning.
- App. C collects the experiments on the dependency on data:
  - App. C.1 studies the sensitivity of CoM to the construction of the calibration set (sampling strategy, data split, input noise, class coverage, and the numerical values of the sample-efficiency plot).
  - App. C.2 evaluates calibration on synthetic images generated with Stable Diffusion v1.5, when no real task data is available.
  - App. C.3 compares merging with fine-tuning when the same examples are available.
- App. D collects additional results:
  - App. D.1 lists the accuracy on each dataset for ViT-B/32, ViT-L/14, and Llama-3-8B with LoRA fine-tuning.
  - App. D.2 reports full fine-tuning results on the language tasks.

A Method Details

A.1 Closed-form derivations

A.1.1 Derivation of the correlation-weighted merging solution

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

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

To simplify the notation, define $\boldsymbol{G}_i^l = \frac{1}{T_i} \tilde{\boldsymbol{X}}_i^l (\tilde{\boldsymbol{X}}_i^l)^\top$. Expanding the norm and applying standard rules of matrix differentiation, the gradient of the objective with respect to $\boldsymbol{W}_M^l$ is:

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

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.

A.1.2 Closed-form merging of normalization layers

Transformer blocks contain element-wise normalization layers (LayerNorm, RMSNorm) parameterized by a per-channel multiplicative scale γ and additive bias β. Since each output channel is processed independently, the activation-matching objective decouples into a collection of scalar problems – one per channel and per layer – and admits a simple closed-form solution: the merged parameters reduce to the arithmetic mean of the task-specific ones, independently of the underlying activation distribution. Derivation for the multiplicative scale below; the additive case is analogous.

Setup. Fix a single channel of a normalization layer and let $x \in \mathbb{R}$ denote its input scalar activation, after the standard normalization performed by the layer itself. Each task-specific model $i \in \{1, \ldots, N\}$ applies a learned scale γ_i, producing $\gamma_i\, x$. We seek the merged scale γ_m that minimizes the expected squared deviation from the average output of the individual models on this channel.

Multiplicative scale. The matching objective is

$$\arg\min_{\gamma_m} \; \mathbb{E}_x \left[\left(\gamma_m\, x - \frac{1}{N} \sum_{i=1}^{N} \gamma_i\, x \right)^2 \right].$$

Factoring out x and pulling the data-independent term outside the expectation gives

$$\arg\min_{\gamma_m} \; \left(\gamma_m - \frac{1}{N} \sum_{i=1}^{N} \gamma_i \right)^2 \mathbb{E}_x[x^2].$$

Assuming the channel is not trivially dead, $\mathbb{E}_x[x^2] > 0$, the global minimum is attained at

$$\gamma_m \;=\; \frac{1}{N} \sum_{i=1}^{N} \gamma_i,$$

independently of the underlying activation distribution.

Additive bias. An analogous derivation (in which x does not appear because the term enters additively) yields

$$\beta_m \;=\; \frac{1}{N}\sum_{i=1}^{N}\beta_i.$$

Implication. Merging normalization layers therefore requires neither activation statistics nor an iterative procedure: the optimal merged γ and β are simply the per-channel arithmetic means of the task-specific parameters. This formally justifies our use of plain averaging for all non-linear layers reported in the implementation details, without sacrificing optimality with respect to the activation-matching objective.

A.2 Further Implementation Details

Activation Normalization. Our proposed approach requires the inversion of $\sum_{i=1}^{N}\widehat{\boldsymbol{X}}_i^l\widehat{\boldsymbol{X}}_i^{l\top}$, 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 [6, 3], especially as fine-tuning typically occurs within a low-dimensional subspace [1, 30]. Drawing on prior work showing that layer normalization [5] mitigates representational collapse [64], we replace the Gram matrix $\widehat{\boldsymbol{X}}_i^l\widehat{\boldsymbol{X}}_i^{l\top}$ with the correlation matrix $\boldsymbol{G}_i^l$, which is computed from the per-token normalized features $\tilde{\boldsymbol{X}}_i^l$, averaged over the T_i tokens, and therefore better conditioned. The full ablation study in terms of average normalized accuracy comprising the effect of the activation normalization is in Tab. 4: normalizing the activations always benefits performance, especially on ViT-L/14, with an increase of over 3 points. The complete procedure of the proposed methodology is summarized in Alg. 1.

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. Values are averaged over three seeds. First row is RegMean with CoM's calibration set.

Components			Architecture		
MCS	Norm	FC	ViT-B/32	ViT-L/14	Llama-3-8B
✗	✗	✗	72.77	80.69	97.31
✗	✓	✗	74.32	83.05	97.52
✓	✗	✗	83.45	84.26	99.71
✗	✗	✓	82.79	85.66	95.02
✗	✓	✓	84.02	88.27	97.52
✓	✓	✗	85.83	87.38	99.76
✓	✗	✓	91.03	90.12	95.03
✓	✓	✓	92.74	91.35	99.63

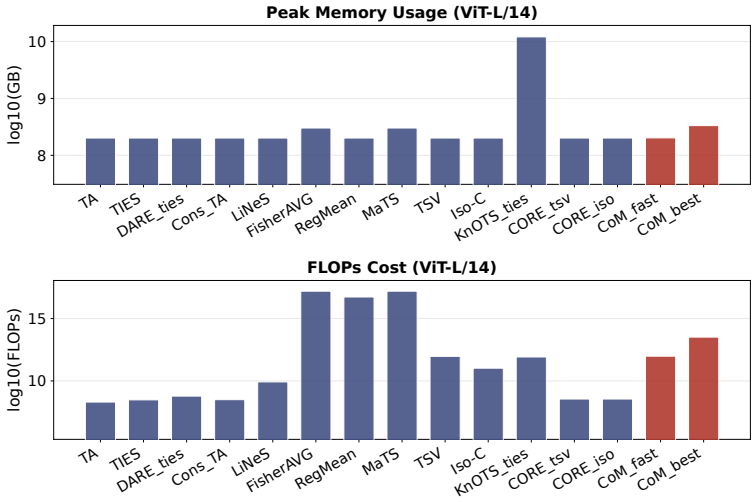
Algorithm 1 – Chain of Merges

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

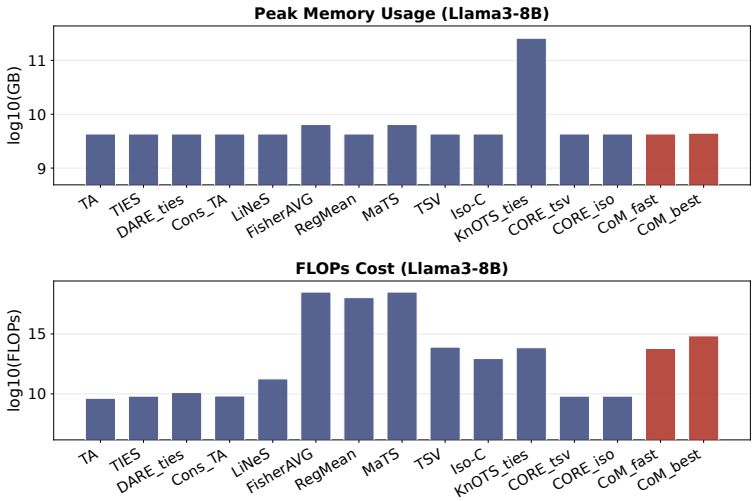
for $l = 1$ to L **do**
  for $i = 1$ to N **do**
    if $l = 1$ **then**
      $\widehat{\boldsymbol{X}}_i^1 \leftarrow \boldsymbol{X}_i^1$
    else
      $\widehat{\boldsymbol{X}}_i^l \leftarrow \boldsymbol{\sigma}(\boldsymbol{W}_M^{l-1}\widehat{\boldsymbol{X}}_i^{l-1})$
    end if
    $\tilde{\boldsymbol{X}}_i^l \leftarrow \widehat{\boldsymbol{X}}_i^l$ with each column (token) $\hat{\boldsymbol{x}}_t$ scaled to $\hat{\boldsymbol{x}}_t/\|\hat{\boldsymbol{x}}_t\|_2$
    $\boldsymbol{G}_i^l \leftarrow \frac{1}{T_i}\tilde{\boldsymbol{X}}_i^l\tilde{\boldsymbol{X}}_i^{l\top}$
    $\omega_i^l \leftarrow \sum_{p<q}|(\boldsymbol{G}_i^l)_{pq}|$
  end for
  Update $\boldsymbol{W}_M^l$ using Eq. 9
end for
return $\{\boldsymbol{W}_M^l\}_{l=1}^L$

A.3 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[3] Figs. 8a and 8b (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 a minimal calibration set of 5 samples per task, with which CoM is already on par with state-of-the-art methods whose hyperparameters are tuned on the whole validation set, as `CoM_fast`, and the variant using the optimal number of samples for maximum performance as `CoM_best`. The results show that CoM has a computational and memory cost comparable to alternative merging strategies.



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



(b) Peak memory usage and merging computational cost for Llama-3-8B.

Figure 8: Comparison of merging methods across computational and memory cost.

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 Fig. 3. These findings highlight that

[3]We refer the reader to Fig. 7 for ViT-B/32.

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.

B Further Analysis of CoM

B.1 Further Analysis of the Components of CoM

This section complements the ablation study of Sec. 4.3. In Tabs. 3 and 4, the rows with MCS = ✗ are layer-wise solutions computed on the activations of each task's own fine-tuned network, using the same class-balanced calibration samples as CoM; all values are averaged over three calibration seeds. Correcting MCS yields consistent gains over the layer-wise solution without chain on all three architectures, even without activation normalization ($+10.7$, $+3.6$ and $+2.4$ points on ViT-B/32, ViT-L/14 and Llama-3-8B; Tab. 4), whereas FC weighting is mostly beneficial on vision and can even hurt on language when feature normalization is off. Equipping the layer-wise solution with the activation normalization used by CoM brings comparatively small gains on all three architectures (Tab. 4), indicating that the improvement stems from the recursion rather than from normalization. The components are also complementary on vision: on ViT-L/14, for instance, the full model improves over MCS alone by about 4 points and over FC alone by about 3 points.

B.2 Analysis of Task Importance Coefficients

In this section, we provide a deeper analysis of CoM's task importance coefficients. 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.

B.2.1 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 general-purpose structures rather than task-specific patterns.

Consequently, high off-diagonal correlations in the input Gram matrix $\boldsymbol{G}$ after fine-tuning indicate that the task concentrates on a narrow subspace of the original data distribution, diverging significantly from the pretraining initialization. Intuitively, when a task heavily relies 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 [10, 38]. Our method leverages this principle: we treat significant deviations from the decorrelated pretrained state (high inter-feature correlation) as a signal of task specificity.

B.2.2 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.

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).

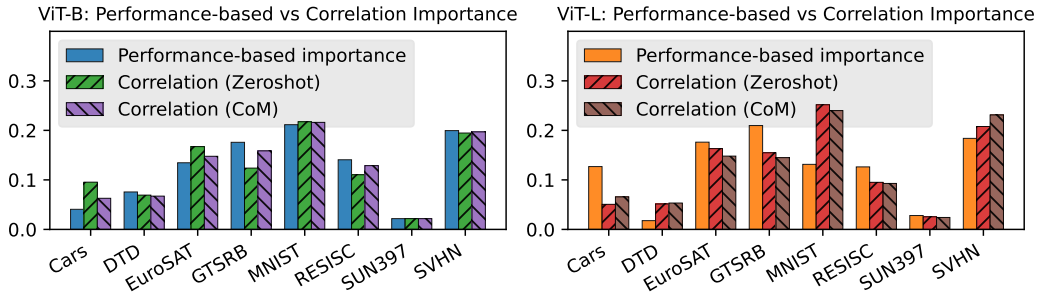


Figure 9: **Ablation of Task Importance Coefficients (ViT-B and ViT-L).** We compare the importance scores derived from our proposed CoM method (which uses the merged model as a reference) against those derived from the zero-shot pretrained model and a performance-based oracle. The results demonstrate that CoM coefficients strongly correlate with the zero-shot reference, justifying the use of the merged model as an efficient proxy.

The results, illustrated in Fig. 9 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.

B.3 Supplementary Ablations for Textual Datasets

Correlation-based importance. In Fig. 10, 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 Llama-3-8B 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. 11, we extend the analysis from Sec. 4.3 to the Llama-3-8B 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.

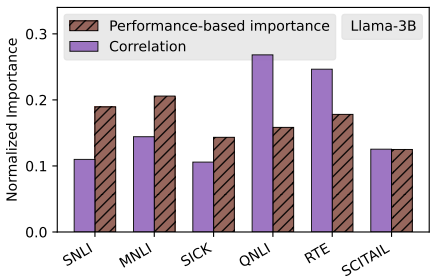


Figure 10: Task importance *vs.* correlation.

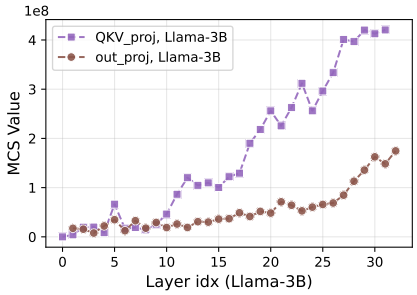


Figure 11: Merging Covariate Shift across layers.

B.4 Larger Gains with LoRA

Both our results and prior work [52, 71] suggest that LoRA checkpoints are intrinsically harder to merge than fully fine-tuned ones. In the full-rank weight space, task-specific solutions can distribute their updates across broader and more aligned subspaces, which reduces interference during merging and compresses the gap between methods. In contrast, LoRA constrains adaptation to a small number of directions, and parameter-efficient updates suffer from direction instability among singular values [71]: different tasks are therefore more likely to encode their knowledge within overlapping dominant directions of the pretrained model, which increases competition and task interference. When interference is more severe, as with LoRA, errors introduced in early layers propagate through a restricted adaptation space, and correcting MCS becomes particularly important. This explains why the largest gains of CoM emerge in the LoRA regime. In addition, on ViT-L/14 with full fine-tuning the benchmark is essentially saturated (Iso-C: 98.2% *vs.* CoM: 97.6% normalized accuracy), leaving little room for significant gains.

C Data Dependency

C.1 Sensitivity to the Calibration Set

We analyze the dependency of CoM on the construction of the calibration set (500 samples per task; by default, from the training split on Llama-3-8B, whose RTE and SICK validation splits are smaller) by comparing: (i) class-balanced *vs.* unconstrained random sampling, (ii) samples drawn from the validation *vs.* the training split, (iii) calibration inputs perturbed with Gaussian noise of standard deviation σ_{noise} added to the preprocessed images, and (iv) calibration restricted to at most 3% of the classes of each task (a single class for the 10-class datasets). Tab. 5 reports the average normalized accuracy over three seeds. Noise is added to the input embeddings with standard deviation σ_{noise} times their root-mean-square.

Table 5: Average normalized accuracy (%) of CoM (LoRA) under different calibration set constructions. Mean $\pm$ standard deviation over 3 seeds.

Calibration set	ViT-B/32	ViT-L/14	Llama-3-8B
Class-balanced, validation split (default, vision)	92.74 ± 0.14	91.35 ± 0.02	99.57 ± 0.15
Random, validation split	92.60 ± 0.06	91.31 ± 0.03	99.88 ± 0.12
Class-balanced, training split (default, Llama)	92.53 ± 0.05	91.30 ± 0.03	99.76 ± 0.12
Random, training split	92.39 ± 0.06	91.25 ± 0.01	99.69 ± 0.10
Noise $\sigma_{\text{noise}} = 0.01$	90.64 ± 0.13	85.52 ± 0.14	99.72 ± 0.11
Noise $\sigma_{\text{noise}} = 0.1$	82.05 ± 0.13	81.61 ± 0.23	99.70 ± 0.12
Noise $\sigma_{\text{noise}} = 1.0$	69.10 ± 0.63	67.44 ± 2.25	99.19 ± 0.07
At most 3% of the classes	79.23 ± 0.58	84.75 ± 0.61	98.36 ± 0.42

On all architectures, the sampling strategy has a limited impact: using the training instead of the validation split, or random instead of class-balanced sampling, changes the accuracy by at most 0.4 points on ViT-B/32, 0.1 points on ViT-L/14 and 0.2 points on Llama-3-8B. Performance degrades progressively as the calibration inputs are corrupted with noise, and the largest drops occur under severe distributional mismatch, i.e., strong noise or calibration on very few classes. On Llama-3-8B, noise perturbation does not substantially impact performance.

Number of calibration examples. Tab. 6 reports the numerical values of Fig. 3 for all the sample sizes of CoM's calibration set, from 5 to 500 examples per task.

Table 6: Average normalized accuracy of CoM (LoRA) varying the number of examples per task: numerical values of Fig. 3.

# of samples	5	10	50	100	200	300	400	500
ViT-B/32	76.94	81.95	88.74	91.02	92.24	92.57	92.67	92.74
ViT-L/14	86.12	88.43	90.85	91.24	91.34	91.32	91.36	91.35
Llama-3-8B	97.17	97.59	98.97	99.43	99.71	99.48	99.66	99.76

C.2 Synthetic Calibration Data

In this section, we evaluate merging methods in a setting where no real task data is available, but samples can be generated. For each vision task, we generate 500 images with Stable Diffusion v1.5 [47], one class-conditioned prompt per image, at 512×512 and downsampled to the native resolution of the dataset. The generated images replace the real data: RegMean and CoM are calibrated on all synthetic images and use their default hyperparameters, while the hyperparameters of the data-free methods are selected on the same synthetic images instead of the real validation split. Tab. 7 reports the results on the LoRA checkpoints (single run). CoM remains the best method on both backbones, ahead of the strongest baseline (Core_{ISO}) by 7.4 points on ViT-B/32 and 4.1 points on ViT-L/14, although the margin is smaller than with real data (Tab. 1).

Table 7: Average normalized accuracy (%) of merged LoRA models using synthetic data (500 generated images per task) in place of the real calibration and validation data.

Arch.	TA	TIES	$\text{DARE}_{\text{TIES}}$	Iso-C	TSV	Core_{ISO}	RegMean	CoM
ViT-B/32	62.91	62.27	62.21	64.73	63.63	<u>66.64</u>	62.95	**74.01**
ViT-L/14	72.14	73.25	71.97	77.52	73.18	<u>77.53</u>	75.24	**81.59**

C.3 Merging *vs.* Fine-Tuning with the Same Data

Since CoM needs a few examples per task for merging, one may ask whether the same examples would be better spent on tuning the hyperparameters of a data-free method or on fine-tuning a merged model. We address this question with a matched-data protocol on the LoRA checkpoints of ViT-B/32 and ViT-L/14 and, since repeatedly fine-tuning Llama-3-8B is too expensive, on Llama-3.2-1B fine-tuned on the six language tasks with LoRA adapters (rank 16 on the attention projections). For $n \in \{10, 50, 100, 500\}$, we draw n examples per task uniformly at random from the split used for calibration in the rest of the paper (validation for vision, training for language; not class-balanced, hence the slight differences in the CoM results w.r.t. Tabs. 1 and 6), and every method receives exactly the same examples.

CoM and RegMean are calibrated on all n examples with their default hyperparameters and use no labels; RegMean is therefore run on n examples here, rather than on the entire training set as in its original setting. The hyperparameters of all other methods are tuned on the same n labeled examples (indicated as *Merged* in Tab. 8).

We then fine-tune the backbone of every merged model on the same examples (+ *Fine-tuning*): 80% of the examples are used for the gradient updates and the remaining 20% for early stopping (patience 5, at most 1000 steps) and for selecting the learning rate in $\{10^{-7}, \dots, 10^{-4}\}$. As a reference without merging, the same fine-tuning is also applied to the pretrained backbone (*Zero-shot*). Not fine-tuning is also a candidate: the fine-tuned model is retained only if it improves over the merged model on the early-stopping examples, and the merged model is reported otherwise. We use AdamW, keep the classification heads frozen (on Llama-3.2-1B, all transformer blocks are fine-tuned and the embeddings are frozen as well), and evaluate the test split once, for the selected model. Results are averaged over three seeds that fix the sampled examples.

Two regimes emerge from Tab. 8. With few examples per task, fine-tuning is unreliable, to the point that it is often not even preferred to the merged model on the held-out examples. In this regime, merging is the most effective way to exploit the available data, and CoM, which does not need labels, yields the strongest model. As the number of labeled examples grows, fine-tuning becomes effective and all methods converge to similar accuracies.

On Llama-3.2-1B, the CoM merge alone is on par with the best fine-tuned models even with 500 examples per task. At that point, the question is no longer which merging method to use, but whether merging is needed at all, as there is enough labeled data to train on all tasks jointly; even then, starting from a merged model remains preferable to fine-tuning the pretrained one. Merging is thus most valuable when little or no labeled data is available, where CoM makes the best use of it.

Finally, this comparison only matches the amount of data, and does not account for the much higher computational cost and memory footprint of fine-tuning: while CoM requires a single forward pass per model, fine-tuning requires backpropagation through the whole backbone, optimizer states, and repeated runs to select the learning rate.

Table 8: Average normalized accuracy (%) of LoRA merged models when all methods use the same n examples per task (mean over 3 seeds).

	Merged				+ Fine-tuning			
Method	$n=10$	50	100	500	$n=10$	50	100	500
ViT-B/32								
Zero-shot	60.6	60.6	60.6	60.6	63.8	73.4	78.4	92.5
Task Arithmetic	63.6	63.2	63.9	63.9	66.9	75.3	80.1	93.6
TIES	63.6	63.9	63.8	64.0	67.3	75.6	80.3	93.4
DARE-TIES	64.1	63.9	64.1	64.1	67.8	75.8	80.4	93.6
LiNeS	63.6	63.2	63.9	63.9	66.9	75.3	80.1	93.6
Consensus	64.2	65.1	65.1	65.3	65.8	77.3	81.3	92.8
TSV	66.8	66.5	66.5	66.5	72.2	77.3	82.1	93.6
KnOTS-TIES	67.5	68.1	68.1	68.3	72.0	78.9	82.0	94.3
Iso-C	69.9	73.2	73.2	73.1	69.2	81.1	84.3	94.4
Core_{TSV}	66.6	66.6	66.6	66.6	70.9	78.2	81.7	93.7
Core_{ISO}	76.0	75.9	75.6	76.1	76.0	82.3	84.7	**95.1**
RegMean	72.4	73.3	72.9	72.9	75.3	81.6	84.0	**95.1**
CoM	**81.2**	**88.7**	**90.9**	**92.6**	**81.2**	**87.1**	**90.9**	94.8
ViT-L/14								
Zero-shot	71.4	71.4	71.4	71.4	72.0	81.0	84.0	92.4
Task Arithmetic	74.1	74.4	74.4	74.8	76.3	83.6	86.1	93.4
TIES	74.6	75.2	75.0	75.5	77.3	84.2	86.5	93.7
DARE-TIES	74.1	74.5	74.5	74.9	75.4	83.6	85.4	93.5
LiNeS	73.9	74.4	74.4	74.8	75.3	83.6	86.0	93.5
Consensus	72.4	74.4	74.4	74.8	76.7	83.6	85.8	92.9
TSV	76.0	77.8	77.8	77.8	77.8	86.6	88.3	94.6
KnOTS-TIES	75.6	78.3	79.1	79.1	78.8	86.8	88.6	94.4
Iso-C	78.2	84.3	84.7	84.7	78.4	88.6	90.0	94.2
Core_{TSV}	76.7	77.2	77.7	78.2	78.0	87.5	88.6	93.9
Core_{ISO}	86.0	85.6	85.6	86.1	81.9	**91.3**	**92.9**	95.9
RegMean	81.3	81.0	81.0	80.7	83.5	90.4	92.2	**96.2**
CoM	**87.9**	**90.6**	**91.0**	**91.3**	**87.9**	90.3	91.9	95.6
Llama-3.2-1B								
Zero-shot	–	–	–	–	51.6	65.1	72.0	89.3
Task Arithmetic	70.3	81.1	81.1	81.1	69.5	85.0	87.5	94.7
TIES	73.7	80.7	80.9	81.5	75.7	77.2	86.6	93.7
DARE-TIES	80.8	80.3	81.4	81.2	75.8	84.5	87.8	96.0
LiNeS	71.7	81.1	81.1	81.1	70.9	85.0	87.6	94.4
Consensus	71.5	71.2	79.5	79.7	70.5	80.8	88.3	95.3
TSV	74.9	87.5	87.5	87.5	74.0	87.4	90.5	95.3
KnOTS-TIES	66.7	82.1	81.2	82.4	66.8	86.6	88.4	95.2
Iso-C	53.4	53.4	53.5	72.9	52.0	66.7	75.3	94.8
Core_{TSV}	75.5	88.0	89.5	89.5	73.9	90.0	89.8	96.2
Core_{ISO}	54.6	64.2	83.3	83.2	54.7	80.4	94.2	94.6
RegMean	88.8	92.5	93.2	93.3	88.8	92.8	93.5	96.3
CoM	**90.9**	**95.3**	**96.4**	**96.3**	**90.9**	**95.3**	**95.2**	**96.4**

D Additional Results

D.1 Accuracies on each dataset

Following [52, 41], we report the normalized accuracies on each dataset in Tab. 9 for ViT-B/32, in Tab. 10 for ViT-L/14 and in Tab. 11 for Llama-3-8B with LoRA fine-tuning.

D.2 Full Fine-Tuning on Language Tasks

For memory reasons, fully fine-tuning and merging the 8B model was not feasible in our setup, so we use a smaller Llama (Llama-3.2-1B) instead of Llama-3-8B. We fully fine-tune all parameters of the model on each of the six tasks with AdamW, a cosine schedule and an effective batch size of 32; for each task, the learning rate is chosen among $\{5\times10^{-6}, 10^{-5}, 2\times10^{-5}\}$ and the number of epochs (at most three) by early stopping on the validation split. As for the other experiments, the hyperparameters of all baselines are tuned on the validation split, RegMean is used in its original setting, and CoM uses 500 class-balanced samples per task. Results are reported in Tabs. 12 and 13. CoM remains the

Table 9: 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	73.99	49.01	41.93	53.45	71.57	97.47	42.08	63.93
TIES	81.57	72.90	49.20	37.87	57.40	70.05	96.64	46.87	64.06
$\text{DARE}_{\text{TIES}}$	82.70	72.81	49.49	37.68	57.73	69.87	96.87	45.92	64.13
Consensus TA	82.72	73.08	48.75	40.79	60.51	71.46	96.95	47.78	65.26
LiNeS	81.59	74.91	50.84	39.88	52.81	71.07	97.99	39.27	63.54
FisherAVG	<u>84.91</u>	73.99	<u>81.74</u>	53.19	51.15	69.58	96.42	34.05	68.13
RegMean	77.20	64.14	36.25	50.47	<u>97.47</u>	77.03	88.99	<u>94.99</u>	73.32
MaTS	84.47	73.72	80.10	52.00	53.10	70.15	96.77	36.36	68.33
TSV	83.12	74.91	49.12	44.85	61.30	73.10	95.74	52.22	66.80
Iso-C	79.07	82.84	52.45	75.62	68.93	79.11	**98.70**	50.49	73.40
$\text{KnOTS}_{\text{TIES}}$	82.87	73.54	48.11	48.67	71.44	70.87	95.52	55.51	68.32
Core_{TSV}	83.77	75.45	52.49	45.21	59.30	72.99	97.71	46.17	66.64
Core_{ISO}	83.29	<u>84.30</u>	48.67	<u>82.43</u>	73.02	<u>83.28</u>	<u>98.65</u>	55.37	<u>76.13</u>
CoM (ours)	**90.58**	**87.80**	**89.99**	**91.99**	**99.42**	**94.54**	89.52	**98.11**	**92.74**

Table 10: 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	80.07	79.73	65.51	60.01	82.27	79.52	86.68	64.95	74.84
TIES	79.54	78.21	64.27	61.18	84.16	79.48	87.36	70.19	75.55
$\text{DARE}_{\text{TIES}}$	79.95	78.89	63.22	60.11	81.66	79.91	87.50	68.77	75.00
Consensus TA	81.88	81.32	68.60	64.55	85.04	81.56	86.90	63.72	76.69
LiNeS	79.70	79.73	66.04	57.88	81.86	78.24	86.48	64.10	74.25
FisherAVG	82.60	81.09	66.68	<u>87.66</u>	75.32	79.70	85.44	61.41	77.49
RegMean	77.56	78.59	65.74	80.93	<u>94.25</u>	90.86	88.16	<u>89.05</u>	83.14
MaTS	82.08	81.17	66.79	85.65	75.80	79.96	85.55	62.60	77.45
TSV	82.58	80.11	64.69	69.31	83.01	83.68	87.91	70.91	77.77
Iso-C	87.86	<u>89.14</u>	<u>81.03</u>	81.36	90.24	89.40	<u>88.77</u>	69.78	84.70
$\text{KnOTS}_{\text{TIES}}$	82.53	80.41	64.58	68.81	88.87	82.32	88.26	76.78	79.07
Core_{TSV}	82.00	79.88	67.02	66.81	87.78	82.24	87.63	72.04	78.17
Core_{ISO}	**91.37**	**90.89**	79.90	86.02	88.50	**91.33**	**89.77**	70.87	<u>86.08</u>
CoM (ours)	<u>88.97</u>	88.51	**87.82**	**93.26**	**99.38**	<u>91.14</u>	83.87	**97.85**	**91.35**

Table 11: Normalized accuracies of merged models on language with Llama-3-8B. Best results are highlighted in bold, second-best underlined.

Method	SNLI	MNLI	SICK	QNLI	RTE	SCITAIL	**Average**
TA	93.66	95.11	88.14	68.66	99.19	96.83	90.27
TIES	95.07	96.32	83.46	70.70	100.00	96.35	90.32
$\text{DARE}_{\text{TIES}}$	94.84	96.40	81.62	71.66	99.19	96.07	89.96
Consensus TA	94.26	94.63	88.78	66.43	<u>101.61</u>	96.93	90.44
LiNeS	92.87	95.35	89.14	73.24	97.58	96.64	90.80
RegMean	**99.91**	**98.65**	82.02	<u>98.55</u>	94.35	95.03	<u>94.75</u>
TSV	94.67	95.71	86.07	78.54	100.81	97.13	92.15
Iso-C	84.52	51.50	93.14	53.94	66.94	75.05	70.85
$\text{KnOTS}_{\text{TIES}}$	89.48	93.64	<u>93.92</u>	81.16	<u>101.61</u>	97.71	92.92
Core_{TSV}	95.91	95.71	89.32	83.92	**102.42**	<u>97.86</u>	94.19
Core_{ISO}	90.93	91.58	85.33	77.11	100.00	97.51	90.41
CoM (ours)	<u>99.60</u>	<u>97.64</u>	**100.28**	**99.98**	101.34	**99.72**	**99.76**

best-performing method by a large margin, outperforming the strongest competitor, Iso-C, by 4.1 points of normalized accuracy; RegMean, whose raw-token Gram matrices are dominated by the largest tasks, collapses on SICK. Core_{TSV}, Core_{ISO} and KnOTS are omitted, as they are designed for LoRA updates and are therefore not applicable to full fine-tuning.

Table 12: Absolute and normalized average accuracy (%) of merged fully fine-tuned Llama-3.2-1B models on the six language tasks. CoM: mean over three calibration seeds ($\pm$ std); DARE-TIES: mean over three drop masks; other rows are deterministic.

Method	Abs. Acc.	Norm. Acc.
Zero-shot	45.47	50.86
Individual FT	89.56	100.00
Task Arithmetic	77.64	86.85
TIES	77.31	86.39
DARE-TIES	77.13	86.29
Consensus	77.96	87.24
LiNeS	77.61	86.81
RegMean	79.49	88.82
TSV	79.38	88.70
Iso-C	<u>82.21</u>	<u>91.96</u>
CoM	**85.99** $\pm$ 0.28	**96.09** $\pm$ 0.34

Table 13: Per-task normalized test accuracy (%) of merged fully fine-tuned Llama-3.2-1B models.

Method	SNLI	MNLI	SICK	QNLI	RTE	SciTail	Avg
Zero-shot	35.32	36.18	60.42	51.73	59.09	62.41	50.86
Task Arithmetic	94.72	94.27	79.70	69.54	92.73	90.15	86.85
TIES	98.62	99.02	68.05	75.02	89.09	88.51	86.39
DARE-TIES	96.17	95.83	74.60	68.09	93.03	90.04	86.29
Consensus	93.07	92.02	85.38	68.90	93.64	90.45	87.24
LiNeS	95.53	95.51	76.87	69.83	92.73	90.40	86.81
RegMean	100.23	100.23	63.44	87.78	93.64	87.62	88.82
TSV	98.54	98.33	73.17	79.67	91.82	90.65	88.70
Iso-C	92.40	93.77	92.76	81.44	98.18	93.24	91.96
CoM	98.40	97.18	92.40	97.50	99.09	91.99	**96.09**

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and Sec. 1 explicitly state our three contributions – (i) the identification and analysis of Merging Covariate Shift (MCS), (ii) the Chain of Merges (CoM) procedure that updates activation statistics autoregressively, and (iii) the correlation-based importance weighting – all of which are formally developed in Sec. 2–3 and empirically validated in Sec. 4.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: Limitations are discussed in Sec. 6: CoM requires a few calibration examples, the evaluation relies on standard model-merging benchmarks, and the gains are smaller with full fine-tuning. How critical the data dependency is in practice is analyzed in Sec. 4.2 (number of calibration samples) and in the appendix (construction of the calibration set, App. C.1; synthetic calibration data, App. C.2; merging *vs.* fine-tuning with the same data, App. C.3).

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best

judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

   Justification: The closed-form solution for the correlation-weighted merging objective (Eq. 9) is fully derived in App. A.1.1, and the closed-form merging of element-wise normalization layers is derived in App. A.1.2. All assumptions (e.g., non-degenerate activations, invertibility of the weighted Gram sum) are stated in the corresponding derivations.

   Guidelines:

   - The answer [N/A] means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.
   - The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
   - Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
   - Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: Sec. 4.1 (Implementation Details) specifies the backbones (ViT-B/32, ViT-L/14, Llama-3-8B), the source of fine-tuned checkpoints (LoRA from [52], full fine-tuning from [23]), the hyperparameters (500 calibration samples per task, Tikhonov coefficient 0.95 for language), and the numerical-stability choices (bfloat16 with float32 Gram inversion, Moore–Penrose pseudoinverse). Algorithm 1 fully specifies CoM, and code is publicly available.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.

(b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
(c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
(d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: An anonymized code repository with instructions to reproduce all main results is provided in the supplementary material, as stated in the abstract. All datasets used (Cars, DTD, EuroSAT, GTSRB, MNIST, RESISC45, SUN397, SVHN, SNLI, MultiNLI, SICK, SciTail, RTE, QNLI) are publicly available, and we rely on publicly released checkpoints from [52, 23].

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: Sec. 4.1 reports the architectures, fine-tuning protocols (LoRA with rank $r = 16$ on attention modules, and full fine-tuning), the calibration set construction (balanced examples across tasks/classes), and all hyperparameters of CoM (number of samples and Tikhonov coefficient). Per-dataset accuracies are provided in App. D.1.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.

- The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: CoM results are averaged over three random draws of the calibration set; the corresponding standard deviations are reported in Tab. 5 (rows of the default configurations, at most 0.14 points), and mean $\pm$ standard deviation over three seeds is also reported for the calibration-set, full fine-tuning and merging-*vs.*-fine-tuning analyses in the appendix. Baselines are deterministic (they either use no data or use the entire training set) and are run once, following the standard protocol of the model merging literature (e.g., [52, 23, 41]).

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).
   - It should be clear whether the error bar is the standard deviation or the standard error of the mean.
   - It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
   - For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
   - If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: Sec. 5 (Single-forward implementation, Complexity) and App. A.3 provide a detailed analysis of FLOPs and peak memory usage for ViT-B/32, ViT-L/14, and Llama-3-8B (Figs. 4 and 7). All experiments fit on a single consumer GPU since merging requires only one forward pass per task model and N single-layer activation buffers.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

Answer: [Yes]

Justification: The research uses publicly available datasets and pretrained checkpoints, does not involve human subjects or sensitive data, and follows the NeurIPS Code of Ethics in every respect. Anonymity is preserved in the submission.

Guidelines:

- The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
- If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
- The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

Answer: [N/A]

Justification: The paper proposes a foundational methodology for combining independently fine-tuned model checkpoints into a single backbone. It does not introduce new capabilities or data and has no direct path to specific positive or negative societal applications beyond those already associated with the underlying pretrained models.

Guidelines:

- The answer [N/A] means that there is no societal impact of the work performed.
- If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

Answer: [N/A]

Justification: The paper does not release new pretrained models, generative systems, or scraped datasets. Only an algorithm and the corresponding implementation are released, operating on already publicly available checkpoints.

Guidelines:

- The answer [N/A] means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: All datasets (Cars, DTD, EuroSAT, GTSRB, MNIST, RESISC45, SUN397, SVHN, SNLI, MultiNLI, SICK, SciTail, RTE, QNLI), backbones (CLIP ViT-B/32, ViT-L/14, Llama-3-8B), and the source LoRA / full fine-tuning checkpoints are cited to their original works in Sec. 4.1, and used in compliance with their licenses.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
    - If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: We release an anonymized implementation of CoM in the supplementary material and code to reproduce the main tables.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

Answer: [N/A]

Justification: The paper does not involve crowdsourcing or research with human subjects.

Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

Answer: [N/A]

Justification: The paper does not involve crowdsourcing or research with human subjects, so IRB approval is not applicable.

Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

Answer: [N/A]

Justification: LLMs are used only as one of the target architectures being merged (Llama-3-8B), not as an original or non-standard component of the methodology. They were not employed beyond standard writing or formatting assistance.

Guidelines:

- The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.