Mining the Potential of Rehearsal Mechanism for VLM-based Class Incremental Learning

Sen Tao[1,2,3,4], Jiawei Liu*[2], Yongchao Xu[2], Guangxi Wan[1,3], and Peng Zeng*[1,3]

[1]State Key Laboratory of Robotics, Shenyang Institute of Automation, CAS, Shenyang, China
[2]University of Science and Technology of China, Hefei, China
[3]Key Laboratory of Networked Control Systems, Shenyang Institute of Automation, CAS, Shenyang China
[4]University of Chinese Academy of Sciences, Beijing, China

taosen23@mails.ucas.ac.cn, jwliu6@ustc.edu.cn, yongchaoxu@mail.ustc.edu.cn, {zp, wanguangxi}@sia.cn

Abstract

Recent advancements in Class Incremental Learning (CIL) have increasingly harnessed the capabilities of Vision-Language Models (VLMs), such as CLIP, owing to their strong generalization performance. However, VLM-based CIL approaches are particularly vulnerable to *catastrophic forgetting* due to the parameter adaptation that CLIP undergoes with each new task. The rehearsal mechanism offers a straightforward solution to mitigate this issue by preserving limited exemplars from old tasks. Despite its simplicity, this strategy often leads to bias in the adapted CLIP model, as a result of the sample size imbalance between old and new classes. Existing CIL methods typically resort to Long-Tailed (LT) learning techniques, originally designed to handle static class imbalance, while overlooking the dynamic characteristic of class distribution shifts inherent to CIL settings. To address this challenge, we propose a novel Debiased Memory-Calibrated Gaussian Discriminant Analysis (DMGDA) method for VLM-based CIL. DMGDA introduces an inverse frequency adjustment to calibrate the prior probabilities of old classes and new classes, thereby yielding a non-parametric, debiased classifier that computes classification probability through Bayes' theorem. This formulation effectively addresses the dynamic class imbalance induced by the rehearsal mechanism, with classifier parameters estimated using the class means and the shared covariance matrix of the current task. Additionally, an Adaptive Memory Iteration Strategy within the rehearsal mechanism is designed to enhance the retention of knowledge from old tasks by adaptively filtering exemplars maintained in memory that align with the test distribution of previously encountered tasks. Extensive experiments conducted on standard CIL and LT-CIL benchmarks demonstrate the effectiveness and generalizability of the proposed approach.

Keywords: ***Class Incremental Learning, Vision-Language Models, Gaussian Discriminant Analysis***

1. Introduction

In an increasingly dynamic world, information is generated in the form of continuous data streams, necessitating the development of intelligent systems capable of adapting to novel environments while preserving previously acquired knowledge. This capability, known as continual learning, refers to a model's ability to incrementally learn new tasks without forgetting old ones. The main challenge in continual learning is *catastrophic forgetting* [16], a phenomenon in which a model's ability to recall previously acquired knowledge deteriorates as it learns new information. The issue of *catastrophic forgetting* has been extensively studied in the context of Class Incremental Learning (CIL) [41, 17, 28] , a representative paradigm of continual learning. The objective of CIL is to incrementally acquire new knowledge in a task-agnostic manner, *i.e.*, to maintain high classification performance during inference even when the task identity of test samples is not provided [43].

CIL methods can be roughly categorized into rehearsal-based approaches [39, 18, 51, 5] and no-exemplar methods [1, 40, 37, 20]. Rehearsal-based approaches preserve a limited number of exemplars from old tasks, which are reused during training to reinforce the model's retention of previous knowledge, thereby enhancing its stability. In con-

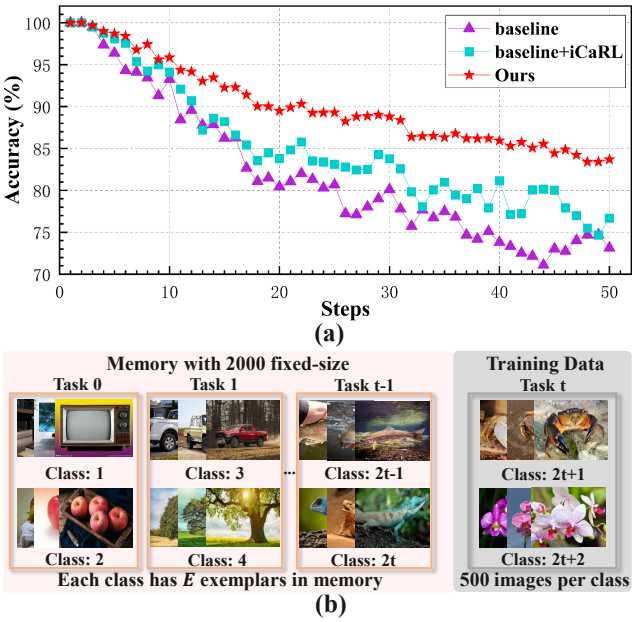

Figure 1: (a) We report the performance evolution on CIFAR100 with 50 steps, where the introduction of iCaRL exhibits an improvement of **1.51**% on average accuracy (Avg) across all steps. (b) An illustration of dynamic class imbalance on CIFAR100 with 50 steps, where $E = \lfloor 2000/2t \rfloor$ denotes the sample size of old classes in memory and variable t indicates task identity.

trast, no-exemplar methods mitigate *catastrophic forgetting* through alternative techniques such as knowledge distillation [1], dynamic expansion architecture [40] and regularization [20]. However, rehearsal-based methods generally achieve superior performance, as no-exemplar methods lack direct access to prior data, making it more difficult to retain previous knowledge effectively. Recently, Vision-Language Models (VLM) have demonstrated powerful transfer ability by jointly leveraging information from textual and visual modalities. A notable example is Contrastive Language-Vision Pre-training (CLIP) [38], a representative vision-language model trained on 400 million image-text pairs, which integrates image and text encoders and is renowned for its strong generalization ability across a wide range of downstream tasks. CIL methods built within the framework of VLM have achieved state-of-the-art performance, *e.g.*, MoE-Adapters [49] which augment a frozen CLIP with lightweight adapter modules. Despite their promise, VLM-based CIL approaches remain vulnerable to *catastrophic forgetting*, primarily due to the continual parameter adaptation that CLIP undergoes during the learning of new tasks. Within this context, the rehearsal mechanism continues to be a compelling strategy to alleviate *catastrophic forgetting* by selectively retaining limited exemplars from old tasks.

Reviewing the classical rehearsal-based method, iCaRL [39] introduced a herding algorithm for selecting exemplars whose features are closest to the class-wise moving barycenter, subsequently incorporating them into the training data via a rehearsal mechanism, which has since been widely adopted in various approaches [13, 18, 51]. In this work, we apply the iCaRL algorithm to the baseline model (MoE-Adapters [49]) to leverage the abundant knowledge preserved within the limited exemplars from old tasks stored in memory. However, as shown in Fig. 1(a), we observe that the incorporation of the rehearsal mechanism into the VLM-based CIL framework yields only limited performance gains. Further analysis reveals a critical limitation: the rehearsal mechanism fails to account for the dynamic class imbalance between the new task and old tasks. As shown in Fig. 1(b), we take CIFAR100 with 50 steps as an example to illustrate the concept of dynamic class imbalance. Each step corresponds to an incremental task involving two new classes with 500 training samples, while exemplars of old classes are stored in memory with a fixed size 2000. In task t, the exemplar size per old class, denoted as E, is computed as $\lfloor 2000/2t \rfloor$. Therefore, the ratio between the number of samples in new and old classes is defined as $r = 500/E$, which increases as the task identity t grows. The class imbalance becomes most severe in task 49, where $r = 25$. In such cases, the adapted model tends to be increasingly biased toward the new classes. Given that CIL requires the model to classify test samples originating from both old and new tasks during inference, this bias significantly impairs the model's ability to retain previously acquired knowledge. To address this issue, existing CIL methods [51, 4, 24] have incorporated Long-Tailed learning techniques (*e.g.*, Focal Loss [32] and SMOTE [6]), which were originally devised to mitigate static class imbalance, yet overlook the dynamic nature of class distribution shifts intrinsic to CIL settings.

Based on the above analyses, we propose a novel Debiased Memory-Calibrated Gaussian Discriminant Analysis (DMGDA) approach for VLM-based CIL to address the challenge of dynamic class imbalance, thereby further enhancing model stability, as demonstrated in Section 3.4. Additionally, we incorporate an Adaptive Memory Iteration Strategy (AMI) within the rehearsal mechanism to facilitate the effective retention of knowledge from old tasks. Specifically, we first fine-tune the lightweight MoE-Adapters [49] within image encoder and text encoder to maintain the model's ability to acquire new knowledge, *i.e.*, plasticity. Subsequently, DMGDA utilizes inverse frequency adjustment to dynamically calibrate the prior probabilities of old classes and new classes. This calibration increases the classification probability of underrepresented old classes, thereby yielding a non-parametric, debiased classifier. The parameters of this debiased classifier are non-learnable and are derived from the class-wise means

and the shared class covariance, estimated jointly using the training data of the current task and the retained exemplars stored in memory. Both the class means and the shared covariance are computed based on image features extracted by the image encoder, where the covariance matrix is estimated via an empirical Bayes ridge-type estimator. During inference, the final prediction logits are obtained by fusing the outputs of the baseline model with those of the non-parametric debiased classifier, thereby achieving a balance between plasticity and stability. Moreover, AMI leverages the semantic similarity between reliable test samples (*i.e.*, with high-confidence pseudo-label) and exemplars of the same class from old tasks as the guidance to identify and retain pivotal exemplars that closely align with the test distribution of old tasks. These selected exemplars are subsequently combined with those from the current task to update the memory for the next task, thereby promoting sustained retention of previously acquired knowledge. Given that real-world data typically exhibit greater complexity and more pronounced class imbalance compared to standard CIL benchmarks—which often assume uniform class distributions—we additionally assess the proposed method on the Long-Tailed Class Incremental Learning (LT-CIL) benchmark [33], where the data distribution for new tasks conforms to a long-tailed structure, representing a specific type of class imbalance. Extensive experiments conducted on both standard CIL and LT-CIL benchmarks demonstrate the effectiveness and generalizability of our method.

Our contribution can be summarized as follows:

- We propose the Debiased Memory-Calibrated Gaussian Discriminant Analysis to effectively handle dynamic class imbalance by correcting the prior probabilities of old classes and new classes through inverse frequency adjustment, thereby obtaining a non-parametric debiased classifier that substantially enhance model stability.
- We design an Adaptive Memory Iteration Strategy integrated within the rehearsal mechanism. This strategy adaptively filters exemplars by exploiting their alignment with the test distribution of previously seen tasks, thereby promoting long-term retention of knowledge from previous tasks.
- Through empirical analysis, we observe that the performance gains of incorporating rehearsal mechanisms into VLM-based CIL methods are limited. We attribute this phenomenon to the often-overlooked issue of dynamic class imbalance, and provide an in-depth explanation based on this observation.
- We conduct comprehensive experiments across both standard Class-Incremental Learning and Long-Tailed Class-Incremental Learning benchmarks to rigorously evaluate the effectiveness and generalizability of the proposed method.

2. Related Work

2.1. Class Incremental Learning

Class Incremental Learning (CIL) strives to continually learn new classes over time while retaining knowledge acquired from previously learned tasks. Existing CIL approaches be broadly categorized into rehearsal-based methods [39, 18, 13] and no-exemplar methods [1, 40, 37, 20]. Rehearsal-based methods select exemplars to store in a fixed memory budget and incorporate them into subsequent training phases to mitigate the effect of *catastrophic forgetting*. For instance, Rebuffi *et al*. [39] proposed the Incremental Classifier and Representation Learning (iCaRL) method, which selects limited exemplars using herding algorithm and replays them during future tasks. Douillard *et al*. [13] introduced the Task-Attention Block (TAB), a structure capable of dynamic expansion that adapts to the task at hand. Gao *et al*. [18] proposed the General Knowledge Attention Block (GKAB) and the Specific Knowledge Attention Block (SKAB) to facilitate inter-task knowledge transfer through memory. Chaudhry *et al*. [5] presented Riemannian Walk (RWalk) to filter out exemplars that are either near the classification boundary or exhibit high entropy. Zhao *et al*. [51] proposed Weighted Align (WA), which corrects weight bias in the fully connected layer following each task's training phase. In contrast, the no-exemplar methods alleviate *catastrophic forgetting* via alternative techniques, such as knowledge distillation [1], dynamic expansion architecture [40] and regularization [20]. For example, Asadi *et al*. [1] proposed Prototype-Sample Relation Distillation (PRD), which employs supervised contrastive learning and simulates previous data distributions by constructing a similarity-based distillation term using new-task data. Roy *et al*. [40] introduced a convolution-based reweighting mechanism for keys, queries, and values within multi-head self-attention layers of transformer, improving generalization across similar tasks. Goswami *et al*. [20] proposed the optimal Bayesian classifier by modeling the covariance relationship of features and employing class prototypes. Given that no-exemplar methods generally underperform compared to rehearsal-based approaches due to the absence of direct access to prior data, our proposed method adopts iCaRL [39] as the rehearsal backbone to reduce *catastrophic forgetting*.

2.2. Vision-Language Models

Vision-Language Models (VLM) have emerged as a new paradigm in the development of foundational models. Among them, Contrastive Language-Image Pre-Training (CLIP) [38] stands out as a prominent pre-trained model

trained on a large-scale proprietary dataset, demonstrating strong generalization capabilities across a wide range of downstream tasks, such as Human Objective Interaction [35, 45, 54], Semantic Segmentation [50, 47, 55, 19], and Objective Detection [30, 44]. Recently, increasing attention has been directed toward adapting VLMs for Class-Incremental Learning (CIL). For example, Zheng *et al.* [52] introduced CLIP into the CIL scenario through knowledge distillation between a frozen CLIP and a fully fine-tuned counterpart, achieving promising results. Inspired by the Mix-of-Experts (MoE) paradigm, Yu *et al.* [49] proposed a dynamic expansion architecture named MoE-Adapters within the frozen CLIP. This method preserved historical knowledge by freezing expert modules that were frequently activated during previous tasks, while learning new expert modules to adapt to emerging tasks, thereby attaining strong performance on standard CIL benchmarks. In contrast to the aforementioned approaches, our DMGDA explores the potential of the rehearsal mechanism within VLM-based CIL context. It explicitly addresses the bias introduced into the adapted CLIP model by dynamic class imbalance, thereby achieving superior performance.

2.3. Long-Tailed Learning

Long-Tailed Learning aims to mitigate the adverse effects of class imbalance, and existing methods can be broadly categorized into data re-sampling approaches [6, 21, 3] and loss adjustment techniques [32, 8]. In terms of data re-sampling, the Synthetic Minority Over-sampling Technique [6] generates synthetic samples through interpolation between instances of minority classes, thereby equalizing the class distribution. For loss adjustment, Focal Loss [32] modifies the standard loss function by down-weighting the contribution of easily classified examples, encouraging the model to focus on harder, misclassified samples. Some existing CIL methods [51, 4, 48] have adopted such strategies to address class imbalance. However, these methods generally assume a static imbalance, overlooking the inherently dynamic nature of class imbalance in the CIL setting, where the imbalance evolves as new tasks are introduced over time. In contrast, our proposed method explicitly addresses this dynamic class imbalance in CIL by dynamically calibrating the prior probabilities of old classes and new classes, yielding a non-parametric debiased classifier that better preserves knowledge across tasks. To further evaluate performance under more realistic, long-tailed scenarios, common in real-world data but overlooked by standard CIL benchmarks, Liu *et al.* [33] introduced the Long-Tailed Class Incremental Learning (LT-CIL) benchmark. Our method achieves strong results on this benchmark, demonstrating superior robustness and generalizability.

3. Methodology

3.1. Preliminary

In Class Incremental Learning (CIL), the model is trained sequentially on a series of $T+1$ classification tasks, denoted as $\{\mathcal{T}^t\}_{t=0}^{T}$. $\mathcal{T}^t = \{\mathcal{D}^t, C^t\}$ represents the classification task with the data $\mathcal{D}^t$ and the corresponding class set C^t. We define the classes in C^t as new classes for task $\mathcal{T}^t$, while all previously encountered classes are referred to as old classes. Each sample in $\mathcal{D}^t$ is represented as a tuple $(\boldsymbol{I}^t, y^t)$, where $\boldsymbol{I}^t$ is the input image and y^t is its associated ground-truth label. The class set C^t comprises Q^t mutually exclusive classes associated with task $\mathcal{T}^t$, and by definition, the class sets from different tasks are disjoint. In this work, a fixed-size memory is employed to retain a limited number of exemplars from previously encountered tasks. Additionally, it is assumed that the model has access to the class labels of the stored exemplars. Evaluation is conducted cumulatively on the test sets of all tasks encountered so far, requiring the model to accurately classify samples from the union of all previously seen classes.

3.2. Overall Framework

As shown in Fig. 2, the overall framework consists of a baseline model, Debiased Memory-Calibrated Gaussian Discriminant Analysis (DMGDA) and Adaptive Memory Iteration Strateg (AMI). This debiased classifier derived from DMGDA enhances the prediction of test images, thereby assisting the AMI within the rehearsal mechanism to improve the retention of knowledge from previous tasks. Conversely, the old exemplars obtained by AMI enable DMGDA to derive a more powerful debiased classifier. The baseline model is built on a frozen CLIP augmented with learnable MoE-Adapters [49], which incorporate new knowledge through a single router and two experts, such as LoRA [25]. After training at each step, both the adapted visual encoder and adapted text encoder are preserved, ensuring the model retains sufficient plasticity for future learning. To seamlessly integrate DMGDA and AMI within a rehearsal-based paradigm, we introduce a structured memory design aimed at improving exemplar management. In task t, the memory is organized as a set of task-specific memory blocks, denoted by $M^{\text{old}} = \{M^0, M^1, \ldots, M^{t-1}\}$, where each block M^k stores exemplars from task k. Each memory block M^k is further divided into block slots, with each block slot $\boldsymbol{S}_i^k$ corresponding to the exemplars of a specific class $i \in C^k$. This structured organization enables class-wise organization and retrieval, facilitating more effective memory calibration and controlled rehearsal.

First, during training on the current task data and exemplars stored in the previous memory blocks M^{old}, we fine-tune the MoE-Adapter integrated into the baseline model's

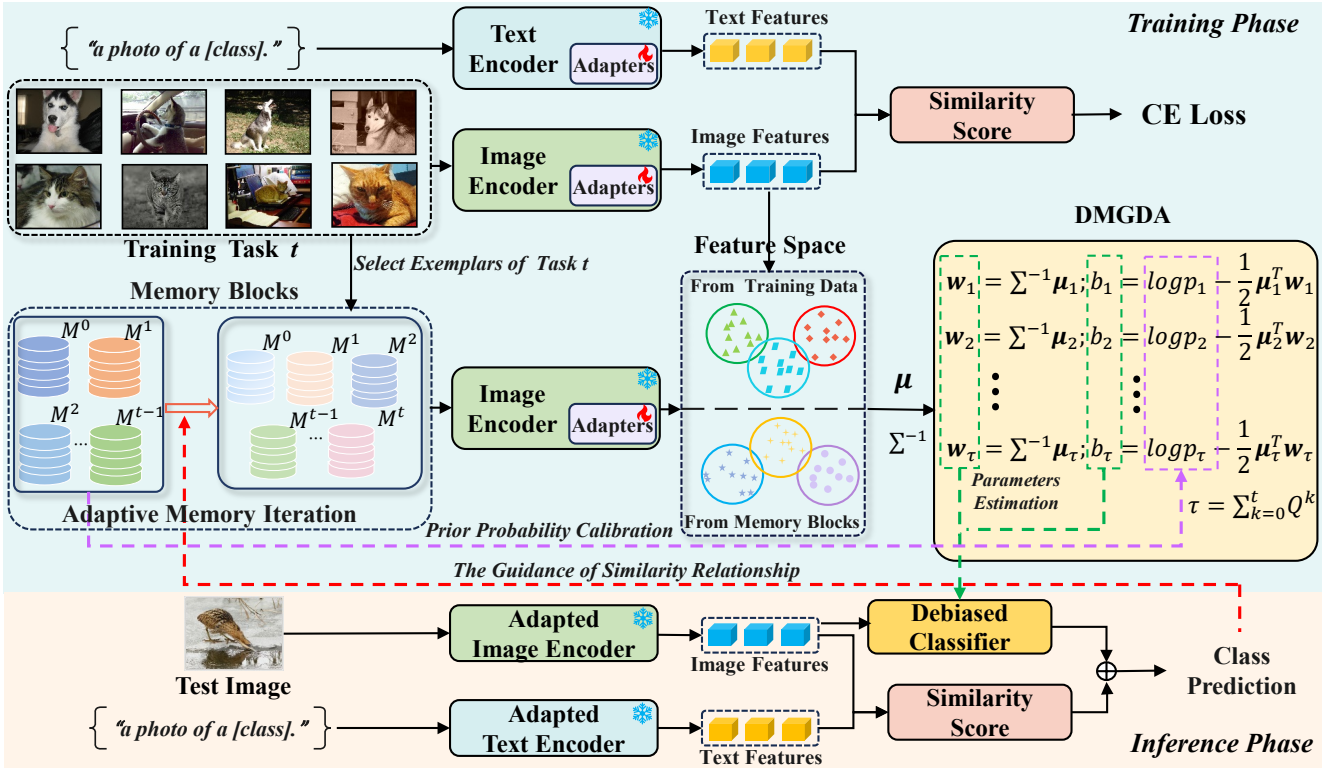

Figure 2: The overview of the proposed framework. It consists of a baseline model with two significant components: a Debiased Memory-Calibrated Gaussian Distribution Analysis (DMGDA) and an Adaptive Memory Iteration Strategy (AMI). DMGDA addresses dynamic class imbalance in class incremental learning by adjusting the prior probabilities of old and new classes through inverse frequency adjustment. During this process, DMGDA estimates the class means and the shared class covariance of the current task, enabling the construction of a non-parametric debiased classifier that further enhances model stability. AMI within the rehearsal mechanism enhances the retention of knowledge from previous tasks by adaptively filtering exemplars that best align with the test distribution of all previously encountered tasks. In this process, AMI reduces exemplars from previous tasks under the limited memory size and selects representative exemplars from the current task to be stored in a new memory block.

image and text encoders. This process, guided by the cross-entropy loss, aims to enhance the model's plasticity and its ability to extract discriminative image features. Specifically, the cross-entropy loss for a training sample $\mathbf{I}_j^t$ is defined as follows:

$$\ell_{CE}^j = -\sum_{m=1}^{K} \mathbb{I}\{y_j^t = m\} \log p(y = m \mid \mathbf{I}_j^t) \qquad (1)$$

where $\mathbb{I}\{\cdot\}$ denotes an indicator function equal to 1 when the condition holds, and $p(y = m \mid \mathbf{I}_j^t)$ represents the predicted probability that sample $\mathbf{I}_j^t$ belongs to class m. These extracted image features are subsequently utilized to estimate the class means and class shared covariance, forming a non-parametric, debiased classifier through DMGDA. This component enhances the model's stability against distributional drift, as proven in Section 3.4. During inference, the final logits $\boldsymbol{z}_f$ are obtained by fusing the logits $\boldsymbol{z}_c$ from CLIP with the debiased logits $\boldsymbol{z}_{de}$ from debiased classifier as follows:

$$\boldsymbol{z}_f = \boldsymbol{z}_c + \alpha \boldsymbol{z}_{de} = \boldsymbol{x}_{test}\boldsymbol{W}_c^T + \alpha(\boldsymbol{x}_{test}\boldsymbol{W}^T + \boldsymbol{b}) \qquad (2)$$

where $\boldsymbol{x}_{test}$ denotes the image feature of the test sample, $\boldsymbol{W}_c$ represents the weights of CLIP's classifier, which are generated by the text encoder using prompt templates "a photo of a [class]". $\boldsymbol{W}$ and $\boldsymbol{b}$ denote the weight and bias of debiased classifier respectively. α is a hyperparameter controls the contribution of the debiased logits, thereby balancing the model's plasticity and stability. Before transitioning to the next task, the rehearsal mechanism employs the herding algorithm to select a representative set of exemplars from the current task, which are stored in a newly allocated memory block M^t. To further enhance memory efficiency, AMI analyzes the similarity between the reliable test samples (*i.e.*, with high-confidence pseudo-label) and class-matched exemplars retained from previous tasks. This similarity-guided selection enables AMI to preserve the most informative exemplars in old memory blocks M^{old},

which are then combined with the new memory block M^t to construct the updated memory for the upcoming task.

3.3. Debiased Memory-Calibrated Gaussian Discriminant Analysis

To address model bias induced by dynamic class imbalance in continual learning, we adopt Gaussian Discriminant Analysis (GDA) [2] and extend its application to the Class-Incremental Learning (CIL) setting. A key insight is that increasing the prior probability of previously encountered classes effectively amplifies their influence in the decision boundary, thereby enhancing the model's capacity to preserve and discriminate old knowledge. GDA constructs its classifier by modeling the data distribution for each class and applying Bayes' theorem to derive classification probabilities. In the context of CIL, the classification probability can naturally be computed using the distributions of the current task's training data and the stored exemplars in memory, along with their prior probabilities.

$$p(y=i|\boldsymbol{x}) = \frac{p(\boldsymbol{x}|y=i)p(y=i)}{A+B} \tag{3}$$

Here, $i = 1, 2, \ldots, K^t$ denotes the set of class labels that the model has encountered from task 0 to task t, where $K^t = \sum_{k=0}^{t} Q^k$ represents the cumulative number of classes over all tasks. The terms $A = \sum_{j=1}^{K^{t-1}} p(\boldsymbol{x}|y = j)p(y = j)$ and $B = \sum_{j=K^{t-1}+1}^{K^t} p(\boldsymbol{x}|y = j)p(y = j)$ denote the total joint probabilities of old classes in memory and new classes in training data, respectively. $\boldsymbol{x} \in \mathbb{R}^D$ represents the D-dimensional feature generated by the image encoder. Following GDA [2], we assume that all features follow a multivariate Gaussian distribution with the shared covariance across classes, *i.e.*, $(\boldsymbol{x}|y = i) \sim \mathcal{N}(\boldsymbol{\mu}_i, \boldsymbol{\Sigma})$. Under this assumption, we derive the following theorem:

Theorem 1. The classifier for CIL can be derived based on Bayes' theorem and the assumption, where each feature contributes to the calculation of classification probabilities for each class.

$$p(y=i|\boldsymbol{x}) = \frac{\exp(G_i + \log p_i)}{\sum_{j=1}^{K^t} \exp(G_j + \log p_j)} \tag{4}$$

Here, $p_i = p(y = i)$ denotes the prior probability of class i, and $G_i = \boldsymbol{\mu}_i^T \boldsymbol{\Sigma}^{-1} \boldsymbol{x} - \frac{1}{2}\boldsymbol{\mu}_i^T \boldsymbol{\Sigma}^{-1} \boldsymbol{\mu}_i$ represents the log-likelihood in a simplified form. We define the logits for class i as $g_i(\boldsymbol{x}) = G_i + \log p_i$, which denotes the classifier's non-normalized prediction score. The classification probabilities thus correspond to applying the softmax function over the logits $g_i(\boldsymbol{x})$. Accordingly, the classifier's weight matrix $\boldsymbol{W} \in \mathbb{R}^{K^t \times D}$ and bias vector $\boldsymbol{b} \in \mathbb{R}^{K^t}$ are computed as follows:

$$\boldsymbol{w}_i = \boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}_i, \quad b_i = \log p_i - \frac{1}{2}\boldsymbol{\mu}_i^T \boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}_i \tag{5}$$

Next, we estimate the class means for all classes as well as the class shared covariance $\boldsymbol{\Sigma}$ to subsequently derive the corresponding weight and bias of the classifier, as specified in Eq. 5. Nevertheless, in higher dimensional spaces, the empirical covariance matrix tends to be a biased estimation of the true shared covariance. To mitigate this bias, we employ the empirical Bayes ridge-type estimator for a more robust estimation of $\boldsymbol{\Sigma}$.

In GDA [2], it is typically assumed that the prior probability of each class is uniform. This assumption, however, does not hold under the rehearsal mechanism used in CIL. As tasks are introduced sequentially, the fixed-size memory must accommodate exemplars from an increasing number of classes, leading to a gradual reduction in the number of exemplars per class for old tasks. Consequently, this disparity between the sample sizes of old and new classes introduces a dynamic class imbalance problem, which biases the adapted CLIP model toward newly introduced classes. To alleviate this issue, we propose a novel Debiased Memory-Calibrated Gaussian Discriminant Analysis (DMGDA) method that dynamically calibrates the class prior probabilities of both stored exemplars and current training data through inverse frequency adjustment mechanism. Specifically, DMGDA leverages the inverse frequencies of new and old class distributions to compute the calibrated prior probability as follows:

$$\begin{cases} p_i^m = \frac{1/f_m}{(1/f_m)K^{t-1} + (1/f_t)Q^t}, \text{for } i \in C^{0:t-1} \\ p_i^t = \frac{1/f_t}{(1/f_m)K^{t-1} + (1/f_t)Q^t}, \text{for } i \in C^t \end{cases} \tag{6}$$

Here, $f_m = \frac{\lfloor N/K^{t-1} \rfloor}{P} = \frac{N-\epsilon}{PK^{t-1}}$ and $f_t = \frac{h}{P}$ denote the empirical probabilities of old classes in memory and new classes in training data respectively. p_i^m and p_i^t represent the prior probabilities of the old classes and the new classes, respectively. N describes the fixed memory size, $\epsilon \ll K^{t-1}$ is a small correction term accounting for floor division, h indicates the sample size of new class training data, and $P = hQ^t + N$ denotes the total number of training instances for the current task combined with the exemplars stored in memory. $C^{0:t-1}$ denotes the class set comprises all classes encountered from the first to the $(t-1)^{th}$ task. By substituting calibrated prior probabilities into Eq. 3, we derive the following theorem.

Theorem 2. Inverse frequency adjustment defined in Eq. 6 increases the prior probability of old classes while keeping the log-likelihood terms unchanged. This calibration leads to an elevation in the corresponding logits for old classes.

According to Eq. 4, increasing the logits directly enhances the classification probabilities, thereby improving the prediction scores for old classes. Consequently, the resulting classifier is effectively debiased, as it compensates for the dynamic class imbalance induced by the rehearsal mechanism in CIL. Notably, this calibration process is non-

parametric, relying solely on frequency statistics and requiring no additional learnable parameters.

3.4. Proof of Theoretical Statement

Proof 1. The likelihood $p(\boldsymbol{x}|y=i)$ is represented using the multivariate Gaussian distribution with class mean $\boldsymbol{\mu}_i$ and class shared covariance $\boldsymbol{\Sigma}$. To simplify calculating the probability density function, we typically work with the logarithm of likelihood, as shown below:

$$\begin{aligned}\log p(\boldsymbol{x}|y=i) = &-\frac{1}{2}\boldsymbol{x}^T\boldsymbol{\Sigma}^{-1}\boldsymbol{x} + \boldsymbol{\mu}_i^T\boldsymbol{\Sigma}^{-1}\boldsymbol{x} - \frac{1}{2}\boldsymbol{\mu}_i^T\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}_i \\ &- \log\left(\sqrt{(2\pi)^D|\boldsymbol{\Sigma}|}\right)\end{aligned} \tag{7}$$

Ignoring constant terms that are independent of the class index i, the expression simplifies to:

$$\log p(\boldsymbol{x}|y=i) \approx \boldsymbol{\mu}_i^T\boldsymbol{\Sigma}^{-1}\boldsymbol{x} - \frac{1}{2}\boldsymbol{\mu}_i^T\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}_i = G_i \tag{8}$$

Next, by incorporating the prior probability $p(y=i)$ as $\log p_i$ and substituting both the simplified log-likelihood and the prior into the Bayes rule under the softmax normalization, we can obtain Eq. 4 with the following procedure.

$$\begin{aligned}p(y=i|\boldsymbol{x}) &= \frac{p(\boldsymbol{x}|y=i)p(y=i)}{\sum_{j=1}^{K^t} p(\boldsymbol{x}|y=j)p(y=j)} \\ &= \frac{\exp\left(G_i\right)p(y=i)}{\sum_{j=1}^{K^t}\exp\left(G_j\right)p(y=j)} \\ &= \frac{\exp\left(G_i+\log p_i\right)}{\sum_{j=1}^{K^t}\exp\left(G_j+\log p_j\right)}\end{aligned} \tag{9}$$

Proof 2. Consider the application of Bayes' theorem in the context of CIL, where the dynamic class imbalance is disregarded by assuming a uniform sample distribution across all classes. Under this simplifying assumption, the classification probabilities of old classes can be computed as below, derived from Eq. 3:

$$p_u(y=i|\boldsymbol{x}) = \frac{\lambda_i}{\sum_{j=1}^{K^{t-1}}\lambda_j + \sum_{j=K^{t-1}+1}^{K^t}\lambda_j} \tag{10}$$

where $\lambda_i = p(\boldsymbol{x}|y=i)$ denotes the likelihood. In contrast, under the proposed DMGDA, which incorporates the calibrated prior probabilities p_i^m for old classes and p_i^t for new classes to account for dynamic class imbalance, the classification probability of old classes can be expressed as:

$$\begin{aligned}p(y=i|\boldsymbol{x}) &= \frac{p_i^m\lambda_i}{p_i^m\sum_{j=1}^{K^{t-1}}\lambda_j + p_i^t\sum_{j=K^{t-1}+1}^{K^t}\lambda_j} \\ &= \frac{\lambda_i}{\sum_{j=1}^{K^{t-1}}\lambda_j + \frac{p_i^t}{p_i^m}\sum_{j=K^{t-1}+1}^{K^t}\lambda_j}\end{aligned} \tag{11}$$

To analyze the effect of prior calibration, we examine the ratio $\frac{p_i^t}{p_i^m}$, which quantifies the relative adjustment between prior probabilities of new and old classes. This ratio is derived from the inverse frequency calibration defined in Eq. 6, and depends on the frequency statistics f_t and f_m corresponding to the new and old classes, respectively. Empirically, we investigate several standard CIL benchmarks and observe that the configurations of ImageNet100 and TinyImageNet satisfy $f_t > f_m$. For CIFAR100 (10/20 steps), this condition also holds. However, CIFAR100 (50 steps) is relatively unique, as $f_t > f_m$ is only satisfied after task 2, as shown in the following equation:

- **CIFAR100 (50 steps):** $N = 2000,\ h = 500,\ Q^0 = 2,$ $Q^j = 2,\ \text{for } j = 1, 2, \ldots, t,\ t \geq 2 \Rightarrow f_t > f_m$

Across all of these settings, we consistently find that $f_t > f_m$, which implies $\frac{p_i^t}{p_i^m} < 1$ for most incremental steps. Substituting this into Eq. 11 reveals that the classification probability of old classes under DMGDA is systematically higher than that under the uncalibrated baseline:

$$p(y=i|\boldsymbol{x}) > p_u(y=i|\boldsymbol{x}), \text{for } i \in C^{0:t-1} \tag{12}$$

In summary, the proposed DMGDA yields a debiased classifier that effectively mitigates the bias of the adapted CLIP model toward new classes induced by dynamic class imbalance, achieved in a fully non-parametric manner. By systematically increasing the classification probabilities of old classes, this calibration strategy enhances the model's sensitivity to previously learned categories, thereby improving stability and retention in CIL scenario.

3.5. Adaptive Memory Iteration Strategy

Our rehearsal mechanism adopts the herding algorithm, as introduced in iCaRL [39], to select representative exemplars for newly introduced classes in the training data. Specifically, for each new class in task t, the herding algorithm first computes the class-wise mean feature vector and subsequently selects samples in an iterative manner based on their proximity to this class mean. The selected exemplars are stored in dedicated block slots corresponding to each class. Once exemplar selection is completed for all Q^t classes in task t, the resulting Q^t block slots merged into a unified memory block M^t for task t. However, the inclusion of new class exemplars inevitably increases the total number of stored exemplars beyond the fixed memory budget N. To accommodate this constraint, an exemplar reduction algorithm must be applied to previously stored samples from earlier tasks. Inspired by recent studies [14], we propose an Adaptive Memory Iteration Strategy (AMI) as the exemplar reduction algorithm that prunes exemplars during inference phase, guided by the data distribution of the test instances. AMI preserves knowledge from previous

Algorithm 1 Adaptive Memory Iteration Strategy

Input: $\{\boldsymbol{X}, \hat{y}^k\}, \hat{y}^k \in C^k$ // features of L test samples with the same high-confidence pseudo-label $\hat{y}^k$
Input: n_{t-1}, n_t // capacity of block slots
Require: Φ // current image feature function
Output: $M^{new} = \{M^0, M^1, \ldots, M^t\}$

/* *shrink old memory block* M^k */
for $k = 0, 1, \ldots, t$-1 **do**
  /* *reduce the capacity of block slot* $\boldsymbol{S}_i^k$ */
  for $\boldsymbol{S}_i^k \in M^k$ **do**
    if $\hat{y}^k = i$ **then**
      $\boldsymbol{J}_i^k : (n_{t-1}, D) \leftarrow \Phi(\boldsymbol{S}_i^k)$ // extract exemplar features from block slot
      $\boldsymbol{W}^s : (n_{t-1}, L) \leftarrow \boldsymbol{J}_i^k * \boldsymbol{X}^T$ // similarity matrix when pseudo-label matches slot number
      $\boldsymbol{S}_i^k \leftarrow \text{Select}(\boldsymbol{W}^s, n_t)$
    end if
  end for
end for
$M^a \leftarrow \{M^0, M^1, \ldots, M^{t-1}\}$
$M^{new} \leftarrow M^a + M^t$
Return: M^{new}

tasks by adaptively filtering exemplars based on their alignment with the cumulative test distribution of all encountered tasks, while ensuring minimal computational overhead.

Specifically, the process of AMI is outlined in the Algorithm 1. The algorithm takes as input the features $\boldsymbol{X} \in \mathbb{R}^{L \times D}$ of L test samples, each associated with the same high-confidence pseudo-label $\hat{y}^k$ (with confidence $\kappa > 0.8$), as well as the target capacity of block slots $n_t = \lfloor \frac{N}{K^t} \rfloor$ for tasks t. For each block slot $\boldsymbol{S}_i^k$ in the old memory block M^k whose index i matches $\hat{y}^k$, we first extract exemplar features $\boldsymbol{J}_i^k \in \mathbb{R}^{n_{t-1} \times D}$ using the current image feature extractor Φ. We then compute the cosine similarity matrix $\boldsymbol{W}^s$ between test features $\boldsymbol{X}$ and exemplar features $\boldsymbol{J}_i^k$. By summing each row of $\boldsymbol{W}^s$, we obtain a similarity score for each exemplar, reflecting its alignment with the current test distribution-higher scores indicate greater representativeness. To filter exemplars, we apply the function Select(), which retains the top-n_t exemplars in each block slot $\boldsymbol{S}_i^k$ based on similarity scores. Repeating this process across all relevant memory blocks yields the updated memory blocks M^a. We then construct the updated memory for the next task as $M^{\text{new}} = M^a \cup M^t$, where M^t is the newly formed memory block for task t. Through AMI, the memory content is adaptively reshaped from fitting the training distribution toward aligning with the test distribution, thereby mitigating distributional shift and enhancing the retention of knowledge from previous tasks.

4. Experiments

4.1. Experimental Setting

Datasets. For the standard CIL benchmark, we employ three commonly used dataset, *i.e.*, CIFAR100 [13], TinyImageNet [48] and ImageNet100 [10] to evaluate the performance of our DMGDA under various task configurations. CIFAR100 consists of 100 classes, which are partitioned into 10 steps (10 classes per step), 20 steps (5 classes per step), and 50 steps (2 classes per step) for sequential training. TinyImageNet, consists of 200 classes, where 100 classes are designated as base classes and the remaining 100 classes is divided into 5 steps, 10 steps, and 20 steps, following a similar incremental protocol. ImageNet100 is evaluated under two common configurations: ImageNet100-B0, where all 100 classes are evenly distributed across 10 incremental steps (10 classes per step), ImageNet100-B50, where the first 50 classes are introduced in the initial step and the remaining 50 are incrementally added over 10 subsequent steps. For the LT-CIL benchmark, we follow the protocol in [33] and conduct experiments on CIFAR100 to assess model performance under class-imbalanced conditions.

Metrics. We compare our method with existing approaches using top-1 classification accuracy as the evaluation metric. Specifically, We report the final accuracy after the last incremental step ("Last") and the average accuracy across all steps ("Avg"), which reflects overall performance throughout the entire continual learning process.

Implementation Details. All experiments are implemented in PyTorch and conducted on NVIDIA GeForce RTX 3090 GPUs. ViT-B/16 is adopted as the default backbone for the baseline model across all settings. Following MoE-Adapters [49], we utilize LoRA as the expert modules, and the router is implemented as a Multi-Layer Perceptron (MLP) that selects the top-2 experts based on gating scores. For a fixed memory budget N, we store 2,000 exemplars for both CIFAR100 and ImageNet100, and 4,000 exemplars for TinyImageNet, consistent with the protocol in [13]. During model adaptation, the baseline model generates predictions by computing cosine similarity between image features and text embeddings. The prompt template used across benchmarks is "a photo of a [class]". The baseline is optimized using cross-entropy loss with the Adam optimizer, employing a learning rate of 0.001, batch size of 64, and training for a single epoch across all datasets. To balance the trade-off between plasticity and stability, we set the hyperparameter α to 1.9 for CIFAR100 and ImageNet100, and to 3.5 for TinyImageNet. For the LT-CIL setting, we strictly follow the experimental configuration outlined in [33].

4.2. Overall Results

4.2.1 The Standard CIL Benchmark

We begin by comparing the proposed DMGDA with state-of-the-art methods on CIFAR100, and the results are summarized in Tab. 1. The findings demonstrate that our method consistently achieves superior performance across CIFAR100 with 10, 20, and 50 incremental steps. Under the "Avg" metric, our DMGDA surpasses the current state-of-the-art method, MoE-Adapters, by **2.94**%, **4.78**%, and **6.50**% at 10, 20, and 50 steps, respectively. Similarly, in terms of the "Last" metric, the proposed DMGDA exceeds MoE-Adapters by up to **4.67**%, **4.71**% and **8.45**%. It is particularly noteworthy that in the 50-step setting, where models typically suffer from significant forgetting due to the extended number of tasks, our method exhibits a pronounced performance advantage, highlighting the robustness and efficacy of the proposed DMGDA.

To further validate the superiorty of DMGDA, we extend performance comparison to TinyImageNet and ImageNet100, with detailed experimental results presented in Tab. 2 and Tab. 3. For TinyImageNet, it is evident that the proposed method attains state-of-the-art performance with 100 base classes across the 5, 10, and 20-step settings. Under the "Avg" metric, the proposed method surpasses MoE-Adapters by **1.06**% and **1.81**% at 10 and 20 steps, respectively. In terms of the "Last" metric, the proposed method outperforms MoE-Adapters by up to **0.93**%, **2.01**% and **2.94**% across these incremental steps. For ImageNet100, our method exceeds competing approaches by a minimum of **0.80**% on ImageNet100-B0 with respect to the "Avg" metric. Moreover, it outperforms alternatives by at least **0.98**% and **0.66**% on ImageNet100-B0 and ImageNet100-B50, respectively, when considering the "Last" metric. Collectively, these results substantiate that DMGDA consistently delivers superior performance relative to existing methods across diverse and challenging real-world datasets.

To corroborate that the proposed method enhances stability over the baseline model, we conduct experiments on the CIFAR100 dataset under the standard CIL benchmark, comparing it with MoE-Adapters and reporting the classification probabilities of old classes, as shown in Table 4. We can observe that the proposed method consistently surpasses MoE-Adapters across all evaluation settings. Specifically, the proposed DMGDA achieves leading performance over MoE-Adapters, with improvements of **0.62**%, **2.67**%, and **6.12**% in the "Avg (o)" metric, as well as **2.46**%, **6.67**%, and **11.11**% in the "Last (o)" metric at 10, 20, and 50 incremental steps, respectively. These experimental results indicate that the proposed method effectively enhances model stability, thereby supporting the theoretical analysis presented in Section 3.4.

Table 1: Performance comparison of different methods on CIFAR100 in CIL. We use **boldface** to indicate the best-performing method and underline to denote the second-best.

Methods	10 steps		20 steps		50 steps	
	Avg	Last	Avg	Last	Avg	Last
UCIR[24]	58.66	43.39	58.17	40.63	56.86	37.09
Bic[46]	68.80	53.54	66.48	47.02	62.09	41.04
PODNet[12]	58.03	41.05	53.97	35.02	51.19	32.99
DER [48]	74.64	64.35	73.98	62.55	72.05	59.76
DyTox+ [13]	74.10	62.34	71.62	57.43	68.90	51.09
HRFSN [15]	58.60	41.94	-	-	-	-
DNE [26]	74.86	70.04	-	-	-	-
CLIP Zero-shot	74.47	65.92	75.20	65.74	75.67	65.94
Fine-tune	65.46	53.23	59.69	43.13	39.23	18.89
LwF [31]	65.86	48.04	60.64	40.56	47.69	32.90
iCaRL [39]	79.35	70.97	73.32	64.55	71.28	59.07
LwF-VR [11]	78.81	70.75	74.54	63.54	71.02	59.45
PROOF [53]	84.88	76.29	85.12	76.13	-	-
ZSCL [52]	82.15	73.65	80.39	69.58	79.92	67.36
MoE-Adapters[49]	85.21	77.52	83.72	76.20	83.60	75.24
Ours	**88.15**	**82.19**	**88.50**	**80.91**	**90.10**	**83.69**

Table 2: Performance comparison of different methods on TinyImageNet with 200 classes in CIL. We designate 100 classes as base classes, and the remaining 100 classes are evenly distributed across the incremental steps.

Methods	5 steps		10 steps		20 steps	
	Avg.	Last	Avg.	Last	Avg.	Last
EWC [27]	19.01	6.00	15.82	3.79	12.35	4.73
EEIL[4]	47.17	35.12	45.03	34.64	40.41	29.72
UCIR [24]	50.30	39.42	48.58	37.29	42.84	30.85
MUC [34]	32.23	19.20	26.67	15.33	21.89	10.32
PASS [56]	49.54	41.64	47.19	39.27	42.01	32.93
DyTox [13]	55.58	47.23	52.26	42.79	46.18	36.21
HRFSN [15]	58.46	50.87	55.92	47.30	-	-
CLIP Zero-shot	69.62	65.30	69.55	65.59	69.49	65.30
Fine-tune	61.54	46.66	57.05	41.54	54.62	44.55
LwF [31]	60.97	48.77	57.60	44.00	54.79	42.26
iCaRL [39]	77.02	70.39	73.48	65.97	69.65	64.68
LwF-VR [11]	77.56	70.89	74.12	67.05	69.94	63.89
ZSCL [52]	80.27	73.57	78.61	71.62	77.18	68.30
MoE-Adapters [49]	**81.12**	76.81	80.23	76.35	79.96	75.77
Ours	80.69	**77.74**	**81.29**	**78.36**	**81.77**	**78.71**

4.2.2 The LT-CIL Benchmark

To demonstrate that the proposed method can effectively address dynamic class imbalance, we conduct experiments on a more challenging the Long-Tailed Class Incremental Learning (LT-CIL) benchmark, with the results summarized in Tab. 5. In the LT-CIL benchmark, the sample sizes across classes follow an exponential decay distribution, where the ratio between the least frequent and the most frequent class is controlled by the imbalance factor δ. A smaller δ indicates a more severe imbalance, whereas a larger δ corresponds to a more balanced distribution. When $\delta = 1$, the setting reduces to the standard CIL benchmark with uniform class distribution. Following the protocol established in [33], we set $\delta = 0.01$ in all experiments on CIFAR100. As illustrated in Tab. 5, the Ordered LT-CIL setting arranges classes in descending order of sample size for sequential task construction, while the Shuffled LT-CIL setting ran-

Table 3: Performance comparison of different methods on ImageNet100 under various CIL settings.

Methods	ImageNet100-B0		ImageNet100-B50	
	Avg	Last	Avg	Last
UCIR [24]	-	-	68.09	57.30
TPCIL [42]	-	-	74.81	66.91
PODNet [12]	-	-	74.33	-
DER [48]	76.12	66.06	77.13	72.06
DyTox [13]	73.96	62.20	-	-
DyTox+ [13]	77.15	67.70	-	-
MAFDRC [7]	79.66	70.41	77.95	71.26
CLIP Zero-shot	84.42	74.92	78.86	74.92
Fine-tune	83.10	70.72	80.31	72.48
LwF [31]	83.35	72.40	80.74	72.22
iCaRL [39]	83.40	70.96	79.76	73.96
LwF-VR [11]	82.53	69.68	80.82	70.18
PROOF [53]	84.71	72.48	-	-
ZSCL[52]	86.39	76.22	**84.28**	79.54
MoE-Adapters [49]	86.13	75.54	83.99	79.00
Ours	**87.19**	**77.20**	84.14	**80.20**

Table 4: Performance comparison for old classes on CIFAR100 in CIL. "Avg (o)" denotes the average accuracy of old classes across all incremental steps, while "Last (o)" refers to the accuracy of old classes at the final step.

Methods	10 steps		20 steps		50 steps	
	Avg (o)	Last (o)	Avg (o)	Last (o)	Avg (o)	Last (o)
MoE-Adapters [49]	86.21	77.94	86.47	74.98	85.64	71.67
Ours	86.83	80.40	89.14	81.65	91.76	82.78

Table 5: Performance comparison of different methods on CIFAR100 in LT-CIL, with average accuracy reported.

Methods	Ordered LT-CIL		Shuffled LT-CIL	
	5 steps	10 steps	5 steps	10 steps
EEIL [4]	38.46	37.50	31.91	32.44
+Two Stage [33]	38.97	37.58	34.19	33.70
LUCIR [24]	42.69	42.15	35.09	34.59
+Two Stage [33]	45.88	45.73	39.40	39.00
PODNet [12]	44.07	43.96	34.64	34.84
+Two Stage [33]	44.38	44.35	36.37	37.03
MAF [23]	35.92	33.70	31.63	30.18
MAFDRC [7]	53.13	49.01	41.41	41.84
MoE-Adapters [49]	77.90	78.78	78.92	76.47
Ours	**83.35**	**84.22**	**84.48**	**85.61**

Table 6: Ablation study of each component of the proposed method on CIFAR100 with 50 steps. The baseline model is incrementally enhanced by incorporating DMGDA and AMI. The rehearsal mechanism defaults to iCaRL, and AMI is employed for exemplar reduction when activated.

Baseline	Rehearsal	DMGDA	AMI	Avg	Last
✓				83.60	75.24
✓	✓			85.11	76.68
✓	✓		✓	86.58	79.96
✓	✓	✓		88.15	82.19
✓	✓	✓	✓	90.10	83.69

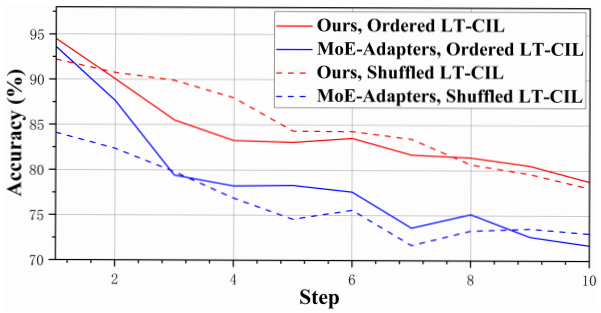

Figure 3: Classification accuracy at each step on CIFAR100 under 10-step setting in LT-CIL. In Ordered LT-CIL, tasks are constructed by arranging classes in descending order of sample size, while Shuffled LT-CIL forms tasks by randomly sampling classes regardless of their frequency. Both variants are designed to evaluate the model's robustness in learning under severe class imbalance in incremental classification scenarios.

Table 7: Ablation study comparing different methods for mitigating class imbalance on CIFAR100 in CIL.

Methods	50 steps	
	Avg	Last
CLIP Zero-shot	75.67	65.94
Baseline (Rehearsal)	85.11	76.68
+RU	84.72	75.93
+RO	86.80	78.52
+SMOTE [6]	85.24	70.79
+Focal Loss [32]	86.29	78.69
+DRC [7]	87.71	81.26
+Ours	90.10	83.69

Table 8: Performance comparison of different task orders on CIFAR100 in CIL.

Task Order	10 steps		20 steps		50 steps	
	Avg	Last	Avg	Last	Avg	Last
Order 1 (default)	**88.15**	82.19	88.50	80.91	**90.10**	**83.69**
Order 2	88.12	**82.28**	**88.55**	80.93	90.03	83.56
Order 3	87.96	82.26	88.42	**80.95**	89.87	83.41

Table 9: Ablation study on hyperparameter α on CIFAR100 under the 50-step CIL setting.

α	Avg	Last	α	Avg	Last
0.6	89.92	83.45	1.9	90.10	83.69
1.0	89.92	84.04	2.0	89.97	83.33
1.4	90.04	83.79	2.1	90.03	83.11
1.8	89.80	83.41	2.2	89.98	84.09

domly selects classes to form incremental tasks. The experimental results show that the proposed method consistently achieves superior performance across all LT-CIL configu-

Table 10: Performance comparison with different random seeds on CIFAR100 in Shuffled LT-CIL. Variations in random seeds lead to significantly different degrees of class imbalance in the incremental classification tasks.

Seed	5 steps		10 steps	
	Avg	Last	Avg	Last
0	83.08	78.10	82.48	77.31
1	83.60	79.09	84.15	78.14
2	82.53	77.28	84.34	78.90
3	82.78	78.91	83.16	78.18
4	84.48	78.21	85.61	78.03

rations, yielding a notable improvement of at least **5.40**% over the baseline model in terms of the "Avg" metric. These results indicate that the proposed method is highly effective in mitigating dynamic class imbalance, even under the more demanding LT-CIL benchmark.

To further analyze the behavior of our method under LT-CIL, we report per-step classification accuracy on CIFAR100 with 10 incremental steps. As shown in Fig. 3, the proposed method consistently achieves leading accuracy at every step in both the Ordered and Shuffled LT-CIL settings. This evidences that our method substantially improves upon MoE-Adapters [49] in dynamically imbalanced scenarios, reinforcing its advantage in real-world incremental learning applications.

4.3. Ablation Study

4.3.1 The Standard CIL Benchmark

Effectiveness of each component in the proposed method. To evaluate the impact of each component within the proposed method, we conduct a series of ablation studies focusing on the effectiveness of DMGDA and AMI. The experiments are performed on the CIFAR100 dataset under the 50-step setting, with performance reported using the "Avg" and "Last" metrics. The rehearsal mechanism is implemented using iCaRL by default, and in scenarios involving AMI, the standard exemplar reduction strategy is replaced accordingly. As shown in Tab. 6, applying iCaRL to the baseline yields a modest improvement of **1.51**% in "Avg" and **1.44**% in "Last", reflecting the impact of memory-based rehearsal. When DMGDA is incorporated into the baseline model, a notable performance increase is observed, with gains of **4.55**% in "Avg" and **6.95**% in "Last". In parallel, the addition of the AMI algorithm results in further enhancement, improving performance by **1.47**% in "Avg" and **3.28**% in "Last" compared to the baseline model with iCaRL. Finally, the integration of DMGDA with the AMI algorithm leads to the most substantial improvement, achieving a **6.50**% gain in "Avg" and an **8.45**% gain in "Last" compared to the baseline model. These results collectively confirm that each component of the proposed method makes a meaningful and complementary contribution to overall model performance.

Effectiveness of various methods for mitigating class imbalance. To validate the effectiveness of the proposed method in handling class imbalance, we conduct a comparative study against several approaches, including Random Oversampling (RO) and Random Undersampling (RU), SMOTE [6], Focal Loss [32], DRC [7]. As shown in Tab. 7, although these methods improve the performance of the baseline model when combined with a rehearsal mechanism, they consistently underperform relative to the proposed approach. These improvements highlight the superiority of our DMGDA in mitigating the dynamic class imbalance by correcting the prior probability based on Bayes' theorem, under the assumption of a multivariate Gaussian distribution, enabling more accurate calibration during incremental learning.

Effectiveness of fixed memory budget N. To evaluate the impact of fixed memory budget N on model performance, we conduct experiments under varying N settings, with the corresponding results illustrated in Fig. 4(a). In terms of the "Avg" metric, the performance exhibits a improvement of **1.36**% as N increases from 1,000 to 2,000. However, further enlarging N to 3,000 and 4,000 yields only marginal performance gains. This suggests that selecting an appropriate memory budget N, tailored to the characteristics of the specific task and dataset, is critical for optimizing model performance.

Effectiveness of different methods for selecting exemplars. To determine the most suitable strategy for exemplar selection, we conduct experiments comparing several approaches, including Random Selection, Cluster, and iCaRL [39]. The results, presented in Fig. 4(b), indicate that iCaRL consistently delivers superior performance. Consequently, we adopt iCaRL as the default exemplar selection method due to its strong compatibility with the proposed model.

Effectiveness of task order in CIL. We follow the class order configuration defined by Dytox [13], referred to as Order 1. To further investigate the sensitivity of model performance to task order, we introduce two additional configurations, Order 2 and Order 3, by applying a random shuffle strategy to the task sequence. The corresponding results are presented in Tab. 8. Consistent with the findings of De *et al.* [9], our results indicate that the model exhibits minimal sensitivity to variations in task order under the standard CIL benchmark.

Effectiveness of hyperparameter α. To analysis the model's sensitivity to hyperparameter α, we conduct experiments on the CIFAR100 with 50 steps using various α values, as shown in Tab. 9. The results suggest that the model demonstrates low sensitivity to changes in α. Moreover, we observe that the "Avg" metric increases as α approaches 1.9 and declines thereafter. Accordingly, we set $\alpha = 1.9$ to

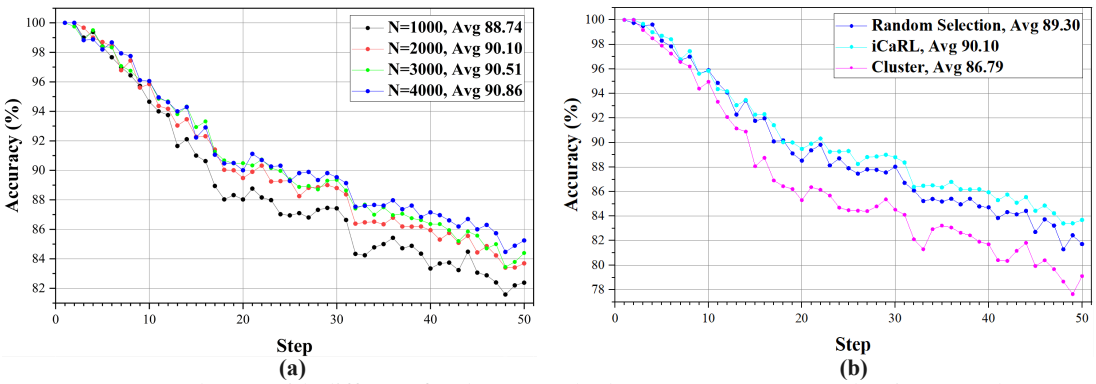

Figure 4: (a) Accuracy at each step with different fixed memory budgets on CIFAR100 under 50 steps. (b) Accuracy at each step using various methods for selecting exemplars. It demonstrates that iCaRL is the optimal choice in the proposed method.

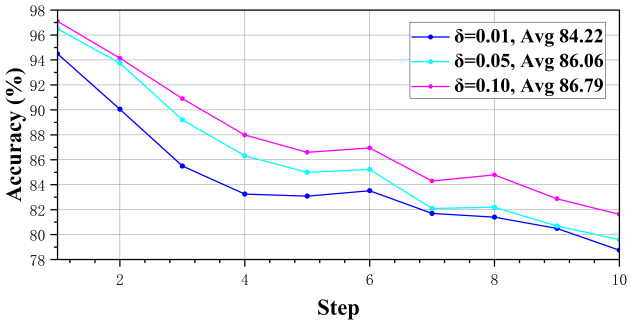

Figure 5: Accuracy at each step on CIFAR100 with 50 steps in Ordered LT-CIL under different imbalance factors δ . A smaller imbalance factor δ signifies more severe class imbalance. When $\delta = 1$, the LT-CIL benchmark reduces to the standard CIL benchmark.

achieve optimal average accuracy across incremental steps.

4.3.2 The LT-CIL Benchmark

Effectiveness of imbalance factor in LT-CIL. The imbalance factor δ in the LT-CIL benchmark plays a critical role in determining the degree of class distribution skewness, with smaller values corresponding to more severe imbalance. To assess the impact of δ, we perform experiments on CIFAR100 datset the 50-step setting, evaluating the model's performance at $\delta = 0.01$, 0.05, and 0.10. As shown in Fig. 5, the results demonstrate that larger values of δ, indicating milder class imbalance, consistently lead to improved accuracy across incremental steps.

Effectiveness of random seed in LT-CIL. In the Shuffled LT-CIL setting, the choice of random seed influences the ordering of classes with varying sample sizes across incremental tasks, thereby introducing different levels of class imbalance into the training data. To investigate this effect, we conduct experiments on CIFAR100 with 5-step and 10-step, comparing random seeds 0, 1, 2, 3, and 4. The results are summarized in Tab. 10. The experimental findings reveal that the proposed method exhibits notable variability in performance depending on the random seed. By default, we use seed 4, which achieves the best performance under the "Avg" metric.

Analysis of model complexity, training efficiency and inference latency. The proposed method does not introduce any additional training parameters compared to the baseline model (*i.e.*, MoE-Adapters [49]). Therefore, the proposed method has the same trainable parameters as the baseline model, specifically **4.02**M, which is significantly fewer than the **153.64**M in ZSCL [52]. We use training time as a metric to represent training efficiency. The training and inference times are measured during incremental steps on CIFAR100 (20 steps) using an NVIDIA GeForce RTX 1080 GPU. The inference time tested on the NVIDIA GeForce RTX 1080 GPU is sufficient to reflect the deployment efficiency on resource-constrained devices. Compared to the training time and inference time of MoE-Adapters, which are **70.22**s and **21.75**s for an incremental step, the proposed method achieves significant performance improvements, although it incurs an increase in time, with **97.21**s for training time and **24.50**s for inference time. We can conclude that our method demonstrates deployment efficiency on resource-constrained devices.

5. Limitation and Future work

The proposed method is primarily designed for CIL within a single domain, and does not currently address the more challenging Cross-Domain Class-Incremental Learning (CCIL) scenario, where new classes may originate from diverse domains, such as OxfordPet [36], EuroSAT [22], and StanfordCars [29]. In the standard CIL benchmark, class labels are disjoint across tasks, thereby eliminating the need to retain task identity information in memory for rehearsal-based methods [39, 13, 18]. Nevertheless, CCIL introduces the additional complexity of overlapping class labels across datasets, which necessitates rehearsal mechanisms for preserving and distinguishing task-specific identities. As a potential future direction, our method could be

extended by integrating memory blocks annotated with task identities, which may enable adaptation to CCIL scenarios. We also plan to construct a comprehensive and challenging CCIL benchmark to systematically evaluate the effectiveness of the proposed framework. This line of research holds promise for advancing continual learning in more realistic and diverse environments.

6. Conclusion

Class-Incremental Learning (CIL) plays a critical role in enabling intelligent systems to learn continually in dynamic real-world environments. In this study, we observe that the performance gains from the rehearsal mechanism are limited when applied to Vision-Language Model (VLM)-based CIL methods, primarily due to dynamic data imbalance. This imbalance causes the adapted CLIP model to exhibit a bias toward newly introduced classes. To mitigate this issue, we propose DMGDA, a novel approach that addresses dynamic class imbalance by calibrating prior probabilities through inverse frequency adjustment. This yields a non-parametric, debiased classifier that enhances the model's stability across incremental tasks. Furthermore, we incorporate an Adaptive Memory Iteration (AMI) strategy into the rehearsal mechanism. AMI adaptively selects exemplars that better align with the test distribution of previous tasks, thereby preserving historical knowledge more effectively. Extensive experimental results across both standard CIL and Long-Tailed CIL (LT-CIL) benchmarks demonstrate that our method consistently outperforms existing approaches, underscoring its effectiveness and strong generalization capabilities.

Acknowledgement

This work was supported by National Natural Science Foundation of China [92467301, U24A20277, 92267205, 62225207, 62476260, 62436008, 62503466, 92367301], the National Key Research, and Development Program of China [2024YFB4711103], Natural Science Foundation of Liaoning Province [2024-MSBA-83, 2025-MS-085], the Fundamental Research Funds for the Central Universities under Grant WK2100000057, the National Program for Funded Postdoctoral Researchers [GZB20230805], Fundamental Research Project of SIA,[2024JC1K10, 2024JC3K03], the State Key Laboratory of Robotics and Intelligent Systems of China [2025-Z12].

References

[1] N. Asadi, M. Davari, S. Mudur, R. Aljundi, and E. Belilovsky. Prototype-sample relation distillation: towards replay-free continual learning. In *International Conference on Machine Learning*, pages 1093–1106, 2023. 1, 2, 3

[2] C. M. Bishop and N. M. Nasrabadi. *Pattern Recognition and Machine Learning*, volume 4. 2006. 6

[3] J. Byrd and Z. Lipton. What is the effect of importance weighting in deep learning? In *International Conference on Machine Learning*, pages 872–881, 2019. 4

[4] F. M. Castro, M. J. Marín-Jiménez, N. Guil, C. Schmid, and K. Alahari. End-to-end incremental learning. In *European Conference on Computer Vision*, pages 233–248, 2018. 2, 4, 9, 10

[5] A. Chaudhry, P. K. Dokania, T. Ajanthan, and P. H. Torr. Riemannian walk for incremental learning: Understanding forgetting and intransigence. In *European Conference on Computer Vision*, pages 532–547, 2018. 1, 3

[6] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer. SMOTE: synthetic minority over-sampling technique. *Journal of Artificial Intelligence Research*, 16:321–357, 2002. 2, 4, 10, 11

[7] X. Chen and X. Chang. Dynamic residual classifier for class incremental learning. In *IEEE/CVF International Conference on Computer Vision*, pages 18743–18752, 2023. 10, 11

[8] Y. Cui, M. Jia, T.-Y. Lin, Y. Song, and S. Belongie. Class-balanced loss based on effective number of samples. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9268–9277, 2019. 4

[9] M. De Lange, R. Aljundi, M. Masana, S. Parisot, X. Jia, A. Leonardis, G. Slabaugh, and T. Tuytelaars. A continual learning survey: Defying forgetting in classification tasks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(7):3366–3385, 2021. 11

[10] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. Imagenet: A large-scale hierarchical image database. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 248–255, 2009. 8

[11] Y. Ding, L. Liu, C. Tian, J. Yang, and H. Ding. Don't stop learning: Towards continual learning for the clip model. *arXiv preprint arXiv:2207.09248*, 2022. 9, 10

[12] A. Douillard, M. Cord, C. Ollion, T. Robert, and E. Valle. Podnet: Pooled outputs distillation for small-tasks incremental learning. In *European Conference on Computer Vision*, pages 86–102, 2020. 9, 10

[13] A. Douillard, A. Ramé, G. Couairon, and M. Cord. Dytox: Transformers for continual learning with dynamic token expansion. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9285–9295, 2022. 2, 3, 8, 9, 10, 11, 12

[14] I. Eddine Marouf, S. Roy, E. Tartaglione, and S. Lathuilière. Rethinking class-incremental learning in the era of large pre-trained models via test-time adaptation. *arXiv e-prints*, pages arXiv–2310, 2023. 7

[15] W. Feng, Z. Wang, Q. Zhang, J. Gong, X. Xu, and Z. Fu. Hybrid rotation self-supervision and feature space normalization for class incremental learning. *Information Sciences*, 691:121618, 2025. 9

[16] R. M. French and N. Chater. Using noise to compute error surfaces in connectionist networks: A novel means of reducing catastrophic forgetting. *Neural Computation*, 14(7):1755–1769, 2002. 1

[17] R. Gao and W. Liu. Ddgr: Continual learning with deep diffusion-based generative replay. In *International Conference on Machine Learning*, pages 10744–10763, 2023. 1

[18] X. Gao, Y. He, S. Dong, J. Cheng, X. Wei, and Y. Gong. Dkt: Diverse knowledge transfer transformer for class incremental learning. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 24236–24245, 2023. 1, 2, 3, 12

[19] Y. Gao, C. Lang, F. Liu, C.-S. Foo, Y. Cao, L. Sun, and Y. Wei. Mining semantic correlations between mispredictions and corrections for interactive semantic segmentation. *IEEE Transactions on Neural Networks and Learning Systems*, 2024. 4

[20] D. Goswami, Y. Liu, B. Twardowski, and J. Van De Weijer. Fecam: Exploiting the heterogeneity of class distributions in exemplar-free continual learning. *Advances in Neural Information Processing Systems*, 36:6582–6595, 2023. 1, 2, 3

[21] H. He, Y. Bai, E. A. Garcia, and S. Li. Adasyn: Adaptive synthetic sampling approach for imbalanced learning. In *IEEE World Congress on Computational Intelligence*, pages 1322–1328, 2008. 4

[22] P. Helber, B. Bischke, A. Dengel, and D. Borth. Eurosat: A novel dataset and deep learning benchmark for land use and land cover classification. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 12(7):2217–2226, 2019. 12

[23] S. Hou, X. Pan, C. C. Loy, Z. Wang, and D. Lin. Lifelong learning via progressive distillation and retrospection. In *European Conference on Computer Vision*, pages 437–452, 2018. 10

[24] S. Hou, X. Pan, C. C. Loy, Z. Wang, and D. Lin. Learning a unified classifier incrementally via rebalancing. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 831–839, 2019. 2, 9, 10

[25] E. J. Hu, Y. Shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, W. Chen, et al. Lora: Low-rank adaptation of large language models. *International Conference on Learning Representations*, 1(2):3, 2022. 4

[26] Z. Hu, Y. Li, J. Lyu, D. Gao, and N. Vasconcelos. Dense network expansion for class incremental learning. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11858–11867, 2023. 9

[27] J. Kirkpatrick, R. Pascanu, N. Rabinowitz, J. Veness, G. Desjardins, A. A. Rusu, K. Milan, J. Quan, T. Ramalho, A. Grabska-Barwinska, et al. Overcoming catastrophic forgetting in neural networks. *Proceedings of the National Academy of Sciences*, 114(13):3521–3526, 2017. 9

[28] Y. Kong, L. Liu, M. Qiao, Z. Wang, and D. Tao. Trust-region adaptive frequency for online continual learning. *International Journal of Computer Vision*, 131(7):1825–1839, 2023. 1

[29] J. Krause, M. Stark, J. Deng, and L. Fei-Fei. 3d object representations for fine-grained categorization. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 554–561, 2013. 12

[30] J. Li, S. Sun, K. Zhang, J. Zhang, and L. Zhuo. Single-stage zero-shot object detection network based on clip and pseudo-labeling. *International Journal of Machine Learning and Cybernetics*, 16(2):1055–1070, 2025. 4

[31] Z. Li and D. Hoiem. Learning without forgetting. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(12):2935–2947, 2017. 9, 10

[32] T. Lin, P. Goyal, R. B. Girshick, K. He, and P. Dollár. Focal loss for dense object detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 42(2):318–327, 2020. 2, 4, 10, 11

[33] X. Liu, Y.-S. Hu, X.-S. Cao, A. D. Bagdanov, K. Li, and M.-M. Cheng. Long-tailed class incremental learning. In *European Conference on Computer Vision*, pages 495–512, 2022. 3, 4, 8, 9, 10

[34] Y. Liu, S. Parisot, G. Slabaugh, X. Jia, A. Leonardis, and T. Tuytelaars. More classifiers, less forgetting: A generic multi-classifier paradigm for incremental learning. In *European Conference on Computer Vision*, pages 699–716, 2020. 9

[35] S. Ning, L. Qiu, Y. Liu, and X. He. Hoiclip: Efficient knowledge transfer for hoi detection with vision-language models. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 23507–23517, 2023. 4

[36] O. M. Parkhi, A. Vedaldi, A. Zisserman, and C. Jawahar. Cats and dogs. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3498–3505, 2012. 12

[37] F. Pelosin, S. Jha, A. Torsello, B. Raducanu, and J. van de Weijer. Towards exemplar-free continual learning in vision transformers: an account of attention, functional and weight regularization. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3820–3829, 2022. 1, 3

[38] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, et al. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning*, pages 8748–8763, 2021. 2, 3

[39] S.-A. Rebuffi, A. Kolesnikov, G. Sperl, and C. H. Lampert. icarl: Incremental classifier and representation learning. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2001–2010, 2017. 1, 2, 3, 7, 9, 10, 11, 12

[40] A. Roy, V. K. Verma, S. Voonna, K. Ghosh, S. Ghosh, and A. Das. Exemplar-free continual transformer with convolutions. In *IEEE/CVF International Conference on Computer Vision*, pages 5897–5907, 2023. 1, 2, 3

[41] H. Shin, J. K. Lee, J. Kim, and J. Kim. Continual learning with deep generative replay. *Advances in Neural Information Processing Systems*, 30, 2017. 1

[42] X. Tao, X. Chang, X. Hong, X. Wei, and Y. Gong. Topology-preserving class-incremental learning. In *European Conference on Computer Vision*, pages 254–270, 2020. 10

[43] G. M. Van de Ven and A. S. Tolias. Three scenarios for continual learning. *arXiv preprint arXiv:1904.07734*, 2019. 1

[44] V. Vidit, M. Engilberge, and M. Salzmann. Clip the gap: A single domain generalization approach for object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3219–3229, 2023. 4

[45] M. Wu, J. Gu, Y. Shen, M. Lin, C. Chen, and X. Sun. End-to-end zero-shot hoi detection via vision and language knowl-

edge distillation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pages 2839–2846, 2023. 4

[46] Y. Wu, Y. Chen, L. Wang, Y. Ye, Z. Liu, Y. Guo, and Y. Fu. Large scale incremental learning. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 374–382, 2019. 9

[47] M. Xu, Z. Zhang, F. Wei, Y. Lin, Y. Cao, H. Hu, and X. Bai. A simple baseline for open-vocabulary semantic segmentation with pre-trained vision-language model. In *European Conference on Computer Vision*, pages 736–753, 2022. 4

[48] S. Yan, J. Xie, and X. He. Der: Dynamically expandable representation for class incremental learning. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3014–3023, 2021. 4, 8, 9, 10

[49] J. Yu, Y. Zhuge, L. Zhang, P. Hu, D. Wang, H. Lu, and Y. He. Boosting continual learning of vision-language models via mixture-of-experts adapters. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 23219–23230, 2024. 2, 4, 8, 9, 10, 11, 12

[50] S. Yun, S. H. Park, P. H. Seo, and J. Shin. Ifseg: Image-free semantic segmentation via vision-language model. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2967–2977, 2023. 4

[51] B. Zhao, X. Xiao, G. Gan, B. Zhang, and S.-T. Xia. Maintaining discrimination and fairness in class incremental learning. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13205–13214, 2020. 1, 2, 3, 4

[52] Z. Zheng, M. Ma, K. Wang, Z. Qin, X. Yue, and Y. You. Preventing zero-shot transfer degradation in continual learning of vision-language models. In *IEEE/CVF International Conference on Computer Vision*, pages 19125–19136, 2023. 4, 9, 10, 12

[53] D.-W. Zhou, Y. Zhang, Y. Wang, J. Ning, H.-J. Ye, D.-C. Zhan, and Z. Liu. Learning without forgetting for vision-language models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2025. 9, 10

[54] Y. Zhou, L. Liu, and C. Gou. Learning from observer gaze: Zero-shot attention prediction oriented by human-object interaction recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 28390–28400, 2024. 4

[55] Z. Zhou, Y. Lei, B. Zhang, L. Liu, and Y. Liu. Zegclip: Towards adapting clip for zero-shot semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11175–11185, 2023. 4

[56] F. Zhu, X.-Y. Zhang, C. Wang, F. Yin, and C.-L. Liu. Prototype augmentation and self-supervision for incremental learning. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5871–5880, 2021. 9