MagMax: Leveraging Model Merging for Seamless Continual Learning

Daniel Marczak*[1,2], Bartłomiej Twardowski[1,5,6], Tomasz Trzciński[1,2,4], and Sebastian Cygert[1,3]

[1] IDEAS NCBR
[2] Warsaw University of Technology
[3] Gdańsk University of Technology
[4] Tooploox
[5] Autonomous University of Barcelona
[6] Computer Vision Center

Abstract. This paper introduces a continual learning approach named MagMax, which utilizes model merging to enable large pre-trained models to continuously learn from new data without forgetting previously acquired knowledge. Distinct from traditional continual learning methods that aim to reduce forgetting during task training, MagMax combines sequential fine-tuning with a maximum magnitude weight selection for effective knowledge integration across tasks. Our initial contribution is an extensive examination of model merging techniques, revealing that simple approaches like weight averaging and random weight selection surprisingly hold up well in various continual learning contexts. More importantly, we present MagMax, a novel model-merging strategy that enables continual learning of large pre-trained models for successive tasks. Our thorough evaluation demonstrates the superiority of MagMax in various scenarios, including class- and domain-incremental learning settings. The code is available on github.

Keywords: Continual Learning · Model Merging

1 Introduction

Large pre-trained models are considered cornerstones of complex machine learning systems, allowing unprecedented performance improvements across many challenging tasks [1, 2, 17, 32, 41, 50]. Yet their remarkable ability to generalize to unseen conditions is intrinsically limited by the stationary character of their training data. To keep up with the ever-changing world, these models should adapt continuously and assimilate knowledge from the stream of new data, which is the objective of Continual Learning (CL) [21, 25, 39].

Traditionally, CL approaches used regularization to retain the knowledge from previous tasks [18, 23], grow the network while learning new tasks [33, 48], or use a replay buffer to limit the catastrophic forgetting [12, 42, 52]. In this

* Corresponding author, email: daniel.marczak.dokt@pw.edu.pl

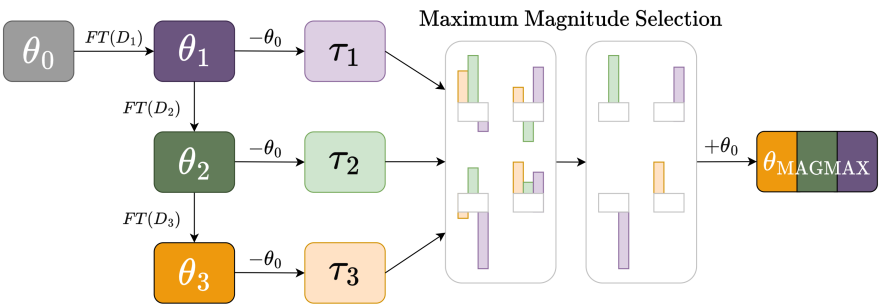

Figure 1: Overview of the proposed MagMax method for continual learning. We sequentially fine-tune the model on the subsequent tasks and create task vectors τ_i by subtracting the weights of the pre-trained model θ_0. Then we merge the task vectors using Maximum Magnitude Selection strategy which selects the parameters of task vectors by highest magnitude. Finally, we apply merged task vector to the pre-trained model to obtain a multitask model θ_{MagMax}. Note that with running statistics implementation we can only store two sets of weights (see Section 5 for details).

work, we argue that in the era of machine learning systems built on top of large pre-trained models, utilizing this foundation seems to present a more intuitive and effective strategy for continuous learning. Model merging is a new paradigm of adapting pre-trained models. It allows to consolidate the knowledge of multiple independently fine-tuned task-specific models into one multi-task model without any additional training. There are various methods that base on selecting or interpolating the weights of task-specific models [13, 26, 29, 36, 47]. Contrary to the traditional CL methods, which focus on alleviating forgetting *during* training on new tasks, model merging allows to seamlessly consolidate the knowledge *after* the training on new tasks leaving the training procedure unchanged.

When evaluated across a single, fixed set of diversified heterogeneous tasks [13, 29, 47], such as recognition of hand-written digits [22], satellite images [10] or car models [19], model merging methods perform well. However, this evaluation benchmark is far from a realistic use case. Furthermore, it does not include real-life applications with the data coming from similar (but disjoint) distributions, *e.g.* various kinds of medical imagery. Here, we fill this gap and extensively evaluate model merging techniques with different levels of task similarity (including class- and domain-incremental scenarios), varying number of tasks, and their granularity. We find that the simplest merging baselines - weight averaging and random weight selection - work surprisingly well, often outperforming sophisticated merging strategies and CL approaches.

Our evaluation highlights a significant drawback of the existing methods. They fine-tune pre-trained models independently for each task foregoing the potential of knowledge transfer. To address this significant limitation, we propose MagMax, a novel method for continual learning that utilizes sequential fine-tuning and model merging via maximum magnitude selection (see Figure 1). We show that sequential fine-tuning simplifies model merging by reducing the number

of sign conflicts – a major source of interference when merging models [47] – between task-specific models while maximum magnitude selection chooses the important parameter values. We investigate the effectiveness of the parameter selection strategy and examine the contribution of task vectors. Finally, we highlight a broader impact of our findings showing that merging via maximum magnitude selection can improve existing CL methods and that sequential fine-tuning improves the performance of models combined using various merging techniques.

To sum up, our contributions are as follows:

- We identify and fill the gaps in model merging evaluation by benchmarking the existing merging strategies in diverse settings with tasks containing different classes or domains when varying the number of tasks, task similarity, and their granularity.
- We find that simple baselines – weight averaging and random weight selection – are very strong and often outperform the existing merging strategies.
- We propose MagMax, a novel method for continual learning that sequentially fine-tunes the model and consolidates the knowledge by merging weights of task-specific models using maximum magnitude selection. MagMax achieves state-of-the-art results on multiple continual learning benchmarks.
- We highlight the broader implications of our results demonstrating that merging with maximum magnitude selection in model merging enhances existing continual learning methods and that sequential fine-tuning facilitates other existing merging techniques.

2 Related Work

Continual learning (CL) is a setting where models learn a sequence of tasks with access to the data from the current task only. The goal is to achieve high performance on all the tasks from the sequence, with catastrophic forgetting of knowledge learned from the previous tasks being the main challenge [7, 27]. One prominent example of CL approaches are the regularization-based methods. In EWC [18], the authors propose to use the Fisher information matrix to estimate model weight importance (for previous tasks) which is then used to penalize changes of important model weight. On the other hand, regularization can be applied on the data level, *e.g.* LwF [23] or DER [48] penalizes changes in model predictions or features. Other CL approaches include adding more parameters as the number of tasks increases [33, 49], or using memory buffer [12, 42, 48, 52] for data from old tasks, which is often undesirable due to the privacy concerns. In general, it seems that the best results are obtained by CL methods that favor stability, that is the model does not change much between consecutive learning tasks [16, 34]. As a result, a plethora of methods were developed for CL scenarios which assumed large first task [31, 53], or Large Pre-trained Model (LPM).

Continual Learning of LPMs became popular as capabilities (e.g., zero shot or out-of-distribution (OOD) performance) of foundation models became apparent [1, 2, 17, 32, 41, 50]. A recent study questioned the utility of some CL

methods, showing that by using a frozen model and nearest mean classifier can obtain competitive results [15]. Further advancements to the use of LPM were driven by using the prompting techniques [37, 43]. Alternatively, SLCA proposed a simple model that fine-tunes only the classification layer with a small learning rate [51]. In general, when using LPMs the focus in CL shifts towards maximal stability.

Weights interpolation has recently emerged as an efficient technique for transfer learning that reduces forgetting. After fine-tuning LPM on target data, its weights are interpolated with the weights of (unchanged) LPM, which allows finding a good balance between accuracy on the target domain and zero-shot capabilities of LPM [45]. Such an approach was further extended when merging models across multiple models for OOD performance [44] or in multi-task learning (i.e., Task Vectors [13]). Since then multiple methods have been developed in this area. TIES-Merging [47] reduces the interference when merging models by trimming parameters and electing signs. In [29], the authors linearize the fine-tuning to disentangle weights and facilitate merging. ZipLoRA [35] adapts diffusion models by merging LoRA weights for different styles and subjects. However, those methods were, up-to-date, evaluated on a limited number of scenarios. In this work, we are interested in using those promising approaches to test how they work for different similarities between tasks, as well as when they are compared with simple CL baselines. A concurrent work, CoFiMA [24], utilizes Fisher Merging [26] sequentially after each task to continually train closed vocabulary image classifiers. In contrast, we focus on reducing parameter-level interferences in open vocabulaty models.

3 Background and motivation

3.1 Problem setting

We consider a problem of continual learning of large pre-trained models. We assume access to a pre-trained model parametrized by d weights $\theta_0 \in \mathbb{R}^d$. Our goal is to adapt the model to a sequence of disjoint tasks $\{D_1, D_2, \ldots, D_n\}$ one task at a time. We investigate *exemplar-free* scenario which assumes no access to data from previous tasks.

We consider two fine-tuning scenarios:

- independent (Ind FT) - starts from pre-trained weights θ_0,
- sequential (Seq FT) - starts from the weights of the model fine-tuned on the sequence of previous tasks, *i.e.* when fine-tuning on task D_t, we start from θ_{t-1} which was trained on $\{D_1, D_2, \ldots, D_{t-1}\}$.

We use a notion of task vector [13] that is an element-wise difference between the fine-tuned model and the pre-trained model, *i.e.* $\tau_i = \theta_i - \theta_0$. Note that independently fine-tuned task vectors contain information about a single task and sequentially fine-tuned task vectors encompass some knowledge about all the tasks in the sequence.

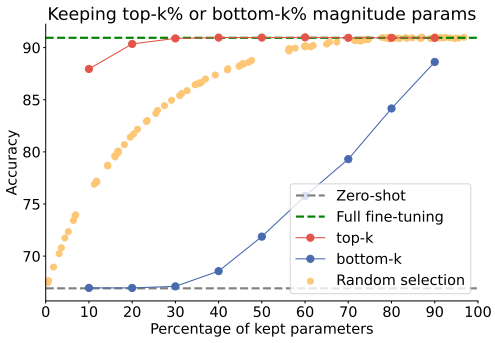

Figure 2: Only a small fraction of parameters that changed the most during fine-tuning is responsible for improved performance.

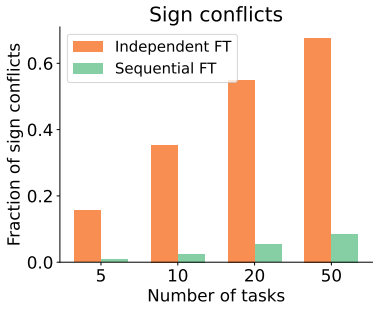

Figure 3: Sequential fine-tuning encourages consistent directions of parameter updates. We report sign conflicts after trimming 80% of the lowest magnitude parameters in each task vector.

3.2 Motivation

In this Section, we set and experimentally validate two hypotheses that serve as a motivation for developing a new method for continual learning via model merging.

$\mathcal{H}1$***: Parameters that change the most during fine-tuning are the most important for the task.*** To verify this hypothesis we conduct the following experiment. We fine-tune a model on CIFAR100 dataset and create a task vector τ. Then, we keep only $k\%$ of parameters that are selected at random, or according to their magnitude (lowest or highest) and remove the rest. Finally, we apply the pruned task vector to the pre-trained model and evaluate its performance[7]. Figure 2 presents the results of this experiment. We observe that only a small fraction of high-magnitude parameters in task vectors are relevant for the model performance. Keeping only 20% of the highest magnitude parameters yields results similar to fully fine-tuned models. To achieve similar performance we need to keep more than 90% of the lowest magnitude parameters or more than 60% of randomly selected parameters. These results validate $\mathcal{H}1$.

$\mathcal{H}2$***: Sequential fine-tuning reduces sign conflicts.*** When fine-tuning the model on several tasks, sometimes we can observe a disagreement between the directions of task-specific updates. Such a situation is denoted as *sign conflict*, as different task vectors have inconsistent signs for the same parameters. As noticed in [47] merging models with sign conflicts results in interference between tasks,

[7] Note, that this experiment considers pruning parameters of task vector instead of pruning the weights of the network. Therefore, the conclusions may differ from neural pruning literature that considers magnitude pruning a strong baseline [6, 8, 9].

and hence reduced performance of the final model. In this work, we postulate that sequential fine-tuning can reduce the number of sign conflicts. To verify this hypothesis, we fine-tune a model on CIFAR100 split into various number of tasks and count the conflicts of top-20% parameters in corresponding task vectors. We perform fine-tuning either independently or sequentially. Figure 3 presents the results. We observe that sequential fine-tuning significantly reduces the sign conflicts validating $\mathcal{H}2$.

4 Maximum Magnitude Selection

Based on the motivations introduced in the previous Section, we introduce Maximum Magnitude Selection (MagMax). It is a novel method for continual learning that utilizes sequential fine-tuning, following $\mathcal{H}2$, and model merging based on selecting the parameters of the highest magnitude, following $\mathcal{H}1$ (see Algorithm 1). Given a new task, D_t, our method consists of two steps:

1. **Sequential adaptation:** We obtain the new weights of the model θ_t by fine-tuning it on D_t. Importantly, we start from the weights of the model fine-tuned on previous tasks θ_{t-1}.
2. **Knowledge consolidation:** We consolidate task-specific knowledge using model merging. Firstly, we create task vectors for all tasks seen so far: $\{\tau_i\}_{i=1}^{t}$, where $\tau_i = \theta_i - \theta_0$. Then, for each parameter $p \in \{1, 2, \ldots, d\}$, we select the value τ^p_{MagMax} by the maximum magnitude out of all the task vectors. Lastly, we apply the resulting task vector τ_{MagMax} to the pre-trained model $\theta_{\text{MagMax}} = \theta_0 + \lambda * \tau_{\text{MagMax}}$, where λ is a scaling factor.

Algorithm 1 Continual learning with MagMax

Input: Pre-trained model θ_0 with d parameters, sequence of tasks $\{D_t\}_{t=1}^{N}$
for t ***in*** $1, ..., N$ **do**
  $\theta_t \leftarrow$ fine-tune(θ_{t-1}, D_t) `// Fine-tune from previous checkpoint on current data`
  for i ***in*** $1, ..., t$ **do**
    $\tau_i = \theta_i - \theta_0$ `// Create task vectors`
  end
  for p ***in*** $1, ..., d$ **do**
    $k \leftarrow \arg\max_i \{|\tau_i^p|\}_{i=1}^{t}$
    $\tau^p_{\text{MagMax}} \leftarrow \tau^p_k$ `// Maximum Magnitude Selection`
  end
  $\theta_{\text{MagMax}} \leftarrow \theta_0 + \lambda * \tau_{\text{MagMax}}$ `// Apply merged task vector to the pre-trained model`
  `// Use model` θ_{MagMax} `until new task`
end

5 Experimental setup

Datasets. For class-incremental learning (CIL) experiments we use CIFAR100 [20] and ImageNet-R [11] as generic image recognition benchmarks and CUB200 [40] and Cars [19] as fine-grained classification datasets. We split the datasets into N equal subsets of disjoint classes, where $N \in \{5, 10, 20, 50\}$ for generic benchmarks and $N \in \{5, 10, 20\}$ for fine-grained benchmarks (which contain less data).

To compare between class- and domain-incremental learning (DIL) we use ImageNet-R and DomainNet [30]. For domain-incremental learning experiments, we split DomainNet into 6 tasks by their domain (clipart, infographics, painting, quickdraw, real and sketch) and ImageNet-R into 15 tasks by their renditions (including cartoons, origami, paintings, sculptures, etc). Moreover, we split these datasets into the corresponding number of tasks following class-incremental protocol (described in the previous paragraph) for a fair comparison of CIL and DIL performance.

We also study the eight task setup proposed by [13] that includes the following datasets: Cars [19], DTD [4], SUN397 [46], EuroSAT [10], GTSRB [38], MNIST [22], SVHN [28] and RESISC45 [3]. This benchmark is widely popular in model merging community [13, 29, 47].

Baselines. We compare MagMax against well-established CL baselines **LwF** [23] and **EWC** [18] as well as recent model merging strategies, Model Soup (**Avg**) [44], Task Arithmetic (**TA**) [13] and TIES-Merging (**TIES**) [47]. Additionally, we introduce a simple baseline dubbed **RandMix** which randomly selects each parameter from one of the fine-tuned models, *i.e.* $\theta_m^p \sim \{\theta_i^p\}_{i=1}^N$. We also evaluate **MaxAbs** baseline, which is basically MagMax with independent fine-tuning instead of sequential. Finally, we present **zero-shot** performance which denotes the capabilities of the pre-trained model, and **joint** performance of a model fine-tuned on the whole dataset.

Implementation details. We use CLIP pre-trained model [32] with ViT/B-16 [5] image encoder. We follow the training procedure from [14], namely we fine-tune the image encoder with a batch size of 128, learning rate 1e-5, and a cosine annealing learning rate schedule and AdamW optimizer with weight decay 0.1. We train CIFAR100, ImageNet-R and DomainNet for 10 epochs each task, and CUB200 and Cars for 30 epochs. We use the final classification layer output by CLIP's text encoder and keep it frozen during fine-tuning, following [14]. This fine-tuning recipe preserves the open-vocabulary nature of the model and does not harm the accuracy compared to training the classification layer [14].

We consider an exemplar-free continual learning scenario in which we cannot store any data from the previous tasks. As a result, we can not tune scaling factor λ at merging time as described in [13]. Therefore, we follow no validation scenario from [47] and set constant λ for each method based on experiments on CIFAR100/5 setting. We choose $\lambda = 0.5$ for MagMax, $\lambda = 0.55$ for TIES and $\lambda = 1/N$ for Task Vectors. Notice, that choosing $\lambda = 1/N$ for Task Vectors simplifies the method to a simple average of task vectors. It makes Task Vectors

and Model Soup identical, and we call this method Avg in further experiments. We tune the hyperparameters of CL methods in the same scenario, setting $\lambda = 1e6$ for EWC and $\lambda = 0.3$ for LwF.

Memory footprint. In Figure 1 and Algorithm 1, we describe that MagMax stores all the previous checkpoints for the sake of simplicity. However, an efficient implementation of the method stores two sets of weights: sequentially fine-tuned θ_t and combined task vector τ_{MagMax_t} of running statistics (maximum magnitude). When task $t+1$ arrives, we start from θ_t and fine-tune the model resulting in θ_{t+1}. Then, we merge τ_{MagMax_t} with τ_{t+1} which is identical to merging $\{\tau_i\}_{i=0}^{t+1}$. That requires a constant memory footprint.

6 Main results

Class-incremental learning. Table 1 presents the comparison of MagMax with CL methods and merging-based baselines on various class-incremental learning benchmarks. MagMax consistently outperforms the competitors across the scenarios that vary in number of tasks and dataset granularity, achieving on average 2.1% better results than the second best method. Interestingly, simple baselines that merge independent fine-tunings by averaging (Avg) or even randomly mixing (RandMix) the weights, are close competitors to CL methods and other merging strategies.

Task-agnostic results. Figure 4 presents task-agnostic results during continual learning for sequential fine-tuning, independent fine-tuning with model merging, and MagMax. We observe that model merging significantly reduces forgetting: sequential fine-tuning exhibited 25.7% forgetting on the first task and 22.3% on the second one while MagMax exhibited only 9.8% and 1.7%, respectively. Moreover, we observe significantly better performance on unseen tasks when using model merging.

Table 1: MagMax outperforms other continual learning methods and merging-based approaches on a wide variety of class-incremental scenarios. We report task-agnostic accuracy (%) after the final task. The best results are in **bold** and the second best underlined.

	CIFAR100				ImageNet-R				CUB200			Cars			Avg
Method	/5	/10	/20	/50	/5	/10	/20	/50	/5	/10	/20	/5	/10	/20	
Zero-shot	66.91				77.73				56.08			64.71			67.21
Joint	90.94				87.55				81.57			88.21			87.38
LwF	83.25	73.45	72.05	68.84	81.15	82.97	81.82	80.32	**65.12**	60.67	**58.90**	71.72	**69.84**	62.98	72.36
EWC	**84.41**	76.24	75.39	72.97	82.15	82.42	81.48	81.47	59.10	54.49	53.31	69.46	60.78	57.42	70.79
RandMix	81.55	77.04	75.36	72.91	83.10	81.88	80.18	78.50	59.86	58.53	58.08	67.32	65.62	64.95	71.78
MaxAbs	81.95	76.75	74.39	73.04	83.03	82.33	80.92	79.33	60.15	58.01	56.59	67.36	63.55	58.95	71.17
Avg	81.41	77.04	75.29	72.92	83.08	81.87	80.27	78.53	59.77	58.44	58.01	67.37	65.59	64.88	71.75
TIES	81.72	77.23	74.66	73.76	83.08	82.27	80.83	79.57	60.94	58.22	56.97	70.45	64.90	61.17	71.84
MagMax	84.16	**80.41**	**78.49**	**76.75**	**83.60**	**83.33**	**82.27**	**81.75**	63.89	**60.74**	**58.90**	**73.61**	69.28	**65.84**	**74.50**

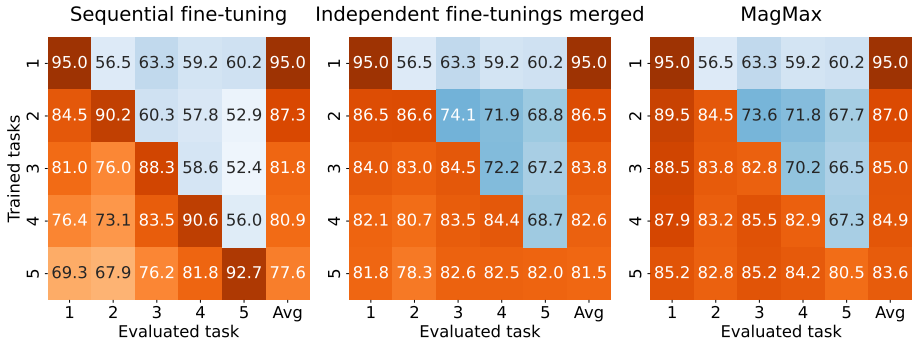

Figure 4: Sequential fine-tuning (left) exhibits high forgetting. Merging independent fine-tunings significantly reduces the forgetting (middle). MagMax further improves this issue (right). We present the results on already learned tasks in orange and zero-shot performance in blue. We report task-agnostic accuracy (%) for each task (columns) after training on the subsequent tasks (rows). The last column is an average accuracy on already seen tasks (lower triangular matrix in orange).

Method	DomainNet CIL	DomainNet DIL	ImageNet-R CIL	ImageNet-R DIL
LwF	**69.59**	69.67	82.05	84.78
EWC	64.73	**70.74**	81.45	83.77
RandMix	65.60	64.31	80.85	82.28
MaxAbs	66.21	67.51	81.50	83.93
Avg	65.71	64.98	80.80	82.98
TIES	66.62	66.42	81.52	83.90
MagMax	69.18	69.00	**82.90**	**85.40**

Table 2: MagMax outperforms other merging-based methods in domain-incremental scenarios and achieves similar results to CL methods. We report task-agnostic accuracy (%) after the final task. The best results are in **bold** and the second best underlined.

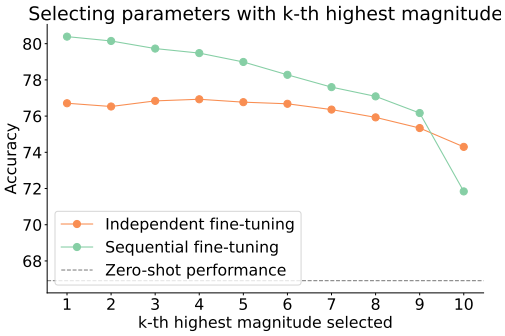

Figure 5: Selecting the highest magnitude parameters results in the best performance when merging sequentially fine-tuned models. We report the accuracy (%) of the model merged by selecting k-th highest magnitude.

Domain-incremental learning. Table 2 presents the results on domain-incremental learning benchmarks. MagMax outperforms other merging strategies in every scenario. It also achieves results on par with CL methods, outperforming them on ImageNet-R but slightly underperforming on DomainNet. We also observe that the top-performing methods achieve higher performance in domain-incremental scenarios than in class-incremental.

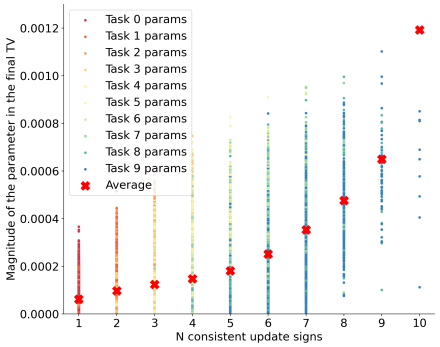

Figure 6: Magnitude of parameters of sequentially fine-tuned task vectors is correlated with the consistency of the update direction in the subsequent tasks. We report the results in CIFAR100/10 setting.

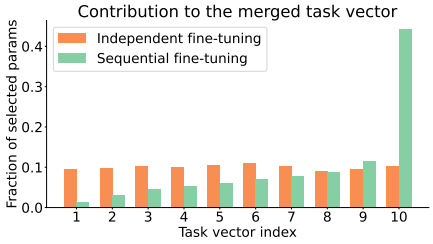

Figure 7: The contribution of parameters is nearly evenly distributed across task when fine-tuning independently. However, for sequential fine-tuning merging prioritizes the later task vectors which accumulated the knowledge about multiple tasks. We report the results in CIFAR100/10 setting.

Merging by k-th magnitude. In this section, we experimentally justify the choice of maximum magnitude when merging models. We perform experiments where we merge task vectors by selecting the parameters that have k-th highest magnitude, where $k = 1$ means maximum magnitude selection. We perform these evaluations for both independent and sequential fine-tuning scenarios. We also normalize the resulting task vectors so they have an equal norm for $k \in \{1, \ldots, N\}$. We present the results in Figure 5. We observe that when fine-tuning independently, the results for $k \in \{1, \ldots, 8\}$ vary by only 1%. It means that the directions of updates defined by the resulting task vectors are similarly beneficial for the final performance. It suggests that parameters of independently fine-tuned models are either redundant (they serve the same purpose therefore the performance does not change) or concurrent (they serve concurrent task-specific purposes). However, for sequential fine-tuning, the performance decreases as k increases. It means that parameters with high magnitude are better indicators of the beneficial update direction than parameters with lower magnitude.

Selecting high magnitude parameters promotes consistent update directions. In this Section we set and verify the following hypothesis: *parameters which update directions were consistent across tasks tend to have higher magnitude.* We define an update direction as a sign of parameter change when trained on a given task, $\text{sgn}(\Delta\theta_t^p) = \text{sgn}(\theta_t^p - \theta_{t-1}^p)$. For each parameter in each sequentially fine-tuned task vector, we calculate the number of consistent update directions n. Figure 6 presents the relation of magnitude of task vectors' parameters and the consistency of update directions. We observe that the parameters with higher consistency tend to have higher magnitude. Therefore, we can think of maximum magnitude selection as a proxy for selecting the updates that multiple tasks agree on.

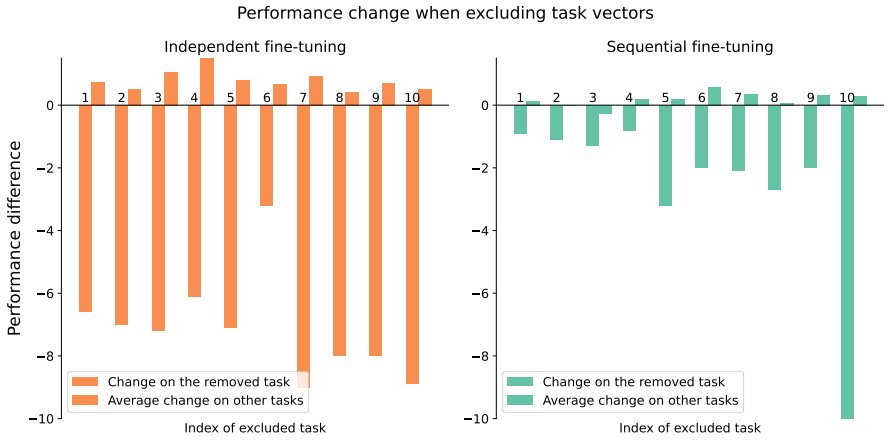

Figure 8: In an independent fine-tuning setting, the exclusion of a single task vector causes significant performance loss on the corresponding task. Models fine-tuned sequentially are much more robust to such an exclusion of non-last task vectors. It shows that the knowledge of previous tasks is partially retained in the later task vectors. We report the difference in accuracy (%) between the model merged out of 10 task vectors and models merged out of 9 task vectors.

Contributions of task vectors. In this section, we present insights into the contributions of the particular task vectors to the final model. Firstly, we perform task vector exclusion experiments in the CIFAR100/10 setting. We merge 9 task vectors, excluding one of them, and compare it to the performance of 10 task vectors merged. We present the results in Figure 8. We observe that for independent fine-tuning, removal of one task vector results in significant performance loss on the corresponding task. However, for sequentially fine-tuned models, the exclusion of a single task vector hurts the performance on the corresponding task much less. The only exception is the exclusion of the last task vector which uniquely contains the knowledge about the last task. This shows that the later task vectors retain some of the information about the previous tasks and the previous task vectors are less critical than when fine-tuning independently.

We demonstrate that this observation corresponds to the extent of contribution from the task vectors towards the merged model, which is quantified as the proportion of parameters chosen for the composite task vector. Figure 7 illustrates these contributions for the CIFAR100/10 experiment. When task-specific models are fine-tuned independently, their contributions are nearly evenly distributed. Yet, in the scenario where the model undergoes sequential fine-tuning, the contribution escalates with the task index, favoring models that have been fine-tuned across an increased number of tasks.

Sensitivity to scaling factor. Exemplar-free continual learning forbids storing data from previous tasks. Therefore we are not able to choose scaling factor λ

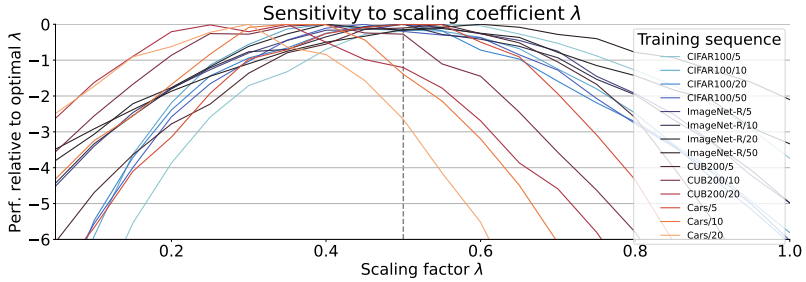

Figure 9: MagMax is fairly stable across different scenarios when it comes to scaling coefficient λ. We report the accuracy (%) relative to the accuracy with optimal λ.

based on validation sets from all tasks as in [13] and we set a constant $\lambda = 0.5$ for our method. Figure 9 presents a sensitivity analysis of the scaling factor. We calculate the difference of the performance for $\lambda \in \{0.05, 0.1, \ldots, 0.95, 1.0\}$ from the performance given an optimal λ. We observe that for 11 out of 14 scenarios, the results for selected $\lambda = 0.5$ differ less than 0.5% from the optimal selection. There are, however, several scenarios where selecting a better scaling coefficient would considerably improve the results. Note, that we only tuned λ on CIFAR100/5 experiment and used the selected value across all other experiments (similar to other methods).

7 Extended analysis

In this Section, we broaden the scope of our analysis. We investigate the impact of maximum magnitude selection merging on existing CL methods. We also study the impact of sequential fine-tuning on other model merging strategies in both CIL setting and on the popular eight datasets benchmark.

Does model merging help CL methods? In this section, we investigate if knowledge consolidation via model merging helps to improve the performance of CL methods. We modify MagMax and instead of performing sequential fine-tuning, we train the model using one of the regularization-based CL methods. We present the results in Table 3. We observe that adding model merging significantly improves the performance of LwF and EWC in almost every scenario. Interestingly, neither of these combinations significantly outperform MagMax which uses naive sequential fine-tuning, traditionally known for causing catastrophic forgetting [7, 27]. These results show that model merging is a promising technique for consolidating the knowledge *after* the training instead of *during* the training.

Sequential fine-tuning improves various merging methods. In this Section, we investigate how well sequential fine-tuning combines with different merging methods. Table 4 presents the results of merging independent and sequential

Table 3: Knowledge consolidation step from MagMax improves the performance of regularization-based CL methods. However, these combinations achieve an average performance on par with MagMax. It suggests that forgetting mitigation techniques are less important when the knowledge is consolidated via model merging.

Method	CIFAR100 /5	CIFAR100 /10	CIFAR100 /20	CIFAR100 /50	ImageNet-R /5	ImageNet-R /10	ImageNet-R /20	ImageNet-R /50	CUB200 /5	CUB200 /10	CUB200 /20	Cars /5	Cars /10	Cars /20	Avg
LwF	83.25	73.45	72.05	68.84	81.15	82.97	81.82	80.32	65.12	60.67	58.89	71.72	69.84	62.98	72.36
LwF + MagMax	82.68	77.61	75.81	72.65	82.55	82.52	81.98	80.63	64.53	61.17	59.60	73.29	71.04	67.85	73.85
Δ	-0.57	+4.16	+3.76	+3.81	+1.40	-0.45	+0.16	+0.31	-0.59	+0.50	+0.71	+1.57	+1.20	+4.87	+1.49
EWC	84.41	76.24	75.39	72.97	82.15	82.42	81.48	81.47	59.10	54.49	53.31	69.46	60.78	57.42	70.79
EWC + MagMax	82.34	77.73	77.66	77.03	82.07	83.02	82.35	81.60	63.57	60.61	59.15	72.83	69.59	66.00	73.97
Δ	-2.07	+1.49	+2.27	+4.06	-0.08	+0.60	+0.87	+0.13	+4.47	+6.12	+5.84	+3.37	+8.81	+8.58	+3.18
MagMax	84.16	80.41	78.49	76.75	83.60	83.33	82.27	81.75	63.89	60.74	58.90	73.61	69.28	65.84	74.50

Table 4: Different merging methods combined with independent (Ind) and sequential (Seq) fine-tuning. RandMix and Avg benefit from sequential fine-tuning in most of the scenarios while TIES and MagMax benefit in all of the evaluated scenarios. The best results are in **bold**.

Method	FT	CIFAR100 /5	CIFAR100 /10	CIFAR100 /20	CIFAR100 /50	ImageNet-R /5	ImageNet-R /10	ImageNet-R /20	ImageNet-R /50	CUB200 /5	CUB200 /10	CUB200 /20	Cars /5	Cars /10	Cars /20	Avg
RandMix	Ind	81.55	77.04	75.36	72.91	83.10	81.88	80.18	78.50	59.86	58.53	58.08	67.32	65.62	64.95	71.78
	Seq	82.70	79.17	77.66	76.48	82.63	82.67	82.18	81.73	62.39	58.18	57.32	71.83	65.90	62.18	73.07
	Δ	+1.15	+2.13	+2.30	+3.57	-0.47	+0.79	+2.00	+3.23	+2.53	-0.35	-0.76	+4.51	+0.28	-2.77	+1.29
Avg	Ind	81.41	77.04	75.29	72.92	83.08	81.87	80.27	78.53	59.77	58.44	58.01	67.37	65.59	64.88	71.75
	Seq	82.68	79.12	77.60	76.46	82.60	82.60	82.18	81.65	62.50	58.20	57.34	71.88	65.94	62.01	73.05
	Δ	+1.27	+2.08	+2.31	+3.54	-0.48	+0.73	+1.91	+3.12	+2.73	-0.24	-0.67	+4.51	+0.35	-2.87	+1.30
TIES	Ind	81.72	77.23	74.66	73.76	83.08	82.27	80.83	79.57	60.94	58.22	56.97	70.45	64.90	61.17	71.84
	Seq	83.03	80.04	77.77	76.28	83.28	82.50	**82.27**	81.23	63.32	**60.77**	**59.36**	**73.90**	**70.40**	**67.06**	74.37
	Δ	+1.31	+2.81	+3.11	+2.52	+0.20	+0.23	+1.44	+1.66	+2.38	+2.55	+2.39	+3.45	+5.50	+5.89	+2.53
MagMax	Ind	81.95	76.75	74.39	73.04	83.03	82.33	80.92	79.33	60.15	58.01	56.59	67.36	63.55	58.95	71.17
	Seq	**84.16**	**80.41**	**78.49**	**76.75**	**83.60**	**83.33**	**82.27**	**81.75**	**63.89**	60.74	58.90	73.61	69.28	65.84	**74.50**
	Δ	+2.21	+3.66	+4.10	+3.71	+0.57	+1.00	+1.35	+2.42	+3.74	+2.73	+2.31	+6.25	+5.73	+6.89	+3.33

fine-tunings with different methods in class-incremental scenarios. We observe that all merging methods benefit from sequential fine-tuning in most of the scenarios, achieving from 1.3% to 3.3% better average results. Table 5 presents the results of a similar experiment on eight dataset benchmark. We observe significant improvement (up to 12 p.p.) introduced by sequential fine-tuning. It shows that sequential fine-tuning can be beneficial even when the tasks are mostly dissimilar. Interestlingly, RandMix, Avg and TIES combined with Seq FT achieve very similar results, while MagMax outperforms them by over 3 p.p.

Starting point for fine-tuning. In this section, we investigate the relevance of the starting weights for fine-tuning when merging with MagMax. When fine-tune the model on task D_t, we experiment with starting from θ_0 (independent fine-tuning), θ_{t-1} (sequential fine-tuning) and θ_1. The last option follows the intuition that the model fine-tuned on the first task is adapted to the particular domain or task, *e.g.* bird species classification, and may serve as an appropriate

Table 5: Sequential fine-tuning leads to significant improvement over the independent fine-tuning even when tasks do not share many similarities. ΔAvg indicates the average gain from using sequential fine-tuning over independent fine-tuning when merging models with different strategies in 8 datasets scenario.

Method	FT	Cars	DTD	EuroSAT	GTSRB	MNIST	RESISC45	SUN397	SVHN	Avg	ΔAvg
RandMix	Ind	69.82	50.27	83.11	59.26	94.51	75.83	68.52	74.24	71.95	
	Seq	83.56	51.06	96.96	50.40	99.35	89.97	69.82	95.47	79.57	+7.62
Avg	Ind	57.11	45.53	75.00	66.78	98.78	62.48	46.07	88.60	67.54	
	Seq	83.58	51.54	96.81	50.36	99.35	90.16	69.88	95.52	79.65	+12.11
TIES	Ind	71.76	55.53	88.33	67.61	98.91	78.67	66.09	89.81	77.09	
	Seq	85.96	52.02	95.85	54.33	99.05	87.21	71.48	90.37	79.53	+2.44
MagMax	Ind	73.14	52.02	83.15	56.33	96.80	79.54	71.42	80.69	74.14	
	Seq	83.65	57.34	94.67	67.59	99.14	91.17	73.23	95.09	82.74	+8.60

starting point for future tasks that share some similarities. We present the results in Table 6. We observe that starting from θ_1 usually hurts the final performance of the model compared to independent fine-tuning except for fine-grained scenarios with sufficiently big first tasks (CUB200/5 and Cars/5). However, both of these approaches underperform compared to the sequential fine-tuning highlighting the importance of knowledge transfer.

Table 6: Starting fine-tuning from the model adapted to a single task (θ_1) does not improve the final performance compared to starting from pre-trained weights (θ_0). Training sequentially (starting from θ_{t-1}) achieves the best results.

Initial θ	CIFAR100				ImageNet-R				CUB200			Cars			Avg
for D_t	/5	/10	/20	/50	/5	/10	/20	/50	/5	/10	/20	/5	/10	/20	
θ_0	81.95	76.75	74.39	73.04	83.03	82.33	80.92	79.33	60.15	58.01	56.59	67.36	63.55	58.95	71.17
θ_1	81.42	75.59	70.64	69.27	82.43	81.37	79.70	78.53	61.48	57.51	55.70	70.45	61.66	59.48	70.37
θ_{t-1}	**84.16**	**80.41**	**78.49**	**76.75**	**83.60**	**83.33**	**82.27**	**81.75**	**63.89**	**60.74**	**58.90**	**73.61**	**69.28**	**65.84**	**74.50**

8 Conclusion

In this paper, we introduced MagMax, a novel approach to continual learning that leverages model merging via maximum magnitude selection alongside sequential fine-tuning. Our findings underscore the potential of model merging as a viable solution to the challenges of continual learning. The synergy between sequential fine-tuning and maximum magnitude weight selection emerges as a pivotal factor in this success. It opens up possibilities for future research direction focused on developing fine-tuning methods that facilitate model merging or finding new, more effective strategies for selecting important parameters in realms of continual learning.

Acknowledgments

Daniel Marczak is supported by National Centre of Science (NCN, Poland) Grant No. 2021/43/O/ST6/02482. This research was partially funded by National Science Centre, Poland, grant no: 2020/39/B/ST6/01511, 2022/45/B/ST6/02817 and 2023/51/D/ST6/02846. Bartłomiej Twardowski acknowledges the grant RYC2021-032765-I. This paper has been supported by the Horizon Europe Programme (HORIZON-CL4-2022-HUMAN-02) under the project "ELIAS: European Lighthouse of AI for Sustainability", GA no. 101120237. We gratefully acknowledge Polish high-performance computing infrastructure PLGrid (HPC Center: ACK Cyfronet AGH) for providing computer facilities and support within computational grant no. PLG/2023/016393.

References

1. Carion, N., Massa, F., Synnaeve, G., Usunier, N., Kirillov, A., Zagoruyko, S.: End-to-end object detection with transformers. ECCV (2020)
2. Caron, M., Touvron, H., Misra, I., Jegou, H., Mairal, J., Bojanowski, P., Joulin, A.: Emerging properties in self-supervised vision transformers. ICCV (2021)
3. Cheng, G., Han, J., Lu, X.: Remote sensing image scene classification: Benchmark and state of the art. Proceedings of the IEEE (2017)
4. Cimpoi, M., Maji, S., Kokkinos, I., Mohamed, S., Vedaldi, A.: Describing textures in the wild. In: CVPR (2014)
5. Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., Houlsby, N.: An image is worth 16x16 words: Transformers for image recognition at scale. In: ICLR (2021)
6. Frankle, J., Dziugaite, G.K., Roy, D.M., Carbin, M.: Linear mode connectivity and the lottery ticket hypothesis. In: ICML (2020)
7. French, R.M.: Catastrophic forgetting in connectionist networks. Trends in Cognitive Sciences (1999)
8. Guo, Y., Yao, A., Chen, Y.: Dynamic network surgery for efficient DNNs. NeurIPS (2016)
9. Han, S., Pool, J., Tran, J., Dally, W.J.: Learning both weights and connections for efficient neural network. In: NeurIPS (2015)
10. Helber, P., Bischke, B., Dengel, A., Borth, D.: Eurosat: A novel dataset and deep learning benchmark for land use and land cover classification. Journal of Selected Topics in Applied Earth Observations and Remote Sensing (2019)
11. Hendrycks, D., Basart, S., Mu, N., Kadavath, S., Wang, F., Dorundo, E., Desai, R., Zhu, T.L., Parajuli, S., Guo, M., Song, D., Steinhardt, J., Gilmer, J.: The many faces of robustness: A critical analysis of out-of-distribution generalization. ICCV (2020)
12. Hou, S., Pan, X., Loy, C.C., Wang, Z., Lin, D.: Learning a unified classifier incrementally via rebalancing. In: CVPR (2019)
13. Ilharco, G., Ribeiro, M.T., Wortsman, M., Schmidt, L., Hajishirzi, H., Farhadi, A.: Editing models with task arithmetic. In: ICLR (2023)
14. Ilharco, G., Wortsman, M., Gadre, S.Y., Song, S., Hajishirzi, H., Kornblith, S., Farhadi, A., Schmidt, L.: Patching open-vocabulary models by interpolating weights. In: NeurIPS (2022)

15. Janson, P., Zhang, W., Aljundi, R., Elhoseiny, M.: A simple baseline that questions the use of pretrained-models in continual learning. In: NeurIPS 2022 Workshop on Distribution Shifts: Connecting Methods and Applications
16. Kim, D., Han, B.: On the stability-plasticity dilemma of class-incremental learning. In: CVPR (2023)
17. Kirillov, A., Mintun, E., Ravi, N., Mao, H., Rolland, C., Gustafson, L., Xiao, T., Whitehead, S., Berg, A.C., Lo, W.Y., Dollár, P., Girshick, R.: Segment anything. ICCV (2023)
18. Kirkpatrick, J., Pascanu, R., Rabinowitz, N., Veness, J., Desjardins, G., Rusu, A.A., Milan, K., Quan, J., Ramalho, T., Grabska-Barwinska, A., et al.: Overcoming catastrophic forgetting in neural networks. PNAS (2017)
19. Krause, J., Stark, M., Deng, J., Fei-Fei, L.: 3D Object representations for fine-grained categorization. In: ICCV Workshops (2013)
20. Krizhevsky, A., Hinton, G., et al.: Learning multiple layers of features from tiny images. University of Toronto (2009)
21. Lange, M.D., Aljundi, R., Masana, M., Parisot, S., Jia, X., Leonardis, A., Slabaugh, G., Tuytelaars, T.: A continual learning survey: Defying forgetting in classification tasks. IEEE TPAMI (2019)
22. LeCun, Y.: The MNIST database of handwritten digits (1998)
23. Li, Z., Hoiem, D.: Learning without forgetting. IEEE TPAMI (2018)
24. Marouf, I.E., Roy, S., Tartaglione, E., Lathuilière, S.: Weighted ensemble models are strong continual learners. arXiv preprint arXiv: 2312.08977 (2023)
25. Masana, M., Liu, X., Twardowski, B., Menta, M., Bagdanov, A.D., van de Weijer, J.: Class-incremental learning: Survey and perfoxrmance evaluation on image classification. IEEE TPAMI (2023)
26. Matena, M., Raffel, C.: Merging models with fisher-weighted averaging. In: NeurIPS (2021)
27. McCloskey, M., Cohen, N.J.: Catastrophic interference in connectionist networks: The sequential learning problem. In: Psychology of Learning and Motivation (1989)
28. Netzer, Y., Wang, T., Coates, A., Bissacco, A., Wu, B., Ng, A.Y.: Reading digits in natural images with unsupervised feature learning. In: NeurIPS Workshops (2011)
29. Ortiz-Jiménez, G., Favero, A., Frossard, P.: Task arithmetic in the tangent space: Improved editing of pre-trained models. In: NeurIPS (2023)
30. Peng, X., Bai, Q., Xia, X., Huang, Z., Saenko, K., Wang, B.: Moment matching for multi-source domain adaptation. In: ICCV (2019)
31. Petit, G., Popescu, A., Schindler, H., Picard, D., Delezoide, B.: Fetril: Feature translation for exemplar-free class-incremental learning. In: WACV (2023)
32. Radford, A., Kim, J.W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., Krueger, G., Sutskever, I.: Learning transferable visual models from natural language supervision. In: ICML (2021)
33. Rusu, A.A., Rabinowitz, N.C., Desjardins, G., Soyer, H., Kirkpatrick, J., Kavukcuoglu, K., Pascanu, R., Hadsell, R.: Progressive neural networks. CoRR (2016)
34. Rypeść, G., Cygert, S., Khan, V., Trzciński, T., Zieliński, B., Twardowski, B.: Divide and not forget: Ensemble of selectively trained experts in continual learning. In: ICLR (2024)
35. Shah, V., Ruiz, N., Cole, F., Lu, E., Lazebnik, S., Li, Y., Jampani, V.: Ziplora: Any subject in any style by effectively merging loras. In: arXiv preprint arxiv:2311.13600
36. Singh, S.P., Jaggi, M.: Model fusion via optimal transport. In: NeurIPS (2020)

37. Smith, J.S., Karlinsky, L., Gutta, V., Cascante-Bonilla, P., Kim, D., Arbelle, A., Panda, R., Feris, R., Kira, Z.: Coda-prompt: Continual decomposed attention-based prompting for rehearsal-free continual learning. In: CVPR (2023)
38. Stallkamp, J., Schlipsing, M., Salmen, J., Igel, C.: The german traffic sign recognition benchmark: a multi-class classification competition. In: IJCNN (2011)
39. van de Ven, G., Tuytelaars, T., Tolias, A.: Three types of incremental learning. Nature Machine Intelligence (2022)
40. Wah, C., Branson, S., Welinder, P., Perona, P., Belongie, S.: The caltech-ucsd birds-200-2011 dataset (2011)
41. Wang, C.Y., Bochkovskiy, A., Liao, H.: Yolov7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors. CVPR (2022)
42. Wang, F., Zhou, D., Ye, H., Zhan, D.: FOSTER: feature boosting and compression for class-incremental learning. In: ECCV (2022)
43. Wang, Z., Zhang, Z., Lee, C., Zhang, H., Sun, R., Ren, X., Su, G., Perot, V., Dy, J.G., Pfister, T.: Learning to prompt for continual learning. In: CVPR (2022)
44. 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: ICML (2022)
45. Wortsman, M., Ilharco, G., Kim, J.W., Li, M., Kornblith, S., Roelofs, R., Lopes, R.G., Hajishirzi, H., Farhadi, A., Namkoong, H., Schmidt, L.: Robust fine-tuning of zero-shot models. In: CVPR (2022)
46. Xiao, J., Ehinger, K.A., Hays, J., Torralba, A., Oliva, A.: Sun database: Exploring a large collection of scene categories. IJCV (2016)
47. Yadav, P., Tam, D., Choshen, L., Raffel, C., Bansal, M.: TIES-merging: Resolving interference when merging models. In: NeurIPS (2023)
48. Yan, S., Xie, J., He, X.: DER: Dynamically expandable representation for class incremental learning. In: CVPR (2021)
49. Yoon, J., Yang, E., Lee, J., Hwang, S.J.: Lifelong learning with dynamically expandable networks. In: ICLR (2018)
50. Zhai, X., Mustafa, B., Kolesnikov, A., Beyer, L.: Sigmoid loss for language image pre-training. ICCV (2023)
51. Zhang, G., Wang, L., Kang, G., Chen, L., Wei, Y.: SLCA: slow learner with classifier alignment for continual learning on a pre-trained model. In: ICCV (2023)
52. Zhao, B., Xiao, X., Gan, G., Zhang, B., Xia, S.: Maintaining discrimination and fairness in class incremental learning. In: CVPR (2020)
53. Zhu, K., Zhai, W., Cao, Y., Luo, J., Zha, Z.J.: Self-sustaining representation expansion for non-exemplar class-incremental learning. In: CVPR (2022)