Toward a Holistic Approach to Continual Model Merging

Hoang Phan
New York University
hvp2011@nyu.edu

Sungmin Cha
New York University
sungmin.cha@nyu.edu

Lam Tran
VinAI Research
lamtt12@vinai.io

Qi Lei
New York University
ql518@nyu.edu

Abstract

We present a holistic framework for continual model merging that intervenes at three critical stages: pre-merging, during merging, and post-merging—to address two fundamental challenges in continual learning. In particular, conventional approaches either maintain a growing list of per-domain task vectors, leading to scalability issues or rely solely on weight-space merging when old data is inaccessible, thereby losing crucial functional information. Our method overcomes these limitations by first fine-tuning the main model within its tangent space on domain-specific data; this linearization amplifies per-task weight disentanglement, effectively mitigating across-task interference. During merging, we leverage functional information from available optimizer states beyond mere parameter averages to avoid the need to revisit old data. Finally, a post-merging correction aligns the representation discrepancy between pre- and post-merged models, reducing bias and enhancing overall performance—all while operating under constant memory constraints without accessing historical data. Extensive experiments on standard class-incremental and domain-incremental benchmarks demonstrate that our approach not only achieves competitive performance but also provides a scalable and efficient solution to the catastrophic forgetting problem. Our code is available at https://github.com/VietHoang1512/CMM.

1. Introduction

The rapid proliferation of large-scale pre-trained models has fundamentally reshaped artificial intelligence research. Models, such as GPT-4 [1, 2], Llama [3, 4], Deepseek [5–7] and Stable Diffusion [8] or Sora [9] exemplify the transformative power of training extensive neural networks on diverse datasets, offering robust performance across numerous tasks and domains. Platforms like HuggingFace[1] currently host approximately 1.5 million publicly available models, each fine-tuned or pretrained with unique data sources, architectures, and training methodologies. This growing repository presents both an opportunity and a challenge: while specialized models can excel in their respective domains, their fragmented nature raises questions about how best to leverage them for broader applications.

Model merging has emerged as a promising direction to address this challenge [10, 11]. Model merging involves combining the parameters of multiple pre-trained models into a single unified model, aiming to retain or even enhance the individual strengths of each constituent model. Unlike traditional ensemble methods that aggregate model outputs at inference, parameter merging directly integrates learned representations within a single model, thereby reducing inference cost and storage requirements, while potentially boosting generalization and robustness.

Beyond efficiency gains, model merging provides versatile methodological tools applicable to several critical areas of machine learning, including multi-task learning [12–14], federated learning [15–17], and particularly continual learning [18–20]. In multi-task learning, merging specialized models trained on distinct tasks can yield a comprehensive model capable of simultaneous task proficiency. In federated learning scenarios, model merging facilitates privacy-preserving aggregation of locally trained models without exchanging sensitive training data. Most notably, in continual learning, merging allows models to sequentially incorporate new knowledge without catastrophic forgetting, significantly advancing the stability and scalability of lifelong learning systems.

Despite growing interest and wide application, existing merging techniques predominantly focus on the "during-merging" phase, where practitioners combine multiple models without access to their original training data [21, 22]. This limitation can lead to suboptimal performance, as naive merging approaches often overlook critical functional information embedded in the training process. Additionally, the number of stored models can grow linearly with the number of tasks, posing scalability challenges. To overcome these limitations, we propose a holistic framework that systematically addresses model merging across three key phases, pre-merging, during-merging, and post-merging. By

[1] https://huggingface.co/models

leveraging task-specific data, our method exploits valuable functional information, leading to improved merging performance compared to conventional linear interpolation methods. Figure 1 highlights this improvement, illustrating accuracy and loss curves when interpolating between models trained on the first two tasks of the Cars [23] dataset, where our proposed method (pink) consistently outperforms linear interpolation (blue) across all settings.

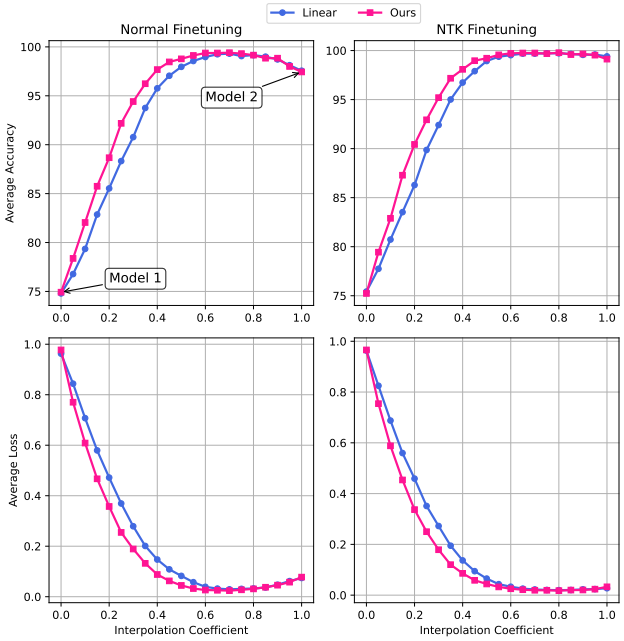

Figure 1. Average accuracy score and loss value when interpolating models after training on first and second tasks, using linear averaging and our proposed merging method.

Based on this finding, this paper aims to address key limitations in existing merging-based continual learning approaches. Specifically, we propose to first fine-tune task-specific models within the tangent space of a pre-trained model, effectively mitigating conflicts during subsequent model merging. Next, we derive a closed-form merging solution that naturally leverages expressive functional information without additional computational cost. Furthermore, we identify and analyze the representation mismatch between merged models and previously learned representations, motivating a representation refinement step as a beneficial post-merging procedure. In summary, our key contributions are:

- We shed light on the fundamental pitfalls encountered when naively applying existing model merging techniques to continual learning scenarios, particularly emphasizing issues such as limited scalability and neglect of available task-specific functional information.
- Based on these insights, we propose a unified merging-based framework specifically designed to alleviate catastrophic forgetting not only during the merging phase but also before and after merging.
- We conduct extensive experiments across several popular benchmarks in both class-incremental and domain-incremental learning settings, empirically validating the effectiveness and versatility of our proposed approach.

2. Related work

2.1. Continual learning

Continual learning (CL) aims to develop models capable of adapting sequentially to new tasks while maintaining performance on previously learned tasks. The main challenge in CL is catastrophic forgetting, where acquiring new information interferes with existing knowledge [24, 25]. Existing approaches to mitigate forgetting can broadly be categorized into three groups: regularization-based, rehearsal-based, and architecture-based methods. Regularization-based approaches constrain updates to parameters crucial for previous tasks, thus reducing interference [24, 26]. Architectural methods expand or adapt the model architecture dynamically, assigning distinct parameter subsets to individual tasks to minimize conflict [27, 28]. Rehearsal-based methods retain exemplars from previous tasks in a memory buffer, leveraging them during training of subsequent tasks to prevent forgetting [25, 29]. However, rehearsal-based methods raise concerns about data privacy and storage requirements. Given these limitations, recent efforts have increasingly focused on memory-free continual learning, utilizing techniques such as generative replay [30], prompt-based adaptation [31, 32], and leveraging pre-trained models to achieve robust performance without data storage [33, 34].

2.2. Model merging

Model merging aims to combine multiple pre-trained or task-specific models into a unified model that integrates their strengths. Early merging methods typically average model parameters directly, based on the mode connectivity assumption that different parameter solutions lie in similar regions of the loss landscape [35, 36]. Subsequent approaches introduced more sophisticated alignment-based strategies, such as weight matching and permutation-invariant transformations, to better handle parameter disparities between merged models [37–39]. Recent techniques, including Task Arithmetic [36], TIES-Merging [40], and Fisher merging [37, 41], further refine merging by considering task-specific importance or parameter influence, significantly improving generalization. Despite these advances, most existing merging methods assume simultaneous access to multiple models, which restricts their direct applicability to continual learning scenarios. For instance, MagMax [42] retains one task

vector per task, resulting in linear memory growth with respect to the number of tasks. It aggregates these vectors by selecting, for each parameter, the value with the largest absolute magnitude across tasks. Since MagMax performs arithmetic operations purely in the parameter space, this method neglects valuable information encoded in task-specific data. The most closely related to our work is CoFiMA [43], which extends Fisher Merging to continual learning by applying an online variant to merge models sequentially. In contrast, our method directly approximates the Fisher Information Matrix by leveraging the second-moment statistics readily available from standard deep-learning optimizers such as Adam [44] or AdamW [45]. Furthermore, our method mitigates task interference during the merging process and incorporates feature refinement post-merging, enabling effective sequential model merging while preserving representational consistency across tasks.

3. Background

3.1. Continual learning

Continual learning for large pre-trained models is crucial for enabling efficient knowledge integration across multiple tasks. In this regard, we focus on continual fine-tuning, where a pre-trained model θ is given the goal is to incrementally learn knowledge from N different domains or tasks, denoted as $\{\mathcal{D}_1, \mathcal{D}_2, \ldots, \mathcal{D}_N\}$. Each domain or task's dataset $\mathcal{D}_n = \{\boldsymbol{X}_n, \boldsymbol{Y}_n\}$ is accessible only during the n-th fine-tuning phase. We investigate two key challenging continual learning scenarios: Class-Incremental Learning (CIL) and Domain-Incremental Learning (DIL). In CIL, the class labels $y_n^j \in \mathcal{Y}_n$ from the incoming domain $\mathcal{D}_n$ are disjoint from the set of seen class labels $\mathcal{Y}_{1:n-1}$. In contrast, DIL assumes that all domains share the same label space $\mathcal{Y}$. For both scenarios, during inference at any task, models must classify input images into all seen classes so far without knowing their domain identity information.

In this paper, we primarily focus on *data-free continual learning*, where data from prior domains is inaccessible. In this setting, the only information retained from past tasks during training on t-th domain is the optimally fine-tuned model parameters from the previous task θ_{t-1}^*.

3.2. CLIP and nearest mean classifier

CLIP [46] is a dual-encoder model that learns a joint embedding space for images and text. It consists of an image encoder $f(\cdot)$ and a text encoder $g(\cdot)$, mapping an image I and text prompt T into normalized embeddings $z_I = f(I)$ and $z_T = g(T)$, respectively. Image classification using CLIP can be performed by computing the cosine similarity between an image embedding and class text embeddings: $s_k = \frac{z_I \cdot z_{T_k}}{\|z_I\| \|z_{T_k}\|}$. The similarity scores are transformed into probabilities using a softmax function with temperature τ:

$$p(y = k \mid I) = \frac{\exp(s_k/\tau)}{\sum_{j=1}^{K} \exp(s_j/\tau)}$$

This formulation allows CLIP to perform zero-shot classification by leveraging the learned joint embedding space without the need for task-specific training. In all our experiments, the encoder $g(\cdot)$ is frozen and we use the nearest mean classifier to classify input images using mean-class embeddings.

3.3. Adam and Fisher Information Matrix

Adam [44] is an adaptive gradient-based optimizer that utilizes first- and second-moment estimates for efficient optimization. AdamW [47] further improves it by decoupling weight decay from gradient updates. The full update rule with decay parameter λ is given by:

$$\begin{aligned}
&\boldsymbol{g}_t \leftarrow \nabla f_t\left(\boldsymbol{\theta}_{t-1}\right) + \lambda \boldsymbol{\theta}_{t-1} \\
&\boldsymbol{m}_t \leftarrow \beta_1 \boldsymbol{m}_{t-1} + \left(1-\beta_1\right) \boldsymbol{g}_t \quad \text{// first-moment estimate} \\
&\boldsymbol{v}_t \leftarrow \beta_2 \boldsymbol{v}_{t-1} + \left(1-\beta_2\right) \boldsymbol{g}_t^2 \quad \text{// second-moment estimate} \\
&\widehat{\boldsymbol{m}_t} \leftarrow \boldsymbol{m}_t / \left(1-\beta_1^t\right) \\
&\widehat{\boldsymbol{v}_t} \leftarrow \boldsymbol{v}_t / \left(1-\beta_2^t\right) \\
&\theta_t \leftarrow \theta_t - \gamma \widehat{\boldsymbol{m}_t} / \left(\sqrt{\widehat{\boldsymbol{v}_t}} + \epsilon\right)
\end{aligned} \tag{1}$$

where t denotes the time step, $\boldsymbol{g}_t \in \mathbb{R}^n$ the gradient vector and $\eta > 0$ the step size. The exponential moving average (EMA) time scales of the first gradient moment $\boldsymbol{m}_t$ and second moment $\boldsymbol{v}_t$ are set by $0 \leqslant \beta_1, \beta_2 < 1$. Note that, when $\beta_1 = 0$, $\boldsymbol{m}_t$ corresponds to the diagonal Fisher Information matrix at the t-th iteration, which is widely used for preconditioning in various optimizers [48–50]

3.4. Neural tangent kernel

In the vicinity of the initial weights θ_0, a neural network's behavior can be closely approximated by the following first-order Taylor expansion:

$$f(x;\theta) \approx f(x;\theta_0) + (\theta - \theta_0)^\top \nabla_\theta f(x;\theta_0). \tag{2}$$

This approximation corresponds to a kernel-based predictor using the *Neural Tangent Kernel* (NTK), represented as $k_{\text{NTK}}(x, x') = \nabla_\theta f(x;\theta_0)^\top \nabla_\theta f(x';\theta_0)$. It establishes a neural tangent space wherein the connection between weights and functional outputs is linear. Note that, as network width approaches infinity, this linearization remains valid throughout training [51–53].

Nonetheless, this linear model often does not hold at finite network widths, as it fails to adequately represent the evolution of parameters during training, according to Equation (2). In these instances, the training dynamics

operate in a *non-linear regime*. Conversely, in scenarios such as fine-tuning, where the evolution of parameters in many pre-trained models is relatively minor, the training largely remains within the tangent space, thus the linear approximation in Equation (2) effectively mirrors the network's behavior [54–58], indicative of a *linear regime*.

4. Proposed method

To solve the above problems, we propose a new algorithm. In this section, we introduce a novel algorithm to address the challenges associated with model merging in continual learning. Our framework improves upon traditional methods by incorporating second-order information, enhancing task-specific representation alignment, and reducing task interference during the merging process.

4.1. Merging during task progression

Let $\mathcal{D}_i$ denote a sequence of tasks indexed by $i = 1, ..., N$. For each task i, the model with parameters θ_i is fine-tuned until convergence. The loss function for i-th task is defined as:

$$L_i(\theta) \;=\; \mathbb{E}_{(x,y)\sim\mathcal{D}_i}\Big[\,\ell\big(f_\theta(\boldsymbol{X}),\,\boldsymbol{Y}\big)\Big],$$

where l represents the loss between the model prediction $f_\theta(\boldsymbol{X})$ and the true label $\boldsymbol{Y}$. Since θ_i is fine-tuned on task i, it implies that θ_i is (approximately) a local minimum of $L_i(\theta)$. A standard second-order Taylor expansion of $L_i(\theta^*)$ with θ^* around θ_i is given by:

$$\begin{aligned} L_i(\theta^*) \;\approx\; & L_i(\theta_i) + \nabla_\theta L_i(\theta_i)^\top \big(\theta^* - \theta_i\big) \\ & + \tfrac{1}{2}\,\big(\theta^* - \theta_i\big)^\top \nabla_\theta^2 L_i(\theta_i)\,\big(\theta^* - \theta_i\big). \end{aligned}$$

Since θ_i is a local optimum for the i-th task, we have $\nabla_\theta L_i(\theta_i) \;\approx\; 0$, which eliminates the first-order term. Therefore, the expansion simplifies to:

$$L_i(\theta^*) \approx L_i(\theta_i) + \tfrac{1}{2}\,\big(\theta^* - \theta_i\big)^\top \nabla_\theta^2 L_i(\theta_i)\,\big(\theta^* - \theta_i\big). \tag{3}$$

In the context of model merging, where task-specific data is unavailable, we approximate the Hessian matrix $\nabla_\theta^2 L_i(\theta_i)$ using the identity matrix I, yielding the approximation:

$$L_i(\theta^*) \;\approx\; L_i(\theta_i) \;+\; \tfrac{1}{2}\,(\theta^* - \theta_i)^\top I\,(\theta^* - \theta_i).$$

Thus, the average loss across tasks is computed as:

$$\mathcal{L}(\theta^*) = \frac{1}{N}\sum_{i=1}^{N} L_i(\theta_i) \;+\; \frac{1}{2N}\sum_{i=1}^{N} \|\theta^* - \theta_i\|^2.$$

Minimizing the above objective leads to the closed-form solution: $\theta^* \;=\; \frac{1}{N}\sum_{i=1}^{N}\theta_i$, which corresponds to a simple weight averaging approach (*i.e.* weight averaging).

However, this approach neglects crucial functional information during the merging process, relying solely on weight space. To address this limitation, we propose leveraging second-order information from the optimizer to more accurately approximate the Hessian matrix in Equation 3. Without loss of generality, we assume that $N = 2$ (representing the previous and current tasks). We thus minimize the weighted sum of per-task objectives $(1-\lambda)L_1(\theta^*) + \lambda L_2(\theta^*)$ for some $\lambda \in (0, 1)$. Using the above quadratic expansions of $L_1(\theta^*)$ and $L_2(\theta^*)$ and ignoring constant terms, the objective becomes:

$$(1-\lambda)\,(\theta^* - \theta_1)^\top F_1\,(\theta^* - \theta_1) + \lambda\,(\theta^* - \theta_2)^\top F_2\,(\theta^* - \theta_2)\,,$$

where $F_i \;\approx\; \nabla_\theta^2 L_i\,(\theta_i)$ is approximated by the Fisher Information Matrix (FIM).

Taking the gradient of the above objective with respect to θ and setting it to zero yields the optimal solution:

$$\theta^* = \left((1-\lambda)F_1 + \lambda F_2\right)^{-1}\left[(1-\lambda)F_1\theta_1 + \lambda F_2\theta_2\right] \tag{4}$$

While directly computing the FIM over the entire training set is computationally expensive, we can approximate it using second-moment information from commonly used optimizers, such as Adam [44], which tracks the moving average of squared gradients $\mathbf{v}_t$ (Equation 1) as an approximation to the diagonal of the Fisher information matrix [59]. Note that this approximation is efficient, requiring no gradient computation or additional data, and thus significantly reduces the computational cost compared to traditional FIM-based methods.

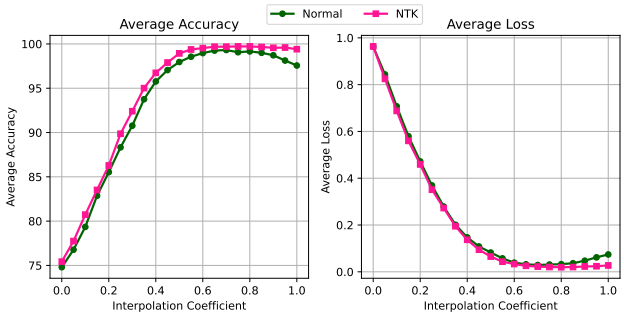

Figure 2. Average accuracy score and loss value along the linear path between models after training on first and second tasks using normal training and our proposed linear fine-tuning method.

4.2. Pre-merging and post-merging strategies

To further improve the merging process, we introduce a linear fine-tuning method aimed at better disentangling the task-specific weights and mitigating task interference [60]. To be more specific, Ortiz-Jimenez et al reveal that weight disentanglement is crucial for model merging, particularly to avoid negative transfer when adding linear combinations

of task vectors. Motivated by this, we propose to fine-tune each model on the tangent space of the pre-trained model θ_0 to effectively facilitate model merging in subsequent stages. As a simple illustrative example, Figure 2 shows the effectiveness of neural tangent kernel (NTK) compared with normal fine-tuning along the linear interpolation between different models.

Additionally, our method involves a representation alignment step post-merging to address feature representation mismatch. Figure 3 shows the mismatch between feature representations before and after merging, which is mitigated by applying refinement after the merging process. By aligning the feature representations, we ensure that the model's output is consistent across tasks, minimizing any detrimental effects caused by the merging process.

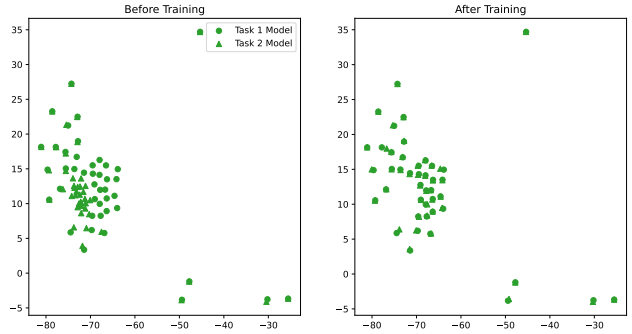

Figure 3. Mismatch between the feature representations of models before merging and after merging (left) or between models before merging and after merging + refinement.

Our proposed approach can be summarized in Algorithm 1, which details the three-stage process of holistic continual model merging. The stages include linear fine-tuning, model merging, and representation alignment, each of which is designed to address specific limitations of conventional methods and improve task transfer in continual learning.

5. Experimental results

To demonstrate the effectiveness of our proposed method, we conduct extensive experiments to compare our approach with baselines on both domain-incremental and class-incremental rehearsal-free scenarios. We implemented our codebase and reproduced relevant results based on implementations of [42, 61].

5.1. Evaluation Benchmarks

Datasets. We conduct comprehensive experiments on several well-established benchmarks commonly used in continual learning research. Specifically, we use CIFAR-100 [63] and ImageNet-R [64] as general image recognition benchmarks and CUB-200 [65] and Cars [23] as fine-grained classification datasets. The CIFAR-100 dataset is split into 10/20/50 tasks, with 10/5/2 distinct classes per task, respectively. ImageNet-R, comprising 200 classes, is split into 5, 10, and 20 disjoint tasks to thoroughly evaluate performance across varying degrees of task granularity. CUB-200 and Cars datasets, representing specialized fine-grained classification scenarios, are evenly partitioned into 5/10/20 tasks, each containing an equal number of classes. Additionally, we evaluate our method on domain-incremental benchmarks: Office-31 [66], Office Home [67],

Algorithm 1 Holistic Continual Model Merging

Input: Pre-trained model θ_0, sequence of tasks $\{D_t\}_{t=1}^{N}$, merging weight λ

Initialize merged model: $\theta_0^* \leftarrow \theta_0$

for $t = 1, \ldots, N$ **do**

`// Step 1: Linear Fine-Tuning`

Update task vector τ_t using data D_t by solving:

$$\tau_t \leftarrow \arg\min_{\tau} \mathcal{L}_t\Big(f_{\text{lin}}(\boldsymbol{x}; \theta_0 + \tau), D_t\Big)$$

where

$$f_{\text{lin}}(\boldsymbol{x}; \theta_0 + \tau) = f(\boldsymbol{x}; \theta_0) + \tau^\top \nabla_\theta f(\boldsymbol{x}; \theta_0)$$

and update:

$$\theta_t \leftarrow \theta_0 + \tau_t.$$

Estimate the diagonal Fisher matrix by the second moment term $\mathbf{v}_t$ from the AdamW optimizer

if $t > 1$ **then**

`// Step 2: Model Merging`

Merge the models as:

$$\theta_t^* \leftarrow \frac{\lambda\, \mathbf{v}_t \odot \theta_t + (1-\lambda)\, \mathbf{v}_{t-1} \odot \theta_{t-1}^*}{\lambda\, \mathbf{v}_t + (1-\lambda)\, \mathbf{v}_{t-1}},$$

where $\odot$ denotes element-wise multiplication.

`// Step 3: Representation Alignment`

Compute feature representations on the entire dataset $\mathcal{D}_t$:

$$\mathcal{Z}_t^t \leftarrow \text{FeatureExtract}(\theta_t^*, \mathcal{D}_t),$$

$$\mathcal{Z}_t^{t-1} \leftarrow \text{FeatureExtract}(\theta_{t-1}^*, \mathcal{D}_t).$$

Calculate the representation bias:

$$d_t = \frac{1}{|\mathcal{D}_t|} \left\| \mathcal{Z}_t^t - \mathcal{Z}_t^{t-1} \right\|.$$

Optimize θ_t^* (e.g. updating the last layer) to minimize d_t.

Set $\mathbf{v}_t \leftarrow \lambda\, \mathbf{v}_t + (1-\lambda)\, \mathbf{v}_{t-1}$ for the next iteration.

end

Output: Final merged model θ_N^*

Table 1. Comparison of continual learning methods across class-incremental setup. We report classification accuracy (%) after the final task. The best results are in **bold** and the second best underlined. $\Theta(1)$ and $\Theta(N)$ indicate constant and linear memory usage, irrespective of the number of tasks N.

Memory	Method	CIFAR100 /10	CIFAR100 /20	CIFAR100 /50	ImageNet-R /5	ImageNet-R /10	ImageNet-R /20	CUB200 /5	CUB200 /10	CUB200 /20	Cars /5	Cars /10	Cars /20	Avg
$\Theta(1)$	Zero-shot		66.91			77.73			56.08			64.71		67.21
N/A	Joint		90.94			87.55			81.57			88.21		87.38
$\Theta(N)$	RandMix [42]	77.04	75.36	72.91	83.10	81.88	80.18	59.86	58.53	58.08	67.32	65.62	64.95	70.40
$\Theta(N)$	MaxAbs [42]	76.75	74.39	73.04	83.03	82.33	80.92	60.15	58.01	56.59	67.36	63.55	58.95	69.59
$\Theta(N)$	Avg [35]	77.04	75.29	72.92	83.08	81.87	80.27	59.77	58.44	58.01	67.37	65.59	64.88	70.38
$\Theta(N)$	TIES [62]	77.23	74.66	73.76	83.08	82.27	80.83	60.94	58.22	56.97	70.45	64.90	61.17	70.37
$\Theta(1)$	LwF [26]	73.45	72.05	68.84	81.15	**82.97**	81.82	**65.12**	60.67	**58.90**	71.72	**69.84**	62.98	70.79
$\Theta(1)$	EWC [24]	76.24	75.39	72.97	82.15	82.42	81.48	59.10	54.49	53.31	69.46	60.78	57.42	68.77
$\Theta(1)$	Ours	**77.72**	**75.95**	**73.88**	**83.41**	82.69	**82.00**	64.67	**61.13**	58.70	**73.52**	69.70	**67.18**	**72.54**

DomainNet [68] and ImageNet-R [64].

Baselines. We compare our proposed method against various established continual learning approaches, including general methods such as Learning without Forgetting (LwF) [26] and Elastic Weight Consolidation (EWC) [24], recent model merging techniques like Model Soup (Avg) [35], RandMix [42], and TIES-Merging (TIES) [40], as well as prompt-based continual learning methods such as L2P [31], DualPrompt [32], AttriCLIP [69], and CODA-Prompt [34]. Additionally, we evaluate our approach against a baseline model that performs linear interpolation and examine performance against the joint-training upper bound. This comprehensive set of baselines ensures a robust and fair evaluation of our merging strategy within diverse continual learning scenarios.

Implementation details. We utilize a ViT/B16 [70] image encoder from CLIP [46] as our base model. The fine-tuning procedure follows the training protocol from previous work [36, 71], using a batch size of 128, an initial learning rate of $1e-5$, and the AdamW [45] optimizer with weight decay of 0.1 and exponential decay rates $\beta_1 = 0.9$, $\beta_2 = 0.999$. A cosine annealing learning rate schedule is applied throughout training. Images are resized to 224×224 pixels [72]. For CIFAR-100 and ImageNet-R, Office-Home, Office-31, DomainNet, datasets, we train each task for 10 epochs, while for the more challenging Cars and CUB-200 datasets, training extends to 30 epochs per task. To preserve the open-vocabulary capability and performance consistency, the classification layer remains frozen as per [36, 71], ensuring the model effectively retains its pre-trained knowledge base while fine-tuning.

Evaluation Metrics Following the conventional continual learning framework, we assess the overall performance of a method by reporting its average accuracy after training on all tasks. Let $R_{N,i}$ represent the classification accuracy of task T_i after training on task T_N, the average accuracy A_N is then defined as: $A_N = \frac{1}{N}\sum_{i=1}^{N} R_{N,i}$.

5.2. Class-incremental experimental results

We evaluate our proposed method in a class-incremental setting, comparing it to various baselines across several datasets. Table 1 showcases the task-agnostic accuracy achieved by each method after the completion of the final task across multiple datasets. The "Zero-shot" method serves as a naive baseline, while the "Join" training method represents an upper bound on performance, assuming full data availability during training. We also distinguish between methods with constant memory usage, denoted as $\Theta(1)$, and methods with memory usage that scales linearly with the number of tasks, denoted as $\Theta(N)$.

While EWC [73] improves over the naive baselines, its performance often lags behind more recent techniques. Note that our method consistently outperforms other constant-memory approaches and frequently matches or even surpasses the performance of methods with linear memory usage. For example, it achieves an average accuracy of 72.54%, outperforming RandMix, which reaches 70.4%. Also, ours achieves the best or second-best performance across all configurations, getting a $\approx 5\%$ margin over Zero-shot CLIP. On CIFAR100, our method achieves strong results across all incremental splits, confirming its robustness as the number of tasks grows. On ImageNet-R, a dataset designed for studying adversarial robustness and transfer learning, our method continues to maintain a leading position among non linear-memory approaches, showing resilience to distribution shifts. On the Cars dataset, our method improves performance by 2–3 percentage points over the second best method. These results demonstrate that our method not only adheres to strict memory constraints but also effectively retains knowledge across multiple tasks, making it a viable option for real-world applications that prioritize memory efficiency. In contrast, methods like MaxAbs show poor performance, even lagging behind LwF, despite requiring memory proportional to the number of tasks, $\Theta(N)$.

Next, we assess the effectiveness of various fine-

tuning strategies for continual learning using ViT [74] and CLIP [46] across two benchmark datasets, CIFAR100 and ImageNet-R. The baseline list includes a comprehensive set of established regularization-based continual learning, model-merging, and prompt-based methods.

Table 2. Results for different fine-tuning strategies. Baselines results are taken from [42, 75, 76]

Memory	Method	CIFAR100	ImageNet-R	Avg
$\Theta(1)$	Zero-shot	66.91	77.73	72.32
N/A	Joint	90.94	87.55	89.24
$\Theta(N)$	RandMix [42]	77.04	81.88	79.46
$\Theta(N)$	MaxAbs [42]	76.75	82.33	79.54
$\Theta(N)$	Avg [35]	77.04	81.87	79.46
$\Theta(N)$	TIES [40]	<u>77.23</u>	82.27	79.75
$\Theta(N)$	iCaRL [25]	57.60	43.71	50.66
$\Theta(N)$	L2P [31]	70.04	69.33	69.69
$\Theta(N)$	DualPrompt [32]	74.31	76.41	75.36
$\Theta(N)$	CODA-P [34]	76.4	79.5	77.95
$\Theta(N)$	PROOF [75]	76.55	79.7	78.13
$\Theta(1)$	LwF [26]	73.45	**82.97**	78.21
$\Theta(1)$	EWC [24]	76.24	82.42	79.33
$\Theta(1)$	Continual-CLIP[77]	68.26	76.94	72.6
$\Theta(1)$	CoOp [78]	70.58	78.66	74.62
$\Theta(1)$	MaPLe [79]	74.52	79.71	77.12
$\Theta(1)$	AttriCLIP [69]	68.45	76.53	72.49
$\Theta(1)$	CLIP-Adapter [80]	68.32	78.1	73.21
$\Theta(1)$	VPT [81]	59.33	74.0	66.66
$\Theta(1)$	Ours	**77.72**	<u>82.69</u>	**80.21**

As summarized in Table 2, our method achieves state-of-the-art performance among $\Theta(1)$ fine-tuning strategies, consistently outperforming or matching more memory-intensive $\Theta(N)$ methods. Specifically, on CIFAR-100, we achieve 77.72% accuracy, surpassing not only baselines such as LwF and EWC but also recent prompt-tuning methods like PROOF by over 1%. Moreover, on ImageNet-R, our method maintains a high accuracy of 82.69%, positioning it as the runner-up for this dataset. These results confirm that our approach narrows the performance gap with $\Theta(N)$ methods, often surpassing them, all while maintaining a constant memory footprint.

5.3. Domain-incremental experimental results

Table 3 reports accuracy (%) after the final task on four domain-incremental benchmarks: Office-Home, Office-31, DomainNet, and ImageNet-R. As shown, our approach consistently attains high accuracy across diverse domains, outperforming or closely matching the strongest merging-based baselines. Despite using significantly less memory than $\Theta(N)$ approaches, our method achieves a superior average accuracy of 84.49%, outperforming or matching specialized continual learning techniques and other merging-based baselines.

From the per-dataset results, we see that our method consistently ranks among the top performers. On Office-31, it achieves 94.05%, which is the best result overall, while on ImageNet-R it reaches 87.35%, again surpassing all baselines by a clear margin. Even on DomainNet—a challenging dataset—our method's 67.27% accuracy remains competitive, trailing only LwF by a small gap. These strong individual dataset performances culminate in the highest average accuracy among all methods. It is also worth noting that, LwF is a strong baseline for this DIL experiment, despite its simplicity.

Table 3. Our proposed method 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 <u>underlined</u>.

Memory	Method	Office-Home	Office-31	DomainNet	ImageNet-R	Avg
$\Theta(N)$	RandMix [42]	**89.45**	93.41	64.31	82.28	82.36
$\Theta(N)$	MaxAbs [42]	89.09	90.63	67.51	83.93	82.79
$\Theta(N)$	Avg [35]	85.21	89.87	64.98	82.98	80.76
$\Theta(N)$	TIES [62]	88.59	89.74	66.42	83.90	82.16
$\Theta(1)$	LwF [26]	88.56	<u>93.92</u>	**69.76**	<u>84.78</u>	<u>84.25</u>
$\Theta(1)$	EWC [24]	88.06	92.78	56.58	83.77	80.29
$\Theta(1)$	Ours	<u>89.29</u>	**94.05**	<u>67.27</u>	**87.35**	**84.49**

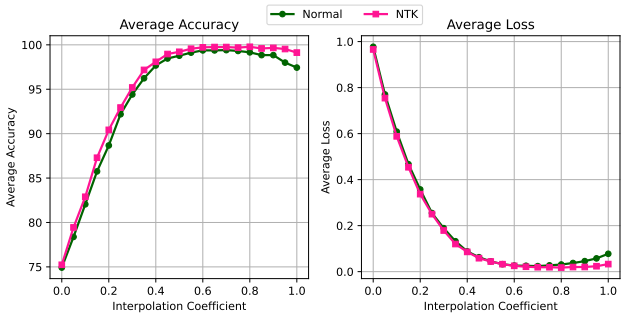

Figure 4. Average accuracy score and loss value when using our merging method in interpolating models after training on first and second tasks using normal training and our proposed linearly fine-tuning method.

6. Ablation studies

To gain a deeper understanding of the contributions made by different components of our proposed method, we conduct a comprehensive ablation study. There are three main components of our proposed method, which correspond to three stages in Algorithm 1. In Figures 2 and 4, we plot the accuracy and cross-entropy loss as a function of the interpolation coefficient. As shown in these figures, the accuracy curves for NTK-based fine-tuning consistently outperform those of conventional fine-tuning, both for linear weight averaging and our proposed merging method. Similarly, the loss curves follow the same general trend, with NTK fine-tuning consistently yielding lower loss values

across the interpolation trajectory. We hypothesize that this improvement can be attributed to the reduction of interference [60] when merging task-specific models.

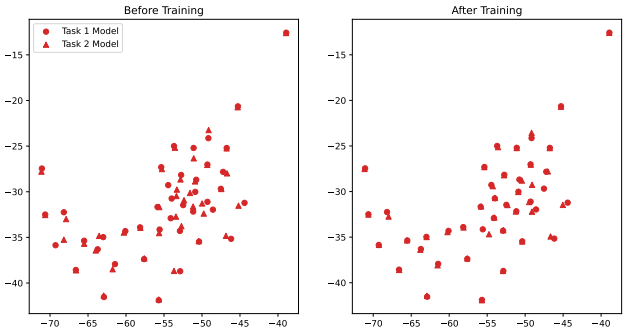

Figure 5. t-SNE of samples from the 12-th class using models from the previous task ○ and current task △.

Next, we examine the impact of feature representation refinement on embedding alignment. Figures 5, 6 and 7 illustrate the t-SNE of embeddings extracted from three different classes randomly sampled from the Cars dataset. These visualizations help assess the alignment of the learned feature representations across tasks.

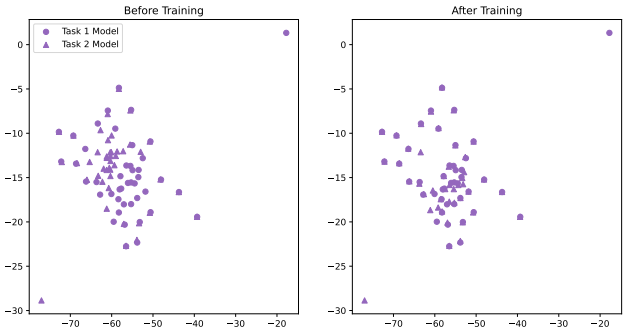

Figure 6. t-SNE of samples from the 18-th class using models from the previous task ○ and current task △.

From the left-hand panels, we observe that the feature representation distributions of the merged model (represented by △ points) and the model from the previous task (represented by ○ points) exhibit significant misalignment. The merged model's embeddings are scattered, and there is little alignment with the previous task's embeddings, suggesting that merging distorts the representations. In contrast, note that the right-hand figures show that these triangle points are brought closer to the circle points following our post-merging procedure. By fine-tuning the lightweight image projection layer, the triangle points shift toward the circle points, confirming the effectiveness of the third stage in our proposed algorithm.

Table 4 presents a detailed breakdown of the impact of each component of our method. As expected, fine-tuning results in a clear performance improvement over zero-shot predictions. When merging fine-tuned current and previous models with our FIM approximation, a significant performance boost is observed. Next, task-specific weight disentanglement and representation alignment help improve accuracy by reducing task interference and aligning learned representations, respectively.

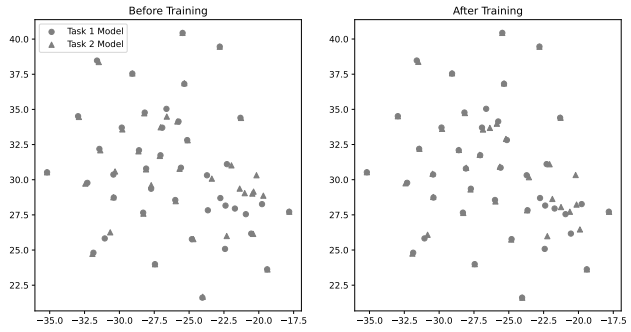

Figure 7. t-SNE of samples from the 29-th class using models from the previous task ○ and current task △.

Table 4. Ablative studies of our proposed method on Office-31. Each individual component of our proposed method contributes a distinct performance gain while combining all components yields the highest overall accuracy.

FT	Merge	NTK	Bias	Amazon	Dslr	Webcam	Avg
×	×	×	×	80.79	83.15	83.89	82.40
✓	×	×	×	88.59	93.26	96.64	90.51
✓	✓	×	×	91.53	95.28	91.95	92.91
✓	✓	✓	×	91.30	96.63	97.32	93.79
✓	✓	✓	✓	91.67	96.63	97.32	94.05

7. Conclusion

In this work, we tackle the challenges of applying model-merging techniques within the context of continual learning. Drawing on our findings, we propose a scalable and effective approach that provides a closed-form solution approximating the minimizer of the weighted task loss. Our extensive experimental results demonstrate the superiority of our method across various class-incremental and domain-incremental scenarios.

Despite these successes, there are opportunities for further enhancement, particularly in adapting pre-trained models such as CLIP to novel downstream tasks exhibiting significant covariate shifts that fall outside the scope of CLIP's original training set. The reliance on linear fine-tuning and other parameter-efficient methods, such as adapter and prompt-tuning, may limit the modeling capacity of the original CLIP model. This limitation presents a challenge in effectively adapting to new tasks and highlights potential areas for future research.

References

[1] J. Achiam, S. Adler, S. Agarwal, L. Ahmad, I. Akkaya, F. L. Aleman, D. Almeida, J. Altenschmidt, S. Altman, S. Anadkat, *et al.*, "Gpt-4 technical report," *arXiv preprint arXiv:2303.08774*, 2023. 1

[2] A. Hurst, A. Lerer, A. P. Goucher, A. Perelman, A. Ramesh, A. Clark, A. Ostrow, A. Welihinda, A. Hayes, A. Radford, *et al.*, "Gpt-4o system card," *arXiv preprint arXiv:2410.21276*, 2024. 1

[3] H. Touvron, L. Martin, K. Stone, P. Albert, A. Almahairi, Y. Babaei, N. Bashlykov, S. Batra, P. Bhargava, S. Bhosale, *et al.*, "Llama 2: Open foundation and fine-tuned chat models," *arXiv preprint arXiv:2307.09288*, 2023. 1

[4] A. Grattafiori, A. Dubey, A. Jauhri, A. Pandey, A. Kadian, A. Al-Dahle, A. Letman, A. Mathur, A. Schelten, A. Vaughan, *et al.*, "The llama 3 herd of models," *arXiv preprint arXiv:2407.21783*, 2024. 1

[5] A. Liu, B. Feng, B. Wang, B. Wang, B. Liu, C. Zhao, C. Dengr, C. Ruan, D. Dai, D. Guo, *et al.*, "Deepseek-v2: A strong, economical, and efficient mixture-of-experts language model," *arXiv preprint arXiv:2405.04434*, 2024. 1

[6] A. Liu, B. Feng, B. Xue, B. Wang, B. Wu, C. Lu, C. Zhao, C. Deng, C. Zhang, C. Ruan, *et al.*, "Deepseek-v3 technical report," *arXiv preprint arXiv:2412.19437*, 2024.

[7] D. Guo, D. Yang, H. Zhang, J. Song, R. Zhang, R. Xu, Q. Zhu, S. Ma, P. Wang, X. Bi, *et al.*, "Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning," *arXiv preprint arXiv:2501.12948*, 2025. 1

[8] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer, "High-resolution image synthesis with latent diffusion models," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 10684–10695, 2022. 1

[9] T. Brooks, B. Peebles, C. Holmes, W. DePue, Y. Guo, L. Jing, D. Schnurr, J. Taylor, T. Luhman, E. Luhman, *et al.*, "Video generation models as world simulators," *OpenAI Blog*, vol. 1, p. 8, 2024. 1

[10] W. Li, Y. Peng, M. Zhang, L. Ding, H. Hu, and L. Shen, "Deep model fusion: A survey," *arXiv preprint arXiv:2309.15698*, 2023. 1

[11] E. Yang, L. Shen, G. Guo, X. Wang, X. Cao, J. Zhang, and D. Tao, "Model merging in llms, mllms, and beyond: Methods, theories, applications and opportunities," *arXiv preprint arXiv:2408.07666*, 2024. 1

[12] E. Yang, Z. Wang, L. Shen, S. Liu, G. Guo, X. Wang, and D. Tao, "Adamerging: Adaptive model merging for multi-task learning," *arXiv preprint arXiv:2310.02575*, 2023. 1

[13] A. Ahmadian, S. Goldfarb-Tarrant, B. Ermis, M. Fadaee, S. Hooker, *et al.*, "Mix data or merge models? optimizing for diverse multi-task learning," *arXiv preprint arXiv:2410.10801*, 2024.

[14] M. Li, Z. Nie, Y. Zhang, D. Long, R. Zhang, and P. Xie, "Improving general text embedding model: Tackling task conflict and data imbalance through model merging," *arXiv preprint arXiv:2410.15035*, 2024. 1

[15] Z. Li, T. Lin, X. Shang, and C. Wu, "Revisiting weighted aggregation in federated learning with neural networks," in *International Conference on Machine Learning*, pp. 19767–19788, PMLR, 2023. 1

[16] Y. Jia, X. Zhang, H. Hu, K.-K. R. Choo, L. Qi, X. Xu, A. Beheshti, and W. Dou, "Dapperfl: Domain adaptive federated learning with model fusion pruning for edge devices," *Advances in Neural Information Processing Systems*, vol. 37, pp. 13099–13123, 2024.

[17] N. Saadati, M. Pham, N. Saleem, J. R. Waite, A. Balu, Z. Jiang, C. Hegde, and S. Sarkar, "Dimat: Decentralized iterative merging-and-training for deep learning models," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 27517–27527, 2024. 1

[18] R. Chitale, A. Vaidya, A. Kane, and A. Ghotkar, "Task arithmetic with lora for continual learning," *arXiv preprint arXiv:2311.02428*, 2023. 1

[19] V. Araujo, M. F. Moens, and T. Tuytelaars, "Learning to route for dynamic adapter composition in continual learning with language models," in *Findings of the Association for Computational Linguistics: EMNLP 2024*, pp. 687–696, 2024.

[20] L. Quarantiello, E. N. Coleman, J. Hurtado, and V. Lomonaco, "Adaptive lora merging for efficient domain incremental learning," in *Latinx in AI@ NeurIPS 2024*. 1

[21] C. Goddard, S. Siriwardhana, M. Ehghaghi, L. Meyers, V. Karpukhin, B. Benedict, M. McQuade, and J. Solawetz, "Arcee's mergekit: A toolkit for merging large language models," in *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing: Industry Track*, pp. 477–485, 2024. 1

[22] A. Tang, L. Shen, Y. Luo, H. Hu, B. Du, and D. Tao, "Fusionbench: A comprehensive benchmark of deep model fusion," *arXiv preprint arXiv:2406.03280*, 2024. 1

[23] J. Krause, M. Stark, J. Deng, and L. Fei-Fei, "3d object representations for fine-grained categorization," in *Proceedings of the IEEE international conference on computer vision workshops*, pp. 554–561, 2013. 2, 5

[24] J. Kirkpatrick, R. Pascanu, N. Rabinowitz, J. Veness, G. Desjardins, A. A. Rusu, K. Milan, J. Quan, T. Ramalho, A. Grabska-Barwinska, *et al.*, "Overcoming catastrophic forgetting in neural networks," *Proceedings of the National Academy of Sciences*, vol. 114, no. 13, pp. 3521–3526, 2017. 2, 6, 7, 13

[25] S.-A. Rebuffi, A. Kolesnikov, G. Sperl, and C. H. Lampert, "icarl: Incremental classifier and representation learning," in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. https://arxiv.org/abs/1611.07725. 2, 7, 13

[26] Z. Li and D. Hoiem, "Learning without forgetting," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2017. https://arxiv.org/abs/1606.09282. 2, 6, 7, 13

[27] A. A. Rusu, N. C. Rabinowitz, G. Desjardins, H. Soyer, J. Kirkpatrick, K. Kavukcuoglu, R. Pascanu, and R. Hadsell, "Progressive neural networks," 2016. https://arxiv.org/abs/1606.04671. 2

[28] S. Yan, J. Xie, and X. He, "Der: Dynamically expandable representation for class incremental learning," in *Proceedings*

of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 3014–3023, 2021. 2

[29] P. Buzzega, M. Boschini, A. Porrello, D. Abati, and S. Calderara, "Dark experience for general continual learning: a strong, simple baseline," *Advances in neural information processing systems*, vol. 33, pp. 15920–15930, 2020. 2

[30] H. Shin, J. K. Lee, J. Kim, and J. Kim, "Continual learning with deep generative replay," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2017. https://arxiv.org/abs/1705.08690. 2

[31] Z. Wang, Z. Zhang, C.-Y. Lee, H. Zhang, R. Sun, X. Ren, G. Su, V. Perot, J. Dy, and T. Pfister, "Learning to prompt for continual learning," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 139–149, 2022. 2, 6, 7, 13

[32] Z. Wang, Z. Zhang, S. Ebrahimi, R. Sun, H. Zhang, C.-Y. Lee, X. Ren, G. Su, V. Perot, J. Dy, *et al.*, "Dualprompt: Complementary prompting for rehearsal-free continual learning," in *European Conference on Computer Vision*, pp. 631–648, Springer, 2022. 2, 6, 7, 13

[33] L. Wang, X. Zhang, H. Su, and J. Zhu, "A comprehensive survey of continual learning: Theory, method and application," *arXiv preprint arXiv:2302.00487*, 2023. 2

[34] J. S. Smith, L. Karlinsky, V. Gutta, P. Cascante-Bonilla, D. Kim, A. Arbelle, R. Panda, R. Feris, and Z. Kira, "Coda-prompt: Continual decomposed attention-based prompting for rehearsal-free continual learning," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 11909–11919, 2023. 2, 6, 7, 13

[35] M. Wortsman, G. Ilharco, S. Y. Gadre, R. Roelofs, R. Gontijo-Lopes, A. S. Morcos, H. Namkoong, A. Farhadi, Y. Carmon, S. Kornblith, *et al.*, "Model soups: averaging weights of multiple fine-tuned models improves accuracy without increasing inference time," in *International Conference on Machine Learning (ICML)*, 2022. https://arxiv.org/abs/2203.05482. 2, 6, 7, 13

[36] G. Ilharco, M. T. Ribeiro, M. Wortsman, S. Gururangan, L. Schmidt, H. Hajishirzi, and A. Farhadi, "Editing models with task arithmetic," in *International Conference on Learning Representations (ICLR)*, 2023. https://arxiv.org/abs/2110.08207. 2, 6

[37] M. S. Matena and C. A. Raffel, "Merging models with fisher-weighted averaging," *Advances in Neural Information Processing Systems*, vol. 35, pp. 17703–17716, 2022. 2

[38] X. Jin, X. Ren, D. Preotiuc-Pietro, and P. Cheng, "Dataless knowledge fusion by merging weights of language models," in *The Eleventh International Conference on Learning Representations*, 2023.

[39] S. K. Ainsworth, J. Hayase, and S. Srinivasa, "Git re-basin: Merging models modulo permutation symmetries," in *International Conference on Learning Representations (ICLR)*, 2023. https://arxiv.org/abs/2209.04836. 2

[40] P. Yadav, D. Tam, L. Choshen, C. A. Raffel, and M. Bansal, "Ties-merging: Resolving interference when merging models," *Advances in Neural Information Processing Systems*, vol. 36, pp. 7093–7115, 2023. 2, 6, 7, 13

[41] N. Dhawan, N. E. Mitchell, Z. Charles, Z. Garrett, and G. K. Dziugaite, "Leveraging function space aggregation for federated learning at scale," *Transactions on Machine Learning Research*, 2024. Expert Certification. 2

[42] D. Marczak, B. Twardowski, T. Trzciński, and S. Cygert, "Magmax: Leveraging model merging for seamless continual learning," in *European Conference on Computer Vision*, pp. 379–395, Springer, 2024. 2, 5, 6, 7, 13

[43] I. E. Marouf, S. Roy, E. Tartaglione, and S. Lathuilière, "Weighted ensemble models are strong continual learners," 2024. 3

[44] D. P. Kingma, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014. 3, 4

[45] I. Loshchilov and F. Hutter, "Decoupled weight decay regularization," in *International Conference on Learning Representations (ICLR)*, 2019. 3, 6

[46] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, *et al.*, "Learning transferable visual models from natural language supervision," in *International Conference on Machine Learning*, pp. 8748–8763, PMLR, 2021. 3, 6, 7

[47] I. Loshchilov, "Decoupled weight decay regularization," *arXiv preprint arXiv:1711.05101*, 2017. 3

[48] S.-I. Amari, "Natural gradient works efficiently in learning," *Neural computation*, vol. 10, no. 2, pp. 251–276, 1998. 3

[49] R. Eschenhagen, A. Immer, R. Turner, F. Schneider, and P. Hennig, "Kronecker-factored approximate curvature for modern neural network architectures," *Advances in Neural Information Processing Systems*, vol. 36, pp. 33624–33655, 2023.

[50] D. Hwang, "Fadam: Adam is a natural gradient optimizer using diagonal empirical fisher information," *arXiv preprint arXiv:2405.12807*, 2024. 3

[51] A. Jacot, F. Gabriel, and C. Hongler, "Neural tangent kernel: Convergence and generalization in neural networks," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2018. https://proceedings.neurips.cc/paper_files/paper/2018/file/5a4be1fa34e62bb8a6ec6b91d2462f5a-Paper.pdf. 3

[52] S. Arora, S. S. Du, W. Hu, Z. Li, R. Salakhutdinov, and R. Wang, "On exact computation with an infinitely wide neural net," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2019. https://proceedings.neurips.cc/paper/2019/hash/dbc4d84bfcfe2284ba11beffb853a8c4-Abstract.html.

[53] J. Lee, L. Xiao, S. Schoenholz, Y. Bahri, R. Novak, J. Sohl-Dickstein, and J. Pennington, "Wide neural networks of any depth evolve as linear models under gradient descent," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2019. https://proceedings.neurips.cc/paper_files/paper/2019/file/0d1a9651497a38d8b1c3871c84528bd4-Paper.pdf. 3

[54] S. Malladi, A. Wettig, D. Yu, D. Chen, and S. Arora, "A kernel-based view of language model fine-tuning," 2022. https://arxiv.org/abs/2210.05643. 4

[55] G. Ortiz-Jiménez, A. Modas, S. Moosavi-Dezfooli, and P. Frossard, "What can linearized neural networks actually say about generalization?," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2021. https://proceedings.neurips.cc/paper_files/paper/2021/file/4b5deb9a14d66ab0acc3b8a2360cde7c-Paper.pdf.

[56] L. Zancato, A. Achille, A. Ravichandran, R. Bhotika, and S. Soatto, "Predicting training time without training," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2020. https://proceedings.neurips.cc/paper/2020/hash/440e7c3eb9bbcd4c33c3535354a51605-Abstract.html.

[57] A. Deshpande, A. Achille, A. Ravichandran, H. Li, L. Zancato, C. C. Fowlkes, R. Bhotika, S. Soatto, and P. Perona, "A linearized framework and a new benchmark for model selection for fine-tuning," 2021. https://arxiv.org/abs/2102.00084.

[58] G. Yüce, G. Ortiz-Jiménez, B. Besbinar, and P. Frossard, "A structured dictionary perspective on implicit neural representations," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. https://doi.org/10.1109/CVPR52688.2022.01863. 4

[59] F. Kunstner, P. Hennig, and L. Balles, "Limitations of the empirical fisher approximation for natural gradient descent," *Advances in neural information processing systems*, vol. 32, 2019. 4

[60] G. Ortiz-Jimenez, A. Favero, and P. Frossard, "Task arithmetic in the tangent space: Improved editing of pre-trained models," *Advances in Neural Information Processing Systems*, vol. 36, pp. 66727–66754, 2023. 4, 8

[61] S. Jha, D. Gong, and L. Yao, "Clap4clip: Continual learning with probabilistic finetuning for vision-language models," *Advances in neural information processing systems*, vol. 37, pp. 129146–129186, 2025. 5

[62] P. Yadav, D. Tam, L. Choshen, C. Raffel, and M. Bansal, "Resolving interference when merging models," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2023. https://arxiv.org/abs/2306.01708. 6, 7

[63] A. Krizhevsky, G. Hinton, *et al.*, "Learning multiple layers of features from tiny images," 2009. https://www.cs.toronto.edu/~kriz/learning-features-2009-TR.pdf. 5

[64] D. Hendrycks, S. Basart, N. Mu, S. Kadavath, F. Wang, E. Dorundo, R. Desai, T. Zhu, S. Parajuli, M. Guo, *et al.*, "The many faces of robustness: A critical analysis of out-of-distribution generalization," in *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 8340–8349, 2021. 5, 6, 13

[65] C. Wah, S. Branson, P. Welinder, P. Perona, and S. Belongie, "The caltech-ucsd birds-200-2011 dataset," 2011. 5

[66] K. Saenko, B. Kulis, M. Fritz, and T. Darrell, "Adapting visual category models to new domains," in *Proceedings of the 11th European Conference on Computer Vision: Part IV*, ECCV'10, (Berlin, Heidelberg), p. 213–226, Springer-Verlag, 2010. 5, 13

[67] H. Venkateswara, J. Eusebio, S. Chakraborty, and S. Panchanathan, "Deep hashing network for unsupervised domain adaptation," *arXiv preprint arXiv: 1706.07522*, 2017. 5, 13

[68] X. Peng, Q. Bai, X. Xia, Z. Huang, K. Saenko, and B. Wang, "Moment matching for multi-source domain adaptation," *IEEE International Conference on Computer Vision*, 2018. 6, 13

[69] R. Wang, X. Duan, G. Kang, J. Liu, S. Lin, S. Xu, J. Lü, and B. Zhang, "Attriclip: A non-incremental learner for incremental knowledge learning," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 3654–3663, 2023. 6, 7, 13

[70] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, *et al.*, "An image is worth 16x16 words: Transformers for image recognition at scale," *arXiv preprint arXiv:2010.11929*, 2020. 6

[71] G. Ilharco, M. Wortsman, S. Y. Gadre, S. Song, H. Hajishirzi, S. Kornblith, A. Farhadi, and L. Schmidt, "Patching open-vocabulary models by interpolating weights," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2022. https://arXiv.org/abs/2208.05592. 6

[72] G. Ilharco, M. Wortsman, R. Wightman, C. Gordon, N. Carlini, R. Taori, A. Dave, V. Shankar, H. Namkoong, J. Miller, H. Hajishirzi, A. Farhadi, and L. Schmidt, "Openclip," 2021. https://github.com/mlfoundations/open_clip. 6

[73] W. Hu, Z. Lin, B. Liu, C. Tao, Z. T. Tao, D. Zhao, J. Ma, and R. Yan, "Overcoming catastrophic forgetting for continual learning via model adaptation," in *International Conference on Learning Representations*, 2019. 6

[74] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby, "An image is worth 16x16 words: Transformers for image recognition at scale," in *International Conference on Learning Representations (ICLR)*, 2021. https://openreview.net/forum?id=YicbFdNTTy. 7

[75] D.-W. Zhou, Y. Zhang, Y. Wang, J. Ning, H.-J. Ye, D.-C. Zhan, and Z. Liu, "Learning without forgetting for vision-language models," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2025. 7, 13

[76] S. Jha, D. Gong, and L. Yao, "Clap4clip: Continual learning with probabilistic finetuning for vision-language models," *CoRR*, vol. abs/2403.19137, 2024. 7

[77] V. Thengane, S. Khan, M. Hayat, and F. Khan, "Clip model is an efficient continual learner," *arXiv preprint arXiv:2210.03114*, 2022. 7

[78] Y. Du, F. Wei, Z. Zhang, M. Shi, Y. Gao, and G. Li, "Learning to prompt for open-vocabulary object detection with vision-language model," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 14084–14093, 2022. 7, 13

[79] M. U. Khattak, H. Rasheed, M. Maaz, S. H. Khan, and F. Khan, "Maple: Multi-modal prompt learning," *Computer Vision and Pattern Recognition*, 2022. 7, 13

[80] P. Gao, S. Geng, R. Zhang, T. Ma, R. Fang, Y. Zhang, H. Li, and Y. Qiao, "Clip-adapter: Better vision-language models with feature adapters," *International Journal of Computer Vision*, vol. 132, no. 2, pp. 581–595, 2024. 7, 13

[81] M. M. Derakhshani, E. Sanchez, A. Bulat, V. G. T. da Costa, C. G. Snoek, G. Tzimiropoulos, and B. Martinez, "Bayesian prompt learning for image-language model generalization," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 15237–15246, 2023. 7, 13

A. Supplementary for Toward a Holistic Approach to Continual Model Merging

Due to space constraints, some details were omitted from the main paper. We therefore include baselines' description used in the main paper (Appendix A.1) and detailed experimental setup (Appendix A.2) in this supplementary material.

A.1. Baseline overview

We compare our proposed method with various continual learning approaches, including general methods, model merging techniques, and prompt learning method that can be applied to continual learning.

- Learning without Forgetting (LwF) [26] overcomes forgetting with a distillation loss between the current model and the previously trained one, both applied to the current task's data.
- Elastic Weight Consolidation (EWC) [24] regularizes weight changes with a task-importance mask that is accumulatively computed using the diagonal empirical Fisher Information matrix at the end of each task.
- iCaRL [25] combines a nearest-mean-of-exemplars classifier with knowledge distillation to retain performance on old classes while learning new ones.
- Model Soup (Avg) [35] simply averages task-specific model weights to obtain the merged one.
- RandMix [42] constructs the final weight by randomly choosing one parameter from one of the fine-tuned models.
- MaxAbs [42] constructs the final task vector by selecting the largest magnitude value from all individual task vectors for each entry.
- TIES-Merging (TIES) [40] mitigates interference between task vectors during model merging by selectively pruning parameters and determining optimal parameter signs.
- Learning to Prompt (L2P) [31] introduced the idea of using a shared pool of prompts, from which the top-k most relevant prompts are chosen for each sample during both training and testing. While this method may facilitate knowledge transfer across tasks, it also poses a risk of catastrophic forgetting.
- DualPrompt [32] addresses the limitations of L2P by attaching auxiliary prompts directly to the pretrained backbone, rather than relying solely on input-level prompting. Additionally, it introduces separate prompt sets for each task, allowing the model to capture more task-specific instructions alongside the task-invariant knowledge drawn from the shared prompt pool.
- CODA-Prompt [34] uses learnable prompts tailored to each task. Similar to L2P, CODA maintains a pool of prompts and keys, generating the final prompt as a weighted sum based on the cosine similarity between queries and keys. To bypass explicit task prediction at test time, it computes the weighted sum using all prompts, regardless of the task.
- CoOp [78] shares a set of learnable vector tokens for all task classes.
- MaPLe [79] employs two set of learnable prompts for visual and textual encoders, respectively.
- CLIP-Adapter [80] incorporates lightweight modules over textual and visual branches to enrich respective features of the frozen CLIP model.
- Variational Prompt Tuning (VPT) [81] learns a distribution over input prompt tokens for generalizability, but also faces a trade-off with in-domain performance.
- AttriCLIP [69] is a non-incremental learner designed for CL, built on top of a frozen CLIP model with fixed image and text encoders. It introduces learnable attribute prompts selected from a predefined word bank to enrich category names, enabling classification via visual-textual similarity without updating the classifier head.
- PROOF [75] extends LwF to vision-language models by freezing the pretrained image and text encoders and introducing task-specific projection layers for each new task. These projections are trained independently and fused via cross-modal attention, allowing the model to adapt to new tasks while preserving knowledge from previous ones.

A.2. Domain-incremental benchmarks

To evaluate our method under domain-incremental setting, we leverage benchmarks from domain adaptation literature: Office-31 [66], Office Home [67], DomainNet [68] and ImageNet-R [64]. For each dataset, tasks are defined as domains. Specifically, Office-31 consists of three different domains, hence is split into 3 tasks. Similarly, Office Home, DomainNet, has 4, 6, and 15 domains, respectively.