MOMA: MASKED ORTHOGONAL MATRIX ALIGNMENT FOR ZERO-ADDITIONAL-PARAMETER MODEL MERGING

Anonymous authors
Paper under double-blind review

ABSTRACT

Model merging offers a scalable alternative to multi-task learning but often suffers from substantial performance drops on classification tasks. We attribute this degradation to a geometric misalignment between the merged encoder and static task-specific classifier heads. Existing post-merging methods rely on auxiliary parameters to enforce strict representation alignment, incurring storage and inference overhead that grows with the number of merged models. We challenge this approach by revealing that the misalignment is well-approximated by an orthogonal transformation, rendering such strict alignment unnecessary. Leveraging this insight, we propose MOMA (Masked Orthogonal Matrix Alignment), a plug-and-play post-merging framework that rectifies the misalignment by jointly optimizing a global multi-task vector mask and task-specific orthogonal transformations. Crucially, MOMA absorbs corresponding new parameters directly into existing model weights, achieving state-of-the-art performance among post-merging methods with zero additional parameters and zero added inference cost. Our code is released in Github `https://anonymous.4open.science/r/MOMA-B7A1`.

1 INTRODUCTION

Model merging aims to directly combine multiple homologous fine-tuned models in the parameter space, bypassing the need for raw training data or retraining, to synthesize a unified model capable of solving multiple tasks Izmailov et al. (2018); Jin et al. (2022); Ilharco et al. (2022); Yang et al. (2024b); Yadav et al. (2023). While model merging has shown success in many areas such as continual learning Marczak et al. (2024); Tang et al. (2025) and information retrieval Li et al. (2024); Braga et al. (2025), we observe that it often yields suboptimal results in classification tasks.

We attribute this performance degradation to a structural misalignment between the merged encoder and the static task-specific classifier heads. Existing efforts to address merging difficulties operate at three distinct stages of the pipeline, yet none adequately resolve this misalignment. *Pre-merging* methods Ortiz-Jimenez et al. (2023); Yoshida et al. (2025) restrict fine-tuning to the tangent space around the base model, motivated by the Neural Tangent Kernel (NTK) Jacot et al. (2018); Chizat et al. (2019) view of merging. However, the non-linear gap arising in practical fine-tuning cannot be fully eliminated, leaving the merged encoder geometrically displaced from the fine-tuned encoders and thus misaligned with the static classifier heads. *During-merging* methods Yadav et al. (2023); Yu et al. (2024); Gargiulo et al. (2025); Marczak et al. (2025) mitigate inter-task interference during aggregation, but their objective is confined to producing a high-quality merged backbone in parameter space, leaving the alignment between the merged encoder and the static classifier heads unaddressed. *Post-merging* methods Yang et al. (2024a); Wei et al. (2025) explicitly target this misalignment by introducing auxiliary parameters that align merged and fine-tuned representations. While effective, this architectural expansion incurs extra storage and inference cost Wortsman et al. (2022), which grows with the number of merged models. In this work, we find that the misalignment can be well approximated by an orthogonal transformation, a simple transformation that can be formulated without introducing additional parameters.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

To validate this finding, we conduct geometric analyses, revealing that the discrepancy between the merged and fine-tuned representations can be characterized by an orthogonal transformation. We substantiate this by demonstrating, via visualization and KNN evaluation, that task-relevant information within the merged representations remains intact. Subsequently, by probing with learnable transformations under varying constraints, we confirm that mitigating the discrepancy via an orthogonal transformation is sufficient to restore the predictive capabilities of the fine-tuned models. This hypothesis is also corroborated by quantitative metric analysis, where high cosine similarity and coefficients between the orthogonally aligned and fine-tuned representations verify the orthogonality.

Building on this insight, we propose a post-merging alignment framework termed MOMA (Masked Orthogonal Matrix Alignment). Distinct from prior post-merging methods that enforce strict representation alignment via auxiliary modules, MOMA prioritizes output logit consistency by rectifying the inherent orthogonal misalignment between the merged encoder and the task-specific classifier heads. Specifically, we jointly optimize task-specific orthogonal transformations and a global multi-task vector mask. The orthogonal transformation rotates the merged representations to align with the classifier heads, while the mask selectively filters redundant parameters to relax optimization constraints and facilitate convergence. Crucially, we exploit the inner-product-preserving property of orthogonal transformations to absorb the matrices directly into the classifier head weights, and permanently apply the learned mask to the multi-task vector. This design ensures that MOMA achieves substantial performance gains with zero additional parameters and zero added inference cost.

Our contributions are summarized as follows:

- We characterize the misalignment between merged encoders and static classifier heads as well-approximated by an orthogonal transformation, revealing that the strict representation alignment used in existing post-merging methods is an unnecessary over-constraint.
- We propose MOMA, a plug-and-play post-merging alignment framework compatible with existing during-merging methods. MOMA rectifies orthogonal misalignment by jointly optimizing a multi-task vector mask and task-specific orthogonal transformations, with the resulting parameters absorbed into existing weights for zero deployment overhead.
- Extensive experiments demonstrate that MOMA, with zero additional parameters and zero added inference cost, achieves state-of-the-art performance among post-merging methods and bridges the gap to fine-tuned models.

2 Background and Related Works

2.1 Notation and Problem Setup

Consider a c-class classification task. Let $\mathcal{X} \subseteq \mathbb{R}^d$ denote the instance space and $\mathcal{Y} \subseteq \mathbb{R}^c$ the label space. We consider a model $f : \mathcal{X} \to \mathcal{Y}$ (e.g., a neural network) that decomposes into a feature encoder $g : \mathcal{X} \to \mathbb{R}^k$, parameterized by θ, and a classifier head $h : \mathbb{R}^k \to \mathcal{Y}$, parameterized by ϕ. Thus, $f = h \circ g$, or equivalently, $f(x) = h\left(g(x;\theta);\phi\right)$.

Given a pre-trained (base) model with parameters (θ_b, ϕ_b), we fine-tune it on T downstream tasks. Each task $t \in \{1, \ldots, T\}$ is associated with a dataset $\mathcal{D}_t = \{(x_i^t, y_i^t)\}_{i=1}^{|\mathcal{D}_t|}$, obtaining task-specific parameters $\{(\theta_t, \phi_t)\}_{t=1}^T$. In the model-merging setting, especially when tasks have heterogeneous label spaces, standard approaches merge the fine-tuned encoders $\{\theta_t\}_{t=1}^T$ into a shared backbone θ_m, while keeping the task-specific heads $\{\phi_t\}_{t=1}^T$ fixed. The goal is to construct θ_m solely from the fine-tuned weights, without accessing labeled data, such that the merged model $h\big(g(x;\theta_m);\phi_t\big)$ performs well on every task t.

2.2 Task Arithmetic and Merging Objective

Task Arithmetic Ilharco et al. (2022) defines a *task vector* as the difference between the fine-tuned encoder parameters θ_t and the base initialization θ_b, namely $\tau_t = \theta_t - \theta_b$. The merged encoder is then formed by a linear combination in parameter space,

$$\theta_m = \theta_b + \lambda \sum_{t=1}^{T} \tau_t, \tag{1}$$

where λ is a scaling hyperparameter. Although this merging is performed in parameter space, the desired behavior is specified in the model's output (logit) space Ortiz-Jimenez et al. (2023). Ideally, when θ_m is paired with the task-specific head ϕ_t, it should reproduce the predictions of the fine-tuned model, i.e.

$$h(g(x;\theta_m);\phi_t) \approx h(g(x;\theta_t);\phi_t), \quad \forall x \in \mathcal{D}_t. \tag{2}$$

This objective indicates the central challenge of model merging: additive updates to the encoder parameters must induce compatible representations so that fixed, task-specific classifier heads continue to produce correct logits.

2.3 Pre-Merging: Constraining Fine-Tuning for Linearity

Pre-merging methods address task interference at its theoretical root: rather than correcting it after it arises, they shape fine-tuning so that task vectors are inherently composable in parameter space. The theoretical motivation comes from the Neural Tangent Kernel (NTK) Jacot et al. (2018) view of merging. In the infinite-width limit, gradient-based training falls into the so-called *lazy regime* Chizat et al. (2019), where the network can be well approximated by its first-order Taylor expansion around the initialization θ_b,

$$g_{\text{lin}}(x;\theta) \triangleq g(x;\theta_b) + (\theta-\theta_b)^\top \nabla_{\theta_b} g(x;\theta_b), \quad g_{\text{lin}}(x;\theta_m) = g(x;\theta_b) + \lambda \sum_{t=1}^{T} \tau_t^\top \nabla_{\theta_b} g(x;\theta_b). \tag{3}$$

Under this linearization, parameter-space addition translates directly into function-space addition, providing a clean justification for additive merging. Motivated by this view, pre-merging methods explicitly restrict fine-tuning to the *tangent space* around θ_b Ortiz-Jimenez et al. (2023); Yoshida et al. (2025); Porrello et al. (2026). However, practical fine-tuning typically operates in the *feature-learning regime* Chizat et al. (2019); Geiger et al. (2020); Woodworth et al. (2020), where parameters move far from initialization and the linear approximation can be substantially violated Yehudai & Shamir (2019); Ghorbani et al. (2019); Karp et al. (2021). Such constraints may therefore leave a residual non-linear gap, causing the merged representation $g(x;\theta_m)$ to be misaligned with the fixed task-specific heads ϕ_t optimized for the fine-tuned representations.

2.4 During-Merging: Mitigating Task Interference in Parameter Space

During-merging methods predominantly focus on mitigating inter-task interference during the aggregation process, with a smaller line of work optimizing per-task or per-layer merging coefficients Qiu et al. (2025); Zhang et al. (2025); Dai et al. (2025); Yang et al. (2024b). Among the most interference-mitigation methods, Ties Yadav et al. (2023) resolves sign conflicts and prunes redundancy; DARE Yu et al. (2024) employs random dropping to reduce interference; TALL-masks Wang et al. (2024) and Model Tailor Zhu et al. (2024) learn masks to localize task-relevant information Miyano & Arase (2025); Zimmer et al. (2023); Marczak et al. (2024). More recently, training-free approaches based on subspace orthogonalization reduce inter-task conflict: TSV-M Gargiulo et al. (2025) reduces task interference by applying a whitening transformation that decorrelates singular vectors across tasks, while Iso-CTS Marczak et al. (2025) further isolates common and task-specific components for isotropic merging. Notably, these methods are predominantly evaluated on classification benchmarks, where each task is paired with a fixed classifier head ϕ_t. This is a structural blind spot during-merging methods are not designed to address: even optimal aggregation cannot guarantee geometric alignment between the merged encoder and the static heads.

2.5 Post-Merging: Aligning the Merged Model in Representation Space

Post-merging methods serve as a plug-in stage that further bridges the representation-space misalignment in the merged model Sun et al. (2025); Shao et al. (2026); Yang et al. (2024a); Wei et al. (2025). Representative methods such as Surgery Yang et al. (2024a) and ProbSurgery Wei et al. (2025) introduce additional trainable parameters ψ to explicitly minimize the distance between merged and fine-tuned representations:

$$\min_{\psi} \mathbb{E}_{x\sim\mathcal{D}_t} \| R(g(x;\theta_m);\psi) - g(x;\theta_t) \|_2, \tag{4}$$

where $R(\cdot;\psi)$ is a learnable function applied to the merged representations. While effective in reducing the representation gap, this reliance on auxiliary modules incurs extra storage and inference cost.

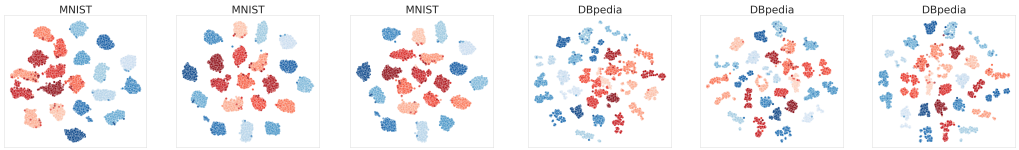
In fact, such heavy-handed structural alignment may be unnecessary. Since predictions are governed by the composite map $h \circ g$, the merged encoder need not replicate the exact fine-tuned coordinates $g(x; \theta_t)$; it only needs to produce representations that remain compatible with the fixed head. In particular, the discrepancy between $g(x; \theta_m)$ and $g(x; \theta_t)$ is, as shown in Section 3, well-approximated by a near-orthogonal transformation. Such a geometry-preserving correction is structurally simple, and can be addressed through lightweight realignment rather than by forcing the encoder back to the fine-tuned representation space with auxiliary representation-mapping parameters.

3 Geometric Analysis of Merging Discrepancy

This section investigates the structure of the merging-induced discrepancy between $g(x; \theta_m)$ and $g(x; \theta_t)$. We hypothesize that this discrepancy can be characterized by a geometry-preserving transformation, in which case the auxiliary parameters used in post-merging methods Yang et al. (2024a); Wei et al. (2025) would be unnecessary. Our qualitative and quantitative results consistently support this hypothesis, showing that the discrepancy is well-captured by an orthogonal transformation.

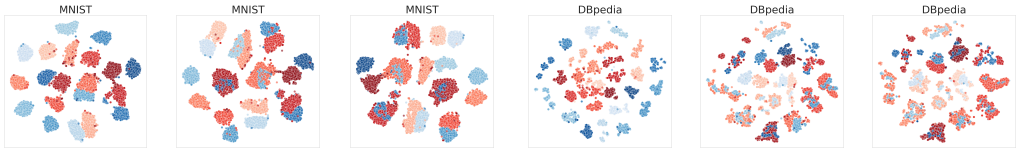
3.1 Visualization of Merged Representations

We use t-SNE to visualize the representation space of the fine-tuned representations $g(\cdot; \theta_t)$ and the merged representations $g(\cdot; \theta_m)$ for several merging methods. To more clearly assess class-conditional separation, we incorporate the ground-truth label information into the visualization process, as in Figure 1. More visualization in Appendix A.



(a) WA (87.5%) (b) TA (97.3%) (c) Ties (96.5%) (d) WA (55.1%) (e) TA (81.5%) (f) Ties (83.4%)

Figure 1: T-SNE visualization of representations from Weight Averaging (WA) Wortsman et al. (2022), Task Arithmetic (TA), and Ties Yadav et al. (2023). Blue points denote fine-tuned representations, while red points denote merged representations, with color shades indicating different class labels and values in parentheses denoting classification accuracy. Despite significant differences in classification accuracy (e.g., WA vs. Ties on DBpedia), all merged representations maintain distinct class-conditional clusters. This evidence suggests that discriminative information is preserved, i.e., the representation intrinsic to class distinction has not collapsed in the merged model.



(a) WA (95.3%) (b) TA (98.2%) (c) Ties (97.7%) (d) WA (97.4%) (e) TA (97.6%) (f) Ties (98.3%)

Figure 2: T-SNE visualization after orthogonal alignment. Compared to the unaligned case, the classification accuracy shows a dramatic recovery.

The visualization reveals a distinct decoupling between classification accuracy and representational structure. As shown in Figure 1, even merged models with suboptimal performance exhibit well-separated clusters that mirror the geometry of the fine-tuned targets. This disconnect implies that the essential task-discriminative information remains intact after merging. Therefore, we hypothesize that the observed performance drops do not arise from a loss of intrinsic classification capability, but rather from a geometric drift where these otherwise healthy clusters have shifted away from the optimal regions defined by the static classifier heads ϕ_t.

3.2 KNN Evaluation of Merged Representations

To empirically verify whether the merged representations retain sufficient task-relevant information, despite the performance drop observed with the static classifier, we employ a *K-Nearest Neighbors*

(KNN) evaluation on $g(\cdot;\theta_m)$. Unlike the static classifier ϕ_t, which requires the merged representations to be aligned with a fixed weight matrix, KNN is a non-parametric method that relies solely on the local geometry (Euclidean distances) of the representation space.

Specifically, for each task t, we construct a support set by sampling K labeled examples per class from the dataset. A test sample x is classified by assigning it to the class that minimizes the average distance to these support samples. Formally, let $\{x_i^{t,c}\}_{i=1}^K$ denote the set of K support samples for class c in task t. The prediction is formulated as:

$$d_c(x) = \frac{1}{K}\sum_{i=1}^{K} \|g(x;\theta_m) - g(x_i^{t,c};\theta_m)\|_2, \quad \hat{y} = \arg\min_{c\in\mathcal{Y}} d_c(x). \tag{5}$$

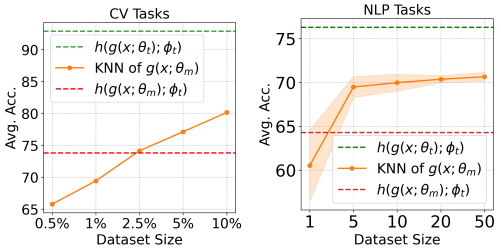


Figure 3: Performance comparison of merged representations using KNN. Interestingly, KNN evaluation (orange line) significantly outperforms the direct inference baseline (red dashed line) with minimal support samples (e.g., 5-shot for NLP). This empirical evidence validates that the static classifier head significantly hinders the model's potential, rendering it inferior to a direct non-parametric evaluation of the representations.

We compare the upper-bound fine-tuned model ($h(g(x;\theta_t);\phi_t)$), the TA merged model ($h(g(x;\theta_m);\phi_t)$), and the KNN evaluation of the merged representations. The x-axis represents the number of support samples K (or the dataset ratio for CV tasks with large class counts). Figure 3 shows that KNN recovers classification accuracy. This finding is pivotal: since a simple nearest-neighbor probe can retrieve the correct labels, it confirms that the merged encoder $g(\cdot;\theta_m)$ preserves the task-specific discriminative information. Thus, the performance degradation is not due to information loss in the representations, but is primarily attributable to the geometric misalignment between the encoder and the static classifier head.

3.3 Hypothesis Validation via Transformation Alignment

The success of the KNN evaluation suggests that the merged representations and the fine-tuned representations share the same intrinsic geometry, yet differ in their spatial alignment. To characterize this misalignment, we hypothesize that there exists a specific transformation mapping the former to the latter. We validate this by formulating an alignment objective with varying degrees of constraints on the transformation function.

$G(\mathbf{x}) = \mathbf{A}\mathbf{x} + \mathbf{b}$, where $\mathbf{A} \in \mathbb{R}^{n\times n}, \mathbf{b} \in \mathbb{R}^n$	
Transformation	Constraints
Affine Transform	None
Linear Transform	$\mathbf{b} = \mathbf{0}$
Invertible Linear Transform	$\det(\mathbf{A}) \neq 0, \mathbf{b} = \mathbf{0}$
Similarity Transform	$\mathbf{A}^\top\mathbf{A} = s\mathbf{I}\ (s > 0), \mathbf{b} = \mathbf{0}$
Orthogonal Transform	$\mathbf{A}^\top\mathbf{A} = \mathbf{I}, \mathbf{b} = \mathbf{0}$
Scaling Transform	$\mathbf{A}_{ij} = 0\ (\forall i \neq j), \mathbf{b} = \mathbf{0}$

Table 1: Definitions of different transformation classes used in our analysis. Here, $\mathbf{x}$ denotes the merged representation and $G(\mathbf{x})$ denotes the target fine-tuned representation. The constraints impose different types of geometric regularization.

Formally, we align the merged representations $g(\cdot;\theta_m)$ with the fine-tuned representations $g(\cdot;\theta_t)$ by optimizing:

$$\min_{G_t} \mathbb{E}_{x\sim\mathcal{D}_t} \|G_t(g(x;\theta_m)) - g(x;\theta_t)\|_2, \tag{6}$$

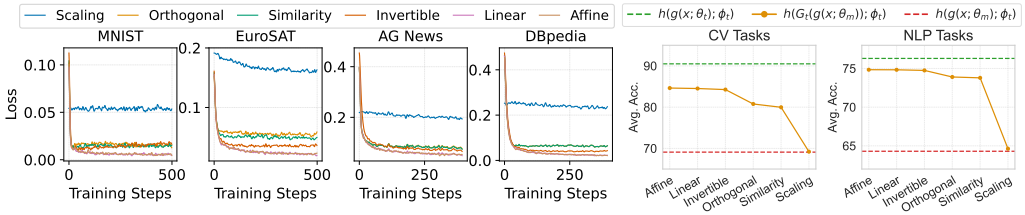
where $G_t(\cdot)$ denotes the transformation function for task t, instantiated according to the families listed in Table 1. We fit G_t on unlabeled training data and evaluate it on disjoint test data, with θ_m and θ_t frozen. This setup serves strictly as an analytical probe to determine the nature of the misalignment.

Taking the Orthogonal Transformation as a primary case study, we solve for the optimal rotation matrix $\mathbf{A}_t$ by minimizing the alignment error subject to an orthogonality regularization:

$$\min_{\mathbf{A}_t} \sum_{x\in\mathcal{D}_t} \|\mathbf{A}_t g(x;\theta_m) - g(x;\theta_t)\|_2 + \alpha\|\mathbf{A}_t^\top\mathbf{A}_t - \mathbf{I}\|_2, \tag{7}$$

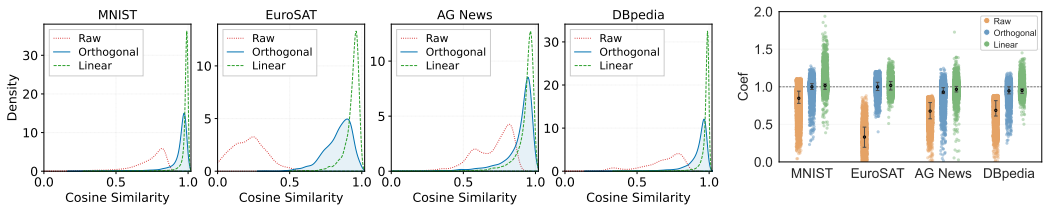
where α is a hyperparameter balancing the alignment fidelity and the orthogonality condition. By tuning α, we enforce $\mathbf{A}_t$ to converge towards a rigid rotation.

As shown in Figure 4 and corroborated by the cluster recovery in Figure 2, an orthogonal transformation is sufficient to effectively rectify the representation drift between merged and fine-tuned representations. Since orthogonal mappings are rigid transformations that preserve the intrinsic geometric structure, this finding supports our hypothesis that the merged representations retain necessary task-relevant information. Consequently, enforcing stricter representational equivalence (via Linear or Affine constraints) or introducing additional parameters and structures offers diminishing returns and is unnecessary for performance recovery.



(a) Training loss via different transformations. (b) Acc. via different transformations.

Figure 4: Analysis of alignment training loss and recovered performance under varying transformation constraints. The transition to an orthogonal transformation yields the most significant performance leap, recovering a substantial portion of the accuracy. While further relaxing the constraints (to Invertible, Linear, or Affine) offers marginal additional gains, these non-rigid deformations provide negligible benefits relative to the increased degrees of freedom. This suggests that the misalignment is largely rotational, identifying orthogonal alignment as the optimal trade-off between performance recovery and geometric structural preservation.



(a) Kernel density estimation of cosine similarity distributions. (b) Distribution of coefficients.

Figure 5: Quantitative Verification of Orthogonality. The legends Raw, Orthogonal, and Linear represent the alignment results between merged and fine-tuned representations under different constraints. (a) The orthogonal alignment drastically shifts the distribution towards 1.0, matching the unconstrained Linear alignment. (b) The orthogonal alignment naturally recovers the regression coefficients to approximately 1.0, indicating that the magnitude scale is inherently preserved. Collectively, these results demonstrate that the orthogonal transformation significantly reduces the geometric discrepancy between the merging and fine-tuned representations.

3.4 Quantitative Verification of Orthogonality

To quantitatively verify our hypothesis that the discrepancy between the merged representations and the fine-tuned representations is nearly an orthogonal transformation, we follow Zhou et al. (2024) and measure cosine similarity for directional alignment and the coefficient for magnitude consistency. We compute these metrics between the fine-tuned representations $g(x;\theta_t)$ and merged representations $g(x;\theta_m)$ under three distinct alignment conditions: (i) *Raw*, representing the original merged representations; (ii) *Orthogonal*, aligned via the optimal orthogonal transformation derived in Eq. 7; and (iii) *Linear*, aligned via the flexible unconstrained linear transformation. Formally, for a pair of representation vectors u and v, these metrics are defined as $\cos(u,v) = u^\top v/(\|u\|\|v\|)$ and $\text{coef}(u,v) = u^\top v/\|v\|^2$.

As visualized in Figure 5, the gap between merged and fine-tuned representations can be almost rectified by a simple rotation. This empirical evidence supports our hypothesis: the representational drift is primarily a rigid geometric transformation, leaving the intrinsic feature structure intact.

4 MASKED ORTHOGONAL MATRIX ALIGNMENT

Building on the orthogonal characterization established in Section 3, we now formalize MOMA. The key observation is that the linearity of the classifier heads allows the orthogonal transformation to be absorbed directly into the classifier weights, avoiding any architectural expansion.

Specifically, let the classifier head for task t be denoted as $h(\cdot;\phi_t)$, parameterized by weights $\mathbf{W}_t$, such that the logit calculation is governed by the inner product between the representation and the weights. Applying an orthogonal transformation $\mathbf{A}_t$ to the merged representations $g(x;\theta_m)$ modifies this operation to $\langle \mathbf{A}_t g(x;\theta_m), \mathbf{W}_t \rangle$. We constrain $\mathbf{A}_t$ to be orthogonal because our diagnostic analysis shows that a geometry-preserving transformation is sufficient to correct most of the misalignment without unnecessarily distorting the representation space. By the linearity of the classifier head, the aligned logits can be equivalently written as

$$\langle \mathbf{A}_t g(x;\theta_m), \mathbf{W}_t \rangle = \langle g(x;\theta_m), \mathbf{A}_t^\top \mathbf{W}_t \rangle. \tag{8}$$

Consequently, we can replace the original weights $\mathbf{W}_t$ with the aligned weights $\mathbf{A}_t^\top \mathbf{W}_t$. This operation achieves the desired alignment without introducing any additional parameters.

While orthogonal alignment is theoretically sound, enforcing such a rigid geometric constraint can be optimizationally challenging and prone to underfitting. Prior studies Wang et al. (2024) have demonstrated that aggregated multi-task vectors can be effectively pruned without compromising representation capability. Motivated by this, we introduce a pruning mechanism to the aggregated multi-task vector. By selectively masking the parameters, we increase the degrees of freedom within the optimization landscape, thereby facilitating the convergence of the orthogonal transformation.

We define the aggregated multi-task vector as $\tau_{\text{agg}} = \sum_{t=1}^{T} \tau_t$, and learn a mask $\mathbf{M}$, yielding the pruned task vector $\tau_m = \mathbf{M} \odot \tau_{\text{agg}}$. To enable gradient-based optimization for the binary mask, we employ a sigmoid reparameterization strategy. Specifically, we optimize a real-valued parameter $\mathbf{S}$ and derive the mask via $\mathbf{M} = \sigma(\mathbf{S})$. The joint optimization objective is formulated as:

$$\min_{\{\mathbf{A}_t:\mathbf{A}_t^\top \mathbf{A}_t=\mathbf{I}\}_{t=1}^{T},\mathbf{S}} \sum_{t=1}^{T} \mathbb{E}_{x\sim\mathcal{D}_t} \|\mathbf{A}_t g(x;\theta_b+\lambda\sigma(\mathbf{S})\odot\tau_{\text{agg}}) - g(x;\theta_t)\|_2, \tag{9}$$

where $\odot$ denotes the Hadamard product. Finally, the learned mask is absorbed in τ_m, and the orthogonal matrices $\mathbf{A}_t$ are absorbed into respective task classifier heads $\mathbf{W}_t$ via matrix multiplication. Thus, our method incurs zero additional parameter storage overhead at deployment.

We also explore a simpler variant, termed Aligned-C, that mitigates the misalignment by directly fine-tuning the classifier heads on few-shot labeled data; see Appendix E for details.

5 EXPERIMENTS

5.1 EXPERIMENTAL SETTING

Datasets Following representative post-merging methods Yang et al. (2024a); Wei et al. (2025), we evaluate on eight image and five text classification datasets. Details are in Appendix B.1.

Comparable Baselines We mainly evaluate MOMA integrated into Task Arithmetic, Ties, LW AdaMerging Yang et al. (2024b), and TSV-M Gargiulo et al. (2025), benchmarking against alignment strategies like Surgery Yang et al. (2024a) and ProbSurgery Wei et al. (2025). See Appendix B.2 for details.

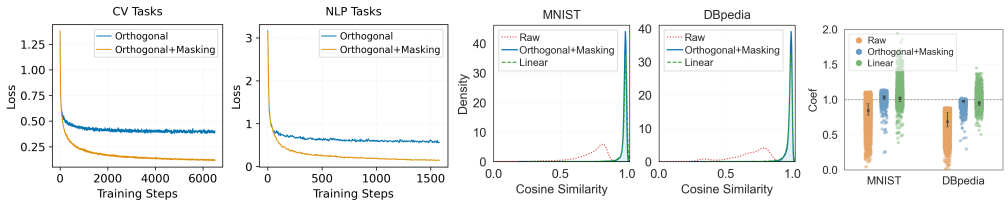
Implementation Details We use ViT-B/16, ViT-B/32, and ViT-L/14 Radford et al. (2021) for CV and bert-base-uncased Devlin et al. (2018) for NLP, adopting fine-tuned checkpoints from Yang et al. (2024a). All experiments are conducted for three random seeds on NVIDIA A100 GPUs. See Appendix B.3 for more details.

5.2 MAIN RESULTS

Main results are shown in Table 2. MOMA surpasses strong competitors such as Surgery and ProbSurgery to achieve SOTA accuracy among all post-merging methods, validating that orthogonal

Method	SUN397	Cars	RESISC45	EuroSAT	SVHN	GTSRB	MNIST	DTD	Avg.
Pre-trained Model	62.3	59.7	60.7	45.5	31.4	32.6	48.5	43.8	48.0
Fine-tuned Model	79.2	77.6	96.1	99.7	97.4	98.7	99.6	79.4	91.0
Traditional MTL	73.9	74.4	93.9	98.2	95.8	98.9	99.5	77.9	88.9
Fisher Merging (Matena & Raffel, 2022)	68.6	69.2	70.7	66.4	72.9	51.1	87.9	59.9	68.3
RegMean (Jin et al., 2022)	65.3	63.5	75.6	78.6	78.1	67.4	93.7	52.0	71.8
Concrete TA (Tang et al., 2023)	62.5	61.1	76.0	95.7	91.0	81.9	98.5	51.9	77.3
Concrete AM (Tang et al., 2023)	67.8	70.0	87.5	96.0	91.6	96.7	98.7	63.8	84.0
Task Arithmetic (Ilharco et al., 2022)	55.2	54.9	66.7	78.9	80.2	69.7	97.3	50.4	69.1
w/ Surgery (Yang et al., 2024a)	63.8	59.9	83.3	97.9	87.0	87.0	98.6	69.4	80.9
w/ ProbSurgery (Wei et al., 2025)	67.0	67.0	94.1	99.8	91.2	98.8	99.4	79.0	87.0
w/ MOMA	72.7	70.0	93.5	99.4	96.7	96.6	99.5	77.3	**88.2**
Ties (Yadav et al., 2023)	65.0	64.4	74.8	77.4	81.2	69.3	96.5	54.5	72.9
w/ Surgery	69.8	66.1	87.3	97.5	86.7	87.6	98.5	71.6	83.1
w/ ProbSurgery	71.5	70.6	94.4	99.7	90.6	98.9	99.4	78.9	88.0
w/ MOMA	76.2	74.8	95.2	99.6	97.4	98.7	99.6	78.7	**90.0**
LW AdaMerging Yang et al. (2024b)	64.5	68.1	79.2	93.8	87.0	91.9	97.5	59.1	80.1
w/ Surgery	71.2	72.0	92.3	99.0	92.2	97.9	99.0	76.1	87.5
w/ ProbSurgery	71.7	73.1	94.8	99.7	93.6	98.8	99.5	79.3	88.8
w/ MOMA	78.4	76.9	95.7	99.7	97.5	98.8	99.7	78.9	**90.7**
TSV-M Gargiulo et al. (2025)	69.0	70.7	85.4	94.2	92.0	91.9	99.2	69.1	83.9
w/ Surgery	71.0	71.8	91.0	99.0	93.8	96.2	99.4	75.5	87.2
w/ ProbSurgery	73.7	74.8	93.5	99.4	93.6	97.7	99.4	78.8	88.9
w/ MOMA	75.0	74.4	94.9	99.6	97.3	98.7	99.6	78.9	**89.8**

Table 2: Comparison on CV datasets via ViT-B/32. MOMA achieves state-of-the-art (SOTA) performance, particularly with Task Arithmetic and Ties. Results on ViT-B/16 and ViT-L/14, and results on NLP tasks are shown in Appendix C.



(a) Training loss of MOMA.

(b) Geometric analysis of the learned representations.

Figure 6: Analysis of MOMA's training and learned representations. (a) Training loss comparison between Orthogonal alone and Orthogonal+Masking (MOMA). (b) Cosine similarity (left) and regression coefficient (right) distributions under Raw, MOMA, and Linear alignment.

alignment is effective for post adaptation. Additionally, MOMA delivers improvements across all four representative during-merging baselines, confirming its flexibility as a plug-and-play module.

5.3 Analysis of MOMA's Training and Representations

To better understand the mechanism behind MOMA's performance, we analyze the training loss and the geometric properties of the aligned representations.

Convergence Behavior Figure 6 (a) visualizes the alignment loss during training. Compared to using Orthogonal alone, the full MOMA converges to a significantly lower loss value on both CV and NLP tasks. This indicates that by masking non-essential weights, MOMA effectively relaxes the optimization landscape, granting the transformation the necessary degrees of freedom to achieve a much tighter and more accurate fit.

Geometry of Aligned Representations Figure 6 (b) further evaluates the geometry of the aligned representations. As shown in Figure 6 (b) left, the cosine similarity distribution of MOMA matches that of the Linear alignment (the empirical upper bound) almost exactly. Similarly, Figure 6 (b) (right) demonstrates that MOMA recovers the regression coefficients to unity, correcting the scale shift observed in the *Raw* baseline. These results validate that MOMA attains the high geometric precision of non-rigid transformations (Linear) while strictly adhering to the parameter-efficiency and structural preservation of rigid orthogonal mappings.

5.4 Further Analysis of MOMA

To better understand MOMA, we present an ablation study, analyze the effect of the masking rate, and compare it with Aligned-C. Additional experiments on the regularization coefficient α, scaling to larger task counts, and LLM backbones are deferred to Appendix D.

Method	CV (Avg.)	NLP (Avg.)
Task Arithmetic	69.1	64.3
w/ Orthogonal	76.4	72.3
w/ Masking	83.5	74.9
w/ Orthogonal + Masking	88.2	76.3
w/ Linear + Masking	89.9	76.5

Table 3: Ablation study of MOMA and constraint analysis. Notably, "Orthogonal" and "Linear" represent the geometric transformation constraints applied to the alignment, while "Orthogonal + Masking" constitutes our proposed MOMA framework.

Ablation Study Table 3 details the contribution of each component. Orthogonal alignment improves the CV average from 69.1% to 76.4%, while Masking alone raises it to 83.5%. Combining the two further improves performance to 88.2% on CV and 76.3% on NLP. These results suggest that Masking facilitates rigid orthogonal alignment by filtering redundant task-vector components, enabling a tighter fit. We do not claim that orthogonal alignment is necessary or uniquely amenable to parameter absorption; rather, it is sufficient to explain and correct most of the discrepancy while preserving representation geometry. Linear + Masking therefore serves as a relaxed empirical upper bound and yields only modest additional gains.

Masking Rate Figure 7 illustrates the test accuracy via the masking rate. MOMA exhibits remarkable robustness across a wide range of sparsity levels (10%–90%) on both CV and NLP tasks. The performance remains stable and consistently matches the fine-tuned upper bound, only degrading under extreme sparsity ($> 90\%$). This observation aligns with the findings in DARE Yu et al. (2024), which demonstrated that removing up to 90% of delta parameters results in minimal performance degradation.

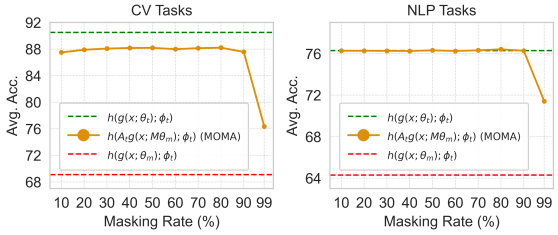


Figure 7: Sensitivity analysis of the masking rate.

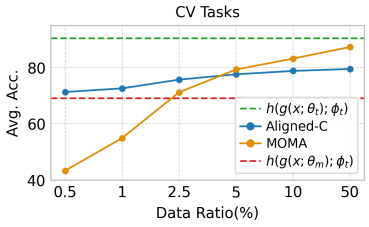


Figure 8: Comparison of MOMA and Aligned-C across different data ratios.

MOMA vs. Aligned-C We further compare Aligned-C and MOMA, which require different forms of supervision (see Figure 8). Aligned-C updates only the classifier head using labeled few-shot examples, making it stable in extremely low-resource settings but limiting its performance ceiling. In contrast, MOMA learns the mask and orthogonal transformations from unlabeled calibration samples. It is less reliable with very few samples, but improves steadily as more unlabeled data are available. From the Table, we observe that Aligned-C is preferable when only a few labeled examples are available, while MOMA becomes more suitable when sufficient unlabeled calibration data can be collected.

6 Conclusion

This paper presents MOMA, a plug-and-play post-merging alignment framework that addresses the geometric misalignment between the merged encoder and the static task-specific classifier heads. By identifying that this misalignment can be well-approximated by an orthogonal transformation, we rectify the drift via learnable orthogonal matrices jointly with a multi-task vector mask. Leveraging the inner-product-preserving property of orthogonal transformations, MOMA absorbs these parameters directly into existing model weights at deployment, preserving the original architecture. Extensive experiments demonstrate that MOMA achieves state-of-the-art performance among post-merging methods with zero additional parameters and zero added inference cost.

AI USE STATEMENT

In this work, we used generative AI tools for manuscript polishing. All AI-assisted edits were manually reviewed and revised by the authors to ensure technical accuracy, consistency with the intended meaning, and faithful presentation of the methods and results. We take full responsibility for the final content of this work, including any text, claims, or artifacts produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

To facilitate reproducibility, we provide our implementation in an anonymous code repository. Experimental settings and implementation details are documented in Section 5 and the Appendix.

REFERENCES

Yelp dataset challenge. `http://www.yelp.com/dataset_challenge`, 2014.

Marco Braga, Pranav Kasela, Alessandro Raganato, and Gabriella Pasi. Investigating task arithmetic for zero-shot information retrieval. In *Proceedings of the 48th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 2738–2743, 2025.

Gong Cheng, Junwei Han, and Xiaoqiang Lu. Remote sensing image scene classification: Benchmark and state of the art. *Proceedings of the IEEE*, pp. 1865–1883, Oct 2017.

Lenaic Chizat, Edouard Oyallon, and Francis Bach. On lazy training in differentiable programming. *Advances in neural information processing systems*, 32, 2019.

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*, pp. 3606–3613, 2014.

Rui Dai, Sile Hu, Xu Shen, Yonggang Zhang, Xinmei Tian, and Jieping Ye. Leveraging submodule linearity enhances task arithmetic performance in llms. *arXiv preprint arXiv:2504.10902*, 2025.

Li Deng. The mnist database of handwritten digit images for machine learning research [best of the web]. *IEEE signal processing magazine*, 29(6):141–142, 2012.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

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

Mario Geiger, Stefano Spigler, Arthur Jacot, and Matthieu Wyart. Disentangling feature and lazy training in deep neural networks. *Journal of Statistical Mechanics: Theory and Experiment*, 2020 (11):113301, 2020.

Behrooz Ghorbani, Song Mei, Theodor Misiakiewicz, and Andrea Montanari. Limitations of lazy training of two-layers neural network. *Advances in Neural Information Processing Systems*, 32, 2019.

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*, pp. 2217–2226, Jul 2019.

Gabriel Ilharco, Marco Tulio Ribeiro, Mitchell Wortsman, Suchin Gururangan, Ludwig Schmidt, Hannaneh Hajishirzi, and Ali Farhadi. Editing models with task arithmetic. *arXiv preprint arXiv:2212.04089*, 2022.

Pavel Izmailov, Dmitrii Podoprikhin, Timur Garipov, Dmitry Vetrov, and Andrew Gordon Wilson. Averaging weights leads to wider optima and better generalization. *arXiv preprint arXiv:1803.05407*, 2018.

Arthur Jacot, Franck Gabriel, and Clément Hongler. Neural tangent kernel: Convergence and generalization in neural networks. *Advances in neural information processing systems*, 31, 2018.

Xisen Jin, Xiang Ren, Daniel Preotiuc-Pietro, and Pengxiang Cheng. Dataless knowledge fusion by merging weights of language models. *arXiv preprint arXiv:2212.09849*, 2022.

Stefani Karp, Ezra Winston, Yuanzhi Li, and Aarti Singh. Local signal adaptivity: Provable feature learning in neural networks beyond kernels. *Advances in Neural Information Processing Systems*, 34:24883–24897, 2021.

Jonathan Krause, Michael Stark, Jia Deng, and Li Fei-Fei. 3d object representations for fine-grained categorization. In *2013 IEEE International Conference on Computer Vision Workshops*, Dec 2013.

Solomon Kullback and Richard A Leibler. On information and sufficiency. *The annals of mathematical statistics*, 22(1):79–86, 1951.

Jens Lehmann, Robert Isele, and et al. Dbpedia – a large-scale, multilingual knowledge base extracted from wikipedia. *Semantic Web*, pp. 167–195, Jan 2015. doi: 10.3233/sw-140134.

Mingxin Li, Zhijie Nie, Yanzhao Zhang, Dingkun Long, Richong Zhang, and Pengjun Xie. Improving general text embedding model: Tackling task conflict and data imbalance through model merging. *arXiv preprint arXiv:2410.15035*, 2024.

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.

Daniel Marczak, Bartłomiej Twardowski, Tomasz Trzcinski, and Sebastian Cygert. Magmax: Leveraging model merging for seamless continual learning. *arXiv preprint arXiv:2407.06322*, 12, 2024.

Daniel Marczak, Simone Magistri, Sebastian Cygert, Bartłomiej Twardowski, Andrew D. Bagdanov, and Joost van de Weijer. No task left behind: Isotropic model merging with common and task-specific subspaces. In *Forty-second International Conference on Machine Learning*, 2025. URL https://openreview.net/forum?id=RBZpAa27ls.

Michael S Matena and Colin A Raffel. Merging models with fisher-weighted averaging. *Advances in Neural Information Processing Systems*, 35:17703–17716, 2022.

Julian McAuley and Jure Leskovec. Hidden factors and hidden topics: understanding rating dimensions with review text. In *Proceedings of the 7th ACM conference on Recommender systems*, pp. 165–172, 2013.

Ryota Miyano and Yuki Arase. Adaptive lora merge with parameter pruning for low-resource generation. *arXiv preprint arXiv:2505.24174*, 2025.

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, pp. 4. Granada, 2011.

Guillermo Ortiz-Jimenez, Alessandro Favero, and Pascal Frossard. Task arithmetic in the tangent space: Improved editing of pre-trained models. *Advances in Neural Information Processing Systems*, 36, 2023.

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 *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=32mrjmaeMP.

Haiquan Qiu, You Wu, Dong Li, Jianmin Guo, and Quanming Yao. Superpose task-specific features for model merging. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 4200–4214, 2025.

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*, pp. 8748–8763, 2021.

Zibo Shao, Baochen Xiong, Xiaoshan Yang, Yaguang Song, Qimeng Zhang, Haifeng Chen, and Changsheng Xu. Pivotmerge: Bridging heterogeneous multimodal pre-training via post-alignment model merging. *arXiv preprint arXiv:2604.22823*, 2026.

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*, Jul 2011.

Wenju Sun, Qingyong Li, Wen Wang, Yang Liu, Yangliao Geng, and Boyang Li. Towards minimizing feature drift in model merging: Layer-wise task vector fusion for adaptive knowledge integration. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025. URL https://openreview.net/forum?id=0KOfAUiHua.

Anke Tang, Li Shen, Yong Luo, Yibing Han, Huisi Ding, and Dacheng Tao. Concrete subspace learning for multi-task model merging. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 36, pp. 61645–61671, 2023.

Anke Tang, Enneng Yang, Li Shen, Yong Luo, Han Hu, Lefei Zhang, Bo Du, and Dacheng Tao. Merging on the fly without retraining: A sequential approach to scalable continual model merging. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025.

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

Qi Wei, Shuo He, Enneng Yang, Tingcong Liu, Haobo Wang, Lei Feng, and Bo An. Representation surgery in model merging with probabilistic modeling. In *International Conference on Machine Learning*, 2025.

Blake Woodworth, Suriya Gunasekar, Jason D Lee, Edward Moroshko, Pedro Savarese, Itay Golan, Daniel Soudry, and Nathan Srebro. Kernel and rich regimes in overparametrized models. In *Conference on Learning Theory*, pp. 3635–3673. PMLR, 2020.

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*, pp. 23965–23998. PMLR, 2022.

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: 3–22, 2016.

Shitao Xiao, Zheng Liu, Peitian Zhang, and Xingrun Xing. Lm-cocktail: Resilient tuning of language models via model merging. *arXiv preprint arXiv:2311.13534*, 2023.

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, 2023.

Enneng Yang, Li Shen, Zhenyi Wang, Guibing Guo, Xiaojun Chen, Xingwei Wang, and Dacheng Tao. Representation surgery for multi-task model merging. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 56332–56356. PMLR, 21–27 Jul 2024a.

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

Gilad Yehudai and Ohad Shamir. On the power and limitations of random features for understanding neural networks. *Advances in neural information processing systems*, 32, 2019.

Kotaro Yoshida, Yuji Naraki, Takafumi Horie, Ryosuke Yamaki, Ryotaro Shimizu, Yuki Saito, Julian McAuley, and Hiroki Naganuma. Mastering task arithmetic: $\backslash\tau$ jp as a key indicator for weight disentanglement. In *The Thirteenth International Conference on Learning Representations*, 2025.

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 *Forty-first International Conference on Machine Learning*, 2024.

Boyuan Zhang, Yingjun Du, Xiantong Zhen, and Ling Shao. Variational task vector composition. *arXiv preprint arXiv:2509.18208*, 2025.

Xiang Zhang, Junbo Zhao, and Yann LeCun. Character-level convolutional networks for text classification. *Advances in neural information processing systems*, 28, 2015.

Zhanpeng Zhou, Zijun Chen, Yilan Chen, Bo Zhang, and Junchi Yan. On the emergence of cross-task linearity in pretraining-finetuning paradigm. In *Forty-first International Conference on Machine Learning*, 2024.

Didi Zhu, Zhongyisun Sun, Zexi Li, Tao Shen, Ke Yan, Shouhong Ding, Chao Wu, and Kun Kuang. Model tailor: Mitigating catastrophic forgetting in multi-modal large language models. In *Forty-first International Conference on Machine Learning*, 2024.

Max Zimmer, Christoph Spiegel, and Sebastian Pokutta. Sparse model soups: A recipe for improved pruning via model averaging. *arXiv preprint arXiv:2306.16788*, 2023.

APPENDIX

The Appendix is organized as follows:

- Appendix A presents t-SNE visualizations of the merged representations. The results demonstrate that representations from different merging methods all form distinct clusters, indicating that they effectively preserve task-relevant discriminative information.
- Appendix B provides supplementary details regarding the experimental setup and implementation.
- Appendix C reports MOMA's performance on additional backbones (ViT-B/16, ViT-L/14) and on NLP tasks. The results confirm that MOMA matches or surpasses restricted alignment methods (e.g., Surgery and ProbSurgery) while maintaining a zero-parameter overhead.
- Appendix D provides additional experiments, including the effect of the hyperparameter α, the performance of MOMA under a larger number of merging tasks, and an extension to LLM backbones.
- Appendix E introduces Aligned-C, an evaluation protocol that mitigates misalignment by fine-tuning classifier heads on few-shot data. Results demonstrate that even with minimal examples, Aligned-C significantly outperforms standard evaluation baselines.

A VISUALIZATION OF MERGED REPRESENTATION

T-SNE visualization of merged representation produced by different methods on other NLP and CV tasks is shown in Figure 9 and Figure 10. From these Figures, we observe that on other datasets, the merged representations also exhibit clear clustering behavior, indicating that the merged representations retain the necessary task-relevant information.

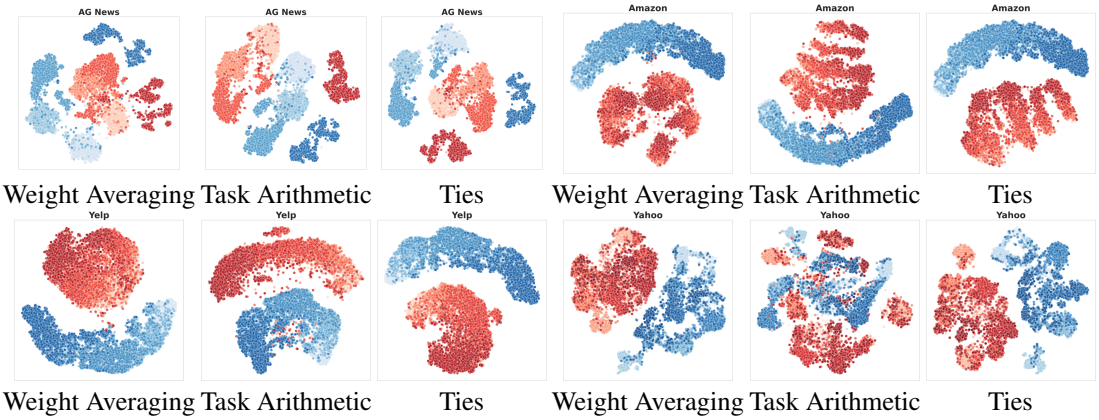


Figure 9: T-SNE visualization of merged representations on NLP tasks.

B EXPERIMENTAL SETUP

B.1 DATASETS

For CV tasks, we select SUN397 (397 classes) Xiao et al. (2016), Cars (196 classes) Krause et al. (2013), RESISC45 (45 classes) Cheng et al. (2017), EuroSAT (10 classes) Helber et al. (2019), SVHN (10 classes) Netzer et al. (2011), GTSRB (43 classes) Stallkamp et al. (2011), MNIST (10 classes) Deng (2012), and DTD (47 classes) Cimpoi et al. (2014).

For NLP tasks, we select AG News (4 classes) Zhang et al. (2015), Yelp (5 classes) yel (2014), Amazon (5 classes) McAuley & Leskovec (2013), Yahoo (10 classes) Zhang et al. (2015), and DBpedia (14 classes) Lehmann et al. (2015).

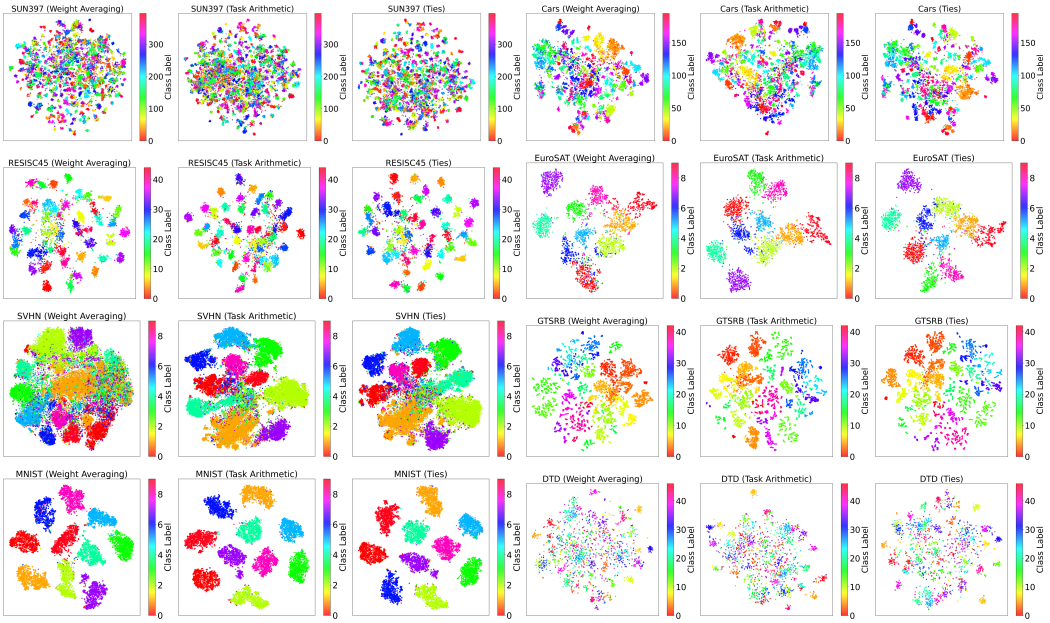


Figure 10: T-SNE visualization of merged representations for each merging method on CV tasks. Noted, due to the large number of classes in the CV datasets, we did not display the T-SNE results of the fine-tuned model for better visualization.

Our dataset selection is consistent with prior studies Yang et al. (2024a); Wei et al. (2025). Detailed descriptions and statistical characteristics of the datasets can be found in these works.

B.2 BASELINES

Following Wei et al. (2025), we conduct comparisons with the following representative methods.

- **Pre-trained Model**: The foundational pre-trained backbone evaluated directly in a zero-shot manner without task-specific tuning.
- **Fine-tuned Model**: Models fine-tuned independently on each specific task, serving as the single-task performance upper bound.
- **Traditional MTL**: A model trained jointly on all datasets using a shared backbone, serving as the multi-task upper bound.
- **Task Arithmetic (TA)** Ilharco et al. (2022): Characterizes task-specific updates as vectors in the parameter space, enabling the manipulation of model behaviors through arithmetic operations such as addition and negation.
- **Ties** Yadav et al. (2023): Mitigates parameter interference by pruning redundant updates, resolving sign conflicts via a "dominant direction" criterion, and keeping only the consensus parameters.
- **Fisher Merging** Matena & Raffel (2022): Utilizes the Fisher Information Matrix to assign higher importance weights to critical parameters, thereby preserving task-specific knowledge during the averaging process.
- **RegMean** Jin et al. (2022): Formulates merging as a least-squares regression problem, minimizing the distance between the activations of the merged model and those of fine-tuned models.
- **Concrete TA & AM** Tang et al. (2023): Proposes a continuous relaxation of Task Arithmetic, employing a concrete distribution to learn optimal fractional masks and merging coefficients.

Metric	Surgery	ProbSurgery	MOMA
Extra Parameters	131,072	1,048,576	**0**
Extra Inference FLOPs	33,296	262,784	**0**

Table 4: Comparison of extra parameters and inference FLOPs.

- **LW AdaMerging** Yang et al. (2024b): An unsupervised approach that adaptively learns layer-wise or parameter-wise merging coefficients by minimizing the prediction entropy on unlabeled test data.
- **TSV-M** Gargiulo et al. (2025): A training-free method that decomposes task vectors via Singular Value Decomposition to extract task-specific singular subspaces, reducing inter-task interference through subspace orthogonalization.
- **Surgery** Yang et al. (2024a): Addresses representation discrepancy by introducing learnable auxiliary modules. It explicitly aligns the representation of fine-tuned models with the merged backbone to minimize geometric misalignment.
- **ProbSurgery** Wei et al. (2025): A state-of-the-art extension of Surgery that incorporates uncertainty modeling. It formulates representation alignment as a probabilistic process to robustly handle noise and parameter conflicts.

B.3 Implementation Details

Setting of Hyperparameter For the scaling hyperparameters of the Task Arithmetic and Ties models, we follow the original paper by using a portion of the labeled data for selection. These parameters are then fixed across all experiments to ensure a fair comparison. We employ the Adam optimizer across all experiments. For both CV and NLP tasks, we set the mask learning rate to 0.1 and the L2 regularization weight to 5×10^{-9}, enabling the model to converge to a sparsity level (masking rate) between 10% and 90%. For task-specific parameters, we use a learning rate of 1×10^{-3} with a batch size of 16 for CV tasks, and 5×10^{-4} with a batch size of 24 for NLP tasks.

Formulation of Classifiers in CV Tasks In the ViT series of experiments, the image input is processed through the image encoder, and text inputs (label names) are processed through the text encoder, obtaining the image embedding $\mathbf{E}_{\text{image}}$ and text embeddings $\mathbf{E}_{\text{text}}$ respectively. Classification is performed by matching $\mathbf{E}_{\text{image}}$ with $\mathbf{E}_{\text{text}}$. In the settings of most model merging papers, the text encoder is frozen, meaning that $\mathbf{E}_{\text{text}}$ remains unchanged. In this case, we consider $\mathbf{E}_{\text{text}}$ can be understood as the classifier parameter $\mathbf{W}_t$, that is, initializing the classifier with label name embeddings. This approach is consistent with the way many text classification methods initialize label embeddings. This structural equivalence explains why we explicitly discuss a 'classifier head' despite employing a ViT-based architecture.

Computational Complexity For training cost, MOMA, Surgery, and ProbSurgery are essentially comparable, since all three require unlabeled data and an additional optimization stage. There is no substantial difference in this regard. For extra parameters, we follow the calculation protocol and parameter settings reported in Table 6 of the ProbSurgery Wei et al. (2025) paper. For the additional inference FLOPs, we compute them per test sample by summing the extra matrix multiplications, activation functions, and related operations introduced by each method. The resulting comparison is summarized in Table 4. These results indicate that MOMA is more efficient at deployment, since it requires no extra parameters and no additional inference-time computation.

C Main Results on More Backbones and Tasks

C.1 Results on CV Tasks via ViT-B/16 and ViT-L/14

Main results of MOMA on ViT-B/16 and ViT-L/14 are shown in Tables 5 and 6. This finding is in line with the results observed for ViT-B/32.

Method	SUN397	Cars	RESISC45	EuroSAT	SVHN	GTSRB	MNIST	DTD	Avg.
Pre-trained	63.8	64.6	65.7	54.5	52.0	43.3	51.7	45.1	55.0
Fine-tuned	81.7	86.7	96.8	99.7	97.8	99.1	99.7	82.0	93.0
Fisher Merging (Matena & Raffel, 2022)	68.5	69.9	75.2	80.4	73.2	61.2	94.5	50.7	71.7
RegMean (Jin et al., 2022)	69.1	71.6	77.6	88.8	83.7	70.2	96.9	54.6	76.6
Task Arithmetic (Ilharco et al., 2022)	61.1	65.9	74.0	76.2	88.0	73.9	98.4	53.0	73.8
w/ Surgery (Yang et al., 2024a)	68.3	72.3	88.7	97.7	91.0	89.5	98.9	72.9	84.9
w/ ProbSurgery (Wei et al., 2025)	71.9	79.3	95.8	99.7	93.5	99.0	99.4	81.5	90.0
w/ MOMA	77.0	81.1	95.4	99.5	97.6	99.0	99.5	80.5	**91.2**
Ties (Yadav et al., 2023)	69.1	72.5	80.5	84.0	85.0	71.5	98.1	54.9	77.0
w/ Surgery	73.0	76.2	90.7	98.1	89.7	86.7	98.7	75.2	86.0
w/ ProbSurgery	75.1	80.9	95.8	99.7	92.2	99.0	99.5	82.0	90.5
w/ MOMA	78.4	83.2	95.9	99.5	97.8	99.1	99.6	81.2	**91.8**
LW AdaMerging Yang et al. (2024b)	70.2	80.7	81.6	94.8	91.6	95.8	98.5	66.2	84.9
w/ Surgery	73.6	81.5	90.4	98.5	93.2	97.4	98.9	77.0	88.8
w/ ProbSurgery	75.3	83.7	95.5	99.8	94.9	99.0	99.4	82.3	91.2
w/ MOMA	80.9	86.1	96.5	99.6	97.8	99.1	99.6	82.0	**92.7**
TSV-M Gargiulo et al. (2025)	72.7	80.4	89.0	96.5	93.9	93.9	99.2	72.6	87.3
w/ Surgery	74.6	81.9	93.4	98.8	94.7	96.6	99.3	77.7	89.6
w/ ProbSurgery	75.9	83.1	95.1	99.4	94.4	98.0	99.3	80.6	90.7
w/ MOMA	77.4	83.2	96.0	99.5	97.8	99.1	99.6	81.7	**91.8**

Table 5: Comparison of MOMA on ViT-B/16.

Method	SUN397	Cars	RESISC45	EuroSAT	SVHN	GTSRB	MNIST	DTD	Avg.
Pre-trained	66.8	77.7	71.0	59.9	58.4	50.5	76.3	55.3	64.5
Fine-tuned	84.8	92.3	97.3	99.7	98.1	99.2	99.6	84.1	94.4
Fisher Merging (Matena & Raffel, 2022)	69.2	88.6	87.5	93.5	80.6	74.8	93.3	70.0	82.2
RegMean (Jin et al., 2022)	73.3	81.8	86.1	97.0	88.0	84.2	98.5	60.8	83.7
Concrete TA	74.6	86.2	89.0	96.7	93.6	93.4	99.1	66.9	87.4
Concrete AM	77.8	91.2	92.1	97.0	94.4	97.9	99.0	79.5	91.1
Task Arithmetic (Ilharco et al., 2022)	73.9	82.1	86.6	94.1	87.9	86.7	98.9	65.6	84.5
w/ Surgery (Yang et al., 2024a)	75.7	84.4	93.1	98.8	91.3	93.4	99.1	76.1	89.0
w/ ProbSurgery (Wei et al., 2025)	77.0	87.5	96.2	99.7	94.2	99.1	99.4	81.8	92.0
w/ MOMA	81.0	90.6	96.7	99.5	97.9	99.2	99.6	83.7	**93.5**
Ties (Yadav et al., 2023)	76.5	85.0	89.3	95.7	90.3	83.3	99.0	68.8	86.0
w/ Surgery	76.5	85.9	93.7	99.2	89.7	92.0	99.1	78.1	89.3
w/ ProbSurgery	77.7	88.4	96.7	99.7	93.2	99.0	99.5	82.4	92.6
w/ MOMA	82.1	90.9	97.1	99.6	98.1	99.3	99.7	83.8	**93.8**
LW AdaMerging Yang et al. (2024b)	79.0	90.3	90.8	96.2	93.4	98.0	99.0	79.9	90.8
w/ Surgery	80.3	90.8	94.3	98.2	94.1	98.7	99.2	82.5	92.3
w/ ProbSurgery	80.7	91.4	96.2	99.6	95.9	99.2	99.5	84.1	93.8
w/ MOMA	84.4	92.1	97.2	99.7	98.1	99.2	99.7	84.0	**94.3**
TSV-M Gargiulo et al. (2025)	79.0	89.8	93.9	98.8	95.3	96.1	99.5	79.1	91.4
w/ Surgery	79.8	90.4	95.8	99.5	96.0	98.0	99.6	81.9	92.6
w/ ProbSurgery	80.0	91.4	96.9	99.6	96.4	98.9	99.6	84.3	93.4
w/ MOMA	81.6	91.3	97.0	99.7	98.1	99.2	99.7	83.7	**93.8**

Table 6: Comparison of MOMA on ViT-L/14.

C.2 Results on NLP Tasks

Main results of MOMA on NLP tasks are shown in Table 7. Note that TA + MOMA and Ties + MOMA both match the fine-tuned upper bound (76.3) on NLP tasks, with several individual datasets even slightly exceeding it. We therefore omit the LW AdaMerging and TSV-M variants for NLP, as no further improvement is possible against the fine-tuned ceiling.

D Additional Experimental Results

D.1 Analysis of the Hyperparameter α

In Eq.7, the hyperparameter α controls the strength of the regularization objective on the learned transformation. Its effect on performance is shown in Table 8. Empirically, a smaller α leads to better performance. This suggests that weaker regularization gives the learned transformation more freedom and makes it closer to a general linear transformation.

Method	AG News	Yelp	Amazon	Yahoo	DBpedia	Avg.
Pre-trained Model	24.9	19.5	19.7	9.9	8.8	16.5
Fine-tuned Model	91.6	60.4	57.9	72.7	98.8	76.3
Traditional MTL	90.6	59.1	55.6	71.3	98.5	75.0
Task Arithmetic	82.9	55.8	48.4	53.1	81.5	64.3
w/ Surgery	89.8	58.4	55.4	70.3	98.0	74.4
w/ ProbSurgery	90.7	60.4	56.4	72.3	98.7	75.7
w/ MOMA	91.5	60.8	57.7	72.9	98.8	**76.3**
Ties	83.3	55.7	47.0	55.9	83.4	65.1
w/ Surgery	90.4	58.8	56.5	71.5	98.7	75.2
w/ ProbSurgery	90.9	59.6	56.3	71.7	98.8	75.5
w/ MOMA	91.4	60.7	57.7	73.0	98.9	**76.3**

Table 7: Comparison on NLP tasks. The results exhibit trends consistent with the CV tasks in Table 2.

α	0.00001	0.00005	0.0001	0.0005	0.001
MOMA	89.3	88.9	88.9	88.0	87.7

Table 8: Effect of α in Eq. 7.

D.2 Evaluation on a Larger Number of Merging Tasks

To evaluate the performance of MOMA with a larger number of merging tasks, we adopt the checkpoints and datasets from Gargiulo et al. (2025); Marczak et al. (2025) and compare MOMA with Surgery and ProbSurgery. It is worth noting that, for the 8-task setting, we also use the checkpoints from Gargiulo et al. (2025); Marczak et al. (2025), which are slightly different from the checkpoints used in the Surgery Yang et al. (2024a) and in our main results. Therefore, the reported results on 8 tasks may differ from those in the main experiments. Nevertheless, all results in Table 9 are reproduced under the same checkpoint setting, ensuring a fair comparison. From the Table, MOMA still demonstrates effective performance when merging a larger number of tasks.

D.3 Generative LLM as Merging Backbone

Similar to most existing model merging methods, the current paper does not include experiments on generative LLMs. In the generative LLM setting, the absorption argument used in our current classification setup may no longer be as direct, since the model relies on a shared LM head rather than fixed task-specific classifier heads.

To better understand whether the same post-merging misalignment phenomenon may also appear in LLMs, we conducted an additional representation analysis. Specifically, we used the checkpoints from Lu et al. (2024), merged LoRA modules via task arithmetic, and took the last-token hidden state from the final layer as the sentence representation. We then compared the embeddings before

Method	ViT-B/32			ViT-B/16			ViT-L/14		
	8 tasks	14 tasks	20 tasks	8 tasks	14 tasks	20 tasks	8 tasks	14 tasks	20 tasks
Zero-shot	48.3	57.2	56.1	55.3	61.3	59.7	64.7	68.2	65.2
Fine-tuned	92.8	90.9	91.3	94.6	92.8	93.2	95.8	94.3	94.7
Weight Averaging	66.3	64.3	61.0	72.2	69.5	65.3	79.6	76.7	71.6
Consensus TA Wang et al. (2024)	75.0	70.4	65.4	79.4	74.4	69.8	86.3	82.2	79.0
TSV-M	85.9	80.1	77.1	89.0	84.6	80.6	93.0	89.2	87.7
Iso-CTS Marczak et al. (2025)	86.2	81.7	78.1	91.1	86.4	82.4	94.7	91.0	90.1
Task Arithmetic	70.8	65.3	60.5	75.4	70.5	65.8	84.9	79.4	74.0
w/ Surgery	81.4	83.1	83.4	86.9	86.9	87.3	91.0	89.8	90.7
w/ ProbSurgery	86.4	84.7	85.2	91.5	88.1	89.0	94.0	90.8	91.7
w/ MOMA	**87.8**	**85.2**	**87.2**	**94.0**	**89.5**	**89.8**	**95.7**	**91.7**	**92.5**

Table 9: Comparison of more merging tasks.

Method	Raw	Affine	Linear	Invertible	Similarity	Orthogonal	Scaling
Cosine Similarity	0.79	0.98	0.97	0.96	0.97	0.97	0.79
Euclidean Distance	22.6	5.9	6.8	6.9	6.1	5.9	21.9

Table 10: Representation misalignment on LLM backbone.

and after merging, and measured their Cosine Similarity and Euclidean Distance to the fine-tuned representations under different alignment transformations, following the protocol in Figure 5.

Our results, shown in Table 10, suggest that a clear representation misalignment still exists before and after merging in this LLM setting, and that this misalignment can be approximated by an orthogonal transformation. Our analysis suggests that the key orthogonal misalignment phenomenon identified in this paper may also arise in LLMs.

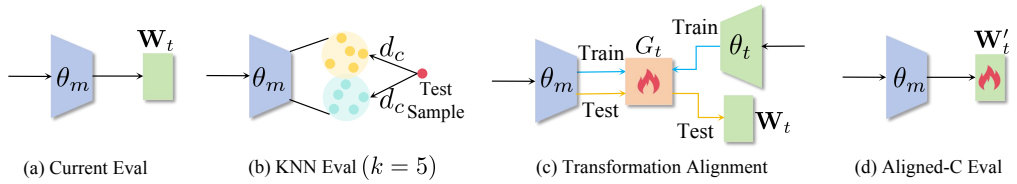
E ALIGNED-C: ALIGNING THE CLASSIFIER FROM THE PERSPECTIVE OF THE EVALUATION

E.1 CURRENT EVALUATION OF MERGED MODELS

Existing merging methods in classification tasks primarily focus on merging fine-tuned encoders, while not performing the merging operation on fine-tuned classifiers. That is, the merging encoders should be evaluated on the fine-tuned classifier, which can be formulated as:

$$\hat{Y}_t = g(x;\theta_m)^\top \mathbf{W}_t, \tag{10}$$

where $\mathbf{W}_t$ represents the weight of the fine-tuned classifier of task t, which maps the merging outputs to the predicted labels $\hat{Y}_t$. In the following descriptions, this currently adopted evaluation protocol is referred to as 'Current Eval', shown in Figure 11.



(a) Current Eval (b) KNN Eval $(k = 5)$ (c) Transformation Alignment (d) Aligned-C Eval

Figure 11: Illustrations of some variants referred to in this paper.

E.2 KNN EVALUATION OF MERGED MODELS

As described in Section 3.2, instead of utilizing the fine-tuned classifier head, we select K samples as anchors to perform k-Nearest Neighbor (KNN) classification to evaluate the merging methods. We refer to this protocol as 'KNN Eval'.

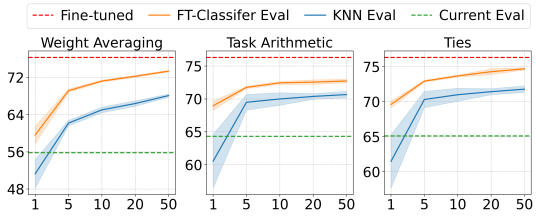
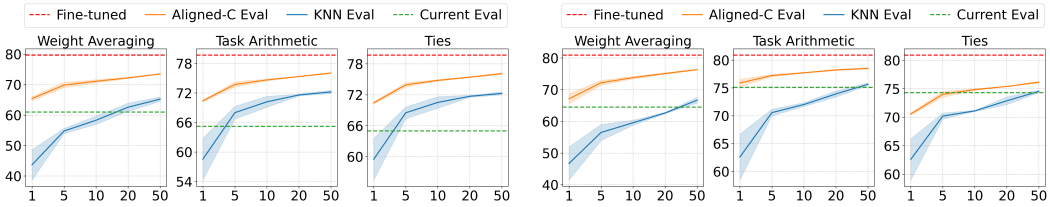


Figure 12: Performance of different evaluation protocols under varying numbers of few-shot examples K using BERT. Corresponding detailed results for $k = 5$ are shown in Table 11.

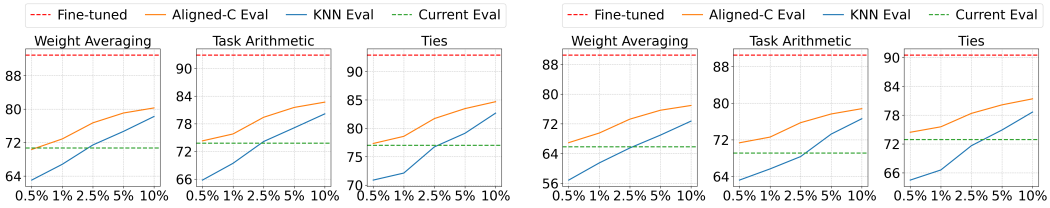
E.3 ALIGNED-C EVALUATION

Model merging combines the parameters of multiple fine-tuned models to produce a merged model. Consequently, as discussed before, the merged representation inevitably shifts in the latent space



(a) Performance on T5-base (b) Performance on T5-large

Figure 13: Performance of different evaluation protocols under T5-base and T5-large. Detailed results on $k = 20$ are reported in Table 12.



(a) CV tasks performance on ViT-B/16 (b) CV tasks performance on ViT-B/32

Figure 14: The performance of different evaluation protocols on CV tasks under ViT-B/16 and ViT-B/32.

relative to the fine-tuned representation, and this shift is unavoidable. In this case, using $\mathbf{W}_t$, which is aligned well with fine-tuned representations, to evaluate the merged model is inappropriate. The evaluation result reflects not the capability of the merged model, but rather its compatibility with the fine-tuned classifier $\mathbf{W}_t$. Moreover, through experiments, we find that simply aligning the merged representation and the fine-tuned representation in the latent space can significantly improve the classification performance. Naturally, this motivates us to propose a new and reasonable evaluation protocol.

Revisiting the objective of model merging, which aims to generate a new model θ_m using $\theta_1, ..., \theta_T$ that performs well across all T tasks, we note that the upper bound of θ_m's performance on a specific task t corresponds to the result from the fine-tuned model θ_t. Therefore, the objective of model merging is to minimize this performance discrepancy of θ_m and θ_t Ortiz-Jimenez et al. (2023). By optimizing this discrepancy, we could directly train an aligned classifier for θ_m. That is, in the new evaluation protocol, we introduce an additional post-processing step while keeping the parameter scale and network structure unchanged, referred to as Aligned-C Eval, which is defined as:

$$\arg\min_{\mathbf{W}'_t} \sum_{t=1}^{T} KL(g(x;\theta_m)^\top \mathbf{W}'_t, g(x;\theta_t)^\top \mathbf{W}_t), \tag{11}$$

where KL is the KL divergence Kullback & Leibler (1951), and the new classifier $\mathbf{W}'_t$ is initialized with the value of $\mathbf{W}_t$.

In model merging, obtaining a few-shot dataset for each task is a relatively flexible requirement. Many studies use few-shot data to guide the merging methods Xiao et al. (2023), and some even rely on large-scale labeled or unlabeled data to determine merging weights Yang et al. (2024b); Jin et al. (2022) or the conflict parameters to be pruned Ortiz-Jimenez et al. (2023); Zhu et al. (2024). As a result, we also incorporate a few-shot sample for aligned classifier training in Aligned-C Eval.

Comparable results of Current Eval, KNN Eval, and Aligned-C Eval are presented in Figure 12, Figure 13 and Figure 14, and detailed results, utilizing few-shot samples are shown Table 11, Table 12, and Table 13. From the results, we observe that after simple training, even on 1-shot samples on NLP tasks, Aligned-C Eval performs significantly better than the Current Eval. This further demonstrates that the misalignment issue is highly manageable. Such a lightweight and computationally feasible post-processing operation facilitates the evaluation of merged models.

Method	AG News	Yelp	Amazon	Yahoo	DBpedia	Average
Fine-tuned model	91.6	60.4	57.9	72.7	98.8	76.3
Multi-task Learning	90.6	59.1	55.6	71.3	98.5	75.0
Weight Averaging	79.2	49.8	45.0	50.3	55.1	55.8
+ KNN Eval	81.9 ± 1.6	42.1 ± 4.3	41.2 ± 1.5	54.1 ± 3.5	91.5 ± 1.8	62.2 ± 0.6
+ Aligned-C Eval	**86.0 ± 1.0**	**51.5 ± 1.9**	**46.3 ± 0.6**	**66.2 ± 1.1**	**95.8 ± 0.4**	**69.2 ± 0.4**
Task Arithmetic	82.9	55.8	48.4	53.1	81.5	64.3
+ KNN Eval	86.4 ± 1.3	54.2 ± 2.6	49.0 ± 1.6	61.6 ± 1.0	96.3 ± 0.2	69.5 ± 1.2
+ Aligned-C Eval	**87.5± 0.7**	**55.6± 0.5**	**51.5 ± 0.6**	**67.6 ± 0.7**	**96.6 ± 0.3**	**71.8 ± 0.2**
Ties	83.3	55.7	47.0	55.9	83.4	65.1
+ KNN Eval	87.0 ± 1.6	54.6 ± 2.9	49.5 ± 1.5	63.4 ± 0.6	97.1 ± 0.2	70.3 ± 1.2
+ Aligned-C Eval	**88.1 ± 0.5**	**57.1 ± 0.9**	**52.6 ± 0.6**	**69.1 ± 0.6**	**97.7 ± 0.1**	**72.9 ± 0.1**

Table 11: Comparison of Aligned-C Eval and KNN Eval with the corresponding Current Eval (represented by merging method name). The results are based on a few-shot setting of $k = 5$.

(a) T5-base

Method	AG News	Yelp	Amazon	Yahoo	DBpedia	Avg.
Fine-tuned	94.5	66.6	62.9	75.2	99.1	79.6
WA	79.0	51.8	51.7	42.2	79.9	60.9
+KNN	**85.4**	48.8	47.3	41.5	94.6	63.5
+Aligned-C	85.3	**60.5**	**55.3**	**61.8**	**97.1**	**72.0**
TA	80.6	59.2	55.2	51.5	79.5	65.2
+KNN	87.7	62.0	57.4	53.5	97.1	71.5
+Aligned-C	**88.2**	**64.0**	**58.2**	**67.6**	**98.4**	**75.3**
Ties	80.0	59.2	55.4	50.7	79.3	64.9
+KNN	87.3	62.4	56.7	54.7	97.2	71.7
+Aligned-C	**88.3**	**63.8**	**58.1**	**68.0**	**98.4**	**75.4**

(b) T5-large

Method	AG News	Yelp	Amazon	Yahoo	DBpedia	Avg.
Fine-tuned	94.9	68.6	64.9	77.1	99.1	80.9
WA	79.2	55.5	44.3	49.0	94.5	64.5
+KNN	87.9	43.6	40.1	47.2	95.6	62.9
+Aligned-C	**88.0**	**62.6**	**59.7**	**67.9**	**98.2**	**75.3**
TA	90.8	58.3	60.5	72.9	92.7	75.1
+KNN	**91.3**	59.0	54.4	62.7	97.7	73.0
+Aligned-C	91.2	**65.9**	**61.9**	**73.5**	**98.7**	**78.3**
Ties	91.3	56.0	**60.3**	**73.2**	90.4	74.3
+KNN	**91.8**	59.1	51.5	65.3	97.5	73.0
+Aligned-C	88.4	**63.9**	58.2	68.0	**98.4**	**75.4**

Table 12: Comparison on five NLP classification tasks using T5-base and T5-large under few-shot setting ($k = 20$). Avg. represents Average. KNN represents KNN Eval, and Aligned-C refers to our proposed Aligned-C Eval.

E.4 θ_m VS. θ_b

Previous analyses demonstrate that different θ_m achieves significant improvements when evaluated with an aligned classifier. To further verify that the performance improvement stems from the inherent capability of the θ_m itself rather than additional information introduced by aligning, we tested the results of the original pre-trained models θ_b under different evaluation protocols. The experimental results are presented in Table 14.

From the Table, we observe that regardless of the evaluation protocol, θ_m consistently outperforms θ_b. This demonstrates that all θ_m models effectively incorporate knowledge from fine-tuned models. Meanwhile, Aligned-C Eval does not show significant improvements over Current Eval when applied to θ_b, indicating that models with limited parameters (only the classifier) cannot retain sufficient task-specific information. This proves that Aligned-C Eval just activates the fine-tuned information that θ_m already contains.

E.5 ALIGNED-C VS. SURGERY

The related work Surgery Yang et al. (2024a) characterizes the overall difference between the merged representations and the fine-tuned representations as "representation bias", and reduces this bias by training additional parameters on a large amount of unlabeled data.

We argue that merging models naturally alters the representation distribution (representation bias) but does not necessarily harm classification performance. The main cause of performance decline is the misalignment between the merged representation and the fine-tuned classifier. We also show that this misalignment is simple (converging to an orthogonal transformation) and does not affect the intrinsic properties of merged representations. Consequently, it can be mitigated by training an

(a) ViT-B/16

Method	SUN397	Cars	RESISC45	EuroSAT	SVHN	GTSRB	MNIST	DTD	Average
Fine-tuned	81.8	86.8	96.9	99.7	97.8	99.1	99.7	82.0	92.9
Weight Averaging	67.7	70.0	75.3	79.5	74.9	60.1	94.4	43.8	70.7
+KNN Eval	65.9	69.4	**87.6**	88.8	77.3	81.7	95.5	**59.6**	78.2
+Aligned-C Eval	**71.5**	**73.6**	86.7	**91.9**	**79.5**	**83.1**	**97.1**	58.7	**80.3**
Task Arithmetic	61.1	65.9	74.0	76.2	88.0	73.9	98.4	53.0	73.8
+KNN Eval	65.0	67.9	84.9	85.9	88.9	88.5	**98.8**	61.3	80.2
+Aligned-C Eval	**68.1**	**72.3**	**86.3**	**90.8**	**90.2**	**90.3**	**98.8**	**64.6**	**82.7**
Ties	69.1	72.5	80.5	84.0	85.0	71.5	98.1	54.9	77.0
+KNN Eval	68.9	73.8	88.4	91.2	86.7	88.2	98.4	66.1	82.7
+Aligned-C Eval	**73.1**	**77.0**	**89.1**	**93.9**	**88.0**	**89.7**	**98.7**	**68.1**	**84.7**

(b) ViT-B/32

Method	SUN397	Cars	RESISC45	EuroSAT	SVHN	GTSRB	MNIST	DTD	Average
Fine-tuned	75.3	77.7	96.1	99.7	97.5	98.7	99.7	79.4	90.5
Weight Averaging	65.3	63.4	71.4	71.7	64.2	52.8	87.5	50.1	65.8
+KNN Eval	63.5	57.7	82.8	83.5	69.0	80.3	93.4	51.8	72.7
+Aligned-C Eval	**69.1**	**67.1**	**82.9**	**88.6**	**73.5**	**83.1**	**95.4**	**55.7**	**76.9**
Task Arithmetic	55.2	54.9	66.7	78.9	80.2	69.7	97.3	50.4	69.1
+KNN Eval	59.7	58.4	79.4	88.7	83.7	89.6	97.8	55.4	76.6
+Aligned-C Eval	**62.5**	**61.5**	**80.4**	**89.9**	**86.4**	**89.9**	**98.4**	**61.3**	**78.8**
Ties	65.0	64.4	74.8	77.4	81.2	69.3	96.5	54.5	72.9
+KNN Eval	65.2	60.8	84.0	87.7	83.2	88.5	97.4	62.5	78.6
+Aligned-C Eval	**69.7**	**68.7**	**84.6**	**89.8**	**85.8**	**89.9**	**98.0**	**64.7**	**81.4**

Table 13: Comparison on CV tasks by ViT. The results in this Table are based on a sampling rate of 10%.

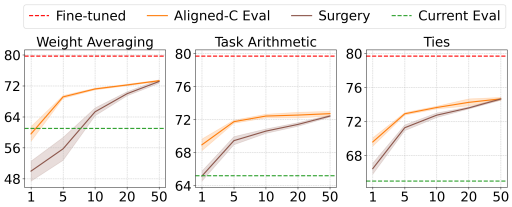


Figure 15: Comparison of Aligned-C Eval and Surgery. Aligned-C outperforms Surgery particularly when only a small number of samples are available.

Method	θ_b	WA	TA	Ties
Current Eval	17.5	55.8	64.3	65.1
KNN Eval	20.9	62.2	69.5	70.3
Aligned-C Eval	22.0	69.2	71.8	72.9

Table 14: Comparison of different θ_m and θ_b on different evaluation protocols.

aligned classifier for each task with very few samples. In addition, the Aligned-C can also keep the parameter scale and network structure unchanged. In the context of model merging, we believe that such a lightweight and computationally efficient approach is more favorable. Results in Figure 15 demonstrate that the Aligned-C outperforms Surgery when only a small number of samples are available.