Learning Adaptive and Expandable Mixture Model for Continual Learning

Fei Ye[1], YongCheng Zhong[1], QiHe Liu[1]*, Adrian G. Bors[2],
JingLing Sun[1], JinYu Guo[1], ShiJie Zhou[1]

[1]School of Information and Software Engineering, University of Electronic Science and Technology of China
[2]Department of Computer Science, University of York
feiye@uestc.edu.cn, 202422090410@std.uestc.edu.cn, qiheliu@uestc.edu.cn,
adrian.bors@york.ac.uk, jlsun@uestc.edu.cn, guojinyu@uestc.edu.cn, sjzhou@uestc.edu.cn

Abstract

Continuous learning constitutes a fundamental capability of artificial intelligence systems, enabling them to incrementally assimilate novel information without succumbing to catastrophic forgetting. Recent research has leveraged Pre-Trained Models (PTMs) to enhance continual learning efficacy. Nevertheless, prevailing methodologies typically depend on a singular pre-trained backbone and freeze all pre-trained parameters to mitigate network forgetting, thereby constraining adaptability to emerging tasks. In this study, we introduce an innovative PTM-based framework featuring a Dual-Representation Backbone Architecture (DRBA), which integrates both invariant and evolved representation networks to concurrently capture static and dynamic features. Building upon DRBA, we propose an Adaptive and Expandable Mixture Model (AEMM) that incrementally incorporates new expert modules with minimal parameter overhead to accommodate the learning of each novel task. To further augment adaptability, we develop a Dynamic Adaptive Representation Fusion Mechanism (DARFM) that processes outputs from both representation networks and autonomously generates data-driven adaptive weights, optimizing the contribution of each representation. This mechanism yields an adaptive, semantically enriched composite representation, thereby maximizing positive knowledge transfer. Additionally, we propose a Dynamic Knowledge Calibration Mechanism (DKCM), comprising prediction and representation calibration processes, to ensure consistency in both predictions and feature representations. This approach achieves a balance between stability and plasticity, even when learning complex datasets. Empirical evaluations substantiate that the proposed approach attains state-of-the-art performance.

Code — https://github.com/CL-Coder236/AEMM
Appendix — https://github.com/CL-Coder236/AEMM

Introduction

Current modern deep learning technologies have achieved significant performance in various computer vision applications. However, the performance of these methods highly relies on the availability of extensive annotated datasets (He et al. 2016; Hinton, Osindero, and Teh 2006). In many practical applications such as autonomous driving and robot navigation, previously learned data samples are usually absent. Such a learning paradigm is called Continual Learning (CL). Applying modern learning technologies in CL can suffer from significant performance degeneration on prior tasks caused by catastrophic forgetting (Parisi et al. 2019). Such performance degeneration happens when the model tries to modify all parameters to adapt to the learning of new tasks. To mitigate network forgetting in CL, recent research has introduced diverse strategies, including dynamic network expansion techniques (McDonnell et al. 2023), regularization-based approaches (Kemker et al. 2018; Martens and Grosse 2015) and memory replay mechanisms (Bang et al. 2021).

While current technologies demonstrate commendable performance, they remain constrained by insufficient plasticity when confronted with more complex datasets. Plasticity, in this context, denotes the capacity to learn novel tasks. Consequently, recent research has investigated the utilization of PTMs (McDonnell et al. 2023; Villa et al. 2023) to bolster plasticity within continual learning frameworks. PTMs are capable of generating semantically enriched representations for input data and can incrementally introduce and integrate additional trainable parameters to accommodate the learning of each new task. Nevertheless, prevailing PTM-based approaches encounter two primary challenges: (1) They usually adopt a single pre-trained Vision Transformer (ViT) (Dosovitskiy et al. 2020) as the backbone, which would suffer from poor generalization when learning several different data domains; (2) They freeze all parameters of PTMs and only optimize a few trainable additional parameters, resulting in limited plasticity; Recent biological research indicates that the interplay between the hippocampus and neocortex establishes a dual-system memory framework, enabling the processing of both gradually changing and rapidly evolving information (McClelland, McNaughton, and O'Reilly 1995). Drawing inspiration from these findings, we introduce a novel Dual-Representation Backbone Architecture (DRBA), which comprises an invariant representation network for encoding stable information and an evolved representation network for modeling dynamically changing data. Both networks leverage pre-trained Vision Transformers (ViT) (Dosovitskiy et al. 2020) as their foundational models. To enhance computational efficiency

*corresponding author

and minimize parameter redundancy, we implement extensive parameter sharing between the invariant and evolved networks. Building upon the DRBA, we further propose an Adaptive and Expandable Mixture Model (AEMM) that incrementally allocates expert modules for each new task. Each expert module incorporates a feature transformation layer that projects the augmented representations from the DRBA into task-specific feature vectors, which are subsequently processed by a linear classifier for prediction.

While integrating features from both invariant and evolved representation networks can yield semantically enriched information, such an approach overlooks the inherent dependencies between data and tasks, thereby limiting the potential for optimal positive knowledge transfer. To enhance model plasticity, we introduce a novel Dynamic Adaptive Representation Fusion Mechanism (DARFM), designed to generate adaptive weights that modulate the contribution of each representation network based on data complexity and specific characteristics. The DARFM employs a trainable function that leverages a self-attention mechanism to process inputs from both invariant and evolved representation networks, autonomously generating data-driven adaptive weights. These weights are subsequently utilized during feature fusion, resulting in adaptive and robust representations that maximize positive knowledge transfer.

Dynamically tuning the parameters of the evolved representation network enables rapid adaptation to novel tasks. Nevertheless, this optimization process may compromise stability, defined as the retention of previously acquired knowledge. To mitigate this challenge, we introduce a Dynamic Knowledge Calibration Mechanism (DKCM) designed to incrementally refine the evolved representation network while preserving stability. The core concept of DKCM is to calibrate the output distributions between the current and prior learning models. DKCM consists of a prediction calibration process that harmonizes the outputs of all historical experts, together with a representation calibration process that regulates the representation networks. The prediction calibration process constructs prediction distributions for both current and historical experts, minimizing the probabilistic divergence between them to ensure consistency. The representation calibration process generates feature distributions from the evolved representation network for both current and prior states, aligning these distributions through a general metric function. This approach sustains stability without constraining the network's plasticity in CL.

We construct a series of experiments, and the empirical results demonstrate that the proposed approach achieves the state-of-the-art performance. The contributions of this paper are summarized as: (1) We propose a novel PTM-based approach that manages and optimizes a DRBA to learn both static and evolved representations, which provide semantically rich information for continual learning; (2) We propose a novel DARFM to automatically produce adaptive weights to regulate the importance of each representation network according to the data complexity, which enhances plasticity; (3) We propose a novel DKCM to calibrate predictions and representations for all history experts and the evolved representation network, which can maintain good stability without suffering from poor plasticity.

Related Work

Continual Learning. CL aims to adapt models to sequential tasks while mitigating catastrophic forgetting. Classical CL methods fall into several major categories. **Rehearsal-based methods** (Bang et al. 2021; Prabhu, Torr, and Dokania 2020; Arani, Sarfraz, and Zonooz 2022) store exemplar buffers or replay synthetic data to consolidate knowledge from previous tasks. While effective, they raise concerns over privacy and memory limitations. **Regularization-based methods** (Kirkpatrick et al. 2017; Chaudhry et al. 2018; Mirzadeh et al. 2020) penalize deviations in important parameters or feature representations to preserve learned knowledge, yet often falter in large-scale or cross-domain setups (Li et al. 2023). **Architecture-based methods** (Douillard et al. 2022; Xue et al. 2022; Ye and Bors 2026, 2025a,b) mitigate forgetting by allocating isolated or shared parameter modules (e.g., adapters, experts) for different tasks. Recent advancements within this category include **Prompt-based approaches** (Wang et al. 2022b,a; Jung et al. 2023; Smith et al. 2023; Wang et al. 2025), which leverage frozen pretrained transformers and optimize lightweight prompts to guide task-specific adaptation, offering memory efficiency.

Multi-Domain Continual Learning (MDCL). requires models to continuously learn from a sequence of tasks originating from distinct visual domains (Zheng et al. 2023; Yu et al. 2024; Jung et al. 2023; Menabue et al. 2024). Each task may involve a previously unseen domain, leading to severe distribution shifts and exacerbating catastrophic forgetting. Traditional continual learning settings often assume that tasks arrive sequentially within the same domain—such as class-incremental learning on CIFAR—which fails to reflect the complexities encountered in real-world applications. To address this limitation, recent research proposes Multi-domain Task-Incremental Learning (MTIL) (Zheng et al. 2023) as a more realistic and challenging benchmark. In MTIL, tasks are drawn from multiple domains that differ significantly in both semantic and visual characteristics. MTIL emphasizes two critical capabilities: domain-specific reasoning during continual adaptation and cross-task generalization, making it a rigorous platform for evaluating robustness under domain shifts.

Methodology

Problem Statement

Let us define $\{\mathcal{T}_1, \ldots, \mathcal{T}_n\}$ as a sequence of tasks in a Multi-domain Task-Incremental Learning (MTIL) setting. Each task $\mathcal{T}_i$ is associated with a training dataset $\mathcal{D}_i = \{\mathbf{x}_j, \mathbf{y}_j\}_{j=1}^{n^i}$, where $\mathbf{x}_j \in \mathcal{X}$ and $\mathbf{y}_j \in \mathcal{Y}$ denote the j-th data and the corresponding class label, respectively. The key characteristic of this setting is that the same set of classes $\mathcal{Y}$ is used across all tasks, but the data distribution changes significantly from one task to another, representing a shift in visual domains (e.g., natural images, medical images, satellite imagery).

In this scenario, the model faces two challenges: adapting to new domain-specific distributions while maintaining per-

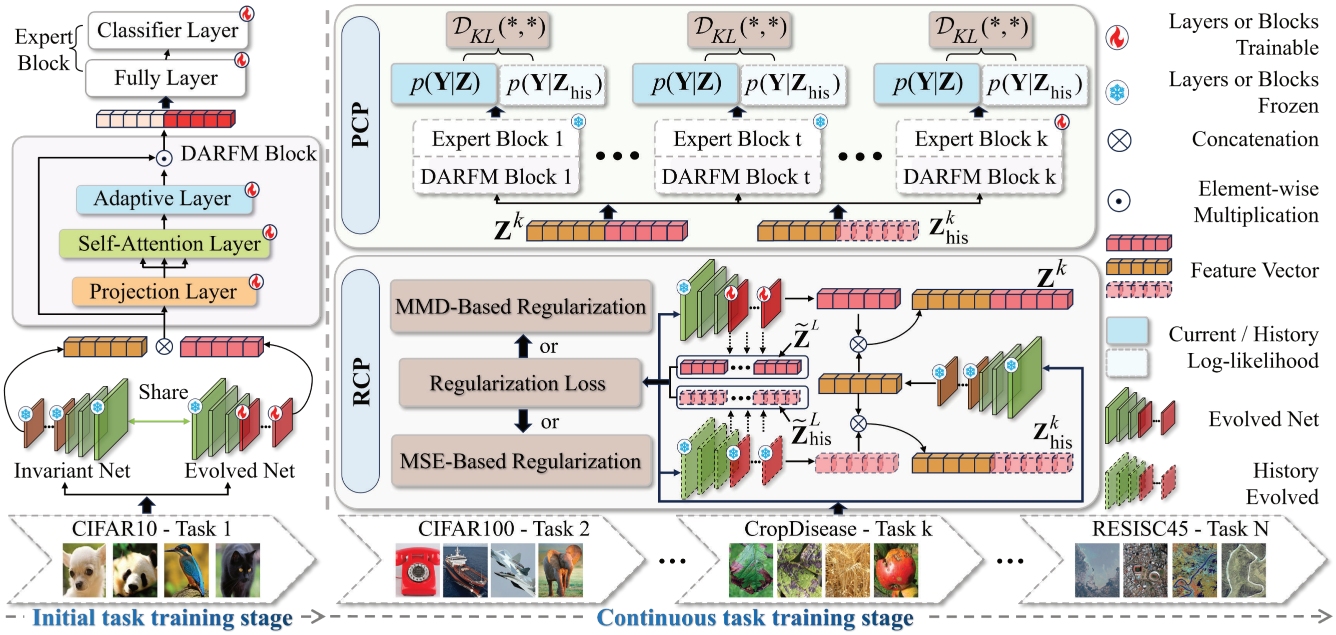


Figure 1: Overview of the Adaptive and Expandable Mixture Model (AEMM) framework. Data samples are input into the DRBA, which generates a consolidated representation through the proposed DARFM. Throughout the training phase, an innovative DKCM is introduced to modulate the optimization dynamics of the DRBA, thereby ensuring robust stability.

formance on previously learned domains. The learning goal of the model at the j-th task is to minimize the loss on the current task's data, expressed as :

$$\theta^\star = \operatorname*{argmin}_{\theta\in\Theta}\left\{\sum\nolimits_{s=1}^{|\mathcal{D}_j|} F_{\text{ce}}(\mathbf{y}_s, F_\theta(\mathbf{x}_s))\right\}, \qquad (1)$$

where θ and Θ denote the model's parameter set and its space. $|\mathcal{D}_j|$ denotes the number of samples within $\mathcal{D}_j$. $\mathbf{x}_s$ and $\mathbf{y}_s$ denote the s-th data sample and the corresponding class label. $F_\theta(\mathbf{x}_s)$ is the prediction made by the model. $F_{\text{ce}}(\cdot)$ is the cross entropy loss function. However, searching for an optimal solution $\theta^\star$ using Eq. (1) is computationally intractable because we cannot access data samples from all previous tasks simultaneously.

Dynamic Expansion Architecture

Most existing dynamic expansion frameworks usually adopt a single pre-trained backbone and dynamically create additional parameters to adapt to new tasks while freezing the backbone's parameters to relieve network forgetting (McDonnell et al. 2023). However, such a design can suffer from poor plasticity when learning several complex data domains since it only provides a static representation, and only a limited number of trainable parameters are available for the new task learning. Functional magnetic resonance imaging studies have shown that the interaction between the hippocampus and the neocortex forms a dual-system memory architecture, which manages a memory system to store slowly changing information and another memory system to preserve the evolved knowledge (McClelland, McNaughton, and O'Reilly 1995). Inspired by this biological mechanism, we aim to enhance the plasticity by proposing a novel Dual-Representation Backbone Architecture (DRBA), which manages an invariant representation network that provides less changing feature information and an evolved representation network that incrementally captures adaptive information over time, where each representation network is implemented using a pre-trained ViT. Specifically, in order to reduce computational costs and storage space, we propose to share most parameters between the invariant and evolved representation networks while only the final L layers of the evolved representation network are trainable.

Let us define $F_{\gamma^s} : \mathcal{X} \to \mathcal{Z}^S$ as a shared representation network, which receives a data sample $\mathbf{x}$ over the input space $\mathcal{X}$ and returns a feature vector $\mathbf{z}^s$ over the output space, where γ^s denotes the parameter set of the shared model. Based on F_{γ^s}, we can build an invariant representation network $F_{\gamma^a} : \mathcal{Z}^S \to \mathcal{Z}$ and an evolved representation network $F_{\gamma^e} : \mathcal{Z}^S \to \mathcal{Z}$, each of which receives a feature vector $\mathbf{z}^s$ from the shared model and returns a representation over the feature space $\mathcal{Z}$. By using the DRBA, we can form a robust representation that contains both invariant and dynamic information, expressed as :

$$F_f(\mathbf{x}) = F_{\gamma^a}(F_{\gamma^s}(\mathbf{x})) \otimes F_{\gamma^e}(F_{\gamma^s}(\mathbf{x}))\,, \qquad (2)$$

where $\otimes$ denotes an operation that concentrates two feature vectors into a single representation. In order to capture the discrimination information for a specific task $\mathcal{T}_k$, we propose to dynamically create a new lightweight expert consisting of a feature transformation module $F_{\theta^f_k} : \mathcal{Z} \to \mathcal{Z}^f$ and a linear classifier $F_{\theta^c_k} : \mathcal{Z}^f \to \mathcal{Y}$, The former receives the robust representation $\mathbf{z}' = F_f(\mathbf{x})$ and returns a task-specific representation $\mathbf{z}^f$ over the low-dimensional feature space. The latter implements the prediction process by feeding the transformed representation. which is fed into the linear classifier $F_{\theta^c_k}$ to make the prediction over the output space $\mathcal{Y}$, expressed as :

$$F_{\text{pre}}(\mathbf{x}, k) = F_{\theta^c_k}(F_{\theta^f_k}(F_{\gamma^a}(F_{\gamma^s}(\mathbf{x})) \\ \otimes F_{\gamma^e}(F_{\gamma^s}(\mathbf{x})))) \,. \tag{3}$$

Dynamic Adaptive Representation Fusion Mechanism

The feature fusion process defined in Eq. (2) simply concatenates features extracted by two representation networks and ignores the contribution of each representation for a given data $\mathbf{x}$. Such a design cannot explore the full representation capacity, resulting in suboptimal plasticity. One simple way to address this issue is to optimize a set of task-specific trainable adaptive weights to determine the importance of each representation during the training process. However, such a design can only produce the global adaptive weight configuration while ignoring the data characteristics. To deal with this issue, we propose a novel Dynamic Adaptive Representation Fusion Mechanism (DARFM) that automatically produces data-driven adaptive weights to regulate each representation during the optimization process. Specifically, the proposed DARFM introduces a data-driven adaptive weight function $F_{\omega_k} : \mathcal{Z}^{\text{aug}} \to \mathcal{Y}'$, which receives the robust feature representation $\mathbf{z}' = F_f(\mathbf{x})$ and gives an adaptive weight $\mathbf{w}$ over the space $\mathcal{Y}'$. Specifically, the adaptive weight function contains a projection matrix $\mathbf{W}^P_{\omega_k}$ to process the robust representation $\mathbf{z}'$ into a low-dimensional feature space :

$$\mathbf{z}'' = \mathbf{W}^P_{\omega_k}\mathbf{z}' \,, \tag{4}$$

where $\mathbf{z}''$ denotes a compact representation. In order to capture the correlation between features, the function F_{ω_k} contains a self-attention mechanism consisting of three trainable parameter matrices $\mathbf{W}^K_{\omega_k}$, $\mathbf{W}^Q_{\omega_k}$, and $\mathbf{W}^V_{\omega_k}$ to process $\mathbf{z}''$, resulting in :

$$\mathbf{z}^a = \text{Softmax}\,(\tilde{\mathbf{Q}}_k(\tilde{\mathbf{K}}_k)^{\text{T}}/\sqrt{d})\tilde{\mathbf{V}}_k \,, \\ \tilde{\mathbf{Q}}_k = \mathbf{W}^Q_{\omega_k}\mathbf{z}'', \tilde{\mathbf{K}}_k = \mathbf{W}^K_{\omega_k}\mathbf{z}'', \tilde{\mathbf{V}}_k = \mathbf{W}^V_{\omega_k}\mathbf{z}'' \,, \tag{5}$$

where $\sqrt{d}$ is a scaling factor and $\mathbf{z}^a$ is a transformed feature vector, which is used to predict the adaptive weight by :

$$\mathbf{w} = \mathbf{W}^w_{\omega_k}\mathbf{z}^a \,, \tag{6}$$

where $\mathbf{W}^w_{\omega_k}$ denotes a parameter matrix that transfers $\mathbf{z}^a$ to the adaptive weight $\mathbf{w} \in \mathbb{R}^2$. As a result, the representation of each backbone is regulated by the adaptive weight and the prediction process of the k-th expert is expressed as :

$$F'_{\text{pre}}(\mathbf{x}, k) = F_{\theta^c_k}(F_{\theta^f_k}(\mathbf{w}[0]F_{\gamma^a}(F_{\gamma^s}(\mathbf{x})) \\ \otimes \mathbf{w}[1]F_{\gamma^e}(F_{\gamma^s}(\mathbf{x})))) \,, \tag{7}$$

where $\mathbf{w}[0]$ and $\mathbf{w}[1]$ denote the adaptive weights for the representations extracted by the invariant and evolved representation network, respectively. Compared to the prediction process defined in Eq. (3), Eq. (7) can provide more robust performance since it adaptively adjusts the contribution of each representation according to the data characteristics and complexity.

Dynamic Knowledge Calibration Mechanism

Most existing PTM-based methods usually fix the parameters of the whole PTM architecture, which can avoid network forgetting but suffer from poor plasticity when the new task contains data samples from unknown data domains. This issue is usually caused by the lack of a considerable number of trainable parameters when learning new tasks. Optimizing the parameters of the evolved backbone can enhance plasticity but would lead to poor stability. To address this issue, we propose a novel Dynamic Knowledge Calibration Mechanism (DKCM) that aims to calibrate the knowledge preserved between the current active and the previously learned evolved representation network, respectively. Specifically, the proposed DKCM achieves this goal through two calibration optimization processes: the prediction calibration process and the representation calibration process, summarized in the following.

The Prediction Calibration Process (PCP). This process considers that each previously learned expert should maintain similar knowledge when changing the parameters of the evolved representation network. To achieve this goal, we first define $F_{\tilde{\gamma}^e}$ as an auxiliary representation network that preserves the knowledge extracted by the evolved representation network trained on the previous task learning. At a certain task learning ($\mathcal{T}_k$), we produce two sets of the current active and the previously learned representations, respectively, expressed as :

$$\mathbf{Z}^k = \{\mathbf{z}_c \,|\, \mathbf{z}_c = \mathbf{w}[0]F_{\gamma^a}(F_{\gamma^s}(\mathbf{x}_c)) \otimes \mathbf{w}[1]F_{\gamma^e}(F_{\gamma^s}(\mathbf{x}_c)), \\ c = 1, \cdots, |\mathbf{X}|\} \,, \tag{8}$$

$$\mathbf{Z}^k_{\text{his}} = \{\mathbf{z}_c \,|\, \mathbf{z}_c = \mathbf{w}[0]F_{\gamma^a}(F_{\gamma^s}(\mathbf{x}_c)) \otimes \mathbf{w}[1]F_{\tilde{\gamma}^e}(F_{\gamma^s}(\mathbf{x}_c)), \\ c = 1, \cdots, |\mathbf{X}|\} \,,$$

where the superscript k in $\mathbf{Z}^k$ denotes that the feature representations is extracted by the evolved representation network at the k-th task learning. $\mathbf{X}$ is a data batch randomly collected from the k-th task learning while $|\mathbf{X}|$ represents the batch size. Based on $\mathbf{Z}^k$ and $\mathbf{Z}^k_{\text{his}}$, each history ($\mathcal{E}_i$) can produce the task-specific representations, expressed as :

$$\mathbf{Z}^{k,i} = \{\mathbf{z}'_c \,|\, \mathbf{z}'_c = F_{\theta^f_i}(\mathbf{z}_c), \mathbf{z}_c \in \mathbf{Z}^k, c = 1, \cdots, |\mathbf{Z^k}|\} \,, \\ \mathbf{Z}^{k,i}_{\text{his}} = \{\mathbf{z}'_c \,|\, \mathbf{z}'_c = F_{\theta^f_i}(\mathbf{z}_c), \mathbf{z}_c \in \mathbf{Z}^k_{\text{his}}, \\ c = 1, \cdots, |\mathbf{Z}^{\mathbf{k}}_{\text{his}}|\} \,, \tag{9}$$

where the superscripts k and i denote that the feature vectors are extracted by the i-th expert at the k-th task learning. As a result, we can form the current active prediction distribution $p(\mathbf{Y} \,|\, \mathbf{Z}^{k,i})$ and the previously learned prediction distribution $p(\mathbf{Y} \,|\, \mathbf{Z}^{k,i}_{\text{his}})$ for the i-th expert. To calibrate the predictions across all previously learned experts, we introduce a regularization loss function defined as :

$$\mathcal{L}_p = \sum\nolimits_{i=1}^{k-1} \Big\{ D_{\text{KL}}(p(\mathbf{Y} \,|\, \mathbf{Z}^{k,i}), p(\mathbf{Y} \,|\, \mathbf{Z}^{k,i}_{\text{his}})) \Big\} \,, \tag{10}$$

where $D_{\text{KL}}(\cdot,\cdot)$ denotes the Kullback–Leibler (KL) divergence.

Method	**(a) Average Accuracy (↑)** **C10**	**C100**	**TIN**	**IN-R**	**CD**	**CUB**	**Resisc45**	**Total Avg (↑)**
DER++ (Buzzega et al. 2020)	94.62±0.01	84.28±0.01	73.94±0.21	59.32±0.13	66.38±0.01	58.92±0.79	58.98±0.58	70.92±0.04
DER++(Re) (Wang et al. 2024)	94.62±0.01	84.18±0.01	73.83±0.07	61.05±0.01	67.99±0.01	60.34±0.22	60.14±0.01	71.74±0.02
CLS-ER (Arani, Sarfraz, and Zonooz 2022)	96.92±0.11	88.84±0.57	66.69±0.09	72.31±0.63	82.21±1.18	80.56±0.23	79.50±0.30	81.00±0.01
L2P (Wang et al. 2022b)	96.09±0.02	82.82±0.04	47.52±0.09	41.53±0.01	45.51±0.29	51.38±0.01	47.87±0.01	58.96±0.03
Dualprompt (Wang et al. 2022a)	**97.34±0.01**	87.18±0.02	53.33±0.04	47.64±0.51	34.07±0.12	58.38±0.04	37.50±0.07	59.35±0.05
CODAPrompt (Smith et al. 2023)	94.77±0.04	62.26±2.44	36.17±1.99	15.92±10.56	25.81±5.30	19.15±5.60	18.85±0.13	38.99±3.01
DAP (Jung et al. 2023)	97.16±0.05	79.46±0.86	51.20±1.05	39.88±1.78	43.38±2.00	49.58±1.04	41.70±0.01	57.48±0.05
Ranpac (McDonnell et al. 2023)	91.29±0.01	77.18±0.01	73.25±0.02	68.69±0.02	73.81±0.03	72.23±0.01	73.10±0.01	75.65±0.02
SLCA (Zhang et al. 2023)	92.45±0.19	62.24±0.88	37.91±0.45	26.20±0.40	29.69±0.13	36.59±0.13	27.44±0.46	44.65±0.15
SEMA (Wang et al. 2025)	76.94±1.92	27.88±3.10	32.59±0.85	30.50±0.10	40.52±2.25	44.34±0.75	44.59±0.12	42.48±1.27
AEMM (mmd)	96.06±0.01	89.59±0.01	86.26±0.01	83.69±0.10	86.25±0.02	**85.25±0.01**	**86.07±0.01**	87.60±0.01
AEMM (mse)	96.38±0.02	**90.04±0.06**	**86.70±0.16**	**83.71±0.13**	**86.50±0.13**	85.11±0.06	85.92±0.12	**87.77±0.05**
Method	**(b) Forgetting Measure (↓)**							**Total Avg (↓)**
DER++	-	1.55±0.11	10.43±0.82	5.89±0.06	4.84±0.02	3.92±0.15	6.95±0.16	4.80±0.01
DER++(Re)	-	1.73±0.03	10.05±0.07	4.06±0.02	3.14±0.07	2.53±0.03	5.93±0.11	3.92±0.01
CLS-ER	-	4.15±1.07	31.54±0.16	15.89±0.75	6.10±1.52	4.49±0.25	6.92±0.37	9.87±0.04
L2P	-	13.28±0.37	58.64±0.32	49.67±0.01	46.03±0.68	35.18±0.17	38.49±0.16	34.47±0.16
Dualprompt	-	11.48±0.37	55.27±0.09	46.93±1.03	64.68±0.25	30.82±0.11	54.30±0.11	37.64±0.07
CODAPrompt	-	49.97±4.98	72.94±1.52	71.32±7.05	60.74±0.73	54.88±2.96	54.79±2.51	52.09±0.68
DAP	-	24.61±3.80	56.54±2.49	57.83±3.74	53.45±3.42	41.72±2.80	50.25±0.14	40.63±0.10
Ranpac	-	1.50±0.01	4.12±0.03	3.17±0.02	3.12±0.01	3.00±0.01	3.62±0.02	2.65±0.01
SLCA	-	44.31±1.53	66.10±0.55	69.40±1.24	65.89±0.42	54.31±0.38	65.32±0.79	52.19±0.55
SEMA	-	57.61±5.10	46.27±0.59	44.00±2.53	37.18±0.96	31.93±0.44	33.06±1.37	35.72±0.33
AEMM (mmd)	-	**0.00±0.00**	0.58±2.25	**0.56±0.02**	0.79±0.03	**0.46±0.01**	**0.55±0.18**	**0.42±0.07**
AEMM (mse)	-	0.17±0.07	**0.53±0.13**	0.74±0.10	**0.71±0.03**	0.54±0.06	0.58±0.05	0.47±0.07
Method	**(c) Learning Accuracy (↑)**							**Total Avg (↑)**
DER++	94.62±0.01	85.06±0.01	80.89±0.02	63.74±0.03	70.26±0.02	62.18±0.31	64.93±0.17	74.53±0.04
DER++(Re)	94.62±0.01	85.05±0.03	80.53±0.21	64.07±0.02	70.47±0.01	62.42±0.12	65.20±0.12	74.62±0.01
CLS-ER	96.92±0.11	90.91±0.03	87.71±0.02	84.22±0.06	87.09±0.03	84.25±0.01	85.39±0.01	88.07±0.01
L2P	96.09±0.03	89.47±0.01	86.62±0.01	78.79±0.02	82.34±0.01	80.69±0.05	80.86±0.05	84.98±0.02
Dualprompt	**97.34±0.01**	**92.23±0.02**	**90.07±0.01**	82.84±0.02	85.81±0.01	84.07±0.03	84.05±0.02	88.10±0.01
CODAPrompt	94.77±0.06	87.24±0.07	84.79±1.37	69.41±7.43	74.40±6.65	64.89±4.46	65.81±3.24	77.33±3.31
DAP	97.16±0.05	91.76±0.01	88.89±0.01	83.26±0.02	86.14±0.01	84.35±0.14	84.77±0.07	88.05±0.01
Ranpac	91.29±0.01	77.93±0.01	76.00±0.02	71.07±0.02	76.30±0.03	74.72±0.01	76.21±0.01	77.65±0.01
SLCA	92.45±0.19	84.40±0.12	81.97±0.09	78.25±0.53	82.40±0.46	81.85±0.19	83.43±0.23	83.54±0.23
SEMA	76.94±1.92	56.69±0.55	63.43±0.46	63.50±1.99	70.26±1.49	70.94±1.11	72.93±1.06	67.81±1.22
AEMM (mmd)	96.06±0.01	89.52±0.01	86.60±0.01	84.07±0.02	86.85±0.01	85.54±0.01	**86.44±0.02**	87.87±0.01
AEMM (mse)	96.38±0.02	90.12±0.02	87.05±0.07	**84.28±0.06**	**87.10±0.10**	**85.54±0.10**	86.38±0.13	**88.12±0.01**

Table 1: Comparison with state-of-the-art (SOTA) methods on the MTIL benchmark in terms of Average Accuracy, Forgetting Measure, Learning Accuracy, and Total Avg (%). Best and second-best results are bold and underlined.

The Representation Calibration Process (RCP). The loss function defined in Eq. (10) can employ the supervised signals from each history expert to regulate the optimization process of the evolved representation network. In order to further improve the stability, the representation calibration process aims to maintain the invariant representation information when changing the parameters of the evolved representation network. Let us define a feature extraction function :

$$F_{\rm t}(F_{\gamma^e}, \mathbf{x}, i) = \begin{cases} F_{\gamma^e_1}(F_{\gamma^s}(\mathbf{x})) & i = 1 \\ F_{\gamma^e_i}(\cdots F_{\gamma^e_1}(F_{\gamma^s}(\mathbf{x}))) & 2 \leq i \leq L\,, \end{cases} \tag{11}$$

where $F_{\gamma^e_i}(\cdot)$ denotes the feature vector extracted by the i-th trainable representation layer of the evolved representation network and L is the total number of trainable representation layers. By using Eq. (11), we can form two sets of feature vectors, expressed as :

$$\begin{aligned} \tilde{\mathbf{Z}}^i &= \{\mathbf{z}_c \,|\, \mathbf{z}_c = F_t(F_{\gamma^e}, \mathbf{x}_c, i), c = 1, \cdots, |\mathbf{X}|\}\,, \\ \tilde{\mathbf{Z}}^i_{\rm his} &= \{\mathbf{z}_c \,|\, \mathbf{z}_c = F_t(F_{\tilde{\gamma}^e}, \mathbf{x}_c, i), c = 1, \cdots, |\mathbf{X}|\}\,, \end{aligned} \tag{12}$$

where $\mathbf{X}$ is data batch randomly collected from the k-th task learning. $\tilde{\mathbf{Z}}^i$ and $\tilde{\mathbf{Z}}^i_{\rm his}$ denote the feature sets extracted using the evolved and auxiliary representation networks, respectively. Then, we can define a general regularization loss function to regulate the optimization process of the evolved representation network at $\mathcal{T}_k$, expressed as :

$$\mathcal{L}_r = \frac{1}{L} \sum\nolimits_{i=1}^{L} \left\{ F_{\rm measure}(\tilde{\mathbf{Z}}^i, \tilde{\mathbf{Z}}^i_{\rm his}) \right\}, \tag{13}$$

where $F_{\rm measure}(\cdot,\cdot)$ denotes a general measure function which can be implemented using the statistical distance. In this paper, we consider implementing $F_{\rm measure}(\cdot,\cdot)$ using the Maximum Mean Discrepancy (MMD) (Tolstikhin, Sriperumbudur, and Schölkopf 2016) and Mean Squared Error (MSE), respectively.

MMD-based regularization : Let $\tilde{\mathcal{Z}}^i$ be a toplogical space. We can define two Borel probability measures $p(\tilde{\mathbf{Z}}^i)$ and $p(\tilde{\mathbf{Z}}^i_{\rm his})$ using two feature sets $\tilde{\mathbf{Z}}^i$ and $\tilde{\mathbf{Z}}^i_{\rm his}$, respectively. Let $\mathbf{z}^i$ and $\mathbf{z}^i_{\rm his}$ be the random variables over $p(\tilde{\mathbf{Z}}^i)$ and $p(\tilde{\mathbf{Z}}^i_{\rm his})$, respectively. We denote by $f \in \mathcal{F}$ a function mapping $\mathcal{X} \rightarrow \mathbf{R}$, where $\mathcal{F}$ represents a class of real-valued functions. We define the MMD between $p(\tilde{\mathbf{Z}}^i)$ and $p(\tilde{\mathbf{Z}}^i_{\rm his})$ as (Tolstikhin, Sriperumbudur, and Schölkopf 2016):

$$\begin{aligned} &\mathcal{L}_{\rm M}(p(\tilde{\mathbf{Z}}^i), p(\tilde{\mathbf{Z}}^i_{\rm his})) \triangleq \\ &\sup_{f \in \mathcal{F}} \left(\mathbb{E}_{\mathbf{z}^i \sim p(\tilde{\mathbf{Z}}^i)} \left[f(\mathbf{z}^i) \right] - \mathbb{E}_{\mathbf{z}^i_{\rm his} \sim p(\tilde{\mathbf{Z}}^i_{\rm his})} \left[f(\mathbf{z}^i_{\rm his}) \right] \right) . \end{aligned} \tag{14}$$

Methods	TIN-CD-RESISC45		RESISC45-TIN-CD		CD-RESISC45-TIN		Average	
	Average	Last	Average	Last	Average	Last	Average	Last
DER++	83.57±0.17	85.83±0.01	84.15±0.22	86.22±0.13	91.54±0.03	85.59±0.36	86.42±3.63	85.88±0.26
DER++(Re)	83.36±0.05	85.55±0.06	84.12±0.26	86.08±0.42	91.49±0.10	85.60±0.21	86.32±3.67	85.74±0.24
CLS-ER	85.76±0.12	88.31±0.36	87.90±0.35	87.64±0.44	93.66±0.24	88.45±0.23	89.11±2.88	88.14±0.37
L2P	79.17±0.41	74.05±1.16	68.10±0.10	62.71±0.05	81.77±0.01	64.83±0.69	76.35±5.93	67.20±4.92
Dualprompt	69.09±0.14	65.89±0.35	64.64±0.68	44.81±0.70	84.64±0.15	68.67±0.17	72.79±8.58	59.79±10.65
CODAPrompt	80.62±0.30	76.74±0.21	64.99±0.23	60.53±0.46	72.76±0.45	35.52±4.91	72.79±6.36	57.60±16.96
Dap	74.79±0.87	62.48±2.91	67.80±0.37	55.21±2.67	76.13±0.04	51.30±0.72	72.91±3.65	56.33±4.63
Ranpac	82.16±0.10	85.86±0.05	85.76±0.05	86.06±0.04	92.88±0.02	85.84±0.01	86.93±4.45	85.92±0.10
SLCA	77.60±1.13	63.87±2.86	76.52±0.30	66.18±0.57	83.89±0.94	60.20±2.25	79.34±3.25	63.42±2.46
SEMA	83.60±0.49	76.70±2.04	70.14±0.42	65.23±0.37	70.29±0.61	41.90±0.55	74.68±6.31	61.28±14.47
AEMM (mmd)	**87.06**±**0.10**	89.85±0.13	89.23±0.31	**90.14**±**0.12**	**94.41**±**0.03**	89.68±0.02	**90.23**±**3.08**	89.89±0.19
AEMM (mse)	86.91±0.20	**90.17**±**0.04**	**89.24**±**0.23**	90.03±0.16	94.29±0.01	**89.88**±**0.14**	90.15±3.08	**90.03**±**0.11**
w/o PCP	86.35±0.01	89.52±0.10	88.85±0.08	89.80±0.14	93.74±0.04	89.36±0.04	89.65±3.07	89.56±0.18
w/o RCP	86.19±0.09	89.43±0.08	88.93±0.59	89.67±0.37	94.18±0.25	89.58±0.32	89.77±3.31	89.56±0.10

Table 2: Comparison of Average and Last accuracy (%) under three task permutations in the MTIL setting: TIN → CD → RESISC45 (TCR), RESISC45 → TIN → CD (RTC), and CD → RESISC45 → TIN (CRT).

where sup denotes the least upper bound of a set of numbers. If two distributions $p(\tilde{\mathbf{Z}}^i)$ and $p(\tilde{\mathbf{Z}}^i_{\text{his}})$ are equal, we have $\mathcal{L}_{\text{M}}(p(\tilde{\mathbf{Z}}^i), p(\tilde{\mathbf{Z}}^i_{\text{his}})) = 0$. The function class $\mathcal{F}$ is considered as a unit ball in an RKHS with a positive definite kernel $k(\mathbf{x}, \mathbf{x}')$. Calculating Eq. (14) is usually computationally intractable. In practice, the MMD is estimated on the embedding space (Li et al. 2017), expressed as :

$$\mathcal{L}^2_{\text{M}}(p(\tilde{\mathbf{Z}}^i), p(\tilde{\mathbf{Z}}^i_{\text{his}})) = ||\boldsymbol{\mu}_{p(\tilde{\mathbf{Z}}^i)} - \boldsymbol{\mu}_{p(\tilde{\mathbf{Z}}^i_{\text{his}})}|| \,, \tag{15}$$

where $\boldsymbol{\mu}_{p(\tilde{\mathbf{Z}}^i)}$ and $\boldsymbol{\mu}_{p(\tilde{\mathbf{Z}}^i_{\text{his}})}$ denote the mean embedding of $p(\tilde{\mathbf{Z}}^i)$ and $p(\tilde{\mathbf{Z}}^i_{\text{his}})$, respectively. $||\cdot||$ denotes the Euclidean distance. $\boldsymbol{\mu}_{p(\tilde{\mathbf{Z}}^i)}$ is defined as $\boldsymbol{\mu}_{p(\tilde{\mathbf{Z}}^i)} = \int k(\mathbf{z}^i, \cdot) \frac{\partial P(\tilde{\mathbf{Z}}^i)}{\partial \mathbf{z}^i} d\mathbf{z}^i$. $P(\tilde{\mathbf{Z}}^i)$ denotes the probability density function for $p(\tilde{\mathbf{Z}}^i)$. $\boldsymbol{\mu}_{p(\tilde{\mathbf{Z}}^i)}$ also satisfies $\mathbb{E}[f(\mathbf{z}^i)] = \langle f, \boldsymbol{\mu}_{p(\tilde{\mathbf{Z}}^i)} \rangle_{\mathcal{H}}$. $\langle f \,, \cdot \rangle_{\mathcal{H}}$ denotes the inner product. Since RKHS has the reproducing property $f \in \mathcal{F}, f(\mathbf{z}^i) = \langle f \,, k(\mathbf{z}^i, \cdot) \rangle_{\mathcal{H}}$, Eq. (15) can be calculated using the kernel functions:

$$\begin{aligned}\mathcal{L}^2_{\text{M}}(p(\tilde{\mathbf{Z}}^i), p(\tilde{\mathbf{Z}}^i_{\text{his}})) = {} & \mathbb{E}_{\mathbf{z}^i, \hat{\mathbf{z}}^i \sim p(\tilde{\mathbf{Z}}^i)}[k(\mathbf{z}^i, \hat{\mathbf{z}}^i)] \\ & - 2\mathbb{E}_{\mathbf{z}^i \sim p(\tilde{\mathbf{Z}}^i), \mathbf{z}^i_{\text{his}} \sim p(\tilde{\mathbf{Z}}^i_{\text{his}})}[k(\mathbf{z}^i, \mathbf{z}^i_{\text{his}})] \\ & + \mathbb{E}_{\mathbf{z}^i_{\text{his}}, \hat{\mathbf{z}}^i_{\text{his}} \sim p(\tilde{\mathbf{Z}}^i_{\text{his}})}[k(\mathbf{z}^i_{\text{his}}, \hat{\mathbf{z}}^i_{\text{his}})] \,,\end{aligned} \tag{16}$$

where $\hat{\mathbf{z}}^i_{\text{his}}$ and $\hat{\mathbf{z}}^i$ are independent copies of $\mathbf{z}^i_{\text{his}}$ and $\mathbf{z}^i$, respectively. In practice, we employ the same number of samples from $p(\tilde{\mathbf{Z}}^i)$ and $p(\tilde{\mathbf{Z}}^i_{\text{his}})$ ($N_P = N_Q$), where N_Q and N_P are the number of samples for $p(\tilde{\mathbf{Z}}^i_{\text{his}})$ and $p(\tilde{\mathbf{Z}}^i)$, respectively. Then Eq. (16) can be estimated using an unbiased empirical estimate, defined as:

$$\mathcal{L}^e_{\text{M}}(p(\tilde{\mathbf{Z}}^i), p(\tilde{\mathbf{Z}}^i_{\text{his}})) = \frac{1}{N_P(N_p - 1)} \sum_{a \neq b}^{N_P} \Big\{ h(a, b) \Big\}, \tag{17}$$

where $h(a, b) = k(\mathbf{z}^i(a), \mathbf{z}^i(b)) + k(\mathbf{z}^i_{\text{his}}(a), \mathbf{z}^i_{\text{his}}(b)) - k(\mathbf{z}^i(a), \mathbf{z}^i_{\text{his}}(b)) - k(\mathbf{z}^i(b), \mathbf{z}^i_{\text{his}}(a))$. We implement $F_{\text{measure}}(\cdot, \cdot)$ using $\mathcal{L}^e_{\text{M}}(\cdot, \cdot)$ and Eq. (13) is redefined by :

$$\mathcal{L}_{\text{MMD}} = \frac{1}{L} \sum\nolimits_{i=1}^{L} \Big\{ \mathcal{L}^e_{\text{M}}(p(\tilde{\mathbf{Z}}^i), p(\tilde{\mathbf{Z}}^i_{\text{his}})) \Big\} \,. \tag{18}$$

The MSE-based regularization : Due to the space limitation, we provide the implementation details about the MSE-based regularization in **Appendix-A** from Supplementary Material (SM). The training process of the proposed AEMM is shown in Figure 1 and the pseudocode is provided in **Algorithm 1** of **Appendix-A** from SM.

Experiment

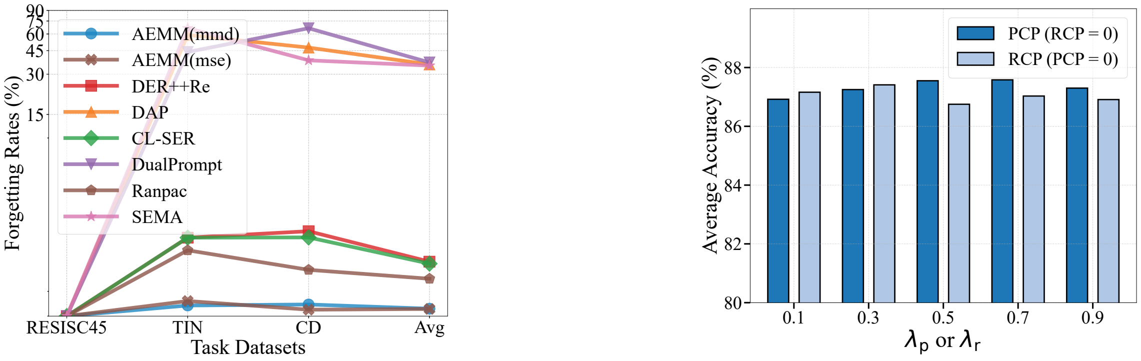
Experiment Setup

We provide the detailed experimental settings in **Appendix B** from SM.

Metrics For the evaluation, each experiment is repeated three times, and the average result with standard errors are reported. We use four widely adopted metrics in CL: Average Accuracy (Average) ↑ (Lopez-Paz et al. 2017), which measures the mean test accuracy over the first k tasks after completing the k-th task; Forgetting Measure ↓ (Chaudhry et al. 2018), which quantifies the extent to which the model forgets previous tasks; and Learning Accuracy ↑ (Riemer et al. 2019), which assesses the model's ability to quickly adapt to new tasks. Additionally, we report the Last Accuracy (Last) (Yu et al. 2024), which evaluates the model's performance on the most recently learned task.

To directly evaluate the model's overall robustness and cross-domain generalization in MTIL, we also report the Total Average (Total Avg ↑), which summarizes performance across all tasks with varying degrees of domain shift.

Datasets: The datasets in our experiments are grouped into four domains: Natural Domains (CIFAR-10 (C10) (Krizhevsky and Hinton 2009), CIFAR-100 (C100) (Krizhevsky and Hinton 2009), Tiny ImageNet (TIN) (Le and Yang 2015) and ImageNet-R (IN-R) (Hendrycks et al. 2021)) for general classification; Medical Domain (CropDiseases (CD) (Mohanty, Hughes, and Salathé 2016)) for plant disease detection; Fine-Grained Domain (CUB-200 (CUB) (Wah et al. 2011)) for fine-grained recognition; and Aerial Domain (RESISC45 (Cheng, Han, and Lu 2017)) for land cover classification using satellite imagery.



(a) Forgetting curves of RTC.

(b) Hyperparameter sensitivity analysis.

Figure 2: (a) Comparison of the forgetting curves after learning the RTC sequence. (b) Average accuracy under cross-domain settings on all seven datasets in Table 1, showing the sensitivity of PCP and RCP to different control coefficients.

Experimental Results

We evaluate two variants of our framework: AEMM (mmd) and AEMM (mse), which differ in the distance metric used in the RCP. Both are evaluated on the MTIL benchmark (Zheng et al. 2023), covering seven different visual domains.

Average Accuracy and Total Performance. As shown in Table 1(a), AEMM (mse) achieves the highest Average Accuracy on five out of seven datasets, including the most challenging ones: TIN (86.70%), IN-R (83.71%), CD (86.50%), and also ranks first in overall performance with a Total Average of 87.77%. AEMM (mmd) closely follows with 87.60%. Both surpass the strongest baseline CLS-ER (81.00%), and significantly outperform prompt-based methods such as DualPrompt (59.35%) and DAP (57.48%), especially under large domain shifts.

Learning Accuracy and Forgetting Measure. Table 1(b) shows that AEMM (mmd) achieves the least forgetting rate with a Total Average of just 0.42%, outperforming DER++(Re) (3.92%) and Ranpac (2.65%) by large margins. AEMM (mse) also maintains low forgetting (0.47%). This indicates strong stability without requiring rehearsal memory or domain-specific prompts. In terms of learning accuracy (Table 1(c)), AEMM (mse) achieves the best overall result of 88.12%, surpassing prompt-based competitors like DAP (88.05%) and DualPrompt (88.10%) while maintaining minimal forgetting. Notably, AEMM (mse) achieves leading task-specific performances on domains like IN-R (84.28%), CD (87.10%), and CUB (85.54%), demonstrating effective knowledge transfer and adaptability via its DARFM and DKCM modules.

Ablation Study

We provide more ablation study results in the **Appendix D** from SM.

Effect of Task Order and Model Components. To assess model robustness under domain heterogeneity and task order variation, we construct three task sequences using TIN, CD, and RESISC45, which differ in visual style and semantic granularity. As shown in Table 2, both AEMM (mmd) and AEMM (mse) consistently outperform all baselines across different orders. Prompt-based methods (e.g., DualPrompt, L2P) suffer significant drops in Last Accuracy, while rehearsal-based methods like DER++ show unstable trends. Figure 2(a) further highlights this difference: AEMM (mmd) maintains near-zero forgetting on the RTC sequence, whereas CODA-Prompt and DualPrompt degrade rapidly. Dynamic methods like RanPac reduce forgetting but still lag behind. We also ablate the Representation Calibration Process (RCP) and Prediction Calibration Process (PCP) in AEMM (mse) to assess their impact. Removing RCP leads to a sharper accuracy drop, especially under domain shifts, while disabling PCP causes moderate but consistent degradation. Both modules are essential to ensuring generalization and stability in continual learning.

Hyperparameter Sensitivity of Calibration Mechanisms. To assess the robustness of the proposed calibration modules, we perform a sensitivity analysis on the control coefficients of the PCP and RCP, independently disabling the other module during tuning. As shown in Figure 2(b), the performance of PCP improves steadily with larger coefficients, peaking at 87.58% before stabilizing, indicating that stronger prediction-level correction enhances generalization until saturation. In contrast, RCP shows a mild gain at 0.3 but drops beyond, suggesting that over-regularizing representation flexibility may limit cross-domain adaptability. Overall, the model exhibits stable behavior across a wide range of settings, validating the design robustness of both modules.

Conclusion

We propose a novel continual learning framework leveraging PTMs with a DRBA, an AEMM, and dynamic mechanisms for fusion (DARFM) and calibration (DKCM). Our method balances plasticity and stability by adaptively fusing invariant and evolving features while aligning current and past knowledge. Experiments on MTIL benchmarks show that our approach achieves state-of-the-art performance in average accuracy and forgetting mitigation, validating its effectiveness across diverse task sequences.

Acknowledgements

This work was supported by the Sichuan Provincial Natural Science Foundation (Grant No. 2025ZNSFSC0510); the National Natural Science Foundation of China (Grants Nos. 62506067 and 62306066); the Fundamental Research Funds for the Central Universities (Grant Nos. ZYGX2025XJ024 and ZYGX2025XJ025); the Postdoctoral Fellowship Program (Grade C) of the China Postdoctoral Science Foundation (Grant No. GZC20251053); and Huawei Funding (Project ID H04W241592).

References

Arani, E.; Sarfraz, F.; and Zonooz, B. 2022. Learning Fast, Learning Slow: A General Continual Learning Method based on Complementary Learning System. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net.

Bang, J.; Kim, H.; Yoo, Y.; Ha, J.-W.; and Choi, J. 2021. Rainbow Memory: Continual Learning with a Memory of Diverse Samples. In *Proc. of IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 8218–8227.

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

Chaudhry, A.; Dokania, P. K.; Ajanthan, T.; and Torr, P. H. 2018. Riemannian walk for incremental learning: Understanding forgetting and intransigence. In *Proceedings of the European conference on computer vision (ECCV)*, 532–547.

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

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

Douillard, A.; Ramé, A.; Couairon, G.; and Cord, M. 2022. Dytox: Transformers for continual learning with dynamic token expansion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 9285–9295.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep residual learning for image recognition. In *Proc. of IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 770–778.

Hendrycks, D.; Basart, S.; Mu, N.; Kadavath, S.; Wang, F.; Dorundo, E.; Desai, R.; Zhu, T.; Parajuli, S.; Guo, M.; Song, D.; Steinhardt, J.; and Gilmer, J. 2021. The Many Faces of Robustness: A Critical Analysis of Out-of-Distribution Generalization. *ICCV*.

Hinton, G. E.; Osindero, S.; and Teh, Y. W. 2006. A fast learning algorithm for deep belief nets. *Neural Computation*, 18(7): 1527–1554.

Jung, D.; Han, D.; Bang, J.; and Song, H. 2023. Generating Instance-Level Prompts for Rehearsal-Free Continual Learning. In *Proc. of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 11847–11857.

Kemker, R.; McClure, M.; Abitino, A.; Hayes, T.; and Kanan, C. 2018. Measuring catastrophic forgetting in neural networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32.

Kirkpatrick, J.; Pascanu, R.; Rabinowitz, N.; Veness, J.; Desjardins, G.; Rusu, A. A.; Milan, K.; Quan, J.; Ramalho, T.; Grabska-Barwinska, A.; Hassabis, D.; Clopath, C.; Kumaran, D.; and Hadsell, R. 2017. Overcoming catastrophic forgetting in neural networks. *Proc. of the National Academy of Sciences (PNAS)*, 114(13): 3521–3526.

Krizhevsky, A.; and Hinton, G. 2009. Learning multiple layers of features from tiny images. Technical report, Univ. of Toronto.

Le, Y.; and Yang, X. 2015. Tiny imageNet visual recognition challenge. Technical report, Univ. of Stanford.

Li, C.-L.; Chang, W.-C.; Cheng, Y.; Yang, Y.; and Póczos, B. 2017. Mmd gan: Towards deeper understanding of moment matching network. *Advances in neural information processing systems*, 30.

Li, X.; Wang, S.; Sun, J.; and Xu, Z. 2023. Variational data-free knowledge distillation for continual learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(10): 12618–12634.

Lopez-Paz, D.; Nishihara, R.; Chintala, S.; Scholkopf, B.; and Bottou, L. 2017. Discovering causal signals in images. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 6979–6987.

Martens, J.; and Grosse, R. B. 2015. Optimizing Neural Networks with Kronecker-factored Approximate Curvature. In Bach, F. R.; and Blei, D. M., eds., *Proceedings of the 32nd International Conference on Machine Learning, ICML 2015, Lille, France, 6-11 July 2015*, volume 37 of *JMLR Workshop and Conference Proceedings*, 2408–2417. JMLR.org.

McClelland, J. L.; McNaughton, B. L.; and O'Reilly, R. C. 1995. Why there are complementary learning systems in the hippocampus and neocortex: insights from the successes and failures of connectionist models of learning and memory. *Psychological review*, 102(3): 419.

McDonnell, M. D.; Gong, D.; Parvaneh, A.; Abbasnejad, E.; and Van den Hengel, A. 2023. Ranpac: Random projections and pre-trained models for continual learning. *Advances in Neural Information Processing Systems*, 36: 12022–12053.

Menabue, M.; Frascaroli, E.; Boschini, M.; Sangineto, E.; Bonicelli, L.; Porrello, A.; and Calderara, S. 2024. Semantic residual prompts for continual learning. In *European Conference on Computer Vision*, 1–18. Springer.

Mirzadeh, S. I.; Farajtabar, M.; Pascanu, R.; and Ghasemzadeh, H. 2020. Understanding the role of training regimes in continual learning. *Advances in Neural Information Processing Systems*, 33: 7308–7320.

Mohanty, S. P.; Hughes, D. P.; and Salathé, M. 2016. Using deep learning for image-based plant disease detection. *Frontiers in plant science*, 7: 215–232.

Parisi, G. I.; Kemker, R.; Part, J. L.; Kanan, C.; and Wermter, S. 2019. Continual lifelong learning with neural networks: A review. *Neural Networks*, 113: 54–71.

Prabhu, A.; Torr, P. H.; and Dokania, P. K. 2020. Gdumb: A simple approach that questions our progress in continual learning. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part II 16*, 524–540. Springer.

Riemer, M.; Cases, I.; Ajemian, R.; Liu, M.; Rish, I.; Tu, Y.; and Tesauro, G. 2019. Learning to Learn Without Forgetting by Maximizing Transfer and Minimizing Interference. In *International Conference on Learning Representations (ICLR)*.

Smith, J. S.; Karlinsky, L.; Gutta, V.; Cascante-Bonilla, P.; Kim, D.; Arbelle, A.; Panda, R.; Feris, R.; and Kira, Z. 2023. 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*, 11909–11919.

Tolstikhin, I. O.; Sriperumbudur, B. K.; and Schölkopf, B. 2016. Minimax estimation of maximum mean discrepancy with radial kernels. *Advances in Neural Information Processing Systems*, 29: 1930–1938.

Villa, A.; Alcázar, J. L.; Alfarra, M.; Alhamoud, K.; Hurtado, J.; Heilbron, F. C.; Soto, A.; and Ghanem, B. 2023. Pivot: Prompting for video continual learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 24214–24223.

Wah, C.; Branson, S.; Welinder, P.; Perona, P.; and Belongie., S. 2011. The Caltech-UCSD Birds-200-2011 Dataset. Technical Report CNS-TR-2011-001, California Institute of Technology.

Wang, H.; Lu, H.; Yao, L.; and Gong, D. 2025. Self-expansion of pre-trained models with mixture of adapters for continual learning. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, 10087–10098.

Wang, Z.; Li, Y.; Shen, L.; and Huang, H. 2024. A Unified and General Framework for Continual Learning. *arXiv preprint*, arXiv:2403.13249.

Wang, Z.; Zhang, Z.; Ebrahimi, S.; Sun, R.; Zhang, H.; Lee, C.-Y.; Ren, X.; Su, G.; Perot, V.; Dy, J.; et al. 2022a. Dual-prompt: Complementary prompting for rehearsal-free continual learning. In *European conference on computer vision*, 631–648. Springer.

Wang, Z.; Zhang, Z.; Lee, C.-Y.; Zhang, H.; Sun, R.; Ren, X.; Su, G.; Perot, V.; Dy, J.; and Pfister, T. 2022b. Learning to prompt for continual learning. In *Proc. of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 139–149.

Xue, M.; Zhang, H.; Song, J.; and Song, M. 2022. Meta-attention for vit-backed continual learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 150–159.

Ye, F.; and Bors, A. G. 2025a. Task-Free Continual Generative Modelling Via Dynamic Teacher-Student Framework. *Expert Systems with Applications*, 129873.

Ye, F.; and Bors, A. G. 2025b. Training a Dynamic Growing Mixture Model for Lifelong Learning. *IEEE Transactions on Neural Networks and Learning Systems*.

Ye, F.; and Bors, A. G. 2026. Online task-free continual learning via Expansible Vision Transformer. *Pattern Recognition*, 169: 111730.

Yu, J.; Zhuge, Y.; Zhang, L.; Hu, P.; Wang, D.; Lu, H.; and He, Y. 2024. Boosting Continual Learning of Vision-Language Models via Mixture-of-Experts Adapters. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 23219–23230.

Zhang, G.; Wang, L.; Kang, G.; Chen, L.; and Wei, Y. 2023. Slca: Slow learner with classifier alignment for continual learning on a pre-trained model. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 19148–19158.

Zheng, Z.; Ma, M.; Wang, K.; Qin, Z.; Yue, X.; and You, Y. 2023. Preventing zero-shot transfer degradation in continual learning of vision-language models. In *Proceedings of the IEEE/CVF international conference on computer vision*, 19125–19136.