Training-Free Task Classification for Multi-Task Model Merging

Jungyong Son[1], Jinwook Jung[1], and Sungyong Baik[1,2⋆]

[1]Department of Artificial Intelligence, [2]Department of Data Science
Hanyang University, Republic of Korea
{jungyongs,jjw970517,dsybaik}@hanyang.ac.kr

Abstract. Ever since the advent of foundation models and the pre-training–finetuning paradigm, there have been numerous efforts to merge multiple task-specific experts into a single multi-task model. Prior work largely focuses on finding a single merged model, but it often underperforms individual experts due to parameter interference. To resolve this, dynamic model merging employs routing to activate task-relevant parameters per input. However, existing routers typically require either additional training with abundant labeled datasets or assume the access to task IDs of each input at inference time. In this work, we aim to close the gap to expert performance without additional training or task-ID-access assumption. To this end, we formulate routing as training-free task classification for each test input. Using singular value decomposition (SVD)-based low-rank manifold approximations for each task, SiM scores tasks by the projection residual of the test input feature onto each task manifold and routes accordingly. The task manifolds are pre-computable offline from a pretrained backbone using a small per-task support set (e.g., 32 examples per task) prior to merging process, requiring no router training and no data during the merging process. Moreover, SiM integrates seamlessly with subspace-/mask-based merging that represents task-expert via lightweight compressed task vectors, avoiding the need to store full expert parameters. Experiments across computer vision and natural language processing benchmarks under task-unknown inference demonstrate that SiM substantially improves merged-model performance and consistently narrows the gap to individual task experts. Our code is available at https://github.com/BAIKLAB/SiM

Keywords: Dynamic model merging · Training-free task classification

1 Introduction

With the emergence of foundation models [1, 36, 47], the pretraining-finetuning paradigm—the practice of pre-training foundation models followed by task-specific fine-tuning—has become one of the dominant approaches in many areas of machine learning [4, 37, 50, 52]. The pretraining-finetuning strategy has led to

⋆ Corresponding author.

Table 1: Dynamic model merging methods and their requirements. Given T task-specific models, N denotes the number of test samples, P_f denotes the number of fine-tuned model parameters, P_c denotes the number of compressed fine-tuned model parameters ($P_c \ll P_f$), P_r denotes the number of router parameters, M denotes the task-specific binary mask, B denotes the number of layer blocks, d denotes the feature dimension, k denotes the number of ranks.

Method	Additional training	Forward passes	Additional memory	Task ID requirement
EMR-Merging [16]	×	$\mathcal{O}(1)$	$\mathcal{O}(P_f + TM)$	✓
TALL Mask [49]	×	$\mathcal{O}(1)$	$\mathcal{O}(P_f + TM)$	✓
TWIN-Merging [28]	✓	$\mathcal{O}(1)$	$\mathcal{O}(P_r + TP_c)$	×
DaWin [33]	×	$\mathcal{O}(N + T)$	$\mathcal{O}(TP_f)$	×
WEMoE [45]	✓	$\mathcal{O}(1)$	$\mathcal{O}(BP_r + TP_f)$	×
MoW-Merging [57]	✓	$\mathcal{O}(1)$	$\mathcal{O}(P_r + TP_f)$	×
SiM(Ours)	×	$\mathcal{O}(1)$	$\mathcal{O}(Tdk + TP_c)$	×

the proliferation of these task-specific models, presenting new challenges related to knowledge and model management.

Accordingly, several works have attempted to integrate the knowledge from these task-specific models (or task experts) by merging them via interpolation on task-experts parameters [13, 17, 19, 29, 30, 55]. By doing so, a merged model can replace all task experts, removing the need to store an expert model for each task and greatly reducing memory footprints. However, these works primarily focus on finding a single static merged model, often failing to reach the performance of individual experts because of parameter interference.

In light of the challenge, few recent works have attempted to dynamically merge model parameters, employing input-adaptive interpolation of parameters [13, 16, 28, 33, 45, 49, 57]. While such input-adaptive interpolation methods have led to substantial performance improvement, few early dynamic model merging works have employed full parameters of each expert [33, 45, 57], undermining the very goal of model merging and incurring memory overhead. Recent works have attempted to seek a middle ground between performance and memory efficiency by compressing task vectors to efficiently store task expert knowledge [13, 16, 28, 49]. However, they either require additional training with labeled task datasets [28, 45, 57] or also require access to task identities of each input sample during inference [13, 16, 49], which limits their applicability to real-world scenarios.

In this work, we introduce a plug-and-play dynamic model merging framework that does not require access to task identity or additional training for routing mechanism. Particularly, we formulate the goal of finding input-adaptive task coefficients as task classification for each input data. To this end, we propose to leverage singular value decomposition (SVD) to construct low-rank manifold approximations for each task. Specifically, the task identity of each test input is de-

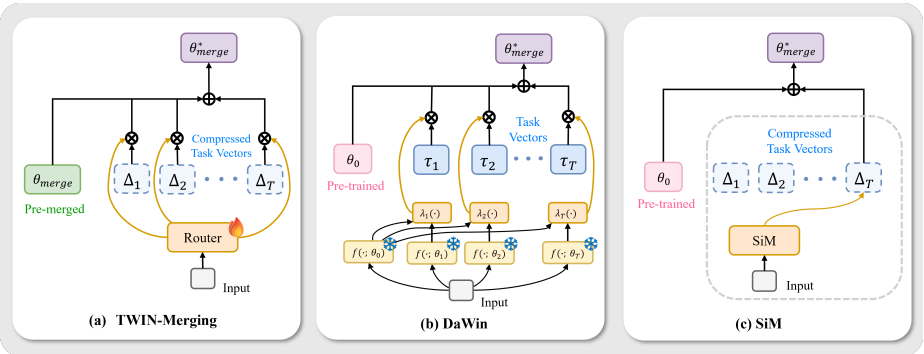

Fig. 1: Comparison of dynamic model merging methods. Subfigure (a) shows that in TWIN-Merging, the coefficients are provided by a router that requires additional training. Subfigure (b) illustrates that DaWin, while training-free, requires maintaining all task-specific experts in memory to calculate coefficients. Subfigure (c) shows that SiM dynamically determines coefficients without any additional training and eliminates the need to hold multiple experts in memory.

termined by the projection residual of the test input feature onto each manifold. A key advantage of this approach is that task manifolds can be pre-computed offline, prior to merging process, using only a small support set (e.g., 32 examples per task) with a pre-trained backbone. Thus, SiM removes the need for router training or the presence of data during the actual merging phase or inference (Tab. 1 explicitly details the comparison between SiM and other dynamic model merging methods). Ephemeral access to a small support set by our method mitigates issues regarding data storage requirements of previous works [17, 55], such as memory overhead and privacy concerns.

We further note that our proposed **Si**ngular-vector-based **M**anifold (**SiM**) framework can be readily integrated with existing subspace-based merging methods that compress task vectors or expert parameters [13, 16, 49]. Notably, the integration with such lightweight compressed task vectors allows SiM to strike a better balance in performance-memory-efficiency trade-off. With compressed task vectors and SiM together, dynamic model merging is performed as follows: (1) task classification; (2) selecting a compressed task vector corresponding to predicted task ID; (3) adding the selected compressed task vector to pretrained model to obtain a model to use for inference.

Extensive experimental results across vision and NLP domains demonstrate that our proposed framework SiM substantially outperforms previous dynamic model merging methods that either rely on additional training or known-task-identity assumption, while overcoming these two limitations.

2 Related work

The purpose of multi-task model merging is to build a multi-task model by combining the parameters of task-specific models. Task Arithmetic [17] merges models by task vectors, which represent the difference between the parameters of the task-specific models and the pre-trained model. TIES-Merging [55] utilizes parameter pruning and sign election to generate task-specific masks to adjust the parameter importance, while DARE [58] employs random dropping to reduce redundancy in task vectors. Additionally, methods such as RegMean [19] and Fisher-Merging [30] leverage statistical information to determine optimal static weights. More recently, TSV-M [13] addresses task interference by applying whitening transformations to task-specific singular vectors, ensuring balanced joint performance. Similarly, Iso-C [29] aims to equalize the task arithmetic spectrum by scaling singular values toward their average to promote a more isotropic and balanced representation. Alongside these approaches, recent works have explored parameter-space subspaces to refine task vectors. DOGE [51] preserves shared knowledge by constraining gradient updates orthogonal to a shared parameter subspace, while Subspace Boosting [41] mitigates rank collapse by explicitly amplifying smaller singular values of task vectors to recover task-specific information. Alternatively, parameter interference can also be mitigated directly during the fine-tuning stage prior to merging [18, 25, 34]. However, these works are limited by their focus on a single static merged model, which frequently falls short of the performance achieved by individual experts.

Moving beyond static approaches, dynamic model merging has been introduced, an approach characterized by its adjustment of parameter importance based on test samples [13, 16, 28, 33, 45, 49, 57]. TWIN-Merging [28] employs a router to determine parameter importance for a shared expert, but this mechanism introduces additional training overhead for the router, limiting its efficiency. DaWin [33] calculates parameter importance using the Shannon entropy of the pre-trained and task-specific models, it faces scalability issues due to the requirement of multiple forward passes for entropy calculation. WEMoE [45] utilizes mixture-of-experts module to combine shared and task-specific experts per input, alleviating parameter interference, however at the cost of training both the router and additional experts and increasing the memory footprint. MoW-Merging [57] introduces a lightweight gating network to produce sample-wise task probabilities used as merging coefficients. However, it still requires additional training for gating network. Apart from these architectural or training-based strategies, other efforts [13, 16, 49] have explored storing task expert knowledge more efficiently. However, these methods often necessitate access to the task identity of each input sample during inference, which restricts their applicability in real-world scenarios.

To address these limitations, we introduce SiM. Unlike previous subspace-based methods [41,51] that exploits parameter-space subspaces to align or reweight task-vector directions for merging, SiM constructs feature-space task subspaces for input-wise task classification and routing via projection residuals. While ReTeX [20] successfully repurposes this feature-space identification to train a pa-

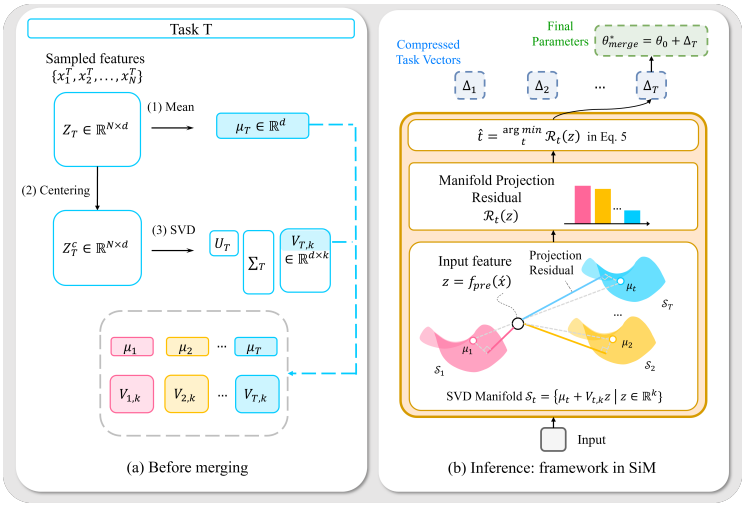

Fig. 2: Overview of SiM. The figure provides a detailed illustration of the SiM framework, which outlines the internal processes within the gray dashed border box shown in Fig. 1 (c). Our proposed method SiM performs task classification based on projection residual to predict task ID and selects the corresponding compressed task vector for the predicted task ID.

rameter recovery module, further validating the broad flexibility of this strategy, SiM applies these subspaces directly to dynamic model merging. Consequently, our approach operates without the need for router training and does not require access to task identities during inference. Fig. 1 highlights the inherent efficiency of our proposed SiM when contrasted with dynamic model merging approaches. Furthermore, a detailed overview of the SiM framework is illustrated in Fig. 2.

3 Background

Problem setting. Under pretraining-finetuning paradigm, Let $f : \mathcal{X} \times \Theta \to \mathcal{Y}$ be a pre-trained model with its parameters $\boldsymbol{\theta}_0 \in \Theta$. For every downstream task $t \in \{1, ..., T\}$, $f_{\boldsymbol{\theta}_0}$ is fine-tuned on the task-specific dataset $\mathcal{D}^{(t)} = \{(\boldsymbol{x}_i^{(t)}, y_i^{(t)})\}_{i=1}^{N_t}$, consisting of input samples $\boldsymbol{x}_i^{(t)} \in X^{(t)} \subseteq \mathcal{X}$ with the corresponding labels $y_i^{(t)} \in Y^{(t)} \subseteq \mathcal{Y}$. Several early works [17,30] have focused on consolidating the knowledge of fine-tuned task experts into a single merged model via weighted combination: $\boldsymbol{\theta} = \sum_{t=1}^{T} \lambda_t \boldsymbol{\theta}_t$. However, a resulting single model often fails to match the performance of each expert, due to parameter interference between experts during merging process.

Dynamic model merging. To address parameter interference, dynamic model merging methods have focused on input-adaptive combination [28, 33, 45, 57]:

$$\boldsymbol{\theta} = \sum_{t=1}^{T} \lambda_t(\boldsymbol{x})\boldsymbol{\theta}_t, \quad \text{or} \quad \boldsymbol{\theta} = \boldsymbol{\theta}_0 + \sum_{t=1}^{T} \lambda_t(\boldsymbol{x})\boldsymbol{\tau}_t, \tag{1}$$

where $\boldsymbol{\tau}_t$ is a task vector [17] that represents task-specific knowledge, isolated from prior knowledge by subtracting the pre-trained model parameters from each task-expert parameters: $\boldsymbol{\tau}_t = \boldsymbol{\theta}_t - \boldsymbol{\theta}_0$. However, several methods incur significant memory overhead because all full-scale expert parameters must be concurrently retained in memory during inference [33, 45, 57]. Furthermore, other dynamic model merging methods often rely on a routing mechanism that necessitates additional training with large labeled datasets [28, 45, 57].

4 Proposed method

4.1 Motivation

In this work, we draw inspiration from classical subspace identification [24] to address task classification challenges within the context of model merging. This approach is predicated on the inherent structure of the feature space, where individual tasks constitute distinct and separable manifolds. Specifically, due to inherent dataset biases, latent representations often retain dataset-specific signals, enabling the identification of the source dataset of a sample from its features [27, 46]. Building upon this, we further show that feature distributions extracted from a pre-trained model are well-separated according to tasks, as illustrated in Fig. 3.

Furthermore, previous works [2, 12] have demonstrated that features extracted from deep neural networks inherently exhibit low-rank manifold in nature. To further support this observation, we conduct spectral analysis on task-specific manifold to quantify their low-rank structure. Specifically, singular values $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_d$ are extracted via Singular Value Decomposition (SVD) of task-specific feature matrix (detailed in Sec. 4.2). We then define the cumulative energy ratio as ρ_k, which represents the proportion of the total Frobenius energy captured by the top-k principal components, as follows:

$$\rho_k = \frac{\sum_{i=1}^{k} \sigma_i^2}{\sum_{j=1}^{d} \sigma_j^2}, \tag{2}$$

where d denotes the feature dimension. As illustrated in Fig. 4, we observe that the cumulative energy ratio ρ_k reveals that the top 10% of the singular values already capture more than 80% of the Frobenius energy across tasks. This high concentration of energy indicates that task-specific features effectively reside on low-dimensional subspaces rather than being uniformly distributed. Consequently, this observation demonstrates the feasibility of a subspace-based approach for task classification.

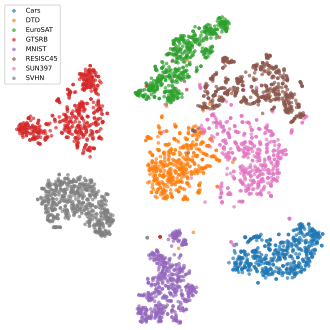

Fig. 3: Task-specific feature distributions from the CLIP ViT-B/32 across 8 vision tasks

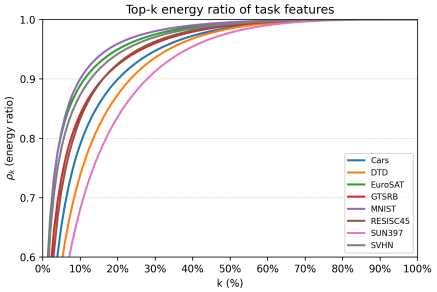

Fig. 4: Top–k energy ratio of feature representations using the CLIP ViT-B/32 on 8 vision tasks

4.2 Task manifold construction

To characterize the distribution for each task, we construct a task-specific subspace using representations extracted from the pre-trained model. For each task $t \in \{1, ..., T\}$, we first compute the empirical mean $\boldsymbol{\mu}_t \in \mathbb{R}^d$, using N randomly gathered samples $\{\boldsymbol{x}_1^{(t)}, \boldsymbol{x}_2^{(t)}, \ldots, \boldsymbol{x}_N^{(t)}\} \subset \mathcal{X}$ of the training set for task t:

$$\boldsymbol{\mu}_t = \frac{1}{N} \sum_{i=1}^{N} f_{\text{pre}}(\boldsymbol{x}_i^{(t)}). \tag{3}$$

To identify the principal structure of the task, we construct a centered feature matrix $\mathbf{Z}_t^c \in \mathbb{R}^{N \times d}$ by subtracting the empirical mean $\boldsymbol{\mu}_t$ from each extracted feature vector $\boldsymbol{z}_t \in \mathbb{R}^d$. We then perform Singular Value Decomposition (SVD) on the centered matrix $\mathbf{Z}_t^c$ to obtain the right-singular vectors:

$$\mathbf{Z}_t^c = \mathbf{U}_t \mathbf{\Sigma}_t \mathbf{V}_t^\top, \tag{4}$$

where the columns of $\mathbf{V}_t \in \mathbb{R}^{d \times d}$ represent the principal components of the feature distribution for task t. Then, we retain the top-k basis vectors to construct the subspace matrix $\mathbf{V}_{t,k} \in \mathbb{R}^{d \times k}$ with $k \ll d$. Consequently, for each task t, we store only the mean $\boldsymbol{\mu}_t$ and the subspace matrix $\mathbf{V}_{t,k}$ as the essential task signatures required for the subsequent inference stage. Notably, task manifolds can be pre-computed from a pre-trained backbone using a small set per task, eliminating the need for additional training or data storage during the merging process. Unless otherwise indicated, a sample size of $N = 32$ and k is set to 10% of the feature dimension. We provide extensive ablation studies justifying these choices, demonstrating robustness of SiM to varying rank k, sample sizes N, and even support-set selection strategies in Appendix Secs. E.1 to E.3.

4.3 Inference

Task classification via manifold projection. Given a test sample $\hat{\boldsymbol{x}} \in \mathcal{D}_{\text{test}} \subset \mathcal{X}$ and its feature vector $\boldsymbol{z} = f_{\text{pre}}(\hat{\boldsymbol{x}}) \in \mathbb{R}^d$, we define the manifold projection residual of $\boldsymbol{z}$ with respect to task t as follows:

$$\mathcal{R}_t(\mathbf{z}) = ||(\mathbf{I} - \mathbf{V}_{t,k}\mathbf{V}_{t,k}^\top)(\boldsymbol{z} - \boldsymbol{\mu}_t)||_2, \tag{5}$$

where $\mathbf{I} \in \mathbb{R}^{d\times d}$ denotes the identity matrix, $\boldsymbol{\mu}_t$ and $\mathbf{V}_{t,k}$ are the empirical mean and the basis matrix of the subspace for task t, respectively. The predicted task ID $\hat{t}$ for each sample $\hat{\boldsymbol{x}}$ is determined by selecting the task that has the smallest projection residual:

$$\hat{t} = \arg\min_t \mathcal{R}_t(\boldsymbol{z}). \tag{6}$$

Intuitively, each sample is assigned to the task whose manifold aligns with the sample most closely in the feature space.

Group batch inference. During inference, we process a batch of B samples to leverage computational efficiency. After determining the predicted task IDs $\hat{t}$ for all samples in the batch, we group them according to their predicted tasks. For each task, we select the corresponding samples $\hat{\boldsymbol{x}}_{\hat{t}}$ and task vectors $\boldsymbol{\tau}_{\hat{t}}$. Then, batch inference is conducted on the selected samples with a resulting merged parameters $\boldsymbol{\theta}^\star_{\text{merge}} = \boldsymbol{\theta}_0 + \boldsymbol{\tau}_{\hat{t}}$.

To achieve strong performance and efficiency, we integrate our method SiM with several subspace-based model merging methods that replace task vectors $\boldsymbol{\tau}_t$ with efficiently compressed task vectors $\boldsymbol{\Delta}_{\text{t}}$.

EMR-Merging (EMR) [16]: EMR first aggregates task vectors into a single pre-merged vector $\boldsymbol{\tau}_{\text{merge}}$ using a sign-and-magnitude rule. Then, for each task t, it derives a task-specific binary mask $\mathcal{M}_t = \mathbb{1}\{\boldsymbol{\tau}_t \odot \boldsymbol{\tau}_{\text{merge}} > 0\}$ and a rescaling factor λ_t. The masked task vector for task t is obtained as $\boldsymbol{\Delta}_{\text{t}} = \lambda_t\,(\mathcal{M}_t \odot \boldsymbol{\tau}_{\text{merge}})$.

TALL-Mask + Task Arithmetic (TM-TA) [49]: This method first computes a merged task vector $\boldsymbol{\tau}_{\text{merge}} = \alpha \sum_{t=1}^{T} \boldsymbol{\tau}_t$ using Task Arithmetic [17]. Then, for each task t, it computes a binary mask $\mathcal{M}_t = \mathbb{1}\{|\boldsymbol{\tau}_t| \geq |\boldsymbol{\tau}_{\text{merge}} - \boldsymbol{\tau}_t| \cdot \lambda_t\}$. The hyperparameter λ_t determines how much information to extract from the merged task vector. The masked task vector update for task t is $\boldsymbol{\Delta}_{\text{t}} = \mathcal{M}_t \odot \boldsymbol{\tau}_{\text{merge}}$.

TALL-Mask + TIES-Merging (TM-TIES) [49]: This method is one of the variants of TALL-Mask. Instead of computing the merged task vector $\boldsymbol{\tau}_{\text{merge}}$ using Task Arithmetic, this variant leverages TIES-Merging for its calculation [55]. The remaining steps are identical to those previously explained.

TSV-C [13]: TSV-C represents task vectors as layer-wise task matrices and applies singular value decomposition (SVD). For each task t, only the most top-k significant singular vectors are retained, yielding a compressed task vector $\boldsymbol{\Delta}_{\text{t}} = \sum_{i=1}^{k} \mathbf{U}_{t,i}\boldsymbol{\Sigma}_{t,i}\mathbf{V}_{t,i}^\top$.

The plug-and-play design nature of SiM allows us to empower efficient subspace-based model merging methods with the ability to perform dynamic merging for task-agnostic scenarios, where task ID of each input is unknown. Furthermore, integration with subspace-based model merging methods creates synergy

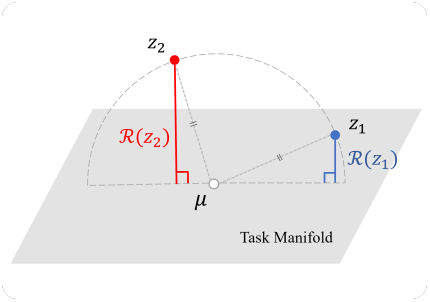

Fig. 5: Manifold projection residual vs. Euclidean distance. Two features $\boldsymbol{z}_1$ and $\boldsymbol{z}_2$ have the same Euclidean distance to the task mean $\boldsymbol{\mu}$, but their projection residuals differ depending on their alignment with the task manifold.

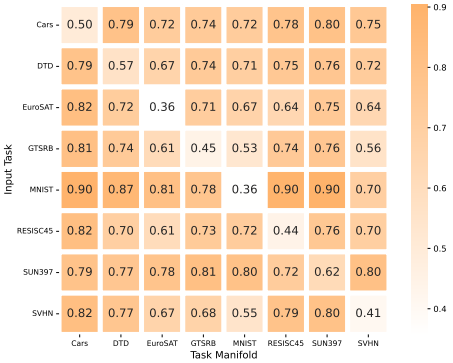

Fig. 6: Average residual ratio $\lambda_{z,t}$ across tasks and subspaces on 8 vision tasks using the CLIP ViT-B/32

as their compressed task vectors enable memory-efficient inference. A detailed description of the complete SiM procedure, along with a detailed comparison of ephemeral data usage of SiM against standard routing pipelines, is provided in Appendix Sec. B.

4.4 Analysis of projection residual

In this section, we justify the manifold projection residual formulation for task classification, in contrast to simple distance-based metric, such as Euclidean distance. Although task feature distributions exhibit inherent separability (shown in Fig. 3), simple distance-based metrics (e.g., Euclidean distance) provide only a coarse signal that neglects the intrinsic low-dimensional structure of each task. We therefore investigate how the projection residual operator $(\mathbf{I} - \mathbf{V}_{t,k}\mathbf{V}_{t,k}^{\top})$ can leverage this geometric orientation to significantly enhance feature discriminability, enabling more robust task classification.

The projection residual operator removes the component of the mean-centered feature aligned with the task subspace and retains only the orthogonal component. Consequently, even when two features have same Euclidean distances to the task mean, the residual magnitude is small if the feature direction aligns with the task subspace and large otherwise, as illustrated in Fig. 5.

To further validate the manifold projection residual formulation for task classification, we analyze how well the projection residual leads to the selection of appropriate task. To quantify the relative residual magnitude induced by each task-specific subspace in a scale-invariant manner, we define the residual ratio $\lambda_{z,t}$ as follows:

$$\lambda_{z,t} = \left\| (\mathbf{I} - \mathbf{V}_{t,k}\mathbf{V}_{t,k}^{\top}) \frac{\boldsymbol{z} - \boldsymbol{\mu}_t}{\|\boldsymbol{z} - \boldsymbol{\mu}_t\|_2} \right\|_2 . \tag{7}$$

Table 2: Multi-task performance of merged CLIP ViT models for computer vision tasks across different number of tasks. We report experimental results for ViT-B/32, ViT-B/16, and ViT-L/14 on 8, 14, and 20 vision tasks. Bold values represent the best performance among all methods, excluding the individual task-specific baselines.

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
Fine-tuned	92.8	90.9	91.3	94.7	92.8	92.8	95.9	94.3	94.8
(Static model merging)									
Task Arithmetic [17]	70.8	65.3	60.5	75.4	70.5	65.8	84.9	79.4	74.0
TIES-Merging [55]	75.1	68.0	63.4	79.7	73.2	68.2	86.9	79.5	75.7
TSV-M [13]	85.7	80.1	77.1	89.0	84.6	80.6	93.0	89.2	87.7
Iso-C [29]	84.7	79.9	74.6	89.2	84.5	79.1	93.3	89.4	87.7
(Dynamic model merging)									
TWIN-Merging [28]	84.0	70.0	57.5	91.4	78.4	63.1	93.7	86.2	74.8
DaWin [33]	89.0	73.8	52.8	87.1	77.8	62.8	91.6	82.6	77.5
WEMoE [45]	90.4	83.1	74.4	93.1	84.0	76.6	94.8	87.0	75.7
MoW-Merging [57]	88.1	83.2	79.3	93.7	79.3	78.2	94.9	78.8	81.8
EMR [16] + SiM (Ours)	90.7	87.0	86.1	92.9	90.0	89.1	95.0	92.5	92.2
TM-TA [49] + SiM (Ours)	92.3	**89.5**	**89.6**	94.0	91.2	**91.5**	92.0	89.1	89.6
TM-TIES [49] + SiM (Ours)	**92.7**	88.5	82.2	**94.3**	90.5	84.5	95.0	91.8	91.4
TSV-C [13] + SiM (Ours)	92.1	89.4	88.6	94.0	**91.5**	91.1	**95.4**	**93.6**	**92.8**

A smaller $\lambda_{z,t}$ indicates stronger alignment between the input and the subspace of task t, while a larger value indicates weaker alignment. Comparing $\lambda_{z,t}$ across subspaces therefore yields a discriminative gap in residual magnitudes. Fig. 6 illustrates the average $\lambda_{z,t}$ computed for each input-task/subspace pair across 8 vision tasks. The result shows that the diagonal entries are consistently the lowest, indicating that inputs from each task are most aligned with their own task subspace. As a result, for a given input, the projection residual operator yields a much smaller residual under its own task subspace and larger residuals under other task subspaces. This gap in residual ratio indicates that task-specific subspaces provide a discriminative signal, improving separability in residual space and facilitating reliable task identification. This corresponds to the superior task classification performance (more details in Sec. 5.4), validating the effectiveness of our proposed method SiM.

5 Experiments

5.1 Setup

Baselines. We compared applying our proposed method to existing dynamic model merging methods against several baselines such as EMR-Merging [16], TALL-Mask + Task Arithmetic, TALL-Mask + TIES [17, 49, 55], and TSV-C [13]. Furthermore, we compare against other dynamic model merging methods

Table 3: Multi-task performance of merged T5-large across 7 NLP tasks. Bold values represent the best performance among all methods, excluding the individual task-specific baselines. Underlines indicate the case where fine-tuning performance is surpassed.

Method	PAWS	QASC	QuaRTz	StoryCloze	WikiQA	Winogrande	WSC	Avg.
Fine-tuned	94.4	98.9	87.8	90.8	96.0	74.7	79.2	88.8
(Static model merging)								
Task Arithmetic [17]	77.8	96.0	78.6	86.4	59.1	62.3	52.8	73.3
TIES-Merging [55]	81.5	96.2	80.1	83.6	64.9	66.5	65.3	76.9
TSV-M [13]	85.0	92.4	66.0	81.1	88.4	87.0	57.0	79.6
Iso-C [29]	72.9	94.8	50.5	80.7	92.5	75.7	54.0	74.4
(Dynamic model merging)								
TWIN-Merging [28]	93.2	96.0	**87.1**	85.6	77.1	70.8	**69.7**	82.8
DaWin [33]	85.6	96.1	81.4	73.2	70.2	68.7	57.9	76.2
WEMoE [45]	93.4	95.4	74.5	79.6	94.1	90.4	57.0	83.4
MoW-Merging [57]	94.7	90.2	79.0	83.1	56.6	**93.8**	51.0	78.3
EMR [16] + SiM	95.0	95.8	83.5	87.8	92.4	80.6	59.0	84.9
TM-TA [49] + SiM	88.3	95.6	81.5	88.0	**95.2**	87.2	53.0	84.1
TM-TIES [49] + SiM	**95.5**	97.0	83.5	89.2	94.2	91.1	49.0	**85.6**
TSV-C [13] + SiM	95.0	**97.6**	84.5	**89.7**	94.4	74.5	55.0	84.4

that require additional training, including TWIN-Merging [28], DaWin [33], WEMoE [45] and MoW-Merging [57] as well as representative static model merging baselines [13, 17, 29, 55].

Evaluation settings. For evaluation, we follow the settings of TALL-Mask [49] for the computer vision tasks, and the settings of TIES-Merging [55] for the 7 NLP tasks. We evaluate our method across four task scenarios. The 8 computer vision task scenario consists of: (1) SUN397 [54], (2) Cars [22], (3) RESISC45 [5], (4) EuroSAT [15], (5) SVHN [31], (6) GTSRB [43], (7) MNIST [10], and (8) DTD [6]. The 14 computer vision task scenario adds: (9) CIFAR100 [23], (10) STL10 [8], (11) Flowers102 [32], (12) OxfordIIITPet [35], (13) PCAM [48], and (14) FER2013 [14]. The 20 computer vision task scenario further includes: (15) EMNIST [9], (16) CIFAR10 [23], (17) Food101 [3], (18) FashionMNIST [53], (19) RenderedSST2 [37, 42], and (20) KMNIST [7].

The 7 NLP task scenario comprises: (1) QASC [21], (2) WikiQA [56], (3) QuaRTz [44] for question answering, (4) PAWS [59] for paraphrase identification, (5) Story Cloze [40] for sentence completion, (6) Winogrande [39], (7) WSC [26] for coreference resolution. We perform merging on CLIP [36] ViT-{B/32, B/16, L/14} [11] for the computer vision tasks and on T5-large [38] for the NLP tasks. The model checkpoints used in our experiments are obtained from the following sources: TALL-Mask [49] (for the computer vision tasks) and TIES-Merging [55] (for the 7 NLP tasks). Further experimental details are provided in Appendix Sec. C.

Table 4: Multi-task performance on seen versus unseen tasks under distribution shift. The "seen" split corresponds to the 8-task vision benchmark used in the main experiments, whereas the "unseen" split consists of the additional 6 datasets from the 14-task benchmark that are not included in the 8-task set. Averaged performance across all 14 tasks is reported in the last column.

Method	Seen (8 tasks)	Unseen (6 tasks)	Overall (14 tasks)
(Static model merging)			
Task Arithmetic [17]	70.8	44.3	59.4
TIES-Merging [55]	75.1	55.2	66.6
TSV-M [13]	85.7	60.5	74.9
Iso-C [29]	84.7	60.2	74.2
(Dynamic model merging)			
TWIN-Merging [28]	84.0	62.6	74.8
DaWin [33]	89.0	57.5	75.5
WEMoE [45]	90.4	59.2	77.0
MoW-Merging [57]	88.1	59.0	75.6
EMR [16] + SiM (Ours)	90.7	57.9	76.6
TM-TA [49] + SiM (Ours)	92.3	62.6	79.6
TM-TIES [49] + SiM (Ours)	**92.7**	**63.3**	**80.1**
TSV-C [13] + SiM (Ours)	92.1	58.0	77.5

5.2 Main results

Computer vision tasks. Experimental results for merging ViT-B/32, ViT-B/16 and ViT-L/14 across the 8, 14, and 20 vision tasks are summarized in Tab. 2. These results demonstrate that our approach, integrating with subspace-based merging methods, consistently surpasses existing model merging methods across various model sizes without requiring prior knowledge of task IDs or additional training. We also investigate the scalability of our method by evaluating its performance on task sets comprising more than 14 tasks. Our method consistently maintains the performance, highlighting the significance of our proposed methodology. For more extreme scenarios, we provide further results in Appendix Secs. D.1 and D.2 demonstrating the robustness of SiM on larger scales (up to 50 tasks) and highly ambiguous fine-grained domains (e.g., CUB-200).

NLP tasks. We further assess the effectiveness of our method by evaluating its performance on NLP tasks, in addition to the vision tasks. The corresponding experimental results for the NLP tasks are shown in Tab. 3. Our method demonstrates near fine-tuning performance even on NLP tasks. Notably, it surpasses fine-tuning performance on PAWS and Winogrande. This suggests that SiM is capable of identifying a superior model compared to the original expert model.

Robustness to unseen tasks. We evaluate task classification on seen and unseen tasks to test robustness to distributional shifts. Concretely, we treat the 8-task vision benchmark used in the main experiments as the set of *seen* tasks, and use the additional 6 datasets from the 14-task vision benchmark as *unseen* tasks. For each input, SiM identifies the task whose subspace yields the minimum projection residual among the seen tasks and selects the corresponding

Table 5: Inference cost with CLIP ViT-B/32. We report the computation costs of model merging methods across all tasks in the 8 computer vision task scenarios, measured on an NVIDIA GeForce RTX 3090.

Method	Batch inference	Latency (per input)	VRAM (GB)	Avg. performance
(Static model merging)				
Task-Arithmetic [17]	✓	0.0008s	1.3	70.8
TIES-Merging [55]	✓	0.0008s	1.3	75.1
TSV-M [13]	✓	0.0008s	1.3	85.7
Iso-C [29]	✓	0.0008s	1.3	84.7
(Dynamic model merging)				
TWIN-Merging [28]	×	0.03s	3.2	84.0
DaWin [33]	×	0.63s	5.5	89.0
WEMoE [45]	×	0.02s	4.3	90.4
MoW-Merging [57]	✓	0.008s	4.9	88.1
EMR [16] + SiM (Ours)	✓	0.004s	1.9	90.7
TM-TA [49] + SiM (Ours)	✓	0.004s	1.8	92.3
TM-TIES [49] + SiM (Ours)	✓	0.003s	1.8	92.7
TSV-C [13] + SiM (Ours)	✓	0.003s	1.6	92.1

Table 6: Training-free task classification performance comparison across different distance-based metrics. We compare task classification performance using Euclidean distance, cosine similarity, Mahalanobis distance, k-NN and our final model.

Method	Cars	DTD	EuroSAT	GTSRB	MNIST	RESISC45	SUN397	SVHN	**Avg.**
Euclidean dist.	99.7	96.9	86.5	90.0	100	95.7	97.8	99.6	95.8
Cosine sim.	99.7	96.5	86.5	89.7	100	95.3	97.0	99.6	95.5
Mahalanobis dist.	98.6	92.3	95.4	97.9	99.1	91.5	99.8	97.5	96.5
k-NN (k=1)	99.9	91.8	99.2	97.1	99.9	79.5	75.7	92.9	92.0
SiM	99.8	98.6	99.9	98.4	100	98.8	95.4	99.9	**99.0**

compressed task vector. As summarized in Tab. 4, among baselines, TWIN-Merging [28] improves robustness to unseen tasks, but still exhibit a noticeable drop in accuracy compared to seen tasks. In contrast, SiM obtains the best overall performance. These results indicate that incorporating the SiM enables dynamic model merging methods to handle unseen tasks effectively, thereby mitigating the impact of feature distribution shift as the task set evolves.

5.3 Computational costs

In this section, we investigate the efficiency of SiM, particularly in comparison to another dynamic model merging method: TWIN-Merging [28], DaWin [33], WEMoE [45] and MoW-Merging [57]. As shown in Tab. 5, SiM demonstrates substantially faster inference, higher memory efficiency, and overall performance compared to other dynamic approaches. Specifically, SiM is highlighted by inference times that are almost 10 times faster than sample-wise dynamic interpolation methods such as TWIN-Merging [28] and WEMoE [45]. Our approach

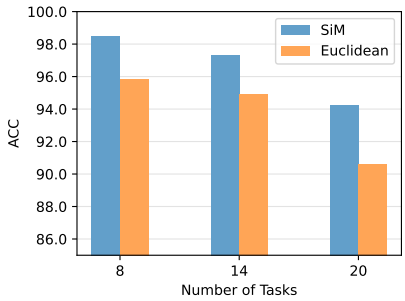

Fig. 7: Task classification performance of SiM vs Euclidean on the CLIP ViT-B/32 across 8, 14, and 20 tasks

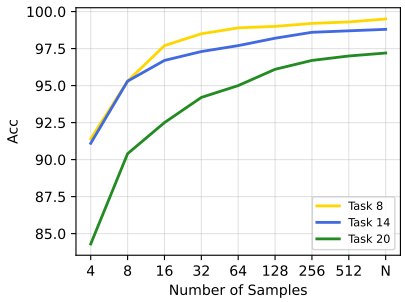

Fig. 8: Task classification accuracy using the CLIP ViT-B/32 across sample sizes

facilitates group batch processing for samples belonging to the same task ID. These results demonstrate the efficiency of our group batch processing strategy.

For the memory costs, SiM significantly reduces memory requirements. Existing dynamic merging methods face substantial memory challenges because they necessitates storing a complete task-specific model for every potential task. This results in a linear increase in memory cost with the number of merged tasks. SiM, however, combines with merging methods that leverage lightweight binary masks or compressed task experts, which store only compact task-specific updates rather than full task-specific models. This design substantially reduces memory cost, requiring memory comparable to static single-model merging methods, while achieving state-of-the-art performance that surpasses dynamic methods. Furthermore, we provide a detailed analysis comparing our hard-routing strategy against soft-routing alternatives in terms of both end-to-end latency and accuracy in Appendix Sec. E.4.

5.4 Task classification

Projection residual vs Euclidean. As discussed in Sec. 4.4, projection residuals enhance task separability by measuring alignment with respect to task-specific subspaces rather than relying solely on mean-based distances. This observation is further supported by Tab. 6, providing a comparison of training-free task classification performance across various distance-based metrics on the 8 vision task scenarios. Overall, SiM achieves superior performance across datasets, demonstrating that improved separability in residual space provides a more reliable signal for task classification. Furthermore, SiM exhibits strong scalability, maintaining robust performance even as the number of tasks increases to 20, which leads to a widening performance gap compared to the Euclidean baseline, as illustrated in Fig. 7. Notably, on EuroSAT, where the diagonal entry in Fig. 6 (i.e., the residual ratio evaluated on the corresponding task subspace)

is the smallest, SiM achieves over a 10% improvement compared to Euclidean and cosine baselines in Tab. 6. Moreover, we observe consistent gains on GTSRB and RESISC45, which also exhibit relatively small diagonal residual ratios, suggesting that smaller in-task residual ratios provide stronger separability in residual space. Furthermore, SiM outperforms the Mahalanobis distance, which accounts for covariance structure and non-parametric k-NN. These results validate the effectiveness of SiM as a training-free task classification approach and confirm that modeling task-specific subspaces yields a more discriminative and robust criterion for task classification. Notably, as detailed in Appendix Sec. E.5, even when routing errors occur, their effect on downstream performance is minimal. Replacing SiM with oracle routing yields only a 0.3–0.6% performance gap, demonstrating that SiM remains highly robust at the downstream prediction level.

The number of samples. We have investigated the impact of sample size on the performance of SiM when estimating task manifolds. As shown in Fig. 8, SiM consistently achieves superior classification performance across diverse task scales, eliminating the need for any additional training. Furthermore, SiM, even with only 32 samples per task, yields strong performance similar to that achieved using the entire training dataset. Here, N represents the total number of training samples. This advantage is consistently demonstrating its robustness and broad applicability. A detailed task classification accuracy for each individual dataset across varying numbers of support samples is provided in Appendix Sec. E.3. Additionally, as detailed in Appendix Sec. D.3, SiM can achieve near-optimal task classification using features from intermediate transformer blocks, offering a practical way to further reduce computational overhead without sacrificing performance.

6 Conclusion

In this work, we introduce SiM, a training-free framework for dynamic model merging in task-agnostic scenarios. By utilizing SVD-based manifold projections, our approach effectively identifies task IDs without any additional router training. Furthermore, by integrating with subspace-based merging, SiM eliminates the need to store full expert parameters, significantly reducing memory overhead. Our experiments across vision and NLP benchmarks demonstrate that SiM substantially improves performance and consistently narrows the gap to individual task experts in real-world settings where task information is unavailable.

Acknowledgments

This work was supported by the Institute of Information and communications Technology Planning and evaluation (IITP) grant (No. RS-2025-25422680, No. RS-2020-II201373, Artificial Intelligence Graduate School Program (Hanyang University)) and the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (No. RS-2025-24533064).

References

1. Achiam, J., Adler, S., Agarwal, S., Ahmad, L., Akkaya, I., Aleman, F.L., Almeida, D., Altenschmidt, J., Altman, S., Anadkat, S., et al.: Gpt-4 technical report. arXiv preprint arXiv:2303.08774 (2023)
2. Ansuini, A., Laio, A., Macke, J.H., Zoccolan, D.: Intrinsic dimension of data representations in deep neural networks. In: NeurIPS (2019)
3. Bossard, L., Guillaumin, M., Gool, L.V.: Food-101 – mining discriminative components with random forests. In: ECCV (2014)
4. Brown, T.B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., Agarwal, S., Herbert-Voss, A., Krueger, G., Henighan, T., Child, R., Ramesh, A., Ziegler, D.M., Wu, J., Winter, C., Hesse, C., Chen, M., Sigler, E., Litwin, M., Gray, S., Chess, B., Clark, J., Berner, C., McCandlish, S., Radford, A., Sutskever, I., Amodei, D.: Language models are few-shot learners. In: NeurIPS (2020)
5. Cheng, G., Han, J., Lu, X.: Remote sensing image scene classification: Benchmark and state of the art. Proceedings of the IEEE (2017)
6. Cimpoi, M., Maji, S., Kokkinos, I., Mohamed, S., Vedaldi, A.: Describing textures in the wild. In: CVPR (2014)
7. Clanuwat, T., Bober-Irizar, M., Kitamoto, A., Lamb, A., Yamamoto, K., Ha, D.: Deep learning for classical japanese literature. arXiv preprint arXiv:1812.01718 (2018)
8. Coates, A., Ng, A., Lee, H.: An analysis of single-layer networks in unsupervised feature learning. In: AISTATS (2011)
9. Cohen, G., Afshar, S., Tapson, J., Schaik, A.v.: Emnist: Extending mnist to handwritten letters. In: IJCNN (2017)
10. Deng, L.: The mnist database of handwritten digit images for machine learning research [best of the web]. IEEE Signal Processing Magazine (SPM) (2012)
11. Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., Houlsby, N.: An image is worth 16x16 words: Transformers for image recognition at scale. In: ICLR (2021)
12. Feng, R., Zheng, K., Huang, Y., Zhao, D., Jordan, M., Zha, Z.J.: Rank diminishing in deep neural networks. In: NeurIPS (2022)
13. Gargiulo, A.A., Crisostomi, D., Bucarelli, M.S., Scardapane, S., Silvestri, F., Rodolà, E.: Task singular vectors: Reducing task interference in model merging. arXiv preprint arXiv:2412.00081 (2024)
14. Goodfellow, I.J., Erhan, D., Carrier, P.L., Courville, A., Mirza, M., Hamner, B., Cukierski, W., Tang, Y., Thaler, D., Lee, D.H., Zhou, Y., Ramaiah, C., Feng, F., Li, R., Wang, X., Athanasakis, D., Shawe-Taylor, J., Milakov, M., Park, J., Ionescu, R., Popescu, M., Grozea, C., Bergstra, J., Xie, J., Romaszko, L., Xu, B., Chuang, Z., Bengio, Y.: Challenges in representation learning: A report on three machine learning contests. In: ICONIP (2013)
15. Helber, P., Bischke, B., Dengel, A., Borth, D.: Eurosat: A novel dataset and deep learning benchmark for land use and land cover classification. IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing (2019)
16. Huang, C., Ye, P., Chen, T., He, T., Yue, X., Ouyang, W.: Emr-merging: Tuning-free high-performance model merging. In: NeurIPS (2024)
17. Ilharco, G., Tulio Ribeiro, M., Wortsman, M., Gururangan, S., Schmidt, L., Hajishirzi, H., Farhadi, A.: Editing models with task arithmetic. In: ICLR (2023)

18. Iurada, L., Ciccone, M., Tommasi, T.: Efficient model editing with task-localized sparse fine-tuning. In: ICLR (2025)
19. Jin, X., Ren, X., Preotiuc-Pietro, D., Cheng, P.: Dataless knowledge fusion by merging weights of language models. In: ICLR (2023)
20. Jung, J., Kim, T., Jo, K., Baik, S.: Learning to recover task experts from a multi-task merged model. arXiv preprint arXiv:2606.26902 (2026)
21. Khot, T., Clark, P., Guerquin, M., Jansen, P., Sabharwal, A.: Qasc: A dataset for question answering via sentence composition. In: AAAI (2020)
22. Krause, J., Stark, M., Deng, J., Fei-Fei, L.: 3d object representations for fine-grained categorization. In: ICCV Workshop (2013)
23. Krizhevsky, A., Hinton, G., et al.: Learning multiple layers of features from tiny images (2009)
24. Lee, K.C., Ho, J., Kriegman, D.J.: Acquiring linear subspaces for face recognition under variable lighting. IEEE Transactions on Pattern Analysis and Machine Intelligence (2005)
25. Lee, Y., Jung, J., Baik, S.: Mitigating parameter interference in model merging via sharpness-aware fine-tuning. In: ICLR (2025)
26. Levesque, H., Davis, E., Morgenstern, L.: The winograd schema challenge. In: KR (2012)
27. Liu, Z., He, K.: A decade's battle on dataset bias: Are we there yet? In: ICLR (2025)
28. Lu, Z., Fan, C., Wei, W., Qu, X., Chen, D., Cheng, Y.: Twin-merging: Dynamic integration of modular expertise in model merging. In: NeurIPS (2024)
29. Marczak, D., Magistri, S., Cygert, S., Twardowski, B., Bagdanov, A.D., van de Weijer, J.: No task left behind: Isotropic model merging with common and task-specific subspaces. In: ICML (2025)
30. Matena, M., Raffel, C.: Merging models with fisher-weighted averaging. In: NeurIPS (2022)
31. Netzer, Y., Coates, A., Wu, B., Wang, T., Ng, A.Y.: Reading digits in natural images with unsupervised feature learning. In: NIPS Workshop (2011)
32. Nilsback, M.E., Zisserman, A.: Automated flower classification over a large number of classes. In: ICVGIP (2008)
33. Oh, C., Li, Y., Song, K., Yun, S., Han, D.: Dawin: Training-free dynamic weight interpolation for robust adaptation. In: ICLR (2025)
34. Ortiz-Jimenez, G., Favero, A., Frossard, P.: Task arithmetic in the tangent space: Improved editing of pre-trained models. In: NeurIPS (2023)
35. Parkhi, O.M., Vedaldi, A., Zisserman, A., Jawahar, C.V.: Cats and dogs. In: CVPR (2012)
36. Radford, A., Kim, J.W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., Krueger, G., Sutskever, I.: Learning transferable visual models from natural language supervision. In: ICML (2021)
37. Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., Sutskever, I.: Language models are unsupervised multitask learners (2019)
38. Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W., Liu, P.: Exploring the limits of transfer learning with a unified text-to-text transformer. In: JMLR (2020)
39. Sakaguchi, K., Bras, R.L., Bhagavatula, C., Choi, Y.: Winogrande: An adversarial winograd schema challenge at scale. In: AAAI (2020)
40. Sharma, R., Allen, J., Bakhshandeh, O., Mostafazadeh, N.: Tackling the story ending biases in the story cloze test. In: ACL (2018)

41. Skorobogat, R., Roth, K., Georgescu, M.I.: Subspace-boosted model merging. arXiv preprint arXiv:2506.16506 (2025)
42. Socher, R., Perelygin, A., Wu, J., Chuang, J., Manning, C.D., Ng, A., Potts, C.: Recursive deep models for semantic compositionality over a sentiment treebank. In: EMNLP (2013)
43. Stallkamp, J., Schlipsing, M., Salmen, J., Igel, C.: The german traffic sign recognition benchmark: a multi-class classification competition. In: IJCNN (2011)
44. Tafjord, O., Gardner, M., Lin, K., Clark, P.: Quartz: An open-domain dataset of qualitative relationship questions. In: EMNLP (2019)
45. Tang, A., Shen, L., Luo, Y., Yin, N., Zhang, L., Tao, D.: Merging multi-task models via weight-ensembling mixture of experts. In: ICML (2024)
46. Torralba, A., Efros, A.A.: Unbiased look at dataset bias. In: CVPR (2011)
47. Touvron, H., Lavril, T., Izacard, G., Martinet, X., Lachaux, M.A., Lacroix, T., Rozière, B., Goyal, N., Hambro, E., Azhar, F., Rodriguez, A., Joulin, A., Grave, E., Lample, G.: Llama: Open and efficient foundation language models. arXiv preprint arXiv:2302.13971 (2023)
48. Veeling, B.S., Linmans, J., Winkens, J., Cohen, T., Welling, M.: Rotation equivariant cnns for digital pathology. In: MICCAI (2018)
49. Wang, K., Dimitriadis, N., Ortiz-Jimenez, G., Fleuret, F., Frossard, P.: Localizing task information for improved model merging and compression. In: ICML (2024)
50. Wei, J., Bosma, M., Zhao, V.Y., Guu, K., Yu, A.W., Lester, B., Du, N., Dai, A.M., Le, Q.V.: Finetuned language models are zero-shot learners. In: ICLR (2022)
51. Wei, Y., Tang, A., Shen, L., Hu, Z., Yuan, C., Cao, X.: Modeling multi-task model merging as adaptive projective gradient descent. In: ICML (2025)
52. Wortsman, M., Ilharco, G., Kim, J.W., Li, M., Kornblith, S., Roelofs, R., Gontijo-Lopes, R., Hajishirzi, H., Farhadi, A., Namkoong, H., Schmidt, L.: Robust fine-tuning of zero-shot models. In: CVPR (2022)
53. Xiao, H., Rasul, K., Vollgraf, R.: Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. arXiv preprint arXiv:1708.07747 (2017)
54. Xiao, J., Ehinger, K.A., Hays, J., Torralba, A., Oliva, A.: Sun database: Exploring a large collection of scene categories. International Journal of Computer Vision (IJCV) (2016)
55. Yadav, P., Tam, D., Choshen, L., Raffel, C., Bansal, M.: Ties-merging: Resolving interference when merging models. In: NeurIPS (2023)
56. Yang, Y., tau Yih, W., Meek, C.: Wikiqa: A challenge dataset for open-domain question answering. In: ACL (2015)
57. Ye, P., Huang, C., Shen, M., Chen, T., Huang, Y., Ouyang, W.: Dynamic model merging with mixture of weights. IEEE Transactions on Circuits and Systems for Video Technology (2025)
58. Yu, L., Yu, B., Yu, H., Huang, F., Li, Y.: Language models are super mario: Absorbing abilities from homologous models as a free lunch. In: ICML (2024)
59. Zhang, Y., Baldridge, J., He, L.: Paws: Paraphrase adversaries from word scrambling. In: NAACL (2019)