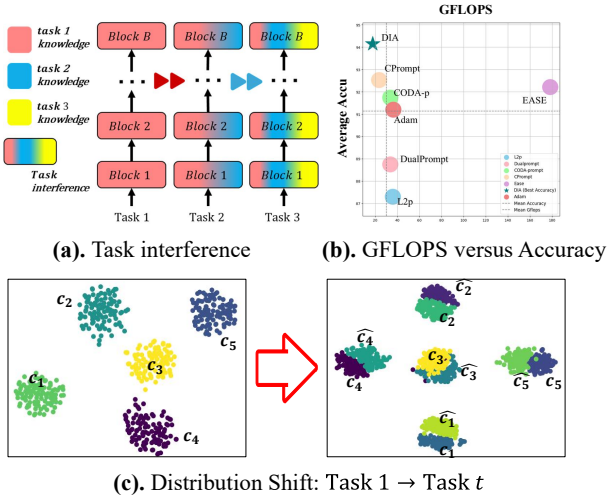
Dynamic Integration of Task-Specific Adapters for Class Incremental Learning

Jiashuo Li[1], **Shaokun Wang**[1], **Bo Qian**[1], **Yuhang He**[2], **Xing Wei**[1], **Yihong Gong**[2]

[1] School of Software Engineering, Xi'an Jiaotong University, Xi'an, 710049, China
[2] College of Artificial Intelligence, Xi'an Jiaotong University, Xi'an, 710049, China
xjtuljs@stu.xjtu.edu.cn; shaokunwang.xjtu@gmail.com; qb990531@stu.xjtu.edu.cn; hyh1379478@163.com; weixing, ygong@mail.xjtu.edu.cn

Abstract

Non-exemplar class Incremental Learning (NECIL) enables models to continuously acquire new classes without retraining from scratch and storing old task exemplars, addressing privacy and storage issues. However, the absence of data from earlier tasks exacerbates the challenge of catastrophic forgetting in NECIL. In this paper, we propose a novel framework called Dynamic Integration of task-specific Adapters (DIA), which comprises two key components: Task-Specific Adapter Integration (TSAI) and Patch-Level Model Alignment. TSAI boosts compositionality through a patch-level adapter integration strategy, which provides a more flexible compositional solution while maintaining low computation costs. Patch-Level Model Alignment maintains feature consistency and accurate decision boundaries via two specialized mechanisms: Patch-Level Distillation Loss (PDL) and Patch-Level Feature Reconstruction method (PFR). Specifically, the PDL preserves feature-level consistency between successive models by implementing a distillation loss based on the contributions of patch tokens to new class learning. The PFR facilitates accurate classifier alignment by reconstructing old class features from previous tasks that adapt to new task knowledge. Extensive experiments validate the effectiveness of our DIA, revealing significant improvements on benchmark datasets in the NECIL setting, maintaining an optimal balance between computational complexity and accuracy. **The full code implementation will be made publicly available upon the publication of this paper.**



(a). Task interference **(b).** GFLOPS versus Accuracy

(c). Distribution Shift: Task 1 → Task t

Figure 1: (a). Shared parameter space leads to task interference. (b). High computation costs characterize current PET-based methods. (c). Gaussian-based feature reconstruction may have a significant deviation from the actual feature distribution. c_i indicates the actual feature distribution of old classes, $\hat{c}_i$ indicates the feature distribution generated by Gaussian Sampling.

Introduction

Class Incremental Learning (CIL) has gained increasing attention within the expansive field of artificial intelligence research (Li et al. 2024b; Zhu et al. 2021b; Wang et al. 2022c; Kurniawan et al. 2024; Li et al. 2024a; Wang et al. 2023a; Bonato et al. 2024; Zhai et al. 2024a), offering the potential for models to continuously learn new knowledge while maintaining the knowledge of previously encountered old classes. Replay-based CIL approaches (Zhu et al. 2021b; Li and Hoiem 2018a; Wang et al. 2022a; Zhou et al. 2023; Rebuffi et al. 2017; Yan, Xie, and He 2021) require the storage of exemplars from previous tasks, which presents challenges in terms of memory limitations and privacy concerns. Benefiting from the advances in Pre-Trained Models (PTMs) (Dosovitskiy et al. 2020), Non-Exemplar Class Incremental Learning (NECIL) has become a promising alternative, enabling the incremental acquisition of knowledge without the need to maintain a buffer of old class exemplars.

Despite the inherent ability of PTMs to produce generalizable features, which has led to superior performance in the NECIL setting, NECIL remains challenging due to two primary factors. **First**, As shown in Fig.1 (a), some studies (Gao et al. 2023; Zhang et al. 2023; Zhou et al. 2024a) that tune a shared parameter space for all tasks unable to segregate parameters for each task. The mixing of parameters leads to task interference, which in turn causes catastrophic forgetting. Particular PET-based methods (Wang et al. 2022b,c; Smith et al. 2023; Gao, Cen, and Chang 2024; Kurniawan et al. 2024) isolate task parameters by designing task-specific prompts. However, these methods establish a strong correlation between prompt selection and the class tokens. This tight coupling impedes the model from leveraging intact old task parameters to reconstruct knowledge without

old exemplars, undermining the model's capacity for effective knowledge retention and reproduction in NECIL scenarios. Moreover, these methods, requiring multiple forward propagations, significantly increase computational costs, as shown in Fig.1 (b). **We summarize the above issues as the compositionality deficiency.**

Second, the absence of exemplars from old tasks prevents the model from applying replay techniques to preserve old task knowledge. Currently, PET-based methods (Wang et al. 2022b,c; Smith et al. 2023; Gao, Cen, and Chang 2024) neglect the importance of maintaining consistency between incrementally trained models. This neglect causes the model to easily overfit new tasks, focusing solely on information relevant to new tasks while discarding others that may be critical for old tasks. It also fails to adapt classifiers to the decision boundary changes introduced by new tasks. There are prototype-based techniques (Zhang et al. 2023; Zhu et al. 2021b; Wang et al. 2023b) that employ the Gaussian distribution to model the distribution of old task features and align the classifier through the generated features. However, as shown in Fig.1 (c), the features created by Gaussian sampling increasingly deviate from the actual features of real images as new tasks are learned. **We attribute the above issues to the model alignment deficiency.**

To mitigate the above two challenges, we introduce the Dynamic Integration of task-specific Adapters (DIA) framework, which comprises two main components: Task-Specific Adapter Integration (TSAI) and Patch-Level Model Alignment. Specifically, the TSAI module is designed to boost compositionality through a patch-level adapter integration strategy, which provides a more flexible compositional solution while maintaining low computation costs. Furthermore, we demonstrate the knowledge retention and reconstruction potential of TSAI on a parameter factorization basis. Building these TSAI's capabilities, we propose a patch-level model alignment strategy to maintain feature consistency and accurate decision boundaries. The proposed model alignment strategy integrates two fundamental parts: 1) The **Patch-Level Distillation Loss (PDL)** maintains feature-level consistency between the old and new models by introducing a distillation loss based on the normalized distance of the patch tokens. PDL evaluates the contribution of patch tokens to the new task learning and penalizes feature drift in non-contributory tokens to maintain feature consistency. 2) The **Patch-Level Feature Reconstruction (PFR)** employs a normalized distance difference to retrieve patch tokens associated with previously acquired task knowledge. These tokens, combined with old class prototypes, are then used to reconstruct old class features that adapt to the newly learned tasks.

Comprehensive experiments on four benchmark datasets demonstrate the superiority of the proposed DIA method, which achieves state-of-the-art (SOTA) performance in the NECIL setting. Moreover, as shown in Fig.1 (a), our DIA maintains up to a 90% decrease in computational complexity while maintaining SOTA performance.

The contributions can be summarized as follows:

- We propose a novel framework entitled Dynamic Integration of task-specific Adapters (DIA) for the NECIL problem, addressing compositionality and model alignment deficiencies.
- We introduce the Task-Specific Adapter Integration (TSAI) module to boost compositionality, which employs a patch-level adapter integration strategy. We also demonstrate its knowledge retention and reconstruction capability through parameter factorization analysis.
- We present a patch-level model alignment strategy based on the knowledge retention and reconstruction capability of TSAI, incorporating PDL to maintain feature consistency and PFR to reconstruct old class features that adapt to new knowledge, improving classifier alignment.
- We conduct extensive experiments across four benchmarks, where the proposed DIA achieves state-of-the-art performance while maintaining an optimal balance between computational complexity and accuracy.

Related Work

Parameter-Efficient Tuning: As an efficient alternative to full fine-tuning, Adapter tuning (Houlsby et al. 2019) was initially introduced to efficiently transfer large pre-trained models to downstream tasks in NLP tasks. Afterward, methods such as Prompt-Tuning (Lester, Al-Rfou, and Constant 2021) and Prefix-Tuning (Li and Liang 2021) adapt models by inserting learnable tokens explicitly tailored to the new tasks. Following the success of Vision Transformers (Dosovitskiy et al. 2020; Liu et al. 2021), PET methods have been adapted for visual transfer learning. Notable examples include Visual Prompt Tuning (VPT) (Jia et al. 2022) and AdapterFormer (Chen et al. 2022), which apply PET techniques to vision tasks. These methods achieve comparable or superior performance to full fine-tuning while maintaining efficiency. In this paper, we propose a Dynamic Integration of task-specific Adapters framework for the NECIL problem based on PET methods (Chen et al. 2022; Houlsby et al. 2019).

Non-exemplar Class Incremental Learning NECIL approaches address privacy and memory concerns by eliminating the need for old task exemplars and employing various techniques such as regularization (Li and Hoiem 2018b; Huang et al. 2024; Yu et al. 2020), augmentation (Zhu et al. 2021a,b; Kim, Park, and Han 2024), and model rectification-based (Wang et al. 2023b; Zhu et al. 2021b) methods to maintain model performance across tasks. Currently, PTM-based methods that sequentially adjust the PTM to stream data with new classes are a promising direction for NECIL. Most methods (Wang et al. 2022c,b; Smith et al. 2023; Gao, Cen, and Chang 2024; Roy et al. 2024) focus on the instance-level prompt selection to segregate task parameters. LAE (Gao et al. 2023) and ADAM (Zhou et al. 2024a) propose unified frameworks for PET methods by model ensemble. SLCA (Zhang et al. 2023) extends the Gaussian modeling (Zhu et al. 2021b) of old task features to rectify classifiers. EASE (Zhou et al. 2024b) utilizes adapters to extract task-specific features through multiple forward propagations.

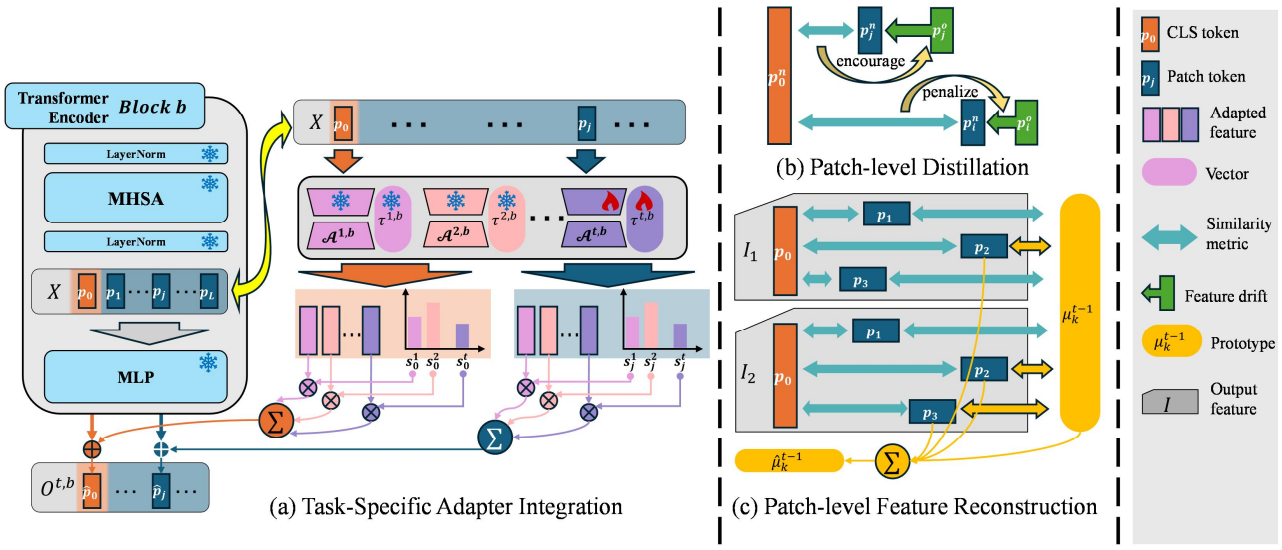


Figure 2: Illustration of DIA. (a) Task-Specific Adapter Integration. For incremental task t, we learn a task adapter $\mathcal{A}^{t,b}$ and a task signature vector $\boldsymbol{\tau}^{t,b} \in \mathcal{R}^d$ at each transformer block b. Each token is routed to the relevant task adapters through the signature vectors, processed independently, and combined into an integrated, task-informed output. (b) Patch-level Distillation. We encourage feature drift in the patch tokens that contribute to new task learning while penalizing others that do not. (c) Patch-level Feature Reconstruction. We identify patch tokens in the patch tokens that are related to old class knowledge and integrate them with the old class prototype $\boldsymbol{\mu}_k^{t-1}$ to reconstruct old class feature $\hat{\boldsymbol{\mu}}_k^{t-1}$.

Problem Setup

In this paper, we focus on the NECIL setting, which prohibits storing exemplars from previous tasks. NECIL involves sequentially learning a set of T tasks, denoted as $\{\mathcal{T}^t\}_{t=1}^T$, where each task $\mathcal{T}^t = \{D^t, C^t\}$ consists of a current training set $D^t = \{(I_i^t, y_i^t)\}_{i=1}^{N^t}$ and a class label set $C^t = \{c_k^t\}_{k=1}^{M^t}$. In this context, I_i^t is the input image, y_i^t is the class label for the i-th image belonging to C^t, N^t represents the number of images, and M^t denotes the number of classes in C^t. There is no overlap between the classes of different tasks, meaning $\forall i, j,\ C^i \cap C^j = \varnothing$. After training on a task $\mathcal{T}^t$, the model is evaluated on all the classes encountered so far, $C^{1:t} = C^1 \cup C^2 \cup \cdots \cup C^t$.

Methodology

Overview

As illustrated in Fig.2, we propose the Dynamic Integration of task-specific Adapters (DIA) framework for the NECIL scenario. At incremental task t, we introduce a task-specific adapter $\mathcal{A}^{t,b}$ parallel to the MLP layer and a task signature vector $\boldsymbol{\tau}^{t,b}$ into each transformer block b. As shown in Fig.2 (a), each image token is routed by the signature vectors $[\boldsymbol{\tau}^{i,b}]_{i=1}^t$ to relevant adapters. These task-specific adapters independently process the input token, and their outputs are merged using scalars determined by the task signature vectors, yielding an integrated output. At task t, only $\mathcal{A}^{t,b}$ and $\boldsymbol{\tau}^{t,b}$ are trainable. Leveraging TSAI's capacity for knowledge retention and reconstruction, along with the rich information embedded in patch tokens, we introduce a patch-level model alignment mechanism at both the feature and classifier levels. Firstly, as shown in Fig.2 (b), we compute the Patch-Level Distillation Loss (PDL) based on the feature drift between the patch tokens p^n, p^o obtained from the model trained on task t and task $t-1$. PDL penalizes the feature drift in patch tokens that do not contribute to the new task learning to preserve old task knowledge. Secondly, we propose a Patch-level Feature Reconstruction (PFR) method to reconstruct old class features without exemplars, as shown in Fig.2 (c). We identify patch tokens related to old task knowledge and integrate them with the old task prototypes. The reconstructed features are utilized to calibrate the decision boundaries of the classifier, adapting to the changes in feature distribution.

Task-Specific Adapters Integration

The proposed TSAI aims to provide a more flexible compositional solution by employing a patch-level adapter integration strategy. To achieve this goal, we learn a task-specific adapter $\mathcal{A}^{t,b}$ and a task signature vector $\boldsymbol{\tau}^{t,b} \in R^d$ for each incremental task, where d indicates feature dimension, t represents the incremental task id, and b represents the block index. For convenience, we omit block index b, as the computation is identical across transformer blocks.

Specifically, the task-specific adapter $\mathcal{A}^t$ is a bottleneck module comprising a down-projection layer $W_{\text{down}} \in \mathbb{R}^{d \times r}$, an up-projection layer $W_{\text{up}} \in \mathbb{R}^{r \times d}$. This bottleneck module extends and adjusts the feature space by modifying the MLP output through the residual connection via a scaling vector $\mathbf{s}^t$ obtained by $\boldsymbol{\tau}^t$. For an input $\mathbf{X} = \left[\mathbf{p}_j{}^\top\right]_{j=0}^L \in R^{(L+1)\times d}$ with $L+1$ image tokens, where $\mathbf{p}_0 \in \mathbb{R}^d$ indicates the class token and $\mathbf{p}_j \in \mathbb{R}^d, j > 1$ indicate the patch token. The task

signature vector $\boldsymbol{\tau}^t$ assigns scalar s_j^t to image token $\mathbf{p}_j$ to evaluate its task relevance:

$$\tilde{\mathbf{p}}_j^t = s_j^t \mathcal{A}^t(\mathbf{p}_j) = s_j^t \text{ReLU}(\mathbf{p}_j W_{\text{down}}) W_{\text{up}}, \quad (1)$$

$$s_j^t = < \bar{\mathbf{p}}_j, \bar{\boldsymbol{\tau}}^t >, \mathbf{s}^t = \left[s_j^t\right]_{j=0}^L, \quad (2)$$

where, $\bar{\mathbf{p}}_j, \bar{\boldsymbol{\tau}}^t$ represent normalized tensors $\bar{\mathbf{p}} = \mathbf{p}/\|\mathbf{p}\|_2$, $\bar{\boldsymbol{\tau}}^t = \boldsymbol{\tau}^t/\|\boldsymbol{\tau}^t\|_2$, and $< \cdot, \cdot >$ represent the dot production.

Afterwards, the output feature $\mathbf{o}_j$ for token $\mathbf{p}_j$ using one adapter $\mathcal{A}^t$ can be calculated as follows:

$$\begin{aligned} \mathbf{o}_j &= \text{MLP}(\mathbf{p}_j) + \tilde{\mathbf{p}}_j^t + \mathbf{p_j} \\ &= \text{MLP}(\mathbf{p}_j) + s_j^t \mathcal{A}^t(\mathbf{p}_j) + \mathbf{p}_j. \end{aligned} \quad (3)$$

For incremental task $t > 1$, the output feature $\mathbf{o}_j$ of TSAI is integrated from multiple task-specific adapters $\{\mathcal{A}^i\}_{i=1}^t$. We use the softmax operation to normalize the scaling vector $[s_j^i]_{i=1}^t$ for patch token $\mathbf{p}_j$, allowing the model to combine the contributions from different adapters while maintaining a balanced and stable output.

$$\mathbf{o}_j = \text{MLP}(\mathbf{p}_j) + \sum_{i=1}^{t} \hat{s}_j^i \mathcal{A}^i(\mathbf{p}_j) + \mathbf{p}_j, \quad (4)$$

$$\mathbf{O} = \left[\mathbf{o}_j^\top\right]_{j=0}^L, \quad [\hat{s}_j^i]_{i=1}^t = \text{softmax}([s_j^i]_{i=1}^t), \quad (5)$$

where $\mathbf{O}$ is the output of TSAI for input $\mathbf{X}$.

Analysis of knowledge of retention and reproduction: We conducted an in-depth analysis of the parameter space for each task-specific adapter, using matrix factorization to demonstrate that our proposed TSAI module can effectively preserve and reproduce knowledge from previous tasks without needing exemplars from old tasks. In particular, we start with no activation function to describe the knowledge retention mechanism under linear conditions and then extend to nonlinear conditions. We provide a more detailed analysis in the supplementary material.

To avoid confusion, assume the input token is $\mathbf{p} \in \mathbb{R}^m$ and the adapter weight is $W = W_{\text{down}} W_{\text{up}} \in \mathbb{R}^{m \times n}$, with a matrix rank of r. The weight matrix can be decomposed through SVD without information loss:

$$\mathbf{W} = \mathbf{U}\text{diag}(\sigma)\mathbf{V}, \quad \mathbf{W} = \sum \mathbf{u}_i \sigma_i \mathbf{v}_i^\top, \quad (6)$$

$$\mathbf{U}^\top = \left[\mathbf{u}_i{}^\top\right]_{i=1}^r \in \mathbb{R}^{r \times m}, \mathbf{V} = \left[\mathbf{v}_i{}^\top\right]_{i=1}^r \in \mathbb{R}^{r \times n}, \quad (7)$$

where diag(σ) is a diagonal matrix with singular values σ_i on the diagonal. $\mathbf{U}$ is an $m \times r$ orthogonal matrix, with the singular vectors $\mathbf{u}_i$ as its columns. $\mathbf{V}$ is an $r \times n$ orthogonal matrix, with the singular vectors $\mathbf{v}_i^\top$ as its rows.

the output $\mathbf{o}$ can be formulated as:

$$\begin{aligned} o = \mathcal{A}(\mathbf{p}) &= \mathbf{W}^\top \mathbf{p} = \sum (\mathbf{u}_i \sigma_i \mathbf{v}_i^\top)^\top \mathbf{p} \\ &= \sum \mathbf{v}_i (\sigma_i \mathbf{u}_i^\top \mathbf{p}) = \sum s_i^{'} \mathbf{v}_i = \mathbf{V}^\top g(\mathbf{p}) \end{aligned} \quad (8)$$

$$g(\mathbf{p}) = \text{diag}(\sigma)^\top U^\top \mathbf{p} = [s_i^{'}]_{i=1}^r \in \mathbb{R}^r. \quad (9)$$

From the above derivation, we can observe that each task adapter allows the model to learn task-specific "keys" and "values" independent of the input features. The output $\mathbf{o}$ for each input token $\mathbf{p}$ is obtained by weighting the task parameter space basis vectors $\mathbf{v}_i$ through a linear function $g(\cdot; U, \text{diag}(\sigma))$ (a non-linear function when ReLU exists). TSAI ensures that the output $\mathbf{o}$ remains within the task subspace, regardless of the input. We can leverage this property by designing appropriate loss functions to maintain feature consistency under the NECIL setting.

Patch-Level Model Alignment

Due to the limitations of NECIL methods that do not store old class exemplars, mainstream methods do not perform model alignment or only align the classifier.

Based on TSAI's knowledge retention and reconstruction ability, as well as the rich information conveyed by patch tokens, we propose a patch-level distillation loss to maintain feature consistency and a patch-level old class feature reconstruction method to align the classifier.

Patch-Level Distillation Loss: To ensure that new tasks can share and reuse old task knowledge while maintaining consistent feature representations for old tasks, we propose patch-level distillation loss (PDL).

As shown in Fig.2 (b), during the training of task t, we obtain the output features $\mathbf{X}^n$ and $\mathbf{X}^o$ for the input image I from both the current model and the model trained on the previous task $t-1$.

$$\mathbf{X}^o = \left[\mathbf{p}_j^{o\top}\right]_{j=0}^L, \mathbf{X}^n = \left[\mathbf{p}_j^{n\top}\right]_{j=0}^L \in \mathbb{R}^{(L+1) \times d}, \quad (10)$$

We first compute the contribution of each patch token to the learning of the new task based on the angular similarity:

$$\alpha_{\cos} = \frac{\mathbf{p}_0^n \cdot \mathbf{p}_j^n}{\|\mathbf{p}_0^n\|_2 \|\mathbf{p}_j^n\|_2}, \alpha_\angle = \frac{\pi - \arccos(-\alpha_{\cos})}{\pi}, \quad (11)$$

here, $\alpha_{\cos}$ and $\alpha_\angle$ represent the cosine similarity and angular similarity, respectively. We also conduct ablation experiments on different similarity metrics in Table 5.

Secondly, we map patch tokens onto a hypersphere using L_2 normalization to ensure numerical stability. Then, we measure the feature drifts by calculating the distances between the corresponding tokens on the hypersphere, as follows:

$$\mathcal{D}(\mathbf{p}_j^n, \mathbf{p}_j^o) = \|\bar{\mathbf{p}_j^n} - \bar{\mathbf{p}_j^o}\|_2. \quad (12)$$

We allow greater flexibility for patch tokens that contribute significantly to new tasks. For patches that contribute less to new tasks, we align their tokens with the output of the old model to maintain feature consistency with previous tasks. Thus, the PDL loss function is defined as:

$$\mathcal{L}_{pdl} = \frac{1}{L} \sum_j \text{SG}\left[\alpha_\angle(\mathbf{p}_0^n, \mathbf{p}_j^n)\right] \mathcal{D}(\mathbf{p}_j^n, \mathbf{p}_j^o), \quad (13)$$

where SG represents the stop gradient operation.

Patch-Level Feature Reconstruction: As new incremental tasks are learned, the decision boundaries established from old tasks could be dramatically changed (Zhu et al. 2021b;

Method	Params	Flops	ImageNet-R		ImageNet-A		CUB-200		Cifar-100	
			$\mathcal{A}^{10}\uparrow$	$\bar{\mathcal{A}}^{10}\uparrow$	$\mathcal{A}^{10}\uparrow$	$\bar{\mathcal{A}}^{10}\uparrow$	$\mathcal{A}^{10}\uparrow$	$\bar{\mathcal{A}}^{10}\uparrow$	$\mathcal{A}^{10}\uparrow$	$\bar{\mathcal{A}}^{10}\uparrow$
FT	86M	17.58B	20.93	40.35	6.03	16.57	22.05	45.67	22.17	41.83
Adam-Ft	86M	33.72B	54.33	61.11	48.52	59.79	86.09	90.97	81.29	87.15
SLCA	86M	17.58B	77.42	82.17	**60.63**	70.04	84.71	90.94	**91.26**	94.09
LAE	0.19M	35.24B	72.39	79.07	47.18	58.15	80.97	87.22	85.33	89.96
Adam-Adapter	1.19M	36.47B	65.29	72.42	48.81	58.84	85.84	91.33	87.29	91.21
EASE	1.19M	177.11B	76.17	81.73	55.04	65.34	84.65	90.51	87.76	92.35
Adam-Prompt-shallow	0.04M	36.28B	65.79	72.97	29.29	39.14	85.28	90.89	85.04	89.49
Adam-Prompt-deep	0.28M	36.28B	72.3	78.75	53.46	64.75	86.6	91.42	83.43	89.47
L2P	0.04M	35.85B	72.34	77.36	44.04	51.24	79.62	85.76	82.84	87.31
DualPrompt	0.13M	33.72B	69.10	74.28	53.19	61.47	81.10	88.23	83.44	88.76
CODA-Prompt	0.38M	33.72B	73.30	78.47	52.08	63.92	77.23	81.90	87.13	91.75
ConvPrompt	0.17M	17.98B	77.86	81.55	—	—	82.44	85.59	88.10	92.39
CPrompt	0.25M	23.62B	77.15	82.92	55.23	65.42	80.35	87.66	88.82	92.53
DIA-r08 (Ours)	0.17M	17.91B	**79.03**	**85.61**	59.78	**70.43**	**86.73**	**92.13**	90.80	**94.29**

Table 1: Experimental results on four CIL benchmarks with 10 incremental tasks. The best results are marked in **bold**, and the second-best are underlined. We report the accuracy of compared methods with their source code. **Params** indicates the trainable parameters in each incremental task. **Flops** represents the number of floating-point operations required to perform inference on a single image.

Wang et al. 2023b). Towards this end, we propose a patch-level feature reconstruction method to generate old class features that adapt to the evolving model, providing better classifier alignment.

Specifically, at each incremental task t, we compute and store a class prototype $\boldsymbol{\mu}_k^t$ for each class k. When learning new incremental tasks, we apply a normalized distance difference $\delta_{(k,i,j)}$ to measure the relevance between patch token $p_{(i,j)}$ and prototype $\boldsymbol{\mu}_k^t$, where $p_{(i,j)}$ indicates the j-th token of X_i within the training batch.

$$\delta_{(k,i,j)} = \frac{\alpha_{\cos}(\boldsymbol{\mu}_k^t, \mathbf{p}_{(i,j)}) - \alpha_{\cos}(\mathbf{p}_{(i,0)}, \mathbf{p}_{(i,j)})}{\alpha_{\cos}(\boldsymbol{\mu}_k^t, \mathbf{p}_{(i,j)}) + \alpha_{\cos}(\mathbf{p}_{(i,0)}, \mathbf{p}_{(i,j)})}, \quad (14)$$

$$\hat{\delta}_{(k,i,j)} = \begin{cases} 0, & \text{if } \delta_{(k,i,j)} \leq 0 \\ \delta_{(k,i,j)}, & \text{if } \delta_{(k,i,j)} > 0 \end{cases}, \quad (15)$$

We retrieve the patch tokens $\mathbf{p}_{(i,j)}$ whose $\hat{\delta}_{(k,i,j)} > 0$ and integrate them with prototype $\boldsymbol{\mu}_k^t$ using the exponential moving average (EMA) (Morales-Brotons, Vogels, and Hendrikx 2024) to generate old task feature $\hat{\boldsymbol{\mu}}_k^t$ as follows:

$$\hat{\boldsymbol{\mu}}_k^t = \beta \cdot \boldsymbol{\mu}_k^t + (1-\beta) \sum_{i,j} \omega_{(i,j)} \cdot \mathbf{p}_{(i,j)}, \quad (16)$$

$$\left[\omega_{(i,j)}\right]_{i,j} = \text{softmax}(\left[\hat{\delta}_{(k,i,j)}\right]_{i,j}), \quad \sum_{i,j} \omega_{(i,j)} = 1, \quad (17)$$

where β is a hyperparameter.

In each training batch, we randomly generate old task features through our PFR and calibrate the classifier using the CrossEntropy loss function as previous methods (Zhang et al. 2023; Zhu et al. 2021b). We provide a more detailed training pipeline in the supplementary material.

Experiments

In this section, we first provide implementation details, then present the experimental results with analyses, and finally show ablation studies and visualization results. More detailed experimental results can be found in the supplementary material.

Implementation Details

Dataset: We conduct experiments on Cifar-100 (Krizhevsky and Hinton 2009), CUB-200 (Wah et al. 2011), ImageNet-R (Hendrycks et al. 2021a), and ImageNet-A (Hendrycks et al. 2021b). These datasets contain typical CIL benchmarks and more challenging datasets with a significant domain gap with ImageNet (i.e., the pre-trained dataset). There are 100 classes in Cifar-100 and 200 classes in CUB, ImageNet-R, and ImageNet-A. For all datasets, we follow the class orders in (Zhou et al. 2024b).

Comparison methods: We compare our method with benchmark PTM-based CIL methods in Table 1 and Table 3. We divide them into three groups: (1) **Prompt-based methods:** L2P (Wang et al. 2022c), DualPrompt (Wang et al. 2022b), CODA-Prompt (Smith et al. 2023), CPrompt (Gao, Cen, and Chang 2024), ConvPrompt (Roy et al. 2024), Adam-Prompt (Zhou et al. 2024a). (2) **Adapter-based methods:** Adam-Adapter (Zhou et al. 2024a), EASE (Zhou et al. 2024b), LAE (Gao et al. 2023), (3) **Finetuning-based methods:** SLCA (Zhang et al. 2023), Adam-Ft (Zhou et al. 2024a).

Training details: We adopt ViT-B/16 (Dosovitskiy et al. 2020) as the pre-trained model, which is pre-trained on ImageNet-21K (Russakovsky et al. 2015). The initial learning rate is set to 0.025 and decays with cosine annealing. We train each task for 20 epochs with a batch size of 32 using Nvidia 3090 GPUs with 24GB of RAM. The down projection of adapters is set to 8 in our main experiments, indicated by DIA-r8 and the hyperparameter β is set to 0.7.

Evaluation metric: Following previous papers (Gao et al. 2023; Zhou et al. 2024b), we denote the model's accuracy after the t-th incremental task as A^t and use $\bar{A}^t = \frac{1}{t}\sum_{i=1}^{t} A^i$ to represent the average accuracy over t incremental tasks. For our evaluation, we specifically focus on two key measurements: A^T (the accuracy after the final(T-th) incremental task) and $\bar{A}^T$ (the average accuracy across all T incremental tasks).

Comparison with SOTA Methods

In this section, we evaluate our proposed DIA method on the ImageNet-R, ImageNet-A, CUB-200, and Cifar-100 datasets, equally dividing the classes into 10 incremental tasks. As shown in Table 1, our method achieves SOTA average accuracies of 85.61%, 70.43%, 92.13%, and 94.29% on four benchmark datasets, respectively.

Prompt-based Methods: In comparison to prompt-based approaches, the proposed DIA demonstrates comprehensive improvements. When compared with the current SOTA method CPrompt (Gao, Cen, and Chang 2024), our DIA exhibits accuracy enhancements ranging from 1.76% to 4.99% across all datasets. Moreover, the DIA-r08 requires fewer training parameters (merely 0.17M) and reduces the floating-point operations (FLOPS) per image by 24.17%. ConvPrompt (Roy et al. 2024) improves inference efficiency by introducing large language models. Our approach, however, offers advantages in both computational efficiency and accuracy across all datasets, notably achieving 4.06% higher average accuracy on ImageNet-R.

Adapter-based Methods: In comparison to adapter-based methods, our DIA still performs exceptionally well. Utilizing only 14.28% of the trainable parameters needed per incremental task, DIA-r08 outperforms EASE (Zhou et al. 2024b) and Adam (Zhou et al. 2024a) across all benchmark datasets. Moreover, DIA-r08 only needs 17.91B Flops to infer an image, reducing inference consumption by 90% compared to EASE (Zhou et al. 2024b) while improving accuracy. Specifically, DIA-r08 surpasses EASE (Zhou et al. 2024b) by 3.88%, 5.09%, 1.62%, and 1.94% on ImageNet-R, ImageNet-A, CUB-200, and Cifar-100, respectively. Our notable improvements on ImageNet-R and ImageNet-A underscore that incorporating compositionality and leveraging knowledge from old tasks can significantly enhance the learning of new classes that have a domain gap with the pre-trained data.

Ft-based Methods: Compared to finetuning-based methods, our DIA not only achieves SOTA average accuracy across four datasets but also saves 99.80% of the training parameters required per task compared to SLCA and Adam-Ft. This demonstrates that our proposed DIA strikes an excellent balance between computational efficiency and performance.

DIA	PDL	PFR	ImageNet-R		Cifar-100	
			$\mathcal{A}^{10}\uparrow$	$\bar{\mathcal{A}}^{10}\uparrow$	$\mathcal{A}^{10}\uparrow$	$\bar{\mathcal{A}}^{10}\uparrow$
			20.93	40.35	22.17	41.83
✓			77.13	83.87	88.37	92.31
✓	✓		78.18	84.55	89.32	93.48
✓		✓	78.22	84.25	89.81	93.78
✓	✓	✓	79.08	85.61	90.80	94.29

Table 2: Ablation Study on Different Components of the DIA Framework

Method	ImageNet-R		Cifar-100	
	$\mathcal{A}^{20}\uparrow$	$\bar{\mathcal{A}}^{20}\uparrow$	$\mathcal{A}^{20}\uparrow$	$\bar{\mathcal{A}}^{20}\uparrow$
Adam-Ft	52.36	61.72	81.27	87.67
SLCA	74.63	79.92	**90.08**	**93.85**
Adam-Prompt-shallow	59.90	68.02	84.57	90.43
Adam-Prompt-deep	70.13	76.91	82.17	88.46
L2P	69.64	75.28	79.93	85.94
DualPrompt	66.61	72.45	81.15	87.87
CODA-Prompt	69.96	75.34	81.96	89.11
ConvPrompt	74.3	79.66	87.25	91.46
CPrompt	<u>74.79</u>	<u>81.46</u>	84.57	90.51
LAE	69.86	77.38	83.69	87.86
Adam-Adapter	57.42	64.75	85.15	90.65
EASE	65.23	77.45	85.80	91.51
DIA-r08 (Ours)	**76.32**	**83.51**	<u>88.74</u>	<u>93.41</u>

Table 3: Experimental results on four CIL benchmarks with 20 incremental tasks. The best results are marked in **bold**, and the second-best are <u>underlined</u>.

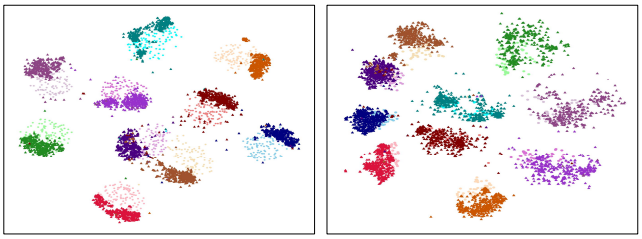
Ablation Study

Component Analysis Our proposed DIA method consists of three main components: 1) Task-Specific Adapter Integration (TSAI), 2) Patch-Level Distillation Loss (PDL), and 3) Patch-Level Feature Reconstruction (PFR). To validate the efficacy of each component, we conduct ablation experiments on two benchmark datasets: Cifar-100, which overlaps with the pre-training data, and ImageNet-R, which exhibits a domain gap from the pre-training data.

The first row in Table 2 shows the average and final accuracy of the pre-trained image encoder with a learnable classifier. The incorporation of the TSAI module yields performance comparable to SOTA methods on both Cifar-100 and ImageNet-R datasets. This indicates that the patch-level integration mechanism atop the adapter effectively reduces interference between new and old tasks while enhancing new task learning, leading to substantial performance gains. The

Method	ImageNet-R		Cifar-100	
	$\mathcal{A}^{10} \uparrow$	$\bar{\mathcal{A}}^{10} \uparrow$	$\mathcal{A}^{10} \uparrow$	$\bar{\mathcal{A}}^{10} \uparrow$
DIA-Gau	78.08	83.72	89.04	93.51
DIA-SLCA	78.37	84.48	89.47	93.79
DIA-PFR	79.03	85.62	90.80	94.29

Table 4: Performance comparison with different feature reconstruction (FR) methods



(a). FR by Gaussian Sampling (b). FR by PFR

Figure 3: Visualization of different feature reconstruction methods. We use dark colored triangle △ to represent actual features and light colored circle ○ to represent pseudo features generated by FR (feature reconstruction) methods.

third and fourth rows highlight the importance of model alignment in NECIL. Leveraging TSAI's retention and reconstruction capabilities, our method gains nearly 1% improvement with the introduction of PDL or PFR. Finally, the fifth row shows that combining compositionality with model alignment achieves SOTA performance, confirming the effectiveness of our framework.

Long Sequence Incremental Analysis: We evaluate the performance of each method under the condition of long sequences. In this setting, datasets are equally divided into 20 tasks with 10 classes/tasks in ImageNet-R and 5 classes/-tasks in Cifar-100, and the results are summarized in Table 3. Our method still maintains excellent performance in terms of A^{20} and $\bar{A}^{20}$. DIA achieves SOTA average accuracies of 83.51% on the ImageNet-R dataset, outperforming the second by 2.05%. We also achieve comparable results with SLCA on Cifar-100, using only 0.17M/86M parameters.

Feature Reconstruction Analysis: We investigate the impact of different old class feature reconstruction methods on classifier alignment. As shown in Table 4, using a Gaussian distribution to simulate old class features yields incremental accuracies of only 93.51% and 93.79% on Cifar-100. In contrast, our proposed PFR method, which leverages TSAI's capability to retain old knowledge, achieves an average accuracy of 94.29%. This indicates that our PFR-based old feature reconstruction method better adapts to the changes in old class features during the incremental process.

We also visualize the feature distributions of old classes generated by Gaussian sampling and compare them to the actual old class features on new tasks, as shown in Fig.3. In the 10th incremental task, we visualize the first 10 classes. Fig.3 (a) shows features generated using a Gaussian distribution, forming two distinct clusters with actual features, highlighting a noticeable disparity. In contrast, Fig.3 (b) displays features generated by PFR, which closely align with actual features, forming a single cohesive cluster. This confirms the effectiveness of our proposed feature generation method and demonstrates the DIA framework's ability to preserve and reproduce old knowledge.

Similarity Metric	ImageNet-R		Cifar-100	
	$\mathcal{A}^{10} \uparrow$	$\bar{\mathcal{A}}^{10} \uparrow$	$\mathcal{A}^{10} \uparrow$	$\bar{\mathcal{A}}^{10} \uparrow$
Euclidean α_{eu}	76.38	81.73	87.42	92.91
Cosine α_{cos}	78.28	84.76	89.67	93.88
Angular $\alpha_{\angle}$	79.03	85.61	90.80	94.29

Table 5: Ablation Study on Similarity Metrics in Patch-Level Distillation Loss (PDL)

PDL ablation: We explore different similarity metrics to assess the contributions of patch tokens to new task learning and analyze their impact on PDL. Specifically, we investigate Euclidean distance (α_{eu}) (Zhai et al. 2024b), cosine similarity (α_{cos}), and angular similarity ($\alpha_{\angle}$) as shown in Table 5. The experimental results indicate that angular similarity yields the best performance, achieving SOTA results across both datasets. In contrast, Euclidean distance performs the worst, with a significant accuracy drop. We attribute this poor performance to the numerical instability of Euclidean distance, which is highly sensitive to variations in numerical range and more prone to outliers. Additionally, cosine similarity results in a slight performance decline, which occurs because it introduces negative loss values, leading to the offsetting of losses between tokens.

Conclusion

In this paper, we propose a novel framework called Dynamic Integration of task-specific Adapters (DIA), which consists of Task-Specific Adapter Integration (TSAI) and Patch-Level Model alignment. The TSAI module enhances compositionality through a patch-level adapter integration mechanism, minimizing task interference while preserving knowledge from old tasks. Patch-level model alignment maintains feature consistency and accurate decision boundaries via a patch-level distillation loss and a patch-level feature reconstruction method. 1) Our PDL preserves feature-level consistency between successive models by implementing a distillation loss based on the contribution of patch tokens to new class learning. 2) Our PFR facilitates accurate classifier alignment by reconstructing features from previous tasks that adapt to new task knowledge. We evaluate DIA on four benchmark datasets proving its efficiency and superior performance.

References

Bonato, J.; Pelosin, F.; Sabetta, L.; and Nicolosi, A. 2024. MIND: Multi-Task Incremental Network Distillation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, 11105–11113.

Chen, S.; Ge, C.; Tong, Z.; Wang, J.; Song, Y.; Wang, J.; and Luo, P. 2022. AdaptFormer: Adapting Vision Transformers for Scalable Visual Recognition. In *Advances in Neural Information Processing Systems*, volume 35, 16664–16678.

Dosovitskiy, A.; Beyer, L.; Kolesnikov, A.; Weissenborn, D.; Zhai, X.; Unterthiner, T.; Dehghani, M.; Minderer, M.; Heigold, G.; Gelly, S.; Uszkoreit, J.; and Houlsby, N. 2020. An Image Is Worth 16x16 Words: Transformers for Image Recognition at Scale. In *International Conference on Learning Representations*.

Gao, Q.; Zhao, C.; Sun, Y.; Xi, T.; Zhang, G.; Ghanem, B.; and Zhang, J. 2023. A Unified Continual Learning Framework with General Parameter-Efficient Tuning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 11449–11459.

Gao, Z.; Cen, J.; and Chang, X. 2024. Consistent Prompting for Rehearsal-Free Continual Learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 28463–28473.

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. 2021a. The Many Faces of Robustness: A Critical Analysis of Out-of-Distribution Generalization. In *IEEE/CVF International Conference on Computer Vision*, 8320–8329.

Hendrycks, D.; Zhao, K.; Basart, S.; Steinhardt, J.; and Song, D. 2021b. Natural Adversarial Examples. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 15257–15266.

Houlsby, N.; Giurgiu, A.; Jastrzebski, S.; Morrone, B.; Laroussilhe, Q. D.; Gesmundo, A.; Attariyan, M.; and Gelly, S. 2019. Parameter-Efficient Transfer Learning for NLP. In *Proceedings of the International Conference on Machine Learning*, 2790–2799.

Huang, L.; Zeng, Y.; Yang, C.; An, Z.; Diao, B.; and Xu, Y. 2024. eTag: Class-Incremental Learning via Embedding Distillation and Task-Oriented Generation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, 12591–12599.

Jia, M.; Tang, L.; Chen, B.-C.; Cardie, C.; Belongie, S.; Hariharan, B.; and Lim, S.-N. 2022. Visual Prompt Tuning. In *Computer Vision – ECCV*, volume 13693, 709–727.

Kim, T.; Park, J.; and Han, B. 2024. Cross-Class Feature Augmentation for Class Incremental Learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, 13168–13176.

Krizhevsky, A.; and Hinton, G. 2009. Learning Multiple Layers of Features from Tiny Images. Technical Report 0, Technical report, University of Toronto / University of Toronto.

Kurniawan, M. R.; Song, X.; Ma, Z.; He, Y.; Gong, Y.; Qi, Y.; and Wei, X. 2024. Evolving Parameterized Prompt Memory for Continual Learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, 13301–13309.

Lester, B.; Al-Rfou, R.; and Constant, N. 2021. The Power of Scale for Parameter-Efficient Prompt Tuning. In Moens, M.-F.; Huang, X.; Specia, L.; and Yih, S. W.-t., eds., *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, 3045–3059.

Li, D.; Wang, T.; Chen, J.; Ren, Q.; Kawaguchi, K.; and Zeng, Z. 2024a. Towards Continual Learning Desiderata via HSIC-Bottleneck Orthogonalization and Equiangular Embedding. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, 13464–13473.

Li, J.; Dong, S.; Gong, Y.; He, Y.; and Wei, X. 2024b. Analogical Learning-Based Few-Shot Class-Incremental Learning. *IEEE Transactions on Circuits and Systems for Video Technology*, 34(7): 5493–5504.

Li, X. L.; and Liang, P. 2021. Prefix-Tuning: Optimizing Continuous Prompts for Generation. In Zong, C.; Xia, F.; Li, W.; and Navigli, R., eds., *Proceedings of the Annual Meeting of the Association for Computational Linguistics and the International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, 4582–4597.

Li, Z.; and Hoiem, D. 2018a. Learning without Forgetting. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(12): 2935–2947.

Li, Z.; and Hoiem, D. 2018b. Learning without Forgetting. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(12): 2935–2947.

Liu, Z.; Lin, Y.; Cao, Y.; Hu, H.; Wei, Y.; Zhang, Z.; Lin, S.; and Guo, B. 2021. Swin Transformer: Hierarchical Vision Transformer Using Shifted Windows. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, 9992–10002.

Morales-Brotons, D.; Vogels, T.; and Hendrikx, H. 2024. Exponential moving average of weights in deep learning: Dynamics and benefits. *Transactions on Machine Learning Research*.

Rebuffi, S.-A.; Kolesnikov, A.; Sperl, G.; and Lampert, C. H. 2017. iCaRL: Incremental Classifier and Representation Learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 5533–5542.

Roy, A.; Moulick, R.; Verma, V. K.; Ghosh, S.; and Das, A. 2024. Convolutional Prompting meets Language Models for Continual Learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 23616–23626.

Russakovsky, O.; Deng, J.; Su, H.; Krause, J.; Satheesh, S.; Ma, S.; Huang, Z.; Karpathy, A.; Khosla, A.; Bernstein, M.; Berg, A. C.; and Fei-Fei, L. 2015. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision*, 115(3): 211–252.

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 (CVPR)*, 11909–11919.

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

Wang, F.-Y.; Zhou, D.-W.; Ye, H.-J.; and Zhan, D.-C. 2022a. FOSTER: Feature Boosting and Compression for Class-Incremental Learning. In *Computer Vision – ECCV*, volume 13685, 398–414. Springer Nature Switzerland.

Wang, S.; Shi, W.; Dong, S.; Gao, X.; Song, X.; and Gong, Y. 2023a. Semantic Knowledge Guided Class-Incremental Learning. *IEEE Transactions on Circuits and Systems for Video Technology*, 33(10): 5921–5931.

Wang, S.; Shi, W.; He, Y.; Yu, Y.; and Gong, Y. 2023b. Non-Exemplar Class-Incremental Learning via Adaptive Old Class Reconstruction. In *Proceedings of the ACM International Conference on Multimedia*, 4524–4534. New York, NY, USA: Association for Computing Machinery.

Wang, Z.; Zhang, Z.; Ebrahimi, S.; Sun, R.; Zhang, H.; Lee, C.-Y.; Ren, X.; Su, G.; Perot, V.; Dy, J.; and Pfister, T. 2022b. DualPrompt: Complementary Prompting for Rehearsal-Free Continual Learning. In *Computer Vision – ECCV*, 631–648.

Wang, Z.; Zhang, Z.; Lee, C.-Y.; Zhang, H.; Sun, R.; Ren, X.; Su, G.; Perot, V.; Dy, J.; and Pfister, T. 2022c. Learning to Prompt for Continual Learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 139–149.

Yan, S.; Xie, J.; and He, X. 2021. DER: Dynamically Expandable Representation for Class Incremental Learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 3013–3022.

Yu, L.; Twardowski, B.; Liu, X.; Herranz, L.; Wang, K.; Cheng, Y.; Jui, S.; and van de Weijer, J. 2020. Semantic Drift Compensation for Class-Incremental Learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 6980–6989.

Zhai, J.-T.; Liu, X.; Yu, L.; and Cheng, M.-M. 2024a. Fine-Grained Knowledge Selection and Restoration for Non-Exemplar Class Incremental Learning. *Proceedings of the AAAI Conference on Artificial Intelligence*, 38(7): 6971–6978.

Zhai, J.-T.; Liu, X.; Yu, L.; and Cheng, M.-M. 2024b. Fine-Grained Knowledge Selection and Restoration for Non-Exemplar Class Incremental Learning. *Proceedings of the AAAI Conference on Artificial Intelligence*, 38(7): 6971–6978.

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 (ICCV)*, 19091–19101.

Zhou, D.-W.; Sun, H.-L.; Ning, J.; Ye, H.-J.; and Zhan, D.-C. 2024a. Continual learning with pre-trained models: A survey. In *IJCAI*.

Zhou, D.-W.; Sun, H.-L.; Ye, H.-J.; and Zhan, D.-C. 2024b. Expandable Subspace Ensemble for Pre-Trained Model-Based Class-Incremental Learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 23554–23564.

Zhou, D.-W.; Wang, Q.-W.; Ye, H.-J.; and Zhan, D.-C. 2023. A Model or 603 Exemplars: Towards Memory-Efficient Class-Incremental Learning.

Zhu, F.; Cheng, Z.; Zhang, X.-y.; and Liu, C.-l. 2021a. Class-Incremental Learning via Dual Augmentation. In *Advances in Neural Information Processing Systems*, volume 34, 14306–14318.

Zhu, F.; Zhang, X.-Y.; Wang, C.; Yin, F.; and Liu, C.-L. 2021b. Prototype Augmentation and Self-Supervision for Incremental Learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 5871–5880.

Dynamic Integration of Task-Specific Adapters for Class Incremental Learning supplementary materials

Anonymous submission

Methodology Supplementary

Analysis of knowledge of retention and reproduction:

As mentioned in the main text, for an input token $\mathbf{p} \in \mathbb{R}^m$, consider an adapter without an activation function, with weights $W = W_{\text{down}} W_{\text{up}} \in \mathbb{R}^{m \times n}, W_{\text{down}} \in \mathbb{R}^{m \times r}, W_{\text{up}} \in \mathbb{R}^{r \times n}$, and a matrix rank of r. The weight matrix can be decomposed using SVD as follows:

$$\mathbf{W} = \mathbf{U}\text{diag}(\sigma)\mathbf{V}, \tag{1}$$

$$\mathbf{U}^\top = \left[\mathbf{u}_i{}^\top\right]_{i=1}^{r} \in \mathbb{R}^{r \times m}, \mathbf{V} = \left[\mathbf{v}_i{}^\top\right]_{i=1}^{r} \in \mathbb{R}^{r \times n}, \tag{2}$$

And the output $\mathbf{o}$ can be formulated as:

$$o = \mathcal{A}(\mathbf{p}) = W^\top \mathbf{p} = \sum s_i^{'} \mathbf{v}_i = \mathbf{V}^\top g(\mathbf{p}), \tag{3}$$

$$g(\mathbf{p}) = \text{diag}(\sigma)^\top U^\top \mathbf{p} = [s_i^{'}]_{i=1}^{r} \in \mathbb{R}^r. \tag{4}$$

After introducing non-linear activation function ReLU, for each input $\mathbf{p} \in \mathbb{R}^m$, the output of the adapter can be reformulated as:

$$\mathcal{A}(\mathbf{p}) = W_{\text{up}}^T \text{ReLU}(W_{\text{down}}^T \mathbf{p}), \tag{5}$$

We perform SVD decomposition on W_{up} as the above Eq. 2:

$$\mathbf{W}_{\text{up}} = \mathbf{U}_{\text{up}} \text{diag}(\sigma_{\text{up}}) \mathbf{V}_{\text{up}}, \tag{6}$$

$$\mathbf{W}_{\text{up}} = \sum \mathbf{u}_{(i,\text{up})} \sigma_{(i,\text{up})} \mathbf{v}_{(i,\text{up})}^\top, \tag{7}$$

the output $\mathbf{o}$ can be reformulated as:

$$\begin{aligned} o = \mathcal{A}(\mathbf{p}) &= W_{\text{up}}^T \text{ReLU}(W_{\text{down}}^T \mathbf{p}) \\ &= \sum \left(\mathbf{u}_{(i,\text{up})} \sigma_{(i,\text{up})} \mathbf{v}_{(i,\text{up})}^\top\right)^\top \text{ReLU}(W_{\text{down}}^T \mathbf{p}) \\ &= \sum \mathbf{v}_{(i,\text{up})} \left(\sigma_{(i,\text{up})} \mathbf{u}_{(i,\text{up})}^\top \text{ReLU}(W_{\text{down}}^T \mathbf{p})\right) \\ &= \sum s_i^{'} \mathbf{v}_{(i,\text{up})} = \mathbf{V}_{\text{up}}^\top g(\mathbf{p}), \end{aligned} \tag{8}$$

$$g(\mathbf{p}) = \text{diag}(\sigma_{(i,\text{up})})^\top \mathbf{U}_{(i,\text{up})}^\top \text{ReLU}(W_{\text{down}}^T \mathbf{p}) \in \mathbb{R}^r. \tag{9}$$

From Eq. 9, we observe that after introducing the nonlinear activation, we obtain an expression similar to Eq. 4, with the function g replaced by a nonlinear function.

Algorithm 1: Training Pipeline for Task t

Input: Training dataset D^t; TSAI module $f(\cdot; \theta_b)$; cosine classifier $\phi(\cdot; W_{cls})$
Parameter: $\theta_b = \{\theta_{ptm}, \theta^o, \theta^n\}$, $W_{cls} = \{W_{cls}^o, W_{cls}^n\}$
Initialization: Initialize θ^n and W_{cls}^n. Freeze parameters θ_{ptm} and θ^o.

```
# Standard Training.
while not converged do
    for {I_i^t, y_i^t} ∈ D^t do
        compute logits ξ_{y_i^t} = φ(f(I_i^t; θ_b); W_cls^n).
        compute CE loss L_CE(ξ_{y_i^t}; s, m)
        compute patch-level distillation loss L_pld
        backward with objective L_obj = L_CE + λL_PDL.
    end for
end while
for each class c_k^t ∈ C^t in task t, we compute its class prototype μ_k^t and store in memory.
# Classifier alignment.
while not converged do
    for Training batch within D^t do
        Sample N class prototypes {μ_{c_i}}_{i=1}^N, c_i ∈ C^{1:t}.
        Construct class feature μ̂_{i,c} using PFR method with the training batch.
        Fine-tune the classifier φ(·; W_cls^o, W_cls^n) .
    end for
end while
```

Training Pipeline

The training pipeline for the proposed DIA method follows previous approaches (**???**) and is composed of two stages: standard training and classifier alignment for each task t, as illustrated in Algorithm.1.

For incremental task t, the TSAI module is parameterized by $\theta_b = \{\theta_{ptm}, \theta^o, \theta^n\}$, where θ_{ptm} denotes the parameters of the pre-trained model (PTM), θ^o refers to the parameters of the adapters and signature vectors for old tasks, and θ^n corresponds to the parameters for the new task t. The cosine classifier $\phi(\cdot; W_{cls})$ comprises two parts: the old class classifier $\phi^o(\cdot; W_{cls}^o)$ and the new class classifier $\phi^n(\cdot; W_{cls}^n)$.

Standard Training: During standard training, for each image $\{I_i^t, y_i^t\} \in D^t$, where $y_i^t \in c_k^t \in C^t$, we first extract features $f(I_i^t; \theta_b)$ using TSAI module $f(\cdot; \theta_b)$, then compute

Method	Cifar-100			
	$\mathcal{A}^{10} \uparrow$	$\bar{\mathcal{A}}^{10} \uparrow$	$\mathcal{A}^{20} \uparrow$	$\bar{\mathcal{A}}^{20} \uparrow$
CLIP-MoE	77.52	85.21	76.20	83.72
DIA-r08	**90.80**	**94.28**	**88.74**	**93.41**

Table 1: Comparision with CLIP-MoE on Cifar-100 dataset. The best results are marked in **bold**.

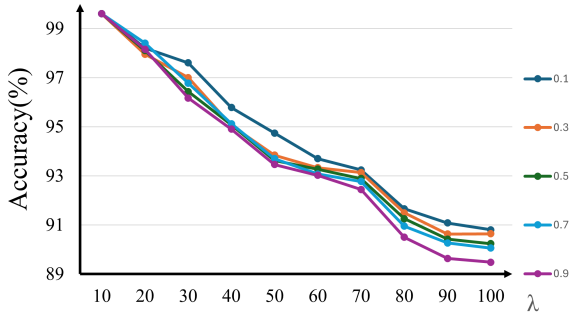
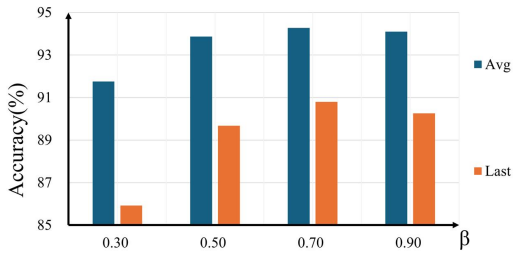
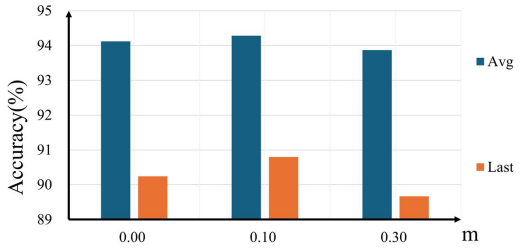


Figure 1: Ablation experiments on hyperparameter λ



(a) Ablation experiments on hyperparameter β



(b) Ablation experiments on hyperparameter m

Figure 2: Ablation experiments on the hyperparameters β and $\mathfrak{m}$ using the CIFAR-100 dataset.

the output logits

$$\xi_{y_i^t} = \phi^n\left(f(I_i^t;\theta_b);W_{cls}^n\right), \tag{10}$$

with the cosine classifier of new task $\phi^n(\cdot;W_{cls}^n)$. We optimize the classification results using a variant of the CE loss $\mathcal{L}_{CE}$ and ensure feature consistency with patch-level distillation loss (PDL) $\mathcal{L}_{pdl}$.

Following methods (**??**), we define $\mathcal{L}_{CE}$ as:

$$\mathcal{L}_{CE}(\xi_{y_i^t};s,\mathfrak{m}) = -\log\frac{e^{s(\xi_{y_i^t}-\mathfrak{m})}}{e^{s(\xi_{y_i^t}-\mathfrak{m})}+\sum_c^{C^t-c_i^t} e^{s(\xi_c)}} \tag{11}$$

Here, s is a scaling factor that adjusts the magnitude of cosine similarity, ensuring that the Softmax function produces a more discriminative probability distribution. $\mathfrak{m}$ introduces an additional angular separation between classes. When $s=0$ and $\mathfrak{m}=0$, $\mathcal{L}_{CE}(\cdot;0,0)$ reduces to the standard cross-entropy loss function. When $s\neq 0$ and $\mathfrak{m}=0$, $\mathcal{L}_{CE}(\cdot;s,0)$ adopts a common format used in cosine classifiers, adjusting the scale of cosine similarity via s.

The loss function during training is defined as:

$$\mathcal{L}_{obj} = \mathcal{L}_{CE} + \lambda\mathcal{L}_{pdl}, \tag{12}$$

where, λ is a hyperparameter that controls the strength of the regularization.

Classifier Alignment: To further refine the classification layer, we perform classifier alignment after the standard training stage, following methods (**??**) (see Algorithm 1). Specifically, during the standard training phase, only the classifiers corresponding to the current task's classes are trained alongside the TSAI module. Once standard training is completed, we compute the class prototype $\boldsymbol{\mu}_k^t = \frac{1}{N_k^t}\sum_i^{N_k^t} f(I_i^t;\theta_b)$ for each class $c_k^t \in C^t$ in the current task, where N_k^t denotes the number of images for class c_k^t.

During the classifier alignment stage for task t, we randomly select N (Set to 32 in our implementation) class prototypes $\{\mu_{c_i}\}_{i=1}^N, c_i \in C^{1:t}$ in each training batch and generate pseudo-features $\{\hat{\mu}_{c_i}\}_{i=1}^N$ using the PFR method. These pseudo-features, combined with training samples, are used as inputs to the classifier $\phi(\cdot;W_{cls}^o,W_{cls}^n)$, which is then fine-tuned using $\mathcal{L}_{CE}(\cdot;0,0)$.

Supplementary Experiments

This section provides additional experiments, including comparisons with CLIP-MoE (**?**), hyperparameter ablation studies, and model structure ablation studies.

Supplementary Implementation Details

In our implementation, we set the margin $\mathfrak{m}$ to 0.1, the hyperparameter λ, which balances $\mathcal{L}_{CE}$ and $\mathcal{L}_{pdl}$, to 0.1, and the parameter β, which balances feature construction, to 0.7.

Comparative Experiments

We conduct comparative experiments with the latest CVPR24 method, CLIP-MoE (**?**). However, since CLIP-MoE did not perform experiments on ImageNet-R, ImageNet-A, or CUB-200, we focus our comparison on CIFAR-100. As shown in Table 1, despite CLIP-MoE (**?**) utilizing a more powerful CLIP backbone and additional semantic information, our method outperforms CLIP-MoE by a significant margin.

Hyperparameter Ablation

We conduct hyperparameter ablation studies in this section. Fig.1 and Fig.2 present the experimental results for the hyperparameters β, $\mathfrak{m}$, and λ. Based on these results, we selected the optimal hyperparameter combination of $\beta=0.7$, $\mathfrak{m}=0.1$, and $\lambda=0.1$.

Method	Params	Flops	ImageNet-R		ImageNet-A		CUB-200		Cifar-100	
			$\mathcal{A}^{10}\uparrow$	$\bar{\mathcal{A}}^{10}\uparrow$	$\mathcal{A}^{10}\uparrow$	$\bar{\mathcal{A}}^{10}\uparrow$	$\mathcal{A}^{10}\uparrow$	$\bar{\mathcal{A}}^{10}\uparrow$	$\mathcal{A}^{10}\uparrow$	$\bar{\mathcal{A}}^{10}\uparrow$
DIA-r08	0.17M	17.91B	79.03	85.61	59.78	70.43	86.73	92.13	90.8	94.28
DIA-r16	0.31M	18.18B	79.82	86.04	57.74	69.84	86.01	91.84	90.38	94.15
DIA-r64	1.19M	19.20B	79.10	85.62	59.91	70.90	87.19	92.26	90.56	94.51

Table 2: Ablation experiments on the adapter rank with 10 incremental tasks.

Method	Params	Flops	ImageNet-R		ImageNet-A		CUB-200		Cifar-100	
			$\mathcal{A}^{10}\uparrow$	$\bar{\mathcal{A}}^{10}\uparrow$	$\mathcal{A}^{10}\uparrow$	$\bar{\mathcal{A}}^{10}\uparrow$	$\mathcal{A}^{10}\uparrow$	$\bar{\mathcal{A}}^{10}\uparrow$	$\mathcal{A}^{10}\uparrow$	$\bar{\mathcal{A}}^{10}\uparrow$
DIA-r08	0.17M	17.91B	79.03	85.61	59.78	70.43	86.73	92.13	90.8	94.28
DIA-r16	0.31M	18.18B	79.82	86.04	57.74	69.84	86.01	91.84	90.38	94.15
DIA-r64	1.19M	19.20B	79.10	85.62	59.91	70.90	87.19	92.26	90.56	94.51

Table 3: Ablation experiments on the adapter rank with 10 incremental tasks.

Method	Params	Flops	ImageNet-R		ImageNet-A		CUB-200		Cifar-100	
			$\mathcal{A}^{10}\uparrow$	$\bar{\mathcal{A}}^{10}\uparrow$	$\mathcal{A}^{10}\uparrow$	$\bar{\mathcal{A}}^{10}\uparrow$	$\mathcal{A}^{10}\uparrow$	$\bar{\mathcal{A}}^{10}\uparrow$	$\mathcal{A}^{10}\uparrow$	$\bar{\mathcal{A}}^{10}\uparrow$
DIA-MLP	0.17M	17.91B	79.03	85.61	59.78	70.43	86.73	92.13	90.80	94.28
DIA-MHSA	0.51M	18.56B	78.50	85.05	57.27	69.12	83.29	90.56	90.33	94.17
DIA-MIX	0.69M	18.9B	79.69	85.41	58.21	69.93	85.68	92.38	90.85	94.30

Table 4: Ablation experiments on the adapter structure with 10 incremental tasks.

Adapter Rank Ablation

We evaluate the model's accuracy across four datasets by varying the adapter down-projection dimensions to 8, 16, and 64. The results show that increasing the number of parameters does not lead to significant accuracy improvements, with the r64 configuration yielding less than a one-point gain over r08. This demonstrates that the importance of model architecture and training strategies outweighs that of merely increasing the number of parameters.

Structure Ablation

We explore the impact of inserting the TSAI module in parallel within both the MHSA and MLP structures of the transformer block. Specifically, we integrate TSAI in parallel with the three QKV projection layers of MHSA. As shown in Table 4, despite tripling the number of trainable parameters per task, DIA-MHSA still slightly underperforms compared to DIA-MLP. We attribute this to the multi-head attention mechanism's role in capturing input sequence dependencies, where its complex structure may be disrupted by the addition of adapters, thereby increasing optimization difficulty. The experimental results further validate the rationality of our current model structure.