LoRA Subtraction for Drift-Resistant Space in Exemplar-Free Continual Learning

Xuan Liu[1], Xiaobin Chang*[1,2,3]

[1]School of Artificial Intelligence, Sun Yat-sen University, China
[2]Key Laboratory of Intelligent Assessment Technology for Sustainable Tourism, Ministry of Culture and Tourism, Sun Yat-sen University
[3]Key Laboratory of Machine Intelligence and Advanced Computing, Ministry of Education, China
liux687@mail2.sysu.edu.cn, changxb3@mail.sysu.edu.cn

Abstract

In continual learning (CL), catastrophic forgetting often arises due to feature drift. This challenge is particularly prominent in the exemplar-free continual learning (EFCL) setting, where samples from previous tasks cannot be retained, making it difficult to preserve prior knowledge. To address this issue, some EFCL methods aim to identify feature spaces that minimize the impact on previous tasks while accommodating new ones. However, they rely on static features or outdated statistics stored from old tasks, which prevents them from capturing the dynamic evolution of the feature space in CL, leading to performance degradation over time. In this paper, we introduce the Drift-Resistant Space (DRS), which effectively handles feature drifts without requiring explicit feature modeling or the storage of previous tasks. A novel parameter-efficient fine-tuning approach called Low-Rank Adaptation Subtraction (LoRA$^-$) is proposed to develop the DRS. This method subtracts the LoRA weights of old tasks from the initial pre-trained weight before processing new task data to establish the DRS for model training. Therefore, LoRA$^-$ enhances stability, improves efficiency, and simplifies implementation. Furthermore, stabilizing feature drifts allows for better plasticity by learning with a triplet loss. Our method consistently achieves state-of-the-art results, especially for long task sequences, across multiple datasets.[1]

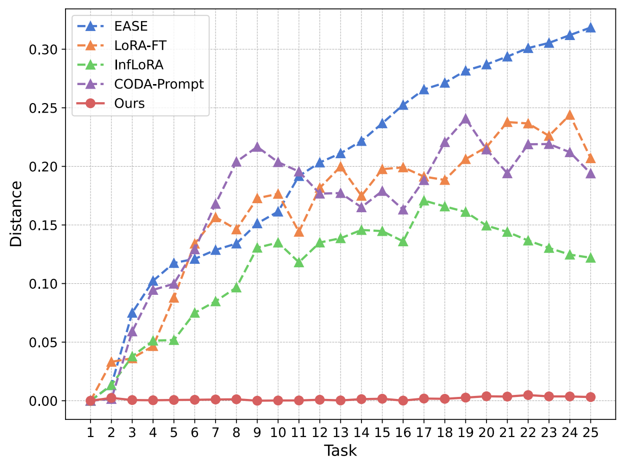

Figure 1. Illustrating the old task feature drifts of different EFCL methods. Imagenet-R dataset under 25 incremental tasks are used.

1. Introduction

Continual learning (CL) methods [5, 27] aim to enable deep models to acquire new knowledge continually, such as recognizing new object categories, while handling catastrophic forgetting [16, 25] of previously learned information. Rehearsal-based methods [2, 3, 8, 30, 43] alleviate forgetting by storing a small subset of past task exemplars and replaying them alongside new data during training. However, access to such exemplars is often restricted due to privacy concerns or memory limitations. Therefore, the study of exemplar-free continual learning (EFCL) becomes necessary. This setting is challenging as no sample of previous tasks can be retained. Without rehearsal, feature representations of old tasks drift over time, leading to severe catastrophic forgetting, as shown in Fig. 1.

Conventional architecture-based EFCL methods [4, 9, 35, 42] involve freezing certain parameters that were trained for the initial task while either adjusting other parameters or expanding the backbone for subsequent tasks. However,

*indicates corresponding author.
[1]Code is available at https://github.com/scarlet0703/LoRA-Sub-DRS.

Table 1. Average accuracy of PEFT methods on Imagenet-R dataset.

Method	Task		
	10	25	50
LoRA-FT	76.03	67.13	58.56
CODA-Prompt	76.95	68.74	52.79
InfLoRA	75.10	74.73	66.27
EASE	81.27	79.32	75.35
LoRA$^-$DRS (Ours)	**81.52**	**79.79**	**77.93**

they require substantial memory consumption and significant computational resources. To handle this, we adopt recent strategies [22] that utilize parameter-efficient fine-tuning (PEFT) [10, 33, 40] to enhance both the effectiveness and efficiency of the EFCL model. Specifically, our model is built on a frozen backbone pre-trained on a large-scale isolated dataset, providing extensive general knowledge. Moreover, low-rank adaptation (LoRA) [14] is integrated to support efficient learning for downstream tasks.

Incorporating the PEFT techniques, e.g., LoRA [14] and prompt tuning [21], in EFCL methods alone may not be sufficient to handle catastrophic forgetting. This is demonstrated by the increasing feature drifts shown in Fig. 1 and the inferior results displayed in Tab. 1. To mitigate feature drifts, existing methods [18, 22, 37] create feature subspaces that aim at preventing the learning of new tasks from interfering with the old ones. Despite their improved overall performance indicated in Tab. 1, these methods still exhibit noticeable feature drifts, as shown by the upward trends in Fig. 1. The key to addressing this issue lies in capturing the dynamic evolution of the feature space for old tasks in CL. However, in the EFCL setting, this is challenging because only static features and statistics from old tasks are available, not to mention the additional memory required to store such data. Therefore, the subspaces created by these methods do not effectively minimize interference from new task learning, especially regarding older tasks that are based on more outdated information.

In this paper, we present Drift-Resistant Space (DRS) for model training in the EFCL setting. As illustrated in Fig. 1, our method maintains a consistently low and stable curve across tasks, indicating effective management of feature drifts. Therefore, this approach significantly mitigates catastrophic forgetting, as evidenced by its superior stability and overall performance in Tab. 1. To avoid relying on static information from old tasks, we propose a novel approach called LoRA Subtraction (LoRA$^-$) for establishing the DRS. Specifically, we first subtract the layer-wise LoRA weights associated with previous tasks from the initial pre-trained weights. The modified model then processes data from new tasks to obtain the projection matrix for the DRS. This simple weight subtraction operation allows the model to "forget" or "unlearn" the knowledge of the corresponding tasks [15]. Consequently, training networks in DRS can focus more on learning new tasks while effectively reducing interference with older ones. The contributions of this work are summarized as follows:

- We introduce DRS, a new space for model training in the EFCL setting that effectively resolves feature drifts and enhances stability in continual learning.
- LoRA$^-$ efficiently establishes DRS without requiring explicit feature modeling or storing static old data. Instead, it aims to reduce the influence of previous tasks in the parameter space.
- Our method LoRA$^-$DRS is less affected by feature drifts, allowing us to use a triplet loss based on prototypes from previous tasks to further enhance its plasticity.

Extensive experiments are conducted in a challenging continual learning setting known as exemplar-free class incremental learning (EFCIL). The proposed method achieves state-of-the-art performance, particularly for long sequences of learning tasks across multiple datasets, highlighting its effectiveness and efficiency.

2. Related Work

2.1. Exemplar-Free Class-Incremental Learning

Class-Incremental Learning (CIL) is a well-established paradigm in continual learning, designed to learn new classes while reducing catastrophic forgetting [1, 17, 46]. Early Computer Vision methods preserved knowledge by storing raw data or features, but rising privacy concerns and storage limits have spurred interest in Exemplar-Free Class-Incremental Learning (EFCIL). One notable approach is LwF [7], which mitigates forgetting by constraining the current model's outputs to stay close to those of the previous model. PASS [51] improves on this by incorporating self-supervised learning in the backbone, followed by functional regularization and feature rehearsal. SSRE [52] proposes a novel architecture design that facilitates the transfer of invariant knowledge across tasks. In FeTRIL [29], the authors freeze the feature extractor and leverage the variance in the current task's data to estimate the location of old class features. FeCAM [11] adopts a similar strategy by using the mean and covariance of previous task features, proposing a Mahalanobis distance-based classifier for better task separation. In this paper, we adopt the EFCIL setting, where old data cannot be stored due to privacy concerns or device limitations, and aim to advance methods that enable continual learning without reliance on exemplars.

2.2. Parameter-Efficient Fine-Tuning

The advent of large-scale pre-trained models (PTMs) has sparked significant interest in fine-tuning techniques for downstream tasks. Traditional continual learning (CL) trains models from scratch, but recent work explores CL with PTMs [38, 41, 48, 50], leveraging their strong feature representations. Parameter-Efficient Fine-Tuning (PEFT) avoids full PTM fine-tuning by inserting and tuning specific sub-modules while keeping pre-trained parameters frozen. Among the most widely used techniques in PEFT are LoRA [14] and Prompt [21], which have shown strong performance in CL [10, 33, 39]. L2P [40] builds on PTMs by learning additional dynamic prompts that guide the model to solve specific tasks. DualPrompt [39] introduces two separate prompt spaces—general and expert prompts—encoding task-invariant and task-specific instructions, respectively. CODAPrompt [33]presents a decomposed attention-based continual learning method that offers greater capacity for learning than previous prompt-based approaches [39, 40]. In contrast, LAE [10] and ADAM [48] introduce unified frameworks for PEFT through model ensembling. SLCA [47] builds upon the Gaussian modeling of old task features proposed by [51], using it to adjust classifiers. However, all these methods rely on fixing the parameters of previous tasks to maintain stability. While they fail to address the issue of feature drift, which prevents the model from maintaining good performance and avoiding forgetting over long-term tasks.

2.3. Mitigating Feature Drift

A critical aspect in EFCIL is the feature drift in old classes when new tasks are learned without old samples. Some approaches aim to correct feature drift. For example, SDC [44] and [34] estimate and compensate for feature drift with current-task samples after each incremental phase. Methods like NAPA-VQ [24] and Prototype Reminiscence [32] enhance or reshape old prototypes using topological information or interpolation with new samples. ADC [12] generates adversarial samples as pseudo-exemplars to measure drift. FeTrIL [29] assumes similar feature distributions across all classes and use prototypes as the nearest feature center of new classes. EASE [49] designs a semantic-guided prototype complement strategy that recalculates prototypes in the new feature space without relying on old exemplars. Other approaches focus on minimize the interference of the new task on the old tasks. InfLoRA [22] uses gradient information from old tasks to design a subspace for LoRA's dimensionality reduction matrix, reducing the impact of new tasks on old ones. Adam-NSCL [37] and AdNS [18] optimizes network parameters by using input features from previous tasks to obtain the approximate null space of old tasks. However, despite the impressive performance achieved by these methods, several challenges remain in effectively controlling feature drift. These approaches assume that the model learns new concepts in an interference-free space of previous tasks. Since this space is computed based on stored, static statistics of old tasks, it becomes difficult to accurately model the highly dynamic feature space in continual learning. As the number of tasks increases, those statistics become increasingly outdated, leading to a noticeable decline in the performance of these methods.

3. Methodology

3.1. Preliminary

Exemplar-free class-incremental learning (EFCIL) is designed to address the challenges of continual learning without exemplars, where the model must learn from a data stream of T sequential tasks, indexed as $\{1, 2, \dots, T\}$, with non-overlapping class distributions. For each task t, the dataset is defined as $D_t = \{(x_{i,t}, y_{i,t})\}_{i=1}^{n_t}$, where $x_{i,t}$ is an input sample, $y_{i,t}$ is its label, and n_t is the number of instances in task t. An instance $x_{i,t}$ belongs to class $y_{i,t}$, where $y_{i,t} \in Y_t$ and Y_t is the label space of task t, and $Y_t \cap Y_{t'} = \emptyset$ for $t \neq t'$, i.e., non-overlapping classes for different tasks. During training on task t, only the dataset D_t is accessible, and no samples from previous tasks can be stored or used.

Following the framework of parameter-efficient fine-tuning (PEFT) for continual learning [36, 39] , we assume the model $\mathcal{M}_{\boldsymbol{\theta}}$ is a pre-trained model with weights $\boldsymbol{\theta}$. Instead of updating the model parameters directly, Low-Rank Adaptation (LoRA) incorporates auxiliary low-rank matrices $\boldsymbol{B}$ and $\boldsymbol{A}$ into the model's linear layers, while keeping the pre-trained weight matrix in each self-attention block $\boldsymbol{W}_0$ fixed. Assume $f(\cdot, \boldsymbol{W})$ denotes the linear transformation layer with parameters $\boldsymbol{W} \in \mathbb{R}^{d\times d}$.

Our approach consists of two main stages. In the first stage, before learning the t-th new task, the parameters $\boldsymbol{W}_{t-1}$ are frozen, and a LoRA branch $\Delta\boldsymbol{W}_t = \boldsymbol{B}_t\boldsymbol{A}_t$ is expanded, where $\boldsymbol{A}_t \in \mathbb{R}^{r\times d}$ is a dimensionality reduction matrix and $\boldsymbol{B}_t \in \mathbb{R}^{d\times r}$ is a dimensionality expansion matrix, with rank $r \ll d$. To make the learning process resistant to feature drift, we apply LoRA Subtraction to obtain a drift-resistant space (DRS), which stabilizes the learning of new tasks while minimizing interference with previously learned knowledge.

In the second stage, we train the LoRA parameters $\boldsymbol{A}_t$ and $\boldsymbol{B}_t$ for the current task t in the DRS. Specifically, during training, when the data D_t from task t is fed to the model, the input and output features at the l-th linear layer within a self-attention block are denoted by $\boldsymbol{X}_t^l$ and $\boldsymbol{Z}_t^l$, respectively. The forward propagation in this layer can be represented by

$$\boldsymbol{Z}_t^l = f(\boldsymbol{X}_t^l, \boldsymbol{W}_t^l), \tag{1}$$

$$\boldsymbol{X}_t^{l+1} = \sigma_l(\boldsymbol{Z}_t^l), \tag{2}$$

$$\boldsymbol{W}_t^l = \boldsymbol{W}_{t-1}^l + \Delta\boldsymbol{W}_t^l = \boldsymbol{W}_0^l + \sum_{j=1}^{t} \boldsymbol{B}_j^l\boldsymbol{A}_j^l, \tag{3}$$

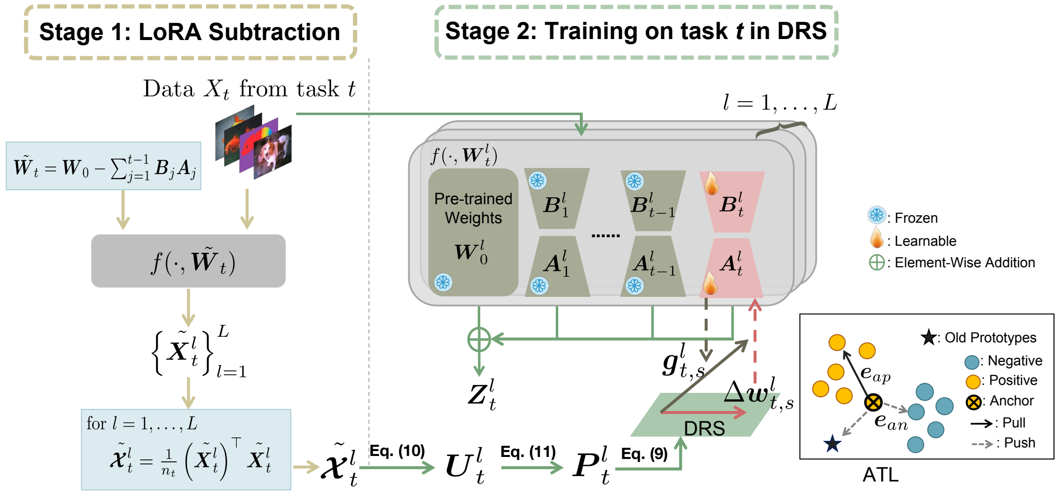

Figure 2. The training pipeline of the proposed LoRA Subtraction for Drift-Resistant Space. Before training on the t-th task, LoRA subtraction is applied to construct the drift-resistant space (DRS). During training, pre-trained weights and previously learned LoRAs are frozen. A_t and B_t of the current task is learned by projecting gradients into DRS, with augmented triplet loss (ATL) enhancing plasticity.

where σ is a nonlinear activation function, $l = 1, \ldots, L$, and $\boldsymbol{X}_t^1 = X_t$. When training on task t at the s-th training step, the parameter update across the L linear layers is denoted as $\Delta\boldsymbol{w}_{t,s} = \{\Delta\boldsymbol{w}_{t,s}^1, \Delta\boldsymbol{w}_{t,s}^2, \ldots, \Delta\boldsymbol{w}_{t,s}^L\}$.

The complete pipeline of our method is illustrated in Fig. 2.

3.2. Stage 1: LoRA Subtraction

The core objective of LoRA subtraction (LoRA$^-$) is to effectively remove information from previous tasks to create a drift-resistant space. Recent studies [15] have shown that a task vector can direct specific changes in the weight space of a pre-trained model. Let $\boldsymbol{\theta}_0$ denote the weights of the pre-trained model and $\boldsymbol{\theta}_t$ the weights after fine-tuning on task t. The task vector for task t is given by

$$\boldsymbol{V}_t = \boldsymbol{\theta}_t - \boldsymbol{\theta}_0. \tag{4}$$

By negating task vectors, performance on specific tasks can be reduced or forgotten with minimal impact on unrelated tasks.

Building on this insight, LoRA$^-$ controls feature drift by selectively negating prior task influences. After training on task $t-1$, the task vector $\boldsymbol{V}_{t-1}^l$ for each linear layer $l = 1, \ldots, L$ is defined as

$$\boldsymbol{V}_{t-1}^l = \boldsymbol{W}_{t-1}^l - \boldsymbol{W}_0^l = \sum_{j=1}^{t-1} \boldsymbol{B}_j^l \boldsymbol{A}_j^l. \tag{5}$$

To achieve forgetting of specific old tasks, LoRA$^-$ applies a negated cumulative task vector $-\boldsymbol{V}_{t-1}$ to the pre-trained weights before training on the new task t

$$\tilde{\boldsymbol{W}_t}^l = \boldsymbol{W}_0^l - \boldsymbol{V}_{t-1}^l = \boldsymbol{W}_0^l - \sum_{j=1}^{t-1} \boldsymbol{B}_j^l \boldsymbol{A}_j^l. \tag{6}$$

We then define a drift-resistant space (DRS) for the new task, ensuring that learning task t does not affect the feature distribution of prior tasks. Existing works [31, 37, 45] have shown that the gradient update in a linear layer lies within the span of the input data. Therefore, we leverage the input matrix of the new task t under $\tilde{\boldsymbol{W}}_t$ to define a learning space that captures features specific to the new task without disturbing previous knowledge.

Before training on task t, we feed D_t to the model with LoRA$^-$ linear layer $f(X_t, \tilde{\boldsymbol{W}}_t^l)$ to obtain the input feature $\tilde{\boldsymbol{X}}_t^l$ at each layer l. Following [37], we compute the uncentered covariance matrix of these input features as follows

$$\tilde{\boldsymbol{\mathcal{X}}}_t^l = \frac{1}{n_t}\left(\tilde{\boldsymbol{X}}_t^l\right)^\top \tilde{\boldsymbol{X}}_t^l, \tag{7}$$

where n_t is the number of samples in task t, and $\tilde{\boldsymbol{\mathcal{X}}}_t^l$ captures input learning space at layer l for task t, playing a key role in defining DRS for stable task-specific learning.

3.3. Stage 2: Training in Drift-Resistant Space

In this section, we introduce a network training algorithm designed for learning sequential tasks in a drift-resistant space (DRS). Our approach ensures that each new task can be learned with minimal interference from previously learned tasks, maintaining a stable feature representation.

Algorithm 1 LoRA Subtraction for Drift-Resistant Space

Inputs: Datasets D_t for task $t \in \{1, 2, \dots, T\}$, a pre-trained ViT model $\mathcal{M}_{\boldsymbol{\theta}}$ with L linear layers $f(\cdot, \boldsymbol{W})$, loss function $\mathcal{L}_{\text{total}}$, learning rate α.
Initialization: Initialized parameters $\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{\theta}, \boldsymbol{W}$.

for task $t \in \{1, 2, \dots, T\}$ **do**
  if $t > 1$ **then**
    `# Stage 1: Compute DRS`
    Compute $\tilde{\boldsymbol{\mathcal{X}}}_t^l$ through LoRA Subtraction with D_t and $f(\cdot, \tilde{\boldsymbol{W}}_t^l)$ as inputs using Eq. (7).
  end if
  `# Stage 2: Train` A_t `and` B_t `in DRS`
  Expand a LoRA branch $\Delta \boldsymbol{W}_t = \boldsymbol{B}_t \boldsymbol{A}_t$.
  Compute $\boldsymbol{P}_t^l$ for each layer $l = 1, \dots, L$ using $\tilde{\boldsymbol{\mathcal{X}}}_t^l$ and Eq. (11).
  Set $s = 0$ and $\boldsymbol{W}_{t,0} = \boldsymbol{W}_{t-1} + \Delta \boldsymbol{W}_{t,0}$;
  while not converged **do**
    Sample a batch $\{X_t^b, Y_t^b\}$ from D_t.
    Compute loss $\mathcal{L}_{\text{total}}$ using $\{X_t^b, Y_t^b\}$ and $\boldsymbol{W}_{t,s}$ based on Eq. (16).
    Get candidate parameter update $\boldsymbol{g}_{t,s} = \{\boldsymbol{g}_{t,s}^1, \dots, \boldsymbol{g}_{t,s}^L\}$ with respect to $\mathcal{L}_{\text{total}}$ with $\{x_t, y_t\}$ using Adam.
    if $t = 1$ **then**
      $\Delta \boldsymbol{w}_{t,s}^l = \boldsymbol{g}_{t,s}^l, \quad l = 1, \dots, L$
    else
      $\Delta \boldsymbol{w}_{t,s}^l = \boldsymbol{P}_t^l \left(\boldsymbol{P}_t^l\right)^\top \boldsymbol{g}_{t,s}^l, \quad l = 1, \dots, L$
    end if
    $\Delta \boldsymbol{W}_{t,s+1}^l = \Delta \boldsymbol{W}_{t,s}^l - \alpha \Delta \boldsymbol{w}_{t,s}^l, \quad l = 1, \dots, L$
    $s = s + 1$
  end while
end for

When training on a new task t, the LoRA weights $\Delta \boldsymbol{W}_t$ are updated iteratively at each training step s, with the parameter update at each linear layer represented as

$$\Delta \boldsymbol{w}_{t,s} = \{\Delta \boldsymbol{w}_{t,s}^1, \Delta \boldsymbol{w}_{t,s}^2, \dots, \Delta \boldsymbol{w}_{t,s}^L\}, \tag{8}$$

where $\Delta \boldsymbol{w}_{t,s}^l$ corresponds to the parameter updates in the LoRA branch for layer l during training step s.

To mitigate feature drift, we project the gradients at each training step into a DRS before updating the weights. DRS is designed to preserve knowledge from previous tasks by restricting updates to directions that minimally interfere with learned representations. Specifically, for each layer l, we denote the gradients at step s as $\boldsymbol{g}_{t,s}^l$ and project them into DRS using a projection operator $\boldsymbol{P}_t^l$. The projected update for each layer l is computed as follows

$$\Delta \boldsymbol{w}_{t,s}^l = \boldsymbol{P}_t^l \left(\boldsymbol{P}_t^l\right)^\top \boldsymbol{g}_{t,s}^l, \tag{9}$$

where $\boldsymbol{P}_t^l \left(\boldsymbol{P}_t^l\right)^\top$ acts as a DRS projection operator.

Specifically, the DRS projection operator is obtained by applying Singular Value Decomposition (SVD) to the feature matrix $\tilde{\boldsymbol{\mathcal{X}}}_t^l$ from LoRA subtraction (LoRA$^-$). Following methods in [26, 37], we apply SVD to $\tilde{\boldsymbol{\mathcal{X}}}_t^l$

$$\boldsymbol{U}_t^l \boldsymbol{\Lambda}_t^l (\boldsymbol{U}_t^l)^\top = \text{SVD}(\tilde{\boldsymbol{\mathcal{X}}}_t^l), \tag{10}$$

where $\boldsymbol{U}_t^l$ represents the directions of the principal components, $\boldsymbol{\Lambda}_t^l$ indicates the significance of each component, and $(\boldsymbol{U}_t^l)^\top$ is the transpose of $\boldsymbol{U}_t^l$.

To construct the DRS basis for layer l, denoted as $\boldsymbol{P}_t^l$, we select a subset of principal components $(\boldsymbol{U}_t^l)_k$ from $\boldsymbol{U}_t^l$ that correspond to the largest singular values. This selection criterion, inspired by Principal Component Analysis (PCA), captures the most significant directions in $\tilde{\boldsymbol{\mathcal{X}}}_t^l$, where updates have minimal interference with previously learned knowledge. Thus, the DRS projection matrix for each layer l is defined as

$$\boldsymbol{P}_t^l = (\boldsymbol{U}_t^l)_k, \tag{11}$$

where $(\boldsymbol{U}_t^l)_k$ includes only the top k singular vectors, selected based on

$$\min_k \frac{\sum_{i=1}^k \lambda_i}{\sum_{i=1}^d \lambda_i} \geq \varepsilon, \tag{12}$$

where λ_i represents the eigenvalues sorted in descending order, ε denote the cumulative variance threshold.

This DRS-based projection ensures that the updated weights remain in DRS, allowing the model to learn new tasks without degrading performance on prior ones.

The detailed steps for both stages of our approach are outlined in Algorithm 1.

3.4. Model Optimization

While drift-resistant space (DRS) controls feature drift to maintain stability in previously learned knowledge, we found that gradient projection can impair the model's plasticity, hindering its ability to separate new classes in the feature space.

To address this, we propose an Augmented Triplet Loss (ATL), denoted as $\mathcal{L}_{\text{TL}}$, which enhances class separation by maximizing the distance between positive and negative pairs for each anchor sample, thus improving plasticity. It is defined as

$$\mathcal{L}_{\text{TL}} = \max(0, e_{ap} - e_{an} + \epsilon), \tag{13}$$

where e_{ap} and e_{an} are the hardest positive and hardest negative distances, and ϵ is a margin parameter that enforces a desired separation between positive and negative pairs.

For an anchor sample $x_{i,t}$ with label $y_{i,t}$ in the t-th task, the hardest positive distance e_{ap} is the maximum distance between $x_{i,t}$ and other samples with the same label

$$e_{ap} = \max_{j:y_{j,t}=y_{i,t}} E_{i,j}, \tag{14}$$

where $E_{i,j}$ denotes the Euclidean distance metric between sample $x_{i,t}$ and sample $x_{j,t}$.

The hardest negative distance e_{an} considers both samples from the current task with labels $y_{j,t} \neq y_{i,t}$ and prototypes $p_c \in P_{1:t-1}$ from previous tasks, where $P_{1:t-1}$ represents the prototype set of all previous tasks

$$e_{an} = \min\left(\min_{j:y_{j,t}\neq y_{i,t}} E_{i,j}, \min_{c} \|\mathcal{M}_{\boldsymbol{\theta}_t}(x_{i,t}) - p_c\|_2\right), \tag{15}$$

where $\mathcal{M}_{\boldsymbol{\theta}_t}(x_{i,t})$ represents the feature embedding of the anchor sample $x_{i,t}$ under the current model parameters $\boldsymbol{\theta}_t$, and $\|\cdot\|_2$ denotes the Euclidean distance to the prototype p_c.

The total loss $\mathcal{L}_{\text{total}}$ combines cross-entropy loss $\mathcal{L}_{\text{CE}}$ and the Augmented Triplet Loss, balanced by a weighting factor λ

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{CE}} + \lambda \mathcal{L}_{\text{TL}}. \tag{16}$$

4. Experiments

4.1. Experimental Setting

Datasets We evaluate our method on multiple CIL benchmarks that are widely used in PEFT research. ImageNet-R [13] is created by applying artistic transformations to 200 classes from the original ImageNet dataset [6]. This dataset was introduced into the continual learning setting by prior work [39] and has since become a standard benchmark for evaluating continual learning methods based on PEFT. CIFAR100 [19], consisting of 100 classes of small-scale images, is a commonly used benchmark dataset in CIL.

Implementation details The proposed method is implemented in PyTorch [28], following similar approaches to previous works [22, 33, 36, 39, 40]. Specifically, we utilize a pre-trained ViT-B/16-IN21K backbone and train the model with the Adam optimizer, using parameters $\beta_1 = 0.9$ and $\beta_2 = 0.999$. Training is conducted with a batch size of 128, across different datasets: each task is trained for 50 epochs on ImageNet-R, and 20 epochs on CIFAR100. Input images are resized to 224 × 224 and normalized within the range [0, 1]. Following previous work [10, 20, 22], we integrate the LoRA architecture into the key and value components of the ViT attention module. Averaged performance of 5 runs is reported in the main results.

Compared methods We compare against PTM-based methods specifically designed to mitigate feature drift, such as EASE [49], InfLoRA [22], and Adam-NSCL [37]. For Adam-NSCL, we implement it with LoRA on PTM. In addition, we select several representative PEFT-based EFCIL methods for comparison, including LAE [10], L2P [40], DualPrompt [39], and CODA-Prompt [33]. We also establish a baseline, denoted as LoRA-FT, which sequentially finetunes the PTM using LoRA. For all methods, their best performance results are reproduced under our experimental settings to ensure fair and direct comparison.

Evaluation metrics Following established practices in continual learning [10, 23, 49], we evaluate the model's performance through three key metrics: accuracy after the last stage(ACC), average accuracy ($\overline{ACC}$) and backward transfer (BWT). For accuracy, we denote ACC_t as the model's accuracy after completing the t-th task. To capture the overall performance, we use ACC_T, which represents the accuracy after the final task, and the average accuracy $\overline{ACC_T}$ across all tasks, defined as:

$$\overline{ACC_T} = \frac{1}{T}\sum_{t=1}^{T} ACC_t, \tag{17}$$

Backward transfer value assesses the effect of learning new tasks on previously learned tasks, capturing the average impact on prior knowledge retention in continual learning. A negative BWT value indicates forgetting, as the model's performance on prior tasks deteriorates when new tasks are added. We compute BWT as:

$$\text{BWT} = \frac{1}{T-1}\sum_{i=1}^{T-1}\left(A_{T,i} - A_{i,i}\right), \tag{18}$$

where $A_{T,i}$ denotes the accuracy on the i-th task after training on the final task T.

4.2. Main Results

In this section, we compare LoRA$^-$DRS with other state-of-the-art PEFT methods on CIL benchmarks, testing across various splits and backbone weights. As shown in Tab. 2, the accuracy results of different methods on ImageNet-R are presented for different numbers of tasks. Similarly, Tab. 3 reports results on CIFAR100. Our method consistently outperforms existing EFCIL methods and other feature drift control methods, such as EASE, InfLoRA, and Adam-NSCL. Notably, as the number of tasks increases, the advantage of our method becomes more pronounced, achieving an accuracy margin of 2.58% to 3.81% over the runner-up method by the 50th task, highlighting its suitability for continual learning scenarios, where maintaining performance over prolonged task sequences is essential.

To evaluate model stability, we assess the Backward Transfer (BWT) values of different methods on CIFAR-100 and ImageNet-R with 25 tasks. As shown in Tab. 4, LoRA$^-$DRS achieves a competitive BWT score of -3.38

Table 2. Averaged results (mean ± standard deviation) of different methods trained under five random seeds of ImageNet-R.

Task	10		20		25		50	
Method	ACC_{10}	$\overline{ACC}_{10}$	ACC_{20}	$\overline{ACC}_{20}$	ACC_{25}	$\overline{ACC}_{25}$	ACC_{50}	$\overline{ACC}_{50}$
LoRA-FT	74.54±6.02	73.43±4.52	60.71±2.39	72.53±1.72	56.80±3.06	70.06±1.74	44.89±2.29	59.25±1.40
LAE [10]	72.56±0.98	78.56±0.47	60.53±1.73	76.64±0.91	68.05±1.88	75.40±1.31	62.83±2.17	70.83±1.63
L2P [40]	64.94±0.90	70.33±1.90	62.15±1.17	68.35±2.12	60.85±1.11	67.17±2.46	55.89±1.59	62.98±2.89
DualPrompt [39]	62.24±0.29	74.63±0.94	66.89±0.40	73.07±1.21	65.76±0.67	72.15±1.19	61.50±0.86	68.63±1.31
CODA-Prompt [33]	72.15±0.12	77.51±0.75	67.53±0.24	73.64±0.95	63.86±0.81	70.47±1.72	48.89±0.90	55.59±2.67
InfLoRA [22]	74.95±0.90	80.99±0.89	71.46±0.95	78.32±1.22	69.09±0.93	76.84±1.32	60.49±1.43	69.95±2.18
Adam-NSCL [37]	72.24±0.69	78.85±1.07	65.52±1.11	72.79±1.39	62.04±1.74	69.69±1.81	49.82±2.67	58.49±3.85
EASE [49]	**75.94**±0.46	**81.67**±0.32	73.78±0.44	80.29±0.72	72.69±0.39	79.65±0.65	68.54±0.71	75.77±0.58
LoRA$^-$DRS (Ours)	74.74±0.78	81.16±0.59	**74.80**±0.73	**80.69**±0.75	**74.19**±0.46	**80.06**±0.76	**72.12**±0.87	**77.94**±0.74

Table 3. Averaged results (mean ± standard deviation) of different methods trained under five random seeds of CIFAR-100.

Task	10		20		25		50	
Method	ACC_{10}	$\overline{ACC}_{10}$	ACC_{20}	$\overline{ACC}_{20}$	ACC_{25}	$\overline{ACC}_{25}$	ACC_{50}	$\overline{ACC}_{50}$
LoRA-FT	82.43±1.47	89.30±1.00	74.51±1.58	84.10±0.95	68.33±3.05	80.81±1.02	43.77±5.06	63.45±2.90
LAE [10]	84.99±0.75	89.36±0.84	83.51±0.36	88.15±0.59	82.20±0.58	87.02±0.87	77.68±2.21	83.16±1.10
L2P [40]	83.41±0.60	88.98±0.31	80.72±1.12	87.18±0.83	79.54±0.72	86.69±0.65	73.91±1.67	81.90±0.98
DualPrompt [39]	86.40±0.62	91.26±1.29	83.82±0.51	90.22±0.68	82.65±1.28	89.44±1.04	76.66±0.74	85.18±0.92
CODA-Prompt [33]	87.02±0.17	91.39±0.23	81.19±0.31	87.27±0.35	77.18±0.49	84.56±0.31	55.45±0.48	68.39±0.53
InfLoRA [22]	86.44±0.81	91.16±0.79	82.19±1.33	88.05±0.64	77.51±1.19	85.02±1.61	56.65±5.55	70.29±1.86
Adam-NSCL [37]	85.39±0.75	89.67±0.53	79.69±1.49	85.52±1.10	75.19±1.61	81.29±1.90	55.02±1.52	66.27±1.34
EASE [49]	88.34±0.52	92.27±0.53	82.21±0.72	90.76±0.54	85.01±0.69	89.98±0.65	82.10±0.66	87.65±0.46
LoRA$^-$DRS (Ours)	**89.14**±0.23	**92.55**±0.25	**88.69**±0.15	**92.25**±0.24	**88.39**±0.33	**92.02**±0.19	**87.29**±0.31	**91.29**±0.29

Table 4. Backward transfer (BWT) for different methods on Imagenet-R under 25 incremental tasks.

Method	CIFAR-100	Imagenet-R
LoRA-FT	-13.08	-27.52
LAE [10]	-7.06	-6.32
L2P [40]	-7.35	-8.85
DualPrompt [39]	-7.25	-6.26
CODA-Prompt [33]	-5.59	-4.24
InfLoRA [22]	-3.70	-7.56
Adam-NSCL [37]	-4.39	-3.91
EASE [49]	-5.54	-4.53
LoRA$^-$DRS (Ours)	**-3.38**	**-3.90**

Table 5. The effectiveness of each component in our method on Imagenet-R. Averaged accuracy ($\overline{ACC}$) is reported.

DRS	ATL	Task		
		10	25	50
✗	✗	76.03	67.13	58.56
✓	✗	80.32	78.00	76.03
✓	✓	81.52	79.79	77.93

Table 6. Averaged new class accuracy (AA_{new}) on Imagenet-R.

Method	Task		
	10	25	50
DRS	79.68	77.31	75.83
+ATL	**81.23** (1.55↑)	**79.18** (1.87↑)	**77.88** (2.05↑)

on CIFAR-100 and −3.90 on Imagenet-R, indicating effective feature drift control and consistent performance across tasks. Notably, while methods like InfLoRA and Adam-NSCL also achieve strong BWT scores, they rely on storing statistics from previous tasks to maintain stability, while our method does not.

4.3. Detailed Analysis

Drift evaluation To further verify that our method effectively controls feature drift, we evaluate it on the ImageNet-R 25 tasks setting and compare feature drift across tasks on previously learned classes. As shown in Fig. 1, the feature drifts of different methods can be illustrated by measuring the shift in feature representations over tasks. Specifically, we compute the average squared Euclidean distance between the class center of task 1 data, as extracted by the network at task t, and the corresponding class prototypes from the first task. The distance for the current state-of-the-art PEFT models tends to increase as tasks progress. Furthermore, InfLoRA attempts to reduce interference, but it still shows persistent feature drift due to reliance on outdated stored task statistics. In contrast, our LoRA$^-$DRS maintains a steady feature drift curve, demonstrating effective drift control.

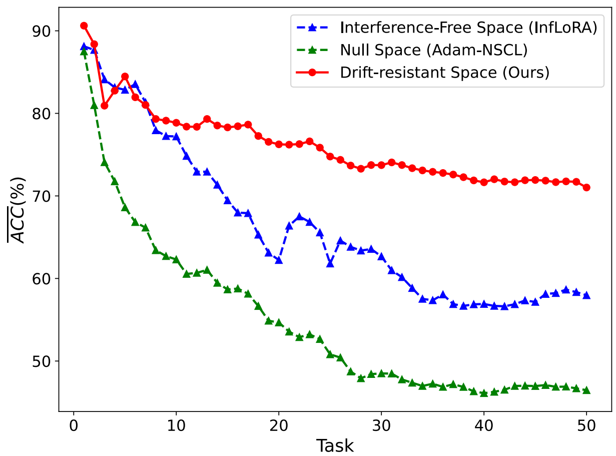

Figure 3. Performance comparison of different space designs on Imagenet-R across 50 incremental tasks.

Ablation Study In this section, we conduct an ablation study to investigate the effectiveness of each component in LoRA$^-$DRS, including Drift-Resistant Space (DRS) and Augmented Triplet Loss (ATL). Specifically, we report the incremental performance of different variations on ImageNet-R.

As shown in Tab. 5, the performance of training without DRS, relying solely on LoRA for adapting to new tasks, is significantly lower than our method. At the 50-task mark, the model without DRS achieves an $\overline{ACC}$ that is 17.47% lower than with DRS. This result suggests that, without the DRS intervention, the model experiences severe forgetting due to feature drift. Additionally, we assess the impact of our ATL on model performance. Tab. 5 shows that incorporating the ATL in the full LoRA$^-$DRS model leads to an improvement of 1.2% to 1.9% in $\overline{ACC}$ compared to the variant without ATL under the same task setting. Tab. 6 further show that ATL significantly improves the model's performance on *new* classes. Therefore, ATL enhances the model's ability to obtain discriminative features, contributing to better plasticity and accuracy.

Further Analysis To validate the effectiveness of our DRS, we compared it with two other space designing methods, Interference-Free space (InfLoRA) and Null Space (Adam-NSCL), on ImageNet-R across 50 incremental tasks. As shown in Fig. 3, which presents the ACC values for each task, our DRS demonstrates superior stability and resilience against performance degradation over longer task sequences. While all methods show some accuracy decline with more tasks, DRS consistently outperforms, maintaining higher accuracy and a more controlled decline. InfLoRA and Adam-NSCL, despite high initial accuracy, experience a marked drop as tasks progress, suggesting their sensitivity to cumulative drift and limitations in preserving old task knowledge. In contrast, our LoRA$^-$DRS effectively mitigates feature drift over time, making it well-suited for long-term incremental learning scenarios.

Table 7. Results of InfLoRA with ATL on Imagenet-R. The effectiveness of ATL depends on controlling feature drift, as unmanaged drift can undermine model's performance.

	$\overline{ACC_{10}}$	$\overline{ACC_{25}}$	$\overline{ACC_{50}}$
InfLoRA	79.93	74.73	66.27
+ATL	79.85↓	71.58↓	61.46↓

During the training phase, we introduced ATL to enhance class separation, aiming to achieve greater plasticity when learning new classes. However, the effectiveness of ATL relies on effectively controlling feature drift. If drift occurs, the feature space for previous tasks may shift, undermining ATL's ability to separate new class samples from old ones. As shown in Tab. 7, comparing InfLoRA with and without ATL on ImageNet-R reveals a performance drop when ATL is applied, and this drop becomes more pronounced as the number of tasks increases. The $\overline{ACC}$ decreases by 4.81% after 50 tasks, highlighting that current methods struggle to control feature drift over long task sequences. Without effective drift control, ATL fails to enhance class separation, negatively impacting overall performance.

5. Conclusion

In conclusion, this paper introduces LoRA$^-$DRS for effective model training in the exemplar-free continual learning setting. Our Drift-Resistant Space (DRS) strategy successfully mitigates feature drift, resulting in enhanced stability and reduced catastrophic forgetting, as demonstrated by consistent performance across long tasks. Through the novel Low-Rank Adaptation Subtraction (LoRA$^-$) approach, DRS minimizes interference from prior tasks without requiring static data storage, allowing the model to focus on new tasks effectively. By enabling Augmented Triplet Loss (ATL) integration to enhance class separation, our method further bolsters learning plasticity. Extensive experiments confirm the state-of-the-art performance of LoRA$^-$DRS across long task sequences in the EFCIL setting, underscoring its effectiveness and adaptability.

Limitations While Low-Rank Adaptation (LoRA) modules reduce parameter counts compared to full fine-tuning, their adapter components still introduce incremental model size overhead. Future research could address this limitation by exploring architectures that seamlessly integrate adaptation mechanisms rather than relying on add-on modules.

Acknowledgment This research was supported by the Natural Science Foundation (NSF) for Young Scientists of China (No.62106289).

References

[1] Hongjoon Ahn, Jihwan Kwak, Subin Lim, Hyeonsu Bang, Hyojun Kim, and Taesup Moon. Ss-il: Separated softmax for incremental learning. In *Proceedings of the IEEE/CVF International conference on computer vision*, pages 844–853, 2021. 2

[2] Lorenzo Bonicelli, Matteo Boschini, Angelo Porrello, Concetto Spampinato, and Simone Calderara. On the effectiveness of lipschitz-driven rehearsal in continual learning. *Advances in Neural Information Processing Systems*, 35: 31886–31901, 2022. 1

[3] Francisco M Castro, Manuel J Marín-Jiménez, Nicolás Guil, Cordelia Schmid, and Karteek Alahari. End-to-end incremental learning. In *Proceedings of the European conference on computer vision (ECCV)*, pages 233–248, 2018. 1

[4] Xiuwei Chen and Xiaobin Chang. Dynamic residual classifier for class incremental learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 18743–18752, 2023. 1

[5] Matthias De Lange, Rahaf Aljundi, Marc Masana, Sarah Parisot, Xu Jia, Aleš Leonardis, Gregory Slabaugh, and Tinne Tuytelaars. A continual learning survey: Defying forgetting in classification tasks. *IEEE transactions on pattern analysis and machine intelligence*, 44(7):3366–3385, 2021. 1

[6] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009. 6

[7] Prithviraj Dhar, Rajat Vikram Singh, Kuan-Chuan Peng, Ziyan Wu, and Rama Chellappa. Learning without memorizing. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5138–5146, 2019. 2

[8] Arthur Douillard, Matthieu Cord, Charles Ollion, Thomas Robert, and Eduardo Valle. Podnet: Pooled outputs distillation for small-tasks incremental learning. In *Computer vision–ECCV 2020: 16th European conference, Glasgow, UK, August 23–28, 2020, proceedings, part XX 16*, pages 86–102. Springer, 2020. 1

[9] Arthur Douillard, Alexandre Ramé, Guillaume Couairon, and Matthieu Cord. Dytox: Transformers for continual learning with dynamic token expansion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9285–9295, 2022. 1

[10] Qiankun Gao, Chen Zhao, Yifan Sun, Teng Xi, Gang Zhang, Bernard Ghanem, and Jian Zhang. A unified continual learning framework with general parameter-efficient tuning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 11483–11493, 2023. 2, 3, 6, 7

[11] Dipam Goswami, Yuyang Liu, Bartłomiej Twardowski, and Joost van de Weijer. Fecam: Exploiting the heterogeneity of class distributions in exemplar-free continual learning. *Advances in Neural Information Processing Systems*, 36, 2024. 2

[12] Dipam Goswami, Albin Soutif-Cormerais, Yuyang Liu, Sandesh Kamath, Bart Twardowski, Joost van de Weijer, et al. Resurrecting old classes with new data for exemplar-free continual learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 28525–28534, 2024. 3

[13] Dan Hendrycks, Steven Basart, Norman Mu, Saurav Kadavath, Frank Wang, Evan Dorundo, Rahul Desai, Tyler Zhu, Samyak Parajuli, Mike Guo, et al. The many faces of robustness: A critical analysis of out-of-distribution generalization. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 8340–8349, 2021. 6

[14] Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*, 2021. 2, 3

[15] Gabriel Ilharco, Marco Tulio Ribeiro, Mitchell Wortsman, Suchin Gururangan, Ludwig Schmidt, Hannaneh Hajishirzi, and Ali Farhadi. Editing models with task arithmetic. *ICLR*, 2023. 2, 4

[16] Ronald Kemker, Marc McClure, Angelina Abitino, Tyler Hayes, and Christopher Kanan. Measuring catastrophic forgetting in neural networks. In *Proceedings of the AAAI conference on artificial intelligence*, 2018. 1

[17] Jong-Yeong Kim and Dong-Wan Choi. Split-and-bridge: Adaptable class incremental learning within a single neural network. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 8137–8145, 2021. 2

[18] Yajing Kong, Liu Liu, Zhen Wang, and Dacheng Tao. Balancing stability and plasticity through advanced null space in continual learning. In *European Conference on Computer Vision*, pages 219–236. Springer, 2022. 2, 3

[19] A Krizhevsky. Learning multiple layers of features from tiny images. *Master's thesis, University of Tront*, 2009. 6

[20] Nupur Kumari, Bingliang Zhang, Richard Zhang, Eli Shechtman, and Jun-Yan Zhu. Multi-concept customization of text-to-image diffusion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1931–1941, 2023. 6

[21] Brian Lester, Rami Al-Rfou, and Noah Constant. The power of scale for parameter-efficient prompt tuning. *arXiv preprint arXiv:2104.08691*, 2021. 2, 3

[22] Yan-Shuo Liang and Wu-Jun Li. Inflora: Interference-free low-rank adaptation for continual learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 23638–23647, 2024. 2, 3, 6, 7, 1

[23] David Lopez-Paz and Marc'Aurelio Ranzato. Gradient episodic memory for continual learning. *Advances in neural information processing systems*, 30, 2017. 6

[24] Tamasha Malepathirana, Damith Senanayake, and Saman Halgamuge. Napa-vq: Neighborhood-aware prototype augmentation with vector quantization for continual learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 11674–11684, 2023. 3

[25] Michael McCloskey and Neal J Cohen. Catastrophic interference in connectionist networks: The sequential learning problem. In *Psychology of learning and motivation*, pages 109–165. Elsevier, 1989. 1

[26] Carl D Meyer. *Matrix analysis and applied linear algebra*. SIAM, 2023. 5

[27] German I Parisi, Ronald Kemker, Jose L Part, Christopher Kanan, and Stefan Wermter. Continual lifelong learning with neural networks: A review. *Neural networks*, 113:54–71, 2019. 1

[28] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32, 2019. 6

[29] Grégoire Petit, Adrian Popescu, Hugo Schindler, David Picard, and Bertrand Delezoide. Fetril: Feature translation for exemplar-free class-incremental learning. In *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, pages 3911–3920, 2023. 2, 3

[30] Sylvestre-Alvise Rebuffi, Alexander Kolesnikov, Georg Sperl, and Christoph H Lampert. icarl: Incremental classifier and representation learning. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 2001–2010, 2017. 1

[31] Andrei A Rusu, Neil C Rabinowitz, Guillaume Desjardins, Hubert Soyer, James Kirkpatrick, Koray Kavukcuoglu, Razvan Pascanu, and Raia Hadsell. Progressive neural networks. *arXiv preprint arXiv:1606.04671*, 2016. 4

[32] Wuxuan Shi and Mang Ye. Prototype reminiscence and augmented asymmetric knowledge aggregation for non-exemplar class-incremental learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1772–1781, 2023. 3

[33] 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. 2, 3, 6, 7

[34] Marco Toldo and Mete Ozay. Bring evanescent representations to life in lifelong class incremental learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16732–16741, 2022. 3

[35] Fu-Yun Wang, Da-Wei Zhou, Han-Jia Ye, and De-Chuan Zhan. Foster: Feature boosting and compression for class-incremental learning. In *European conference on computer vision*, pages 398–414. Springer, 2022. 1

[36] Liyuan Wang, Jingyi Xie, Xingxing Zhang, Mingyi Huang, Hang Su, and Jun Zhu. Hierarchical decomposition of prompt-based continual learning: Rethinking obscured suboptimality. *Advances in Neural Information Processing Systems*, 36, 2024. 3, 6

[37] Shipeng Wang, Xiaorong Li, Jian Sun, and Zongben Xu. Training networks in null space of feature covariance for continual learning. In *Proceedings of the IEEE/CVF conference on Computer Vision and Pattern Recognition*, pages 184–193, 2021. 2, 3, 4, 5, 6, 7

[38] Yabin Wang, Zhiwu Huang, and Xiaopeng Hong. S-prompts learning with pre-trained transformers: An occam's razor for domain incremental learning. *Advances in Neural Information Processing Systems*, 35:5682–5695, 2022. 3

[39] 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. 3, 6, 7

[40] 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 *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 139–149, 2022. 2, 3, 6, 7

[41] Tz-Ying Wu, Gurumurthy Swaminathan, Zhizhong Li, Avinash Ravichandran, Nuno Vasconcelos, Rahul Bhotika, and Stefano Soatto. Class-incremental learning with strong pre-trained models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9601–9610, 2022. 3

[42] Shipeng Yan, Jiangwei Xie, and Xuming He. Der: Dynamically expandable representation for class incremental learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 3014–3023, 2021. 1

[43] Jaehong Yoon, Divyam Madaan, Eunho Yang, and Sung Ju Hwang. Online coreset selection for rehearsal-based continual learning. *arXiv preprint arXiv:2106.01085*, 2021. 1

[44] Lu Yu, Bartlomiej Twardowski, Xialei Liu, Luis Herranz, Kai Wang, Yongmei Cheng, Shangling Jui, and Joost van de Weijer. Semantic drift compensation for class-incremental learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 6982–6991, 2020. 3

[45] Guanxiong Zeng, Yang Chen, Bo Cui, and Shan Yu. Continual learning of context-dependent processing in neural networks. *Nature Machine Intelligence*, 1(8):364–372, 2019. 4

[46] Duzhen Zhang, Xiuyi Chen, Shuang Xu, and Bo Xu. Knowledge aware emotion recognition in textual conversations via multi-task incremental transformer. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 4429–4440, 2020. 2

[47] 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. 3

[48] Da-Wei Zhou, Zi-Wen Cai, Han-Jia Ye, De-Chuan Zhan, and Ziwei Liu. Revisiting class-incremental learning with pre-trained models: Generalizability and adaptivity are all you need. *arXiv preprint arXiv:2303.07338*, 2023. 3

[49] Da-Wei Zhou, Hai-Long Sun, Han-Jia Ye, and De-Chuan Zhan. Expandable subspace ensemble for pre-trained model-based class-incremental learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 23554–23564, 2024. 3, 6, 7, 1

[50] Kaiyang Zhou, Jingkang Yang, Chen Change Loy, and Ziwei Liu. Learning to prompt for vision-language models. *In-*

ternational Journal of Computer Vision, 130(9):2337–2348, 2022. 3

[51] Fei Zhu, Xu-Yao Zhang, Chuang Wang, Fei Yin, and Cheng-Lin Liu. Prototype augmentation and self-supervision for incremental learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5871–5880, 2021. 2, 3

[52] Kai Zhu, Wei Zhai, Yang Cao, Jiebo Luo, and Zheng-Jun Zha. Self-sustaining representation expansion for non-exemplar class-incremental learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9296–9305, 2022. 2