Navigating the Accuracy-Size Trade-Off with Flexible Model Merging

Anonymous authors
Paper under double-blind review

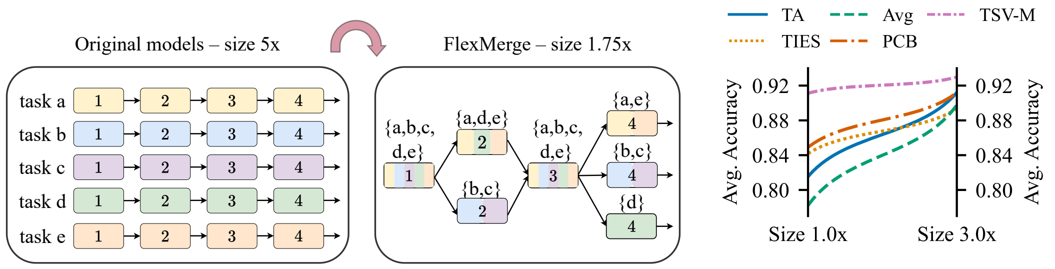
Abstract

Model merging has emerged as an efficient method to combine multiple single-task fine-tuned models. The merged model can enjoy multi-task capabilities without expensive training. While promising, merging into a single model often suffers from an accuracy gap with respect to individual fine-tuned models. On the other hand, deploying all individual fine-tuned models incurs high storage costs. We propose FlexMerge, a novel data-free model merging framework that: *(a)* flexibly generates merged models of varying sizes, spanning the full spectrum from a single merged model to retaining all individual fine-tuned models; and *(b)* supports multiple merging algorithms in a unified framework. Using FlexMerge, we systematically characterize the accuracy–size trade-off of different algorithms. Our study reveals two key findings: first, even modestly larger merged models can yield steep accuracy gains (up to 13.5% when just doubling the size); second, algorithm rankings are not consistent as size increases, with some methods overtaking others beyond the one-model regime. These results uncover a new design dimension for model merging: developing and comparing algorithms across the full spectrum of sizes rather than only at the single-model limit. Extensive experiments on vision and NLP benchmarks, with up to 30 tasks, confirm the generality and practicality of FlexMerge.

1 Introduction

In recent years, the pre-training followed by fine-tuning paradigm has become the leading approach in both natural language processing (NLP) and computer vision, showcasing remarkable success on a wide range of tasks (Devlin et al., 2018; Dodge et al., 2020; Dosovitskiy et al., 2021; Bommasani et al., 2021). Pre-trained models (PTMs), which learn generalized features from large-scale datasets, serve as powerful starting points, enabling fine-tuning to achieve superior performance on downstream tasks with less labeled data. This has led to an exponential growth in the number of fine-tuned models driven further by the availability of open-source repositories (maintainers & contributors, 2016; Wolf et al., 2019). However, *deploying individual fine-tuned models* for specific tasks incurs high storage and deployment costs. The alternative is Multi-task learning (MTL), which aims to jointly train *a single model* across multiple tasks (Vandenhende et al., 2021; Sanh et al., 2022). But MTL comes with its own drawbacks, such as significant computational overhead and the need to simultaneously access the data from all tasks, which might be infeasible due to privacy constraints (Jin et al., 2023).

To mitigate these limitations, model merging has emerged as a promising solution, allowing the combination of multiple fine-tuned models into a *single model* without access to training data. To this end, several model merging methods have been proposed (Gargiulo et al., 2025; Huang et al., 2024; Yang et al., 2024a; Yadav et al., 2023; Ilharco et al., 2023; Matena & Raffel, 2022). However, a single model is often unable to perfectly resolve parameter conflicts between tasks, leaving an accuracy gap with respect to the individual fine-tuned models (Zhang et al., 2025; Huang et al., 2024). This gap becomes more significant as a higher number of models are merged (Yadav et al., 2023; Ilharco et al., 2023). To mitigate this issue, some methods leverage additional data to facilitate merging (Lu et al., 2024; Yang et al., 2024a; Tang et al., 2024a; Yang et al., 2024b). Yet, the data-dependency might be difficult to meet in practice due to privacy constraints or proprietary restrictions, leading to a growing focus on data-free model merging techniques (Gargiulo et al., 2025; Huang et al., 2024; Du et al., 2024; Yu et al., 2024; Yadav et al., 2023). Nevertheless, in the absence of data, the accuracy gap remains significant, highlighting the need for novel solutions.



(a) FlexMerge illustration (b) FlexMerge on 8 tasks (ViT-L/14)

Figure 1: (a) Fine-tuned models are sequences of blocks. FlexMerge iteratively merges block pairs until reaching the desired size (*e.g.*, size 1.75×). (b) Algorithm rankings change as size is increased.

We argue that an effective solution to this challenge is to go beyond the conventional one model approach, and merge into model(s) of bigger sizes. Merging multiple fine-tuned models naturally presents a trade-off between maintaining accuracy and achieving model compactness, dictated by the size of the merged model. This trade-off spans a spectrum: at one extreme, retaining all individual fine-tuned models for each task achieves maximal accuracy but at the cost of larger overall size; at the other, fully merging all tasks into a single model minimizes storage size but sacrifices accuracy. Despite this clear trade-off, a systematic investigation of the accuracy-size relationship in model merging has been lacking. In this light, we pose two key research questions: *(RQ1) How can we derive merged models across the full range of model sizes in a data-free manner?* and *(RQ2) What is the nature of the accuracy-size trade-off exhibited by different data-free merging algorithms?*

In response to (RQ1), we propose FlexMerge, a flexible framework that enables *data-free* fusion into model(s) of any desired size. At its core, FlexMerge treats each fine-tuned model as composed of sequential blocks, as illustrated in Figure 1(a), whose granularity can be controlled (*e.g.*, a transformer block, a few layers, or even a single layer). It then takes a bottom-up approach starting with all fine-tuned models with their respective blocks and greedily merging a pair of blocks with the highest cosine similarity in each merging iteration. This merging can leverage *any* existing data-free merging method such as Task Arithmetic (TA) (Ilharco et al., 2023), TIES-Merging (Yadav et al., 2023), EMR-Merging (Huang et al., 2024), TSV-M (Gargiulo et al., 2025), *etc*., applied at the block-level. With each merging iteration, the size of the deployed model is reduced, and the process can be halted once the desired size is met. For instance, in Figure 1(a), the merging is halted when the merged model is 1.75× the size of a single fine-tuned model. The entire merging process in FlexMerge needs no additional data or tuning, making FlexMerge fully *data-free*.

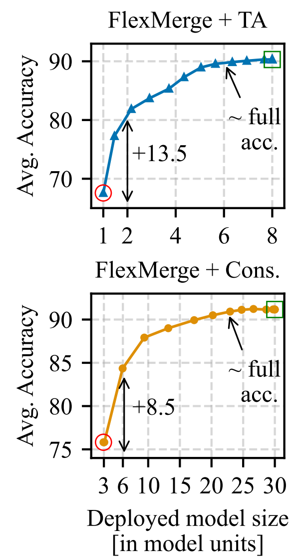


Figure 2: FlexMerge enables large accuracy gains when just doubling the deployed model size and attains full accuracy well before the maximum size.

In response to (RQ2), we demonstrate with FlexMerge that a range of data-free merging algorithms exhibit highly favorable accuracy-size trade-offs. Remarkably, the accuracy-size trade-off is characterized by steep gains in accuracy for even modestly bigger merged models beyond one model, followed by steady improvements, reaching near fine-tuning accuracy well before the maximum size. To illustrate this in practice, Figure 2 charts the merged model accuracy versus deployed size for 8 tasks (top) and 30 tasks (bottom) using the ViT-B/32 model, with TA (Ilharco et al., 2023) and Consensus (Wang et al., 2024) as the respective merging methods. ○ and □ annotate the accuracy at both ends of the spectrum *i.e.*, lowest fused size and retaining all fine-tuned models respectively. FlexMerge + TA gains 13.5% in average accuracy when going from 1× to 2× while FlexMerge + Consensus gains 8.5% when doubling the size from approximately 3× to 6×. We note that Consensus requires storing masks and the pre-trained parameters alongside the unified parameters (Wang et al., 2024), resulting in the lowest possible size of ≈ 3× for 30 tasks. We observe that the steep rise is followed by relatively slower accuracy growth in the middle. Yet, a near fine-tuning accuracy is attained well before the maximum size. For 8 tasks, this is obtained around size 6× and for 30 tasks, around size 23.5×. Secondly, we observe that algorithm rankings are not consistent even at modestly bigger sizes. Figure 1(b) shows that vanilla averaging exceeds TIES-Merging while TA attains the performance of PCB-Merging at size 3× despite starting from a large gap at 1×. *Our findings open a new design dimension: encouraging algorithm development and comparison for sizes* > 1× *instead of restricting only to* 1×.

Contributions. To the best of our knowledge, we present the first study of model merging that:

- Generates merged models across full spectrum of sizes, *including non-integer sizes*;
- Supports a wide range of data-free merging algorithms, *within a unified framework*;
- Provides a systematic characterization of the accuracy-size trade-off in data-free model merging, *revealing general trends, highly favorable regions and inconsistency of algorithm rankings*;
- Demonstrates that larger merged sizes incur negligible inference-time overhead, *enabled by our efficient implementation*.

We confirm our findings through extensive experiments spanning language and vision modalities, multiple model families, multi-modal datasets, using both full-parameter fine-tuning (FFT) and parameter efficient fine-tuning (PEFT), scaling up to 30 tasks.

2 Related Work

Initial studies on model merging focused on vanilla averaging as a way of combining models obtained from same or different training runs of a task into one higher performing model (Izmailov et al., 2018; Gupta et al., 2020; Wortsman et al., 2022; Cha et al., 2021). Vanilla averaging is also used in federated learning to merge different client models (McMahan et al., 2017; Konečný et al., 2016). Ilharco et al. (2023) introduced task vectors, representing the difference between fine-tuned and pre-trained models, enabling model combination through vector arithmetic.

Data-based merging methods leverage validation data to facilitate merging. Techniques like Fisher Merging (Matena & Raffel, 2022) and RegMean (Jin et al., 2023) compute the Fisher Information and Gram matrices, respectively, for weighted averaging of model parameters. Surgery (Yang et al., 2024a) trains task-specific adapters to debias the representations produced by the merged model. AdaMerging (Yang et al., 2024b) introduces per-task, per-layer merging co-efficients, and proposes to learn these co-efficients by solving an entropy minimization objective. WEMoE (Tang et al., 2024a) merges all modules except for task-specific MLPs, which are retained as weight-ensembled mixture-of-experts (MoE) with learned routers. Twin-Merging (Lu et al., 2024) leverages MoE on difference vectors *i.e*., the difference between the fine-tuned models and the merged model. While the availability of validation data enhances accuracy, such data might be difficult to obtain in practice.

Data-free merging directly merges model parameters without any data. TIES-Merging (Yadav et al., 2023) resolves parameter interference by trimming redundant parameters and resolving sign conflicts. PCB-Merging (Du et al., 2024) considers both intra- and inter-parameter competition balancing. DARE (Yu et al., 2024) reduces parameter interference by randomly dropping parameters and proportionally rescaling remaining ones. EMR-Merging (Huang et al., 2024) introduces the paradigm of maintaining light-weight task specific masks in addition to the merged model to enhance performance. Consensus (Wang et al., 2024) also relies on task specific masks, but creates them differently compared to EMR-Merging. Both approaches significantly improve accuracy over previous methods, albeit at the cost of test-time reconstruction overhead (Gargiulo et al., 2025). TSV-M (Gargiulo et al., 2025) merges SVD-decomposed task singular vectors, reducing interference by retaining only prominent singular directions and orthogonalizing them across tasks.

Recent work by Zhang et al. (2025) explores merging into sizes > 1×. Their method, Channel Merging, relies on layer-wise K-Means clustering followed by merging within each cluster using only TA. However, this approach is restrictive as it cannot generate fractional-sized models. Despite the emergence of advanced methods and attempts at merging into bigger sizes, to the best of our knowledge, no prior work has systematically investigated the accuracy–size trade-off in model merging under a single unified framework. For completeness, we provide additional related work and a taxonomy of existing algorithms based on their data-free/data-based nature in Appendix A.

3 FlexMerge

3.1 Preliminaries

We consider a set of M tasks: $\{T_1, \ldots, T_M\}$, where the fine-tuned model parameters for task T_i are denoted by $\boldsymbol{\theta}_i$. These fine-tuned parameters are typically obtained by adapting a pre-trained model, such as ViT (Dosovitskiy et al., 2021) or T5 (Raffel et al., 2020) using either full parameter fine-tuning (FT) or parameter-efficient fine-tuning (PEFT) methods (Liu et al., 2022). Thus, it is assumed that all the fine-tuned models have the same size and the model architecture as the pre-trained model, as also considered in prior work (Ilharco et al., 2023; Yadav et al., 2023). To analyze the changes introduced by fine-tuning, we use the concept of task vectors $\boldsymbol{\tau}_i$ introduced by Ilharco et al. (2023), where $\boldsymbol{\tau}_i = \boldsymbol{\theta}_i - \boldsymbol{\theta}_{\text{pre}}$, with $\boldsymbol{\theta}_{\text{pre}}$ being the pre-trained weights. These task vectors capture the specific modifications needed for each task and provide a compact representation for merging.

Standard model merging approaches involve combining the task-vectors $\{\boldsymbol{\tau}_1, \ldots, \boldsymbol{\tau}_M\}$ into a unified task vector $\boldsymbol{\tau}_{\text{uni}} = \mathcal{F}(\{\boldsymbol{\tau}_1, \ldots, \boldsymbol{\tau}_M\})$ and then adding the unified task vector to the pre-trained weights to get the final merged model, $\boldsymbol{\theta}_{\text{uni}} = \boldsymbol{\theta}_{\text{pre}} + \boldsymbol{\tau}_{\text{uni}}$. Here $\mathcal{F}$ denotes the merging algorithm used to obtain the unified task vector's weights. For example, the unified task vector $\boldsymbol{\tau}_{\text{uni}}$ can be computed via simple averaging $\boldsymbol{\tau}_{\text{uni}} = \frac{1}{M}\sum_{i=1}^{M} \boldsymbol{\tau}_i$ or via TA (Ilharco et al., 2023) that uses a coefficient λ to weigh the contribution[1] of the unified task vector $\boldsymbol{\tau}_{\text{uni}} = \lambda \cdot \frac{1}{M}\sum_{i=1}^{M} \boldsymbol{\tau}_i$ in the final merged model. It is shown that just by tuning λ, one can outperform weight averaging (Ilharco et al., 2023).

Motivation. Merging into one model $\boldsymbol{\theta}_{\text{uni}}$ may cause accuracy deterioration due to parameter interference between different fine-tuned models (Zhang et al., 2025; Yadav et al., 2023). This behavior becomes prominent as more and more fine-tuned models are merged, as discussed in Section 1. On the other hand, retaining all fine-tuned models preserves full fine-tuning accuracy but results in a net size $M\times$ that of one fine-tuned model, which is impractical due to the high memory requirements. In this work, we investigate the problem of generating models of any desired size in the range $[1, M]$, including models with fractional size such as $2.25\times$ model units.

3.2 Proposed approach

To enable a more granular fusion, we consider the model to be composed of B sequential blocks, for instance transformer blocks in a ViT model or even layers within each transformer block such as attention or MLP layers could be considered as unique blocks. Assuming B total blocks, we consider the task vectors for each block as $\{\boldsymbol{\tau}_k^b\}_{b=1}^B$ corresponding to the original task vector $\boldsymbol{\tau}_k$ for a task k. Our proposed framework, FlexMerge, takes a greedy approach to efficiently merge task vectors from multiple tasks at the granularity of blocks, aiming to reduce the deployed model size while maintaining utility. The pseudo-code for FlexMerge is presented in Algorithm 1.

Algorithm 1: FlexMerge framework

Input: Task vectors $\{\boldsymbol{\tau}_k^b\}$ for all $k \in [M], b \in [B]$; merging algorithm $\mathcal{F}$; target size S_{target}
Output: Merged task vectors with reduced size

$S \leftarrow 0$ ▷ *Initialize deployed size*
for $b = 1$ **to** B **do**
  $\mathcal{G}^b \leftarrow \emptyset$
  for $k = 1$ **to** M **do**
    $\mathcal{G}^b \leftarrow \mathcal{G}^b \cup \left(\{k\}, \boldsymbol{\tau}_k^b\right)$
    $S \leftarrow S + \text{size}(\boldsymbol{\tau}_k^b)$
while $S > S_{target}$ ***or*** *not all blocks merged* **do**
  Find block b^* and pair $(g_{i^*}, g_{j^*}) \in \mathcal{G}^{b^*}$ with the highest similarity:
  $$(b^*, g_{i^*}, g_{j^*}) = \underset{b \in [B],\, g_i, g_j \in \mathcal{G}^b}{\arg\max} \ \textsc{similarity}(g_i, g_j)$$
  $\mathcal{T}_{i^*}^{b^*}, \mathcal{T}_{j^*}^{b^*} \leftarrow g_{i^*}(0),\ g_{j^*}(0)$ ▷ *Get task subsets*
  $\mathcal{T}_{\text{uni}}^{b^*} \leftarrow \mathcal{T}_{i^*}^{b^*} \cup \mathcal{T}_{j^*}^{b^*}$ ▷ *Merge task subsets*
  $\boldsymbol{\tau}_{\text{uni}}^{b^*} \leftarrow \mathcal{F}(\{\boldsymbol{\tau}_k^{b^*} \mid k \in \mathcal{T}_{\text{uni}}^{b^*}\})$ ▷ *Merge task vectors*
  $\mathcal{G}^{b^*} \leftarrow \mathcal{G}^{b^*} \cup \left(\mathcal{T}_{\text{uni}}^{b^*}, \boldsymbol{\tau}_{\text{uni}}^{b^*}\right) \setminus \{g_{i^*}, g_{j^*}\}$ ▷ *Update the block*
  $S \leftarrow S - \text{size}(\boldsymbol{\tau}_{\text{uni}}^{b^*})$ ▷ *Update current size*

[1] We add a scaling factor of $1/M$ to the standard definition $\boldsymbol{\tau}_{\text{uni}} = \lambda \cdot \sum_{i=1}^{M} \boldsymbol{\tau}_i$ given in (Ilharco et al., 2023) to better suit its usage in FlexMerge where M can vary across blocks.

Initialization (Lines 1–6). The merging proceeds bottom-up. Initially, no merging has occurred, and we retain τ_k^b for all tasks $k \in [M]$ and all blocks $b \in [B]$ (see Figure 1(a)). For each block b, we initialize a set of tuples: $\mathcal{G}^b = \left\{ \left(\{k\}, \tau_k^b \right) \mid k \in [M] \right\}$. Each tuple in $\mathcal{G}^b$ consists of: *(i)* a task set $\{k\}$ (tracking which tasks are represented) and *(ii)* the corresponding block task vector τ_k^b. For example, in Figure 1(a) for the first block, we would have $\mathcal{G}^1 = \left\{ \left(\{a\}, \tau_a^1 \right), \ldots, \left(\{e\}, \tau_e^1 \right) \right\}$. When the merging terminates, the resulting $\mathcal{G}^1$ for Figure 1(a) would be $\mathcal{G}^1 = \left\{ \left(\{a, \ldots, e\}, \hat{\tau}_{\text{uni}}^1 \right) \right\}$, where $\hat{\tau}_{\text{uni}}^1$ is the merged task vector for the first block for all tasks. The initial size S is calculated as the cumulative size of all block parameters across M tasks.

Iteration (lines 7-14). In each iteration, the algorithm identifies a block b^* and pair of tuples $(g_{i^*}, g_{j^*}) \in \mathcal{G}^{b^*}$, which have the highest similarity (as defined below). Then they are merged as follows. Let $\mathcal{T}_{i^*}^{b^*}$ and $\mathcal{T}_{j^*}^{b^*}$ be the subset of tasks associated with g_{i^*} and g_{j^*} respectively, *i.e.*, the first elements of g_{i^*} and g_{j^*} respectively. First, $\mathcal{T}_{i^*}^{b^*}$ and $\mathcal{T}_{j^*}^{b^*}$ are merged via a union operation: $\mathcal{T}_{\text{uni}}^{b^*} = \mathcal{T}_{i^*}^{b^*} \cup \mathcal{T}_{j^*}^{b^*}$. Next, the merged task vector corresponding to block b^* and set $\mathcal{T}_{\text{uni}}^{b^*}$ is created as follows: $\tau_{\text{uni}}^{b^*} = \mathcal{F}(\{\tau_k^{b^*} \mid k \in \mathcal{T}_{\text{uni}}^{b^*}\})$. Here $\mathcal{F}$ can be *any* data-free merging algorithm. The tuple set $\mathcal{G}^{b^*}$ is then updated by removing the tuples g_{i^*}, g_{j^*} and adding the new merged tuple $(\mathcal{T}_{\text{uni}}^{b^*}, \tau_{\text{uni}}^{b^*})$. Each merge reduces the model size by the size of the task vector corresponding to block b^*, and the process continues until the current size S meets the desired size S_{target} or no further merges are possible.

Similarity function. We measure the similarity between two groups g_i, g_j in any block b using the lowest cosine similarity between any pair of original task vectors corresponding to the tasks in the sets $\mathcal{T}_i^b$ and $\mathcal{T}_j^b$:

$$\textsc{similarity}(g_i, g_j) = \min_{k_1 \in \mathcal{T}_i^b,\, k_2 \in \mathcal{T}_j^b} \text{cosine\_sim}(\tau_{k_1}^b, \tau_{k_2}^b). \tag{1}$$

Our choice of the min similarity derives from our ablations comparing different strategies—max, min, and average—as well as computing similarity between merged group task vectors directly. Among these, min yields the best performance. Thus at each iteration, we merge the pair of groups with the highest of these minimum similarities (line 9, Algorithm 1). While the cosine similarity between full task vectors can be relatively low (Ilharco et al., 2023), the block-level similarities tend to be higher and effective for merging. Channel Merging (Zhang et al., 2025) also employs cosine similarity.

Enhancing efficiency. The pairwise similarities can be precomputed once for all pairs and accessed in constant time during the merging process. Furthermore, we leverage the Disjoint Set Union (DSU) (Cormen et al., 2009) data structure to efficiently track and unify task sets for each block. Our design enables FlexMerge to perform very efficient merging even under many tasks (see Table 2).

Complexity Analysis. Algorithm 1 identifies most similar block pairs in each iteration. We presented the algorithm in this form for conceptual clarity. However, in practice, we generate the global merging order for all blocks first and then apply merges. To analyze the time complexity of FlexMerge, we consider three distinct stages of this process: similarity pre-computation, generating merging order, and actual parameter merging.

- **Similarity pre-computation:** We compute the pairwise cosine similarities between M tasks in each block, across all B blocks. Let the maximum size of any block task vector be $d_{\max}$, then the similarity computes takes $O(d_{\max})$ per pair. With $\binom{M}{2}$ pairs per block, this step is $O(BM^2 d_{\max})$.
- **Generating merging order:** Our greedy merging using Equation (1) is an instance of a specific form of clustering, called single-linkage clustering. We thus use the SLINK algorithm (Sibson, 1973) which takes as input the similarity matrix and generates a sorted list of merge orders for each block in $O(M^2)$. For B blocks, this takes $O(BM^2)$. We then need to combine these per-block sorted lists, each of size $M - 1$, into a single global sorted list. Using a min-heap, this takes $O(BM \log B)$ (Knuth, 1997). In total, this step takes $O(BM^2 + BM \log B)$ and results in a global merge ordering across all blocks.
- **Applying parameter merging:** We now merge the blocks in the generated order. For linear algorithms like TA, merging a group g of size $|g|$ with task vectors of size d takes $O(|g|d)$. Summing over all groups $\mathcal{G}^b$ within a block takes $\sum_{g \in \mathcal{G}^b} O(|g| \cdot d) = O((\sum_{g \in \mathcal{G}^b} |g|)d) = O(Md)$. Repeating this for B blocks and upper bounding d with $d_{\max}$ results in $O(BMd_{\max})$.

Table 1: Summary of existing data-free merging methods. Column $\mathcal{F}(\{\tau_k^b \mid k \in \mathcal{T}_{\text{uni}}^b\})$ denotes the result of merging. Figure 7 (Section B) provides an illustrative diagram.

Algorithm	$\mathcal{F}(\{\tau_k^b \mid k \in \mathcal{T}_{\text{uni}}^b\})$	Final Model	What is stored?
TA (Ilharco et al., 2023), TIES (Yadav et al., 2023), Avg. (Ilharco et al., 2023), PCB (Du et al., 2024), TSV-M (Gargiulo et al., 2025)	τ_{uni}^b	$\theta_{\text{uni}}^b = \theta_{\text{pre}}^b + \tau_{\text{uni}}^b$	θ_{uni}^b
Consensus (Wang et al., 2024)	$\tau_{\text{uni}}^b, \{m_k^b \mid k \in \mathcal{T}_{\text{uni}}\}$	$\hat{\theta}_k^b = \theta_{\text{pre}}^b + \tau_{\text{uni}}^b \circ m_k^b$ (reconstructed per-task k)	$\theta_{\text{pre}}^b, \tau_{\text{uni}}^b, \{m_k^b \mid k \in \mathcal{T}_{\text{uni}}\}$
EMR-Merging (Huang et al., 2024)	$\tau_{\text{uni}}^b, \{m_k^b, \gamma_k^b \mid k \in \mathcal{T}_{\text{uni}}^b\}$	$\hat{\theta}_k^b = \theta_{\text{pre}}^b + \gamma_k^b \cdot \tau_{\text{uni}}^b \circ m_k^b$ (reconstructed per-task k)	$\theta_{\text{pre}}^b, \tau_{\text{uni}}^b, \{m_k^b, \gamma_k^b \mid k \in \mathcal{T}_{\text{uni}}\}$

The total complexity is dominated by the similarity pre-computation (as $d_{\max}$ is typically larger than B), resulting in a final complexity of $O(BM^2 d_{\max})$. Note however that $d_{\max}$ is much smaller than the total model dimension, at it only corresponds to the maximum size of any block of the model.

3.3 Existing merging methods in combination with FlexMerge

FlexMerge provides the flexibility to choose any data-free merging algorithm $\mathcal{F}$ from a diverse set of existing approaches. Unlike traditional methods that operate at the level of full task vectors, FlexMerge applies merging algorithms at the block level, fusing block task vectors. We detail the exact block-level merging procedure for different algorithms next. In standard approaches like TA, TSV-M, and PCB-Merging, task vectors are merged into a single unified task vector. When applied at the block-level, the merging outcome for any block b can be denoted as: $\tau_{\text{uni}}^b \leftarrow \mathcal{F}(\{\tau_k^b \mid k \in \mathcal{T}_{\text{uni}}^b\})$ where $\mathcal{F}$ is the specific merging algorithm and $\mathcal{T}_{\text{uni}}^b$ is the subset of tasks for which the merging occurs. The final block parameters are then computed as $\theta_{\text{uni}}^b = \theta_{\text{pre}}^b + \tau_{\text{uni}}^b$. Approaches such as Consensus generate task-specific masks in addition to the unified vector: $\tau_{\text{uni}}^b, \{m_k^b \mid k \in \mathcal{T}_{\text{uni}}\} \leftarrow \mathcal{F}(\{\tau_k^b \mid k \in \mathcal{T}_{\text{uni}}^b\})$. Then during inference, the task-specific weights for task k are reconstructed as $\hat{\theta}_k^b = \theta_{\text{pre}}^b + \tau_{\text{uni}}^b \circ m_k^b$. Consensus thus stores θ_{pre}^b, τ_{uni}^b, and the binary masks $\{m_k^b \mid k \in \mathcal{T}_{\text{uni}}\}$ and defers per-task reconstruction to the inference time. This leads to a storage cost exceeding $2\times$ that of standard methods, which only store θ_{uni}^b. EMR-Merging further generates task-specific scalars $\{\gamma_k^b \mid k \in \mathcal{T}_{\text{uni}}\}$ in addition to the masks, however the storage cost of these scalars is negligible. Table 1 summarizes the merging outcomes for different algorithms, applied at block-level within FlexMerge. Figure 7 (Section B) provides an illustrative diagram.

4 Experiments

We split our evaluation as follows: *(i)* Merging on vision, PEFT and FFT benchmarks in Section 4.1; (ii) FlexMerge vs Channel Merging in Section 4.2; and *(iii)* ablation and efficiency analysis in Section 4.3. Lastly, multi-modal and OOD results are in Appendices C.4 and C.6.[2].

Merging algorithms. We investigate the accuracy-size trade-off for several data-free merging algorithms including Vanilla Averaging, TA (Ilharco et al., 2023), TIES-Merging (Yadav et al., 2023), PCB-Merging (Du et al., 2024), TSV-M (Gargiulo et al., 2025), Consensus (Wang et al., 2024) and EMR-Merging (Huang et al., 2024) on extensive vision and NLP benchmarks. As noted earlier, the focus of our work is data-free model merging. Hence, existing data-based algorithms such as Surgery (Yang et al., 2024a), AdaMerging (Yang et al., 2024b), Twin-Merging (Lu et al., 2024), *etc.* are not directly comparable in our setting.

Hyperparameters. For TA, we set $\lambda = 1.5$. For TIES-Merging, we use a sparsity ratio of 0.1 and employ the recommended value of $\lambda = 1$. For Consensus, we set the hyperparameter responsible for controlling the amount of information extracted by masks to 0.6 for all tasks and use TIES-Merging as the algorithm to generate unified task vectors. For FlexMerge, we set the block granularity at the level of individual components within the transformer layer, *i.e.*, the attention, MLP, and layer normalization modules are treated as separate blocks during the merging process.

[2]Our anonymized code is available at: `https://anonymous.4open.science/r/model-merging-84F2`

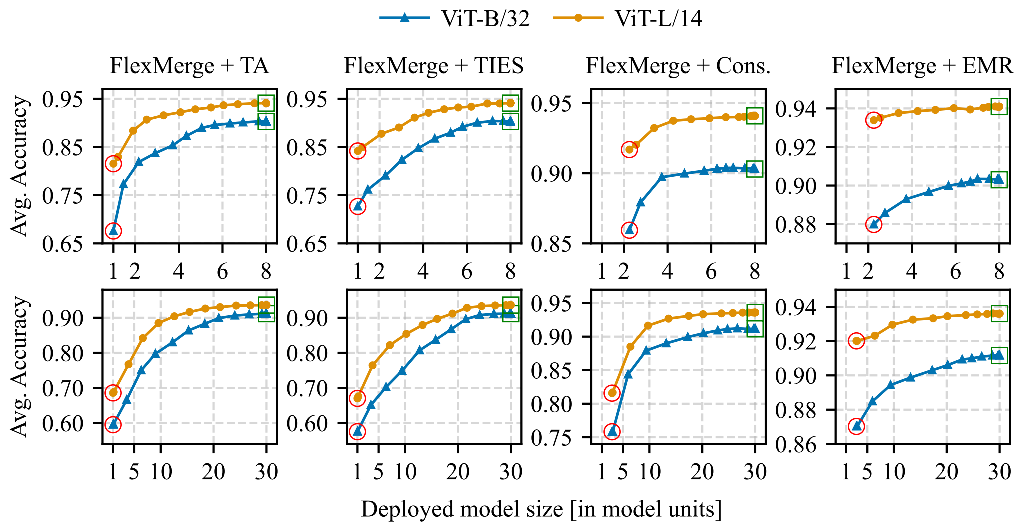


Figure 3: Merging 8 (top) and 30 (bottom) tasks. The accuracy-size trade-off shows rapid initial gains, followed by gradual improvement, reaching near fine-tuning accuracy well before the maximum size.

4.1 Merging Results

Merging 8 **and** 30 **vision models.** For the image classification tasks, we follow the setup from existing work (Huang et al., 2024; Yadav et al., 2023). Specifically, we use two versions of the CLIP model (Radford et al., 2021), incorporating ViT-B/32 and ViT-L/14 as visual encoders (Dosovitskiy et al., 2021). We evaluate on the standard 8 task benchmark (Ilharco et al., 2023) as well as an extended 30 task benchmark (detailed in Appendix B.2). Figure 3 plots average accuracy vs. deployed model size (in multiples of a single fine-tuned model). For FlexMerge + TA, the accuracy increases fairly rapidly as the model size grows beyond 1×. The gains are significant (top row), where the accuracy reaches > 80% at size 2× from only 67.5% at size 1× for the ViT-B/32 model in the 8 task setup. Similar gains are also observed for 30 tasks (bottom row).

Masking-based approaches, Consensus and EMR-Merging, begin with substantially higher accuracy than TA and TIES-Merging, but their smallest size exceeds 1× due to the need to store pre-trained weights and binary masks (Section 3.3). On 8 tasks, Consensus was shown to match fine-tuned accuracy at small sizes, but only when its extraction parameter is separately tuned per task (Wang et al., 2024). FlexMerge + Consensus also shows strong gains, improving from 76% at ≈ 3× to 84.5% at ≈ 6× for ViT-B/32 in 30 tasks. EMR-Merging maintains high accuracy even at the smallest size. Yet, it exhibits an accuracy gap w.r.t the fine-tuned models, which can be effectively reduced by increasing the deployed model size. Larger ViT-L/14 models achieve higher accuracy across all methods, but the accuracy-size trade-off remains similar: rapid initial gains followed by gradual improvements. Most algorithms approach the fine-tuning accuracy (denoted by □) well before maximum size, around 6× for 8 tasks and 23.5× for 30 tasks. Thus, in cases requiring storage of all fine-tuned models, FlexMerge can reduce size by about 25% with little accuracy loss.

Merging 11 **PEFT models.** We adopt the experimental setup from prior work (Huang et al., 2024; Yadav et al., 2023). Specifically, we employ the $(IA)^3$ (Liu et al., 2022) PEFT method on the T0-3B (Sanh et al., 2022) base model using 11 diverse datasets sourced from (Yadav et al., 2023) (detailed in Section B.3). Figure 4(a) demonstrates the benefits of deploying larger model sizes, where in this case the model size is measured with respect to the $(IA)^3$ modules. FlexMerge + TA achieves notable gains, increasing accuracy from 59% at size 1× to 66.2% at 3×, a 7.2% improvement. Similarly, FlexMerge + EMR-Merging surpasses 70% accuracy at 5×, starting from 67.6% at the lowest size of 2.34×. We observe similar trends for other algorithms, included in Appendix C.2.

Merging 7 **FFT models.** For this experiment, we closely follow the setup from prior work (Du et al., 2024; Yadav et al., 2023). We use T5-Base and T5-Large as base models, applying full-parameter

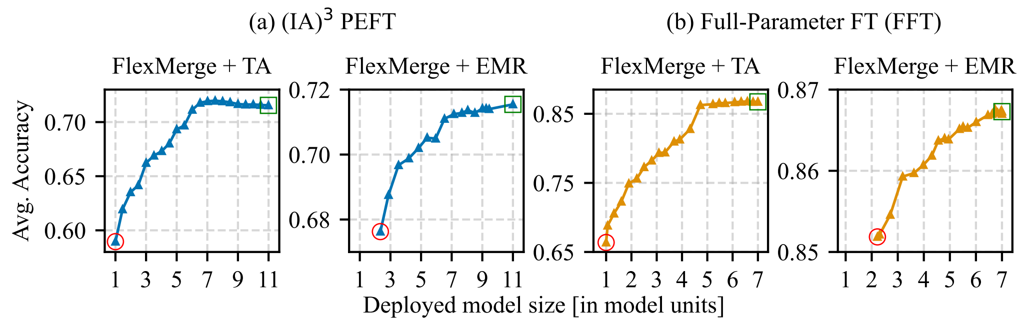


Figure 4: FlexMerge + TA gains 7.2% for $(IA)^3$ going from 1× to 3× and more than 9% for FFT when just doubling the size from 1× to 2×. EMR begins with higher accuracy, yet, substantially benefits from increased size.

fine-tuning on 7 datasets sourced from (Yadav et al., 2023) (detailed in Appendix B.4). Figure 4(b) illustrates the trade-off between model size and accuracy for the T5-Large model. Here, one unit of model size corresponds to the full size of a single model. FlexMerge + TA gains more than 9% to reach an accuracy of 75% when just doubling the size from 1× to 2×. Similarly, FlexMerge + EMR-Merging surpasses 86% at size 4×, starting from 85.2% at its lowest size of 2.2×. Consistent with our observations on vision tasks, FlexMerge + TA reaches very close to the fine-tuning accuracy around size 5×, much in advance of full size 7×. Thus, scaling the model size benefits both ends of the spectrum. Results for other combinations are included in Appendix C.3.

Cross-algorithm analysis. Thus far, we evaluated the accuracy-size trade-off per algorithm. We now compare algorithms at same size, yielding two interesting findings: *(i)* the performance gap between different algorithms significantly narrows at slightly larger sizes; and *(ii)* the algorithms rankings also alter in many cases, with simpler algorithms rivaling or surpassing advanced ones. This behavior aligns with the phenomenon we observed earlier: as scale increases, parameter interference between tasks reduces substantially, leading to greater natural parameter disentanglement and higher accuracy. Consequently, the explicit interference-reduction mechanisms built into advanced approaches such as TRIM in TIES or competition balancing in PCB offer marginal added benefit, because much of the interference is already mitigated simply by the increase in scale. Thus simple algorithms begin rivaling their more sophisticated counterparts at larger sizes. In Figure 5 on vision tasks, vanilla averaging exceeds TIES-Merging at size 3.25× while TA overlaps with PCB. While EMR-Merging and Consensus stay atop on vision, they are surpassed by PCB on PEFT at size 4.5×. Crucially, all algorithms remain within $3-4\%$ on both benchmarks at increased sizes despite originating with a

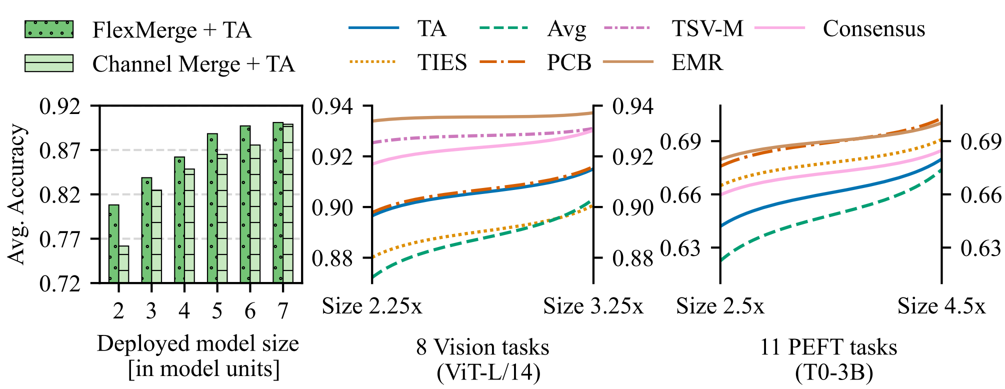


Figure 5: (Left) FlexMerge + TA outperforms Channel Merging + TA across all sizes. (Center, Right) Algorithm rankings shift even at modestly larger sizes, with simpler methods rivaling advanced ones. We show sizes just over Consensus and EMR-Merging's lowest size for a wholistic comparison.

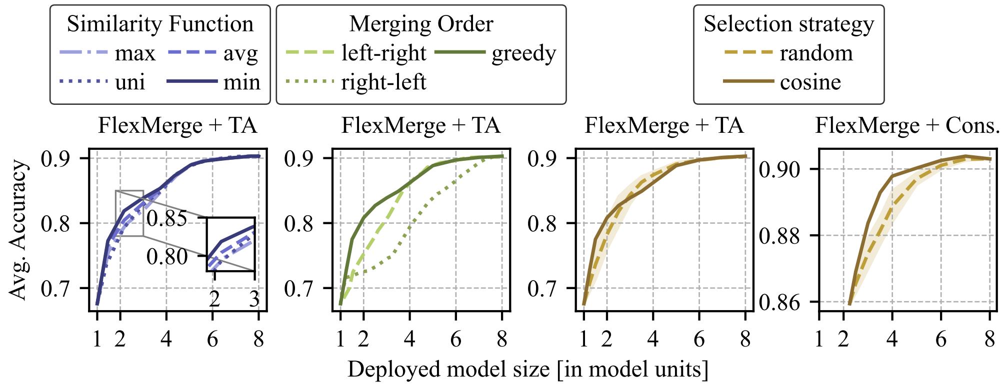


Figure 6: Ablation results for FlexMerge reveal that the min similarity strategy and greedy merging perform the best, while cosine similarity generally outperforms random selection.

much larger gap at size 1× (see Figure 1(b)). *Our findings provide encouraging evidence to develop and compare algorithms at sizes > 1× rather than only at 1×.*

4.2 FlexMerge vs Channel Merging

Channel Merging (Zhang et al., 2025) uses K-Means clustering per layer, following a fixed same value of K for every layer. Each choice of $K \in \{2, 3, \ldots, M - 1\}$ results in a merged model of the corresponding size. Figure 5 charts the average accuracy with TA and ViT-B/32 for a set of integer model sizes, excluding the extremes 1× and 8× where both approaches have the identical accuracy. Recall that Channel Merging does not support fractional sizes. FlexMerge achieves higher accuracy than Channel Merging in all cases, thanks to its greedy pairwise merging approach which allows flexible number of groups per layer instead of restrictive clustering. Results with TIES-Merging and visualization of clusters is included in Appendix C.5.

4.3 Analysis

Ablations on the merging procedure. We ablate on the similarity functions (min, max, average, comparing unified vectors) for Equation (1) and merging orders (left-to-right, right-to-left, greedy) in FlexMerge using the ViT-B/32 model on 8 tasks. We also investigate random block selection over cosine similarity. Figure 6 shows that the min strategy performs the best, though other strategies are also competitive. For merging order, right to left performs the worst as expected since the final layers in neural networks tend to be more specialized and merging them first hurts accuracy. While left to right seems ideal, it can be too strict and therefore greedy emerges as the best. We further analyze the merging order of greedy in Appendix C.10. Random selection is competitive, but generally underperforms when compared across algorithm. Based on these findings, we set FlexMerge to use greedy with cosine similarity (min strategy) by default. For more ablations, see Appendix C.8.

Merging and inference efficiency. Table 2 shows that FlexMerge achieves highly efficient data-free merging, generating all deployed sizes in about 20 sec for up to 30 tasks. For inference with FlexMerge, each request follows a unique forward path through the merged model using task-specific blocks (Figure 1(a)). For a model of size 1×, all tasks share a single path, but the classification heads are always applied separately. We load the tensors of merged model (size > 1×) into the GPU memory once and create M task-specific model views that reference these shared tensors to process task batches in parallel. Standard merging, by contrast, processes all tasks in a single batch before splitting for task-specific heads. We simulate the worst case arrival, where inference batches corresponding to all tasks arrive at once. We consider 50 consecutive batches of size 256 (totaling 12800 samples). Each batch contains 32 samples per task across 8 tasks. Table 3 shows that FlexMerge maintains inference speed comparable to standard merging for both ViT-B/32 and ViT-L/14, demonstrating that larger models can enhance accuracy without slowing inference.

Table 2: Merging time (ViT-B/32 vs ViT-L/14).

Method	ViT-B/32 (s)	ViT-L/14 (s)
8 Tasks		
TA (size 1×)	≈ 0.8	**≈ 2.6**
FlexMerge (all sizes)	≈ 2.3	**≈ 3.4**
30 Tasks		
TA (size 1×)	≈ 1.9	**≈ 6.1**
FlexMerge (all sizes)	≈ 20	**≈ 31**

Table 3: Comparing inference time of FlexMerge against standard model merging. The overheads are negligible.

Model	Algorithm	Size	Inference Cost (/12800 items)
ViT-B-32	Standard Merging	1×	12.30 ± 0.21 ms
	FlexMerge	> 1×	12.21 ± 0.41 ms
ViT-L-14	Standard Merging	1×	118.70 ± 1.78 ms
	FlexMerge	> 1×	120.53 ± 0.32 ms

5 Discussion and Conclusion

Benefits. Different merging algorithms have different advantages: EMR and Consensus achieve high accuracy but require task-specific reconstruction during inference, incurring overheads. FlexMerge can also mitigate this overhead as larger deployed models need fewer blocks to be reconstructed (see Appendix C.9). In contrast, TIES and TA avoid reconstruction but have lower accuracy. FlexMerge provides flexibility, letting practitioners choose algorithms and balance accuracy, reconstruction overhead, and model size for various deployment scenarios.

Limitations. Most works, including FlexMerge, are limited to merging models with the same architecture as merging heterogeneous models remains challenging (Singh & Jaggi, 2020; Imfeld et al., 2024). Secondly, the theoretical insights for effective model merging are limited (Ortiz-Jimenez et al., 2023). For FlexMerge, how to obtain the optimal merged model for any given size remains unclear. Although extensive ablations help guide (Section 4.3), further investigation is needed to understand the bounds of the accuracy-size trade-off.

We introduced FlexMerge, a flexible, data-free model merging framework that extends beyond traditional single-model fusion and offers precise control over fused model size. Extensive experiments show that the accuracy-size trade-off exhibits favorable properties for several algorithms, benefiting from rapid accuracy gains with modest size increments. Future work may explore specialized algorithms for block-level merging.

References

M Israk Ahmed, Shahriyar Mahmud Mamun, and Asif Uz Zaman Asif. Dcnn-based vegetable image classification using transfer learning: A comparative study. In *2021 5th International Conference on Computer, Communication and Signal Processing (ICCCSP)*, pp. 235–243. IEEE, 2021.

Sarder Iftekhar Ahmed, Muhammad Ibrahim, Md Nadim, Md Mizanur Rahman, Maria Mehjabin Shejunti, Taskeed Jabid, and Md Sawkat Ali. Mangoleafbd: A comprehensive image dataset to classify diseased and healthy mango leaves. *Data in Brief*, 47:108941, 2023.

Stephen H Bach, Victor Sanh, Zheng-Xin Yong, Albert Webson, Colin Raffel, Nihal V Nayak, Abheesht Sharma, Taewoon Kim, M Saiful Bari, Thibault Fevry, et al. Promptsource: An integrated development environment and repository for natural language prompts. *arXiv preprint arXiv:2202.01279*, 2022.

Satanjeev Banerjee and Alon Lavie. METEOR: An automatic metric for MT evaluation with improved correlation with human judgments. In Jade Goldstein, Alon Lavie, Chin-Yew Lin, and Clare Voss (eds.), *Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization*, pp. 65–72, Ann Arbor, Michigan, June 2005. Association for Computational Linguistics. URL `https://aclanthology.org/W05-0909/`.

Puneet Bansal. Intel image classification. *Available on https://www. kaggle. com/puneet6060/intel-image-classification, Online*, 2019.

Rishi Bommasani, Drew A Hudson, Ehsan Adeli, Russ Altman, Simran Arora, Sydney von Arx, Michael S Bernstein, Jeannette Bohg, Antoine Bosselut, Emma Brunskill, et al. On the opportunities and risks of foundation models. *arXiv preprint arXiv:2108.07258*, 2021.

Lukas Bossard, Matthieu Guillaumin, and Luc Van Gool. Food-101 – Mining Discriminative Components with Random Forests. In *IEEE European Conference on Computer Vision (ECCV)*, 2014.

CCHANG. Garbage classification. `https://www.kaggle.com/ds/81794`, 2018.

Junbum Cha, Sanghyuk Chun, Kyungjae Lee, Han-Cheol Cho, Seunghyun Park, Yunsung Lee, and Sungrae Park. SWAD: domain generalization by seeking flat minima. In Marc'Aurelio Ranzato, Alina Beygelzimer, Yann N. Dauphin, Percy Liang, and Jennifer Wortman Vaughan (eds.), *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, pp. 22405–22418, 2021. URL `https://proceedings.neurips.cc/paper/2021/hash/bcb41ccdc4363c6848a1d760f26c28a0-Abstract.html`.

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

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.

Tarin Clanuwat, Mikel Bober-Irizar, Asanobu Kitamoto, Alex Lamb, Kazuaki Yamamoto, and David Ha. Deep learning for classical japanese literature. *arXiv*, 2018. URL `http://arxiv.org/abs/1812.01718v1`.

Adam Coates, Andrew Ng, and Honglak Lee. An analysis of single-layer networks in unsupervised feature learning. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2011. `https://proceedings.mlr.press/v15/coates11a.html`.

Gregory Cohen, Saeed Afshar, Jonathan Tapson, and Andre Van Schaik. EMNIST: Extending MNIST to handwritten letters. In *International Joint Conference on Neural Networks (IJCNN)*, 2017.

Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein. *Introduction to Algorithms, Third Edition*. The MIT Press, 3rd edition, 2009. ISBN 0262033844.

Marie-Catherine De Marneffe, Mandy Simons, and Judith Tonhauser. The commitmentbank: Investigating projection in naturally occurring discourse. In *proceedings of Sinn und Bedeutung*, volume 23, pp. 107–124, 2019.

DeepNets. Landscape recognition. `https://www.kaggle.com/datasets/utkarshsaxenadn/landscape-recognition-image-dataset-12k-images`.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 248–255, 2009. doi: 10.1109/CVPR.2009.5206848.

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.

Jesse Dodge, Gabriel Ilharco, Roy Schwartz, Ali Farhadi, Hannaneh Hajishirzi, and Noah Smith. Fine-tuning pretrained language models: Weight initializations, data orders, and early stopping. *arXiv preprint arXiv:2002.06305*, 2020.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2021.

Guodong Du, Junlin Lee, Jing Li, Runhua Jiang, Yifei Guo, Shuyang Yu, Hanting Liu, Sim Kuan Goh, Ho-Kin Tang, Daojing He, and Min Zhang. Parameter competition balancing for model merging. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. URL `https://openreview.net/forum?id=l5SbrtvSRS`.

Chris Fifty, Ehsan Amid, Zhe Zhao, Tianhe Yu, Rohan Anil, and Chelsea Finn. Efficiently identifying task groupings for multi-task learning. *Advances in Neural Information Processing Systems*, 34: 27503–27516, 2021.

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.

Danilo Giampiccolo, Bernardo Magnini, Ido Dagan, and William B Dolan. The third pascal recognizing textual entailment challenge. In *Proceedings of the ACL-PASCAL workshop on textual entailment and paraphrasing*, pp. 1–9, 2007.

Ian J Goodfellow, Dumitru Erhan, Pierre Luc Carrier, Aaron Courville, Mehdi Mirza, Ben Hamner, Will Cukierski, Yichuan Tang, David Thaler, Dong-Hyun Lee, et al. Challenges in representation learning: A report on three machine learning contests. In *International Conference on Neural Information Processing (ICONIP)*, 2013. URL http://arxiv.org/abs/1307.0414v1.

Vipul Gupta, Santiago Akle Serrano, and Dennis DeCoste. Stochastic weight averaging in parallel: Large-batch training that generalizes well. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net, 2020. URL https://openreview.net/forum?id=rygFWAEFwS.

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

Chenyu Huang, Peng Ye, Tao Chen, Tong He, Xiangyu Yue, and Wanli Ouyang. EMR-merging: Tuning-free high-performance model merging. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. URL https://openreview.net/forum?id=lYdjzx3DYu.

Gabriel Ilharco, Marco Tulio Ribeiro, Mitchell Wortsman, Ludwig Schmidt, Hannaneh Hajishirzi, and Ali Farhadi. Editing models with task arithmetic. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=6t0Kwf8-jrj.

Moritz Imfeld, Jacopo Graldi, Marco Giordano, Thomas Hofmann, Sotiris Anagnostidis, and Sidak Pal Singh. Transformer fusion with optimal transport. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=LjeqMvQpen.

Pavel Izmailov, Dmitrii Podoprikhin, Timur Garipov, Dmitry P. Vetrov, and Andrew Gordon Wilson. Averaging weights leads to wider optima and better generalization. In Amir Globerson and Ricardo Silva (eds.), *Proceedings of the Thirty-Fourth Conference on Uncertainty in Artificial Intelligence, UAI 2018, Monterey, California, USA, August 6-10, 2018*, pp. 876–885. AUAI Press, 2018. URL http://auai.org/uai2018/proceedings/papers/313.pdf.

Mona Jalal, Kaihong Wang, Sankara Jefferson, Yi Zheng, Elaine O Nsoesie, and Margrit Betke. Scraping social media photos posted in kenya and elsewhere to detect and analyze food types. In *Proceedings of the 5th International Workshop on Multimedia Assisted Dietary Management*, pp. 50–59, 2019.

Ruochen Jin, Bojian Hou, Jiancong Xiao, Weijie J Su, and Li Shen. Fine-tuning attention modules only: Enhancing weight disentanglement in task arithmetic. In *The Thirteenth International Conference on Learning Representations*, 2025.

Xisen Jin, Xiang Ren, Daniel Preotiuc-Pietro, and Pengxiang Cheng. Dataless knowledge fusion by merging weights of language models. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=FCnohuR6AnM.

Aditya Khosla, Nityananda Jayadevaprakash, Bangpeng Yao, and Li Fei-Fei. Novel dataset for fine-grained image categorization. In *First Workshop on Fine-Grained Visual Categorization, IEEE Conference on Computer Vision and Pattern Recognition*, Colorado Springs, CO, June 2011.

Tushar Khot, Peter Clark, Michal Guerquin, Peter Jansen, and Ashish Sabharwal. Qasc: A dataset for question answering via sentence composition. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(05):8082–8090, Apr. 2020. doi: 10.1609/aaai.v34i05.6319. URL https://ojs.aaai.org/index.php/AAAI/article/view/6319.

Donald Ervin Knuth. *The art of computer programming*, volume 3. Pearson Education, 1997.

Jakub Konečný, H. Brendan McMahan, Daniel Ramage, and Peter Richtárik. Federated optimization: Distributed machine learning for on-device intelligence. *CoRR*, abs/1610.02527, 2016. URL http://arxiv.org/abs/1610.02527.

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

Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images, 2009. https://www.cs.toronto.edu/~kriz/learning-features-2009-TR.pdf.

Makerere AI Lab. Bean disease dataset, January 2020. URL https://github.com/AI-Lab-Makerere/ibean/.

Yann LeCun. The mnist database of handwritten digits. *http://yann. lecun. com/exdb/mnist/*, 1998.

Yeoreum Lee, Jinwook Jung, and Sungyong Baik. Mitigating parameter interference in model merging via sharpness-aware fine-tuning. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=eaTqsptDPL.

Hector Levesque, Ernest Davis, and Leora Morgenstern. The winograd schema challenge. In *Thirteenth international conference on the principles of knowledge representation and reasoning*, 2012.

Chin-Yew Lin. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pp. 74–81, Barcelona, Spain, July 2004. Association for Computational Linguistics. URL https://aclanthology.org/W04-1013/.

Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft coco: Common objects in context. In David Fleet, Tomas Pajdla, Bernt Schiele, and Tinne Tuytelaars (eds.), *Computer Vision – ECCV 2014*, pp. 740–755, Cham, 2014. Springer International Publishing. ISBN 978-3-319-10602-1.

Haokun Liu, Derek Tam, Mohammed Muqeeth, Jay Mohta, Tenghao Huang, Mohit Bansal, and Colin A Raffel. Few-shot parameter-efficient fine-tuning is better and cheaper than in-context learning. *Advances in Neural Information Processing Systems (NeurIPS)*, 35:1950–1965, 2022.

Zhenyi Lu, Chenghao Fan, Wei Wei, Xiaoye Qu, Dangyang Chen, and Yu Cheng. Twin-merging: Dynamic integration of modular expertise in model merging. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. URL https://openreview.net/forum?id=81YIt63TTn.

TorchVision maintainers and contributors. Torchvision: Pytorch's computer vision library. https://github.com/pytorch/vision, 2016.

Michael S Matena and Colin Raffel. Merging models with fisher-weighted averaging. In Alice H. Oh, Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho (eds.), *Advances in Neural Information Processing Systems*, 2022. URL https://openreview.net/forum?id=LSKlp_aceOC.

Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Agüera y Arcas. Communication-efficient learning of deep networks from decentralized data. In Aarti Singh and Xiaojin (Jerry) Zhu (eds.), *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics, AISTATS 2017, 20-22 April 2017, Fort Lauderdale, FL, USA*, volume 54 of *Proceedings of Machine Learning Research*, pp. 1273–1282. PMLR, 2017. URL http://proceedings.mlr.press/v54/mcmahan17a.html.

Yixin Nie, Adina Williams, Emily Dinan, Mohit Bansal, Jason Weston, and Douwe Kiela. Adversarial nli: A new benchmark for natural language understanding. *arXiv preprint arXiv:1910.14599*, 2019.

Maria-Elena Nilsback and Andrew Zisserman. Automated flower classification over a large number of classes. In *2008 Sixth Indian conference on computer vision, graphics & image processing*, 2008.

Guillermo Ortiz-Jimenez, Alessandro Favero, and Pascal Frossard. Task arithmetic in the tangent space: Improved editing of pre-trained models. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL `https://openreview.net/forum?id=0A9f2jZDGW`.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, ACL '02, pp. 311–318, USA, 2002. Association for Computational Linguistics. doi: 10.3115/1073083.1073135. URL `https://doi.org/10.3115/1073083.1073135`.

Omkar M Parkhi, Andrea Vedaldi, Andrew Zisserman, and CV Jawahar. Cats and dogs. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012.

Mohammad Taher Pilehvar and Jose Camacho-Collados. Wic: the word-in-context dataset for evaluating context-sensitive meaning representations. *arXiv preprint arXiv:1808.09121*, 2018.

Konstantin Pogorelov, Kristin Ranheim Randel, Carsten Griwodz, Sigrun Losada Eskeland, Thomas de Lange, Dag Johansen, Concetto Spampinato, Duc-Tien Dang-Nguyen, Mathias Lux, Peter Thelin Schmidt, et al. Kvasir: A multi-class image dataset for computer aided gastrointestinal disease detection. In *Proceedings of the 8th ACM on Multimedia Systems Conference*, pp. 164–169, 2017.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language Models are Unsupervised Multitask Learners, 2019. `https://openai.com/blog/better-language-models/`.

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. PMLR, 2021.

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

Melissa Roemmele, Cosmin Adrian Bejan, and Andrew S Gordon. Choice of plausible alternatives: An evaluation of commonsense causal reasoning. In *2011 AAAI Spring Symposium Series*, 2011.

Keisuke Sakaguchi, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. Winogrande: An adversarial winograd schema challenge at scale. *Communications of the ACM*, 64(9):99–106, 2021.

Victor Sanh, Albert Webson, Colin Raffel, Stephen Bach, Lintang Sutawika, Zaid Alyafeai, Antoine Chaffin, Arnaud Stiegler, Arun Raja, Manan Dey, M Saiful Bari, Canwen Xu, Urmish Thakker, Shanya Sharma Sharma, Eliza Szczechla, Taewoon Kim, Gunjan Chhablani, Nihal Nayak, Debajyoti Datta, Jonathan Chang, Mike Tian-Jian Jiang, Han Wang, Matteo Manica, Sheng Shen, Zheng Xin Yong, Harshit Pandey, Rachel Bawden, Thomas Wang, Trishala Neeraj, Jos Rozen, Abheesht Sharma, Andrea Santilli, Thibault Fevry, Jason Alan Fries, Ryan Teehan, Teven Le Scao, Stella Biderman, Leo Gao, Thomas Wolf, and Alexander M Rush. Multitask prompted training enables zero-shot task generalization. In *International Conference on Learning Representations*, 2022. URL `https://openreview.net/forum?id=9Vrb9D0WI4`.

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

R. Sibson. Slink: An optimally efficient algorithm for the single-link cluster method. *The Computer Journal*, 16(1):30–34, 01 1973. ISSN 0010-4620. doi: 10.1093/comjnl/16.1.30. URL https://doi.org/10.1093/comjnl/16.1.30.

Sidak Pal Singh and Martin Jaggi. Model fusion via optimal transport. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 22045–22055. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper_files/paper/2020/file/fb2697869f56484404c8ceee2985b01d-Paper.pdf.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D Manning, Andrew Ng, and Christopher Potts. Recursive deep models for semantic compositionality over a sentiment treebank. In *Empirical Methods in Natural Language Processing (EMNLP)*, 2013. https://aclanthology.org/D13-1170/.

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*, pp. 1453–1460. IEEE, 2011.

Trevor Standley, Amir Zamir, Dawn Chen, Leonidas Guibas, Jitendra Malik, and Silvio Savarese. Which tasks should be learned together in multi-task learning? In *International conference on machine learning*, pp. 9120–9132. PMLR, 2020.

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

Alane Suhr, Stephanie Zhou, Ally Zhang, Iris Zhang, Huajun Bai, and Yoav Artzi. A corpus for reasoning about natural language grounded in photographs, 2019. URL https://arxiv.org/abs/1811.00491.

Oyvind Tafjord, Matt Gardner, Kevin Lin, and Peter Clark. QuaRTz: An open-domain dataset of qualitative relationship questions. In Kentaro Inui, Jing Jiang, Vincent Ng, and Xiaojun Wan (eds.), *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pp. 5941–5946, Hong Kong, China, November 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-1608. URL https://aclanthology.org/D19-1608/.

Anke Tang, Li Shen, Yong Luo, Nan Yin, Lefei Zhang, and Dacheng Tao. Merging multi-task models via weight-ensembling mixture of experts. In *Forty-first International Conference on Machine Learning*, 2024a. URL https://openreview.net/forum?id=nLRKnO74RB.

Anke Tang, Li Shen, Yong Luo, Yibing Zhan, Han Hu, Bo Du, Yixin Chen, and Dacheng Tao. Parameter-efficient multi-task model fusion with partial linearization. In *The Twelfth International Conference on Learning Representations*, 2024b.

Simon Vandenhende, Stamatios Georgoulis, Wouter Van Gansbeke, Marc Proesmans, Dengxin Dai, and Luc Van Gool. Multi-task learning for dense prediction tasks: A survey. *IEEE transactions on pattern analysis and machine intelligence*, 44(7):3614–3633, 2021.

Ramakrishna Vedantam, C. Lawrence Zitnick, and Devi Parikh. Cider: Consensus-based image description evaluation. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4566–4575, 2015. doi: 10.1109/CVPR.2015.7299087.

Bastiaan S Veeling, Jasper Linmans, Jim Winkens, Taco Cohen, and Max Welling. Rotation equivariant CNNs for digital pathology. In *International Conference on Medical Image Computing and Computer Assisted Intervention (MICCAI)*, 2018. URL http://arxiv.org/abs/1806.03962v1.

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. URL https://openreview.net/forum?id=DWT9uiGjxT.

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

Wenhui Wang, Hangbo Bao, Li Dong, Johan Bjorck, Zhiliang Peng, Qiang Liu, Kriti Aggarwal, Owais Khan Mohammed, Saksham Singhal, Subhojit Som, and Furu Wei. Image as a foreign language: Beit pretraining for all vision and vision-language tasks, 2022. URL https://arxiv.org/abs/2208.10442.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, et al. Huggingface's transformers: State-of-the-art natural language processing. *arXiv preprint arXiv:1910.03771*, 2019.

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 *Proceedings of the International Conference on Machine Learning (ICML)*, pp. 23965–23998, 2022.

Haixia Xiao, Feng Zhang, Zhongping Shen, Kun Wu, and Jinglin Zhang. Classification of weather phenomenon from images by using deep convolutional neural network. *Earth and Space Science*, 8(5):e2020EA001604, 2021.

Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. *arXiv*, 2017. URL http://arxiv.org/abs/1708.07747v2.

Jianxiong Xiao, James Hays, Krista A Ehinger, Aude Oliva, and Antonio Torralba. Sun database: Large-scale scene recognition from abbey to zoo. In *2010 IEEE computer society conference on computer vision and pattern recognition*, pp. 3485–3492. IEEE, 2010.

Prateek Yadav, Derek Tam, Leshem Choshen, Colin A Raffel, and Mohit Bansal. Ties-merging: Resolving interference when merging models. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine (eds.), *Advances in Neural Information Processing Systems*, volume 36, pp. 7093–7115. Curran Associates, Inc., 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/1644c9af28ab7916874f6fd6228a9bcf-Paper-Conference.pdf.

Enneng Yang, Li Shen, Zhenyi Wang, Guibing Guo, Xiaojun Chen, Xingwei Wang, and Dacheng Tao. Representation surgery for multi-task model merging. In Ruslan Salakhutdinov, Zico Kolter, Katherine Heller, Adrian Weller, Nuria Oliver, Jonathan Scarlett, and Felix Berkenkamp (eds.), *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. URL https://proceedings.mlr.press/v235/yang24t.html.

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. URL https://openreview.net/forum?id=nZP6NgD3QY.

Yi Yang, Wen-tau Yih, and Christopher Meek. WikiQA: A challenge dataset for open-domain question answering. In Lluís Màrquez, Chris Callison-Burch, and Jian Su (eds.), *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pp. 2013–2018, Lisbon, Portugal, September 2015. Association for Computational Linguistics. doi: 10.18653/v1/D15-1237. URL https://aclanthology.org/D15-1237/.

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.

Netzer Yuval. Reading digits in natural images with unsupervised feature learning. In *Proceedings of the NIPS Workshop on Deep Learning and Unsupervised Feature Learning*, 2011.

Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. Hellaswag: Can a machine really finish your sentence? *arXiv preprint arXiv:1905.07830*, 2019.

Mingyang Zhang, Jing Liu, Ganggui Ding, Linlin Ou, Xinyi Yu, and Bohan Zhuang. Channel merging: Preserving specialization for merged experts. *Proceedings of the AAAI Conference on Artificial Intelligence*, 39(21):22479–22487, Apr. 2025. doi: 10.1609/aaai.v39i21.34405. URL https://ojs.aaai.org/index.php/AAAI/article/view/34405.

Yuan Zhang, Jason Baldridge, and Luheng He. PAWS: Paraphrase adversaries from word scrambling. In Jill Burstein, Christy Doran, and Thamar Solorio (eds.), *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pp. 1298–1308, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1131. URL https://aclanthology.org/N19-1131/.

Ziyu Zhao, Tao Shen, Didi Zhu, Zexi Li, Jing Su, Xuwu Wang, and Fei Wu. Merging loras like playing lego: Pushing the modularity of lora to extremes through rank-wise clustering. In *The Thirteenth International Conference on Learning Representations*, 2025.