Appendix for Learning Adaptive and Expandable Mixture Model for Continual Learning

Contents

A MSE-based Regularization and Algorithm Implementation **2**

A.1 MSE-based Regularization 2

A.2 The Algorithm Implementation 2

B Additional Information for the Experiment Setting **3**

B.1 Device Configurations. 3

B.2 Datasets 3

B.3 SOTA Methods 4

B.4 Implementation 5

B.5 Evaluation Metrics 6

C Computational Cost Analysis **6**

D The Additional Ablation Studies **8**

D.1 Effectiveness of Dual-Branch Representation Architecture (DRBA) 8

D.2 Influence of Calibration Loss and Backbone Pretraining Strategy 9

D.3 Effectiveness of the Task-Adaptive Feature Fusion in DARFM 10

D.4 Analysis of RCP Variants Using Alternative Calibration Losses 10

A MSE-based Regularization and Algorithm Implementation

A.1 MSE-based Regularization

In addition to the MMD criterion, we also consider implementing $F_{\mathrm{measure}}(\cdot,\cdot)$ using the mean squared error (MSE) metric. Compared to MMD, the MSE has an easy implementation and is computed fast. According to Eq. (13), the MSE-based regularization loss function is defined by :

$$\mathcal{L}_{\mathrm{MSE}} = \frac{1}{L-1}\sum\nolimits_{i=1}^{L}\left\{\frac{1}{|\tilde{\mathbf{Z}}^i|}\sum\nolimits_{c=1}^{|\tilde{\mathbf{Z}}^i|}\left\{F_{\mathrm{mse}}(\tilde{\mathbf{Z}}^i[c],\tilde{\mathbf{Z}}^i_{\mathrm{his}}[c])\right\}\right\}, \tag{1}$$

where $\tilde{\mathbf{Z}}^i[c]$ and $\tilde{\mathbf{Z}}^i_{\mathrm{his}}[c]$ are the c-th feature from $\tilde{\mathbf{Z}}^i$ and $\tilde{\mathbf{Z}}^i_{\mathrm{his}}$, respectively. $|\tilde{\mathbf{Z}}^i|$ is the total number of features in $\tilde{\mathbf{Z}}^i$ and $F_{\mathrm{MSE}}(\tilde{\mathbf{Z}}^i[c],\tilde{\mathbf{Z}}^i_{\mathrm{his}}[c])$ denotes the MSE function, defined as :

$$F_{\mathrm{MSE}}(\tilde{\mathbf{Z}}^i[c],\tilde{\mathbf{Z}}^i_{\mathrm{his}}[c]) = \frac{1}{|\tilde{\mathbf{Z}}^i[c]|}\sum\nolimits_{t=1}^{|\tilde{\mathbf{Z}}^i[c]|}\left\{(\tilde{\mathbf{Z}}^i[c][t]-\tilde{\mathbf{Z}}^i_{\mathrm{his}}[c][t])^2\right\}, \tag{2}$$

where $\tilde{\mathbf{Z}}^i[c][t]$ and $\tilde{\mathbf{Z}}^i_{\mathrm{his}}[c][t]$ are the t-th dimension of the c-th feature from $\tilde{\mathbf{Z}}^i$ and $\tilde{\mathbf{Z}}^i_{\mathrm{his}}$, respectively. $|\tilde{\mathbf{Z}}^i[c]|$ denotes the feature dimension.

A.2 The Algorithm Implementation

We provide the detailed pseudocode in **Algorithm 1**. The process can be summarized in four main steps:

Step 1: The DARFM phase. When building a new lightweight expert module $\mathcal{E}_k$ for the k-th task learning, we also create the associated function F_{ω_k} to regulate the importance of each representation for a given sample.

Step 2: Calibrate knowledge via dual regularization. To mitigate catastrophic forgetting, we introduce two calibration losses using the auxiliary network $F_{\tilde{\gamma}^e}$. The prediction calibration loss $\mathcal{L}_p$ aligns the output distributions of all previous experts with their historical counterparts, formulated as in Eq. (10). Additionally, the representation calibration loss $\mathcal{L}_r$ regularizes the intermediate features of the trainable layers in the evolved network, computed layer-wise using MMD or MSE.

Step 3. The model updating. The final optimization function combines the cross-entropy loss with the calibration terms :

$$\mathcal{L}_{\mathrm{total}} = \mathcal{L}_{ce} + \lambda_p\mathcal{L}_p + \lambda_r\mathcal{L}_r\ , \tag{3}$$

where λ_p and λ_r are the hyperparameters that regulate the importance of the prediction calibration and representation calibration terms, respectively. $\mathcal{L}_{ce}$ represents the cross-entropy loss. During the optimization process, we only update the last L layers of F_{γ^e}, the current expert $\mathcal{E}_k$, and the current active DARFM F_{ω_k} while freezing all other components.

Algorithm 1 The training process of the proposed framework.

```
Require: A sequence of tasks $\{\mathcal{T}_1, \dots, \mathcal{T}_N\}$;
    Pre-trained Vision Transformer (ViT);
    Number of trainable layers $L$ in evolved network;
    Calibration hyperparameters $\lambda_p$, $\lambda_r$.
Initialize shared representation network $F_{\gamma^s}$, invariant network $F_{\gamma^a}$, and initial evolved network $F_{\gamma^e}$ from pre-trained ViT. Freeze $F_{\gamma^s}$ and $F_{\gamma^a}$.
for task $k = 1$ to $N$ do
    Instantiate a new lightweight expert $\mathcal{E}_k = \{F_{\theta_k^f}, F_{\theta_k^c}\}$.
    Instantiate a new Dynamic Adaptive Representation Fusion Mechanism $F_{\omega_k}$.
    if $k > 1$ then
        Create auxiliary network $F_{\tilde{\gamma}^e}$ by copying $F_{\gamma^e}$ from task $\mathcal{T}_{k-1}$ and freeze it.
    end if
    for each batch $(\mathbf{x}, \mathbf{y}) \in \mathcal{D}_k$ do
        Compute fused feature representation via Eq.(4): $\mathbf{z}' = F_f(\mathbf{x})$.
        Generate adaptive weights $\mathbf{w} = [w_0, w_1]$ using $F_{\omega_k}$.
        Form final prediction input via Eq. (7).
        if $k > 1$ then
            Compute current and historical representations $\mathbf{Z}^{k,i}$ and $\mathbf{Z}_{\text{his}}^{k,i}$ for all $i < k$.
            Calculate prediction calibration loss $\mathcal{L}_p$ via Eq. (10).
        end if
        if $k > 1$ then
            Extract intermediate features $\tilde{\mathbf{Z}}^i$ and $\tilde{\mathbf{Z}}_{\text{his}}^i$ from each of the $L$ trainable layers.
            Calculate representation calibration loss $\mathcal{L}_r$ via Eq. (13) using MMD or MSE.
        end if
        Compute total loss: $\mathcal{L}_{total} = \mathcal{L}_{ce} + \lambda_p \mathcal{L}_p + \lambda_r \mathcal{L}_r$.
        Update parameters of: (i) last $L$ layers of $F_{\gamma^e}$, (ii) $\mathcal{E}_k$, (iii) $F_{\omega_k}$.
    end for
    Preserve the learned $\mathcal{E}_k$, $F_{\omega_k}$, and updated $F_{\gamma^e}$ for future use.
end for
```

B Additional Information for the Experiment Setting

B.1 Device Configurations.

All experiments were conducted on the same hardware environment running Ubuntu 22.04.2 LTS, with 256 GB of RAM and Intel Xeon Silver4320. A single NVIDIA A100 GPU provides the computing acceleration in experiments.

B.2 Datasets

In this section, we provide a detailed description of the datasets used in the MTIL scenario. The proposed method is evaluated on a task sequence consisting of seven diverse datasets: CIFAR-10 (C10) [9], CIFAR-100 (C100) [10], Tiny ImageNet (TIN) [11], ImageNet-R (IN-R) [6], CropDisease (CD) [14], CUB-200-2011 (CUB) [18], and RESISC45 [4]. The order of these datasets during training is specifically designed to reflect a progressive learning process, starting with general visual recognition tasks and progressing to more specialized and complex tasks involving fine-grained classification and cross-domain understanding.

These datasets vary significantly in terms of image resolution, class granularity, semantic content, and their respective application domains. The CIFAR series consists of small-sized images with relatively broad categories, while Tiny ImageNet introduces a greater number of classes, increasing the complexity of the recognition task. ImageNet-R provides stylized images to simulate real-world distribution shifts, testing the robustness of the model under such changes. CropDisease offers a fine-grained classification task in the context of agricultural disease detection, and CUB-200-2011 challenges the model with highly similar bird species, requiring fine distinctions. Finally, RESISC45, a remote sensing dataset, presents a completely different modality, focusing on land cover classification using satellite imagery, which is distinct in spatial structure and content.

The datasets are carefully selected to evaluate the continual learning capability and generalization performance of the model under evolving data distributions and varying task characteristics. This task sequence is designed to reflect real-world scenarios where a model must adapt to new domains and tasks while maintaining performance on previously learned tasks.

We present a comprehensive overview of the dataset characteristics and preprocessing strategies employed in our MTIL framework. Table 1 summarizes the key properties of each dataset, including their learning order, category descriptions, original image dimensions, and specific resizing and cropping transformations applied during training and testing phases.

Table 1: Basic information and transformations for datasets.

Dataset	Order	Category	Original Size	Phase	Resizing & Cropping
CIFAR-10	1	General Recognition	$3 \times 32 \times 32$	Training Testing	Resize 224, Random crop (padding 28) Resize 224
CIFAR-100	2	General Recognition	$3 \times 32 \times 32$	Training Testing	Random resized crop 224 Resize 224
Tiny ImageNet	3	Object Detection	$3 \times 64 \times 64$	Training Testing	Random resized crop 224 Resize 256, Center crop 224
ImageNet-R	4	Robustness Evaluation	$3 \times 224 \times 224$	Training Testing	Random resized crop 224 Resize 256, Center crop 224
CropDisease	5	Agricultural Disease Detection	$3 \times 224 \times 224$	Training Testing	Random resized crop 224 Resize 256, Center crop 224
CUB-200	6	Fine-grained Bird Classification	$3 \times 224 \times 224$	Training Testing	Resize 300, Random crop 224 Resize 256, Center crop 224
RESISC45	7	Remote Sensing Scene Classification	$3 \times 224 \times 224$	Training Testing	Random resized crop 224 Resize 256, Center crop 224

B.3 SOTA Methods

We conduct a comprehensive evaluation of our proposed method against a diverse set of state-of-the-art (SOTA) continual learning approaches, reflecting the major categories discussed in Related Work. This comparison includes both rehearsal-based and rehearsal-free methods, leveraging prompting, parameter-efficient fine-tuning (PEFT), dynamic adaptation strategies, and other techniques.

Rehearsal-based Methods: We compare against recent SOTA rehearsal-based techniques that utilize episodic memory to mitigate forgetting. This includes Dark Experience Replay++ (DER++) [2], which enhances standard experience replay with dark knowledge regularization using stored logits, and its improved variant DER++Refresh (DER++(Re)) [20], which introduces a periodic refresh mechanism to better consolidate old knowledge. Additionally, we include CLS-ER [1], inspired by complementary learning systems, which employs dual semantic memories alongside experience replay for effective knowledge consolidation. These methods serve as strong baselines leveraging memory buffers.

Prompt-based Methods: We benchmark against recent advancements in rehearsal-free continual learning using prompt-based strategies. This includes Learning to Prompt (L2P) [22], which learns a shared prompt pool selected via input-dependent queries; DualPrompt [21], which decouples knowledge into general and task-specific prompts for better transfer and reduced forgetting; and the recent SOTA CODA-Prompt [16], which employs a decomposed attention mechanism to dynamically assemble prompts from learned components, offering greater learning capacity. Furthermore, we compare with Domain-Adaptive Prompting (DAP) [7], which generates instance-level prompts to improve domain scalability.

Dynamic Adaptation Strategies: We evaluate methods that efficiently adapt pre-trained models through various mechanisms. This category includes SEMA [19], a self-expansion method that dynamically adds modularized adapters based on detected feature distribution shifts, enabling sub-linear parameter growth and effective knowledge reuse. Finally, we consider RanPAC [13], which leverages frozen random projections and class prototypes, demonstrating strong performance by effectively utilizing pre-trained features without backbone modification or rehearsal.

Other Rehearsal-free Strategies: Our comparison also includes other effective rehearsal-free approaches. This includes Slow Learner with Classifier Alignment (SLCA) [25], which employs a slow learning rate for representation layers and a post-hoc classifier alignment strategy to balance stability and plasticity.

To ensure consistency and fair comparison, for all methods that do not involve modifications to the backbone architecture—such as DER++, DER++(Re), and CLS-ER—we employ a Vision Transformer (ViT) pre-trained on ImageNet-21K and further fine-tuned on ImageNet-1K, with only the last three layers unfrozen (consistent with our backbone setup) and a trainable classification head. For methods utilizing episodic memory, we set the memory buffer to a maximum capacity of 5120. Other methods are implemented according to the original configurations reported in their respective papers, particularly for prompt- and adapter-based approaches.

B.4 Implementation

We initialize our Dual-Route Backbone Architecture (DRBA) using the ViT-B/16 model pre-trained on ImageNet-21K and further fine-tuned on ImageNet-1K. The invariant representation branch remains fully frozen throughout continual learning to preserve strong general visual features, while only the top three Transformer layers of the evolving representation branch are unfrozen to enable task-specific adaptation.

The Dynamic Adaptive Representation Fusion Mechanism (DARFM), employs a task-adaptive weighting function F_{ω_k} to dynamically fuse invariant and evolving features. In our implementation, the concatenated feature vector $\mathbf{z}' \in \mathbb{R}^{2d}$ is first projected into a 128-dimensional latent space via a linear transformation $\mathbf{W}^{P}_{\omega_k}$. A single-head self-attention layer (Eq. (5)) is then applied, with the query, key, and value vectors all set to 64 dimensions. The attention output is subsequently transformed by a projection matrix $\mathbf{W}^{w}_{\omega_k}$ to produce a 2-dimensional fusion weight vector $\mathbf{w}$, representing the adaptive importance of each feature route under different task distributions.

In the Dynamic Knowledge Calibration Mechanism (DKCM), we incorporate both the Prediction Calibration Process (PCP) and the Representation Calibration Process (RCP). For the PCP, we apply a regularization term $\lambda_{\text{p}}\mathcal{L}_p$ (Appendix Eq. (3)) with a coefficient $\lambda_{\text{p}} = 0.3$ to encourage alignment between the predictions of the current and previous experts, mitigating knowledge divergence. Similarly, for the RCP, we use a regularization term $\lambda_{\text{r}}\mathcal{L}_r$ (Appendix Eq. (3)) with a coefficient

$\lambda_\text{r} = 0.3$ to promote alignment between the intermediate representations of the current and previous evolved backbones, enhancing feature consistency. For optimization, we adopt the stochastic gradient descent (SGD) optimizer. Each task is trained for a single epoch with a batch size of 32.

B.5 Evaluation Metrics

To comprehensively evaluate the performance of our method in both MTIL settings, we adopt a set of widely used continual learning metrics. These metrics assess different aspects of model behavior, including knowledge retention, adaptability to new tasks, and overall performance.

Average Accuracy (Average) [12]. It denotes the average accuracy over all tasks after the model is trained on the final task T, and is commonly used in the continual learning community. It is formulated as:

$$\text{Average Accuracy} = a_T = \frac{1}{T}\sum_{j=1}^{T} a_{T,j}, \tag{4}$$

where $a_{i,j}$ denotes the accuracy on the test set of task j after the model is trained on task i.

Forgetting Measure [3]. This metric quantifies the extent to which knowledge from previous tasks is forgotten. The forgetting for task j after learning the final task T is defined as:

$$\text{Forgetting Measure} = \frac{1}{T-1}\sum_{j=1}^{T-1} f_T^j, \quad \text{where } f_T^j = \max_{l\in\{1,\dots,T-1\}} a_{l,j} - a_{T,j}. \tag{5}$$

Learning Accuracy [15]. This metric evaluates how effectively the model acquires knowledge from new tasks. It is computed as the average accuracy on each task immediately after it is learned:

$$\text{Learning Accuracy} = \frac{1}{T}\sum_{j=1}^{T} a_{j,j}. \tag{6}$$

Last [24]. This metric measures the model's accuracy on the most recently learned task:

$$\text{Last} = a_{T,T}. \tag{7}$$

indicating the model's ability to retain plasticity and effectively learn the final task.

C Computational Cost Analysis

To thoroughly assess the computational overhead and resource demands of our proposed framework, we conduct a detailed comparison with representative SOTA CL methods across the MTIL benchmark, covering seven diverse domains. The comparison encompasses four critical dimensions: number of trainable parameters, peak and average GPU/CPU memory usage, and average training iteration time. As shown in Tab. 2, our method, AEMM (mmd), achieves an excellent balance between representation power and computational efficiency, often outperforming existing baselines in both memory consumption and speed.

Table 2: Comparison of our method with other SOTA methods in terms of trainable parameters, GPU usage, and average time per training iteration. All results are from the seven datasets in the MTIL scenario. Best and second-best results are **bold** and underlined.

Methods	**Params ↓**	**GPU Max ↓**	**GPU Avg ↓**	**CPU Max ↓**	**CPU Avg ↓**	**Iteration ↓**
DER++	21.42M	7138.63MiB	7138.63MiB	20160.90MiB	15423.49MiB	2.68 s/it
DER++(Re)	21.42M	11567.83MiB	11567.83MiB	16845.37MiB	12205.65MiB	2.80 s/it
CLS-ER	21.42M	15718.00MiB	15718.00MiB	18081.05MiB	13354.19MiB	2.64 s/it
L2P	**0.20M**	3420.06MiB	3420.06MiB	17072.70MiB	12338.38MiB	0.98 s/it
Dualprompt	0.41M	3556.64MiB	3556.64MiB	17046.43MiB	12315.18MiB	**0.45 s/it**
CODAPrompt	3.81M	4744.15MiB	4744.15MiB	**16773.28MiB**	12218.85MiB	0.94 s/it
Dap	0.51M	3686.00MiB	3686.00MiB	17181.89MiB	12455.47MiB	0.48 s/it
Ranpac	1.34M	3064.90MiB	3064.90MiB	17686.42MiB	13459.57MiB	0.43 s/it
SEMA	1.06M	4940.86MiB	4940.86MiB	18493.46MiB	16727.44MiB	2.07s/it
SLCA	85.95M	16998.00MiB	15578.00MiB	17535.84MiB	12647.40MiB	2.01s/it
AEMM (mmd)	22.85M	**2701.66MiB**	**2701.66MiB**	16827.99MiB	**12074.17MiB**	0.75 s/it

Rehearsal-based approaches, including DER++, DER++(Re), and CLS-ER, exhibit the highest computational burden across all metrics. Their elevated GPU memory consumption stems primarily from the need to replay a large buffer of previously seen data, effectively doubling the input load per training iteration. In particular, CLS-ER consumes over 15.7 GB of peak GPU memory due to its semantic memory module, which maintains rich intermediate feature statistics for each task, significantly increasing both parameter and activation storage demands. DER++(Re), which integrates an additional buffer refresh mechanism, shows even higher GPU usage compared to the base DER++, underscoring the cost of managing dynamic memory updates. While these methods offer strong performance in some settings, their resource requirements pose clear limitations for large-scale or online learning scenarios.

In contrast, prompt-based approaches such as DualPrompt, DAP, and CODAPrompt achieve significantly lower memory usage and faster iteration times due to their parameter-efficient designs. These methods typically freeze the backbone model and learn only a small set of prompt vectors or prototypes, resulting in a minimal number of trainable parameters (typically < 1M) and reduced backpropagation overhead. For instance, DualPrompt achieves the fastest training iteration (0.45 s/it), while L2P and CODAPrompt remain below 1 second per iteration. However, these gains in efficiency are often accompanied by limited representational capacity, making such methods less effective in scenarios that require substantial domain-specific adaptation or dynamic representation learning.

Our method, AEMM (mmd), offers a favorable trade-off between representation richness and computational efficiency. Despite using a full Transformer backbone and dual-branch architecture, it achieves the lowest GPU memory consumption (2.70 GB peak and average) and the lowest average CPU memory usage (12.07 GB), outperforming all other baselines. Its average training iteration time (0.75 s/it) is also highly competitive, surpassing all non-prompting methods and approaching the efficiency of prompt-based ones. This is made possible by our DRBA, which freezes the invariant branch during training while only updating the lightweight evolved branch and calibration modules. This design drastically reduces the volume of gradient-tracked activations, thereby minimizing memory footprint without compromising adaptation ability. The result is a method that maintains high flexibility and generalization across domains while remaining exceptionally efficient—a key requirement for practical continual learning in real-world systems.

Table 3: Average accuracy (%) of different DRBA architectural variants on the seven datasets in the MTIL scenario.

Methods	C10	C100	TIN	IN-R	CD	CUB	Resisc45	Avg
Evolved-only	92.79±1.49	84.00±2.74	81.16±0.80	75.48±1.42	79.48±2.80	78.77±0.14	79.46±0.10	81.59±5.56
Invariant-only	91.66±3.98	83.68±1.42	80.95±1.05	77.98±0.92	81.69±0.73	80.63±0.68	81.52±1.51	82.59±4.34
Dual-Evolved	95.51±0.37	88.31±0.42	84.97±0.02	79.47±0.68	82.74±0.17	81.98±0.15	82.21±0.10	85.03±5.39
Full DRBA	**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**

Table 4: Comparison of Average and Last accuracy (%) under four task permutations in the MTIL setting: C100 → RESISC45, RESISC45 → C100, and CD → CUB, CUB → CD.

Methods	C100-RESISC45		RESISC45-C100		CD-CUB		CUB-CD	
	Average	Last	Average	Last	Average	Last	Average	Last
AEMM (mmd) + ViT-B/16-1K	84.63±1.13	85.90±0.22	**86.29±0.13**	**84.66±0.56**	**93.21±0.25**	**88.53±0.32**	**81.53±1.02**	**86.80±0.44**
+ ViT-B/16-21K	**84.84±0.03**	**85.93±0.05**	86.19±0.24	83.88±0.10	92.60±0.15	87.14±0.13	80.68±0.13	86.28±0.40
+ ViT-B/16-CLIP	80.71±0.05	83.79±0.21	85.76±0.42	81.28±1.14	86.79±0.31	77.00±0.30	66.03±0.45	72.55±0.15
+ ViT-B/32-21K	84.32±0.09	85.08±0.20	84.49±0.34	82.50±0.23	90.38±0.55	83.59±1.04	76.20±1.22	82.96±1.05
AEMM (mse) + ViT-B/16-1K	**85.18±0.05**	**86.16±0.21**	**86.78±0.08**	**84.72±0.22**	**93.01±0.03**	**87.70±0.08**	**82.65±0.11**	**87.54±0.20**
+ ViT-B/16-21K	84.35±0.04	85.33±1.09	85.71±0.16	83.75±0.84	92.56±0.73	87.27±0.44	81.29±0.23	86.79±0.86
+ ViT-B/16-CLIP	80.71±0.22	83.79±0.54	86.56±0.51	82.69±1.04	87.32±0.08	77.10±0.21	68.51±0.20	75.60±0.66
+ ViT-B/32-21K	83.91±0.56	84.53±1.22	84.94±1.14	83.22±1.55	90.08±0.46	82.55±0.87	75.09±0.96	82.63±1.00

D The Additional Ablation Studies

In this section, we provide additional ablation studies to analyze the performance of the proposed framework.

D.1 Effectiveness of Dual-Branch Representation Architecture (DRBA)

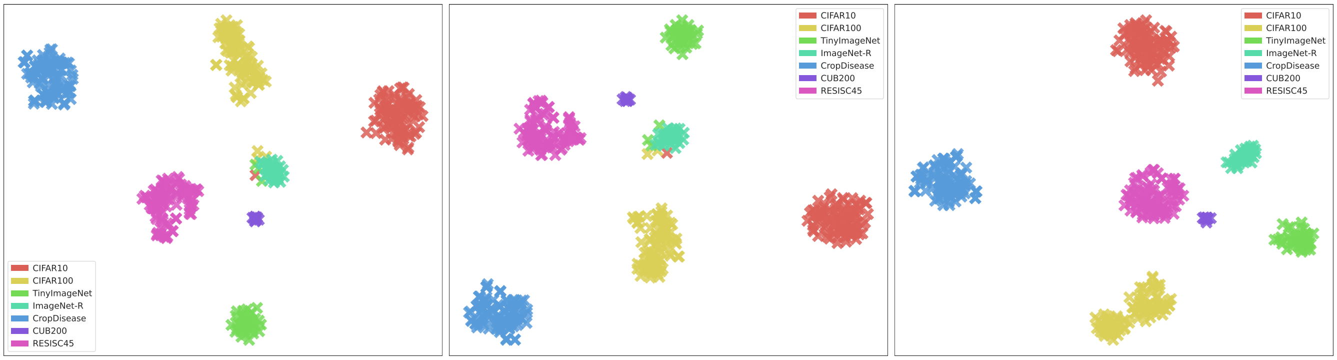
To verify the necessity and effectiveness of the proposed Dual-Representation Backbone Architecture (DRBA), we compare our full model against three architectural variants with modified representation pathways. Specifically, the Evolved-only variant removes the invariant branch entirely and relies solely on task-adaptive layers; the Invariant-only variant discards the evolved branch and retains only the frozen backbone features; and the Dual-Evolved variant replaces the frozen invariant branch with a second trainable evolved branch, effectively emphasizing adaptability over stability by making both pathways task-adaptive.

Experimental results summarized in Tab.3 reveal clear performance distinctions across the variants. Both Evolved-only and Invariant-only models underperform the full DRBA, achieving average accuracies of 81.59% and 82.59% respectively, confirming that neither isolated plasticity nor rigidity is sufficient for effective multi-domain continual learning. The Dual-Evolved variant shows notable improvement (85.03%), suggesting that expanding adaptive capacity brings performance gains. However, it still falls short of the full DRBA model, which achieves the highest average accuracy of 87.77%.

These findings confirm that simply increasing model capacity through additional adaptive branches cannot substitute the complementary stability provided by an invariant representation. Instead, the explicit architectural decoupling in DRBA—where a fixed backbone captures transferable general features and a lightweight evolved head accommodates task-specific shifts—strikes a more effective balance. This result underscores that the synergy between stable and adaptive representations is critical for achieving robust generalization and resistance to forgetting in complex MTIL scenarios.

Table 5: Average accuracy (%) of three fusion variants in DARFM across seven datasets in the MTIL scenario. DF (Dynamic Fusion) denotes our full task-adaptive design, while NC and SF are ablation baselines with naïve and static fusion, respectively.

Fusion Variant	C10	C100	TIN	IN-R	CD	CUB	Resisc45	Avg
Naïve Concatenation (NC)	95.40±0.73	87.27±0.11	84.34±0.29	80.40±0.41	80.56±0.27	81.77±2.12	81.28±3.17	84.43±5.43
Static Fusion (SF)	**96.53±0.75**	89.08±0.50	84.02±1.22	81.31±1.54	82.53±2.20	83.85±2.11	83.16±3.02	85.78±5.33
Dynamic Fusion (DF)	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**



((a)) Naïve Concatenation (NC)

((b)) Static Fusion (SF)

((c)) Dynamic Fusion (DF)

Figure 1: t-SNE visualization of fused features extracted at the end of each task in the MTIL sequence. Compared to NC and SF, our Dynamic Fusion strategy (DF) achieves better separation and compact clustering across domains, indicating enhanced domain discrimination and lower representational interference.

D.2 Influence of Calibration Loss and Backbone Pretraining Strategy

We further investigate how the choice of representation calibration loss and pretraining strategy of the backbone affect performance under the DKCM module. Specifically, we evaluate two versions of our AEMM model, differing only in the loss function used in the Representation Calibration Process (RCP): one based on Maximum Mean Discrepancy (MMD) and the other on Mean Squared Error (MSE). Both variants are tested with four widely adopted pretrained backbones: ViT-B/16 pretrained on ImageNet-21K (ViT-B/16-21K), ViT-B/16 further fine-tuned on ImageNet-1K (ViT-B/16-1K), CLIP's ViT-B/16 pretrained on image-text pairs (ViT-B/16-CLIP), and ViT-B/32 pretrained on ImageNet-21K (ViT-B/32-21K).

Across all task permutations, we find that both AEMM (mmd) and AEMM (mse) variants produce competitive results, highlighting the general robustness of the DKCM design to calibration loss choices. Notably, AEMM (mse) combined with ViT-B/16-1K consistently yields the highest Last Accuracy, especially in more challenging sequences such as CUB→CD (87.54%) and CD→CUB (87.7%). This suggests that MSE provides a strong inductive bias for stable domain transition, possibly due to its smooth gradient dynamics.

In contrast, CLIP-pretrained backbones underperform across both calibration settings, likely because their multimodal pretraining objectives are less suited to domain-specific classification under continual adaptation. ViT-B/16-1K emerges as the most effective backbone overall, benefiting from task-aligned fine-tuning, while ViT-B/32-21K provides a solid trade-off between performance and computational efficiency.

These findings underscore the importance of tailoring backbone pretraining to the MTIL setting and demonstrate that DKCM remains effective under multiple alignment strategies.

Table 6: Comparison with RCP variants on MTIL benchmark in terms of ”Average Accuracy (Avg Acc)”, ”Forgetting Measure (FM)”, ”Learning Accuracy (Lrn Acc)”, and ”Total Avg” scores (%).

RCP	C100			CD			CUB			Total Avg		
	Avg Acc	FM	Lrn Acc	Avg Acc	FM	Lrn Acc	Avg Acc	FM	Lrn Acc	Avg Acc	FM	Lrn Acc
MSE	86.66±0.33	-	86.66±0.12	**91.90±0.55**	1.41±0.16	**92.61±1.21**	87.23±0.08	0.37±0.41	0.48±0.30	88.60±0.26	0.89±0.74	**88.92±0.15**
MMD	**86.71±1.13**	-	**86.71±0.97**	91.80±0.54	0.31±0.22	91.96±0.81	**87.54±0.33**	0.07±0.05	**87.58±0.67**	**88.68±0.44**	0.19±0.11	88.75±0.76
COSINE	85.77±1.05	-	85.77±1.22	90.92±0.69	**0.19±0.14**	91.01±1.33	86.74±0.58	**0.06±0.03**	86.78±0.47	87.81±0.92	**0.12±0.09**	87.85±0.55
HSIC	85.02±0.87	-	85.02±1.02	90.96±0.63	1.13±0.41	91.52±1.35	86.87±0.29	0.22±0.18	87.01±0.77	87.62±1.05	0.67±0.33	87.85±0.51
CKA	85.21±0.96	-	85.21±0.74	91.16±1.12	0.93±0.25	91.62±0.68	86.92±1.44	0.03±0.08	86.87±0.39	87.76±0.82	0.48±0.55	87.90±1.21

D.3 Effectiveness of the Task-Adaptive Feature Fusion in DARFM

To evaluate the effectiveness of the task-adaptive fusion mechanism in DARFM, we compare three fusion strategies under the MTIL benchmark with seven datasets: (i) Naïve Concatenation (NC): direct concatenation of invariant and evolved features without fusion weighting; (ii) Static Fusion (SF): fusion using a fixed ratio (0.5:0.5); (iii) Dynamic Fusion (DF): our full DARFM, which learns task-specific fusion weights.

As shown in Tab. 5, DF significantly outperforms both NC and SF, achieving the highest average accuracy (87.77%). NC yields the lowest performance (84.43%), indicating that unweighted feature stacking is suboptimal. SF shows moderate improvement (85.78%), but still lacks task-level adaptability. These results confirm that dynamic, learnable fusion provides more effective integration of invariant and adaptive features, enhancing generalization across heterogeneous domains.

To further validate this, we visualize the fused features using t-SNE. As shown in Fig. 1, Compared to NC and SF, where features from different domains are entangled and poorly clustered, DF yields clearer separation and compactness. This suggests that task-adaptive fusion improves domain disentanglement and mitigates representational interference throughout continual learning.

D.4 Analysis of RCP Variants Using Alternative Calibration Losses

To further investigate the flexibility and effectiveness of the proposed RCP module, we conducted an extended ablation study by substituting the regularization term with several widely adopted feature alignment losses. These alternative losses were chosen to capture feature-level similarity or distribution alignment from different perspectives, as outlined below:

- **Mean Squared Error (MSE)**: A point-wise Euclidean loss that penalizes deviations between corresponding feature vectors from the current and historical models. It encourages direct alignment of individual features.

- **Maximum Mean Discrepancy (MMD) [17]**: A distribution-level metric that compares the means of two feature distributions in Reproducing Kernel Hilbert Space (RKHS), enabling soft alignment without requiring point-wise correspondences.

- **Cosine Distance (COSINE) [23]**: Measures the angular similarity between vectors. It promotes alignment in direction rather than magnitude, and is often used in metric learning tasks.

- **Hilbert-Schmidt Independence Criterion (HSIC) [5]**: A kernel-based dependency measure that evaluates statistical dependence between two sets of features. Lower HSIC implies higher independence, so it is typically maximized in disentanglement; here we use its negated form to encourage alignment.

- **Centered Kernel Alignment (CKA) [8]**: A similarity measure between two sets of features that remains invariant to isotropic scaling. It has been shown to be effective in comparing representations learned across layers or models.

Tab. 6 presents the average accuracy, forgetting measure, and learning accuracy for three representative C100, CD, and CUB—as well as the overall Total Average across all domains. From the results, we can observe that MMD achieves the highest average accuracy on both C100 (86.71%) and CUB (87.54%) datasets, along with the lowest forgetting measure overall (0.19%). This demonstrates MMD's effectiveness in maintaining stable representations across tasks, aligning with its function as a distribution-matching criterion. On the other hand, MSE, while showing slightly higher forgetting (FM = 0.89%), achieves the best performance on the CD dataset (91.90%) and the highest learning accuracy overall (88.92%), highlighting its strength in adapting to new tasks by precisely aligning feature representations.

On the other hand, Cosine distance performs competitively with a forgetting measure of 0.12% and decent average accuracy of 87.81%. This suggests that directional alignment strikes a reasonable balance between stability and plasticity in continual learning. In contrast, HSIC and CKA, while powerful tools for measuring feature correlation and similarity, do not perform as well in the task-specific calibration context. Their higher forgetting measures (0.67% and 0.48%, respectively) indicate that these losses, which focus on abstract similarity or feature independence, might not be as effective for preserving fine-grained task representations across learning tasks.

This analysis clearly demonstrates the trade-off between stability and adaptability in different RCP loss designs. MMD proves to be the most effective in preserving prior knowledge and minimizing forgetting, while MSE excels in learning new tasks by aligning representations more effectively. Both MMD and MSE outperform the other similarity-based alternatives, which justifies their selection as the primary regularization terms in the RCP module of our framework.

References

[1] Elahe Arani, Fahad Sarfraz, and Bahram Zonooz. 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, 2022. 4

[2] Pietro Buzzega, Matteo Boschini, Angelo Porrello, Davide Abati, and Simone Calderara. Dark experience for general continual learning: a strong, simple baseline. *Advances in neural information processing systems*, 33:15920–15930, 2020. 4

[3] Arslan Chaudhry, Puneet K Dokania, Thalaiyasingam Ajanthan, and Philip HS Torr. Riemannian walk for incremental learning: Understanding forgetting and intransigence. In *Proceedings of the European conference on computer vision (ECCV)*, pages 532–547, 2018. 6

[4] Gong Cheng, Junwei Han, and Xiaoqiang Lu. Remote sensing image scene classification: Benchmark and state of the art. *Proceedings of the IEEE*, 105(10):1865–1883, 2017. 3

[5] Arthur Gretton, Kenji Fukumizu, Choon H Teo, Le Song, Bernhard Schölkopf, and Alex J Smola. A kernel statistical test of independence. In *Advances in neural information processing systems*, pages 585–592, 2008. 10

[6] Dan Hendrycks, Steven Basart, Norman Mu, Saurav Kadavath, Frank Wang, Evan Dorundo, Rahul Desai, Tyler Zhu, Samyak Parajuli, Mike Guo, Dawn Song, Jacob Steinhardt, and Justin Gilmer. The many faces of robustness: A critical analysis of out-of-distribution generalization. *ICCV*, 2021. 3

[7] Dahuin Jung, Dongyoon Han, Jihwan Bang, and Hwanjun Song. Generating instance-level prompts for rehearsal-free continual learning. In *Proc. of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 11847–11857, 2023. 5

[8] Simon Kornblith, Mohammad Norouzi, Honglak Lee, and Geoffrey Hinton. Similarity of neural network representations revisited. In *International conference on machine learning*, pages 3519–3529. PMlR, 2019. 11

[9] Alex Krizhevsky and Geoffrey Hinton. Learning multiple layers of features from tiny images. Technical report, Univ. of Toronto, 2009. 3

[10] Alex Krizhevsky and Geoffrey Hinton. Learning multiple layers of features from tiny images. Technical report, University of Toronto, 2009. Technical report. 3

[11] Ya Le and Xuan Yang. Tiny imageNet visual recognition challenge. Technical report, Univ. of Stanford, 2015. 3

[12] David Lopez-Paz, Robert Nishihara, Soumith Chintala, Bernhard Scholkopf, and Léon Bottou. Discovering causal signals in images. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6979–6987, 2017. 6

[13] Mark D McDonnell, Dong Gong, Amin Parvaneh, Ehsan Abbasnejad, and Anton Van den Hengel. Ranpac: Random projections and pre-trained models for continual learning. *Advances in Neural Information Processing Systems*, 36:12022–12053, 2023. 5

[14] Sharada P Mohanty, David P Hughes, and Marcel Salathé. Using deep learning for image-based plant disease detection. *Frontiers in plant science*, 7:215–232, 2016. 3

[15] Matthew Riemer, Ignacio Cases, Robert Ajemian, Miao Liu, Irina Rish, Yuhai Tu, and Gerald Tesauro. Learning to learn without forgetting by maximizing transfer and minimizing interference. In *International Conference on Learning Representations (ICLR)*, 2019. 6

[16] James Seale Smith, Leonid Karlinsky, Vyshnavi Gutta, Paola Cascante-Bonilla, Donghyun Kim, Assaf Arbelle, Rameswar Panda, Rogerio Feris, and Zsolt 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*, pages 11909–11919, 2023. 5

[17] Ilya O Tolstikhin, Bharath K Sriperumbudur, and Bernhard Schölkopf. Minimax estimation of maximum mean discrepancy with radial kernels. *Advances in Neural Information Processing Systems*, 29:1930–1938, 2016. 10

[18] Catherine Wah, Steve Branson, Peter Welinder, Pietro Perona, and Serge Belongie. The caltech-ucsd birds-200-2011 dataset. Technical Report CNS-TR-2011-001, California Institute of Technology, 2011. 3

[19] Huiyi Wang, Haodong Lu, Lina Yao, and Dong Gong. Self-expansion of pre-trained models with mixture of adapters for continual learning. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 10087–10098, 2025. 5

[20] Zhenyi Wang, Yan Li, Li Shen, and Heng Huang. A unified and general framework for continual learning. *arXiv preprint*, arXiv:2403.13249, 2024. 4

[21] Zifeng Wang, Zizhao Zhang, Sayna Ebrahimi, Ruoxi Sun, Han Zhang, Chen-Yu Lee, Xiaoqi Ren, Guolong Su, Vincent Perot, Jennifer Dy, et al. Dualprompt: Complementary prompting for rehearsal-free continual learning. In *European conference on computer vision*, pages 631–648. Springer, 2022. 5

[22] Zifeng Wang, Zizhao Zhang, Chen-Yu Lee, Han Zhang, Ruoxi Sun, Xiaoqi Ren, Guolong Su, Vincent Perot, Jennifer Dy, and Tomas Pfister. Learning to prompt for continual learning. In *Proc. of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 139–149, 2022. 5

[23] SK Michael Wong, Wojciech Ziarko, and Patrick CN Wong. Generalized vector spaces model in information retrieval. In *Proceedings of the 8th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 18–25, 1985. 10

[24] Jiazuo Yu, Yunzhi Zhuge, Lu Zhang, Ping Hu, Dong Wang, Huchuan Lu, and You He. 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)*, pages 23219–23230, 2024. 6

[25] Gengwei Zhang, Liyuan Wang, Guoliang Kang, Ling Chen, and Yunchao Wei. 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*, pages 19148–19158, 2023. 5