Localizing Task Information for Improved Model Merging and Compression

Ke Wang [*1] **Nikolaos Dimitriadis** [*1] **Guillermo Ortiz-Jiménez** [2 3] **François Fleuret** [4] **Pascal Frossard** [1]

Abstract

Model merging and task arithmetic have emerged as promising scalable approaches to merge multiple single-task checkpoints to one multi-task model, but their applicability is reduced by significant performance loss. Previous works have linked these drops to interference in the weight space and erasure of important task-specific features. Instead, in this work we show that the information required to solve each task is still preserved after merging as different tasks mostly use non-overlapping sets of weights. We propose TALL-masks, a method to identify these task supports given a collection of task vectors and show that one can retrieve $> 99\%$ of the single task accuracy by applying our masks to the multi-task vector, effectively compressing the individual checkpoints. We study the statistics of intersections among constructed masks and reveal the existence of *selfish* and *catastrophic* weights, i.e., parameters that are important exclusively to one task and irrelevant to all tasks but detrimental to multi-task fusion. For this reason, we propose Consensus Merging, an algorithm that eliminates such weights and improves the general performance of existing model merging approaches. Our experiments in vision and NLP benchmarks with up to 20 tasks, show that Consensus Merging consistently improves existing approaches. Furthermore, our proposed compression scheme reduces storage from 57Gb to 8.2Gb while retaining 99.7% of original performance.

1. Introduction

In recent years, the field of ML has witnessed a paradigm shift with the release of foundation models and the influx of associated checkpoints, significantly improving the performance on downstream applications (Devlin et al., 2019; Ilharco et al., 2022; Wortsman et al., 2022b; Pruksachatkun et al., 2020; Zhou et al., 2023). The widespread adoption of foundation models has followed a proliferation of works addressing practical challenges arising from their sheer computational and storage requirements. For example, parameter-efficient fine-tuning (Hu et al., 2022; Liu et al., 2022), quantization (Dettmers et al., 2022; 2023) address aspects of training, fine-tuning and inference. An important question remains how to efficiently leverage the existing fine-tuned models towards improving models and building generalist agents (Reed et al., 2022).

Recent work has illuminated the benefits of interpolating the weights of different models (Frankle et al., 2020; Ilharco et al., 2022; Wortsman et al., 2022a; Ortiz-Jimenez et al., 2023; Dimitriadis et al., 2023), offering scalable and effective techniques to edit the knowledge of pre-trained models. Task arithmetic (TA) (Ilharco et al., 2023) has emerged as a promising solution to fuse the knowledge of disparate checkpoints into a single model with multi-objective capabilities, forgoing the need for additional joint training (Caruana, 1997) or optimizing over the exponentially large number of task combinations (Standley et al., 2020; Fifty et al., 2021). Prior studies have proposed more involved merging techniques by resolving *weight* interference (Yadav et al., 2023b), matching activations (Jin et al., 2023) or by preserving task-specific important parameters (Matena & Raffel, 2022; Tam et al., 2024). Despite these recent advances, weight space interpolation for multi-task fusion still suffers from significant drops in performance compared to individual fine-tuned models.

In this paper, we present a novel view and show that performance of the merged model can degrade even without weight interference or information erasure through a controlled experiment. In contrast, the discriminant information for individual tasks is preserved and embedded in the multi-task vector after merging disparate task vectors, and we propose an algorithm, `TALL-masks`, that identifies the subset of important parameters for each task

[*]Equal contribution [1]École Polytechnique Fédérale de Lausanne, Lausanne, Switzerland [2]Google Deepmind [3]Work done while at EPFL. [4]University of Geneva, Geneva, Switzerland.. Correspondence to: Ke Wang <k.wang@epfl.ch>, Nikolaos Dimitriadis <nikolaos.dimitriadis@epfl.ch>.

Proceedings of the 41st International Conference on Machine Learning, Vienna, Austria. PMLR 235, 2024.

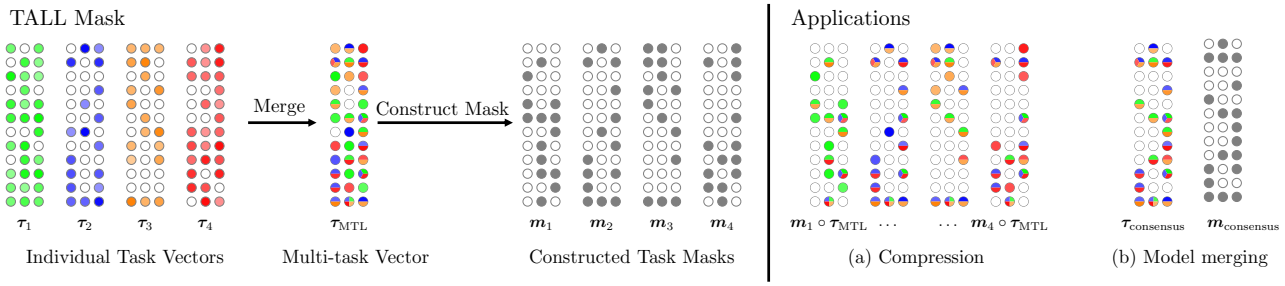


Figure 1. Illustration of our mask construction algorithm (left) along with the applications (right) on model compression and model merging. Each block corresponds to the same weight matrix, and color intensity reflects the value of each parameter – empty means zero value. Given single-task vectors $\{\boldsymbol{\tau}_t\}_{t=1}^4$ and the merged vector $\boldsymbol{\tau}_{\text{MTL}}$, our method constructs per-task masks $\{\boldsymbol{m}_t\}_{t=1}^4$, pinpointing the important parameters for each original task vector. For *model merging*, we keep only the '*general*' weights selected by more than one mask and produce the consensus mask $\boldsymbol{m}_{\text{consensus}}$ and the final merged vector. For compression, we evaluate on each task with reconstructed task vectors by masking out the irrelevant weights, retaining almost full performance without saving the individual task vectors.

(Panigrahi et al., 2023; Dai et al., 2022; Bayazit et al., 2023). We cast the problem of localizing important information as approximating each original task vector via erasing task-irrelevant information in the merged multi-task vector with a data-driven way, resulting in the construction of task-specific binary masks. We study the statistics of mask agreements among tasks, and reveal the existence of *catastrophic* and *selfish* weights, i.e., parameters that are deemed important by none and exclusively one task, respectively.

We then propose Consensus Merging, a method that utilizes the constructed masks to eliminate the *catastrophic* and *selfish* weights and is complementary to existing model merging approaches. Through extensive experimental validation, we show that our proposed Consensus Merging consistently improves prior methods. For instance, building upon task arithmetic (Ilharco et al., 2023) yields 4.9% gain in absolute average accuracy on a 20 task vision benchmark, while we improve TIES (Yadav et al., 2023b) by 6.3% on an 8-task NLP benchmark.

We also employ the constructed masks towards compressing the individually fine-tuned checkpoints. Motivated by our findings that task-specific information is preserved and by virtue of the masks, we can localize the knowledge of each task in the merged vector and extract it to approximate the original single-task vector. We compress the collection of checkpoints to the zero-shot model, the merged task vector and binary masks. Our experimental validation shows that our algorithms retains $> 99\%$ of original performance in various vision settings, ranging from small to large ViTs (Dosovitskiy et al., 2021) and benchmarks from 8 to 20 tasks, showing remarkable robustness to the increase in number of tasks. For instance, in a vision benchmark we compress 20 fine-tuned models from 57Gb to 8.2Gb retaining 99.7% of performance, while model merging methods almost reset to mere zero-shot performance. [1]

In short, our contributions are the following:

- We show that the task-specific information is preserved after merging, but task arithmetic cannot properly utilize it due to *task interference*. We provide an efficient algorithm, `TALL-masks`, to localize the task-specific information in the multi-task vector, which deactivates irrelevant parts for each task in the merged multi-task vector with binary masks.
- With the constructed task-specific masks, we are able to eliminate task interference and compress multiple fine-tuned checkpoints to only the zero-shot model, the merged task vector and the aforementioned binary masks while preserving the performance of individual models.
- We analyze the profile of mask agreements and identify the existence of weights deemed important by only one task or even none. We then propose Consensus Merging, a model merging method that eliminates these *selfish* and *catastrophic* weights, keeping only *general* weights. Our method can be combined with existing approaches, such as Task Arithmetic or TIES, and consistently improve over them, showing better robustness to increasing number of tasks.
- We perform extensive evaluation on Computer Vision and NLP benchmarks and show the benefits of our proposed methods. For model merging, our Consensus Merging consistently improves prior merging methods, setting state-of-the-art results. For compression, we achieve $> 99\%$ performance retention in across all

[1]The source code can be found at https://github.com/nik-dim/tall_masks.

vision benchmarks and model sizes while requiring much less storage compared to the original collection of fine-tuned checkpoints.

2. Related Work

Weight Interpolation and Model Merging Model editing directly in the weight space has attracted a lot of attention in recent years with many works showing that interpolating the weights of different models results in low-loss paths (Garipov et al., 2018; Draxler et al., 2018; Frankle et al., 2020). Wortsman et al. (2021) enacted on these insights and trained a weight ensemble from scratch, showing better generalization (Foret et al., 2021; Chaudhari, 2018) for the midpoint, while Dimitriadis et al. (2023) extended these ideas to Multi-Task Learning and showed that linear weight subspaces can encode tradeoffs and map to the Pareto Front. While these works focus on end-to-end training, Ilharco et al. (2023) studied pre-trained models and observed that arithmetic operations among fine-tuned weights generate similar functional responses and allow for a scalable framework to endow multi-objective capabilities. Several approaches have improved this idea by performing merging guided by various heuristics (Davari & Belilovsky, 2023; Luo et al., 2023; Jin et al., 2023), such as resolving interference due to redundant parameter values and sign disagreements (Yadav et al., 2023b), by preserving the important parameters defined via the Fisher Information Matrix (Matena & Raffel, 2022; Tam et al., 2024), or by learning the model merging weights with unlabeled test data (Yang et al., 2024). Ortiz-Jimenez et al. (2023) offered more theoretical foundations on the field of *model merging* and identified *weight disentanglement* as the necessary condition for task arithmetic, while showing that performing fine-tuning on a linearized model leads to improved model merging.

Reducing complexity for foundation models The capabilities of foundation models have commanded the development of methods that address their computational and memory requirements. Parameter-efficient fine-tuning (PEFT) (Hu et al., 2022; Liu et al., 2022; Houlsby et al., 2019) approaches heavily reduce the number of trainable parameters, enabling efficient adaptation. Several works (Dettmers et al., 2023; Liang et al., 2021) perform quantization after training and reduce the memory footprint of the model by requiring less bits to represent each parameter. ComPEFT (Yadav et al., 2023a) addresses an orthogonal issue, namely communication costs in expert model merging, by compressing fine-tuning residuals via sparsification and quantization. Task Arithmetic can also be viewed from a compression standpoint; multiple functionally diverse models are combined into one, but severe performance degradation is observed. BYOM (Jiang et al., 2024) sparsifies the residuals before merging but the performance heavily depends on the chosen sparsity level. In this paper, we introduce a mechanism that addresses this drop while significantly compressing the multiple initial checkpoints. Mixture of Experts (MoE) architectures (Shazeer et al., 2017; Riquelme et al., 2021), where different sub-networks specialize in various tasks or input regions, offer an effective approach for handling diverse objectives. However, training such models from scratch can be complex (Chen et al., 2022; Fedus et al., 2022). This work explores an alternative approach, leveraging the power of pre-trained models and task-specific information localization to create expert models, potentially streamlining MoE development.

3. Task Interference Causes Performance Degradation

We consider the case of T tasks, where training for each task t starts from pre-trained model $\boldsymbol{\theta}_0$ and fine-tunes on $\mathcal{D}_t^{\text{train}}$ to obtain $\boldsymbol{\theta}_t$. Task arithmetic (Ilharco et al., 2023) merges the fine-tuned checkpoints by decoupling the contributions of the zero-shot model and operating on the space of residuals or *task vectors* $\boldsymbol{\tau}_t = \boldsymbol{\theta}_t - \boldsymbol{\theta}_0$, generating the *multi-task vector* through simple summation: $\boldsymbol{\tau}_{\text{MTL}} = \sum_{t\in[T]} \boldsymbol{\tau}_t$. The final multi-task model corresponds to $\boldsymbol{\theta} = \boldsymbol{\theta}_0 + \alpha\boldsymbol{\tau}_{\text{MTL}}$, where $\alpha > 0$ is a scaling factor tuned on a held-out validation set.

While task arithmetic offers a computationally cheap way to fuse multiple fine-tuned checkpoints, it suffers from significant performance drops compared to the single-task counterparts. Previous works have attributed the performance drop to loss of valuable task-specific information due to parameter interference during merging process (Yadav et al., 2023b). To better understand the causes for performance degradation, we make two hypotheses:

- *Information erasure*: large amount of information specific to each task is erased when merging the *multi-task vector*.
- *Task interference*: the task-specific information is preserved in the *multi-task vector*, but can not manifest properly due to interference between the tasks.

To validate these hypotheses, we start with a controlled experiment where information erasure would not happen. Specifically, for the 8-task vision benchmark proposed by Ilharco et al. (2023), we randomly select a subset of weights for each task and perform gradient updates only for those parameters. Hence, task vectors $\{\boldsymbol{\tau}_t\}_{t=1}^{8}$ form a partition of the weight space, where all non-overlapping subsets are of equal size. By design, parameter interference (Yadav et al., 2023b) is nonexistent and tasks do not compete for important parameters (Matena & Raffel, 2022; Tam et al., 2024). Importantly, all the task-relevant information is preserved inside the *multi-task vector*.

Table 1. Performance comparison between standard and non-overlapping fine-tuning, averaged over 8 tasks (Ilharco et al., 2023); the lack of *weight interference* and the preservation of all task-specific knowledge in the controlled experiment is not beneficial for task arithmetic.

		Abs. acc.	Norm. acc.
Fine-tuning method	Standard	92.8	100.0
	Controlled	89.8	100.0
Task arithmetic accuracy	Standard	**71.5**	**77.0**
	Controlled	68.8	76.6

The results for this control experiment are presented in Table 1, compared with task arithmetic where the models are fine-tuned in a standard way. Looking at the normalized accuracy, defined in Appendix A, we observe that the performance of task arithmetic in the controlled setting deteriorates at the same rate as standard fine-tuning, where the accuracy of the merged model is 2.7% worse than standard case. This suggests that, even when the task-specific knowledge is perfectly preserved inside the *multi-task vector*, task arithmetic fails to properly utilize the relevant information to restore the fine-tuned performance. It hints that *task interference* is the culprit for the performance decline of task arithmetic rather than *weight interference*. Specifically, while task-specific parameters remain constant, alterations in other tasks lead to changes in discriminating features for that task, perturbing the mapping from the task's input distribution to output.

4. `TALL-masks`: Localizing Task-specific Information in *multi-task vector*

In the controlled experiment, the fine-tuning performance can be easily restored by localizing task-specific information in the *multi-task vector* with the masks used for fine-tuning. Now we shift our focus to the general setting of standard fine-tuning and investigate the percentage of information preserved after merging.

We formulate the problem of localization of task-specific knowledge (Panigrahi et al., 2023; Dai et al., 2022) as extracting relevant weight subsets from the *multi-task vector* with binary masks, such that the extracted weights approximate the original task vector $\boldsymbol{\tau}_t$. The binary mask deactivates irrelevant weights in *multi-task vector* while keeping only the task-specific information. Our algorithm, `TALL-masks` for TAsk LocaLization Masks, constructs masks $\boldsymbol{m}_t$ targeting to construct $\hat{\boldsymbol{\theta}}_t$ such that:

$$\hat{\boldsymbol{\theta}}_t = \boldsymbol{\theta}_0 + \boldsymbol{m}_t \circ \boldsymbol{\tau}_{\text{MTL}} \approx \boldsymbol{\theta}_t \tag{1}$$

We minimize the ℓ_1 distance between the reconstructed $\hat{\boldsymbol{\theta}}_t$ and fine-tuned model $\boldsymbol{\theta}_t$:

$$\begin{aligned} \boldsymbol{m}_t^* &= \underset{\boldsymbol{m}_t \in \{0,1\}^P}{\text{argmin}} \|\hat{\boldsymbol{\theta}}_t - \boldsymbol{\theta}_t\|_1 && (2) \\ &= \underset{\boldsymbol{m}_t \in \{0,1\}^P}{\text{argmin}} \|\boldsymbol{m}_t \circ \boldsymbol{\tau}_{\text{MTL}} - \boldsymbol{\tau}_t\|_1 && (3) \\ &= \mathbb{1}\{|\boldsymbol{\tau}_t| \geq |\boldsymbol{\tau}_{\text{MTL}} - \boldsymbol{\tau}_t|\} && (4) \end{aligned}$$

Here P stands for the total number of parameters, the detailed derivation are given in Appendix B. Furthermore, we add a hyper-parameter λ_t on the right hand side of Equation 4 to tune the amount of information for the mask to extract from *multi-task vector*; the smaller λ_t, the more parameters get selected by $\boldsymbol{m}_t$. Finally, we construct the task-specific masks based on:

$$\boldsymbol{m}_t = \mathbb{1}\left\{|\boldsymbol{\tau}_t| \geq |\boldsymbol{\tau}_{\text{MTL}} - \boldsymbol{\tau}_t| \cdot \lambda_t\right\} \tag{5}$$

Note that λ_t is selected based on the validation accuracy of each task respectively, allowing for the task-specific problems to be solved in parallel and independently.

We validate the efficacy of our mask construction by checking if the original performance in the same 8-task computer vision benchmark, evaluated on a held-out dataset, can be restored. Specifically, we construct the masks for each dataset via Equation 5 for the benchmark proposed by Ilharco et al. (2023), and evaluate with reconstructed models as in Equation 1. Figure 2 confirms that full performance can be retained by simply deactivating irrelevant parameter subsets with binary masks. Thus, it shows that all the information embedded in the original checkpoints is not erased but rather preserved in the *multi-task vector*.

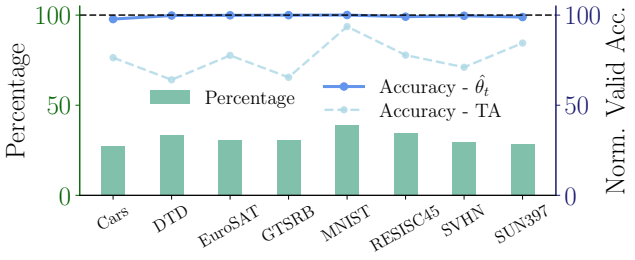


Figure 2. `TALL-masks` localizes task-specific information. The bar plot shows the percentage of parameters selected by `TALL-masks`, while the blue line shows the normalized validation accuracy achieved by the re-constructed $\hat{\boldsymbol{\theta}}_t$ with the selected masks using Equation 1. The lightblue dashed line shows the task arithmetic baseline where the information is not localized. Our task-specific masks allow the restoration of full performance, showing that all knowledge embedded in the initial fine-tuned checkpoints is preserved post merging.

5. Applications

Based on these observations, we present two application scenarios of the masking algorithm for compressing the task vectors and improving model merging methods.

5.1. Compressing Task Vectors

Motivated by the previous results, we employ the masks for compressing the fine-tuned checkpoints. Since full performance can be retained by the constructed models using the masks, it allows us to significantly reduce the required storage cost without sacrificing performance.

Specifically, instead of the collection of fine-tuned checkpoints $\{\boldsymbol{\theta}_t\}_{t=1}^T$, we can save only the pre-trained model $\boldsymbol{\theta}_0$, the *multi-task vector* $\boldsymbol{\tau}_{\text{MTL}}$ and the individual task-specific binary masks $\{\boldsymbol{m}_t\}_{t=1}^T$. For evaluation on task t, we construct a specialized model by adding to the pre-trained only task-relevant subsets from the *multi-task vector*:

$$\hat{\boldsymbol{\theta}}_t = \boldsymbol{\theta}_0 + \boldsymbol{m}_t \circ \boldsymbol{\tau}_{\text{MTL}} \tag{6}$$

In this way, it allows significant compression while maintaining the majority of performance for fine-tuned models without saving the individual checkpoints. For example, it requires only 13.7% of storage compared to saving the ViT-L/14 checkpoints for a 20-task benchmark. We provide the details for storage comparison in Appendix C.

5.2. Improving Model Merging

While the storage of task-specific masks introduces extra storage cost compared with Task Arithmetic, we present here another application of `TALL-masks`, Consensus Merging, which improves over model merging methods without requiring extra storage.

The construction of the task-specific masks $\{\boldsymbol{m}_t\}_{t=1}^T$ allows us to investigate the relevance of parameters in the *multi-task vector* to each task, where we assume that if a parameter is included in a mask, it is relevant for the associated task.

We find that many parameters in the *multi-task vector* are relevant to only a subset of tasks. Let P be the total number of parameters, we define *mask agreement percentage* as the fraction of weights deemed important by exactly n out of T tasks:

$$\alpha(\{\boldsymbol{m}_t\}_{t=1}^T, n) = \frac{1}{P}\left\| \mathbb{1}\left\{ \sum_{t\in[T]} \boldsymbol{m}_t = n \right\} \right\|_0 \tag{7}$$

The histogram for the mask agreement percentage is shown in Figure 3, for both the *multi-task vector* merged with Task Arithmetic and TIES, while we provide in Appendix D.4 the cases for more tasks. We observe that there exists a subset of parameters which are used by no task at all, which we term

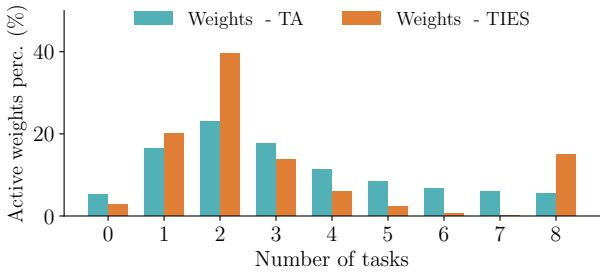


Figure 3. The distribution of mask agreements in the merged vector produced by two model merging methods, Task Arithmetic and TIES. A non-negligent fraction of weights is important exclusively to one task (selfish) while another fraction is irrelevant to all tasks (catastrophic). Our method eliminates both categories to improve model merging.

catastrophic weights as their existence would only introduce unnecessary interference and hurt the performance of model merging. Furthermore, we also identify there exists a non-negligible fraction of weights used by only one task, which we term *selfish* weights, as their existence only benefits one task whereas causing *task interference* for all other tasks. We term the rest of weights as *general* weights, as they are relevant to at least two tasks and their importance grows with the number of relevant tasks. Similarly, we term *universal* the weights deemed important for all tasks ($n = T$).

Based on these observations, we present Consensus Merging, which is targeted to reduce *task interference* for better model merging. Formally, we form the *consensus mask* for threshold $k \in \{0, \dots, T\}$ as:

$$\boldsymbol{m}_{\text{consensus}} = \mathbb{1}\left\{ \sum_{t\in[T]} \boldsymbol{m}_t \geq k \right\}, \tag{8}$$

and filter the *multi-task vector* through a Hadamard product:

$$\boldsymbol{\tau}_{\text{consensus}} = \boldsymbol{m}_{\text{consensus}} \circ \boldsymbol{\tau}_{\text{MTL}}, \tag{9}$$

where k in Equation 8 is defined as the weight-pruning threshold, e.g., the minimal number of activated masks for preventing the weights from being pruned. By setting $k = 2$ in Equation 8 we can eliminate both *catastrophic* weights and *selfish* weights in the *multi-task vector* to reduce *task interference*, keeping only *general* weights that are globally important to at least two tasks. The *threshold* k affects performance and depends on the task number and combination, as well as the underlying model merging method, as Figure 3 showcases the different *mask agreement profiles* for Task Arithmetic and TIES. In the following, we use $k = 2$, unless specified otherwise.

Finally, we note that both the proposed compression method and Consensus Merging are orthogonal to existing model

Table 2. Comparison of model merging (top) and compression (bottom) methods on three sets of NLP benchmarks with a T5-large model, in terms of accuracy, absolute and normalized (in parentheses), as well as storage cost.

	Method	7 NLP tasks (Yadav et al., 2023b)		8 QA tasks (Zhou et al., 2022)		All 11 tasks	
		Acc.(%) ↑	Bits(Gb) ↓	Acc.(%) ↑	Bits(Gb) ↓	Acc.(%) ↑	Bits(Gb) ↓
Merging	Zero-shot	44.9	25.1	33.1	25.1	36.9	25.1
	Weight averaging	60.5 (72.7)	25.1	56.4 (69.9)	25.1	55.2 (71.6)	25.1
	Task arithmetic	71.9 (85.3)	25.1	63.8 (79.6)	25.1	63.6 (81.8)	25.1
	TIES	69.6 (83.5)	25.1	62.8 (78.9)	25.1	64.0 (82.6)	25.1
	Consensus TA [ours]	**73.5** (**87.7**)	25.1	68.6 (85.4)	25.1	**67.5** (**86.8**)	25.1
	Consensus TIES [ours]	71.0 (84.2)	25.1	**69.1** (**85.8**)	25.1	66.8 (85.7)	25.1
Compression	Fine-tuned	85.9	169.1	80.7	193.1	78.7	265.1
	Magnitude Pruning	81.6 (93.5)	>54.3	70.8 (85.7)	>55.1	65.8 (81.4)	>57.3
	Magnitude Masking	78.9 (90.7)	54.3	72.5 (88.8)	55.1	69.8 (87.2)	57.3
	TALL Mask + TA [ours]	**86.8** (**102.2**)	54.3	79.6 (98.7)	55.1	76.5 (96.2)	57.3
	TALL Mask + TIES [ours]	83.4 (95.4)	54.3	**79.7** (**98.8**)	55.1	**77.4** (**97.5**)	57.3

merging approaches and can be easily plugged in since they operate on the *multi-task vector* τ_{MTL}, e.g., τ_{MTL} can be produced by Task Arithmetic (Ilharco et al., 2023), TIES (Yadav et al., 2023b) or other algorithms (Matena & Raffel, 2022; Tam et al., 2024). Practitioners can toggle between these two applications depending on the usage scenario.

6. Experiments

6.1. Model Merging

Baselines We compare Consensus Merging with several train-free model-merging methods, including weight averaging, Task Arithmetic (Ilharco et al., 2023), and TIES (Yadav et al., 2023a). Our method is complementary to all and we opt to validate its efficacy when combined with the latter two. Specifically, we term our methods Consensus Task Arithmetic and Consensus TIES when combined with them respectively. We include also individually fine-tuned models and the zero-shot model as higher and lower bounds on performance, respectively. We assess the performance based on both the averaged absolute accuracy, and normalized accuracy, defined in detail in Appendix A.

Natural Language Processing We explore NLP benchmarks following Yadav et al. (2023b); Tam et al. (2024). We use a variant of T5-large model (Raffel et al., 2020), T5-large-LM-Adapt (Lester et al., 2021), and evaluate our method on three sets of benchmarks studied in previous works, a 7-task NLP benchmark (Yadav et al., 2023b), an 8-task benchmark geared towards Question-Answering (Zhou et al., 2022), as well as their union amounting to 11 tasks overall. More details about the task composition for each benchmark are provided in Appendix A. We use the publicly released checkpoints from Tam et al. (2024).

Table 2 presents the results for all NLP benchmarks. Consensus Merging consistently improves over both Task Arithmetic and TIES, leading to significant performance gains across settings. For example, in the 8-QA benchmark, our methods respectively improve over Task arithmetic by 4.8% and TIES by 6.3% in absolute accuracy, while performance is enhanced by over 2.9% and 2.8%, respectively, for the 11-task benchmark. We also conduct experiments with checkpoints finetuned with parameter-efficient methods in $(IA)^3$ (Liu et al., 2022); Appendix D.3 shows that our proposed method again results in significant gains over Task Arithmetic and TIES.

Computer Vision We consider 3 test scenarios, where the number of tasks in each scenario increases gradually from 8 to 14 and 20. The 8-task benchmark coincides with the experimental setup originally introduced by Ilharco et al. (2023) and is expanded to further illuminate the effect of larger number of tasks. The full details for the benchmarks are provided in Appendix A.3. For each test scenario, we assess the efficacy of our method on three CLIP model variants (Radford et al., 2021) with ViT-B/32, ViT-B/16, and ViT-L/14 as visual encoders (Dosovitskiy et al., 2021). All methods use the same checkpoints, fine-tuned with the setting outlined in Ilharco et al. (2023).

The results in image classification are shown in Table 3. Similar to NLP, Consensus Merging provides the best results in 5 out of 6 scenarios, with its superiority becoming increasingly apparent as both the number of tasks and model size grow. This pattern is also noted for the ViT-B/16, as detailed in Table 7 in Appendix D.5. Our algorithm offers consistent enhancements over Task Arithmetic, while its advantages over TIES are most pronounced in larger models. For example, in the most extensive evaluation involving 20 tasks with a ViT-L/14 encoder, our approach improves on Task Arithmetic and TIES by 4.9% and 1.1%, respectively.

Table 3. Comparison of model merging (top) and compression (bottom) methods across three test scenarios in image classification with different ViT encoders for CLIP models, in terms of accuracy, absolute and normalized (in parentheses), as well as storage cost (in Gb).

	Method	ViT-B/32						ViT-L/14					
		8 tasks		14 tasks		20 tasks		8 tasks		14 tasks		20 tasks	
		Acc.(%) ↑	Bits↓	Acc.(%) ↑	Bits↓	Acc.(%) ↑	Bits↓	Acc.(%) ↑	Bits↓	Acc.(%) ↑	Bits↓	Acc.(%) ↑	Bits↓
Merging	Zeroshot	48.4	3.6	57.3	3.6	56.1	3.6	64.4	11.0	68.0	11.0	65.1	11.0
	Weight averaging	66.5 (72.3)	3.6	64.4 (71.2)	3.6	61.1 (67.5)	3.6	79.4 (83.0)	11.0	76.6 (81.0)	11.0	71.5 (75.5)	11.0
	Task arithmetic	70.8 (76.5)	3.6	65.4 (72.2)	3.6	60.6 (66.8)	3.6	84.8 (88.5)	11.0	79.3 (83.8)	11.0	74.0 (78.0)	11.0
	TIES	**75.1** (**81.0**)	3.6	68.0 (74.8)	3.6	63.4 (69.9)	3.6	**86.9** (**90.7**)	11.0	79.5 (84.1)	11.0	75.7 (79.8)	11.0
	Consensus TA [ours]	75.0 (80.8)	3.6	**70.4** (**77.4**)	3.6	**65.4** (**72.0**)	3.6	86.2 (89.9)	11.0	**82.2** (**86.9**)	11.0	**78.9** (**83.2**)	11.0
	Consensus TIES [ours]	74.8 (80.6)	3.6	67.7 (74.5)	3.6	63.2 (69.6)	3.6	**86.9** (**90.7**)	11.0	81.5 (86.1)	11.0	76.8 (80.9)	11.0
Compression	Fine-tuned	92.8	23.3	90.9	40.2	91.3	57.0	95.8	79.1	94.3	137.4	94.7	195.8
	Magnitude Pruning	91.3 (98.4)	>7.1	85.3 (93.7)	>7.7	83.4 (91.2)	>8.2	95.4 (99.6)	>23.1	91.8 (97.2)	>24.9	91.2 (96.1)	>26.8
	Magnitude Masking	86.8 (93.3)	7.1	80.7 (88.4)	7.7	75.3 (82.1)	8.2	94.6 (98.7)	23.1	91.6 (97.0)	24.9	91.6 (96.5)	26.8
	TALL Mask + TA [ours]	92.6 (99.7)	7.1	90.1 (99.1)	7.7	90.6 (99.2)	8.2	95.7 (99.9)	23.1	93.1 (98.8)	24.9	93.7 (98.9)	26.8
	TALL Mask + TIES [ours]	**93.0** (**100.3**)	7.1	**90.9** (**100.0**)	7.7	**91.1** (**99.7**)	8.2	**95.9** (**100.1**)	23.1	**93.4** (**99.0**)	24.9	**93.9** (**99.1**)	26.8

6.2. Compression

We adopt the compression technique discussed in Section 5.1 to compress the individual checkpoints, and use the prefix '*TALL Mask +*' when combined with different model merging methods.

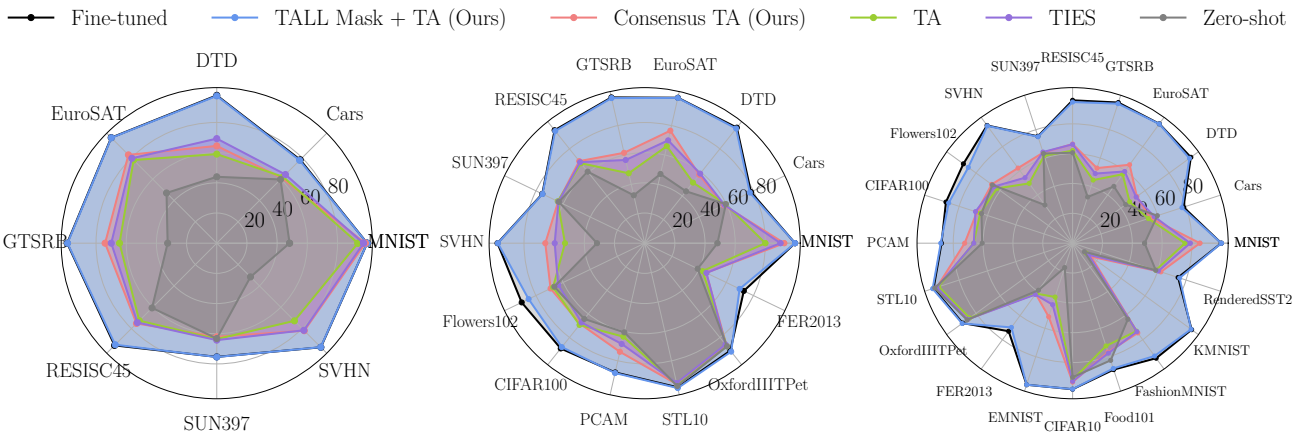
Baselines We compare against methods on the same level of storage as our proposed solution. We first consider Magnitude Masking, where per-task masks are constructed by replacing our procedure in Equation 4 by keeping the top $k\%$ of the parameters. After experimental validation, we found that $k = 10$ works well in general and therefore use it throughout. We also consider unstructured Magnitude Pruning of the individual task vectors, where we set the level of pruning so that the storage cost of one pre-trained model and T pruned task vectors is equal to ours. Note that, in practice, Magnitude Pruning takes more storage than our method as we do not consider the storage cost for the positions of the parameters.

Natural Language Processing Following the same experimental setting for merging, the results are presented in the bottom half of Table 2. Our compression scheme effectively retains performance while requiring much less storage cost than storing individual fine-tuned models. In contrast, magnitude-based solutions suffer from severe performance losses, especially as the number of tasks increases. For example, for the 7-task benchmark, our method keeps all the performance from the fine-tuned models with even over 100% normalized accuracy, while requiring less than 1/3 storage cost than storing the checkpoints. The advantage on storage compression becomes more pronounced with larger number of tasks. For example, for the 11-task benchmark, both our methods require only around 1/5 of the storage of individual checkpoints, while keeping at least 96.2% of the performance. By keeping the same level of storage, separately pruned models preserve lower number of parameters and the cost in performance mirrors this lack of task-specific features; for the 11-task benchmark only 81.4% of performance is retained compared to our performance of 97.5%. In contrast, our method takes advantage of the redundancy in the task vector to compress effectively with minimal performance losses.

Computer Vision The bottom half of Table 3 presents the results for compression in vision. While we observe that both Magnitude Pruning and Magnitude Masking show a clear performance degradation with increasing number of tasks, our proposed methods deliver almost full performance in all test scenarios.

Specifically, TALL Mask + Task Arithmetic achieves around 99% normalized accuracy on all cases, where there is almost no performance degradation with increasing number of tasks. TALL Mask + TIES performs even better, with its normalized accuracy being over 99% for all the test cases. For ViT-B/32, it achieves around 100% for all test scenarios without any loss of performance. For ViT-L/14, TALL Mask + TIES still performs exceptionally, with its normalized accuracy being around 100% for 8 tasks and over 99% for 14 and 20 tasks. These results show that our method is able to capture the crucial task-specific information buried in the merged vector and is not bound by model scale or number of tasks.

In terms of storage, TALL Mask + Task Arithmetic requires much less storage cost compared to storing the individual fine-tuned models, where the storage saving is more pronounced with a larger number of tasks. Importantly, our compression scheme with `TALL-masks`, provides



(a) 8 Tasks (b) 14 Tasks (c) 20 Tasks

Figure 4. Comparison of absolute accuracy (%) of individual tasks for the computer vision benchmarks and ViT-B/32. Results for ViT-B/16 and ViT-L/14 are provided in the appendix. Our Consensus Merging shows higher performance compared to model merging baselines, especially for the settings with more tasks. Our compression algorithm consistently matches the performance of the individual fine-tuned models at a fraction of the memory, while model merging techniques are not robust to the increase of tasks.

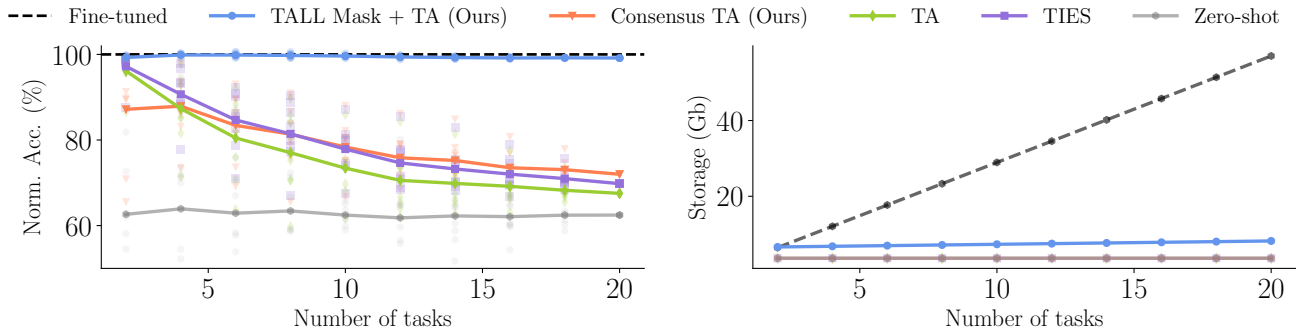


Figure 5. Averaged normalized accuracy *vs.* number of tasks for computer vision benchmarks. Our proposed specialist algorithm maintains initial performance regardless of task combination and heavily compresses the fine-tuned checkpoints.

an efficient trade-off between performance and storage compared to the model-merging methods. For example, using TALL Mask + Task Arithmetic on 20 tasks with ViT-B/32 takes 8.2 Gb storage while achieving 90.6% absolute accuracy, whereas using TIES on the same 20 tasks with ViT-L/14 takes 11.0 Gb but delivers an absolute accuracy of merely 75.7%. Overall, our method delivers a desirable trade-off in the Pareto Front of performance retention vs total storage cost.

6.3. Individual-task Performance

We now shift our focus from statistics over all tasks and present the performance on individual tasks. For vision settings, we compare the performance of TALL Mask + Task Arithmetic and Consensus Task Arithmetic with baselines methods, and plot in Figure 4 the individual accuracies on three benchmarks with ViT-B/32, while we provide the vision results with two other models as well as the results in NLP settings in Appendix D.6. We observe that TALL Mask + Task Arithmetic consistently delivers performance the same level as individual fine-tuned models, across datasets and total number of tasks.

On the other hand, the expansion of the considered tasks results in significant performance drop for model merging methods, where for some datasets the performance is even reset back to zero-shot model. Yet, we observe that Consensus Task Arithmetic suffers the least from increasing number of tasks and shines among the model merging methods, especially in the 20-task setting, where it outperforms the other model merging methods in almost all the datasets.

6.4. Performance with Varying Task Combinations

We perform a systematic study on the effect of different task combinations on our methods and model-merging baselines,

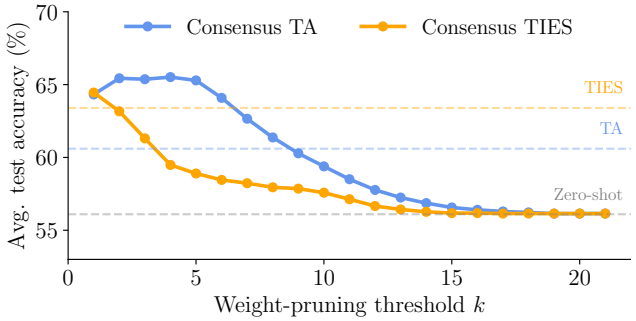


Figure 6. Performance of Consensus Merging with varying weight-pruning threshold k, on the 20-task image classification benchmark with ViT-B/32.

and present both the normalized accuracy and the storage cost with respect to different numbers of tasks in Figure 5. Due to the vast array of 2^{20} potential combinations, the performance is averaged on 8 carefully selected representative sample task combinations for each number of tasks; the details for selection process are given in Appendix E. Our observations indicate that our TALL Mask + Task Arithmetic consistently matches the performance of individually fine-tuned models across all combinations, regardless of the total task count. By comparison, model merging methods show a stark degradation in performance, especially given a large number of tasks where their performance degrade gradually to nearly zero-shot performance. Nonetheless, Consensus Task Arithmetic outperforms the two other model merging methods in general, while the performance gain gradually becomes apparent with larger number of tasks.

While the model merging methods have constant storage cost, their applicability is undermined by low performance. Conversely, maintaining individual models is excessively prohibitive in terms of storage but guarantees strong performance. Localizing the task-specific parameters with our proposed masks offers a favorable trade-off between performance and storage. Crucially, we deliver consistent performance with near 100% normalized accuracy across various task combinations and numbers of tasks, while heavily compressing the required storage cost, as we see in the right plot of Figure 5. It presents a more viable solution in scenarios where balancing performance with storage efficiency is essential.

6.5. Effect of Weight-Pruning Threshold

The weight-pruning threshold k is used to form the consensus mask in Equation 8 and, determines least number of activated tasks to prevent weights from being pruned. Extending the number of tasks, modifying the task combination and the model merging method itself affect the *mask agreement profile*, defined in Equation 7, and consequently the optimal pruning-threshold and the overall performance. We present the performance of Consensus Merging with ViT-B/32 on various image classification benchmarks in Figure 6 for the 20-task benchmark and Figure 7 in the appendix for the 8 and 14-task benchmarks when k gradually increases from 1 to over the total number of tasks. We observe from the figure that while the optimal performance of Consensus TA is usually achieved by setting $k = 2$, i.e., removing both *catastrophic* and *selfish* weights, Consensus TIES achieves its optimal performance by setting $k = 1$, i.e., removing only catastrophic weights. We present also the results for removing only catastrophic weights in Appendix D.1 for all test scenarios with Computer Vision, where we observe the performance of Consensus TIES consistently outperform TIES.

The difference in optimal thresholds k between Task Arithmetic and TIES originates in their profiles, as shown in Figure 3 and Figure 8; the pruning and sign resolution mechanisms of TIES shift the distribution towards increased *universal* and *selfish* weights. Removing the latter altogether results in a significant reduction of salient weights. Studying how different merging strategies affect the weight profiles remains an interesting future direction.

7. Conclusion

In this study, we introduced `TALL-masks`, a method with dual practical applications: it effectively addresses the significant issue of compressing foundation models and facilitates model merging. We identify the reason for performance degradation of model merging methods is not due to information erasure during merging process, but because of task interference during evaluation. Our research demonstrates that the multi-task vector retains crucial task-specific information, which `TALL-masks` localizes and extracts through the use of task-specific binary masks, thereby enabling the recovery of original fine-tuned performance levels. We utilize this observation to compress a collection of fine-tuned checkpoints into storing only the zero-shot model, the information-rich merged vector and the specified masks. Further, our examination of mask statistics uncovered weights harmful to model merging. We thus proposed their elimination and achieved state-of-the-art performance in common model merging benchmarks. Our proposed solution not only enhances the utility of foundation models but also makes them more accessible and sustainable, paving the way for broader applications and advancements in the field.

Impact Statement

This paper presents work whose goal is to advance the field of Machine Learning. There are many potential societal consequences of our work, none which we feel must be specifically highlighted here.

Acknowledgments

We thank Alessandro Favero, Thibault Sejourné and the anonymous reviewers for helpful feedback and comments.

References

Bayazit, D., Foroutan, N., Chen, Z., Weiss, G., and Bosselut, A. Discovering knowledge-critical subnetworks in pretrained language models. *arXiv*, 2023. URL http://arxiv.org/abs/2310.03084v1.

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

Caruana, R. Multitask Learning. *Machine Learning*, 28(1): 41–75, 1997.

Chaudhari, P. A. *A Picture of the Energy Landscape of Deep Neural Networks*. PhD thesis, 2018.

Chen, Z., Deng, Y., Wu, Y., Gu, Q., and Li, Y. Towards understanding the mixture-of-experts layer in deep learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022.

Cheng, G., Han, J., and Lu, X. Remote sensing image scene classification: Benchmark and state of the art. *Proceedings of the IEEE*, 2017. URL http://arxiv.org/abs/1703.00121v1.

Cimpoi, M., Maji, S., Kokkinos, I., Mohamed, S., and Vedaldi, A. Describing textures in the wild. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014. URL http://arxiv.org/abs/1311.3618v2.

Clanuwat, T., Bober-Irizar, M., Kitamoto, A., Lamb, A., Yamamoto, K., and Ha, D. Deep learning for classical japanese literature. *arXiv*, 2018. URL http://arxiv.org/abs/1812.01718v1.

Coates, A., Ng, A., and Lee, H. 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.

Cohen, G., Afshar, S., Tapson, J., and Van Schaik, A. EMNIST: Extending MNIST to handwritten letters. In *International Joint Conference on Neural Networks (IJCNN)*, 2017.

Dai, D., Dong, L., Hao, Y., Sui, Z., Chang, B., and Wei, F. Knowledge Neurons in Pretrained Transformers. In *Association for Computational Linguistics (ACL)*, 2022. URL https://aclanthology.org/2022.acl-long.581.

Davari, M. and Belilovsky, E. Model Breadcrumbs: Scaling Multi-Task Model Merging with Sparse Masks. *arXiv*, 2023. URL http://arxiv.org/abs/2312.06795v1.

Dettmers, T., Lewis, M., Belkada, Y., and Zettlemoyer, L. GPT3.int8(): 8-bit Matrix Multiplication for Transformers at Scale. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022. URL https://openreview.net/forum?id=dXiGWqBoxaD.

Dettmers, T., Pagnoni, A., Holtzman, A., and Zettlemoyer, L. Qlora: Efficient finetuning of quantized llms. *arXiv*, 2023. URL http://arxiv.org/abs/2305.14314v1.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *North American Chapter of the Association for Computational Linguistics (NAACL)*, 2019. https://aclanthology.org/N19-1423.

Dimitriadis, N., Frossard, P., and Fleuret, F. Pareto Manifold Learning: Tackling multiple tasks via ensembles of single-task models. In *International Conference on Machine Learning (ICML)*, 2023.

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., and Houlsby, N. An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. In *International Conference on Learning Representations (ICLR)*, 2021. URL http://arxiv.org/abs/2010.11929v2.

Draxler, F., Veschgini, K., Salmhofer, M., and Hamprecht, F. A. Essentially No Barriers in Neural Network Energy Landscape. In *International Conference on Machine Learning (ICML)*, 2018. URL http://arxiv.org/abs/1803.00885v5.

Fedus, W., Zoph, B., and Shazeer, N. Switch transformers: Scaling to trillion parameter models with simple and efficient sparsity. *Journal of Machine Learning Research (JMLR)*, 23(120):1–39, 2022. URL http://arxiv.org/abs/2101.03961v3.

Fifty, C., Amid, E., Zhao, Z., Yu, T., Anil, R., and Finn, C. Efficiently Identifying Task Groupings for Multi-Task Learning. *arXiv*, 2021. URL http://arxiv.org/abs/2109.04617v2.

Foret, P., Kleiner, A., Mobahi, H., and Neyshabur, B. Sharpness-Aware Minimization for Efficiently Improving Generalization. In *International Conference on*

Learning Representations (ICLR), 2021. URL http://arxiv.org/abs/2010.01412v3.

Frankle, J., Dziugaite, G. K., Roy, D., and Carbin, M. Linear mode connectivity and the lottery ticket hypothesis. In *International Conference on Machine Learning (ICML)*, 2020. URL http://arxiv.org/abs/1912.05671v4.

Garipov, T., Izmailov, P., Podoprikhin, D., Vetrov, D. P., and Wilson, A. G. Loss Surfaces, Mode Connectivity, and Fast Ensembling of DNNs. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2018. URL http://arxiv.org/abs/1802.10026v4.

Goodfellow, I. J., Erhan, D., Carrier, P. L., Courville, A., Mirza, M., Hamner, B., Cukierski, W., Tang, Y., Thaler, D., Lee, D.-H., 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.

Helber, P., Bischke, B., Dengel, A., and Borth, D. EuroSAT: A Novel Dataset and Deep Learning Benchmark for Land Use and Land Cover Classification. *IEEE J. Sel. Top. Appl. Earth Obs. Remote. Sens.*, 12(7):2217–2226, 2019. URL https://doi.org/10.1109/JSTARS.2019.2918242.

Houlsby, N., Giurgiu, A., Jastrzebski, S., Morrone, B., De Laroussilhe, Q., Gesmundo, A., Attariyan, M., and Gelly, S. Parameter-efficient transfer learning for NLP. In *International Conference on Machine Learning (ICML)*, 2019. URL http://arxiv.org/abs/1902.00751v2.

Hu, E. J., Shen, Y., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., Wang, L., and Chen, W. LoRA: Low-Rank Adaptation of Large Language Models. In *International Conference on Learning Representations (ICLR)*, 2022. https://openreview.net/forum?id=nZeVKeeFYf9.

Huang, L., Bras, R. L., Bhagavatula, C., and Choi, Y. Cosmos QA: Machine reading comprehension with contextual commonsense reasoning. *arXiv*, 2019. URL http://arxiv.org/abs/1909.00277v2.

Ilharco, G., Wortsman, M., Wightman, R., Gordon, C., Carlini, N., Taori, R., Dave, A., Shankar, V., Namkoong, H., Miller, J., Hajishirzi, H., Farhadi, A., and Schmidt, L. OpenCLIP, 2021. https://doi.org/10.5281/zenodo.5143773.

Ilharco, G., Wortsman, M., Gadre, S. Y., Song, S., Hajishirzi, H., Kornblith, S., Farhadi, A., and Schmidt, L. Patching open-vocabulary models by interpolating weights. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022. URL http://arxiv.org/abs/2208.05592v2.

Ilharco, G., Ribeiro, M. T., Wortsman, M., Gururangan, S., Schmidt, L., Hajishirzi, H., and Farhadi, A. Editing models with task arithmetic. In *International Conference on Learning Representations (ICLR)*, 2023. https://arxiv.org/abs/2110.08207.

Jiang, W., Lin, B., Shi, H., Zhang, Y., Li, Z., and Kwok, J. T. BYOM: Building Your Own Multi-Task Model For Free, 2024. URL http://arxiv.org/abs/2310.01886v3.

Jin, X., Ren, X., Preotiuc-Pietro, D., and Cheng, P. Dataless Knowledge Fusion by Merging Weights of Language Models. In *International Conference on Learning Representations (ICLR)*, 2023. URL https://openreview.net/forum?id=FCnohuR6AnM.

Khot, T., Clark, P., Guerquin, M., Jansen, P., and Sabharwal, A. QASC: A Dataset for Question Answering via Sentence Composition, 2020. URL http://arxiv.org/abs/1910.11473v2.

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

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

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

Lester, B., Al-Rfou, R., and Constant, N. The power of scale for parameter-efficient prompt tuning. *arXiv*, 2021. URL http://arxiv.org/abs/2104.08691v2.

Levesque, H., Davis, E., and Morgenstern, L. The winograd schema challenge. In *Thirteenth international conference on the principles of knowledge representation and reasoning*, 2012.

Liang, T., Glossner, J., Wang, L., Shi, S., and Zhang, X. Pruning and quantization for deep neural network acceleration: A survey. *Neurocomputing*, 461:370–403, 2021. URL http://arxiv.org/abs/2101.09671v3.

Lin, K., Tafjord, O., Clark, P., and Gardner, M. Reasoning over paragraph effects in situations. *arXiv*, 2019. URL http://arxiv.org/abs/1908.05852v2.

Liu, H., Tam, D., Muqeeth, M., Mohta, J., Huang, T., Bansal, M., and Raffel, C. A. Few-shot parameter-efficient fine-tuning is better and cheaper than in-context learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022. URL http://arxiv.org/abs/2205.05638v2.

Luo, S., Tan, Y., Patil, S., Gu, D., von Platen, P., Passos, A., Huang, L., Li, J., and Zhao, H. Lcm-lora: A universal stable-diffusion acceleration module. *arXiv*, 2023.

Matena, M. S. and Raffel, C. A. Merging models with fisher-weighted averaging. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022. URL http://arxiv.org/abs/2111.09832v2.

Netzer, Y., Wang, T., Coates, A., Bissacco, A., Wu, B., and Ng, A. Y. Reading Digits in Natural Images with Unsupervised Feature Learning. In *NIPS Workshop on Deep Learning and Unsupervised Feature Learning 2011*, 2011. URL http://ufldl.stanford.edu/housenumbers/nips2011_housenumbers.pdf.

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

Ortiz-Jimenez, G., Favero, A., and Frossard, P. Task Arithmetic in the Tangent Space: Improved Editing of Pre-Trained Models. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023. URL http://arxiv.org/abs/2305.12827v3.

Panigrahi, A., Saunshi, N., Zhao, H., and Arora, S. Task-specific skill localization in fine-tuned language models. In *International Conference on Machine Learning (ICML)*, 2023. URL http://arxiv.org/abs/2302.06600v2.

Parkhi, O. M., Vedaldi, A., Zisserman, A., and Jawahar, C. Cats and dogs. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012.

Pruksachatkun, Y., Phang, J., Liu, H., Htut, P. M., Zhang, X., Pang, R. Y., Vania, C., Kann, K., and Bowman, S. R. Intermediate-Task Transfer Learning with Pretrained Language Models: When and Why Does It Work? In *Association for Computational Linguistics (ACL)*, 2020. URL https://aclanthology.org/2020.acl-main.467.

Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., and Sutskever, I. Language Models are Unsupervised Multitask Learners, 2019. https://openai.com/blog/better-language-models/.

Radford, A., Kim, J. W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., Krueger, G., and Sutskever, I. Learning Transferable Visual Models From Natural Language Supervision. In *International Conference on Machine Learning (ICML)*, 2021. URL http://arxiv.org/abs/2103.00020v1.

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

Reed, S., Zolna, K., Parisotto, E., Colmenarejo, S. G., Novikov, A., Barth-maron, G., Giménez, M., Sulsky, Y., Kay, J., Springenberg, J. T., Eccles, T., Bruce, J., Razavi, A., Edwards, A., Heess, N., Chen, Y., Hadsell, R., Vinyals, O., Bordbar, M., and de Freitas, N. A Generalist Agent. *Transactions on Machine Learning Research*, 2022. URL https://openreview.net/forum?id=1ikK0kHjvj.

Riquelme, C., Puigcerver, J., Mustafa, B., Neumann, M., Jenatton, R., Pinto, A. S., Keysers, D., and Houlsby, N. Scaling Vision with Sparse Mixture of Experts. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2021.

Rogers, A., Kovaleva, O., Downey, M., and Rumshisky, A. Getting closer to AI complete question answering: A set of prerequisite real tasks. In *AAAI Conference on Artificial Intelligence (AAAI)*, 2020.

Sakaguchi, K., Bras, R. L., Bhagavatula, C., and Choi, Y. WinoGrande: An Adversarial Winograd Schema Challenge at Scale. *Commun. ACM*, 64(9):99–106, 2021. URL http://arxiv.org/abs/1907.10641v2.

Sap, M., Rashkin, H., Chen, D., LeBras, R., and Choi, Y. Socialiqa: Commonsense reasoning about social interactions. *arXiv*, 2019. URL http://arxiv.org/abs/1904.09728v3.

Sharma, R., Allen, J., Bakhshandeh, O., and Mostafazadeh, N. Tackling the Story Ending Biases in The Story Cloze Test. In *Association for Computational Linguistics (ACL)*, 2018. URL https://aclanthology.org/P18-2119.

Shazeer, N., Mirhoseini, A., Maziarz, K., Davis, A., Le, Q., Hinton, G., and Dean, J. Outrageously Large Neural Networks: The Sparsely-Gated Mixture-of-Experts Layer. In *International Conference on Learning Representations (ICLR)*, 2017.

Socher, R., Perelygin, A., Wu, J., Chuang, J., Manning, C. D., Ng, A., and Potts, C. Recursive deep models

for semantic compositionality over a sentiment treebank. In *Empirical Methods in Natural Language Processing (EMNLP)*, 2013. https://aclanthology.org/D13-1170/.

Stallkamp, J., Schlipsing, M., Salmen, J., and Igel, C. The German traffic sign recognition benchmark: a multi-class classification competition. In *International Joint Conference on Neural Networks (IJCNN)*, 2011. https://ieeexplore.ieee.org/document/6033395.

Standley, T., Zamir, A. R., Chen, D., Guibas, L. J., Malik, J., and Savarese, S. Which Tasks Should Be Learned Together in Multi-task Learning? In *International Conference on Machine Learning (ICML)*, 2020. URL http://arxiv.org/abs/1905.07553v4.

Tafjord, O., Gardner, M., Lin, K., and Clark, P. QuaRTz: An Open-Domain Dataset of Qualitative Relationship Questions. In *Empirical Methods in Natural Language Processing (EMNLP)*, 2019. URL https://aclanthology.org/D19-1608.

Tam, D., Bansal, M., and Raffel, C. Merging by Matching Models in Task Parameter Subspaces. *Transactions on Machine Learning Research*, 2024. URL https://openreview.net/forum?id=qNGo6ghWFB.

Veeling, B. S., Linmans, J., Winkens, J., Cohen, T., and Welling, M. 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.

Wortsman, M., Horton, M. C., Guestrin, C., Farhadi, A., and Rastegari, M. Learning Neural Network Subspaces. In *International Conference on Machine Learning (ICML)*, 2021. URL http://arxiv.org/abs/2102.10472v3.

Wortsman, M., Ilharco, G., Gadre, S. Y., Roelofs, R., Gontijo-Lopes, R., Morcos, A. S., Namkoong, H., Farhadi, A., Carmon, Y., Kornblith, S., et al. Model soups: averaging weights of multiple fine-tuned models improves accuracy without increasing inference time. In *International Conference on Machine Learning (ICML)*, 2022a. URL http://arxiv.org/abs/2203.05482v3.

Wortsman, M., Ilharco, G., Kim, J. W., Li, M., Kornblith, S., Roelofs, R., Lopes, R. G., Hajishirzi, H., Farhadi, A., Namkoong, H., and Schmidt, L. Robust Fine-Tuning of Zero-Shot Models. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022b. URL http://arxiv.org/abs/2109.01903v3.

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

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

Yadav, P., Choshen, L., Raffel, C., and Bansal, M. ComPEFT: Compression for Communicating Parameter Efficient Updates via Sparsification and Quantization. *arXiv*, 2023a. URL http://arxiv.org/abs/2311.13171v1.

Yadav, P., Tam, D., Choshen, L., Raffel, C., and Bansal, M. TIES-Merging: Resolving Interference When Merging Models. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023b. URL http://arxiv.org/abs/2306.01708v2.

Yang, E., Wang, Z., Shen, L., Liu, S., Guo, G., Wang, X., and Tao, D. AdaMerging: Adaptive Model Merging for Multi-Task Learning. In *International Conference on Learning Representations (ICLR)*, 2024. URL https://openreview.net/forum?id=nZP6NgD3QY.

Yang, Y., Yih, W.-t., and Meek, C. WikiQA: A Challenge Dataset for Open-Domain Question Answering. In *Empirical Methods in Natural Language Processing (EMNLP)*, 2015. URL https://aclanthology.org/D15-1237.

Zhang, Y., Baldridge, J., and He, L. PAWS: Paraphrase Adversaries from Word Scrambling. In *North American Chapter of the Association for Computational Linguistics (NAACL)*, 2019. URL https://aclanthology.org/N19-1131.

Zhou, C., Li, Q., Li, C., Yu, J., Liu, Y., Wang, G., Zhang, K., Ji, C., Yan, Q., He, L., et al. A comprehensive survey on pretrained foundation models: A history from bert to chatgpt. *arXiv*, 2023. URL http://arxiv.org/abs/2302.09419v3.

Zhou, J., Lin, Z., Zheng, Y., Li, J., and Yang, Z. Not All Tasks Are Born Equal: Understanding Zero-Shot Generalization. In *International Conference on Learning Representations (ICLR)*, 2022.

A. Experimental Details

All our experiments were performed using the same hardware consisting of four V100 NVIDIA GPUs with 32GB of memory each.

A.1. Fine-tuning

Fine-tuning: For all fine-tuning experiments, we stick to the training procedure outlined in Ilharco et al. (2023). Specifically, we fine-tune the same pre-trained CLIP checkpoint obtained from the *openclip* repository (Ilharco et al., 2021). We fine-tune for 2,000 iterations, using a batch size of 128, a learning rate of $1e^{-5}$, and a cosine annealing learning rate schedule with 200 warm-up steps, along with the AdamW optimizer. Following Ilharco et al. (2023) and Ortiz-Jimenez et al. (2023), we freeze the weights of the classification layer during fine-tuning process.

Normalized Accuracy To account for the task difficulties, we provide the normalized accuracies as well as the absolute accuracies in our results. Specifically, the normalization is performed with respect to the accuracy achieved by the individual fine-tuned models:

$$\text{Normalized Accuracy} = \frac{1}{T}\sum_{t=1}^{T}\frac{\underset{x\sim\mu_t}{\text{acc}}\left[f_{\text{merged}}(x)\right]}{\underset{x\sim\mu_t}{\text{acc}}\left[f_{\text{fine-tuned}}(x)\right]} \tag{10}$$

Note that the normalized accuracy depends on the fine-tuning methods as well as the merging methods.

A.2. Hyper-parameter tuning

Mask sparsity factor λ For constructing task-specific masks, we tune the hyper-parameter λ for each task over $\{0.2, 0.3, 0.4, 0.5, 0.6\}$. The best λ for each task is selected based the validation performance on each individual tasks.

Task vector scaling factor α Following Ilharco et al. (2023), we use a single scaling factor α to scale the multi-task vector for the model merging methods in Table 2 and Table 3. The scaling factor is tuned over a range of $\{0.0, 0.1, ..., 0.9, 1.0\}$, selected based on the performance on the validation set averaged on all tasks.

A.3. Benchmark task contents

A.3.1. Computer vision

The **8-task scenario** takes into account the 8 tasks studied in Radford et al. (2021), Ilharco et al. (2023), including 1. Cars (Krause et al., 2013), 2. DTD (Cimpoi et al., 2014), 3. EuroSAT (Helber et al., 2019), 4. GTSRB (Stallkamp et al., 2011), 5. MNIST (LeCun, 1998), 6. RESISC45 (Cheng et al., 2017), 7. SUN397 (Xiao et al., 2016), 8. SVHN (Netzer et al., 2011).

The **14-task scenario** adds to the 8 tasks mentioned above the following tasks: 9. CIFAR100 (Krizhevsky et al., 2009), 10. STL10 (Coates et al., 2011), 11. Flowers102 (Nilsback & Zisserman, 2008), 12. OxfordIIITPet (Parkhi et al., 2012), 13. PCAM (Veeling et al., 2018), 14. FER2013 (Goodfellow et al., 2013).

The **20-task scenario** adds to the 14 tasks mentioned above the following tasks: 15. EMNIST (Cohen et al., 2017), 16. CIFAR10 (Krizhevsky et al., 2009), 17. Food101 (Bossard et al., 2014), 18. FashionMNIST (Xiao et al., 2017), 19. RenderedSST2 (Socher et al., 2013; Radford et al., 2019) 20. KMNIST (Clanuwat et al., 2018),

For the benchmarks beyond 8 tasks, we use available datasets from `torchvision` library. For the 14-task scenario, we aim for diversisty in the tasks as much as possible. After removing the MNIST-variants in the 20-task benchmark, we rank the tasks based on the performance comparison for zero-shot CLIP v.s. linear probe ResNet-50 (Radford et al., 2019) with ascending order and select the top 6 tasks.

A.3.2. Natural language Processing

7 NLP Tasks This benchmark is studied in Yadav et al. (2023b) and contains the following datasets 1. QASC (Khot et al., 2020), 2. QuaRTz (Tafjord et al., 2019), 3. PAWS (Zhang et al., 2019), 4. Story Cloze (Sharma et al., 2018), 5. WikiQA (Yang et al., 2015), 6. Winogrande (Sakaguchi et al., 2021) and 7. WSC (Levesque et al., 2012)

8 QA Tasks Following Tam et al. (2024), we evaluate on another benchmark, containing the following tasks 1. CosmosQA (Huang et al., 2019), 2. QASC (Khot et al., 2020), 3. QuAIL (Rogers et al., 2020), 4. QuaRTz (Tafjord et al., 2019)), 5. PAWS (Zhang et al., 2019). 6. ROPES (Lin et al., 2019), 7. SocialIQA (Sap et al., 2019), 8. Wiki QA (Yang et al., 2015).

B. Derivation of Equation 4

This short section shows the derivation of the mask criterion in more detail:

$$\boldsymbol{m}_t^* = \underset{\boldsymbol{m}_t \in \{0,1\}^P}{\operatorname{argmin}} (\|\hat{\boldsymbol{\theta}}_t - \boldsymbol{\theta}_t\|_1) \tag{11}$$

$$= \underset{\boldsymbol{m}_t \in \{0,1\}^P}{\operatorname{argmin}} (\|\boldsymbol{m}_t \circ \boldsymbol{\tau}_{\text{MTL}} - \boldsymbol{\tau}_t\|_1) \tag{12}$$

$$= \underset{\boldsymbol{m}_t \in \{0,1\}^P}{\operatorname{argmin}} \sum_{n=1}^{P} |m_t^{(n)} \cdot \tau_{\text{MTL}}^{(n)} - \tau_t^{(n)}| \tag{13}$$

$$\implies m_t^{(n)*} = \underset{m_t^{(n)} \in \{0,1\}}{\operatorname{argmin}} |m_t^{(n)} \cdot \tau_{\text{MTL}}^{(n)} - \tau_t^{(n)}| \tag{14}$$

$$= \begin{cases} 1 & \text{if } |\tau_t^{(n)}| \geq |\tau_{\text{MTL}}^{(n)} - \tau_t^{(n)}| \\ 0 & \text{otherwise} \end{cases} \tag{15}$$

$$= \mathbb{1}\{|\tau_t^{(n)}| \geq |\tau_{\text{MTL}}^{(n)} - \tau_t^{(n)}|\} \tag{16}$$

From Equation 13 to Equation 14 we use the independence of each sub-problem. Aggregating over all sub-problems yields that the optimal mask is given by:

$$\mathbf{m}_t^* = \mathbb{1}\{|\boldsymbol{\tau}_t| \geq |\boldsymbol{\tau}_{\text{MTL}} - \boldsymbol{\tau}_t|\} \tag{17}$$

C. Storage cost calculation

This section show the calculation of the storage cost for each method in Table 2 and Table 3. Let T be the number of tasks, P be the number of all parameters, P' be the number of trainable parameters in the model, and F be the number of frozen parameters in the model. Assuming one float parameter takes 32 bits, for each method, their respective storage cost for T tasks is calculated as:

- Fine-tuned models: $32(TP'+F)$. $32TP'$ is for storing T trainable parameters and $32F$ is for storing frozen parameters.
- Task arithmetic: $32P$; Stores a single model.
- Ties-merging: $32P$; Stores a single model.
- Consensus Task arithmetic: $32P$; Stores a single model.
- Consensus Ties: $32P$; Stores a single model.
- Zero-shot: $32P$; Stores a single model.
- Magnitude Masking: $(64+T)P' + 32F$; $64P' + 32F$ is for storing zero-shot model and multi-task vector, while TP' is for storing T binary masks.
- Magnitude Pruning: $> (64+T)P' + 32F$; For Magnitude Pruning, we need to store a single model, as well as the sparsified task vectors for each task. We calculate the respective sparsity such that the total storage cost will be higher than $(64+T)P' + 32F$.
- TALL Mask + Task arithmetic: $(64+T)P' + 32F$; $64P' + 32F$ is for storing zeroshot model and multi-task vector, while TP' is for storing T binary masks.
- TALL Mask + Ties-merging: $(64+T)P' + 32F$; $64P' + 32F$is for storing zeroshot model and multi-task vector, while TP' is for storing T binary masks.

D. Additional results

D.1. Performance for removing only catastrophic weights

For Consensus Merging, we propose to remove both catastrophic weights and selfish weights, as their contribution to the multi-task performance is limited. In this section, we study the performance of Consensus Merging when removing only catastrophic weights, i.e., weights beneficial for none of the tasks. The results for the experiments in image classification are presented in Table 4 with ViT-B/32 and ViT-L/14, and Table 5 with ViT-B/16.

Table 4. Performance of Consensus when removing only catastrophic weights with ViT-B/32 and ViT-L/14 on image classification benchmarks.

	ViT-B/32			ViT-L/14		
	8 tasks	14 tasks	20 tasks	8 tasks	14 tasks	20 tasks
Task arithmetic	70.8 (76.5)	65.4 (72.2)	60.6 (66.8)	84.8 (88.5)	79.3 (83.8)	74.0 (78.0)
Consensus TA ($k = 1$)	73.8 (79.4)	69.0 (75.8)	64.5 (71.0)	85.5 (89.2)	81.4 (86.1)	77.7 (81.9)
Consensus TA ($k = 2$)	75.0 (80.7)	**70.4** (**77.4**)	**65.4** (**72.0**)	86.2 (89.9)	**82.3** (**86.9**)	**78.9** (**83.2**)
TIES	75.2 (81.1)	68.0 (74.9)	63.3 (69.8)	86.9 (90.7)	79.5 (84.1)	75.7 (79.8)
Consensus TIES ($k = 1$)	**75.9** (**81.0**)	69.0 (75.9)	64.5 (71.0)	**87.5** (**91.3**)	81.6 (86.3)	78.0 (82.1)
Consensus TIES ($k = 2$)	74.8 (80.6)	67.7 (74.5)	63.2 (69.6)	86.9 (90.7)	81.5 (86.1)	76.8 (80.9)

Table 5. Performance of Consensus when removing only catastrophic weights with ViT-B/16 on image classification benchmarks.

	ViT-B/16		
	8 tasks	14 tasks	20 tasks
Task arithmetic	76.2 (80.6)	70.5 (75.8)	65.7 (70.9)
Consensus TA ($k = 1$)	78.0 (82.4)	72.9 (78.4)	68.9 (74.0)
Consensus TA ($k = 2$)	79.2 (83.6)	74.3 (79.8)	**70.0** (**75.2**)
TIES	**79.7** (**84.3**)	73.2 (78.7)	68.1 (73.3)
Consensus TIES ($k = 1$)	79.4 (83.8)	**74.4** (**80.0**)	69.9 (75.1)
Consensus TIES ($k = 2$)	79.4 (83.9)	74.1 (79.5)	68.7 (73.9)

D.2. Effect of weight-pruning threshold for 8 and 14 tasks

We plot the performance for Consensus Merging with varying weight-pruning threshold k, on the 8-task and 14-task benchmarks in image classification with ViT-B/32, in Figure 7. Similar to the observation from Figure6, the performance differs from one method to another (TIES and task arithmetic in our case). While the optimal performance of Consensus TA is achieved by removing both catastrophic and selfish weights (setting the threshold to 2), for Consensus TIES it delivers the best results by removing only the catastrophic weights (setting the threshold to 1). Gradually the performance of Consensus Merging reduces to zero-shot as more and more weights get removed.

D.3. Performance with Parameter-efficient fine-tuning methods

In the main text we have presented the performance of our proposed methods for models with full fine-tuning. As Parameter-efficient fine-tuning (PEFT) methods have been widely adopted as cost-effective alternatives to full fine-tuning, in this section we provide the performance of our methods on PEFT models as well.

Following previous work (Yadav et al., 2023b), we provide the performance for models fine-tuned with $(IA)^3$ (Liu et al., 2022) on the three NLP benchmarks.

The results are presented in Table 6. We observe from the results that while removing the selfish weights lead to performance degradation, possibly due to different profiles of the weights being tuned by $(IA)^3$ and full fine-tuning, removing only the catastrophic weights leads to consistent performance improvement of Consensus Merging over Task Arithmetic and TIES.

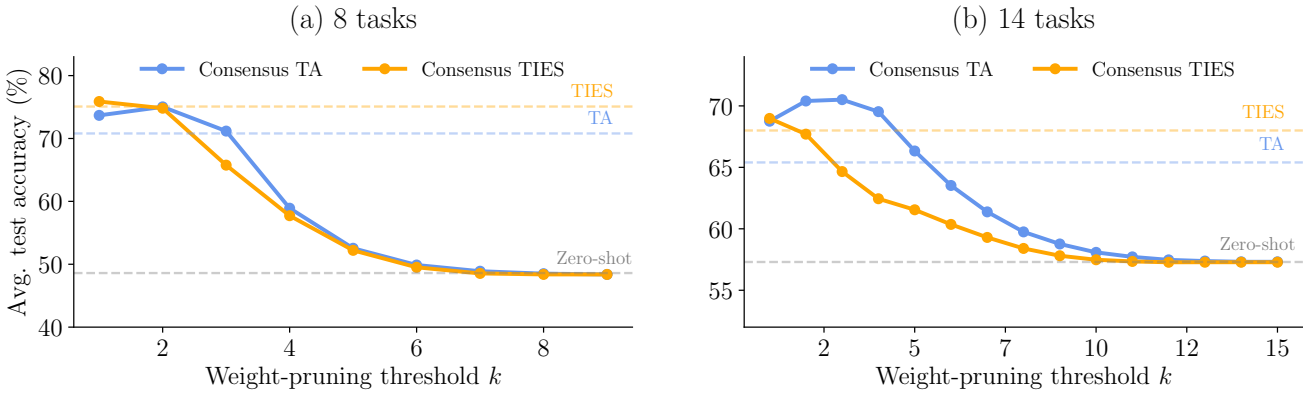


Figure 7. Performance of Consensus Merging with varying weight-pruning threshold k.

For example, for the 7 NLP tasks, Consensus TA leads to 3.8% gain to Task Arithmetic and Consensus TIES leads to 5.3% gain to TIES.

Table 6. Results on NLP benchmarks using $(IA)^3$ models, comparing the performance of Consensus Merging with baseline methods.

	7 NLP tasks	8 QA tasks	All 11 tasks
	$(IA)^3$	$(IA)^3$	$(IA)^3$
Zeroshot	44.9	33.1	36.9
Task arithmetic	67.1 (79.5)	57.6 (74.3)	59.7 (77.7)
Consensus TA ($k = 1$)	70.9 (83.6)	**60.2** (**77.1**)	**60.8** (**79.1**)
Consensus TA ($k = 2$)	66.6 (78.6)	54.1 (68.1)	55.7 (71.1)
TIES	66.0 (77.6)	56.8 (72.6)	56.1 (72.5)
Consensus TIES ($k = 1$)	**71.3** (**84.2**)	58.7 (74.9)	59.2 (76.5)
Consensus TIES ($k = 2$)	67.6 (79.8)	56.3 (71.6)	53.1 (68.8)

D.4. Distribution of mask agreements with more tasks

We provide in Figure 8 the same histogram as in Figure 3 for 14 tasks and 20 tasks, respectively for Task Arithmetic and TIES.

D.5. Result for ViT-B/16

We provide here the same results shown in Table 3 for ViT-B/16 in Table 7, where we observe similar findings as the case for ViT-B/32 and ViT-L/14 in main text.

D.6. Full results on individual tasks

D.6.1. Full results for vision

In main text we have provided the full results on individual tasks for ViT-B/32, here we provide the same radar plots for ViT-B/16 in Figure 9, and the radar plots for ViT-L/14 in Figure 10, respectively. We observe similar findings for the case with ViT-B/32 in main text.

D.6.2. Full results for NLP

We provide the full results for individual tasks in NLP in Figure 11

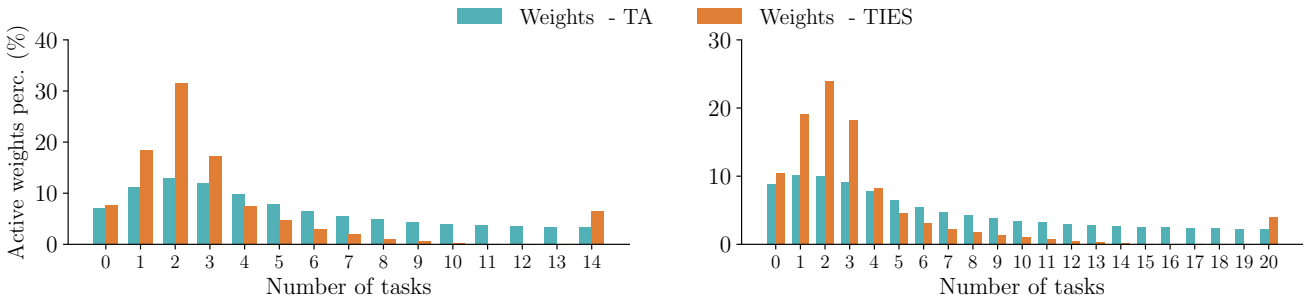


Figure 8. The mask agreement profile, defined in Equation 7, in the case of 14 and 20 vision tasks. This figure complements Figure 3.

Table 7. Complementary to Table 3, for results obtained with ViT-B/16.

	Method	ViT-B/16					
		8 tasks		14 tasks		20 tasks	
		Acc.(%)↑	Bits(Gb)↓	Acc.(%)↑	Bits(Gb)↓	Acc.(%)↑	Bits(Gb)↓
Merging	Zeroshot	55.2	3.6	61.2	3.6	59.7	3.6
	Weight averaging	72.2 (76.6)	3.6	69.4 (74.7)	3.6	65.3 (70.3)	3.6
	Task arithmetic	75.8 (80.2)	3.6	70.5 (75.8)	3.6	65.7 (70.7)	3.6
	TIES	**79.7** (**84.3**)	3.6	73.2 (78.7)	3.6	68.2 (73.3)	3.6
	Consensus Task arithmetic [ours]	79.2 (83.6)	3.6	**74.3** (**79.8**)	3.6	**69.7** (**74.9**)	3.6
	Consensus TIES [ours]	79.4 (83.9)	3.6	74.1 (79.5)	3.6	68.7 (73.9)	3.6
Compression	Fine-tuned	94.6	22.9	92.8	39.4	93.2	56.0
	Mag. Prunning	93.3 (98.5)	>7.0	88.1 (94.7)	>7.5	86.5 (92.7)	>8.1
	Mag. Masking	89.7 (94.6)	7.0	84.8 (91.1)	7.5	81.6 (87.3)	8.1
	TALL Mask + Task arithmetic	94.2 (99.6)	7.0	92.0 (99.2)	7.5	92.5 (99.3)	8.1
	TALL Mask + TIES	**94.6** (**99.9**)	7.0	**92.6** (**99.8**)	7.5	**93.0** (**99.8**)	8.1

E. Sample subset selection protocol

Figure 5 presents the performance comparison as a function of different numbers of tasks. For a given number of tasks, we select 8 representative task combinations. Specifically, we sort all 20 tasks based on the following 8 orders:

- Ascending order on zero-shot performance:

  KMNIST, EMNIST, SVHN, GTSRB, FER2013, DTD, EuroSAT, MNIST, RenderedSST2, Cars, PCAM, RESISC45, FashionMNIST, SUN397, CIFAR100, Flowers102, Food101, OxfordIIITPet, CIFAR10, STL10

- Descending order on zero-shot performance:

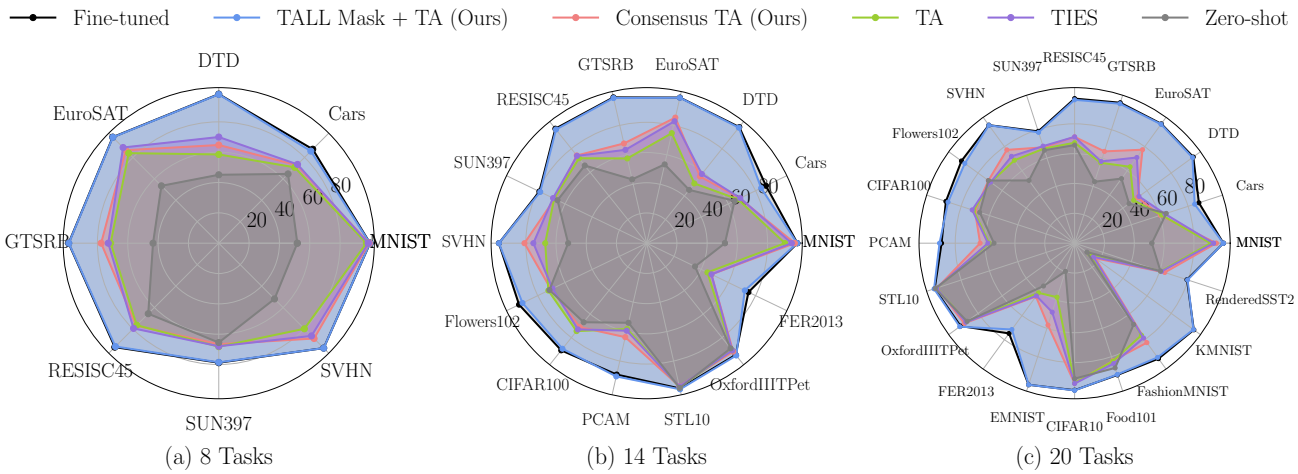
  STL10, CIFAR10, OxfordIIITPet, Food101, Flowers102, CIFAR100, SUN397, FashionMNIST, RESISC45, PCAM, Cars, RenderedSST2, MNIST, EuroSAT, DTD, FER2013, GTSRB, SVHN, EMNIST, KMNIST

- Wave order on zero-shot performance:

  Cars, PCAM, RenderedSST2, RESISC45, MNIST, FashionMNIST, EuroSAT, SUN397, DTD, CIFAR100, FER2013, Flowers102, GTSRB, Food101, SVHN, OxfordIIITPet, EMNIST, CIFAR10, KMNIST, STL10

- Zigzag order on zero-shot performance:

  STL10, KMNIST, CIFAR10, EMNIST, OxfordIIITPet, SVHN, Food101, GTSRB, Flowers102, FER2013, CIFAR100, DTD, SUN397, EuroSAT, FashionMNIST, MNIST, RESISC45, RenderedSST2, PCAM, Cars



(a) 8 Tasks (b) 14 Tasks (c) 20 Tasks

Figure 9. Absolute accuracy (%) of individual tasks for ViT-B/16, comparing the accuracy of individual fine-tuned models, Our method, Task arithmetic, Ties-merging, and Zero-shot.

- Ascending alphabetic order:

  CIFAR10, CIFAR100, Cars, DTD, EMNIST, EuroSAT, FER2013, FashionMNIST, Flowers102, Food101, GTSRB, KMNIST, MNIST, OxfordIIITPet, PCAM, RESISC45, RenderedSST2, STL10, SUN397, SVHN

- Descending alphabetic order:

  SVHN, SUN397, STL10, RenderedSST2, RESISC45, PCAM, OxfordIIITPet, MNIST, KMNIST, GTSRB, Food101, Flowers102, FashionMNIST, FER2013, EuroSAT, EMNIST, DTD, Cars, CIFAR100, CIFAR10

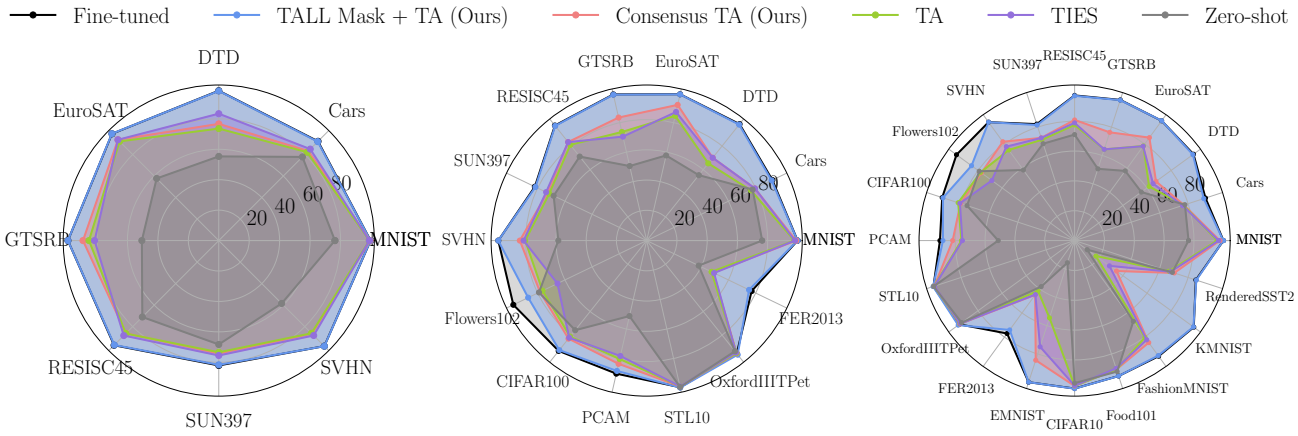
- Wave order on alphabetic order:

  CIFAR10, SVHN, CIFAR100, SUN397, Cars, STL10, DTD, RenderedSST2, EMNIST, RESISC45, EuroSAT, PCAM, FER2013, OxfordIIITPet, FashionMNIST, MNIST, Flowers102, KMNIST, Food101, GTSRB

- Zigzag order on alphabetic order:

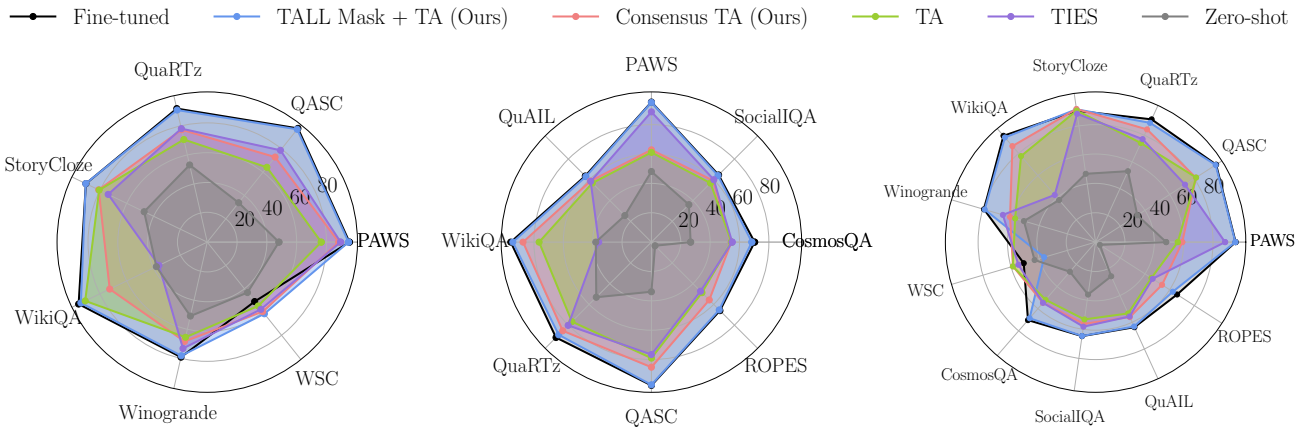
  GTSRB, Food101, KMNIST, Flowers102, MNIST, FashionMNIST, OxfordIIITPet, FER2013, PCAM, EuroSAT, RESISC45, EMNIST, RenderedSST2, DTD, STL10, Cars, SUN397, CIFAR100, SVHN, CIFAR10

For a given number of tasks n, we retrieve the first n tasks from these 8 sequences respectively, such that we account for both task difficulty (in zero-shot performance order) and randomness (in alphabetic order) when generating Figure 5.



(a) 8 Tasks (b) 14 Tasks (c) 20 Tasks

Figure 10. Absolute accuracy (%) of individual tasks for ViT-L-14, comparing the accuracy of individual fine-tuned models, Our method, Task arithmetic, Ties-merging, and Zero-shot.



(a) 7 NLP Tasks (b) 8 QA Tasks (c) 11 Tasks

Figure 11. Absolute accuracy (%) of individual tasks for T5-large on three benchmarks, comparing the accuracy of individual fine-tuned models, Our method, Task arithmetic, Ties-merging, and Zero-shot.