CL-LoRA: Continual Low-Rank Adaptation for Rehearsal-Free Class-Incremental Learning

Jiangpeng He[1] Zhihao Duan[2] Fengqing Zhu[2]
[1] Massachusetts Institute of Technology, Cambridge, Massachusetts, U.S.A.
[2] Purdue University, West Lafayette, Indiana, U.S.A.
jpenghe@mit.edu, {duan90, zhu0}@purdue.edu

Abstract

*Class-Incremental Learning (CIL) aims to learn new classes sequentially while retaining the knowledge of previously learned classes. Recently, pre-trained models (PTMs) combined with parameter-efficient fine-tuning (PEFT) have shown remarkable performance in rehearsal-free CIL without requiring exemplars from previous tasks. However, existing adapter-based methods, which incorporate lightweight learnable modules into PTMs for CIL, create new adapters for each new task, leading to both parameter redundancy and failure to leverage shared knowledge across tasks. In this work, we propose Continua**L** **Low-R**ank **A**daptation (CL-LoRA), which introduces a novel dual-adapter architecture combining **task-shared adapters** to learn cross-task knowledge and **task-specific adapters** to capture unique features of each new task. Specifically, the shared adapters utilize random orthogonal matrices and leverage knowledge distillation with gradient reassignment to preserve essential shared knowledge. In addition, we introduce learnable block-wise weights for task-specific adapters, which mitigate inter-task interference while maintaining the model's plasticity. We demonstrate CL-LoRA consistently achieves promising performance under multiple benchmarks with reduced training and inference computation, establishing a more efficient and scalable paradigm for continual learning with pre-trained models.*

1. Introduction

Modern computer vision models have achieved remarkable progress in various downstream tasks but typically require training on static datasets. However, in real-world scenarios, data often arrives sequentially with new classes gradually becoming available, necessitating Class-Incremental Learning (CIL) [4, 25, 31, 38, 46] that can continuously incorporate new knowledge while retaining previously learned information. A key challenge in CIL is catastrophic forgetting [33], where the model's performance on old tasks dramatically degrades after learning new classes. Recently, the advent of pre-trained models (PTMs) has revolutionized the CIL paradigm by providing robust and generalizable representations learned from large-scale datasets [6, 40]. Through parameter-efficient fine-tuning (PEFT) [19], recent works [41, 48, 49, 57, 58] have demonstrated promising results in CIL even without the need to store old task data as exemplars, marking a significant advancement over traditional rehearsal-based CIL methods with model training from scratch [4, 18, 31, 38, 52].

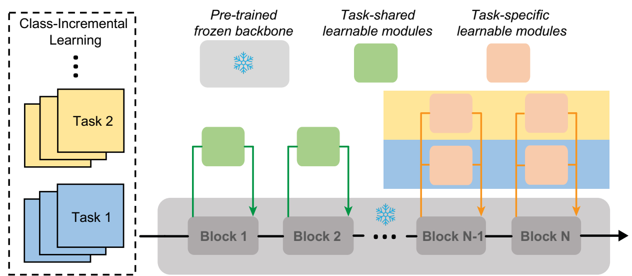

Figure 1. Overview of our dual-adapter architecture. The task-shared learnable modules are continuously updated to capture cross-task knowledge while task-specific modules preserve the unique characteristics of each individual task.

The goal of PTM-based CIL is to use a small set of trainable parameters to adaptively learn new classes while keeping the backbone frozen [49, 57]. Adapter-based methods [10, 34, 56, 58] achieve this goal by inserting lightweight learnable modules (adapters) into pre-trained models, offering a flexible and parameter-efficient approach with remarkable performance. Meanwhile, Low-Rank Adaptation (LoRA) [20] has emerged as a promising PEFT method by decomposing the weight updates into low-rank matrices, significantly reducing the number of trainable parameters. Recent works [26, 47, 50] have demonstrated the effectiveness of applying LoRA as learnable adapters in CIL. However, similar to other adapter-based methods [56, 58], this approach of learning new adapters for each new task leads to parameter redundancy [57] dur-

Code is available at: https://github.com/JiangpengHe/CL-LoRA ing inference and fails to leverage potential shared knowledge across tasks [48]. This motivates us to explore how to continuously update the learnable modules to enable cross-task knowledge transfer towards scalable CIL.

While the ideal PTM-based learnable modules for CIL should be capable of capturing both shared knowledge across tasks and task-specific characteristics [48], designing such a dual-purpose module presents significant challenges: (i) For shared knowledge preservation, continuously updating existing modules on new tasks leads to catastrophic forgetting of previously learned patterns and biases towards recent classes [56]. (ii) For task-specific feature learning, the key challenge lies in effectively capturing discriminative patterns unique to each task while preventing inter-task interference, particularly under the parameter efficiency constraints of low-rank decomposition [26, 47].

In this work, we propose Continual Low-Rank Adaptation (CL-LoRA), which introduces a novel dual-adapter architecture to combine both task-shared and task-specific LoRA modules as shown in Figure 1. Specifically, to preserve shared knowledge, we leverage random orthogonal matrices as down-projection in shared adapters and propose early exit knowledge distillation with gradient reassignment to maintain essential cross-task patterns while preventing catastrophic forgetting. Meanwhile, to capture task-specific characteristics, we introduce learnable block-wise weights with orthogonal constraints that enable discriminative feature learning while minimizing interference between tasks. This dual-module design allows CL-LoRA to simultaneously achieve efficient knowledge sharing and task-specific representation learning with minimal parameter overhead. Our key contributions are summarized as follows:

- We explored cross-task knowledge sharing in adapter-based CIL, providing new insights into how LoRA can be effectively utilized for CIL with PTMs.
- We propose a novel method that leverages gradient reassignment for effective cross-task knowledge preservation and introduces block weights with orthogonal constraints to capture discriminative task-specific features.
- Through comprehensive evaluations on various benchmarks, we demonstrate the effectiveness of our method and provide valuable insights into the trade-offs between parameter efficiency and performance in PTM-based CIL.

2. Related Work

2.1. Class-Incremental Learning

Class-Incremental Learning (CIL) aims to continuously learn new classes while preserving knowledge of previously learned classes. Conventional CIL methods typically start with randomly initialized models and train them from scratch, which can be broadly categorized into three groups: replay-based, regularization-based, and model expansion-based methods. Replay-based methods [1, 3–5, 18, 28, 29, 31, 32, 38, 45, 52] have shown remarkable performance by storing a small number of exemplars from previous tasks for rehearsal to maintain the performance on old classes. However, storing real data samples raises significant concerns in real-world applications, including privacy issues, storage limitations, and computational overhead of maintaining and processing the exemplar set [14, 60]. Regularization-based methods attempt to introduce penalties to constrain the update of important parameters [22, 25, 27], while model expansion methods allocate new components for incoming tasks [9, 30]. However, these conventional CIL methods face fundamental limitations in performance due to the challenges of training from scratch with limited data.

CIL with Pre-trained Models: With the recent success of pre-training in various vision tasks, leveraging pre-trained models (PTMs), typically the Vision Transformers (ViTs), for CIL has emerged as a promising paradigm that eliminates the need for training model from scratch and storing exemplars. Most existing PTM-based CIL work can be broadly categorized as *prompt-based* and *adapter-based* methods. Specifically, *prompt-based methods* [21, 39, 41, 48, 49] adapt PTMs through learnable tokens (prompts) prepended to input embeddings. L2P [49] first introduced a fixed prompt pool with instance-specific prompt selection through key query matching. DualPrompt [48] improved L2P by using both general and expert prompts to capture task-shared and task-specific knowledge, which aligns with the motivation in our work. CODA-Prompt [41] was introduced to improve prompt selection through an end-to-end attention mechanism. *Adapter-based methods* [10, 26, 34, 47, 56, 58] focus on efficient adaptation through lightweight learnable modules inserted at different layers of the ViT. These methods differ in their adaptation mechanism where some approaches [11, 56, 58] insert bottleneck modules among MLP or projection layers, while others [10, 26, 47, 50, 51] leverage Low-Rank Adaptation (LoRA) in Multi-Head Self-Attention (MHSA) layers. Despite their effectiveness, these methods either train new adapters for each new task or completely freeze learned adapters, leading to parameter inefficiency and limited knowledge transfer between tasks. In this work, we leverage LoRA as the adapter due to its parameter efficiency and effective adaptation through MHSA. But different from existing approaches, we propose a novel dual-adapter architecture that enables both shared knowledge accumulation and task-specific feature learning in a unified framework.

2.2. Parameter-Efficient Fine-Tuning

Parameter-Efficient Fine-Tuning (PEFT) [19] aims to adapt pre-trained models to downstream tasks by updating only a small subset of parameters while keeping the backbone frozen. Among various PEFT methods, Low-Rank Adaptation (LoRA) [20] has emerged as a promising approach that decomposes weight updates into low-rank matrices.

LoRA's unique design enables direct modification of weight matrices in MHSA layers through lightweight rank decomposition, making it particularly suitable for adapting large transformer models. Several variants have been proposed to enhance LoRA through dynamic rank adjustment [44, 54]. However, these methods are primarily designed for single-task fine-tuning and face significant limitations in CIL where continuously updating LoRA modules leads to catastrophic forgetting of previous tasks. These limitations motivate our work to design task-shared LoRA that can effectively preserve previously learned patterns while enabling efficient knowledge transfer across tasks.

3. Preliminaries

3.1. CIL with Pre-trained Models

The objective of CIL is to learn new classes continuously while maintaining knowledge of previously learned classes. Formally, we consider a sequence of tasks $\{\mathcal{T}_1, \mathcal{T}_2, ..., \mathcal{T}_T\}$, where each task $\mathcal{T}_t = \{(x_i^t, y_i^t)\}_{i=1}^{n_t}$ contains image data x and class-labels y from non-overlapping classes, *i.e.*, $\mathcal{C}_i \cap \mathcal{C}_j = \emptyset$ for any $i \neq j$. In this work, we target the rehearsal-free setup [14, 58, 60] where no samples from previous tasks can be stored for knowledge replay. In PTM-based CIL, the pre-trained vision transformer (ViT) [8] with feature extractor $f_\theta(\cdot)$ is widely adopted as the backbone [49]. To achieve parameter-efficient adaptation, adapter-based methods insert lightweight learnable modules f_{adapt} into transformer blocks while keeping pre-trained weights frozen. The extraction of adapter-augmented feature z can be generally expressed as:

$$z = f_\theta(x) + f_{adapt}(x) \tag{1}$$

Prototype-based classifier is widely adopted in rehearsal-free PTM-based CIL methods [56, 58]. During the training process, a local classifier $h_\phi^t \in \mathbb{R}^{d\times|\mathcal{C}_t|}$ is initialized for each new task t, and the model learns through the local cross-entropy loss $\mathcal{L}_{ce}$ on current task data:

$$\min_{f_{\text{adapt}}, h_\phi^t} \frac{1}{|\mathcal{T}_t|} \sum_{(x,y)\in\mathcal{T}_t} \mathcal{L}_{ce}(h_\phi^t(f_\theta(x) + f_{\text{adapt}}(x)), y) \tag{2}$$

where $|\mathcal{C}_t|$ denotes the number of classes in task t. After training, the class prototypes $\mathbf{P}^t = \{\mathbf{p}_i^t\}_{i\in\mathcal{C}_t}$, representing each class i in task t, are computed as the mean feature vectors of all training instances in that class using the corresponding task adapter f_{adapt}^t. During inference, given a test sample x, each task adapter $f_{\text{adapt}}^i \in \{f_{\text{adapt}}^1, \ldots, f_{\text{adapt}}^t\}$ extracts task-specific features that are matched with their corresponding task prototypes. The final prediction $\hat{y}$ is determined by the maximum cosine similarity across all tasks:

$$\hat{y} = \underset{y\in\bigcup_{i=1}^t \mathcal{C}_i}{\arg\max} \; \cos\left(\mathbf{p}_y^i, f_\theta(x) + f_{\text{adapt}}^i(x)\right) \tag{3}$$

where $\mathbf{p}_y^i$ is the prototype vector for class y in task i. In this work, we adopt the prototype classifier but instead leverage low-rank adaptation as learnable adapters, integrating both task-shared and task-specific adapters, which differs from existing methods that rely solely on task-specific adapters [26, 47, 56, 58].

3.2. Low-Rank Adaptation

Low-Rank Adaptation (LoRA) [20] is an efficient approach for adapting pre-trained models by decomposing weight updates into low-rank matrices. Specifically, for a pre-trained weight matrix $W \in \mathbb{R}^{d\times k}$, LoRA decomposes its update into a pair of rank decomposition matrices, including a down-projection matrix $\mathbf{B} \in \mathbb{R}^{r\times k}$ and an up-projection matrix $\mathbf{A} \in \mathbb{R}^{d\times r}$ with rank $r \ll min(d, k)$, achieving learning efficiency using only $r \times (d + k)$ trainable parameters. The modified feature extraction can be expressed as:

$$z = Wx + \Delta Wx, \quad \text{where} \quad \Delta W = \mathbf{AB} \tag{4}$$

In vision transformers (ViT) [8], these low-rank matrices are typically attached to Multi-Head Self-Attention (MHSA) layers, specifically the query (W_q), key (W_k), and value (W_v) projection matrices. To leverage LoRA as adapters in CIL, existing work [26, 47] introduces task-specific low-rank matrices to learn $\{\mathbf{A}_t, \mathbf{B}_t\}$ for each task t. Therefore, the adapted feature extraction for task t in Equation 1 can be implemented through LoRA as

$$z = f_\theta(x) + \mathbf{A}_t\mathbf{B}_t x \tag{5}$$

During the training of task t, only the task-specific matrices $\mathbf{A}_t$ and $\mathbf{B}_t$ receive gradient updates while freezing the pre-trained weights W and previous task matrices $\{\mathbf{A}_i, \mathbf{B}_i\}_{i=1}^{t-1}$. However, such task-specific LoRA designs lead to not only linear parameter growth with new tasks but also fail to leverage shared knowledge between tasks.

4. Method

In this work, we introduce Continual Low-Rank Adaptation (CL-LoRA), a novel dual-adapter architecture for class-incremental learning. As shown in Figure 2, CL-LoRA incorporates two complementary components including a task-shared LoRA adapter f_{shared} for preserving cross-task knowledge, and task-specific LoRA adapters f_{spec} for capturing unique characteristics of each task. During the learning process of task t, we insert f_{shared} into the first l transformer blocks and deploy f_{spec} in the remaining blocks. Following the LoRA framework [20], these LoRA adapters are integrated into Multi-Head Self-Attention (MHSA) layers, specifically modifying the query (W_q) and value (W_v) projection matrices and keeping the key (W_k) unchanged. To enhance knowledge preservation in f_{shared}, we introduce knowledge distillation with early exit mechanism and gradient reassignment based on the L_2 norm of weight vectors

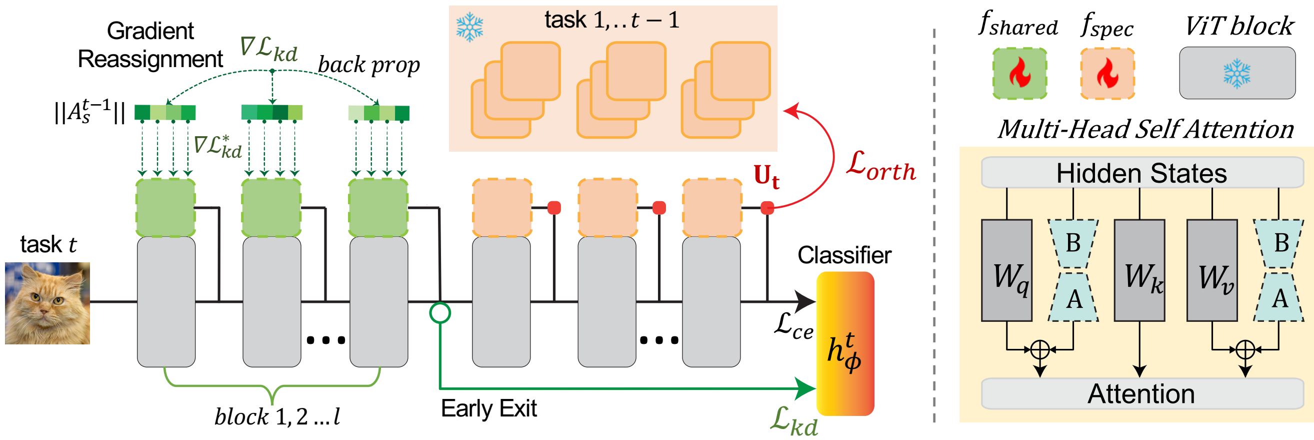

Figure 2. Overview of CL-LoRA for class-incremental learning. We insert shared adapters $(\mathbf{A}_s, \mathbf{B}_s)$ in the first l transformer blocks and task-specific adapters $(\mathbf{A}_t, \mathbf{B}_t)$ with learnable block weights $\mathbf{U}_t$ in the remaining blocks. To preserve cross-task knowledge, we apply knowledge distillation loss $\mathcal{L}_{\text{kd}}$ at the early exit point (l-th block) and reassign its gradient $\nabla_{\mathbf{A}_s^t}\mathcal{L}_{\text{kd}}$ based on L_2 weight norms of previous task's shared adapters $\|\mathbf{A}_s^{t-1}\|$ to obtain $\nabla_{\mathbf{A}_s^t}\mathcal{L}_{\text{kd}}^*$. Meanwhile, orthogonality constraints $\mathcal{L}_{\text{orth}}$ are imposed on block weights $\mathbf{U}_t$ to capture unique knowledge. Both shared and specific LoRA are inserted into MHSA layers on query (W_q) and value (W_v) projection matrices.

in up-projection matrices. Meanwhile, for f_{spec}, we employ learnable block-wise weights with orthogonality constraints to enable effective task-specific feature learning while preventing interference between tasks.

4.1. Dual-Adapter Architecture

In this section, we introduce our dual-adapter architecture that integrates task-shared and task-specific adapters into pre-trained vision transformer blocks.

Design of task-shared LoRA adapter: Our task-shared LoRA module leverages a fixed random orthogonal down-projection matrix $\mathbf{B}_s \in \mathbb{R}^{r\times k}$ and a cross-task shared trainable up-projection matrix $\mathbf{A}_s \in \mathbb{R}^{d\times r}$ initialized as zero. Therefore, our task-shared LoRA can be formulated as

$$f_{\text{shared}}(x) = \mathbf{A}_s\mathbf{B}_s x, \text{ where } \mathbf{B}_s\mathbf{B}_s^\top = \mathbf{I} \tag{6}$$

where $\mathbf{I} \in \mathbb{R}^{r\times r}$ is identity matrix. Our design is inspired by recent theoretical analysis [61], which highlights the asymmetric roles of LoRA matrices, showing that fine-tuning $\mathbf{A}$ is inherently more effective than fine-tuning $\mathbf{B}$. Moreover, the analysis suggests that a random, untrained $\mathbf{B}$ can perform nearly as well as a fully fine-tuned one. In this work, we initialize $\mathbf{B}_s$ by first generating a random matrix $\mathbf{M} \in \mathbb{R}^{r\times k}$ with elements sampled from standard normal distribution $\mathcal{N}(0, 1)$, then obtain the random orthogonal matrix through Singular Value Decomposition [23] of $\mathbf{M}$:

$$\mathbf{M} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^\top, \mathbf{B}_s = \mathbf{U}\mathbf{V}_r^\top \tag{7}$$

where $\mathbf{V}_r^\top$ denotes the transpose of the matrix formed by the first r rows of $\mathbf{V}$. $\mathbf{B}_s$ remains fixed during training and $\mathbf{A}_s$ is continuously updating over tasks. This design of shared adapter effectively preserves the structure of input data in the low-dimensional space and mitigate forgetting.

Dual-Adapter in Vision Transformer: Based on this design, we introduce our dual-adapter architecture for vision transformer (ViT) with N blocks. Specifically, we apply the shared LoRA adapters (Eq. 6) to the first l blocks ($l \le N$), and use original LoRA modules (Eq.5) as task-specific adapters for each task t to the remaining $N - l$ blocks. The output of the i-th block z^i can be calculated:

$$z^i = f_\theta^i(z^{i-1}) + \begin{cases} \mathbf{A}_s^i\mathbf{B}_s^i z^{i-1} & i \le l \\ \mathbf{A}_t^i\mathbf{B}_t^i z^{i-1} & l < i \le N \end{cases} \tag{8}$$

where z^{i-1} is the output from previous block with $z^0 = x$. The f_θ^i represents the i-th block of frozen pre-trained backbone, $\mathbf{A}_s^i$ is the shared trainable up-projection, $\mathbf{B}_s^i$ is the fixed orthogonal down-projection, and $\{\mathbf{A}_t^i, \mathbf{B}_t^i\}$ are task-specific adapters f_{spec} for task t block i. The dual-adapter design leverages the hierarchical structure of vision transformers [37], where early layers (blocks 1 to l) capture generalizable patterns for sharing across tasks, while deeper layers (blocks $l + 1$ to N) focus on task-specific details, aligning with CIL's goals of balancing shared knowledge retention with task-specific adaptation.

4.2. Learning Cross-Task Knowledge

While our dual-adapter architecture leverages random orthogonal down-projection matrix $\mathbf{B}_s$ to provide fixed projection to low rank space, the trainable up-projection matrix $\mathbf{A}_s$ still poses challenges for cross-task knowledge preservation. Specifically, as $\mathbf{A}_s$ is continuously updated during the learning of new tasks ($t > 1$), the output of task-shared adapters may drift significantly from previously learned feature mappings, leading to potential catastrophic forgetting. To address this challenge, we propose to use knowledge distillation with early exiting and gradient reassignment.

Knowledge Distillation with Early Exiting: Knowledge distillation has been widely adopted in CIL for learning cross-task knowledge [4, 25, 38] to mitigate forgetting. However, directly applying it with our dual-adapter architecture would harm the plasticity of our task-specific adapters to capture unique characteristics of each new task. Therefore, we introduce an early exit mechanism that strategically applies knowledge distillation at the transition point between task-shared and task-specific adapters (the l-th block). In detail, we extract the [CLS] token representation through shared adapters (1 to l-th blocks) for both current task z_t^l[CLS] and previous task z_{t-1}^l[CLS]. Then, the knowledge distillation is performed using the local classifier h_ϕ^t (Eq.2) and formulated as

$$\mathcal{L}_{\text{kd}} = \sum_{i \in \mathcal{C}_t} s_{t-1,i}^\tau \log(s_{t,i}^\tau) \tag{9}$$

where $s_t^\tau = Softmax(h_\phi^t(z_t^l[\text{CLS}])/\tau)$ represents the softened probability distribution over current task classes $\mathcal{C}_t$, and $\tau = 2$ is the temperature. However, knowledge distillation provides only implicit guidance for knowledge transfer [36]. To enable more explicit and targeted knowledge retention, we introduce gradient reassignment that directly leverages the structural information in shared adapters.

Gradient Reassignment: Since the shared up-projection matrix $\mathbf{A}_s$ is initialized as zero and works as an auxiliary adaptation branch to the original pre-trained projection matrix, its post-training element norms naturally reflect where the pre-trained weights need more adaptation. Specifically, each weight vector $\mathbf{a}_s \in \mathbb{R}^{1\times r}$ in $\mathbf{A}_s$ represents the up-projection mapping from the low-rank space ($\mathbb{R}^r$) back to the original feature dimension ($\mathbb{R}^d$), so the weight vector with larger norm indicates the feature dimensions where more significant modifications to the pre-trained weights are required, suggesting these dimensions are more crucial for knowledge adaptation across tasks. Thus, motivated by [13], we propose to redistribute gradients from $\mathcal{L}_{\text{kd}}$ based on the relative importance weights learned in the previous task. We adopt L_2 norm following existing CIL [52, 55] to measure the importance of weight vectors for knowledge accumulation. Let $\|\mathbf{a}_{s,j}^{t-1}\|_2$ denote the L_2 norm of the j-th weight vector in up-projection matrix $\mathbf{A}_s^{t-1}$ from last task $t-1$. The gradient for current task $\mathbf{A}_s^t \in \mathbb{R}^{d\times r}$ is modified as

$$\nabla_{\mathbf{A}_s^t}\mathcal{L}_{\text{kd}}^* = \nabla_{\mathbf{A}_s^t}\mathcal{L}_{\text{kd}} \odot \sigma(\{\|\mathbf{a}_{s,j}^{t-1}\|_2\}_{j=1}^d) \tag{10}$$

where $\odot$ represents element-wise multiplication and $\sigma(w) = d \times w / \sum_{i=1}^d w_i$ is the dimension-preserving normalization function. Our gradient reassignment ensures the essential dimensions for previous task knowledge receive stronger preservation gradients while allowing other dimensions more flexibility to adapt to new tasks. Combined with early exiting knowledge distillation, this creates a more targeted approach to knowledge retention in shared adapters.

4.3. Learning Task-Specific Knowledge

Though shared adapters capture generalizable patterns across tasks, learning unique characteristics for each new task is equally critical. The major challenge of capturing unique knowledge is task interference, where task-specific features can become entangled, making it difficult to discriminate between different tasks especially as LoRA only fine-tunes a very small portion of parameters compared to the large pre-trained backbone. The objective is to maintain discriminative features for each task while preventing them from collapsing into similar representations. Existing methods [26, 47] typically impose orthogonality constraints directly on all adapter parameters, which may over-restrict the model's adaptation capability and harm the performance.

Block-wise Weight Learning: To enable more fine-grained task-specific adaptation without interference, we introduce learnable block-wise scaling factors $\{\mu_t^i\}_{i=l+1}^N$ for task-specific adapters $\mathbf{A}_t$ of each task t. Rather than simply adding LoRA modules to the pre-trained backbone, these scaling factors modulate the contribution of each adaptation

$$z^i = f_\theta^i(z^{i-1}) + \mu_t^i \times \mathbf{A}_t\mathbf{B}_t z^{i-1}, \quad l < i \leq N \tag{11}$$

We also encourage the block-wise weights between different tasks to be distinct, which helps prevent them from updating similar blocks and thus reduces task interference. We include the regularization term

$$\mathcal{L}_{\text{orth}} = \sum_{i=1}^{t-1}\sum_{j,k} \|(\mathbf{U}_t^\top \mathbf{U}_i)_{j,k}\|_2 \tag{12}$$

where $\mathbf{U}_t = [\mu_t^{l+1}, ..., \mu_t^N]$ concatenates each block scaling factors $\mu_t > 0$ for task t. The proposed learnable block-wise weights allow each task to adaptively scale the importance of different transformer blocks. By minimizing the overlap between scaling factors of different tasks, rather than enforcing orthogonality directly on full adapter parameters, we efficiently reduce interference while maintaining plasticity [7] for learning new knowledge in CIL.

4.4. Integrated Objectives

The overall training objective can be expressed as

$$\mathcal{L} = \mathcal{L}_{\text{ce}} + \lambda_1\mathcal{L}_{\text{kd}} + \lambda_2\mathcal{L}_{\text{orth}} \tag{13}$$

including the local cross-entropy loss $\mathcal{L}_{\text{ce}}$ (Eq. 2), knowledge distillation loss $\mathcal{L}_{\text{kd}}$ for learning shared knowledge (Eq. 9), and orthogonality loss $\mathcal{L}_{\text{orth}}$ (Eq. 12) for encouraging task-specific characteristics. λ_1 and λ_2 are tunable hyperparameters. During inference, we adopt the prototype classifier as described in Section 3.1, but adapt it to our dual-adapter architecture. Specifically, given an input data, we first obtain its shared representation z^l through the first

Method	Params (%)	CIFAR-100 [24] T=20 A_T	CIFAR-100 [24] T=20 $\overline{A}$	ImageNet-R [16] T=40 A_T	ImageNet-R [16] T=40 $\overline{A}$	ImageNet-A [17] T=10 A_T	ImageNet-A [17] T=10 $\overline{A}$	VTAB [53] T=5 A_T	VTAB [53] T=5 $\overline{A}$
L2P [49]	0.2	79.51±0.67	85.50±1.23	60.62±1.12	65.82±0.71	37.62±1.89	39.81±1.36	76.41±2.26	78.96±1.62
DualPrompt [48]	0.5	80.44±1.38	86.96±1.98	61.73±0.93	67.41±0.30	47.45±0.96	56.43±2.33	80.94±2.87	82.51±3.49
CODA-Prompt [41]	4.6	81.36±0.88	88.17±0.61	63.93±0.82	70.39±0.49	51.61±0.63	60.70±0.94	89.49±0.42	92.27±0.61
LAE w/ LoRA [10]	0.8	79.67±1.06	85.17±1.53	57.04±1.13	67.55±1.22	54.28±0.94	63.25±2.21	76.00±8.21	82.24±2.45
APER [56]	1.4	83.26±0.52	89.09±0.56	67.13±0.63	74.05±0.30	56.60±1.81	65.53±1.16	84.99±0.06	88.27±0.16
RanPAC [34]	3.1	**87.62**±0.16	**91.63**±0.28	71.06±0.71	**78.53**±0.73	54.85±1.36	66.14±1.54	88.85±1.36	89.61±4.21
EASE [58]	1.4	**85.71**±0.76	90.96±0.83	**71.43**±0.18	78.04±0.67	**59.25**±0.88	**68.92**±2.06	**92.85**±0.88	**93.01**±0.33
O-LoRA [47]	0.4	81.26±0.68	89.63±0.61	63.19±0.26	72.52±0.29	47.53±0.84	55.02±0.74	86.98±0.89	87.22±1.21
InfLoRA [26]	0.3	80.97±0.74	88.84±0.90	64.51±1.25	73.22±1.12	47.04±0.90	56.91±1.27	87.16±1.17	88.83±0.94
CL-LoRA (Ours)	0.3	85.32±0.08	**91.02**±0.12	**74.51**±0.14	**81.58**±0.59	**60.54**±0.63	**70.15**±2.23	**94.29**±0.34	**94.57**±0.60

Table 1. The results of average ($\overline{A}$) and final (A_T) accuracy (%) comparison on CIFAR-100, ImageNet-R, ImageNet-A and VTAB benchmarks with total number of tasks T. We also report the trainable parameters (%) of each method relative to the pre-trained backbone. All results are averaged over 10 runs with mean ± standard deviation. **Best** and **Second Best** results are highlighted.

l blocks using the shared adapters. Then, for each task seen so far $i \in \{1, 2...t\}$, we compute the task-specific feature z_i^N[CLS] using its corresponding specific adapter $(\mathbf{A}_i, \mathbf{B}_i)$ in the remaining $N - l$ blocks following Eq. 11. This gives us t different features and we then compute cosine similarities between each task-specific feature and its corresponding task prototypes. The final prediction is determined as the class with the highest similarity score across all tasks:

$$\hat{y} = \underset{y \in \bigcup_{i=1}^{t} \mathcal{C}_i}{\arg\max} \ \cos(\mathbf{p}_y^i, z_i^N[\text{CLS}]) \tag{14}$$

where $\mathbf{p}_y^i$ is the prototype vector for class y in task i, computed using the same adapter combination during training.

5. Experiments

5.1. Experimental Setup

Datasets: We conduct comprehensive experiments on four representative CIL benchmarks including CIFAR-100 [24], ImageNet-R [16], ImageNet-A [17], and VTAB [53]. CIFAR-100 contains 100 natural object classes, ImageNet-R and ImageNet-A each contain 200 classes selected from ImageNet [40] with artistic renditions and adversarial filtering, respectively. VTAB consists of diverse visual classification tasks ranging from natural scenes to specialized medical images and we follow [56] to construct CIL with 50 selected classes. For all datasets, we create different task sequences by dividing the classes into T equal-sized tasks (*e.g.*, $T = 20$ tasks with 5 classes each for CIFAR-100).

Compared Methods: We compare with rehearsal-free PTM-based CIL methods from both prompt-based and adapter-based methods including L2P [49], DualPrompt [48], CODA-Prompt [41], EASE [58], APER [56], LAE [10], RanPAC [34], InfLoRA [26], and O-LoRA [47].

Evaluation Metrics: Following standard evaluation protocol [38], we adopt average accuracy $\overline{A} = \frac{1}{T}\sum_{t=1}^{T} A_t$, where A_t is the accuracy on all seen classes after learning task t, and final accuracy A_T, which measures the performance on all classes after learning the last task.

Implementation Details: We adopt ViT-B/16 [8] with $N = 12$ transformer blocks pretrained on ImageNet-21K [6] as our backbone architecture across all experiments. We use rank $r = 10$ in LoRA ($\mathbf{A} \in \mathbb{R}^{768\times10}$, $\mathbf{B} \in \mathbb{R}^{10\times768}$) and insert the task-shared LoRA to first half of $l = 6$ blocks and task-specific LoRA to remaining blocks. We use fixed hyper-parameters with $\lambda_1 = 5$ and $\lambda_2 = 0.0001$ for all experiments. The implementations of existing methods are based on the LAMDA-PILOT [43, 57, 59] and Mammoth [2, 3]. All results averaged over ten independent runs.

5.2. Experimental Results

As shown in Table 1, CL-LoRA demonstrates strong performance on various benchmarks with different tasks T while only using a small number of trainable parameters compared to existing work. Specifically, CL-LoRA achieves the best performance on ImageNet-R, ImageNet-A, and VTAB benchmarks. While RanPAC [34] achieves higher accuracy on CIFAR-100, it requires ten times more parameters (3.1% v.s. 0.3%). In addition, CL-LoRA shows significant improvements on challenging benchmarks such as ImageNet-R and ImageNet-A with distribution shifts through artistic renditions and natural adversarial examples, where it outperforms the second best accuracies by a large margin. Compared to prompt-based methods, CL-LoRA shows substantial advantages across all benchmarks with better parameter efficiency, particularly for longer sequences such as $T = 40$ on ImageNet-R. For adapter-based approaches, our method achieves similar promising results compared to methods inserting adapter in MLP Layer such as EASE [58] but requiring fewer parameters. In particular, among LoRA-based methods including O-LoRA [47] and InfLoRA [26], CL-LoRA demonstrates superior perfor-

KD	GR	BW	CIFAR-100 $T=10$	CIFAR-100 $T=20$	ImageNet-R $T=20$	ImageNet-R $T=40$
			88.20	88.13	82.24	79.61
✓			90.83	90.06	83.42	80.77
✓	✓		91.69	90.72	84.08	81.25
		✓	89.01	88.72	82.93	80.48
✓	✓	✓	**91.85**	**91.02**	**84.77**	**81.58**

Table 2. Ablation study with average accuracy $\overline{A}$ (%) on CIFAR-100 and ImageNet-R. Best results are marked in bold.

mance while maintaining similar parameter efficiency by leveraging both shared and task-specific knowledge effectively. These comprehensive results demonstrate that our dual-adapter architecture effectively balances parameter efficiency, knowledge preservation, and task-specific adaptation across diverse image classification scenarios.

5.3. Ablation Study

We conduct ablation studies to analyze our design choices in CL-LoRA. Specifically, we investigate (1) the effectiveness of different components (2) the position and the variants of task-shared adapters in transformer blocks.

Effect of Different Components. We first conduct ablation studies to validate the effectiveness of key components in CL-LoRA, including the early exit Knowledge Distillation (**KD**) with Gradient Reassignment (**GR**) as described in Section 4.2, and the use of Block-wise Weights (**BW**) as illustrated in Section 4.3. Table 2 shows that each component contributes to the overall performance improvement. Without KD, the shared adapter tends to bias towards recent tasks and gradually forget previously learned knowledge since it is continuously updated on new tasks without any knowledge retention mechanism. Adding KD with early exit brings significant improvements on both datasets by enforcing the shared adapter to preserve essential cross-task knowledge. The addition of GR further enhances the performance by explicitly identifying and preserving important information among the shared adapters, providing a more targeted approach to knowledge preservation. Using BW enables more flexible task-specific adaptation by learning optimal scaling factors for each transformer block while maintaining orthogonality between tasks to prevent interference. When combining all components, CL-LoRA achieves the best performance across all settings, particularly for longer tasks, demonstrating that our proposed components effectively complement each other in balancing knowledge preservation and task-specific adaptation.

Position and Variants of Task-Shared Adapters. In our main experiments, we set $l = 6$ to divide the $N = 12$ transformer blocks into two parts where the first 6 blocks with task-shared adapters and the remaining 6 blocks with task-specific adapters. To understand how this division affects performance, we conduct experiments by varying the position $l \in \{0, 2, 4, 6, 8, 10, 12\}$, where l indicates the

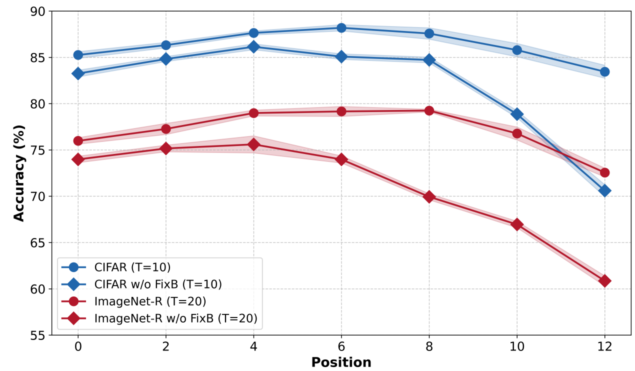

Figure 3. Final step accuracy A_T (%) on CIFAR-100 ($T = 10$) and ImageNet-R ($T = 20$) by varying the position $l \in \{0, 2, 4, 6, 8, 10, 12\}$ to split task-shared and specific adapters. Shaded regions indicate $\pm$ standard deviation around the mean.

position after which we switch from task-shared to task-specific adapters. For example, $l = 0$ means all adapters are task-specific (without knowledge distillation), while $l = 12$ means all adapters are task-shared (without learnable block weights). In addition, we also compare our design of using random orthogonal matrices illustrated in Section 4.1 with using regular trainable down-projection matrix $\mathbf{B}$ in task-shared adapters, denoted as (*w/o FixB*).

Figure 3 shows the final step accuracy A_T on CIFAR-100 ($T = 10$) and ImageNet-R ($T = 20$) across different values of l. First of all, our fixed random orthogonal down-projection consistently outperforms the trainable variant across all positions. Even with $l = 0$ (all task-specific adapters), the fixed untrained down-projection achieves better performance, aligning with recent theoretical findings [61]. This suggests that using fixed random orthogonal down-projection not only simplifies the adaptation process by reducing trainable parameters but also effectively prevents catastrophic forgetting in the low-rank space. We further observe that performance improves significantly when l increases, validating our core motivation that incorporating task-shared adapters helps leverage cross-task knowledge for better CIL performance. However, performance degrades noticeably when l becomes too large, particularly when all adapters are task-shared. While this degradation occurs because excessive sharing of adapters leads to few task-specific learning and potential bias towards the most recent tasks, our method with fixed down-projection demonstrates more stable performance compared to using trainable down-projection. These observations demonstrate that both leveraging cross-task knowledge through shared adapters and maintaining sufficient task-specific capacity are crucial components for effective CIL. While we adopt $l = 6$ across all experiments for fair comparison, the optimal position could be determined through validation on the first few tasks as in [48], allowing better adaptation to different application scenarios.

MHSA Layer	$r=1$		$r=5$		$r=10$		$r=20$		$r=64$	
	CIFAR-100	ImageNet-R	CIFAR-100	ImageNet-R	CIFAR-100	ImageNet-R	CIFAR-100	ImageNet-R	CIFAR-100	ImageNet-R
W_v	90.52	81.23	90.72	83.65	90.85	83.48	90.09	83.70	90.68	82.42
	0.14×10^5		*0.69×10^5*		*1.38×10^5*		*2.76×10^5*		*8.85×10^5*	
W_k, W_v	90.45	82.97	90.86	84.44	91.09	84.93	91.17	84.76	91.07	83.34
	0.28×10^5		*1.38×10^5*		*2.76×10^5*		*5.53×10^5*		*17.69×10^5*	
W_q, W_v	90.48	82.03	90.91	84.95	91.02	84.77	91.28	84.68	91.30	83.67
	0.28×10^5		*1.38×10^5*		*2.76×10^5*		*5.53×10^5*		*17.69×10^5*	
W_q, W_k, W_v	90.28	82.21	90.24	84.55	90.66	84.92	91.33	84.69	90.64	82.82
	0.41×10^5		*2.07×10^5*		*4.14×10^5*		*8.29×10^5*		*26.54×10^5*	

Table 3. Results of various LoRA configurations for CIFAR-100 (T=20) and ImageNet-R (T=20). For each configuration, we report average accuracy $\overline{A}$ (%) and the corresponding total number of *trainable parameters*.

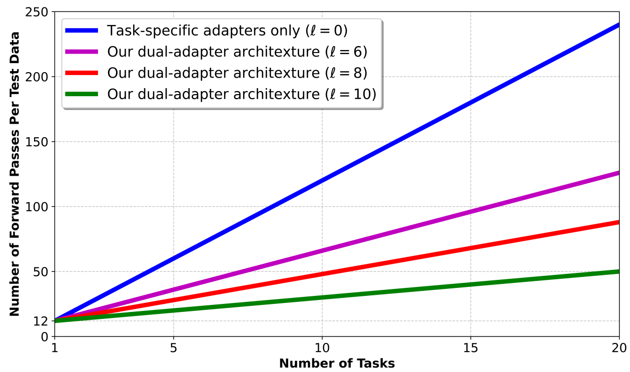

Figure 4. Inference scalability comparison with varied position l.

5.4. Discussions

In this section, we discuss the model's inference scalability and different LoRA configurations in terms of rank r and MHSA layer choices.

Inference Scalability. One of the strengths of using the task-shared adapter is to reduce the inference computation and enhance framework scalability. Specifically, for existing methods that rely solely on task-specific adapters [26, 58], each test sample must forward through all task-specific adapters across every transformer block to extract features. Our method alleviates this burden by leveraging task-shared adapters in the first l blocks. This design reduces the inference complexity from $O(NT)$ to $O(l+(N-l)T)$, where N represents the total number of transformer blocks attached with adapters and T denotes the number of tasks. Figure 4 compares the required number of adapter forward passes for each test data with $T = 20$ tasks and $N = 12$ blocks. Finally, the use of task-shared adapters also shows potentiality in reducing the reliance on clear task identity during training, which could further advance continual learning in online [15] and blurry task boundary settings [35].

LoRA Configurations. We also investigate the impact of different LoRA configurations, including the rank r and commonly adopted combinations of MHSA projection matrices (query W_q, key W_k, and value W_v) as suggested in previous studies [12, 20, 26, 42, 47]. Table 3 shows the results on CIFAR-100 and ImageNet-R with $T = 20$ tasks.

We first observe that even with an extremely low rank of $r = 1$, CL-LoRA still achieves promising results, particularly on CIFAR-100, demonstrating its reliability and parameter efficiency. In addition, increasing the rank r does not lead to consistent performance improvements on both datasets, suggesting that a small rank is sufficient for effective adaptation. This finding also indicates the effectiveness of CIL depends more on efficiently utilizing the adaptation capacity rather than simply increasing trainable parameters.

Regarding the choice of MHSA layers, we find that applying LoRA to more projection matrices does not necessarily yield better performance. For instance, adapting all three matrices (W_q, W_k, W_v) requires significantly more parameters yet achieves similar or slightly worse performance compared to adapting only two matrices (W_q, W_v) or (W_k, W_v). This suggests the importance of maintaining some pre-trained knowledge instead of adapting all components. Moreover, we observe that ImageNet-R is more sensitive to these configurations compared to CIFAR-100, showing larger performance variations, especially with very small ranks or with single matrix W_v. This sensitivity occurs due to its challenging distribution shifts, which require a more robust adaptation capacity. While our empirical analysis provides useful insights, developing algorithms to select optimal configurations based on task characteristics remains an important open challenge for future work.

6. Conclusion

In this work, we presented Continual Low-Rank Adpation (CL-LoRA), a novel dual-adapter architecture for rehearsal-free class-incremental learning that effectively combines task-shared and task-specific adapters. Through comprehensive experiments across multiple challenging benchmarks, we demonstrate CL-LoRA consistently achieves promising performance while using minimal trainable parameters. In addition, our systematic ablation studies provide valueable insights into both the importance of cross-task knowledge sharing and the effective utilization of low-rank adaptation in CIL. Finally, our work opens new future directions in exploring knowledge sharing to move beyond current task-specific adaptation paradigms.

References

[1] Eden Belouadah and Adrian Popescu. Il2m: Class incremental learning with dual memory. *Proceedings of the IEEE International Conference on Computer Vision*, pages 583–592, 2019. 2

[2] Matteo Boschini, Lorenzo Bonicelli, Pietro Buzzega, Angelo Porrello, and Simone Calderara. Class-incremental continual learning into the extended der-verse. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2022. 6

[3] Pietro Buzzega, Matteo Boschini, Angelo Porrello, Davide Abati, and Simone Calderara. Dark experience for general continual learning: a strong, simple baseline. *Advances in neural information processing systems*, 33:15920–15930, 2020. 2, 6

[4] Francisco M. Castro, Manuel J. Marin-Jimenez, Nicolas Guil, Cordelia Schmid, and Karteek Alahari. End-to-end incremental learning. *Proceedings of the European Conference on Computer Vision*, 2018. 1, 5

[5] Arslan Chaudhry, Marc'Aurelio Ranzato, Marcus Rohrbach, and Mohamed Elhoseiny. Efficient lifelong learning with a-gem. *arXiv preprint arXiv:1812.00420*, 2018. 2

[6] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255, 2009. 1, 6

[7] Shibhansh Dohare, J Fernando Hernandez-Garcia, Qingfeng Lan, Parash Rahman, A Rupam Mahmood, and Richard S Sutton. Loss of plasticity in deep continual learning. *Nature*, 632(8026):768–774, 2024. 5

[8] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. *International Conference on Learning Representations*, 2021. 3, 6

[9] Arthur Douillard, Matthieu Cord, Charles Ollion, Thomas Robert, and Eduardo Valle. Podnet: Pooled outputs distillation for small-tasks incremental learning. *Proceedings of the European Conference on Computer Vision*, pages 86–102, 2020. 2

[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. *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 11483–11493, 2023. 1, 2, 6

[11] Xinyuan Gao, Songlin Dong, Yuhang He, Qiang Wang, and Yihong Gong. Beyond prompt learning: Continual adapter for efficient rehearsal-free continual learning. *Proceedings of European Conference on Computer Vision*, 2024. 2

[12] Haiyang Guo, Fei Zhu, Wenzhuo Liu, Xu-Yao Zhang, and Cheng-Lin Liu. Pilora: Prototype guided incremental lora for federated class-incremental learning. *Proceedings of the European Conference on Computer Vision*, 2024. 8

[13] Jiangpeng He. Gradient reweighting: Towards imbalanced class-incremental learning. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16668–16677, 2024. 5

[14] Jiangpeng He and Fengqing Zhu. Exemplar-free online continual learning. *arXiv preprint arXiv:2202.05491*, 2022. 2, 3

[15] Jiangpeng He, Runyu Mao, Zeman Shao, and Fengqing Zhu. Incremental learning in online scenario. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 13926–13935, 2020. 8

[16] 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. *Proceedings of the IEEE/CVF international conference on computer vision*, pages 8340–8349, 2021. 6

[17] Dan Hendrycks, Kevin Zhao, Steven Basart, Jacob Steinhardt, and Dawn Song. Natural adversarial examples. *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 15262–15271, 2021. 6

[18] Saihui Hou, Xinyu Pan, Chen Change Loy, Zilei Wang, and Dahua Lin. Learning a unified classifier incrementally via rebalancing. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 831–839, 2019. 1, 2

[19] Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin De Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. Parameter-efficient transfer learning for nlp. *International conference on machine learning*, pages 2790–2799, 2019. 1, 2

[20] 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. 1, 2, 3, 8

[21] Dahuin Jung, Dongyoon Han, Jihwan Bang, and Hwanjun Song. Generating instance-level prompts for rehearsal-free continual learning. *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 11847–11857, 2023. 2

[22] James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, et al. Overcoming catastrophic forgetting in neural networks. *The National Academy of Sciences*, 114(13): 3521–3526, 2017. 2

[23] Virginia Klema and Alan Laub. The singular value decomposition: Its computation and some applications. *IEEE Transactions on automatic control*, 25(2):164–176, 1980. 4

[24] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. *Technical Report*, 2009. 6

[25] Zhizhong Li and Derek Hoiem. Learning without forgetting. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(12):2935–2947, 2017. 1, 2, 5

[26] Yan-Shuo Liang and Wu-Jun Li. Inflora: Interference-free low-rank adaptation for continual learning. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 23638–23647, 2024. 1, 2, 3, 5, 6, 8

[27] Xialei Liu, Marc Masana, Luis Herranz, Joost Van de Weijer, Antonio M Lopez, and Andrew D Bagdanov. Rotate your networks: Better weight consolidation and less catastrophic forgetting. *2018 24th International Conference on Pattern Recognition (ICPR)*, pages 2262–2268, 2018. 2

[28] Xialei Liu, Chenshen Wu, Mikel Menta, Luis Herranz, Bogdan Raducanu, Andrew D Bagdanov, Shangling Jui, and Joost van de Weijer. Generative feature replay for class-incremental learning. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 226–227, 2020. 2

[29] Yaoyao Liu, Yuting Su, An-An Liu, Bernt Schiele, and Qianru Sun. Mnemonics training: Multi-class incremental learning without forgetting. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 12245–12254, 2020. 2

[30] Yaoyao Liu, Bernt Schiele, and Qianru Sun. Adaptive aggregation networks for class-incremental learning. *Proceedings of the IEEE/CVF conference on Computer Vision and Pattern Recognition*, pages 2544–2553, 2021. 2

[31] David Lopez-Paz and Marc'Aurelio Ranzato. Gradient episodic memory for continual learning. *Advances in neural information processing systems*, pages 6467–6476, 2017. 1, 2

[32] Zilin Luo, Yaoyao Liu, Bernt Schiele, and Qianru Sun. Class-incremental exemplar compression for class-incremental learning. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11371–11380, 2023. 2

[33] Michael McCloskey and Neal J Cohen. Catastrophic interference in connectionist networks: The sequential learning problem. pages 109–165. Elsevier, 1989. 1

[34] Mark D McDonnell, Dong Gong, Amin Parvaneh, Ehsan Abbasnejad, and Anton van den Hengel. Ranpac: Random projections and pre-trained models for continual learning. *Advances in Neural Information Processing Systems*, 36, 2024. 1, 2, 6

[35] Jun-Yeong Moon, Keon-Hee Park, Jung Uk Kim, and Gyeong-Moon Park. Online class incremental learning on stochastic blurry task boundary via mask and visual prompt tuning. *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 11731–11741, 2023. 8

[36] Hyungkeun Park and Jong-seok Lee. Adaptive explicit knowledge transfer for knowledge distillation. *arXiv preprint arXiv:2409.01679*, 2024. 5

[37] Namuk Park and Songkuk Kim. How do vision transformers work? *International Conference on Learning Representations*, 2022. 4

[38] Sylvestre-Alvise Rebuffi, Alexander Kolesnikov, Georg Sperl, and Christoph H. Lampert. iCaRL: Incremental classifier and representation learning. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017. 1, 2, 5, 6

[39] Anurag Roy, Riddhiman Moulick, Vinay K Verma, Saptarshi Ghosh, and Abir Das. Convolutional prompting meets language models for continual learning. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 23616–23626, 2024. 2

[40] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision*, 115(3): 211–252, 2015. 1, 6

[41] 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. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11909–11919, 2023. 1, 2, 6

[42] James Seale Smith, Yen-Chang Hsu, Lingyu Zhang, Ting Hua, Zsolt Kira, Yilin Shen, and Hongxia Jin. Continual diffusion: Continual customization of text-to-image diffusion with c-loRA. *Transactions on Machine Learning Research*, 2024. 8

[43] Hai-Long Sun, Da-Wei Zhou, Han-Jia Ye, and De-Chuan Zhan. Pilot: A pre-trained model-based continual learning toolbox. *arXiv preprint arXiv:2309.07117*, 2023. 6

[44] Mojtaba Valipour, Mehdi Rezagholizadeh, Ivan Kobyzev, and Ali Ghodsi. Dylora: Parameter efficient tuning of pre-trained models using dynamic search-free low-rank adaptation. *arXiv preprint arXiv:2210.07558*, 2022. 3

[45] Fu-Yun Wang, Da-Wei Zhou, Han-Jia Ye, and De-Chuan Zhan. Foster: Feature boosting and compression for class-incremental learning. *Proceedings of the European Conference on Computer Vision*, pages 398–414, 2022. 2

[46] Liyuan Wang, Xingxing Zhang, Hang Su, and Jun Zhu. A comprehensive survey of continual learning: theory, method and application. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2024. 1

[47] Xiao Wang, Tianze Chen, Qiming Ge, Han Xia, Rong Bao, Rui Zheng, Qi Zhang, Tao Gui, and Xuanjing Huang. Orthogonal subspace learning for language model continual learning. 2023. 1, 2, 3, 5, 6, 8

[48] 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. *European Conference on Computer Vision*, pages 631–648, 2022. 1, 2, 6, 7

[49] 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. *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 139–149, 2022. 1, 2, 3, 6

[50] Xiwen Wei, Guihong Li, and Radu Marculescu. Online-lora: Task-free online continual learning via low rank adaptation. *NeurIPS 2024 Workshop on Scalable Continual Learning for Lifelong Foundation Models*, 2024. 1, 2

[51] Martin Wistuba, Prabhu Teja Sivaprasad, Lukas Balles, and Giovanni Zappella. Continual learning with low rank adaptation. *arXiv preprint arXiv:2311.17601*, 2023. 2

[52] Yue Wu, Yinpeng Chen, Lijuan Wang, Yuancheng Ye, Zicheng Liu, Yandong Guo, and Yun Fu. Large scale in-

cremental learning. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019. 1, 2, 5

[53] Xiaohua Zhai, Joan Puigcerver, Alexander Kolesnikov, Pierre Ruyssen, Carlos Riquelme, Mario Lucic, Josip Djolonga, Andre Susano Pinto, Maxim Neumann, Alexey Dosovitskiy, et al. A large-scale study of representation learning with the visual task adaptation benchmark. *arXiv preprint arXiv:1910.04867*, 2019. 6

[54] Qingru Zhang, Minshuo Chen, Alexander Bukharin, Nikos Karampatziakis, Pengcheng He, Yu Cheng, Weizhu Chen, and Tuo Zhao. Adalora: Adaptive budget allocation for parameter-efficient fine-tuning. *arXiv preprint arXiv:2303.10512*, 2023. 3

[55] Bowen Zhao, Xi Xiao, Guojun Gan, Bin Zhang, and Shu-Tao Xia. Maintaining discrimination and fairness in class incremental learning. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 13208–13217, 2020. 5

[56] 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. *International Journal of Computer Vision*, pages 1–21, 2024. 1, 2, 3, 6

[57] Da-Wei Zhou, Hai-Long Sun, Jingyi Ning, Han-Jia Ye, and De-Chuan Zhan. Continual learning with pre-trained models: A survey. *arXiv preprint arXiv:2401.16386*, 2024. 1, 6

[58] Da-Wei Zhou, Hai-Long Sun, Han-Jia Ye, and De-Chuan Zhan. Expandable subspace ensemble for pre-trained model-based class-incremental learning. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 23554–23564, 2024. 1, 2, 3, 6, 8

[59] Da-Wei Zhou, Qi-Wei Wang, Zhi-Hong Qi, Han-Jia Ye, De-Chuan Zhan, and Ziwei Liu. Class-incremental learning: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 46(12):9851–9873, 2024. 6

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

[61] Jiacheng Zhu, Kristjan Greenewald, Kimia Nadjahi, Haitz Sáez de Ocáriz Borde, Rickard Brüel Gabrielsson, Leshem Choshen, Marzyeh Ghassemi, Mikhail Yurochkin, and Justin Solomon. Asymmetry in low-rank adapters of foundation models. *arXiv preprint arXiv:2402.16842*, 2024. 4, 7