Weighted Ensemble Models Are Strong Continual Learners

Imad Eddine Marouf[1], Subhankar Roy[2], Enzo Tartaglione[1], and Stéphane Lathuilière[1]

[1]Télécom-Paris, Institut Polytechnique de Paris, [2]University of Trento, Italy
[1]`{first.lastname}@telecom-paris.fr`
[2]`{first.lastname}@unitn.it`

Abstract. In this work, we study the problem of continual learning (CL) where the goal is to learn a model on a sequence of tasks, under the assumption that the data from the previous tasks becomes unavailable while learning on the current task data. CL is essentially a balancing act between learning on the new task (*i.e.* plasticity) and maintaining the performance on the previously learned concepts (*i.e.* stability). To address the stability-plasticity trade-off, we propose to perform weight-ensembling of the model parameters of the previous and current tasks. This weighted-ensembled model, which we call **Co**ntinual **M**odel **A**veraging (or **CoMA**), attains high accuracy on the current task by leveraging plasticity, while not deviating too far from the previous weight configuration, ensuring stability. We also propose an improved variant of CoMA, named **Co**ntinual **Fi**sher-weighted **M**odel **A**veraging (or **CoFiMA**), that selectively weighs each parameter in the weights ensemble by leveraging the Fisher information of the weights of the model. Both variants are conceptually simple, easy to implement, and effective in attaining state-of-the-art performance on several standard CL benchmarks. Code is available at: https://github.com/IemProg/CoFiMA.

Keywords: Continual Learning · Model Averaging.

1 Introduction

Continually learning from a sequence of tasks with a unified model is a challenging problem due to *catastrophic forgetting* (CF) [15] – a phenomenon that is marked by deterioration of performance on previously seen data. Continual learning (CL) has emerged as a solution to CF that allows models to assimilate information from new tasks while retaining classification capability for the previously learned classes [36]. Until recently, CL approaches predominantly focused on relatively small networks, often ResNets [18], starting from random initialization [36, 75]. Lately, the prominence of large Pre-Trained Models (PTMs) [11, 27, 51] – Vision Transformer (ViT) [27, 35] pre-trained on large datasets (*e.g.* ImageNet [55], LAION-400M [58]) – has led to an influx of CL methods that are leveraging the strong representation of the PTMs, causing a

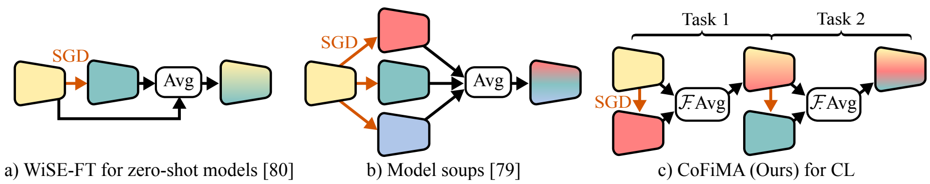

Fig. 1: Comparison of existing model averaging techniques with our proposed technique for CL. (a) Averaging the weights of the pre-trained model and the one fine-tuned leads to simultaneous improvement in out-of-distribution and target dataset performance. (b) Models soups combines multiple fine-tuned models resulting in a robust unified model. (c) In the proposed CoFiMA the weights of the current and past models are weighted based on their Fisher Information matrices (represented by $\mathcal{F}$), resulting in balanced performance for both the current and old tasks.

paradigm shift in CL [69, 76–78, 87]. In detail, numerous PTM-based CL methods [10, 40, 81] have empirically validated that a good initial representation, obtained with the help of large and diverse pre-training, facilitates incremental learning since new tasks can be learned with fewer training steps. However, sequential full fine-tuning of the PTM backbone results in deterioration of the original PTM representation, alongside significant forgetting on the previously learned tasks [38, 47, 85, 87]. To counteract overfitting, many methods have been proposed that either heuristically confine the PTM fine-tuning only to the first adaptation session [38, 47, 87] or carefully choose a low learning rate to fine-tune the backbone [85]. Nevertheless, achieving a *satisfactory balance between quickly accruing knowledge on new tasks while preserving the generalizability of the PTMs* remains an open research question.

In the quest for achieving robust fine-tuning of PTMs, several studies have investigated the application of weight averaging (WA) methods [1, 25, 37, 42, 79, 80]. The essence of these methods is to ensemble several fine-tuned PTMs to obtain a single model that encapsulates the representational capabilities of multiple models. While, to our knowledge, WA has not yet been investigated for CL, two recent studies [79, 80] have sparked our interest in examining the feasibility of WA for CL. Firstly, WiSE-FT [80] improves the robustness of the fine-tuning procedure of zero-shot classifiers like CLIP by averaging the weights of both fine-tuned and pre-trained models (Fig. 1(a)). Here, the resulting ensemble exhibits high accuracy on the target distribution while preserving the out-of-distribution (OOD) performance of the original PTM. It is important to recognize that model robustness and CL intersect: the balance between *OOD performance and target performance* mirrors the CL *stability-plasticity trade-off*, where the goal is to accommodate new tasks while preserving efficacy on prior tasks [1, 25, 42]. Secondly, "Model soups" [79] shows that combining multiple fine-tunings of the same PTM through WA enhances performance on both in- and out-of-distribution tasks (Fig. 1(b)). Their experiments underscore the potential of WA with a relatively large pool of models (*e.g.* 32), mirroring the number of tasks typically encountered in CL. Nevertheless, Model soups do not establish the feasibility of using WA when models are trained on different tasks, a capability required for CL.

Motivated by these observations, *we cast PTM-based CL as a robust fine-tuning problem* and propose **Co**ntinual **M**odel **A**veraging (**CoMA**) as a response to the PTM fine-tuning conundrum in CL. In CoMA we linearly combine the weights from the previous task $t-1$ with the weights of the fully fine-tuned model from the current task t (Fig. 1(c)). Full fine-tuning on the current task ensures plasticity, while ensembling the weights with the previous task checkpoint preserves performance on the previous task. The ensemble at task t is then used as an initialization for the next task $t+1$. In essence, our proposed CoMA exploits the model averaging techniques [79, 80] to the CL scenario by extending the model averaging to a sequence of downstream tasks. Intuitively, CoMA is effective because task-specific models optimized sequentially could lie in the same basin of the total loss landscape [41, 43].

The model averaging in CoMA assumes that all the network's weights have the same importance for a given task. While effective, putting equal importance to all weights could result in a weight-ensembled model that lies in an error basin with high loss [79]. To mitigate this problem, we aim to selectively ensemble the weights based on the importance of each weight parameter for a given task. Inspired by Elastic Weight Consolidation (EWC) [26], and the more recent work of Matena *et al*. [37], we leverage the Fisher information [9, 62] to weigh the model parameters during model averaging. Fisher information inherently captures the importance of each weight parameter on the dataset (or task) the model has been trained on. We call this variant of CoMA as **Co**ntinual **Fi**sher-weighted **M**odel **A**veraging (**CoFiMA**), and shown in Fig. 1(c). CoFiMA stores only the Fisher values of the previous task, thereby rendering Fisher-weighted averaging compatible with CL constraints. CoFiMA maintains computational efficiency, necessitating only a singular forward and backward pass on the data of the current task to estimate Fisher values [26, 48] and yields state-of-the-art performance surpassing both CoMA and existing PTM-based CL solutions.

Our **contributions** are summarised as follows:

- We draw a parallel between *robust fine-tuning* and *continual learning* and show that model-averaging is a simple yet effective solution for *PTM-based* CL problem. We propose CoMA, a weight-ensemble inspired CL approach, that addresses the challenging task of stability-plasticity trade-off.
- We extend CoMA to CoFiMA by employing Fisher information to adaptively weigh the parameters of the previous and current task model.
- We run extensive experiments on several standard CL benchmarks and demonstrate that CoFiMA yet being simple, it consistently outperforms PTM-based CL solutions.

2 Related Work

Continual Learning with PTM. Not a long time ago, the predominant focus in CL has been on the sequential training of deep neural networks from scratch, aiming to proficiently acquire new tasks while mitigating forgetting of preceding tasks. Typical CL strategies encompass *regularization*-based approaches [2, 8, 26, 33, 84], which maintain the initial model and selectively stabilize parameter or prediction alterations; *replay*-based approaches [4, 50, 72, 82], that seek to approximate and regenerate previously learned data distributions; and *architecture*-based approaches [56, 59, 83], which allocate discrete parameter sub-spaces for each task.

Differently, the recent trajectory of CL research has probed into the advantages of PTMs [10, 77, 78]. Representations derived from pre-training have demonstrated the capacity to facilitate not only knowledge transfer but also resilience against catastrophic forgetting during downstream continual learning [40, 52]. Moreover, learning on substantial base classes during the pre-training phase permits CL with minimal adaptations [81]. For example, L2P [78] leveraged techniques inspired by pre-trained knowledge utilization in NLP, employing an additional set of learnable parameters, termed "prompts", which guide a pre-trained representation layer in learning incremental tasks. DualPrompt [77] elaborated on this concept by attaching supplementary prompts to the pre-trained representation layer to facilitate the learning of both task-invariant and task-specific instructions. Though prompt-based approaches have been documented to significantly outperform conventional CL baselines, they introduce an extra inference cost. Recently, Zhang *et al.* [85] showed that sequential fine-tuning with a small learning rate using PTMs outperforms traditional CL approaches. Wang *et al.* [73, 74] propose an architecture that employs multiple narrower sub-networks to manage incremental tasks, effectively reducing generalization errors in CL. However, this approach introduces increased complexity.

Unlike prior approaches in CL that enhance PTMs using prompts [77, 78], ensemble of experts [57, 73, 74] or replay-buffers [4, 50, 72, 82], CoFiMA employs a different strategy. CoFiMA enables plasticity by unlocking all model parameters to be fine-tuned unlike [38, 77, 78]. Furthermore, it reduces forgetting during training by averaging the weights of parameters from previous models. In contrast to EWC [26], which uses the Fisher Information Matrix (FIM) as a regularization term for L2-transfer between tasks, CoFiMA applies the FIM as a weighting factor to assess the significance of weights for each task, without any regularization constraints (more details in supplementary material).

Output-space/weight-space ensembles. Traditional ensemble methods, or output-space ensembles, combine multiple classifiers' predictions, often outperforming single models and providing more calibrated uncertainty estimates under distribution shifts [16, 31, 37, 46, 64]. These output-space ensembles, however, demand substantial computational resources at inference. Weight-space ensembles offer a computationally efficient alternative by interpolating between model weights [22, 44, 65, 79, 80]. Wortsman *et al.* [80] achieved this by interpolating between zero-shot CLIP and fine-tuned model weights, resulting in performance gains on both the fine-tuning task and under-distribution shifts. Matena *et al.* [37] propose an advanced WA technique that uses Fisher-values for different text classification tasks but they do not investigate the feasibility of this approach for CL. In Federated Learning (FL), model averaging, notably FedAvg, is a fundamental technique for amalgamating insights from decentralized data while upholding privacy [39]. This approach involves training local models on distributed nodes and averaging their parameters to update a global model, thus enhancing learning efficiency and data privacy [24, 32, 39]. Our approach differs from existing WA approaches [37, 79, 80] as it focuses on sequential fine-tuning with weight-averaging to be accustomed to the CL setting, we iteratively perform our procedure once per task, using the averaged model at each task as the initialization for the next task.

WA is also strongly related to the notion of linear model connectivity, introduced by Frankle *et al*. [14]. This concept identifies a condition where the accuracy levels are maintained throughout the linear interpolation between the weights of two separate networks. Interpolation of neural network weights has been shown to maintain high accuracy across various scenarios, along a shared optimization trajectory [7, 12, 14, 22, 79, 80]. Analogously, Neyshabur *et al*. [43] demonstrates that there exists a connection between minima obtained by pre-trained models versus freshly initialized ones. They note that there is no performance barrier between solutions coming from pre-trained models, but there can be a barrier between solutions of different randomly initialized models. Mizradeh *et al*. [41] investigated linear connectivity in the context of multi-task learning and CL. They show that there exists a linear path solution between two models trained on two tasks "A" and "B": one excelling in task A, and the other fine-tuned on both tasks A and B. These works offer a solid foundation of both theoretical and empirical evidence supporting the efficacy of linear interpolation in model performance. Building on these findings, our work presents a novel solution tailored to CL, an area yet to be explored.

3 Method

3.1 Problem formulation and overview

Continual learning with pre-trained models. We consider a classification model $M_\varphi(\cdot)=h(f(\cdot))$, where $f(\cdot)$ is a feature extractor and $h(\cdot)$ is a classification head, both parameterized by a unified set of parameters φ. The feature extractor $f_{\boldsymbol{\theta}}$ is initialized with parameters $\boldsymbol{\theta}_0$ of a PTM and then trained sequentially on a series of incremental tasks, each represented by the corresponding training set $\mathcal{D}_t$, for $t \in \{1, \dots, T\}$. The primary objective is to achieve robust performance across the test sets associated with these tasks. Specifically, for each task t, the dataset $\mathcal{D}_t$ is defined as $\mathcal{D}_t = \bigcup_{c\in C_t} \mathcal{D}_t^c$ where $\mathcal{D}_t^c = (\mathbf{x}_n^c, \boldsymbol{y}_n^c)_{n=1}^{N_c}$, and C_t denotes the set of novel classes introduced in task t. Here, N_c represents the number of training instances for each class c, with $(\mathbf{x}_n^c, \boldsymbol{y}_n^c)$ denoting the n-th training instance and its corresponding label. In Class-Incremental Learning (CIL), evaluation is conducted across all the observed classes without the need for task-index labels [68].

This problem poses two primary challenges: (i) the necessity to adapt the knowledge acquired from the PTM to new tasks; and (ii) the importance of

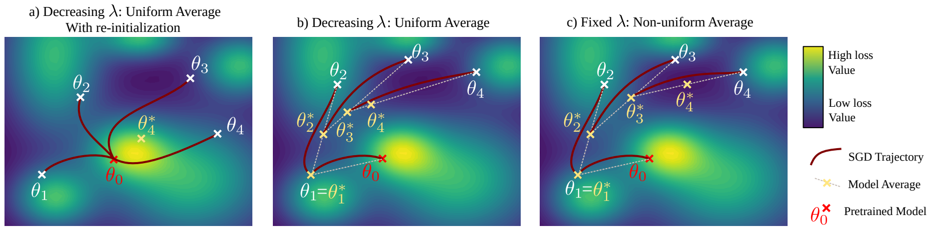

Fig. 2: Illustration of the parameter trajectory with model averaging in the loss landscape. **(a)** The trajectory where models are re-initialized at the start of each task leading to disparate solutions. **(b)** Depicts decreasing $\lambda = 1/t$: Uniform Averaging without re-initialization, showing the convergence of model parameters towards a solution that balances between tasks. **(c)** Recent tasks are given more weight, resulting in a solution that remains close to the latest task's model while considering previous tasks.

maintaining the model's comprehensive learning capabilities to avoid forgetting previously acquired knowledge while assimilating new tasks.

Overview. In this work we propose *model averaging* as an effective solution for PTM-based CIL. As fine-tuning a PTM on a new task causes the weights to deviate from the original PTM and previous task configuration, model averaging avoids forgetting on a previous task and maintains the generalizability of PTM by averaging the weights of the models of the previous task and current task. Each task concludes with inference on all tasks seen so far using the averaged model and using it as an initialization for the next one. We name this approach *Continual Model Averaging* (CoMA) and will be discussed in detail in Sec. 3.2.

We extend CoMA by introducing Continual Fisher-weighted Model Averaging (CoFiMA), wherein we average two models at any task t based on an additional weighing coefficient that is determined by Fisher information [9, 62]. Fisher information of a given model parameter dictates the importance of that particular weight towards a task. This procedure is detailed in Sec. 3.3.

3.2 Continual Model Averaging (CoMA)

Temporarily disregarding the typical memory limitations of CIL, a practical approach through model ensembling is proposed as follows: individual models are trained for each task sequentially, starting from the pre-trained model, and the model is saved after each task. Suppose we have trained on a sequence of T tasks: this method will result in T distinct networks, each with its own set of parameters, denoted as $\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_T$. The objective is to create a composite neural network with parameters $\boldsymbol{\theta}$ ensuring good performance across all the tasks.

We model the posterior over the composite parameters $\boldsymbol{\theta}$, conditioned on the task-specific parameters $\boldsymbol{\theta}_t$, denoted by $p(\boldsymbol{\theta}|\boldsymbol{\theta}_t)$, as an isotropic Gaussian distribution $\boldsymbol{\theta} \sim \mathcal{N}(\boldsymbol{\theta}_t, \mathbf{I})$, where $\mathbf{I}$ is the identity matrix [37, 80]. We treat the models $\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_T$ as independent observations of the composite model $\boldsymbol{\theta}$ and maximizing the log-likelihoods of the posterior distribution of $\boldsymbol{\theta}$ over all T the tasks leads to the following optimization problem:

$$\boldsymbol{\theta}^* = \arg\max_{\boldsymbol{\theta}} \frac{1}{T} \sum_{t=1}^{T} \log p(\boldsymbol{\theta}|\boldsymbol{\theta}_t). \tag{1}$$

The solution to this optimization problem has closed-form and is the simple average of the model parameters [37,39]. With a slight abuse in notation, where we use the sum operator to denote the element-wise summation across sets, it can be written:

$$\boldsymbol{\theta}^* = \frac{1}{T} \sum_{t=1}^{T} \boldsymbol{\theta}_t. \tag{2}$$

In addition to the support of the well-established likelihood maximization framework, averaging models as in Eq. (2) gains further validation from the insights provided by Mirzadeh *et al.* [41]. Their research shows that when two models are each trained on distinct tasks, a model adept in both tasks often exists within the linear interpolation of their parameter spaces. Below, we detail how we adapt this model-averaging method to the constraints of CIL where the task data arrives sequentially, and without storing previous data.

First, initiating the training of each task from the same pre-trained model $\boldsymbol{\theta}_0$ can result in convergence to distinct regions within the parameter space [14,53]. This case is illustrated in Fig. 2(a). If the learned parameters are too distant, the Gaussian assumption of the posterior distribution is no longer valid, and averaging the network could lead to high-loss regions and poor performance of the aggregated model (see Sec. 4.3). Therefore, for each task t, we initiate finetuning from $\boldsymbol{\theta}^*_{t-1}$ rather than the initial pre-trained model $\boldsymbol{\theta}_0$. This change limits the risk of reaching distant parameter regions.

Second, one of the goals of our approach is to prevent the number of models from growing linearly with t. Therefore, we shift from simultaneous averaging all models $\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_T$ to the iterative computation of the average $\boldsymbol{\theta}^*_t$ at each task t:

$$\boldsymbol{\theta}^*_t = \lambda \boldsymbol{\theta}_t + (1-\lambda)\boldsymbol{\theta}^*_{t-1}, \tag{3}$$

where $\lambda \in [0,1]$. To initialize this recursion, we start at $t=1$ with $\boldsymbol{\theta}^*_1$ which is set to the model parameters obtained at the end of the first task. Note that, when $\lambda = \frac{1}{t}$, Eq. (3) is strictly equivalent to the average in Eq. (2) (see supplementary material), aligning with the maximum-likelihood solution and adhering to our initial hypothesis of isotropic Gaussian posteriors. However, our experiments highlight that such a parameter choice might result in suboptimal performances and that assigning variable weights to each model proves advantageous. This phenomenon might be attributed to the possibility of encountering a region with a high loss value in between all the task-specific models (see Fig. 2(b)). Therefore, we propose to perform a non-uniform averaging giving higher importance to the latest tasks as shown in Fig. 2(c). This is obtained by using a constant weight parameter λ. The motivation for giving more importance to the latest tasks is that early models are trained only on the first few tasks, while the last model has encoded the knowledge from both old and recent tasks via sequential fine-tuning (see supplementary material). In this way, it is expected that a better trade-off would exist along this trajectory, and the two endpoints $\boldsymbol{\theta}_{t-1}$ and $\boldsymbol{\theta}_t$ are smoothly connected without significant loss barrier or performance drop along the path [14].

In terms of memory, during each training phase, the storage overhead of our approach compared to naive sequential fine-tuning over all the tasks is limited to the size of a single model. However, when transitioning to subsequent tasks, only $\boldsymbol{\theta}_t^*$ requires storage.

Handling Classifier Parameters. In each novel task t, there are unique parameters (specifically, new head parameters associated with new classes) not present in the preceding models that are subject to averaging. To accommodate this, we restrict the averaging process (as in Eq. (3)) exclusively to the parameters that are common across models (this includes both backbone parameters and the shared portions of head parameters) while excluding the new head weights (pertaining to new classes) from the averaging.

3.3 Continual Fisher-weighted Model Averaging (CoFiMA)

Uniform weight-averaging operates under the implicit assumption that all parameters of the model have the same importance for the training task t. This assumption could potentially compromise model performance since different parameters can have various impacts on the network output [26,37]. To enhance the averaging process, we propose a more refined model averaging for CIL. Specifically, we employ the Fisher information matrix [9, 62], which encapsulates the quantity of information that observed data $\mathbf{x}$ provides about the network parameters $\boldsymbol{\theta}_t$. The Fisher information matrix $F_{\boldsymbol{\theta}_t}$ of a neural network $p_{\boldsymbol{\theta}_t}(\boldsymbol{y}|\mathbf{x})$ trained to predict an output $\boldsymbol{y}$ from input data $\mathbf{x}$ is computed as in [3]:

$$\mathbf{F}_{\boldsymbol{\theta}_t} = \mathbb{E}_{\mathbf{X}\sim\mathcal{D}_t}\big[\mathbb{E}_{\boldsymbol{y}\sim p(\boldsymbol{y}|\mathbf{X},\boldsymbol{\theta}_t)}[\nabla_{\boldsymbol{\theta}_t}\log p(\boldsymbol{y}|\mathbf{x},\boldsymbol{\theta}_t)\nabla_{\boldsymbol{\theta}_t}\log p(\boldsymbol{y}|\mathbf{x},\boldsymbol{\theta}_t)^T]\big]. \tag{4}$$

Given the computational cost of storing the full Fisher matrix, this study adopts the *diagonal* of the Fisher information matrix for practicality [26, 48, 62]. The computation of the diagonal Fisher is feasible within the same order of complexity as the standard back-propagation training process, as it necessitates only a single gradient computation per data point, corresponding to an extra epoch at the end of each task training. Through Monte-Carlo sampling over $\mathbf{x}$ [62,63], the diagonal Fisher matrix can be estimated as follows:

$$\hat{F}_{\boldsymbol{\theta}_t} = \frac{1}{N}\sum_{i=1}^{N}\mathbb{E}_{\boldsymbol{y}\sim p(\boldsymbol{y}|\mathbf{x}_i,\boldsymbol{\theta}_t)}\left[[\nabla_{\boldsymbol{\theta}_t}\log p(\boldsymbol{y}|\mathbf{x}_i,\boldsymbol{\theta}_t)]^2\right], \tag{5}$$

where the expectation over $\boldsymbol{y}$ is estimated from the data samples N. The estimation of this Fisher information matrix $F_{\boldsymbol{\theta}}$ is based on the assumption that the statistical properties of the data are locally similar around the parameter values [34, 61]. This assumption aligns with the Gaussian assumption previously utilized in deriving CoMA. Furthermore, as discussed in the work of Chizat *et al*. [6], the use of PTMs limits the risk of facing significant changes in parameter distributions which would violate the *locality assumption* [9].

We consider that the posterior of each task model $p(\boldsymbol{\theta}|\boldsymbol{\theta}_t)$ is defined as $\boldsymbol{\theta} \sim \mathcal{N}(\boldsymbol{\theta}_t, \mathbf{F}_t^{-1})$. As in Eq. (2), maximization of the average of the task-specific log-likelihoods brings us to the following closed-form solution:

$$\boldsymbol{\theta}^* = \frac{\sum_{t=1}^{T} \mathbf{F}_t \boldsymbol{\theta}_t}{\sum_{t=1}^{T} \mathbf{F}_t}, \tag{6}$$

with a slight abuse in notation, where we use the division operator to denote the element-wise division. As in Sec. 3.2, we adapt this formulation making it iterative and increasing the importance of the latest task in the parameter update. This brings us to the following update rule for θ_t^*:

$$\boldsymbol{\theta}_t^* = \frac{\lambda \mathbf{F}_t \boldsymbol{\theta}_t + (1-\lambda)\mathbf{F}_{t-1}\boldsymbol{\theta}_{t-1}^*}{\lambda \mathbf{F}_t + (1-\lambda)\mathbf{F}_{t-1}}, \tag{7}$$

where F_{t-1} is the Fisher matrix estimated with $\boldsymbol{\theta}_{t-1}^*$ on the data $\mathcal{D}_{t-1}$. In terms of memory requirements, this solution requires the storage of only the model parameters $\boldsymbol{\theta}_t^*$ and the Fisher matrix $\mathbf{F}_{t-1}$ in between two consecutive tasks t and $t-1$. Similarly to CoMA, the recursion is initialized at the end of the first task with $\boldsymbol{\theta}_1^* = \boldsymbol{\theta}_1$ and estimating $\mathbf{F}_1$ with $\boldsymbol{\theta}_1$ on $\mathcal{D}_1$.

4 Experiments

In this section, we first briefly describe the experimental setups and then present the experimental results.

4.1 Experimental Setups

Datasets and Settings. We conduct experiments on four CIL benchmarks: CIFAR-100 [30], ImageNet-R [19], CUB-200 [71], and Cars-196 [28]. The CIFAR-100 dataset [30] has 100 classes of natural images, each with 500 training images. The ImageNet-R dataset [19] includes images from 200 classes, divided into 24,000 for training and 6,000 for testing. These images, although related to ImageNet-21K, are considered challenging for the PTM because they are either hard examples from ImageNet or new images in different styles. The CUB-200 dataset [71] consists of images from 200 bird classes, with about 60 images per class, half for training and half for testing. The Cars-196 dataset [28] is made up of 196 types of car images, split into 8,144 for training and 8,040 for testing, maintaining a similar class ratio. The first two focus on fine-grained classifications, while the last two datasets (*i.e.*, CIFAR-100, and ImageNet-R) are standard benchmarks for CL. Following SLCA [85] we split each benchmark into 10 tasks. We report the results in the class-incremental setting, *i.e.* the task id is not known during inference.

Table 1: State-of-the-art comparison on CUB-200, Cars-196, CIFAR-100, and ImageNet-R using **ViT-B/16** [27] supervisedly pre-trained on ImageNet-21K [55].

Method	Memory -Free	CUB-200		Cars-196		CIFAR-100		ImageNet-R	
		Last-Acc	Inc-Acc	Last-Acc	Inc-Acc	Last-Acc	Inc-Acc	Last-Acc	Inc-Acc
Joint-Training	-	$88.00_{\pm 0.34}$	-	$80.31_{\pm 0.13}$	-	$93.22_{\pm 0.16}$	-	$79.60_{\pm 0.87}$	-
Prototype-classifier [23]	✓	$80.66_{\pm 0.00}$	$88.95_{\pm 0.00}$	$28.58_{\pm 0.00}$	$39.83_{\pm 0.00}$	$60.29_{\pm 0.00}$	$69.18_{\pm 0.00}$	$38.45_{\pm 0.00}$	$45.59_{\pm 0.00}$
GDumb [50]		$61.80_{\pm 0.77}$	$79.76_{\pm 0.18}$	$25.20_{\pm 0.84}$	$49.48_{\pm 0.74}$	$81.92_{\pm 0.15}$	$89.46_{\pm 0.94}$	$24.23_{\pm 0.35}$	$43.48_{\pm 0.49}$
DER++ [4]		$77.42_{\pm 0.71}$	$87.61_{\pm 0.09}$	$60.41_{\pm 1.76}$	$75.04_{\pm 0.57}$	$84.50_{\pm 1.67}$	$91.49_{\pm 0.61}$	$67.75_{\pm 0.93}$	$78.13_{\pm 1.14}$
BiC [82]		$81.91_{\pm 2.59}$	$89.29_{\pm 1.57}$	$63.10_{\pm 5.71}$	$73.75_{\pm 2.37}$	$88.45_{\pm 0.57}$	$93.37_{\pm 0.32}$	$64.89_{\pm 0.80}$	$73.66_{\pm 1.61}$
L2P [78]	✓	$62.21_{\pm 1.92}$	$73.83_{\pm 1.67}$	$38.18_{\pm 2.33}$	$51.79_{\pm 4.19}$	$82.76_{\pm 1.17}$	$88.48_{\pm 0.83}$	$66.49_{\pm 0.40}$	$72.83_{\pm 0.56}$
DualPrompt [77]	✓	$66.00_{\pm 0.57}$	$77.92_{\pm 0.50}$	$40.14_{\pm 2.36}$	$56.74_{\pm 1.78}$	$85.56_{\pm 0.33}$	$90.33_{\pm 0.33}$	$68.50_{\pm 0.52}$	$72.59_{\pm 0.24}$
EWC [26]	✓	$68.32_{\pm 2.64}$	$79.95_{\pm 2.28}$	$52.50_{\pm 3.18}$	$64.01_{\pm 3.25}$	$89.30_{\pm 0.23}$	$92.31_{\pm 1.66}$	$70.27_{\pm 1.99}$	$76.27_{\pm 2.13}$
LwF [33]	✓	$69.75_{\pm 1.37}$	$80.45_{\pm 2.08}$	$49.94_{\pm 3.24}$	$63.28_{\pm 1.11}$	$87.99_{\pm 0.05}$	$92.13_{\pm 1.16}$	$67.29_{\pm 1.67}$	$74.47_{\pm 1.48}$
Seq FT	✓	$68.07_{\pm 1.09}$	$79.04_{\pm 1.69}$	$49.74_{\pm 1.25}$	$62.83_{\pm 2.16}$	$88.86_{\pm 0.83}$	$92.01_{\pm 1.71}$	$71.80_{\pm 1.45}$	$76.84_{\pm 1.26}$
RanPAC [38]	✓	$85.82_{\pm 0.53}$	$91.47_{\pm 0.96}$	$53.84_{\pm 0.84}$	$66.39_{\pm 1.18}$	$90.09_{\pm 0.25}$	$93.31_{\pm 0.98}$	$72.62_{\pm 0.11}$	$78.35_{\pm 0.58}$
SLCA [85]	✓	$84.71_{\pm 0.40}$	$90.94_{\pm 0.68}$	$67.73_{\pm 0.85}$	$76.93_{\pm 1.21}$	$91.53_{\pm 0.28}$	$94.09_{\pm 0.87}$	$77.00_{\pm 0.33}$	$81.17_{\pm 0.64}$
CoMA (Ours)	✓	$85.95_{\pm 0.29}$	$90.75_{\pm 0.39}$	$73.35_{\pm 0.59}$	$78.55_{\pm 0.42}$	$92.00_{\pm 0.13}$	$94.12_{\pm 0.63}$	$77.47_{\pm 0.05}$	$81.32_{\pm 0.17}$
CoFiMA (Ours)	✓	**87.11**$_{\pm 0.56}$	**91.87**$_{\pm 0.69}$	**76.96**$_{\pm 0.64}$	**82.65**$_{\pm 0.96}$	**92.77**$_{\pm 0.24}$	**94.89**$_{\pm 0.94}$	**78.25**$_{\pm 0.26}$	**81.48**$_{\pm 0.56}$

Metrics. We report the average classification accuracy of all the classes ever seen after learning each incremental task (denoted as *Inc-Acc* (%)) and the accuracy after learning the last task (denoted as *Last-Acc* (%)).

Baselines and Competitors. We compare with the state-of-the-art PTM-based CIL methods L2P [78], DualPrompt [77], SLCA [85], and RanPAC [38]. We also utilize the same PTM as the initialization for classical CL methods GDumb [50], LwF [33], DER [4], BIC [82], and EWC [26]. Additionally, we report the following baselines: sequentially fine-tuning of the model (denoted as Seq FT), and Prototype-classifier [23], which is a cosine similarity classifier on the extracted features of the PTM. Joint-Training is an upper bound, where the model has been trained on all the tasks at the same time.

Implementation details. We adopted two kinds of PTMs in our experiments: a ViT-B/16 [27] backbone supervisedly pre-trained on ImageNet-21K [55], the default PTM unless otherwise stated; and ViT-B/16 backbone with self-supervised pre-training using MoCo-V3 [5] on ImageNet-1K. We follow the implementation of SLCA [85] that adapts a small learning rate of 0.0001 for the representation layer and 0.01 for the classification layer, along with the class-alignment strategy. We set the batch size to 128, $\lambda = 0.4$ for supervised and $\lambda = 0.2$ for self-supervised pre-training, in all our experiments.

4.2 State-of-the-art Comparison

This section analyzes the performance of CoMA and CoFiMA across various CL benchmarks. We report in Tabs. 1 and 2 the comparison of CoMA and CoFiMA with state-of-the-art CL methods using the ViT-B/16 backbone that was pre-trained supervisedly and unsupervisedly, respectively.

As shown in Tab. 1, our proposed CoMA consistently outperforms the best-performing CIL baselines, SLCA [85] and RanPAC [38], across all benchmarks. This confirms the benefits of model averaging in CIL. Moreover, CoFiMA, the improved variant of CoMA, shows even further performance improvement over

Table 2: State-of-the-art comparison on CUB-200, Cars-196, CIFAR-100, and ImageNet-R using **ViT-B/16** with self-supervised pre-training (**MoCo-V3** [5]) on ImageNet-1K.

Method	**Memory -Free**	**CUB-200** Last-Acc	Inc-Acc	**Cars-196** Last-Acc	Inc-Acc	**CIFAR-100** Last-Acc	Inc-Acc	**ImageNet-R** Last-Acc	Inc-Acc
Joint-Training	-	$79.55_{\pm 0.04}$	-	$74.52_{\pm 0.09}$	-	$89.11_{\pm 0.06}$	-	$72.80_{\pm 0.23}$	-
Prototype-classifier [23]	✓	$51.57_{\pm 0.00}$	$63.43_{\pm 0.00}$	$20.97_{\pm 0.00}$	$30.10_{\pm 0.00}$	$73.50_{\pm 0.00}$	$81.75_{\pm 0.00}$	$37.60_{\pm 0.00}$	$44.95_{\pm 0.00}$
GDumb [50]		$45.29_{\pm 0.97}$	$66.86_{\pm 0.63}$	$20.95_{\pm 0.42}$	$45.40_{\pm 0.66}$	$69.72_{\pm 0.20}$	$80.95_{\pm 1.19}$	$28.24_{\pm 0.58}$	$43.64_{\pm 1.05}$
DER++ [4]		$61.47_{\pm 0.32}$	$77.15_{\pm 0.61}$	$50.64_{\pm 0.70}$	$67.64_{\pm 0.45}$	$63.64_{\pm 1.30}$	$79.55_{\pm 0.87}$	$53.11_{\pm 0.44}$	$65.10_{\pm 0.91}$
BiC [82]		$74.39_{\pm 1.12}$	$82.13_{\pm 0.33}$	$65.57_{\pm 0.93}$	$73.95_{\pm 0.29}$	$80.57_{\pm 0.86}$	$89.39_{\pm 0.33}$	$57.36_{\pm 2.68}$	$68.07_{\pm 0.22}$
EWC [26]	✓	$61.36_{\pm 1.43}$	$72.84_{\pm 2.18}$	$53.16_{\pm 1.45}$	$63.61_{\pm 1.06}$	$81.62_{\pm 0.34}$	$87.56_{\pm 0.97}$	$64.50_{\pm 0.36}$	$70.37_{\pm 0.41}$
LwF [33]	✓	$61.66_{\pm 1.95}$	$73.90_{\pm 1.91}$	$52.45_{\pm 0.48}$	$63.87_{\pm 0.31}$	$77.94_{\pm 1.00}$	$86.90_{\pm 0.90}$	$60.74_{\pm 0.30}$	$68.55_{\pm 0.65}$
Seq FT	✓	$61.67_{\pm 1.37}$	$73.25_{\pm 1.83}$	$52.91_{\pm 1.61}$	$63.32_{\pm 1.31}$	$81.47_{\pm 0.55}$	$87.55_{\pm 0.95}$	$64.43_{\pm 0.44}$	$70.48_{\pm 0.54}$
RanPAC [38]	✓	$74.43_{\pm 0.43}$	$83.63_{\pm 0.01}$	$63.21_{\pm 0.02}$	$74.01_{\pm 0.47}$	$86.47_{\pm 0.52}$	$90.81_{\pm 1.05}$	$69.11_{\pm 0.69}$	$75.20_{\pm 0.34}$
SLCA [85]	✓	$73.01_{\pm 0.16}$	$82.13_{\pm 0.34}$	$66.04_{\pm 0.08}$	$72.59_{\pm 0.04}$	$85.27_{\pm 0.08}$	$89.51_{\pm 1.04}$	$68.07_{\pm 0.21}$	$73.04_{\pm 0.56}$
CoMA (Ours)	✓	$75.12_{\pm 0.27}$	$82.76_{\pm 0.16}$	$67.48_{\pm 0.19}$	$74.90_{\pm 0.87}$	$86.59_{\pm 0.51}$	$91.02_{\pm 0.47}$	$69.33_{\pm 0.22}$	$75.64_{\pm 0.13}$
CoFiMA (Ours)	✓	$\mathbf{77.65}_{\pm 0.18}$	$\mathbf{83.54}_{\pm 0.16}$	$\mathbf{69.51}_{\pm 0.16}$	$\mathbf{76.21}_{\pm 0.83}$	$\mathbf{87.44}_{\pm 0.47}$	$\mathbf{91.13}_{\pm 0.53}$	$\mathbf{70.87}_{\pm 0.31}$	$\mathbf{76.09}_{\pm 0.78}$

CoMA, achieving new state-of-the-art results in CIL. This highlights the need for adaptive model averaging based on the importance of the parameters for a given task. In detail, CoFiMA achieves a Last-Acc of 87.11% and an Inc-Acc of 91.87% on CUB-200. This performance surpasses SLCA's performance, demonstrating a gain of +**2.4%** and +**0.93%** in Last-Acc and Inc-Acc, respectively. Similarly, on Cars-196, CoFiMA outperforms SLCA with +**9.23%** and +**5.72%** improvements in Last-Acc and Inc-Acc, respectively. This trend of outperforming SLCA is also observed in Imagenet-R, with +**1.25%** and +**0.31%** improvements in Last-Acc and Inc-Acc over SLCA, respectively.

From Tab. 2 we observe that CoMA and CoFiMA both surpass the state-of-the-art methods by substantial margins while using a PTM pre-trained unsupervisedly on ImageNet-1K. The results confirm that having access to an unsupervised PTM is sufficient to reach satisfactory performance in CIL, although the absolute performance is lower for all methods compared to Tab. 1. In detail, on CUB-200, CoFiMA's Last-Acc of 77.65% and Inc-Acc of 83.54% continue to show an advantage over SLCA. CoFiMA also leads in Cars-196, CIFAR-100, and ImageNet-R, showcasing its consistent performance across different datasets.

Notably, CoFiMA's performance is not far from the joint-training baselines in both Tabs. 1 and 2. For instance, in Tab. 1 for CIFAR-100, CoFiMA's Last-Acc is only **0.45%** lower than the joint-training baseline of 93.22%. This gap narrows further in other benchmarks, indicating CoFiMA's effectiveness in approaching the upper bounds of CL performance. CoFiMA, which efficiently balances the retention of old knowledge with the acquisition of new information, contributes to its strong performance using both supervised and unsupervised PTMs.

4.3 Ablation Studies

Analysis of Model Averaging. This section evaluates our continual model-averaging approach against two baselines:

Table 3: Experimental results comparing our methods (CoMA, and CoFiMA) to weight-averaging baselines using **ViT-B/16** [27] supervised PTM.

Variant	Init.	λ	CUB-200		Cars-196		CIFAR-100		Imagenet-R	
			Last-Acc	Inc-Acc	Last-Acc	Inc-Acc	Last-Acc	Inc-Acc	Last-Acc	Inc-Acc
Prototype-classifier [23]	-	-	$80.66_{\pm 0.00}$	$88.95_{\pm 0.00}$	$28.58_{\pm 0.00}$	$39.83_{\pm 0.00}$	$60.29_{\pm 0.00}$	$69.18_{\pm 0.00}$	$38.45_{\pm 0.00}$	$45.59_{\pm 0.00}$
Seq FT	-	-	$68.07_{\pm 1.09}$	$79.04_{\pm 1.69}$	$49.74_{\pm 1.25}$	$62.83_{\pm 2.16}$	$88.86_{\pm 0.83}$	$92.01_{\pm 1.71}$	$71.80_{\pm 1.45}$	$76.84_{\pm 1.26}$
SLCA [85]	-	-	$84.71_{\pm 0.40}$	$90.94_{\pm 0.68}$	$67.73_{\pm 0.85}$	$76.93_{\pm 1.21}$	$91.53_{\pm 0.28}$	$94.09_{\pm 0.87}$	$77.00_{\pm 0.33}$	$81.17_{\pm 0.64}$
Weight-Ens.	θ_0	$1/t$	$82.49_{\pm 0.27}$	$87.20_{\pm 0.56}$	$36.20_{\pm 0.27}$	$45.31_{\pm 0.31}$	$61.68_{\pm 0.14}$	$70.24_{\pm 0.46}$	$44.90_{\pm 0.14}$	$52.03_{\pm 0.46}$
Weight-Ens.	θ^*_{t-1}	$1/t$	$84.28_{\pm 0.47}$	$90.07_{\pm 0.22}$	$71.82_{\pm 0.47}$	$78.85_{\pm 0.31}$	$91.69_{\pm 0.23}$	$94.52_{\pm 0.95}$	$75.85_{\pm 0.73}$	$81.51_{\pm 0.60}$
EMA	-	-	$84.99_{\pm 0.49}$	$90.84_{\pm 0.74}$	$65.41_{\pm 0.18}$	$73.68_{\pm 0.46}$	$91.40_{\pm 0.12}$	$93.89_{\pm 0.59}$	$77.35_{\pm 0.83}$	$83.07_{\pm 0.94}$
CoMA (Ours)	θ^*_{t-1}	λ	$85.95_{\pm 0.29}$	$90.75_{\pm 0.39}$	$73.35_{\pm 0.59}$	$78.55_{\pm 0.42}$	$92.00_{\pm 0.13}$	$94.12_{\pm 0.63}$	$77.47_{\pm 0.05}$	$81.32_{\pm 0.17}$
CoFiMA (Ours)	θ^*_{t-1}	λF_t	**87.11**$_{\pm 0.56}$	**91.87**$_{\pm 0.69}$	**76.96**$_{\pm 0.64}$	**82.65**$_{\pm 0.96}$	**92.77**$_{\pm 0.24}$	**94.89**$_{\pm 0.94}$	**78.25**$_{\pm 0.26}$	**81.48**$_{\pm 0.56}$

- ***Weight-Ensemble***, which uniformly averages model weights (*e.g.* $\lambda = 1/t$), initializing each model M_t from either the pre-trained PTM ($\boldsymbol{\theta}_0$) or the previous task's parameters ($\boldsymbol{\theta}_{t-1}$).
- ***Exponential Moving Average (EMA)*** [66], a technique for a running average of model parameters computed at every gradient descent iteration m as follows: $\boldsymbol{\theta}_m = \beta \boldsymbol{\theta}_m + (1-\beta)\boldsymbol{\theta}_{m-1}$ with $\beta = 0.999$.

Tab. 3 presents the results. CoFiMA shows superior performance across all datasets. The *Weight-Ensemble* with θ_{t-1} initialization yields competitive results in CIFAR-100 (Last-Acc: 91.69%) but underperforms in datasets like Cars-196 (Last-Acc: 71.82%). The *Weight-Ensemble* starting from θ_0 shows lower performance. This indicates the limitations of reinitialization to θ_0; as models are trained on different tasks, it leads to different optima, as depicted in Fig. 2(a).

However, *Weight-Ensemble* with $\boldsymbol{\theta}_{t-1}$ initialization outperforms the variant starting from $\boldsymbol{\theta}_0$, emphasizing the importance of initialization. Initializing $\boldsymbol{\theta}_t$ from $\boldsymbol{\theta}^*_{t-1}$ improves performance on both task t and $t-1$, as done in [21,80]. This gain in performance indicates that averaging successive models leads to good performance on the current task t while preserving previous knowledge.

Averaging all models trained up to task t is suboptimal, as weights at task t likely differ significantly from those at task $t=1$. Such averaging shifts the $\boldsymbol{\theta}^*_t$ values toward suboptimal minima, leading to decreased performance (refer to Fig. 2(b)). In contrast, CoFiMA avoids this by averaging only with the preceding parameters $\boldsymbol{\theta}^*_{t-1}$, leading to better performance.

The *EMA* method, though superior to both *Weight-Ensemble* variants in most scenarios, falls short of CoFiMA's performance. This discrepancy may stem from excessive averaging in *EMA*, where $\boldsymbol{\theta}^*_t$ is modified multiple times within the same task, potentially leading to suboptimal outcomes. Differently, our method applies weight averaging only after completing each task, enhancing computational efficiency by selectively focusing on pertinent model weights while retaining previous knowledge.

Effect of PTMs. In this section, we evaluate the performance of the CoFiMA approach across a variety of backbone architectures, including self-supervised (MAE [17], MoCoV3 [5], and DINOv2 [45]) and supervised (ViT-Tiny [27] and ViT-B/16-SAM [13]) models. This comprehensive analysis aims to ascertain the adaptability and performance consistency of CoFiMA in diverse training paradigms. Results are visualized in Fig. 3.

Our results indicate that CoFiMA enhances performance across almost all tested backbones relative to the baseline SLCA method. For instance, with the ViT-Tiny backbone, CoFiMA improves the Last-Accuracy of SLCA from 80.25% to 82.96%. This trend is similarly observed with ViT-Large, where CoFiMA achieves an accuracy of 86.81%, surpassing SLCA's 85.93%. Only in the case of ViT-B/16-DINOv2, there is a slight reduction in performance, likely due to the benchmark reaching its saturation point: both SLCA and CoFiMA exhibit performance akin to the joint-learning gold standard.

In the context of self-supervised learning models, CoFiMA demonstrates its efficacy by outperforming SLCA on ViT-B/16-MAE and ViT-B/16-MoCoV3 backbones. Importantly, the ViT-B/16-SAM backbone [13] achieves the best performance among the evaluated models. This can be attributed to the effective generalization features ingrained in the SAM backbone, a result of being trained with the SAM optimizer. This optimizer is known for its capability to enhance model generalizability, which is reflected in the superior performance metrics observed in our experiments, as also mentioned in Mehta *et al*. [40] work. We also notice that self-supervised pre-training usually results in larger performance gaps between continual learning baselines and joint training. Especially for ViT-B/16-MAE, as noted in Zhang *et al*. [85] work, because joint training using MAE requires smaller updates to learn all tasks compared to incremental learning using SLCA or CoFiMA.

When compared to joint training, a standard upper bound in CL settings, CoFiMA demonstrates competitive performance. Although joint training always leads, as seen with ViT-Large (94.45%) and ViT-B/16-SAM (91.87%), CoFiMA remains close, particularly with ViT-B/16-SAM, where it achieves 90.48%. These results indicate that CoFiMA is a versatile approach, effective in enhancing performance with various backbones in both supervised and self-supervised learning contexts. However, the performance boost from our approach varies with the choice of backbone (size) and its pre-training paradigm.

Balancing the Information of an Old and a New Task. In CL, a primary objective is to balance knowledge retention from previous tasks with the acquisition of new information from current tasks. This study examines the impact of λ used to balance the stability/plasticity of the model during averaging.

In addition to our method, we also include an adaptation of the aggregation scheme of WiSE-FT [80] for CL, which we refer to as **WiSE-FT-CL**. After learning task t, stability is achieved via linear interpolation between the pre-trained and the fine-tuned model: $\boldsymbol{\theta}_t^* = \lambda\boldsymbol{\theta}_t + (1-\lambda)\boldsymbol{\theta}_0$. For every new task t, fine-tuning starts from $\boldsymbol{\theta}_{t-1}^*$. This method aligns with WiSE-FT [80] and relies only on the initial pre-training to achieve stability.

According to Fig. 4, CoFiMA demonstrates superior performance compared to WiSE-FT-CL across various λ settings. This improvement suggests that our method is more effective at incorporating new knowledge while preserving old information. The best performance is achieved with $\lambda = 0.3$, giving the best

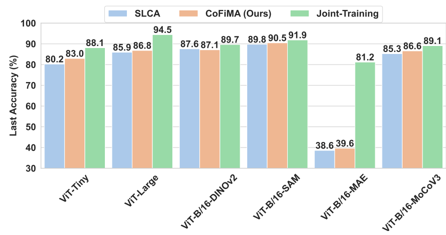

Fig. 3: CoFiMA with various PTMs on CIFAR-100. CoFiMA enhances the results of SLCA.

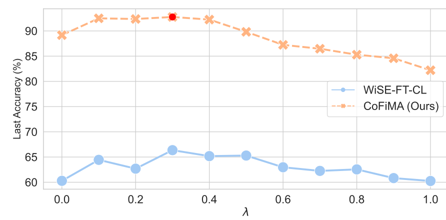

Fig. 4: Ablation study on the effect of λ on CIFAR-100 dataset. The red marker point represents the best performance.

trade-off between learning new task t (*i.e.*, plasticity) and preserving knowledge from previous tasks (*i.e.*, stability).

The underperformance of WiSE-FT-CL is attributed to the impact of averaging with the model $\boldsymbol{\theta}_0$, which possesses significantly different parameter values than $\boldsymbol{\theta}_t$ due to fine-tuning. Consequently, this averaging process results in sub-optimal parameter values as explained in Sec. 3. For CoFiMA, higher values for λ put more emphasis on the current model at task t, thus leading to forgetting previous tasks.

In summary, our method, CoFiMA, effectively maintains a balance between retaining old task information and adapting to new task data. This is achieved by leveraging weight averaging between successive models. The key benefit here is that the parameter values of models at tasks t and $t-1$ do not diverge significantly, ensuring that the averaged parameters remain effective for both tasks [6, 21] (more details in supplementary material).

5 Conclusion

In this work, we have introduced CoFiMA, the first method based on weight averaging techniques to address catastrophic forgetting in the CIL setting. This approach builds its grounding on two pillars. First, it leverages model averaging, providing a balanced mechanism for retaining prior knowledge while accommodating new information. Second, Fisher information is incorporated to intelligently weigh the averaging of parameters. This refinement allows for the adjustment of each parameter's value based on its importance, as determined by its Fisher information, thereby effectively reducing catastrophic forgetting.

Our benchmarking on diverse datasets with various PTM backbones demonstrates that CoFiMA consistently outperforms state-of-the-art CIL methods. Our findings underscore the efficacy of CoFiMA in mitigating forgetting and highlight its versatility across different PTM backbones and benchmark datasets.

Acknowledgements

We thank Yasser Benigmim and Ekaterina Iakovleva for helpful discussions and draft feedback. This paper has been supported by the French National Research Agency (ANR) in the framework of its JCJC, and was funded by the European Union's Horizon Europe research and innovation program under grant agreement No. 101120237 (ELIAS). This research was additionally funded by the EU project AI4TRUST (No.101070190). Furthermore, this research was partially funded by Hi!PARIS Center on Data Analytics and Artificial Intelligence. This work was granted access to the HPC resources of IDRIS under the allocation AD011013860 made by GENCI.

References

1. Abraham, W.C., Robins, A.: Memory retention–the synaptic stability versus plasticity dilemma. Trends in neurosciences **28**(2), 73–78 (2005) 2
2. Aljundi, R., Babiloni, F., Elhoseiny, M., Rohrbach, M., Tuytelaars, T.: Memory aware synapses: Learning what (not) to forget. In: Eur. Conf. Comput. Vis. (2018) 4
3. Amari, S.i.: Neural learning in structured parameter spaces - natural riemannian gradient. In: Mozer, M., Jordan, M., Petsche, T. (eds.) Advances in Neural Information Processing Systems. vol. 9. MIT Press (1996) 8
4. Buzzega, P., Boschini, M., Porrello, A., Abati, D., Calderara, S.: Dark experience for general continual learning: a strong, simple baseline. Adv. Neural Inform. Process. Syst. **33** (2020) 4, 10, 11, 9
5. Chen*, X., Xie*, S., He, K.: An empirical study of training self-supervised vision transformers. arXiv preprint arXiv:2104.02057 (2021) 10, 11, 12, 7
6. Chizat, L., Oyallon, E., Bach, F.: On lazy training in differentiable programming (2020) 9, 14
7. Choshen, L., Venezian, E., Slonim, N., Katz, Y.: Fusing finetuned models for better pretraining (2022) 5
8. Dhar, P., Singh, R.V., Peng, K.C., Wu, Z., Chellappa, R.: Learning without memorizing. In: IEEE Conf. Comput. Vis. Pattern Recog. (2019) 4
9. Dogucu, M., Johnson, A., Ott, M.: bayesrules: Datasets and Supplemental Functions from Bayes Rules! Book (2021), r package version 0.0.2.9000 3, 6, 8, 9
10. Dyer, E., Lewkowycz, A., Ramasesh, V.: Effect of scale on catastrophic forgetting in neural networks. In: Int. Conf. Learn. Represent. (2022) 2, 4
11. d'Ascoli, S., Touvron, H., Leavitt, M.L., Morcos, A.S., Biroli, G., Sagun, L.: Convit: Improving vision transformers with soft convolutional inductive biases. In: International Conference of Machine Learning (2021) 1
12. Entezari, R., Sedghi, H., Saukh, O., Neyshabur, B.: The role of permutation invariance in linear mode connectivity of neural networks. In: Int. Conf. Learn. Represent. (2022) 5
13. Foret, P., Kleiner, A., Mobahi, H., Neyshabur, B.: Sharpness-aware minimization for efficiently improving generalization. In: International Conference on Learning Representations (2021) 12, 13, 7
14. Frankle, J., Dziugaite, G.K., Roy, D., Carbin, M.: Linear mode connectivity and the lottery ticket hypothesis. In: Int. Conf. Machine Learn. (2020) 5, 7, 8

15. French, R.M.: Catastrophic forgetting in connectionist networks. Trends in cognitive sciences **3**(4) (1999) 1
16. Friedman, J., Hastie, T., Tibshirani, R., et al.: The elements of statistical learning. Springer series in statistics New York (2001) 4
17. He, K., Chen, X., Xie, S., Li, Y., Dollár, P., Girshick, R.: Masked autoencoders are scalable vision learners (2021) 12, 7
18. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: IEEE Conf. Comput. Vis. Pattern Recog. (2015) 1
19. Hendrycks, D., Basart, S., Mu, N., Kadavath, S., Wang, F., Dorundo, E., Desai, R., Zhu, T., Parajuli, S., Guo, M., et al.: The many faces of robustness: A critical analysis of out-of-distribution generalization. In: International Conference of Computer Vision (ICCV) (2021) 9
20. Hinton, G., Vinyals, O., Dean, J.: Distilling the knowledge in a neural network. arXiv preprint arXiv:1503.02531 (2015) 9
21. Ilharco, G., Wortsman, M., Gadre, S.Y., Song, S., Hajishirzi, H., Kornblith, S., Farhadi, A., Schmidt, L.: Patching open-vocabulary models by interpolating weights (2022) 12, 14
22. Izmailov, P., Podoprikhin, D., Garipov, T., Vetrov, D., Wilson, A.G.: Averaging weights leads to wider optima and better generalization. In: Conference on Uncertainty in Artificial Intelligence (UAI) (2018) 4, 5
23. Janson, P., Zhang, W., Aljundi, R., Elhoseiny, M.: A simple baseline that questions the use of pretrained-models in continual learning (2023) 10, 11, 12
24. Kairouz, P., McMahan, H.B., Avent, B., Bellet, A., Bennis, M., Bhagoji, A.N., Bonawitz, K., Charles, Z., Cormode, G., Cummings, R., D'Oliveira, R.G.L., Eichner, H., Rouayheb, S.E., Evans, D., Gardner, J., Garrett, Z., Gascón, A., Ghazi, B., Gibbons, P.B., Gruteser, M., Harchaoui, Z., He, C., He, L., Huo, Z., Hutchinson, B., Hsu, J., Jaggi, M., Javidi, T., Joshi, G., Khodak, M., Konečný, J., Korolova, A., Koushanfar, F., Koyejo, S., Lepoint, T., Liu, Y., Mittal, P., Mohri, M., Nock, R., Özgür, A., Pagh, R., Raykova, M., Qi, H., Ramage, D., Raskar, R., Song, D., Song, W., Stich, S.U., Sun, Z., Suresh, A.T., Tramèr, F., Vepakomma, P., Wang, J., Xiong, L., Xu, Z., Yang, Q., Yu, F.X., Yu, H., Zhao, S.: Advances and open problems in federated learning (2021) 5
25. Kim, D., Han, B.: On the stability-plasticity dilemma of class-incremental learning. In: IEEE Conf. Comput. Vis. Pattern Recog. (2023) 2
26. 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 **114**(13) (2017) 3, 4, 8, 10, 11
27. Kolesnikov, A., Dosovitskiy, A., Weissenborn, D., Heigold, G., Uszkoreit, J., Beyer, L., Minderer, M., Dehghani, M., Houlsby, N., Gelly, S., Unterthiner, T., Zhai, X.: An image is worth 16x16 words: Transformers for image recognition at scale. In: Int. Conf. Learn. Represent. (2021) 1, 10, 12, 7
28. Krause, J., Stark, M., Deng, J., Fei-Fei, L.: 3d object representations for fine-grained categorization. In: Proceedings of the IEEE International Conference on Computer Vision Workshops (2013) 9
29. Krause, J., Stark, M., Deng, J., Fei-Fei, L.: 3d object representations for fine-grained categorization. In: 2013 IEEE International Conference on Computer Vision Workshops (2013) 9
30. Krizhevsky, A., Hinton, G., et al.: Learning multiple layers of features from tiny images (2009) 9

31. Lakshminarayanan, B., Pritzel, A., Blundell, C.: Simple and scalable predictive uncertainty estimation using deep ensembles. In: Adv. Neural Inform. Process. Syst. (2017) 4
32. Li, T., Sahu, A.K., Talwalkar, A., Smith, V.: Federated learning: Challenges, methods, and future directions. IEEE Signal Processing Magazine **37**(3) (2020) 5
33. Li, Z., Hoiem, D.: Learning without forgetting. TPAMI **40**(12) (2017) 4, 10, 11, 9
34. Liao, Z., Drummond, T., Reid, I., Carneiro, G.: Approximate fisher information matrix to characterize the training of deep neural networks. IEEE Transactions on Pattern Analysis and Machine Intelligence **42**(1) (2020) 8
35. Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., Lin, S., Guo, B.: Swin transformer: Hierarchical vision transformer using shifted windows (2021) 1
36. Masana, M., Liu, X., Twardowski, B., Menta, M., Bagdanov, A.D., van de Weijer, J.: Class-incremental learning: survey and performance evaluation on image classification. IEEE Trans. Pattern Anal. Mach. Intell. (2022) 1
37. Matena, M.S., Raffel, C.A.: Merging models with fisher-weighted averaging. Adv. Neural Inform. Process. Syst. **35**, 17703–17716 (2022) 2, 3, 4, 5, 6, 7, 8
38. McDonnell, M.D., Gong, D., Parvaneh, A., Abbasnejad, E., van den Hengel, A.: Ranpac: Random projections and pre-trained models for continual learning. Adv. Neural Inform. Process. Syst. **36** (2024) 2, 4, 10, 11, 9
39. McMahan, H.B., Moore, E., Ramage, D., Hampson, S., y Arcas, B.A.: Communication-efficient learning of deep networks from decentralized data (2023) 5, 7
40. Mehta, S.V., Patil, D., Chandar, S., Strubell, E.: An empirical investigation of the role of pre-training in lifelong learning (2023) 2, 4, 13
41. Mirzadeh, S.I., Farajtabar, M., Gorur, D., Pascanu, R., Ghasemzadeh, H.: Linear mode connectivity in multitask and continual learning (2020) 3, 5, 7
42. Murata, K., Ito, S., Ohara, K.: Learning and transforming general representations to break down stability-plasticity dilemma. In: Proceedings of the Asian Conference on Computer Vision (2022) 2
43. Neyshabur, B., Sedghi, H., Zhang, C.: What is being transferred in transfer learning? In: Adv. Neural Inform. Process. Syst. (2020) 3, 5, 8
44. Nichol, A., Achiam, J., Schulman, J.: On first-order meta-learning algorithms. arXiv preprint arXiv:1803.02999 (2018) 4
45. Oquab, M., Darcet, T., Moutakanni, T., Vo, H.V., Szafraniec, M., Khalidov, V., Fernandez, P., Haziza, D., Massa, F., El-Nouby, A., Howes, R., Huang, P.Y., Xu, H., Sharma, V., Li, S.W., Galuba, W., Rabbat, M., Assran, M., Ballas, N., Synnaeve, G., Misra, I., Jegou, H., Mairal, J., Labatut, P., Joulin, A., Bojanowski, P.: Dinov2: Learning robust visual features without supervision (2023) 12, 7
46. Ovadia, Y., Fertig, E., Ren, J., Nado, Z., Sculley, D., Nowozin, S., Dillon, J.V., Lakshminarayanan, B., Snoek, J.: Can you trust your model's uncertainty? evaluating predictive uncertainty under dataset shift. In: Adv. Neural Inform. Process. Syst. (2019) 4
47. Panos, A., Kobe, Y., Reino, D.O., Aljundi, R., Turner, R.E.: First session adaptation: A strong replay-free baseline for class-incremental learning. In: Int. Conf. Comput. Vis. (2023) 2
48. Pascanu, R., Bengio, Y.: Revisiting natural gradient for deep networks (2014) 3, 8
49. Pham, Q., Liu, C., Steven, H.: Continual normalization: Rethinking batch normalization for online continual learning. In: Int. Conf. Learn. Represent. (2022) 4
50. Prabhu, A., Torr, P.H., Dokania, P.K.: Gdumb: A simple approach that questions our progress in continual learning. In: Eur. Conf. Comput. Vis. (2020) 4, 10, 11

51. Radford, A., Kim, J.W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., et al.: Learning transferable visual models from natural language supervision. In: Int. Conf. Machine Learn. PMLR (2021) 1
52. Ramasesh, V.V., Lewkowycz, A., Dyer, E.: Effect of scale on catastrophic forgetting in neural networks. In: Proceedings of the International Conference on Learning Representations (2021) 4
53. Ramé, A., Kirchmeyer, M., Rahier, T., Rakotomamonjy, A., Gallinari, P., Cord, M.: Diverse weight averaging for out-of-distribution generalization (2023) 7
54. Ren, W., Li, X., Wang, L., Zhao, T., Qin, W.: Analyzing and reducing catastrophic forgetting in parameter efficient tuning (2024) 5
55. Ridnik, T., Ben-Baruch, E., Noy, A., Zelnik-Manor, L.: Imagenet-21k pretraining for the masses (2021) 1, 10
56. Rusu, A.A., Rabinowitz, N.C., Desjardins, G., Soyer, H., Kirkpatrick, J., Kavukcuoglu, K., Pascanu, R., Hadsell, R.: Progressive neural networks. arXiv preprint arXiv:1606.04671 (2016) 4
57. Rypeść, G., Cygert, S., Khan, V., Trzcinski, T., Zieliński, B.M., Twardowski, B.: Divide and not forget: Ensemble of selectively trained experts in continual learning. In: Int. Conf. Learn. Represent. (2023) 4
58. Schuhmann, C., Vencu, R., Beaumont, R., Kaczmarczyk, R., Mullis, C., Katta, A., Coombes, T., Jitsev, J., Komatsuzaki, A.: Laion-400m: Open dataset of clip-filtered 400 million image-text pairs. arXiv preprint arXiv:2111.02114 (2021) 1
59. Serra, J., Suris, D., Miron, M., Karatzoglou, A.: Overcoming catastrophic forgetting with hard attention to the task. In: Int. Conf. Machine Learn. (2018) 4
60. Simon, C., Faraki, M., Tsai, Y.H., Yu, X., Schulter, S., Suh, Y., Harandi, M., Chandraker, M.: On generalizing beyond domains in cross-domain continual learning. arXiv preprint arXiv:2203.03970 (2022) 4
61. Soen, A., Sun, K.: On the variance of the fisher information for deep learning. Adv. Neural Inform. Process. Syst. **34**, 5708–5719 (2021) 8
62. Spall, J.C.: Monte carlo computation of the fisher information matrix in nonstandard settings. Journal of Computational and Graphical Statistics **14**(4) (2005) 3, 6, 8
63. Spall, J.C.: Improved methods for monte carlo estimation of the fisher information matrix. In: 2008 American Control Conference (2008) 8
64. Stickland, A.C., Murray, I.: Diverse ensembles improve calibration. In: International Conference on Machine Learning (ICML) Workshop on Uncertainty and Robustness in Deep Learning (2020) 4
65. Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., Wojna, Z.: Rethinking the inception architecture for computer vision. In: IEEE Conf. Comput. Vis. Pattern Recog. (2016) 4
66. Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., Wojna, Z.: Rethinking the inception architecture for computer vision. In: IEEE Conf. Comput. Vis. Pattern Recog. (2016) 12, 4
67. Titsias, M.K., Schwarz, J., de G. Matthews, A.G., Pascanu, R., Teh, Y.W.: Functional regularisation for continual learning with gaussian processes (2020) 5
68. van de Ven, G.M., Tolias, A.S.: Three scenarios for continual learning (2019) 5
69. Villa, A., Alcázar, J.L., Alfarra, M., Alhamoud, K., Hurtado, J., Heilbron, F.C., Soto, A., Ghanem, B.: Pivot: Prompting for video continual learning. arXiv preprint arXiv:2212.04842 (2022) 2
70. Wah, C., Branson, S., Welinder, P., Perona, P., Belongie, S.: The Caltech-UCSD Birds-200-2011 Dataset. Tech. Rep. CNS-TR-2011-001, California Institute of Technology (2011) 9

71. Wah, C., Branson, S., Welinder, P., et al.: The caltech-ucsd birds-200-2011 dataset (2011) 9
72. Wang, L., Yang, K., Li, C., Hong, L., Li, Z., Zhu, J.: Ordisco: Effective and efficient usage of incremental unlabeled data for semi-supervised continual learning. In: IEEE Conf. Comput. Vis. Pattern Recog. (2021) 4
73. Wang, L., Zhang, X., Li, Q., Zhang, M., Su, H., Zhu, J., Zhong, Y.: Incorporating neuro-inspired adaptability for continual learning in artificial intelligence. Nature Machine Intelligence **5**(12) (Nov 2023) 4
74. Wang, L., Zhang, X., Li, Q., Zhu, J., Zhong, Y.: Coscl: Cooperation of small continual learners is stronger than a big one (2022) 4
75. Wang, L., Zhang, X., Su, H., Zhu, J.: A comprehensive survey of continual learning: Theory, method and application (2023) 1, 5
76. Wang, Y., Huang, Z., Hong, X.: S-prompts learning with pre-trained transformers: An occam's razor for domain incremental learning. arXiv preprint arXiv:2207.12819 (2022) 2
77. Wang, Z., Zhang, Z., Ebrahimi, S., Sun, R., Zhang, H., Lee, C.Y., Ren, X., Su, G., Perot, V., Dy, J., et al.: Dualprompt: Complementary prompting for rehearsal-free continual learning. In: Eur. Conf. Comput. Vis. Springer (2022) 2, 4, 10, 5, 9
78. Wang, Z., Zhang, Z., Lee, C.Y., Zhang, H., Sun, R., Ren, X., Su, G., Perot, V., Dy, J., Pfister, T.: Learning to prompt for continual learning. In: IEEE Conf. Comput. Vis. Pattern Recog. (2022) 2, 4, 10, 5, 9
79. 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: Int. Conf. Machine Learn. PMLR (2022) 2, 3, 4, 5
80. Wortsman, M., Ilharco, G., Li, M., Kim, J.W., Hajishirzi, H., Farhadi, A., Namkoong, H., Schmidt, L.: Robust fine-tuning of zero-shot models. In: IEEE Conf. Comput. Vis. Pattern Recog. (2022) 2, 3, 4, 5, 6, 12, 13
81. Wu, T.Y., Swaminathan, G., Li, Z., Ravichandran, A., Vasconcelos, N., Bhotika, R., Soatto, S.: Class-incremental learning with strong pre-trained models. In: IEEE Conf. Comput. Vis. Pattern Recog. (2022) 2, 4
82. Wu, Y., Chen, Y., Wang, L., Ye, Y., Liu, Z., Guo, Y., Fu, Y.: Large scale incremental learning. In: IEEE Conf. Comput. Vis. Pattern Recog. (2019) 4, 10, 11
83. Yang, B., Deng, X., Shi, H., Li, C., Zhang, G., Xu, H., Zhao, S., Lin, L., Liang, X.: Continual object detection via prototypical task correlation guided gating mechanism. In: IEEE Conf. Comput. Vis. Pattern Recog. (2022) 4
84. Zenke, F., Poole, B., Ganguli, S.: Continual learning through synaptic intelligence. In: Int. Conf. Machine Learn. (2017) 4
85. 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: Int. Conf. Comput. Vis. (2023) 2, 4, 9, 10, 11, 12, 13, 1, 5
86. Zhang, J., Zhang, J., Ghosh, S., Li, D., Zhu, J., Zhang, H., Wang, Y.: Regularize, expand and compress: Nonexpansive continual learning. In: WACV (2020) 4
87. Zhou, D.W., Ye, H.J., Zhan, D.C., Liu, Z.: Revisiting class-incremental learning with pre-trained models: Generalizability and adaptivity are all you need (2023) 2