AIR: Analytic Imbalance Rectifier for Continual Learning

Di Fang[1], Yinan Zhu[1], Runze Fang[1], Cen Chen[1], Ziqian Zeng[2], Huiping Zhuang[2]*

[1]School of Future Technology, South China University of Technology, Guangzhou, China
[2]Shien-Ming Wu School of Intelligent Engineering, South China University of Technology, Guangzhou, China
{fti, ftynz, ftfrz}@mail.scut.edu.cn, {chencen, zqzeng, hpzhuang}scut.edu.cn

Abstract

Continual learning enables AI models to learn new data sequentially without retraining in real-world scenarios. Most existing methods assume the training data are balanced, aiming to reduce the catastrophic forgetting problem that models tend to forget previously generated data. However, data imbalance and the mixture of new and old data in real-world scenarios lead the model to ignore categories with fewer training samples. To solve this problem, we propose an analytic imbalance rectifier algorithm (AIR), a novel online exemplar-free continual learning method with an analytic (i.e., closed-form) solution for data-imbalanced class-incremental learning (CIL) and generalized CIL scenarios in real-world continual learning. AIR introduces an analytic re-weighting module (ARM) that calculates a re-weighting factor for each class for the loss function to balance the contribution of each category to the overall loss and solve the problem of imbalanced training data. AIR uses the least squares technique to give a non-discriminatory optimal classifier and its iterative update method in continual learning. Experimental results on multiple datasets show that AIR significantly outperforms existing methods in long-tailed and generalized CIL scenarios. The source code is available at https://github.com/fang-d/AIR.

1 Introduction

Humans can continuously learn new knowledge and expand their capabilities in real-world scenarios where data comes in a sequential data stream. Inspired by this ability, continual learning (CL) is proposed to enable AI models to learn new knowledge and capabilities without retraining and forgetting. Exploring this learning paradigm is significant for deep neural networks, especially for large pre-trained models, as it reduces the considerable cost of retraining models. Many methods have been carried out around class-incremental learning (CIL), one of the most challenging paradigms in CL for the severe catastrophic forgetting problem (McCloskey and Cohen 1989; Ratcliff 1990) that models tend to forget previously learned data.

Most existing CIL methods assume that the training dataset is balanced. However, in real-world scenarios, the number of samples for each category usually follows a long-tailed distribution, and the data of new and old classes can arrive mixed. Thus, CIL in real-world scenarios is roughly divided into two types: long-tail CIL (Liu et al. 2022) and generalized CIL (Aljundi et al. 2019). LT-CIL in Figure 1 (a) refers to the process of CIL where the number of samples for each category follows a long-tailed distribution, extending conventional CIL to the real-world imbalanced dataset. GCIL in Figure 1 (b) refers to the scenario where new and old classes may appear simultaneously in the same phase during CL, and it focuses on the dynamic changes in the number of training samples for each category, represented by the Si-blurry (Moon et al. 2023) setting. Besides, methods for GCIL can be applied to all CL settings, such as task-incremental learning and domain-incremental learning.

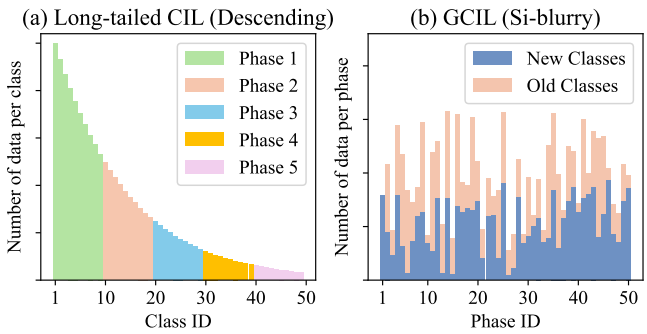

Figure 1: Different settings of imbalanced CIL.

Therefore, existing CL methods face a significant performance decline under real-world scenarios where the training dataset is usually imbalanced for the following reasons. (1) The number of samples for each category in real-world datasets is imbalanced, which leads to the model ignoring categories with fewer training samples (tail class) and tending to output categories with more training samples (head class). (2) Real-world data is often generated sequentially and requires models to learn continuously online. In GCIL, the ratio of the number of samples between different categories changes dynamically, making many long-tailed learning techniques inapplicable. (3) Many applications in real-world scenarios have rigorous privacy requirements and replay-based methods that rely on storing past training samples as exemplars cannot be applied in these scenarios.

Existing CL methods cannot solve the above three challenges at the same time. For example, to address the chal-

*Corresponding author (e-mail: hpzhuang@scut.edu.cn).

lenge (1), a common approach is to use a two-stage training method to alleviate the imbalance (Wu et al. 2019; Liu et al. 2022), but storing training samples as exemplars is required. For challenge (2), some methods introduce Transformer-based models and use techniques like P-Tuning for exemplar-free CIL (Wang et al. 2022c,b; Smith et al. 2023). However, the catastrophic forgetting problem is still significant in imbalanced training data. For challenge (3), state-of-the-art (SOTA) methods based on analytic CL (ACL) (Zhuang et al. 2022) solve catastrophic forgetting with a frozen pre-trained model to extract features and a ridge-regression (Hoerl and Kennard 1970) classifier with an analytic (i.e., closed-form) solution of the classifier. Existing ACL methods treat each training sample equally and optimize the classifier with the recursive least squares (RLS) algorithm, leading to a significant performance decline under data-imbalanced scenarios.

Head classes are likely to contribute more to the loss function than tail classes under imbalanced scenarios. This phenomenon emphasizes the head classes when optimizing the overall loss, resulting in discrimination and performance degradation. To address this issue, we propose the analytic imbalance rectifier (AIR), a novel online exemplar-free approach with an analytic solution for LT-CIL and GCIL scenarios in CL. AIR introduces an analytic re-weighting module (ARM) that calculates a re-weighting factor for each class for the loss function to balance the contribution of each category to the overall loss. We give an optimal unbiased classifier and its iterative update method. The key contributions of this paper are summarized as follows.

- We propose AIR, an online exemplar-free CL method for data-imbalanced scenarios with a closed-form solution.
- We point out that the unequal weight of each class in the loss function is the reason for discrimination and performance degradation under data-imbalanced scenarios.
- AIR introduces ARM that calculates a re-weighting factor for each class to balance the contribution of each class to the overall loss, giving an iterative analytic solution on imbalanced datasets.
- Evaluation under both the LT-CIL and GCIL scenarios shows that AIR significantly outperforms previous SOTA methods on several benchmark datasets.

2 Related Works

2.1 Conventional CIL

Conventional CIL focuses on classification scenarios where classes from different phases strictly disjoint in each incremental phase, and the data from each class are balanced or nearly balanced.

Classic CL Techniques Many outstanding works have proposed various methods to solve the problem of catastrophic forgetting in conventional CIL. Here, we introduce two types of them that significantly impact imbalanced CIL.

Exemplar replay is first proposed by iCaRL (Rebuffi et al. 2017) and retains past training samples as exemplars to hint models of old classes when learning new ones. The bigger memory for exemplars, the better performance that replay-based CIL achieves. Although it is a popular anti-forgetting technique that has inspired many excellent subsequent works (Hou et al. 2019; Douillard et al. 2020; Liu, Schiele, and Sun 2021; Wang et al. 2022a; Liu et al. 2023), storing original training samples poses a challenge for applying these methods in scenarios where stringent data privacy is mandated.

Regularization is used to prevent the activation and the parameter drift in CL. EWC (Kirkpatrick et al. 2017), Path Integral (Zenke, Poole, and Ganguli 2017), and RWalk (Chaudhry et al. 2018) apply weight regularization based on parameter importance evaluated by the Fisher Information Matrix. LwF (Li and Hoiem 2017), LfL (Jung et al. 2016), and DMC (Zhang et al. 2020) introduce Knowledge Distillation (Hinton, Vinyals, and Dean 2015) to prevent previous knowledge by distilling the activations of output, hidden layers, or both of them, respectively. Many regularization-based methods are exemplar-free but still face considerable catastrophic forgetting when there are many learning phases.

Analytic Continual Learning (ACL) ACL is a recently emerging CL branch exhibiting competitive performance due to its equivalence between CL and joint learning. Inspired by pseudoinverse learning (Guo and Lyu 2001, 2004), the ACL classifiers are trained with an RLS-like technique to generate a closed-form solution. ACIL (Zhuang et al. 2022) restructures CL programs into a recursive learning process, while RanPAC (McDonnell et al. 2023) gives an iterative one. To enhance the classification ability, the DS-AL (Zhuang et al. 2024c) introduces another recursive classifier to learn the residue, and the REAL (He et al. 2024) introduces the representation enhancing distillation to boost the plasticity of backbone networks. In addition, GKEAL (Zhuang et al. 2023) focuses on few-shot CL scenarios by leveraging a Gaussian kernel process that excels in zero-shot learning, AFL (Zhuang et al. 2024b) extends the ACL to federated learning, transitioning from temporal increment to spatial increment, Liu et al. (2024) apply similar techniques to the reinforcement learning, and GACL (Zhuang et al. 2024a) first extends ACL into GCIL. Our AIR is the first member of this branch to address the data imbalance issue in CIL.

CIL with Large Pre-trained Models Large pre-trained models bring backbone networks with strong feature representation ability to the CL field. On the one hand, inspired by fine-tuning techniques in NLP (Lester, Al-Rfou, and Constant 2021; Hu et al. 2022), DualPrompt (Wang et al. 2022b), CODA-Prompt (Smith et al. 2023), and MVP (Moon et al. 2023) introduce prompts into CL, while EASE (Zhou et al. 2024b) introduces a distinct lightweight adapter for each new task, aiming to create task-specific subspace. On the other hand, SimpleCIL (Zhou et al. 2024a) shows that with the help of a simple incremental classifier and a frozen large pre-trained model as a feature extractor that can bring generalizable and transferable feature embeddings, it can surpass many previous CL methods. Thus, it is with great potential to combine the large pre-trained models with the CL approaches with a powerful incremental classifier, such as SLDA (Hayes and Kanan 2020) and the ACL methods.

2.2 Long-Tailed CIL (LT-CIL)

To address data-imbalance problem in CIL, several approaches are proposed including LUCIR (Hou et al. 2019), BiC (Wu et al. 2019), PRS (Kim, Jeong, and Kim 2020), and CImbL (He, Wang, and Chen 2021). LST (Hu et al. 2020) and ActiveCIL (Belouadah et al. 2020) are designed for few-shot CL and active CL, respectively. Liu et al. (2022) propose a two-stage learning paradigm, bridging the existing CL methods to imbalanced CL. The experiments conducted by them on long-tailed datasets inspire a series of subsequent works (Chen and Chang 2023; Xu et al. 2024; He 2024; Wang et al. 2024; Hong et al. 2024).

Under online scenarios, CBRS (Chrysakis and Moens 2020) introduces a memory population approach for data balance, CBA (Wang et al. 2023) proposes an online bias adapter, LAS (Huang et al. 2024) introduces a logit adjust softmax to reduce inter-class imbalance, and DELTA (Raghavan, He, and Zhu 2024) introduces a decoupled learning approach to enhance learning representations and address the substantial imbalance.

2.3 Generalized CIL (GCIL)

GCIL simulates real-world incremental learning, as data category and size distributions could be unknown in one task. The GCIL arouses problems such as intra- and inter-phase forgetting and class imbalance (Moon et al. 2023). In the BlurryM (Aljundi et al. 2019) setting, $a\%$ of the classes disjoint between phases, with the rest appearing in each phase. The i-Blurry-N-M (Koh et al. 2022) setting has blurry phase boundaries and requires the model to perform inference at any time. The i-Blurry scenario has a fixed number of classes in each phase with the same proportion of new and old classes. In contrast, the Si-Blurry (Moon et al. 2023) setting has an ever-changing number of classes. It can effectively simulate newly emerging or disappearing data, highlighting the problem of uneven distribution in real-world scenarios.

Several approaches, such as GSS (Aljundi et al. 2019), RM (Bang et al. 2021), CLIB (Koh et al. 2022), DualPrompt (Wang et al. 2022b), and MVP (Moon et al. 2023), are proposed to address this issue.

3 Method

3.1 Class-Incremental Learning Problem

Let $\{\mathcal{D}_1, \mathcal{D}_2, \dots, \mathcal{D}_k, \dots\}$ be the classification dataset that arrives phase by phase sequentially to train the model. $\mathcal{D}_k = \{(\boldsymbol{\mathcal{X}}_{k,1}, y_{k,1}), (\boldsymbol{\mathcal{X}}_{k,2}, y_{k,2}), \cdots, (\boldsymbol{\mathcal{X}}_{k,N_k}, y_{k,N_k})\}$ of size N_k is the training set at phase k, where $\boldsymbol{\mathcal{X}}$ is the input tensor and y is an integer representing each distinct class. C_k is the maximum value of y from phase 1 to k, indicating the number of classes to classify at phase k.

In conventional CIL, classes from different phases are strictly disjoint and $C_k < C_{k+1}$. However, classes from the latter phases could either appear or not appear in the previous phases and $C_k \leq C_{k+1}$ in GCIL.

3.2 Analytic Classifier for Balanced Dataset

AIR extracts features with a frozen backbone network followed by a frozen buffer layer. The backbone network $f_{\text{backbone}}(\boldsymbol{\mathcal{X}}, \boldsymbol{\Theta})$ of AIR is a deep neural network, where $\boldsymbol{\Theta}$ is the network parameters either trained on the base training dataset $\mathcal{D}_1$ or pre-trained on a large-scale dataset. The buffer layer $\mathcal{B}$ non-linearly projects the features to a higher dimensional space. The extracted feature vector $\boldsymbol{x}$ is a raw vector, where

$$\boldsymbol{x} = \mathcal{B}(f_{\text{backbone}}(\boldsymbol{\mathcal{X}}, \boldsymbol{\Theta})). \tag{1}$$

There are several options for the buffer layer, such as a random projection matrix followed by an activation function in ACIL (Zhuang et al. 2022) and RanPAC (McDonnell et al. 2023) or a Gaussian kernel in GKEAL (Zhuang et al. 2023).

The feature extractor and the classification model are decoupled in AIR. The classifier maps an extracted feature to a one-hot raw vector. We can use $\boldsymbol{X}_k$ and $\boldsymbol{Y}_k$ to represent the dataset $\mathcal{D}_k$ at phase k by stacking the extracted features $\boldsymbol{x}$ and the corresponding one-hot labels $\text{onehot}(y)$ vertically. Similarly, by stacking $\boldsymbol{X}_k$ and $\boldsymbol{Y}_k$ from each phase, we can get $\boldsymbol{X}_{1:K}$ and $\boldsymbol{Y}_{1:K}$ representing overall training data.

AIR trains a ridge-regression model (Hoerl and Kennard 1970) with weight $\boldsymbol{W}_k$ at phase k as the classifier like existing ACL approaches, but uses a different loss function. However, when the training dataset is strictly balanced, the loss of AIR and existing ACL methods are the same

$$\mathcal{L}(\boldsymbol{W}_k) = \|\boldsymbol{X}_{1:K}\boldsymbol{W}_k - \boldsymbol{Y}_{1:K}\|_{\text{F}}^2 + \gamma\|\boldsymbol{W}_k\|_{\text{F}}^2, \tag{2}$$

where $\|\cdot\|_{\text{F}}$ indicates the Frobenius norm and γ is the coefficient of the regularization term.

The goal of AIR is to find the optimal weight under data-imbalanced scenarios, which is inspired by existing ACL methods that find a recursive form (Zhuang et al. 2022) or an iterative form (McDonnell et al. 2023) of the optimal solution at phase k

$$\hat{\boldsymbol{W}}_k = \underset{\boldsymbol{W}_k}{\text{argmin}}\, \mathcal{L}(\boldsymbol{W}_k) = (\sum_{t=1}^{k} \boldsymbol{A}_t + \gamma \boldsymbol{I})^{-1} (\sum_{t=1}^{k} \boldsymbol{C}_t), \tag{3}$$

where $\boldsymbol{A}_t = \boldsymbol{X}_t^\top \boldsymbol{X}_t$ is the *auto-correlation feature matrix*, and $\boldsymbol{C}_t = \boldsymbol{X}_t^\top \boldsymbol{Y}_t$ is the *cross-correlation feature matrix*.

3.3 Diagnosis: Classifier Need to Be Rectified

The loss function in Equation (2) teats each sample equally, bringing discrimination under the class imbalance scenarios.

We sort the samples at each phase by their labels to illustrate this issue. Let $\boldsymbol{x}_{k,i}^{(y)}$ be the i-th extracted features with label y at phase k. Similarly, we use $\boldsymbol{X}_k^{(y)}$ and $\boldsymbol{Y}_k^{(y)}$ to represent the extracted features and labels with the same label y at phase k. $\boldsymbol{X}_{1:k}^{(y)}$ and $\boldsymbol{Y}_{1:k}^{(y)}$ are all the features and labels with the same label y from phase 1 to k. $N_k^{(y)}$ is the number of samples at phase k with label y, and $N_{1:k}^{(y)} = \sum_{t=1}^{k} N_t^{(y)}$ is the number all training samples with label y.

Rearranging the samples by their labels, the training loss (2) can be written in

$$\begin{aligned}\mathcal{L}(\boldsymbol{W}_k) &= \sum_{t=1}^{k}\sum_{i=1}^{N_k} \|\boldsymbol{x}_{t,i}\boldsymbol{W}_k - \text{onehot}(y_{t,i})\|_{\text{F}}^2 + \gamma\|\boldsymbol{W}_k\|_{\text{F}}^2 \\ &= \sum_{y=0}^{C_k} \mathcal{L}^{(y)}(\boldsymbol{W}_k) + \gamma\|\boldsymbol{W}_k\|_{\text{F}}^2,\end{aligned} \tag{4}$$

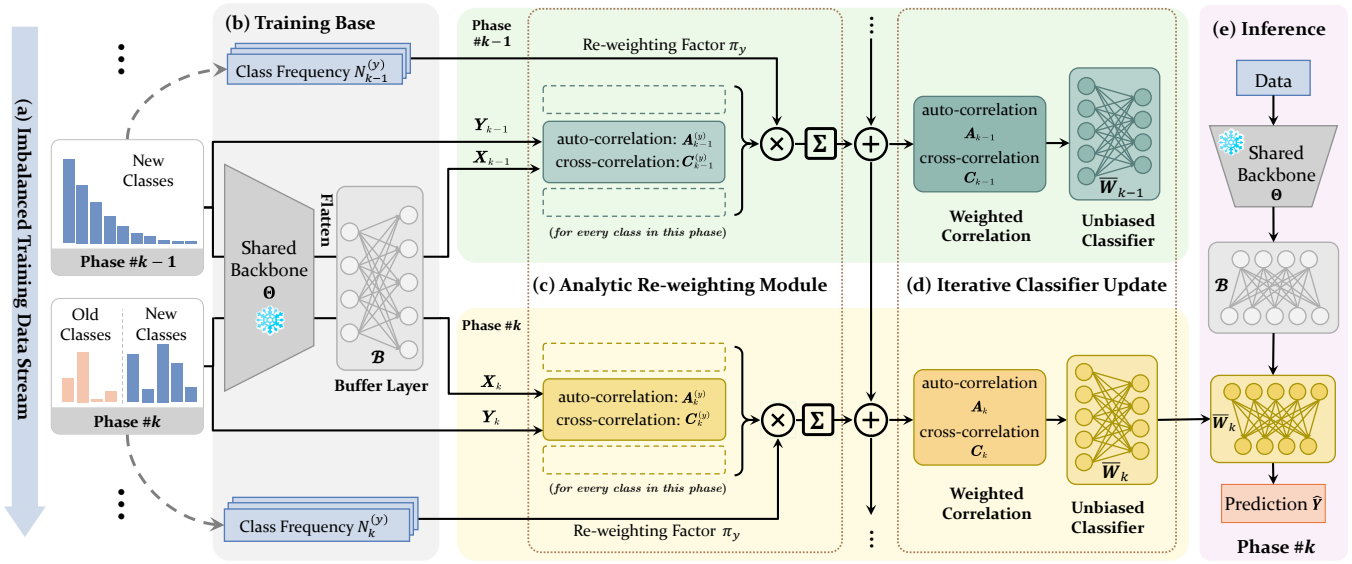

Figure 2: The flowchart of AIR, including (**a**) the input data stream that arrives phase by phase, where data is imbalanced, and the number of classes may change dynamically; (**b**) a frozen backbone network followed by a buffer layer that extracts features and maps into a higher dimensional space; (**c**) the analytic re-weighting module (ARM) calculating the re-weighting factor π_y for each class π_y; (**d**) the unbiased classifiers that are iteratively updated at each phase; (**d**) the frozen backbone network, the frozen buffer layer, and the unbiased classifier are used for inference.

where

$$\begin{aligned}\mathcal{L}^{(y)}(\boldsymbol{W}_k) &= \sum_{t=1}^{k}\sum_{i=1}^{N_t^{(y)}} \|\boldsymbol{x}_{t,i}^{(y)}\boldsymbol{W}_k - \text{onehot}(y)\|_{\text{F}}^2 \\ &= \|\boldsymbol{X}_{1:k}^{(y)}\boldsymbol{W}_k - \boldsymbol{Y}_{1:k}^{(y)}\|_{\text{F}}^2\end{aligned} \tag{5}$$

is the loss on the specific class y. The total loss $\mathcal{L}(\boldsymbol{W}_k)$ is the sum of the loss on each class $\mathcal{L}^{(y)}(\boldsymbol{W}_k)$ plus the regularization term $\gamma\|\boldsymbol{W}_k\|_{\text{F}}^2$.

Each training sample contributes equally to the total loss $\mathcal{L}(\boldsymbol{W}_k)$ in existing ACL approaches. In class-imbalance scenarios, head classes with more training samples are more likely to have a larger contribution $\mathcal{L}^{(y)}(\boldsymbol{W}_k)$ to the total loss. As the goal of the classifier is to find a classifier with a minimum loss, this imbalance in the contribution to the total loss leads to a bias towards the classes with more samples, causing discrimination under the data-imbalanced scenarios. Therefore, the ridge-regression classifier needs to be rectified under the data-imbalanced scenarios.

3.4 Analytic Imbalance Rectifier (AIR)

A simple but effective strategy is to re-weight the loss of each class. Inspired by this idea, we introduce ARM to balance the loss of each class, adding a scalar term π_y for each class to the overall loss function

$$\begin{aligned}\mathcal{L}_{\text{we}}(\boldsymbol{W}_k) &= \sum_{y=0}^{C_k}\pi_y\mathcal{L}^{(y)}(\boldsymbol{W}_k) + \gamma\|\boldsymbol{W}_k\|_{\text{F}}^2 \\ &= \sum_{y=0}^{C_k}\pi_y\|\boldsymbol{X}_{1:k}^{(y)}\boldsymbol{W}_k - \boldsymbol{Y}_{1:k}^{(y)}\|_{\text{F}}^2 + \gamma\|\boldsymbol{W}_k\|_{\text{F}}^2.\end{aligned} \tag{6}$$

Although the scalar term π_y for each class can be arbitrarily configured, we just set it to the reciprocal of the number of training samples (i.e., $\pi_y = 1/N_t^{(y)}$) in this paper, so that each class contributes equally to the global loss no matter how many training samples in this class.

The global optimal weight of the classifier $\bar{\boldsymbol{W}}_k$ can be obtained by mincing the weighted loss function $\mathcal{L}_{\text{we}}(\boldsymbol{W}_k)$.

Theorem 1. *The global optimal weight of the weighted classifier at phase k is*

$$\begin{aligned}\bar{\boldsymbol{W}}_k &= \underset{\boldsymbol{W}_k}{\text{argmin}}\ \mathcal{L}_{\text{we}}(\boldsymbol{W}_k) \\ &= (\sum_{y=0}^{C_k}\pi_y\boldsymbol{A}_{1:k}^{(y)} + \gamma\boldsymbol{I})^{-1}(\sum_{y=0}^{C_k}\pi_y\boldsymbol{C}_{1:k}^{(y)}),\end{aligned} \tag{7}$$

where

$$\begin{cases}\boldsymbol{A}_{1:k}^{(y)} = \sum_{t=1}^{k}\boldsymbol{X}_t^{(y)\top}\boldsymbol{X}_t^{(y)} = \boldsymbol{A}_{1:k-1}^{(y)} + \boldsymbol{X}_k^{(y)\top}\boldsymbol{X}_k^{(y)} \\ \boldsymbol{C}_{1:k}^{(y)} = \sum_{t=1}^{k}\boldsymbol{X}_t^{(y)\top}\boldsymbol{Y}_t^{(y)} = \boldsymbol{C}_{1:k-1}^{(y)} + \boldsymbol{X}_k^{(y)\top}\boldsymbol{Y}_k^{(y)}\end{cases} \tag{8}$$

can be obtained iteratively.

Proof. To minimize the loss function, we first calculate the gradient of the loss function $\mathcal{L}_{\text{we}}$, with respect to the weight:

$$\begin{aligned}&\frac{\partial}{\partial\boldsymbol{W}_k}\left(\sum_{y=0}^{C_k}\pi_y\|\boldsymbol{X}_{1:k}^{(y)}\boldsymbol{W}_k - \boldsymbol{Y}_{1:k}^{(y)}\|_{\text{F}}^2 + \gamma\|\boldsymbol{W}_k\|_{\text{F}}^2\right) \\ &= -2\sum_{y=0}^{C_k}\pi_y\boldsymbol{X}_{1:k}^{(y)\top}(\boldsymbol{Y}_{1:k}^{(y)} - \boldsymbol{X}_{1:k}^{(y)}\boldsymbol{W}_k) + 2\gamma\boldsymbol{W}_k\end{aligned} \tag{9}$$

Setting the gradient to zero matrix yields the optimal weight:

$$\begin{aligned}\bar{\boldsymbol{W}}_k &= (\sum_{y=0}^{C_k} \pi_y \boldsymbol{X}_{1:k}^{(y)\top} \boldsymbol{X}_{1:k}^{(y)} + \gamma \boldsymbol{I})^{-1} \sum_{y=0}^{C_k} (\pi_y \boldsymbol{X}_{1:k}^{(y)\top} \boldsymbol{Y}_{1:k}^{(y)}) \\ &= (\sum_{y=0}^{C_k} \pi_y \boldsymbol{A}_{1:k}^{(y)} + \gamma \boldsymbol{I})^{-1} (\sum_{y=0}^{C_k} \pi_y \boldsymbol{C}_{1:k}^{(y)}),\end{aligned} \tag{10}$$

which completes the proof. □

Therefore, we give the pseudo-code of AIR in Algorithm 1.

Algorithm 1 The training process of AIR for CIL.

procedure TRAINFORONEPHASE($\mathcal{D}_k$, γ, $\boldsymbol{\Theta}$)

▷ *Extract features.* ◁

for all $(\boldsymbol{\mathcal{X}}, y) \in \mathcal{D}_k$ **do**

$\boldsymbol{x} \leftarrow \mathcal{B}(f_{\text{backbone}}(\boldsymbol{\mathcal{X}}, \boldsymbol{\Theta}))$

$\boldsymbol{A}_k^{(y)} \leftarrow \boldsymbol{A}_k^{(y)} + \boldsymbol{x}^\top \boldsymbol{x}$

$\boldsymbol{C}_k^{(y)} \leftarrow \boldsymbol{C}_k^{(y)} + \boldsymbol{x}^\top \text{onehot}(y)$

$N^{(y)} \leftarrow N^{(y)} + 1$

▷ *Calculate the unbiased classifier.* ◁

for all $y \in \mathcal{D}_k$ **do**

$\pi_y \leftarrow 1/N^{(y)}$

$\boldsymbol{A}_k \leftarrow \boldsymbol{A}_k + \pi_y \boldsymbol{A}_k^{(y)}$

$\boldsymbol{C}_k \leftarrow \boldsymbol{C}_k + \pi_y \boldsymbol{C}_k^{(y)}$

▷ *Accumulate $\boldsymbol{A}_k$ and $\boldsymbol{C}_k$ to reduce memory.* ◁

$\boldsymbol{A}_{1:k} \leftarrow \boldsymbol{A}_{1:k-1} + \boldsymbol{A}_k$

$\boldsymbol{C}_{1:k} \leftarrow \boldsymbol{C}_{1:k-1} + \boldsymbol{C}_k$

return $\bar{\boldsymbol{W}}_k \leftarrow (\boldsymbol{A}_{1:k} + \gamma \boldsymbol{I})^{-1} \boldsymbol{C}_{1:k}$

3.5 Generalized AIR

The programming trick in Algorithm 1 that accumulates the sums of *auto-correlation feature matrix* and the *cross-correlation feature matrix* in $\boldsymbol{A}_{1:k}$ and $\boldsymbol{C}_{1:k}$ to reduce the memory is based on the assumption that classes from different phases are strictly disjoint in conventional CIL. In CIL

$$\boldsymbol{A}_{1:k} = \sum_{t=1}^{K} \sum_{y=0}^{C_t} \boldsymbol{A}_t^{(y)} = \sum_{y=0}^{C_k} \sum_{t=1}^{K} \boldsymbol{A}_t^{(y)} \tag{11}$$

as $\boldsymbol{A}_t^y = \boldsymbol{0}$ when $t \leq C_{t-1}$. The memory consumption of this algorithm is $\Theta(f^2 + fC_k)$, where f is the length of the feature vector $\boldsymbol{x}$.

However, classes training samples in each phase may either appear or not appear in the previous phases in GCIL scenarios, so that eq. (11) is no longer available in GCIL scenarios. To solve this problem, all we need to do is store $\boldsymbol{A}^{(y)} = \sum_{t=1}^{K} \boldsymbol{A}_t^{(y)}$ for each class. The memory consumption of the algorithm for GCIL is $\Theta(C_k(f^2 + fC_k))$, which could be a limitation of our algorithm when the feature size f and the number of classes C_k are both large.

The pseudo-code of generalized AIR for GCIL is listed in Algorithm 2.

Algorithm 2 The training process of AIR for GCIL.

procedure TRAINFORONEPHASE($\mathcal{D}_k$, γ, $\boldsymbol{\Theta}$)

for all $(\boldsymbol{\mathcal{X}}, y) \in \mathcal{D}_k$ **do**

$\boldsymbol{x} \leftarrow \mathcal{B}(f_{\text{backbone}}(\boldsymbol{\mathcal{X}}, \boldsymbol{\Theta}))$

$\boldsymbol{A}^{(y)} \leftarrow \boldsymbol{A}^{(y)} + \boldsymbol{x}^\top \boldsymbol{x}$

$\boldsymbol{C}^{(y)} \leftarrow \boldsymbol{C}^{(y)} + \boldsymbol{x}^\top \text{onehot}(y)$

$N^{(y)} \leftarrow N^{(y)} + 1$

$\pi_y \leftarrow 1/N^{(y)}$

$C_y \leftarrow \max(C_y, y)$

$\boldsymbol{A}_{1:k} \leftarrow \sum_{y=0}^{C_y} \pi_y \boldsymbol{A}^{(y)}$

$\boldsymbol{C}_{1:k} \leftarrow \sum_{y=0}^{C_y} \pi_y \boldsymbol{A}^{(y)}$

return $\bar{\boldsymbol{W}}_k \leftarrow (\boldsymbol{A}_{1:k} + \gamma \boldsymbol{I})^{-1} \boldsymbol{C}_{1:k}$

4 Experiments

4.1 Scenario 1: Long-Tailed CIL (LT-CIL)

We compare our AIR on CIFAR-100 (Krizhevsky, Nair, and Hinton 2009) and ImageNet-R (Hendrycks et al. 2021) under the LT-CIL scenario with baseline and SOTA methods.

Setting We follow Hong et al. (2024) to use the CIFAR-100 and the ImageNet-R datasets by splitting them into the long-tailed distribution. The imbalance ratio ρ, the ratio between the least and the most frequent class, is configured to $1/500$ for CIFAR-100 and $1/120$ for ImageNet-R[1]. The training/testing split is $80\%/20\%$ for ImageNet-R.

We follow Hong et al. (2024) to split the dataset into 10 incremental phases. The number of classes in each phase is 10 for CIFAR-100 and 20 for ImageNet-R. The class distribution in each phase is divided into 3 settings: ascending, descending, and shuffled. In the ascending scenario, the learning process starts with data-scarce phases followed by data-rich ones. In contrast, in the descending scenario, the learning process begins with data-rich tasks followed by data-scarce ones. In the shuffled scenario, the classes are randomly shuffled in each phase.

Evaluation Metrics We use the average accuracy $\mathcal{A}_{\text{avg}}$ and the last-phase accuracy $\mathcal{A}_{\text{last}}$ as the evaluation metrics. $\mathcal{A}_{\text{last}}$ is the average accuracy of each class in the last phase, while $\mathcal{A}_{\text{avg}}$ is the average accuracy of each phase.

Implementation Details We follow Hong et al. (2024) to use ViT-B/16 (Dosovitskiy et al. 2021) pre-trained on ImageNet as the shared backbone. For all the ACL methods, we follow ACIL (Zhuang et al. 2022) and RanPAC (McDonnell et al. 2023) to use the random buffer layer with a ReLU activation, projecting the extracted features to 2048. The coefficient of the regularization term of classifier γ of our methods is set to 1000. The batch size is configured to 64.

Result Analysis As shown in Table 1, AIR significantly outperforms other methods in most metrics on both CIFAR-100 and ImageNet-R datasets under the LT-CIL scenario.

[1]Hong et al. (2024) have not reported their imbalance ratio for ImageNet-R yet. Thus, we use the most challenging value so that the number of tail classes is 1 for the correctness of conclusions.

Method	Memory	CIFAR-100 (LT)						ImageNet-R (LT)					
		Ascending		Descending		Shuffled		Ascending		Descending		Shuffled	
		$\mathcal{A}_{\text{avg}}$	$\mathcal{A}_{\text{last}}$	$\mathcal{A}_{\text{avg}}$	$\mathcal{A}_{\text{last}}$	$\mathcal{A}_{\text{avg}}$	$\mathcal{A}_{\text{last}}$	$\mathcal{A}_{\text{avg}}$	$\mathcal{A}_{\text{last}}$	$\mathcal{A}_{\text{avg}}$	$\mathcal{A}_{\text{last}}$	$\mathcal{A}_{\text{avg}}$	$\mathcal{A}_{\text{last}}$
Fine-tuning	0	65.83	22.02	19.52	25.58	43.30	33.56	40.60	7.68	18.22	21.15	21.37	22.62
iCaRL (2017)	20/cls	53.00	28.73	41.70	26.88	48.62	31.02	48.41	29.55	24.40	29.17	40.21	23.02
ACIL / RanPAC (2022; 2023)	0	72.51	57.40	81.66	57.40	71.72	57.40	42.97	42.55	60.19	42.55	50.07	42.55
L2P (2022c)	0	66.51	50.26	53.50	48.73	51.43	49.43	50.05	31.72	27.24	29.42	30.19	26.21
Dual-Pormpt (2022b)	0	70.51	51.79	54.50	45.72	49.49	48.82	51.47	31.12	25.03	25.42	34.68	27.38
CODA-Prompt (2023)	0	81.91	58.98	54.54	41.84	60.90	42.56	52.39	35.21	28.21	32.62	40.02	34.78
DS-AL (2024c)	0	72.08	56.59	85.17	64.15	72.63	59.02	42.84	42.23	63.07	48.32	50.88	44.06
DAP (2024)	0	79.09	61.49	56.30	55.47	61.43	56.12	**58.47**	40.25	31.42	36.47	43.22	36.38
AIR	0	**82.39** ±0.03	**79.70** ±0.06	**89.43** ±0.02	**79.70** ±0.06	**85.75** ±0.92	**79.70** ±0.06	49.01 ±0.11	**55.49** ±0.06	**68.95** ±0.05	**55.49** ±0.06	**61.53** ±2.11	**55.49** ±0.06

Table 1: Accuracy (%) among AIR and other methods under the LT-CIL setting. Data **in bold** and underlined represent the **best** and the second-best results, respectively. We run experiments 7 times and show the results of AIR in "mean±standard error".

Gradient-based methods such as DAP usually achieve higher performance in average accuracy for better adaptation in imbalanced datasets. In contrast, ACL methods such as DS-AL reach higher last-phase accuracy for their non-forgetting property. AIR inherits the non-forgetting property of ACL and solves the data imbalance problem at the same time, thus achieving competitive average accuracy and outperforming the $\mathcal{A}_{\text{last}}$ of the SOTA method by over 7%.

Besides, the last-phase accuracy $\mathcal{A}_{\text{last}}$ of AIR are the same (i.e., 79.70% for CIFAR-100 and 55.49% for ImageNet-R) no matter the classes are in ascending, descending, or shuffled order, which indicates that AIR is robust to the data order in the LT-CIL scenario, keeping the same *weight-invariant property* as the other ACL approaches. For comparison, the last-phase accuracy of the gradient-based approaches is significantly affected by the order of the classes.

4.2 Scenario 2: Generalized CIL (GCIL)

We compare our AIR on CIFAR-100 (Krizhevsky, Nair, and Hinton 2009), ImageNet-R (Hendrycks et al. 2021), and Tiny-ImageNet (Deng et al. 2009) under the Si-blurry (Moon et al. 2023) scenario, one of the most challenging scenarios of GCIL with baseline and SOTA methods.

Setting We follow Moon et al. (2023) to use the Si-blurry scenario to test our proposed method. In the Si-blurry scenario, classes are partitioned into two groups: disjoint classes that cannot overlap between tasks and blurry classes that might reappear. The ratio of partition is controlled by the *disjoint class ratio* $\boldsymbol{r}_{\text{D}}$, which is defined as the ratio of the number of disjoint classes to the number of all classes. Each blurry task further conducts the blurry sample division by randomly extracting part of samples to assign to other blurry tasks based on *blurry sample ratio* $\boldsymbol{r}_{\text{B}}$, which is defined as the ratio of the extracted sample within samples in all blurry tasks. In this experiment, we set $\boldsymbol{r}_{\text{D}} = 0.1$ and $\boldsymbol{r}_{\text{B}} = 0.5$.

Evaluation Metrics We use the average accuracy $\mathcal{A}_{\text{avg}}$ and the last-phase accuracy $\mathcal{A}_{\text{last}}$ as the evaluation metrics, which are the same as the first experiment. Besides, we follow Moon et al. (2023) to validate the performance per 1000 samples and use the area under the curve (AUC) as the evaluation metric $\mathcal{A}_{\text{auc}}$.

Implementation Details We use DeiT-S (Touvron et al. 2021) pre-trained on 611 ImageNet classes after excluding 389 classes that overlap with CIFAR-100 and Tiny-ImageNet to prevent data leakage. The memory sizes of compared replay-based methods are set to 500 and 2000. For the ACL methods, we set the output of the buffer layer to 5000 and the coefficient of the regularization term γ by grid search. The best γ to AIR is 1000 on CIFAR-100 and ImageNet-R.

Result Analysis We can see from Table 2 that AIR outperforms all exemplar-free methods in all metrics on CIFAR-100, ImageNet-R, and Tiny-ImageNet datasets under the Si-blurry setting. The results are competitive, even compared with the replay-based methods.

Our AIR outperforms replay-based methods when the memory is limited (e.g., 500) and reaches a competitive result when the memory is 2000. Although replay-based methods can be further improved using more exemplars, they could bring more training memory and costs.

Compared with GACL, AIR shows a significant improvement in $\mathcal{A}_{\text{auc}}$ and $\mathcal{A}_{\text{avg}}$, indicating that the proposed method is more effective in the data-imbalanced scenario. However, for the balanced dataset in total (e.g., CIFAR-100), the last-phase accuracy of AIR and GACL is closed, showing that the GACL is just a particular case of AIR.

4.3 AIR Solves the Imbalance Issue

Classification Compared with ACIL, AIR has a more balanced classification result, indicating that our method gives a more balanced prediction for each class. As shown in the confusion matrix in Figure 3 (a), ACIL is more likely to predict the classes with more samples, resulting in worse performance for the tail classes. In contrast, the AIR gives a more balanced prediction for each class in Figure 3 (b).

Method	Memory	CIFAR-100			ImageNet-R			Tiny-ImageNet		
		$\mathcal{A}_{\text{auc}}$	$\mathcal{A}_{\text{avg}}$	$\mathcal{A}_{\text{last}}$	$\mathcal{A}_{\text{auc}}$	$\mathcal{A}_{\text{avg}}$	$\mathcal{A}_{\text{last}}$	$\mathcal{A}_{\text{auc}}$	$\mathcal{A}_{\text{avg}}$	$\mathcal{A}_{\text{last}}$
EWC++ (2017)	2000	$53.31_{\pm 1.70}$	$50.95_{\pm 1.50}$	$52.55_{\pm 0.71}$	$36.31_{\pm 0.72}$	$39.87_{\pm 1.35}$	$29.52_{\pm 0.43}$	$52.43_{\pm 0.52}$	$54.61_{\pm 1.54}$	$37.67_{\pm 0.77}$
ER (2019)	2000	$56.17_{\pm 1.84}$	$53.80_{\pm 1.46}$	$55.60_{\pm 0.69}$	$39.31_{\pm 0.70}$	$43.03_{\pm 1.19}$	$32.09_{\pm 0.44}$	$55.69_{\pm 0.47}$	$57.87_{\pm 1.42}$	$41.10_{\pm 0.57}$
RM (2021)	2000	$53.22_{\pm 1.82}$	$52.99_{\pm 1.69}$	$55.25_{\pm 0.61}$	$32.34_{\pm 1.88}$	$36.46_{\pm 2.23}$	$25.26_{\pm 1.08}$	$49.28_{\pm 0.43}$	$57.74_{\pm 1.57}$	$41.79_{\pm 0.34}$
MVP-R (2023)	2000	$\underline{63.09}_{\pm 2.01}$	$60.63_{\pm 2.20}$	$\underline{65.77}_{\pm 0.65}$	$\mathbf{47.96}_{\pm 0.78}$	$\mathbf{51.75}_{\pm 0.93}$	$41.40_{\pm 0.71}$	$62.85_{\pm 0.47}$	$64.95_{\pm 0.70}$	$50.72_{\pm 0.31}$
EWC++ (2017)	500	$48.31_{\pm 1.81}$	$44.56_{\pm 0.96}$	$40.52_{\pm 0.83}$	$32.81_{\pm 0.76}$	$35.54_{\pm 1.69}$	$23.43_{\pm 0.61}$	$45.30_{\pm 0.61}$	$46.34_{\pm 2.05}$	$27.05_{\pm 1.35}$
ER (2019)	500	$51.59_{\pm 1.94}$	$48.03_{\pm 0.80}$	$44.09_{\pm 0.80}$	$35.96_{\pm 0.72}$	$39.01_{\pm 1.54}$	$26.14_{\pm 0.44}$	$48.95_{\pm 0.58}$	$50.44_{\pm 1.71}$	$29.97_{\pm 0.75}$
RM (2021)	500	$41.07_{\pm 1.30}$	$38.10_{\pm 0.59}$	$32.66_{\pm 0.34}$	$22.45_{\pm 0.62}$	$22.08_{\pm 1.78}$	$9.61_{\pm 0.13}$	$36.66_{\pm 0.40}$	$38.83_{\pm 2.33}$	$18.23_{\pm 0.22}$
MVP-R (2023)	500	$59.25_{\pm 2.19}$	$56.03_{\pm 1.89}$	$56.79_{\pm 0.54}$	$44.33_{\pm 0.80}$	$47.25_{\pm 1.05}$	$35.92_{\pm 0.94}$	$56.78_{\pm 0.60}$	$58.34_{\pm 1.39}$	$40.49_{\pm 0.71}$
LwF (2017)	0	$40.71_{\pm 2.13}$	$38.49_{\pm 0.56}$	$27.03_{\pm 2.92}$	$29.41_{\pm 0.83}$	$31.95_{\pm 1.86}$	$19.67_{\pm 1.27}$	$39.88_{\pm 0.90}$	$41.35_{\pm 2.59}$	$24.93_{\pm 2.01}$
SLDA (2020)	0	$53.00_{\pm 3.85}$	$50.09_{\pm 2.77}$	$61.79_{\pm 3.81}$	$33.11_{\pm 3.17}$	$33.78_{\pm 1.76}$	$39.02_{\pm 1.30}$	$49.17_{\pm 4.41}$	$47.93_{\pm 4.43}$	$53.13_{\pm 2.29}$
Dual-Prompt (2022b)	0	$41.34_{\pm 2.59}$	$38.59_{\pm 0.68}$	$22.74_{\pm 3.40}$	$30.44_{\pm 0.88}$	$32.54_{\pm 1.84}$	$16.07_{\pm 3.20}$	$39.16_{\pm 1.13}$	$39.81_{\pm 3.03}$	$20.42_{\pm 3.37}$
L2P (2022c)	0	$42.68_{\pm 2.70}$	$39.89_{\pm 0.45}$	$28.59_{\pm 3.34}$	$30.21_{\pm 0.91}$	$32.21_{\pm 1.73}$	$18.01_{\pm 3.07}$	$41.67_{\pm 1.17}$	$42.53_{\pm 2.52}$	$24.78_{\pm 2.31}$
MVP (2023)	0	$48.95_{\pm 2.62}$	$48.95_{\pm 1.11}$	$36.97_{\pm 3.06}$	$36.64_{\pm 0.91}$	$38.09_{\pm 1.39}$	$25.03_{\pm 2.38}$	$46.80_{\pm 0.96}$	$47.83_{\pm 1.85}$	$29.31_{\pm 1.91}$
GACL (2024a)	0	$60.36_{\pm 1.34}$	$\underline{61.50}_{\pm 2.05}$	$\mathbf{72.33}_{\pm 0.07}$	$41.68_{\pm 0.78}$	$47.30_{\pm 0.84}$	$\underline{42.22}_{\pm 0.10}$	$\underline{63.23}_{\pm 1.74}$	$\underline{68.17}_{\pm 2.57}$	$\underline{64.17}_{\pm 0.07}$
AIR	0	$\mathbf{67.86}_{\pm 1.16}$	$\mathbf{68.82}_{\pm 1.53}$	$\mathbf{72.33}_{\pm 0.07}$	$\underline{45.49}_{\pm 0.93}$	$\underline{48.85}_{\pm 1.49}$	$\mathbf{42.88}_{\pm 0.18}$	$\mathbf{67.87}_{\pm 1.21}$	$\mathbf{70.34}_{\pm 1.76}$	$\mathbf{64.26}_{\pm 0.09}$

Table 2: Accuracy (%) among AIR and other methods under the Si-Blurry setting. Data **in bold** and underlined represent the **best** and the second-best results, respectively. We run all experiments 5 times and show the results in "mean±standard error".

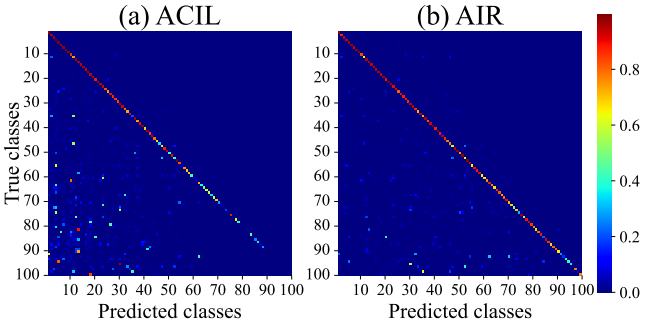

Figure 3: Last-phase performance on the testing set of CIFAR-100 under the descending LT-CIL scenario.

Accuracy As shown in Figure 4, AIR has a more balanced accuracy for each class. Although the accuracy of the head classes is slightly lower than ACIL, the accuracy of the middle and the tail classes is significantly improved, resulting in a better overall performance.

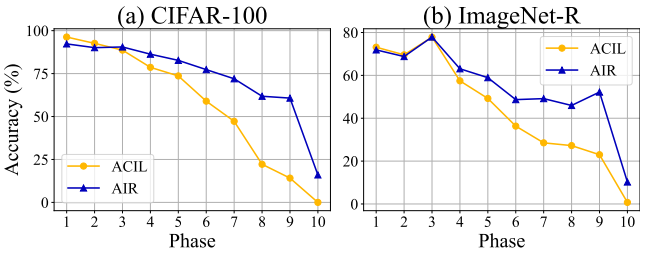

Figure 4: Last-phase accuracy for classes in each phase.

Weight We plot the L2 norm of the weight for each class in the last-phase classifier on CIFAR-100. Figure 5 (a) shows that the weight of the head classes is significantly larger than the tail classes in ACIL. That is why ACIL is more likely to predict the head classes. In contrast, AIR has a more balanced weight for each class shown in Figure 5 (b), showing that AIR learns a more balanced classifier.

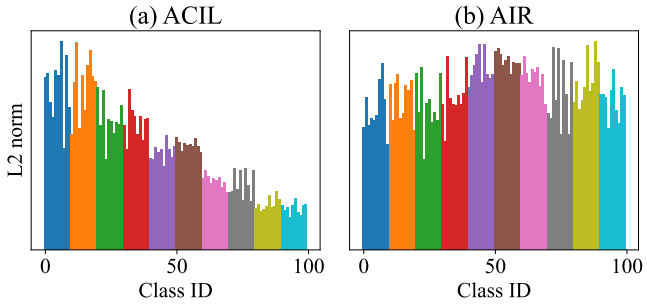

Figure 5: L2 norm of the weight for each class in the last-phase classifier under the descending LT-CIL scenario.

4.4 Analysis on the Loss

We validate our claim that the unequal weight of each class in the loss function is the reason for discrimination and performance degradation under data-imbalanced scenarios by experiments with the same setting as the LT-CIL experiment.

We train models under the descending order (where head classes are with smaller class IDs) and plot the average loss of samples in each class below in Figure 6. We use the mean square error (MSE) loss on the testing set of CIFAR-100. The losses of head classes of ACIL and DS-AL are significantly lower than the tail classes, indicating that the head

classes are more important than the tail classes in training, leading to discrimination. In contrast, the losses of each class in AIR are unbiased, addressing this issue.

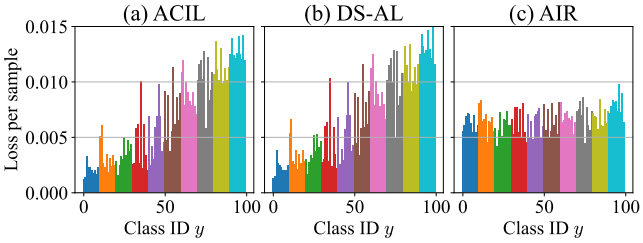

Figure 6: MSE loss on CIFAR-100 (LT) testing set.

We also plot the sum of loss on the training set on the training dataset in Figure 7. Classes with more training samples contribute more loss to the total loss. However, AIR can alleviate this issue by balancing the loss of each class. The sum loss of tail classes of AIR is much less than in other methods, leading to better performance.

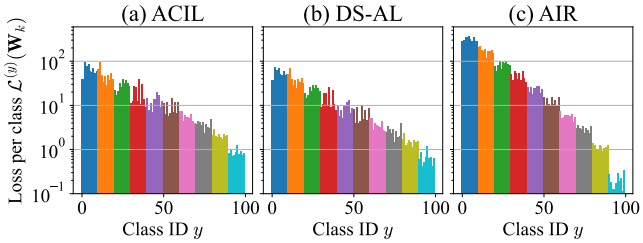

Figure 7: MSE loss on CIFAR-100 (LT) training set.

4.5 Discussion

Why AIR outperforms Gradient-based Methods? AIR significantly improves under the LT-CIL and the Si-blurry scenario compared with gradient-based methods. AIR, as a new member of ACL, inherits the non-forgetting property of ACL by giving an iterative closed-form solution, which avoids task-recency bias caused by gradient descent.

Why AIR outperforms Existing ACL Methods? Existing ACL methods are not designed for data-imbalanced scenarios. AIR introduces ARM to balance the loss of each class, treating each class equally in the total loss function, thus performing better and without discrimination.

5 Conclusions

In this paper, we point out that the unequal weight of each class in the loss function is the reason for discrimination and performance degradation under data-imbalanced scenarios. We propose AIR, a novel online exemplar-free CL method with an analytic solution for LT-CIL and GCIL scenarios to address this issue.

AIR introduces ARM, which calculates a weighting factor for each class for the loss function to balance the contribution of each category to the overall loss and solve the problem of imbalanced training data and mixed new and old classes without storing exemplars simultaneously.

Evaluations on the CIFAR-100, ImageNet-R, and TinyImageNet datasets under the LT-CIL and the Si-blurry scenarios show that our AIR outperforms SOTA methods in most metrics, indicating that AIR is effective in real-world data-imbalanced CIL scenarios.

References

Aljundi, R.; Lin, M.; Goujaud, B.; and Bengio, Y. 2019. Gradient based sample selection for online continual learning. In Wallach, H.; Larochelle, H.; Beygelzimer, A.; d'Alché-Buc, F.; Fox, E.; and Garnett, R., eds., *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc.

Bang, J.; Kim, H.; Yoo, Y.; Ha, J.-W.; and Choi, J. 2021. Rainbow Memory: Continual Learning With a Memory of Diverse Samples. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 8218–8227.

Belouadah, E.; Popescu, A.; Aggarwal, U.; and Saci, L. 2020. Active Class Incremental Learning for Imbalanced Datasets. In Bartoli, A.; and Fusiello, A., eds., *Computer Vision – ECCV 2020 Workshops*, 146–162. Cham: Springer International Publishing. ISBN 978-3-030-65414-6.

Chaudhry, A.; Dokania, P. K.; Ajanthan, T.; and Torr, P. H. S. 2018. Riemannian Walk for Incremental Learning: Understanding Forgetting and Intransigence. In Ferrari, V.; Hebert, M.; Sminchisescu, C.; and Weiss, Y., eds., *Computer Vision – ECCV 2018*, 556–572. Cham: Springer International Publishing. ISBN 978-3-030-01252-6.

Chen, X.; and Chang, X. 2023. Dynamic Residual Classifier for Class Incremental Learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 18743–18752.

Chrysakis, A.; and Moens, M.-F. 2020. Online Continual Learning from Imbalanced Data. In III, H. D.; and Singh, A., eds., *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, 1952–1961. PMLR.

Deng, J.; Dong, W.; Socher, R.; Li, L.-J.; Li, K.; and Fei-Fei, L. 2009. ImageNet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, 248–255.

Dosovitskiy, A.; Beyer, L.; Kolesnikov, A.; Weissenborn, D.; Zhai, X.; Unterthiner, T.; Dehghani, M.; Minderer, M.; Heigold, G.; Gelly, S.; Uszkoreit, J.; and Houlsby, N. 2021. An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. In *International Conference on Learning Representations*.

Douillard, A.; Cord, M.; Ollion, C.; Robert, T.; and Valle, E. 2020. PODNet: Pooled Outputs Distillation for Small-Tasks Incremental Learning. In Vedaldi, A.; Bischof, H.; Brox, T.; and Frahm, J.-M., eds., *Computer Vision – ECCV 2020*, 86–102. Cham: Springer International Publishing. ISBN 978-3-030-58565-5.

Guo, P.; and Lyu, M. R. 2001. Pseudoinverse Learning Algorithm for Feedforward Neural Networks. *Advances in Neural Networks and Applications*, 1: 321–326.

Guo, P.; and Lyu, M. R. 2004. A Pseudoinverse Learning Algorithm for Feedforward Neural Networks with Stacked Generalization Applications to Software Reliability Growth Data. *Neurocomputing*, 56: 101–121.

Hayes, T. L.; and Kanan, C. 2020. Lifelong Machine Learning With Deep Streaming Linear Discriminant Analysis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*.

He, C.; Wang, R.; and Chen, X. 2021. A Tale of Two CILs: The Connections Between Class Incremental Learning and Class Imbalanced Learning, and Beyond. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, 3559–3569.

He, J. 2024. Gradient Reweighting: Towards Imbalanced Class-Incremental Learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 16668–16677.

He, R.; Zhuang, H.; Fang, D.; Chen, Y.; Tong, K.; and Chen, C. 2024. REAL: Representation Enhanced Analytic Learning for Exemplar-free Class-incremental Learning. arXiv:2403.13522.

Hendrycks, D.; Basart, S.; Mu, N.; Kadavath, S.; Wang, F.; Dorundo, E.; Desai, R.; Zhu, T.; Parajuli, S.; Guo, M.; Song, D.; Steinhardt, J.; and Gilmer, J. 2021. The Many Faces of Robustness: A Critical Analysis of Out-of-Distribution Generalization. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 8340–8349.

Hinton, G.; Vinyals, O.; and Dean, J. 2015. Distilling the Knowledge in a Neural Network. arXiv:1503.02531.

Hoerl, A. E.; and Kennard, R. W. 1970. Ridge Regression: Biased Estimation for Nonorthogonal Problems. *Technometrics*, 12(1): 55–67.

Hong, C.; Jin, Y.; Kang, Z.; Chen, Y.; Li, M.; Lu, Y.; and Wang, H. 2024. Dynamically Anchored Prompting for Task-Imbalanced Continual Learning. In Larson, K., ed., *Proceedings of the Thirty-Third International Joint Conference on Artificial Intelligence, IJCAI-24*, 4127–4135. International Joint Conferences on Artificial Intelligence Organization. Main Track.

Hou, S.; Pan, X.; Loy, C. C.; Wang, Z.; and Lin, D. 2019. Learning a Unified Classifier Incrementally via Rebalancing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Hu, E. J.; Shen, Y.; Wallis, P.; Allen-Zhu, Z.; Li, Y.; Wang, S.; Wang, L.; and Chen, W. 2022. LoRA: Low-Rank Adaptation of Large Language Models. In *International Conference on Learning Representations*.

Hu, X.; Jiang, Y.; Tang, K.; Chen, J.; Miao, C.; and Zhang, H. 2020. Learning to Segment the Tail. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Huang, Z.; Li, T.; Yuan, C.; Wu, Y.; and Huang, X. 2024. Online Continual Learning via Logit Adjusted Softmax. *Transactions on Machine Learning Research*.

Jung, H.; Ju, J.; Jung, M.; and Kim, J. 2016. Less-forgetting Learning in Deep Neural Networks. arXiv:1607.00122.

Kim, C. D.; Jeong, J.; and Kim, G. 2020. Imbalanced Continual Learning with Partitioning Reservoir Sampling. In Vedaldi, A.; Bischof, H.; Brox, T.; and Frahm, J.-M., eds., *Computer Vision – ECCV 2020*, 411–428. Cham: Springer International Publishing. ISBN 978-3-030-58601-0.

Kirkpatrick, J.; Pascanu, R.; Rabinowitz, N.; Veness, J.; Desjardins, G.; Rusu, A. A.; Milan, K.; Quan, J.; Ramalho, T.; Grabska-Barwinska, A.; Hassabis, D.; Clopath, C.; Kumaran, D.; and Hadsell, R. 2017. Overcoming Catastrophic Forgetting in Neural Networks. *Proceedings of the National Academy of Sciences*, 114(13): 3521–3526.

Koh, H.; Kim, D.; Ha, J.-W.; and Choi, J. 2022. Online Continual Learning on Class Incremental Blurry Task Configuration with Anytime Inference. In *International Conference on Learning Representations*.

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

Lester, B.; Al-Rfou, R.; and Constant, N. 2021. The Power of Scale for Parameter-Efficient Prompt Tuning. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, 3045–3059. Online and Punta Cana, Dominican Republic: Association for Computational Linguistics.

Li, Z.; and Hoiem, D. 2017. Learning without Forgetting. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(12): 2935–2947.

Liu, X.; Hu, Y.-S.; Cao, X.-S.; Bagdanov, A. D.; Li, K.; and Cheng, M.-M. 2022. Long-Tailed Class Incremental Learning. In Avidan, S.; Brostow, G.; Cissé, M.; Farinella, G. M.; and Hassner, T., eds., *Computer Vision – ECCV 2022*, 495–512. Cham: Springer Nature Switzerland. ISBN 978-3-031-19827-4.

Liu, Y.; Li, Y.; Schiele, B.; and Sun, Q. 2023. Online Hyperparameter Optimization for Class-Incremental Learning. *Proceedings of the AAAI Conference on Artificial Intelligence*, 37(7): 8906–8913.

Liu, Y.; Schiele, B.; and Sun, Q. 2021. Adaptive Aggregation Networks for Class-Incremental Learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2544–2553.

Liu, Z.; Du, C.; Lee, W. S.; and Lin, M. 2024. Locality Sensitive Sparse Encoding for Learning World Models Online. In *The Twelfth International Conference on Learning Representations*, 1–19. Vienna, Austria: OpenReview.net.

McCloskey, M.; and Cohen, N. J. 1989. Catastrophic Interference in Connectionist Networks: The Sequential Learning Problem. In Bower, G. H., ed., *Psychology of Learning and Motivation*, volume 24, 109–165. Academic Press.

McDonnell, M. D.; Gong, D.; Parvaneh, A.; Abbasnejad, E.; and van den Hengel, A. 2023. RanPAC: Random Projections and Pre-trained Models for Continual Learning. In Oh, A.; Neumann, T.; Globerson, A.; Saenko, K.; Hardt, M.; and Levine, S., eds., *Advances in Neural Information Processing Systems*, volume 36, 12022–12053. Curran Associates, Inc.

Moon, J.-Y.; Park, K.-H.; Kim, J. U.; and Park, G.-M. 2023. Online Class Incremental Learning on Stochastic Blurry Task Boundary via Mask and Visual Prompt Tuning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 11731–11741.

Raghavan, S.; He, J.; and Zhu, F. 2024. DELTA: Decoupling Long-Tailed Online Continual Learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, 4054–4064.

Ratcliff, R. 1990. Connectionist Models of Recognition Memory: Constraints Imposed by Learning and Forgetting Functions. *Psychological Review*, 97(2): 285–308.

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

Rolnick, D.; Ahuja, A.; Schwarz, J.; Lillicrap, T.; and Wayne, G. 2019. Experience Replay for Continual Learning. In Wallach, H.; Larochelle, H.; Beygelzimer, A.; d'Alché-Buc, F.; Fox, E.; and Garnett, R., eds., *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc.

Smith, J. S.; Karlinsky, L.; Gutta, V.; Cascante-Bonilla, P.; Kim, D.; Arbelle, A.; Panda, R.; Feris, R.; and Kira, Z. 2023. CODA-Prompt: COntinual Decomposed Attention-Based Prompting for Rehearsal-Free Continual Learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 11909–11919.

Touvron, H.; Cord, M.; Douze, M.; Massa, F.; Sablayrolles, A.; and Jegou, H. 2021. Training data-efficient image transformers & distillation through attention. In Meila, M.; and Zhang, T., eds., *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, 10347–10357. PMLR.

Wang, F.-Y.; Zhou, D.-W.; Ye, H.-J.; and Zhan, D.-C. 2022a. FOSTER: Feature Boosting and Compression for Class-Incremental Learning. In Avidan, S.; Brostow, G.; Cissé, M.; Farinella, G. M.; and Hassner, T., eds., *Computer Vision – ECCV 2022*, 398–414. Cham: Springer Nature Switzerland. ISBN 978-3-031-19806-9.

Wang, Q.; Wang, R.; Wu, Y.; Jia, X.; and Meng, D. 2023. CBA: Improving Online Continual Learning via Continual Bias Adaptor. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 19082–19092.

Wang, X.; Yang, X.; Yin, J.; Wei, K.; and Deng, C. 2024. Long-Tail Class Incremental Learning via Independent Sub-prototype Construction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 28598–28607.

Wang, Z.; Zhang, Z.; Ebrahimi, S.; Sun, R.; Zhang, H.; Lee, C.-Y.; Ren, X.; Su, G.; Perot, V.; Dy, J.; and Pfister, T. 2022b. DualPrompt: Complementary Prompting for Rehearsal-Free Continual Learning. In Avidan, S.; Brostow, G.; Cissé, M.; Farinella, G. M.; and Hassner, T., eds., *Computer Vision – ECCV 2022*, 631–648. Cham: Springer Nature Switzerland. ISBN 978-3-031-19809-0.

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

Wu, Y.; Chen, Y.; Wang, L.; Ye, Y.; Liu, Z.; Guo, Y.; and Fu, Y. 2019. Large Scale Incremental Learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Xu, S.; Meng, G.; Nie, X.; Ni, B.; Fan, B.; and Xiang, S. 2024. Defying Imbalanced Forgetting in Class Incremental Learning. *Proceedings of the AAAI Conference on Artificial Intelligence*, 38(14): 16211–16219.

Zenke, F.; Poole, B.; and Ganguli, S. 2017. Continual Learning Through Synaptic Intelligence. In Precup, D.; and Teh, Y. W., eds., *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, 3987–3995. PMLR.

Zhang, J.; Zhang, J.; Ghosh, S.; Li, D.; Tasci, S.; Heck, L.; Zhang, H.; and Kuo, C.-C. J. 2020. Class-incremental Learning via Deep Model Consolidation. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, 1131–1140.

Zhou, D.-W.; Cai, Z.-W.; Ye, H.-J.; Zhan, D.-C.; and Liu, Z. 2024a. Revisiting Class-Incremental Learning with Pre-Trained Models: Generalizability and Adaptivity are All You Need. arXiv:2303.07338.

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

Zhuang, H.; Chen, Y.; Fang, D.; He, R.; Tong, K.; Wei, H.; Zeng, Z.; and Chen, C. 2024a. G-ACIL: Analytic Learning for Exemplar-Free Generalized Class Incremental Learning. arXiv:2403.15706.

Zhuang, H.; He, R.; Tong, K.; Fang, D.; Sun, H.; Li, H.; Chen, T.; and Zeng, Z. 2024b. Analytic Federated Learning. arXiv:2405.16240.

Zhuang, H.; He, R.; Tong, K.; Zeng, Z.; Chen, C.; and Lin, Z. 2024c. DS-AL: A Dual-Stream Analytic Learning for Exemplar-Free Class-Incremental Learning. *Proceedings of the AAAI Conference on Artificial Intelligence*, 38(15): 17237–17244.

Zhuang, H.; Weng, Z.; He, R.; Lin, Z.; and Zeng, Z. 2023. GKEAL: Gaussian Kernel Embedded Analytic Learning for Few-Shot Class Incremental Task. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 7746–7755.

Zhuang, H.; Weng, Z.; Wei, H.; Xie, R.; Toh, K.-A.; and Lin, Z. 2022. ACIL: Analytic Class-Incremental Learning with Absolute Memorization and Privacy Protection. In Koyejo, S.; Mohamed, S.; Agarwal, A.; Belgrave, D.; Cho, K.; and Oh, A., eds., *Advances in Neural Information Processing Systems*, volume 35, 11602–11614. Curran Associates, Inc.