GACL: Exemplar-Free Generalized Analytic Continual Learning

Huiping Zhuang[1*] **Yizhu Chen**[1*] **Di Fang**[1] **Run He**[1] **Kai Tong**[1]
Hongxin Wei[2] **Ziqian Zeng**[1†] **Cen Chen**[1,3,4†]
[1]South China University of Technology, China
[2]Southern University of Science and Technology, China
[3]Shenzhen Institute, Hunan University, China
[4]Pazhou Lab, China

Abstract

Class incremental learning (CIL) trains a network on sequential tasks with separated categories in each task but suffers from catastrophic forgetting, where models quickly lose previously learned knowledge when acquiring new tasks. The generalized CIL (GCIL) aims to address the CIL problem in a more real-world scenario, where incoming data have mixed data categories and unknown sample size distribution. Existing attempts for the GCIL either have poor performance or invade data privacy by saving exemplars. In this paper, we propose a new exemplar-free GCIL technique named generalized analytic continual learning (GACL). The GACL adopts analytic learning (a gradient-free training technique) and delivers an analytical (i.e., closed-form) solution to the GCIL scenario. This solution is derived via decomposing the incoming data into exposed and unexposed classes, thereby attaining a weight-invariant property, a rare yet valuable property supporting an equivalence between incremental learning and its joint training. Such an equivalence is crucial in GCIL settings as data distributions among different tasks no longer pose challenges to adopting our GACL. Theoretically, this equivalence property is validated through matrix analysis tools. Empirically, we conduct extensive experiments where, compared with existing GCIL methods, our GACL exhibits a consistently leading performance across various datasets and GCIL settings. Source code is available at `https://github.com/CHEN-YIZHU/GACL`.

1 Introduction

Class incremental learning (CIL) [1], an important form of continual learning, aims to effectively tune an off-the-shelf network on incoming new datasets, with data excluding various categories from its previous states. The CIL has gained significant traction due to its ability to refine learned models for new and unfamiliar data classes, eliminating the need to start the training process from scratch. This elimination of retraining saves valuable computational resources, which is especially important in the era of pre-trained models that have absorbed a massive amount of data.

One significant challenge in CIL is *catastrophic forgetting* [2, 3], which causes trained models to lose existing knowledge when gaining new information quickly. This can be attributed to the fundamental property of gradient-based iterative algorithms that impose a *task-recency* bias, i.e., predictions favor recently updated categories [4]. To the authors' knowledge, no solutions exist for these gradient-trained CIL models to fully tackle catastrophic forgetting.

*These authors contribute equally.
†Corresponding authors: Ziqian Zeng (zqzeng@scut.edu.cn) and Cen Chen (chencen@scut.edu.cn).

38th Conference on Neural Information Processing Systems (NeurIPS 2024).

On the other hand, traditional CIL assumes that the number of samples in each task is fixed and that new tasks are entirely disjoint from previous ones. This paradigm does not align with real-world scenarios, where training data may include both new and previously encountered categories, and the number of data points often exhibits arbitrariness in each task. This extended CIL setting is referred to as generalized CIL (GCIL) [5, 6]. Such an uneven task-wise distribution of training samples and data categories further complicates the forgetting issue. For instance, GCIL may lead to the neglect of minority samples within a batch, thereby undermining representation during the training process.

To mitigate catastrophic forgetting, a simple but effective approach is to replay historical samples. Replay-based CIL [1, 4] mitigates forgetting by storing a small number of samples from historical categories for the model to review while learning new information. However, this replay mechanism poses risks to data privacy. Thus, the exemplar-free CIL (EFCIL) without saving old exemplars gains prominence due to the increasing concern for privacy. However, many EFCIL methods perform poorly due to the task-recency bias caused by the nature of gradient-based algorithms [4]. Recently, this dilemma has been alleviated by the analytic continual learning (ACL) [7, 8], an emerging EFCIL branch that first achieves comparable or even more competitive performance over the replay-based CIL. This improvement occurs because, for the first time, ACL achieves a near "complete non-forgetting" by allowing an equivalence between the incremental learning and its joint training (i.e., the weight-invariant property).

The ACL provides a powerful toolbox for traditional EFCIL scenarios where data categories among training tasks are mutually exclusive. However, an apparent gap exists between the existing ACL techniques and the more desired and real-world GCIL scenario. Exploring the possibility of incorporating the weight-invariant property into the GCIL framework is both a significant and natural motivation, as it has the potential to enhance overall performance. To achieve this, we propose a generalized analytic continual learning (GACL), a new and compensated ACL member, offering a weight-invariant property solution to the GCIL. The key contributions are summarized as follows.

- We present the GACL, an exemplar-free technique that achieves the equivalence between the GCIL (with split incoming data) and its joint training (with data centralized in a single task).
- We theoretically establish the GACL's weight-invariant property. It is achieved and proved by separating the incoming data into exposed and unexposed components and aligning them structurally with matrix decomposition techniques.
- We isolate the distinctive component of the GACL, namely the *exposed class label gain* (ECLG), from the existing ACL. This module explains the feasibility of achieving GCIL's analytic learning, offering a high interpretability in the GCIL realm.
- Experiments on various benchmark datasets are presented, showing that the GACL outperforms the existing EFCIL by a large margin. It also exceeds most state-of-the-art replay-based methods.

2 Related Works

This section reviews existing methods for CIL and its more real-world counterpart, i.e., GCIL.

2.1 CIL Techniques

Existing CIL methods can be roughly divided into three categories: replay-based methods, regularization-based methods, and prototype-based methods.

The *replay-based CIL* methods such as iCaRL [1], LUCIR [4], PODNet [9], AANets [10], FOSTER [11], and OHO [12], retain past training samples as exemplars and utilize them during the learning of new ones. However, storing original training samples presents a significant challenge, particularly in scenarios with strict data privacy requirements.

The *regularization-based CIL* aims to design a loss function that prevents the change of activations or important weights. Methods such as the Less-forgetting learning [13] and the LwF [14] introduce knowledge distillation [15] into their loss function to prevent the forgetting caused by activation drift. EWC [16], EWC++, RWalk [17], and Rotate your Networks [18], introduce regularization that slows down learning on the weights important for old tasks by calculating the Fisher information matrix.

The *prototype-based CIL* maintains distinct prototypes for each category, which prevents overlapping representations of new and old categories. For example, the PASS [19] distinguishes prior categories

by augmenting feature prototypes. The SSRE technique [20] enhances the dissimilarity between old and new categories via selecting prototypes to incorporate with new samples into a distillation process. The FeTrIL [21] uses new representations to generate pseudo-features of old categories.

2.2 Analytic Continual Learning

The ACL is a recently developed EFCIL branch inspired by the analytic learning [22, 23, 24] where the training of neural networks yields a closed-form solution using least squares. The ACIL [7] first converts a continual learning problem to a batch recursive least-squares problem, eliminating the need to store samples by preserving the correlation matrix, and the RanPAC [25] applies this trick to pre-trained models. The GKEAL [8] focuses on the few-shot CIL scenarios by leveraging a Gaussian kernel projection. The DS-AL [26] introduces an additional linear classifier to learn the residue of the ACIL to enhance the plasticity, while the REAL [27] introduces the representation enhancing distillation to improve the backbone's generalization capabilities. The AFL [28] extends the ACL to federated learning, transitioning from temporal increment to spatial increment, and similar techniques are applied to the reinforcement learning [29]. The ACL is an emerging competitive CIL branch with a closed-form solution that leads to a valuable weight-invariant property, securing the equivalence between CIL and its joint learning. However, existing ACL methods are designed for the CIL scenario in which the categories of samples in each task must be entirely distinct. This restricts their applicability in real-world scenarios.

2.3 The Generalized Class Incremental Learning

The GCIL simulates real-world incremental learning, as distributions of data category and size could be unknown in one task. The GCIL arouses problems such as intra- and inter-task forgettings and the class imbalance problem [30]. The key GCIL properties can be summarized as follows: (i) the number of classes across different tasks is not fixed; (ii) classes shown in prior tasks could reappear in later tasks; (iii) training samples are imbalanced across different classes in each task [6] (See Appendix B).

There are several GCIL settings. In the BlurryM setting [5], $a\%$ of the classes are disjoint between tasks, while the remaining classes appear in every task. The i-Blurry-N-M [31] setting has blurry task boundaries and requires the model to perform anytime inference. However, the i-Blurry scenario has a fixed number of classes in each task with the same proportion of new and old classes. The Si-Blurry [30] is the most complex and realistic GCIL setting satisfying all three GCIL properties since it has an ever-changing number of classes and is capable of effectively simulating newly emerging or disappearing data, highlighting the problem of uneven distribution in real-world scenarios.

To address the issue of the GCIL, gradient-based sample selection methods such as the GSS-IQP and the GSS-Greedy are proposed by [5]. The RM [32] proposes a memory management strategy based on per-sample classification uncertainty and data augmentation, while the management in the CLIB [31] eliminates samples based on a per-sample importance score. The DualPrompt [33], as an EFCIL method, introduces the prompt-based learning to the CIL problem, and the MVP [30] proposes an instance-wise logit masking and contrastive visual prompt tuning loss.

3 The Proposed Method

In this section, we deliver details of the proposed GACL. We first define the learning problem. Then, we derive the GACL by employing matrix decomposition techniques. A corresponding theoretical analysis follows to indicate the interpretability of our work. An overview is depicted in Figure 1.

3.1 Problem Definition

We denote the complete set of available data as $\mathcal{D}$. When $\mathcal{D}$ is partitioned into a sequence of GCIL tasks, we assume that $\mathcal{D}_k^{\text{train}} \sim \{\boldsymbol{X}_k^{\text{train}}, \boldsymbol{Y}_k^{\text{train}}\}$ is the set of training samples that are present in task k. The training dataset $\mathcal{D}_k^{\text{train}}$ consists of labeled samples, where $\boldsymbol{X}_k^{\text{train}} \in \mathbb{R}^{N_k \times c \times w \times h}$ represents N_k input image samples with a shape of $c \times w \times h$. $\boldsymbol{Y}_k^{\text{train}} \in \mathbb{R}^{N_k \times d_{y_k}}$ represents N_k-stacked one-hot encoded label tensors with d_{y_k} classes that have been seen from task 1 to task k. $\mathcal{D}_k^{\text{test}} \sim \{\boldsymbol{X}_k^{\text{test}}, \boldsymbol{Y}_k^{\text{test}}\}$

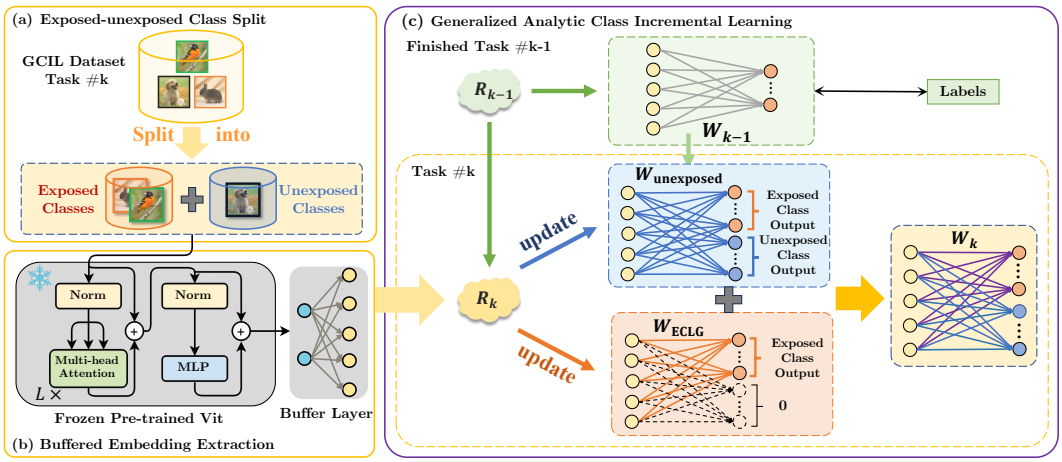

Figure 1: An overview of our proposed GACL. (a) Labels of the *exposed class* and the *unexposed class* are extracted in each GCIL task (see definition in Section 3.2), respectively. (b) A frozen pre-trained ViT and a buffer layer are utilized to extract features from the inputs. (c) The key to the recursively updated formulation of the GACL contains two components. The $\hat{\boldsymbol{W}}_{\text{unexposed}}^{(k)}$ takes in the contribution of unexposed class data (see (11)). The other is contributed by the ECLG module $\hat{\boldsymbol{W}}_{\text{ECLG}}^{(k)}$ (e.g., see (12)), which reflects the gain of exposed class data on the seen categories. The recursive formulation flows aided by the *autocorrelation memory matrix* $\boldsymbol{R}$ throughout the GCIL.

is the test dataset in task k. The goal of GCIL in task k is to train networks using $\mathcal{D}_k^{\text{train}}$ and evaluate their performance on the test dataset $\mathcal{D}_{1:k}^{\text{test}}$. Here, $\mathcal{D}_{1:k}$ denotes the joint dataset spanning tasks 1 to k.

3.2 Exposed-unexposed Class Split

In each GCIL task, classes may not appear exclusively. Hence, in any GCIL task k, we refer to classes that have appeared in previous tasks 1 to $k-1$ as the *exposed classes* of task k, while classes making their initial appearance are the *unexposed classes* of task k as shown in Figure 1 (a). This distinction helps to characterize the evolving nature of class occurrences throughout different GCIL tasks.

In a task-wise GCIL scenario, we can involve all class labels in a set $\mathcal{S}$. In task k, the set of the exposed class labels is denoted as $\mathcal{S}_{\text{exposed, k}} \subseteq \mathcal{S}$, while the set of unexposed class labels is marked by $\mathcal{S}_{\text{unexposed, k}} \subseteq \mathcal{S}$, where $\mathcal{S}_{\text{exposed, k}} \cap \mathcal{S}_{\text{unexposed, k}} = \varnothing$. Note that $\mathcal{S}_{\text{exposed, k}}$ and $\mathcal{S}_{\text{unexposed, k}}$ may evolve from task $k-1$ to task k, that is

$$\mathcal{S}_{\text{exposed, k}} = S_{\text{unexposed, k-1}} \cup \mathcal{S}_{\text{exposed, k-1}} = S_{\text{unexposed, k-1}} \cup S_{\text{unexposed, k-2}} \ldots \cup \mathcal{S}_{\text{unexposed, 1}}.$$

From the scope of exposed-unexposed classes, the d_{y_k} can be represented as $d_{y_k} = |\mathcal{S}_{\text{exposed, k}}| + |\mathcal{S}_{\text{unexposed, k}}| = d_{y_{k-1}} + |\mathcal{S}_{\text{unexposed, k}}|$, where $|\cdot|$ denotes the cardinality of a set.

In task k, given training dataset $\mathcal{D}_k^{\text{train}} \sim \{\boldsymbol{X}_k^{\text{train}}, \boldsymbol{Y}_k^{\text{train}}\}$, class labels $\boldsymbol{Y}_k^{\text{train}}$ can be partitioned due to the *exposed-unexposed split* as follows:

$$\boldsymbol{Y}_k^{\text{train}} = \left[\bar{\boldsymbol{Y}}_k^{\text{train}} \quad \tilde{\boldsymbol{Y}}_k^{\text{train}}\right], \tag{1}$$

where $\bar{\boldsymbol{Y}}_k^{\text{train}} \in \mathbb{R}^{N_k \times d_{y_{k-1}}}$ is the *exposed class label matrix* and $\tilde{\boldsymbol{Y}}_k^{\text{train}} \in \mathbb{R}^{N_k \times (d_{y_k} - d_{y_{k-1}})}$ is the *unexposed class label matrix*. They correspond to segments displaying the appearance of exposed classes and unexposed classes.

3.3 Buffered Embedding Extraction

The power of pre-trained models allows the GACL to adopt a frozen backbone from structures such as ViT [34] to extract the features of images shown in Figure 1 (b). Let

$$\boldsymbol{X}^{(\text{E})} = f_{\text{backbone}}(\boldsymbol{X}, \boldsymbol{\Theta}_{\text{backbone}}) \tag{2}$$

be the features extracted by the backbone, where $\mathbf{\Theta}_{\text{backbone}}$ indicates the backbone weight. Then, we use a buffer layer to project features, i.e.,

$$\boldsymbol{X}_i^{(\text{B})} = f_{\text{buffer}}(\boldsymbol{X}^{(\text{E})}), \tag{3}$$

where f_{buffer} indicates the operation of the buffer layer. Several options for the buffer layer exist, including a randomly initialized linear mapping in the ACIL [7] or a kernel embedding projection in the GKEAL [8]. The selection of the buffer layer is not our focus. For convenience, we follow the ACIL, taking the random linear projection followed by a non-linear activation function as the buffer layer, i.e. $f_{\text{buffer}}(\boldsymbol{X}^{(\text{E})}) = \text{ReLU}(\boldsymbol{X}^{(\text{E})}\boldsymbol{W}_{\text{B}})$, where the elements of the buffer layer weight $\boldsymbol{W}_{\text{B}}$ are randomly sampled from a normal distribution.

3.4 Generalized Analytic Class Incremental Learning

Here, we derive the GACL by partitioning training samples into unexposed and exposed categories, as shown in Figure 1 (c). Let $\boldsymbol{X}_{1:k}^{\text{total}}$ and $\boldsymbol{Y}_{1:k}^{\text{total}}$ be the accumulated feature and label matrices in task k, which can be extended from the accumulated matrices $\boldsymbol{X}_{1:k-1}^{\text{total}}$ and $\boldsymbol{Y}_{1:k-1}^{\text{total}}$ in task $k-1$ as follows.

$$\boldsymbol{X}_{1:k}^{\text{total}} = \begin{bmatrix} \boldsymbol{X}_{1:k-1}^{\text{total}} \\ \boldsymbol{X}_k^{(\text{B})} \end{bmatrix}, \quad \boldsymbol{Y}_{1:k}^{\text{total}} = \begin{bmatrix} \boldsymbol{Y}_{1:k-1}^{\text{total}} & \boldsymbol{0} \\ \bar{\boldsymbol{Y}}_k^{\text{train}} & \tilde{\boldsymbol{Y}}_k^{\text{train}} \end{bmatrix}.$$

Subsequently, one could formulate the learning problem in task k by a fully connected network (FCN) as the classifier

$$\underset{\boldsymbol{W}_{\text{FCN}}^{(k)}}{\text{argmin}} \left\| \boldsymbol{Y}_{1:k}^{\text{total}} - \boldsymbol{X}_{1:k}^{\text{total}} \boldsymbol{W}_{\text{FCN}}^{(k)} \right\|_{\text{F}}^2 + \gamma \left\| \boldsymbol{W}_{\text{FCN}}^{(k)} \right\|_{\text{F}}^2, \tag{4}$$

where $\|\cdot\|_{\text{F}}$ is Frobenius-norm, $\gamma \geq 0$ is the regularization term and $\boldsymbol{W}_{\text{FCN}}^{(k)}$ indicates the FCN layer weight. The optimal solution to (4) is

$$\hat{\boldsymbol{W}}_{\text{FCN}}^{(k)} = (\boldsymbol{X}_{1:k}^{\text{total}\top} \boldsymbol{X}_{1:k}^{\text{total}} + \gamma \boldsymbol{I})^{-1} \boldsymbol{X}_{1:k}^{\text{total}\top} \boldsymbol{Y}_{1:k}^{\text{total}}. \tag{5}$$

The goal of the GACL is then to obtain $\hat{\boldsymbol{W}}_{\text{FCN}}^{(k)}$ recursively from $\hat{\boldsymbol{W}}_{\text{FCN}}^{(k-1)}$ without directly involving historical samples (e.g., $\boldsymbol{X}_{1:k-1}^{\text{total}}$ and $\boldsymbol{Y}_{1:k-1}^{\text{total}}$). That is to solve

$$\underset{\boldsymbol{W}_{\text{FCN}}^{(k)}}{\text{argmin}} \left\| \begin{bmatrix} \boldsymbol{Y}_{1:k-1}^{\text{total}} & \boldsymbol{0} \\ \bar{\boldsymbol{Y}}_k^{\text{train}} & \tilde{\boldsymbol{Y}}_k^{\text{train}} \end{bmatrix} - \begin{bmatrix} \boldsymbol{X}_{1:k-1}^{\text{total}} \\ \boldsymbol{X}_k^{(\text{B})} \end{bmatrix} \boldsymbol{W}_{\text{FCN}}^{(k)} \right\|_{\text{F}}^2 + \gamma \left\| \boldsymbol{W}_{\text{FCN}}^{(k)} \right\|_{\text{F}}^2 \tag{6}$$

by recursively updating the previous-task weight $\hat{\boldsymbol{W}}_{\text{FCN}}^{(k)}$. To achieve this, we define an *autocorrelation memory matrix* as follows.

$$\boldsymbol{R}_k = (\boldsymbol{X}_{1:k}^{\text{total}\top} \boldsymbol{X}_{1:k}^{\text{total}} + \gamma \boldsymbol{I})^{-1}. \tag{7}$$

Accordingly, we summarize the recursive formulation of the proposed GACL in Theorem 3.1.

Theorem 3.1. *Let $\hat{\boldsymbol{W}}_{\text{FCN}}^{(k)}$ be the optimal estimation of* (6) *with all the training data from task* 1 *to task k. Then $\hat{\boldsymbol{W}}_{\text{FCN}}^{(k)}$ is equivalent to its recursive form*

$$\hat{\boldsymbol{W}}_{\text{FCN}}^{(k)} = \left[\hat{\boldsymbol{W}}_{\text{FCN}}^{(k-1)} - \boldsymbol{R}_k \boldsymbol{X}_k^{(\text{B})\top} \boldsymbol{X}_k^{(\text{B})} \hat{\boldsymbol{W}}_{\text{FCN}}^{(k-1)} + \boldsymbol{R}_k \boldsymbol{X}_k^{(\text{B})\top} \bar{\boldsymbol{Y}}_k^{\text{train}} \quad \boldsymbol{R}_k \boldsymbol{X}_k^{(\text{B})\top} \tilde{\boldsymbol{Y}}_k^{\text{train}} \right], \tag{8}$$

where

$$\boldsymbol{R}_k = \boldsymbol{R}_{k-1} - \boldsymbol{R}_{k-1} \boldsymbol{X}_k^{(\text{B})\top} (\boldsymbol{I} + \boldsymbol{X}_k^{(\text{B})} \boldsymbol{R}_{k-1} \boldsymbol{X}_k^{(\text{B})\top})^{-1} \boldsymbol{X}_k^{(\text{B})} \boldsymbol{R}_{k-1}. \tag{9}$$

Proof. See Appendix A. □

As indicated in Theorem 3.1, the weight $\hat{\boldsymbol{W}}_{\text{FCN}}^{(k)}$ in task k recursively obtained using the previous-task weight $\hat{\boldsymbol{W}}_{\text{FCN}}^{(k-1)}$ is identical to its joint-learning counterpart formulated in (6). That is, the GACL maintains the same *weight-invariant property* in the GCIL scenario as other ACL methods.

Algorithm 1 The pseudo-code of GACL.

Input: GCIL tasks $\mathcal{D}_1^{\text{train}}, \ldots, \mathcal{D}_K^{\text{train}}$ with $\mathcal{D}_k^{\text{train}} \sim \{\boldsymbol{X}_k^{\text{train}}, \boldsymbol{Y}_k^{\text{train}}\}$, the pre-trained backbone with frozen weight $\boldsymbol{\Theta}_{\text{backbone}}$

Initialization: $\boldsymbol{R}_0 \leftarrow \gamma \boldsymbol{I}$, $\boldsymbol{W}_{\text{FCN}}^{(0)} \leftarrow \boldsymbol{0}$

for task $k = 1$ **to** K **do**

$\boldsymbol{X}_k^{(\text{E})} \leftarrow f_{\text{backbone}}(\boldsymbol{X}_k^{\text{train}}, \boldsymbol{\Theta}_{\text{backbone}})$ (2)

$\boldsymbol{X}_k^{(\text{B})} \leftarrow f_{\text{buffer}}(\boldsymbol{X}_k^{(\text{E})})$ (3)

Decompose $\boldsymbol{Y}_k^{\text{train}}$ into exposed and unexposed class components $\bar{\boldsymbol{Y}}_k^{\text{train}}$ and $\tilde{\boldsymbol{Y}}_k^{\text{train}}$

$\boldsymbol{R}_k \leftarrow \boldsymbol{R}_{k-1} - \boldsymbol{R}_{k-1}\boldsymbol{X}_k^{(\text{B})\top}(\boldsymbol{I} + \boldsymbol{X}_k^{(\text{B})}\boldsymbol{R}_{k-1}\boldsymbol{X}_k^{(\text{B})\top})^{-1}\boldsymbol{X}_k\boldsymbol{R}_{k-1}$ (9)

$\boldsymbol{W}_{\text{unexposed}}^{(k)} \leftarrow \left[\boldsymbol{W}_{\text{FCN}}^{(k-1)} - \boldsymbol{R}_k\boldsymbol{X}_k^{(\text{B})\top}\boldsymbol{X}_k^{(\text{B})}\hat{\boldsymbol{W}}_{\text{FCN}}^{(k-1)} \quad \boldsymbol{R}_k\boldsymbol{X}_k^{(\text{B})\top}\tilde{\boldsymbol{Y}}_k^{\text{train}}\right]$ (11)

$\boldsymbol{W}_{\text{ECLG}}^{(k)} \leftarrow \left[\boldsymbol{R}_k\boldsymbol{X}_k^{(\text{B})\top}\bar{\boldsymbol{Y}}_k^{\text{train}} \quad \boldsymbol{0}\right]$ (12)

$\boldsymbol{W}_{\text{FCN}}^{(k)} \leftarrow \boldsymbol{W}_{\text{unexposed}}^{(k)} + \boldsymbol{W}_{\text{ECLG}}^{(k)}$

end for

The pseudo-code of the GACL is listed in Algorithm 1.

Exemplar-free. The recursive formulation is aided by $\boldsymbol{R}_k$ as indicated in (9). Note that this autocorrelation memory matrix records the inverse of inner products among the historical embedding matrices as shown in (7). Hence, the embeddings (e.g., $\boldsymbol{X}_k^{(\text{B})}$) are not reversible. Saving $\boldsymbol{R}_k$ instead of used samples is a safe alternative to preserve past knowledge. That is, our GACL is an *exemplar-free* technique without the need to keep any historical samples.

To more properly explain our GACL, as indicated in Figure 1 (c), the recursive solution in (8) can be rewritten as the sum of the unexposed-class contributed weight $\hat{\boldsymbol{W}}_{\text{unexposed}}^{(k)}$ and the ECLG weight $\hat{\boldsymbol{W}}_{\text{ECLG}}^{(k)}$, i.e.,

$$\hat{\boldsymbol{W}}_{\text{FCN}}^{(k)} = \hat{\boldsymbol{W}}_{\text{unexposed}}^{(k)} + \hat{\boldsymbol{W}}_{\text{ECLG}}^{(k)}, \tag{10}$$

where

$$\hat{\boldsymbol{W}}_{\text{unexposed}}^{(k)} = \left[\hat{\boldsymbol{W}}_{\text{FCN}}^{(k-1)} - \boldsymbol{R}_k\boldsymbol{X}_k^{(\text{B})\top}\boldsymbol{X}_k^{(\text{B})}\hat{\boldsymbol{W}}_{\text{FCN}}^{(k-1)} \quad \boldsymbol{R}_k\boldsymbol{X}_k^{(\text{B})\top}\tilde{\boldsymbol{Y}}_k^{\text{train}}\right], \tag{11}$$

$$\hat{\boldsymbol{W}}_{\text{ECLG}}^{(k)} = \left[\boldsymbol{R}_k\boldsymbol{X}_k^{(\text{B})\top}\bar{\boldsymbol{Y}}_k^{\text{train}} \quad \boldsymbol{0}\right]. \tag{12}$$

Unexposed-class Contributed Weight. The unexposed-class contributed weight $\hat{\boldsymbol{W}}_{\text{unexposed}}^{(k)}$ is recursively updated by the data of the unexposed class only. Note that the unexposed class label $\tilde{\boldsymbol{Y}}_k^{\text{train}}$ is applied on the concatenated weight along with new data $\boldsymbol{X}_k^{(\text{B})\top}$, which is reasonable as historical information should not intervene with the weight update of unseen classes. On the other hand, new data $\boldsymbol{X}_k^{(\text{B})\top}$ could also affect historical knowledge. This is marked by the gain of $-\boldsymbol{R}_k\boldsymbol{X}_k^{(\text{B})\top}\boldsymbol{X}_k^{(\text{B})}\hat{\boldsymbol{W}}_{\text{FCN}}^{(k-1)}$ to the original weight $\hat{\boldsymbol{W}}_{\text{FCN}}^{(k-1)}$ as indicated in (11).

Exposed-class Label Gain Weight. The ECLG module indicated in (12) captures knowledge from exposed-class labels. The supervision of this weight component marked by $\boldsymbol{R}_k\boldsymbol{X}_k^{(\text{B})\top}\bar{\boldsymbol{Y}}_k^{\text{train}}$ is mainly contributed by the exposed-class labels (i.e., $\bar{\boldsymbol{Y}}_k^{\text{train}}$). It is important to note that when $\bar{\boldsymbol{Y}}_k^{\text{train}}$ is empty (i.e., no classes reappear in task k), this component does not contribute to the update of $\hat{\boldsymbol{W}}_{\text{FCN}}^{(k)}$. This module is also isolated to distinguish GACL's difference from the existing ACL methods in a mathematical analysis manner (indicated as follows).

Difference from Existing ACL Methods. Overall, the GACL can be treated as a nontrivial generalization of ACIL [7], GKEAL [8], and various other ACL methods. For instance, in conventional CIL where no classes reappear in new tasks (i.e., $\forall k, \bar{\boldsymbol{Y}}_k^{\text{train}} \in \mathbb{R}^{*\times 0}$), the classifier of the GACL $\hat{\boldsymbol{W}}_{\text{FCN}}^{(k)} = \hat{\boldsymbol{W}}_{\text{unexposed}}^{(k)}$, which is equivalent to the recursive classifier of the ACIL. That is, the ACIL is a special case of our proposed GACL. The major difference lies in the ECLG module, corresponding to the exposed-class gain. This pattern makes sense as there must be compensation on top of ACIL updates (specifically designed for traditional CIL) when exposed data (out of setting) participate.

Table 1: Comparison of $\mathcal{A}_{\text{AUC}}$, $\mathcal{A}_{\text{Avg}}$, and $\mathcal{A}_{\text{Last}}$ among the GACL and other methods under the Si-Blurry setting. Data in **bold** represent the best EFCIL results, and data underlined are the best among all settings. We run all experiments 5 times and show "mean$_{\pm\text{standard error}}$".

Mem Size	Method	EFCIL	CIFAR-100 (%)			ImageNet-R (%)			Tiny-ImageNet (%)		
			$\mathcal{A}_{\text{AUC}}$	$\mathcal{A}_{\text{Avg}}$	$\mathcal{A}_{\text{Last}}$	$\mathcal{A}_{\text{AUC}}$	$\mathcal{A}_{\text{Avg}}$	$\mathcal{A}_{\text{Last}}$	$\mathcal{A}_{\text{AUC}}$	$\mathcal{A}_{\text{Avg}}$	$\mathcal{A}_{\text{Last}}$
2000	EWC++ [16]	×	$53.31_{\pm 1.70}$	$50.95_{\pm 1.50}$	$52.55_{\pm 0.71}$	$36.31_{\pm 0.72}$	$39.87_{\pm 1.35}$	$29.52_{\pm 0.43}$	$52.43_{\pm 0.52}$	$54.61_{\pm 1.54}$	$37.67_{\pm 0.77}$
	ER [35]	×	$56.17_{\pm 1.84}$	$53.80_{\pm 1.46}$	$55.60_{\pm 0.69}$	$39.31_{\pm 0.70}$	$43.03_{\pm 1.19}$	$32.09_{\pm 0.44}$	$55.69_{\pm 0.47}$	$57.87_{\pm 1.42}$	$41.10_{\pm 0.57}$
	RM [32]	×	$53.22_{\pm 1.82}$	$52.99_{\pm 1.69}$	$55.25_{\pm 0.61}$	$32.34_{\pm 1.88}$	$36.46_{\pm 2.23}$	$25.26_{\pm 1.08}$	$49.28_{\pm 0.43}$	$57.74_{\pm 1.57}$	$41.79_{\pm 0.34}$
	MVP-R [30]	×	$60.62_{\pm 1.03}$	$57.58_{\pm 0.56}$	$64.30_{\pm 0.29}$	$47.16_{\pm 1.00}$	$50.36_{\pm 0.90}$	$42.05_{\pm 0.15}$	$61.15_{\pm 0.86}$	$62.41_{\pm 0.50}$	$51.12_{\pm 0.67}$
500	EWC++ [16]	×	$48.31_{\pm 1.81}$	$44.56_{\pm 0.96}$	$40.52_{\pm 0.83}$	$32.81_{\pm 0.76}$	$35.54_{\pm 1.69}$	$23.43_{\pm 0.61}$	$45.30_{\pm 0.61}$	$46.34_{\pm 2.05}$	$27.05_{\pm 1.35}$
	ER [35]	×	$51.59_{\pm 1.94}$	$48.03_{\pm 0.80}$	$44.09_{\pm 0.80}$	$35.96_{\pm 0.72}$	$39.01_{\pm 1.54}$	$26.14_{\pm 0.44}$	$48.95_{\pm 0.58}$	$50.44_{\pm 1.71}$	$29.97_{\pm 0.75}$
	RM [32]	×	$41.07_{\pm 1.30}$	$38.10_{\pm 0.59}$	$32.66_{\pm 0.34}$	$22.45_{\pm 0.62}$	$22.08_{\pm 1.78}$	$9.61_{\pm 0.13}$	$36.66_{\pm 0.40}$	$38.83_{\pm 2.33}$	$18.23_{\pm 0.22}$
	MVP-R [30]	×	$56.20_{\pm 1.47}$	$53.61_{\pm 0.04}$	$55.35_{\pm 0.43}$	$43.28_{\pm 1.41}$	$45.74_{\pm 0.97}$	$35.60_{\pm 1.18}$	$55.28_{\pm 1.42}$	$55.45_{\pm 1.02}$	$40.12_{\pm 0.40}$
0	LwF [14]	✓	$40.71_{\pm 2.13}$	$38.49_{\pm 0.56}$	$27.03_{\pm 2.92}$	$29.41_{\pm 0.83}$	$31.95_{\pm 1.86}$	$19.67_{\pm 1.27}$	$39.88_{\pm 0.90}$	$41.35_{\pm 2.59}$	$24.93_{\pm 2.01}$
	L2P [36]	✓	$42.68_{\pm 2.70}$	$39.89_{\pm 0.45}$	$28.59_{\pm 3.34}$	$30.21_{\pm 0.91}$	$32.21_{\pm 1.73}$	$18.01_{\pm 3.07}$	$41.67_{\pm 1.17}$	$42.53_{\pm 2.52}$	$24.78_{\pm 2.31}$
	DualPrompt [33]	✓	$41.34_{\pm 2.59}$	$38.59_{\pm 0.68}$	$22.74_{\pm 3.40}$	$30.44_{\pm 0.88}$	$32.54_{\pm 1.84}$	$16.07_{\pm 3.20}$	$39.16_{\pm 1.13}$	$39.81_{\pm 3.03}$	$20.42_{\pm 3.37}$
	MVP [30]	✓	$45.07_{\pm 2.43}$	$44.93_{\pm 0.54}$	$39.94_{\pm 0.47}$	$35.77_{\pm 2.55}$	$35.58_{\pm 1.20}$	$22.06_{\pm 5.01}$	$46.43_{\pm 3.07}$	$45.41_{\pm 1.09}$	$28.21_{\pm 2.89}$
	SLDA [37]	✓	$53.00_{\pm 3.85}$	$50.09_{\pm 2.77}$	$61.79_{\pm 3.81}$	$33.11_{\pm 3.17}$	$33.78_{\pm 1.76}$	$39.02_{\pm 1.30}$	$49.17_{\pm 4.41}$	$47.93_{\pm 4.43}$	$53.13_{\pm 2.29}$
	GACL (ours)	✓	$\mathbf{57.99}_{\pm \mathbf{2.46}}$	$\mathbf{56.24}_{\pm \mathbf{3.12}}$	$\mathbf{70.31}_{\pm \mathbf{0.06}}$	$\mathbf{41.68}_{\pm \mathbf{0.78}}$	$\mathbf{47.30}_{\pm \mathbf{0.84}}$	$\mathbf{42.22}_{\pm \mathbf{0.10}}$	$\mathbf{63.14}_{\pm \mathbf{0.66}}$	$\mathbf{69.32}_{\pm \mathbf{0.87}}$	$\mathbf{62.68}_{\pm \mathbf{0.08}}$

4 Experiments

4.1 Experimental Setup

In the section, we conduct experiments on various benchmark datasets and compare the GACL with both EFCIL and replay-based state-of-the-art methods, including LwF [14], L2P [36], DualPrompt [33], ER [35], EWC++ [16], SLDA [37], RM [32], MVP [30], and MVP-R (MVP with exemplars).[3]

Datasets. We conduct experiments on three datasets: CIFAR-100 [38], ImageNet-R [39], and Tiny-ImageNet [40]. We evaluate each method under the Si-Blurry setting [30] (the most complex GCIL setting) with 5 independent seeds. For the Si-Blurry setting, we set the disjoint class ratio $\boldsymbol{r}_{\text{D}}$ to 50% and the blurry sample ratio $\boldsymbol{r}_{\text{B}}$ to 10%. More details about Si-Blurry are listed in Appendix C.

Implementation Details. We utilize the DeiT-S/16 [41] as our backbone. Following [42, 43], we pre-train the backbone on 611 ImageNet classes after excluding 389 classes that overlap with CIFAR and Tiny-ImageNet to prevent data leakage. To ensure a fair comparison, all methods utilize a frozen backbone. All methods under comparison are implemented as specified in [30]. The memory sizes of compared relay-based methods are set to 500 and 2000.

There are two hyperparameters in the GACL, the regularization term γ and the size of the buffer layer. Here, we adopt $\gamma = 100$, which is determined by the grid search of {0, 10, 100, 500, 1000, 10000} on CIFAR-100 (by a 90%-10% train-val split). As the regularization term γ is not sensitive in a proper range [7], we adopt this value for all datasets for convenience. We relocate its analysis to Appendix E. The size for the buffer layer $\boldsymbol{W}_{\text{B}}$ is set to 5000 for both the GACL and ACIL for convenience.

Evaluation Protocol. Three metrics are adopted to evaluate GCIL tasks. The real-time performance is evaluated by the *area under the curve of accuracy* $\mathcal{A}_{\text{AUC}}$ [31], i.e., $\mathcal{A}_{\text{AUC}} = \sum_{i=1}^{k} f(i \cdot \triangle n) \cdot \triangle n$, where $\triangle n$ is the number of samples observed between evaluation and $f(\cdot)$ is the curve in the accuracy-to-{number of training samples} plot, measuring anytime inference performance during training. A higher $\mathcal{A}_{\text{AUC}}$ corresponds to a method consistently maintaining high accuracy throughout the training. The overall performance is evaluated by the *average incremental accuracy* (or average accuracy) $\mathcal{A}_{\text{Avg}} = \frac{1}{K+1}\sum_{k=1}^{K}\mathcal{A}_k$, where the task-wise accuracy $\mathcal{A}_k$ indicates the average test accuracy in task

[3]The results for MVP and MVP-R are based on their official implementation, committed on October 26, 2024 (commit ID: ad8d1426a497545ba634521c00008c34ceece799).

k by testing the network on $\mathcal{D}_{1:k}^{\text{test}}$. A higher $\mathcal{A}_{\text{Avg}}$ score is preferred when evaluating algorithms. The last evaluation metric is the *last-task accuracy* $\mathcal{A}_{\text{Last}}$ evaluating the network's last-task performance after completing all tasks.

4.2 Comparison with State-of-the-arts

As shown in Figure 2, we comprehensively compare the GACL with both EFCIL and replay-based methods.

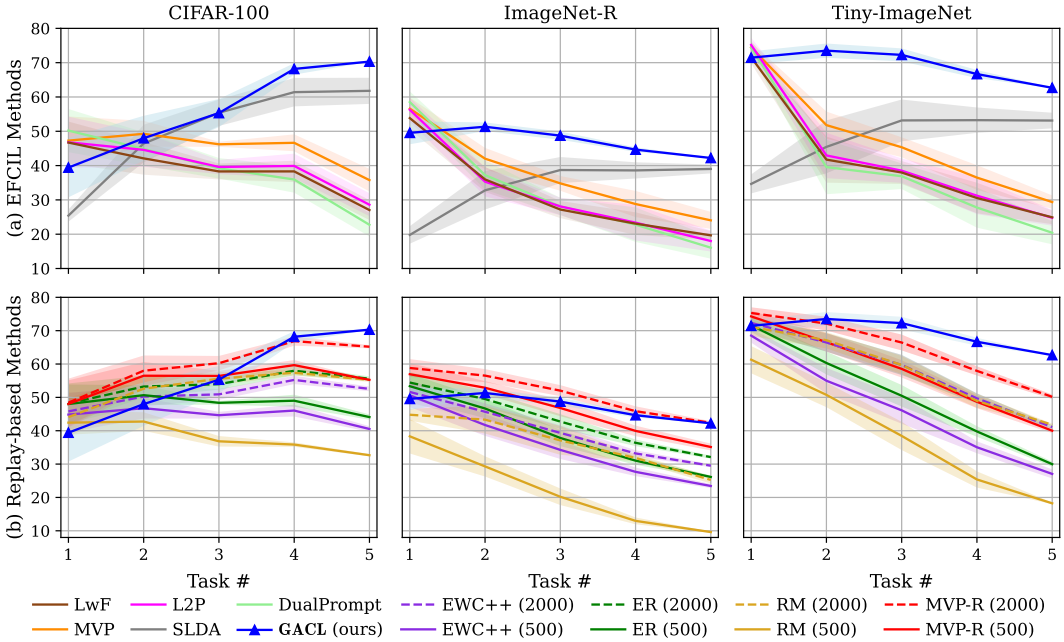

Figure 2: The task-wise accuracy $\mathcal{A}_k$ of the GACL with EFCIL methods (top) and replay-based methods (bottom) on benchmark datasets with the $K = 5$.

Compare with EFCIL Methods. EFCIL methods address privacy concerns and mitigate catastrophic forgetting without exemplars. Among EFCIL methods, our GACL consistently exhibits superior performance across all three datasets, as illustrated in the lower panel of Table 1.

For instance, on CIFAR-100, our method surpasses the second-best method SLDA, by **4.99**%, **6.15**%, and **8.52**% for $\mathcal{A}_{\text{AUC}}$, $\mathcal{A}_{\text{Avg}}$, and $\mathcal{A}_{\text{Last}}$, respectively. On Tiny-ImageNet, the GACL achieves impressive results with $\mathcal{A}_{\text{AUC}}$, $\mathcal{A}_{\text{Avg}}$, and $\mathcal{A}_{\text{Last}}$ reaching 63.14%, 69.32%, and 62.68%, respectively, surpassing the previous best EFCIL by **13.97**%, **21.39**%, and **9.55**%. Similar patterns are evident in the results of ImageNet-R, further confirming that the GACL is an exceptional tool for GCIL.

Owing to the weight-invariant property, the GACL exhibits more accurate and stable evolutions as k increases as observed in Figure 2 (a). All compared EFCIL methods exhibit sharp declines in accuracy, while the GACL delivers nearly non-declining curves. In particular, on CIFAR-100, the GACL shows an unnatural improvement of task-wise accuracy throughout the learning tasks, with the GACL initially lagging behind other EFCIL methods. This is because the Si-Blurry samples more than 70% of the CIFAR-100 categories in the first two tasks (see Appendix F), constructing a scenario where gradient-based algorithms could largely avoid the forgetting issue. Moreover, our method produces more stable predictions across diverse scenarios, as indicated by much smaller standard errors (colored shades in Figure 2 (a)). In summary, the experimental results demonstrate that our proposed GACL is exceedingly accurate and robust, exhibiting exceptional generalization ability.

Compare with Replay-based Methods. Replay-based methods are considerably competitive as they leverage historical samples. The memory size is a key adjustment, as increasing it typically leads to performance improvements by allowing more historical knowledge to be reviewed. For instance, the MVP-R achieves 4.42%, 3.97%, and 8.95% gains for $\mathcal{A}_{\text{AUC}}$, $\mathcal{A}_{\text{Avg}}$, and $\mathcal{A}_{\text{Last}}$ (see Table 1) on CIFAR-100 when increasing the memory size from 500 to 2000.

As an exemplar-free technique, our GACL avoids re-using the historical samples. However, as indicated in Table 1, the GACL still outperforms most existing replay-based results. For instance, the GACL achieves the best $\mathcal{A}_{\text{Last}}$ results among all settings. The GACL's $\mathcal{A}_{\text{AUC}}$ and $\mathcal{A}_{\text{Avg}}$ results are also mostly superior, except that our performance is slightly weaker than that of MVP-R with a memory size of 2000 on CIFAR-100 and ImageNet-R. Although increasing the number of exemplars can further improve the results of replay-based methods, this approach could lead to higher training and memory costs and, more importantly, more severe privacy invasion.

As indicated in Figure 2 (b), replay-based methods experience accuracy declines similar to those observed in the EFCIL case. This decline is due to an inherent limitation of gradient-based iterative algorithms, which tend to favor recently trained categories and thus lead to catastrophic forgetting. The GACL is iterative-free and then not constrained by this forgetting issue, thereby achieving nearly no performance reduction as K increases.

Why the GACL Gives Leading Performance. The above comparisons show that the proposed GACL is a powerful GCIL technique. Its competitive performance can be explained as follows. (i) Weight-invariant property. As shown in Theorem 3.1, the weight obtained recursively is equal to its joint-learning counterpart, indicating that the GACL is a "completely non-forgetting" technique (under the condition of a frozen backbone). (ii) Analytical solution. Existing GCIL techniques are gradient-based iterative algorithms prone to catastrophic forgetting by nature. The GACL is a new member of the ACL and inherits its non-iterative gradient-free essence with an analytical solution, thereby avoiding the task-recency bias to address forgetting.

4.3 Ablation Study on the ECLG Module

The ECLG module is a core component that allows the GACL to obtain the weight-invariant property in the GCIL scenario. Here, we conduct an ablation study to justify the ECLG's contributions under various blurry sample ratios r_B with $\boldsymbol{r}_{\text{D}} = 50\%$. Larger $\boldsymbol{r}_{\text{B}}$ indicates more complex data distributions in the Si-Blurry setting. As shown in Table 2, the GACL without ECLG exhibits poor performance with a visible gap for $\mathcal{A}_{\text{AUC}}$, $\mathcal{A}_{\text{Avg}}$, and $\mathcal{A}_{\text{Last}}$. For instance, on CIFAR-100 with r_B = 10%, the ECLG contributes a 23.01% $\mathcal{A}_{\text{Last}}$ gain to the GACL.

Table 2: Ablation study on the ECLG module of our GACL.

$\boldsymbol{r}_{\text{B}}$	Dataset	With ECLG			Without ECLG		
		$\mathcal{A}_{\text{AUC}}(\%)$	$\mathcal{A}_{\text{Avg}}(\%)$	$\mathcal{A}_{\text{Last}}(\%)$	$\mathcal{A}_{\text{AUC}}(\%)$	$\mathcal{A}_{\text{Avg}}(\%)$	$\mathcal{A}_{\text{Last}}(\%)$
10%	CIFAR-100	**57.99**$_{\pm 2.46}$	**56.24**$_{\pm 3.12}$	**70.31**$_{\pm 0.06}$	45.68$_{\pm 7.74}$	42.04$_{\pm 4.52}$	47.30$_{\pm 2.61}$
	ImageNet-R	**41.68**$_{\pm 0.78}$	**47.30**$_{\pm 0.84}$	**42.22**$_{\pm 0.10}$	40.29$_{\pm 2.23}$	46.95$_{\pm 1.15}$	41.67$_{\pm 0.36}$
	Tiny-ImageNet	**63.14**$_{\pm 0.66}$	**69.32**$_{\pm 0.87}$	**62.68**$_{\pm 0.08}$	60.21$_{\pm 1.86}$	65.80$_{\pm 1.20}$	60.13$_{\pm 0.37}$
30%	CIFAR-100	**57.33**$_{\pm 1.03}$	**58.74**$_{\pm 1.59}$	**69.90**$_{\pm 0.01}$	42.53$_{\pm 1.97}$	42.26$_{\pm 1.75}$	45.49$_{\pm 1.17}$
	ImageNet-R	**42.19**$_{\pm 0.44}$	**47.82**$_{\pm 1.11}$	**42.90**$_{\pm 0.08}$	42.01$_{\pm 0.26}$	46.95$_{\pm 1.15}$	41.67$_{\pm 0.56}$
	Tiny-ImageNet	**60.73**$_{\pm 1.15}$	**67.31**$_{\pm 1.14}$	**59.73**$_{\pm 2.55}$	60.63$_{\pm 1.86}$	57.03$_{\pm 1.98}$	60.13$_{\pm 0.55}$
50%	CIFAR-100	**56.74**$_{\pm 1.14}$	**58.29**$_{\pm 1.95}$	**70.02**$_{\pm 0.05}$	40.91$_{\pm 3.57}$	47.25$_{\pm 2.64}$	58.61$_{\pm 2.62}$
	ImageNet-R	**41.33**$_{\pm 1.46}$	**46.42**$_{\pm 2.30}$	**42.92**$_{\pm 0.17}$	40.44$_{\pm 3.14}$	42.50$_{\pm 3.43}$	39.05$_{\pm 1.65}$
	Tiny-ImageNet	**60.96**$_{\pm 1.83}$	**66.28**$_{\pm 2.69}$	**62.24**$_{\pm 0.10}$	60.32$_{\pm 4.20}$	60.70$_{\pm 4.30}$	56.97$_{\pm 1.89}$

As claimed in Theorem 3.1, the classifier without the ECLG module fails to absorb knowledge from joint classes in each task (i.e., classes that reappear), leading to substantial information loss under the GCIL setting. The GACL, equipped with the ECLG module, demonstrates competence in handling overlapping classes in realistic scenarios.

4.4 Robustness Analysis in Si-Blurry Setting

Here, we conduct a robust analysis by varying the disjoint class ratio $\boldsymbol{r}_{\text{D}}$ and the blurry sample ratio $\boldsymbol{r}_{\text{B}}$. The comparison happens among the GACL, the second-best EFCIL method SLDA, and the top-performing replay-based method MVP-R with a memory size of 500.

We evaluate our method under various r_{D}, including extreme cases where each task shares classes ($r_{\mathrm{D}} = 0\%$) and traditional CIL scenarios ($r_{\mathrm{D}} = 100\%$). Table 3 illustrates that our GACL consistently outperforms the compared methods (e.g., leads the SLDA by 2%-10%) and produces near-identical A_{Last} values with varying r_{D}. This shows the accurate and robust traits of the GACL.

We also evaluate our method using various r_{B} values, as shown in Table 4. Similar patterns observed here align with those in Table 3, further demonstrating the robustness of the proposed GACL, which delivers exceptional performance across different GCIL settings.

Table 3: The performance at different r_{D} with $r_{\mathrm{B}} = 10\%$ on CIFAR-100.

r_{D}	Method	$\mathcal{A}_{\mathrm{AUC}}(\%)$	$\mathcal{A}_{\mathrm{Avg}}(\%)$	$\mathcal{A}_{\mathrm{Last}}(\%)$
0%	SLDA [37]	**55.51**$_{\pm 1.93}$	**53.94**$_{\pm 0.92}$	67.45$_{\pm 0.26}$
	MVP-R [30]	53.49$_{\pm 1.40}$	50.73$_{\pm 0.37}$	60.54$_{\pm 2.03}$
	GACL (ours)	49.96$_{\pm 0.61}$	50.56$_{\pm 0.49}$	**69.94**$_{\pm 0.09}$
50%	SLDA [37]	53.00$_{\pm 3.85}$	50.09$_{\pm 2.77}$	61.79$_{\pm 3.81}$
	MVP-R [30]	56.20$_{\pm 1.47}$	53.61$_{\pm 0.04}$	55.35$_{\pm 0.43}$
	GACL (ours)	**57.99**$_{\pm 2.46}$	**56.24**$_{\pm 3.12}$	**70.31**$_{\pm 0.06}$
100%	SLDA [37]	65.46$_{\pm 4.79}$	67.29$_{\pm 5.28}$	63.56$_{\pm 2.68}$
	MVP-R [30]	68.43$_{\pm 0.28}$	68.04$_{\pm 1.48}$	53.14$_{\pm 0.72}$
	GACL (ours)	**70.72**$_{\pm 0.32}$	**77.57**$_{\pm 1.02}$	**69.97**$_{\pm 0.03}$

Table 4: The performance at different r_{B} with $r_{\mathrm{D}} = 50\%$ on CIFAR-100.

r_{B}	Method	$\mathcal{A}_{\mathrm{AUC}}(\%)$	$\mathcal{A}_{\mathrm{Avg}}(\%)$	$\mathcal{A}_{\mathrm{Last}}(\%)$
10%	SLDA [37]	53.00$_{\pm 3.85}$	50.09$_{\pm 2.77}$	61.79$_{\pm 3.81}$
	MVP-R [30]	56.20$_{\pm 1.47}$	53.61$_{\pm 0.04}$	55.35$_{\pm 0.43}$
	GACL (ours)	**57.99**$_{\pm 2.46}$	**56.24**$_{\pm 3.12}$	**70.31**$_{\pm 0.06}$
30%	SLDA [37]	54.55$_{\pm 4.66}$	54.06$_{\pm 2.41}$	63.04$_{\pm 2.56}$
	MVP-R [30]	59.65$_{\pm 2.04}$	58.31$_{\pm 1.52}$	58.16$_{\pm 1.38}$
	GACL (ours)	**57.33**$_{\pm 1.03}$	**58.74**$_{\pm 1.59}$	**69.90**$_{\pm 0.01}$
50%	SLDA [37]	53.81$_{\pm 3.43}$	52.93$_{\pm 2.36}$	63.45$_{\pm 2.72}$
	MVP-R [30]	59.10$_{\pm 1.98}$	57.34$_{\pm 1.96}$	54.81$_{\pm 0.21}$
	GACL (ours)	**56.74**$_{\pm 1.14}$	**58.29**$_{\pm 1.95}$	**70.02**$_{\pm 0.05}$

4.5 Limitation and Future Work

Overall, the GACL exhibits various good characteristics as an exemplar-free GCIL technique. The major limitation here is the need for a well-trained backbone because the GACL does not update backbone weights. This could motivate the exploration of adjustable backbones to continuously improve their feature extraction abilities, thereby further enhancing GACL's performance.

5 Conclusion

In this paper, we introduce the exemplar-free generalized analytic class incremental learning (GACL) approach to address the GCIL problem. Building upon analytic learning, the GACL delivers closed-form solutions to GCIL through the decomposition of GCIL data into exposed and unexposed classes. The GACL achieves the weight-invariant property that provides identical solutions for GCIL to its joint learning counterpart. We theoretically validate this property and provide high interpretability through the matrix analysis tool. Various experiments are conducted under the Si-Blurry setting, demonstrating that our proposed GACL achieves remarkable performance with high robustness compared to state-of-the-art EFCIL and replay-based methods.

Acknowledgments and Disclosure of Funding

This research was supported by the National Natural Science Foundation of China (62306117), the Guangzhou Basic and Applied Basic Research Foundation (2024A04J3681, 2023A04J1687), the South China University of Technology-TCL Technology Innovation Fund, the Fundamental Research Funds for the Central Universities (2023ZYGXZR023, 2024ZYGXZR074), the Guangdong Basic and Applied Basic Research Foundation (2024A1515010220), the CAAI-MindSpore Open Fund developed on Openl Community, the Shenzhen Fundamental Research Program (JCYJ20230807091809020), and Shenzhen Science and Technology Plan (Grant No. JCYJ20210324123802006).

References

[1] Sylvestre-Alvise Rebuffi, Alexander Kolesnikov, Georg Sperl, and Christoph H. Lampert. iCaRL: Incremental classifier and representation learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

[2] Michael McCloskey and Neal J. Cohen. Catastrophic interference in connectionist networks: The sequential learning problem. In Gordon H. Bower, editor, *Psychology of Learning and Motivation*, volume 24, pages 109–165. Academic Press, 1989.

[3] Ian J. Goodfellow, Mehdi Mirza, Da Xiao, Aaron Courville, and Yoshua Bengio. An empirical investigation of catastrophic forgetting in gradient-based neural networks, 2013.

[4] Saihui Hou, Xinyu Pan, Chen Change Loy, Zilei Wang, and Dahua Lin. Learning a unified classifier incrementally via rebalancing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[5] Rahaf Aljundi, Min Lin, Baptiste Goujaud, and Yoshua Bengio. Gradient based sample selection for online continual learning. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019.

[6] Fei Mi, Lingjing Kong, Tao Lin, Kaicheng Yu, and Boi Faltings. Generalized class incremental learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, June 2020.

[7] Huiping Zhuang, Zhenyu Weng, Hongxin Wei, RENCHUNZI XIE, Kar-Ann Toh, and Zhiping Lin. ACIL: Analytic class-incremental learning with absolute memorization and privacy protection. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh, editors, *Advances in Neural Information Processing Systems*, volume 35, pages 11602–11614. Curran Associates, Inc., 2022.

[8] Huiping Zhuang, Zhenyu Weng, Run He, Zhiping Lin, and Ziqian Zeng. GKEAL: Gaussian kernel embedded analytic learning for few-shot class incremental task. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7746–7755, June 2023.

[9] Arthur Douillard, Matthieu Cord, Charles Ollion, Thomas Robert, and Eduardo Valle. PODNet: Pooled outputs distillation for small-tasks incremental learning. In Andrea Vedaldi, Horst Bischof, Thomas Brox, and Jan-Michael Frahm, editors, *Computer Vision – ECCV 2020*, pages 86–102, Cham, 2020. Springer International Publishing.

[10] Yaoyao Liu, Bernt Schiele, and Qianru Sun. Adaptive aggregation networks for class-incremental learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2544–2553, June 2021.

[11] Fu-Yun Wang, Da-Wei Zhou, Han-Jia Ye, and De-Chuan Zhan. FOSTER: Feature boosting and compression for class-incremental learning. In Shai Avidan, Gabriel Brostow, Moustapha Cissé, Giovanni Maria Farinella, and Tal Hassner, editors, *Computer Vision – ECCV 2022*, pages 398–414, Cham, 2022. Springer Nature Switzerland.

[12] Yaoyao Liu, Yingying Li, Bernt Schiele, and Qianru Sun. Online hyperparameter optimization for class-incremental learning. *Proceedings of the AAAI Conference on Artificial Intelligence*, 37(7):8906–8913, Jun. 2023.

[13] Heechul Jung, Jeongwoo Ju, Minju Jung, and Junmo Kim. Less-forgetting learning in deep neural networks, 2016.

[14] Zhizhong Li and Derek Hoiem. Learning without forgetting. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(12):2935–2947, 2018.

[15] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network, 2015.

[16] James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A. Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, Demis Hassabis, Claudia Clopath, Dharshan Kumaran, and Raia Hadsell. Overcoming catastrophic forgetting in neural networks. *Proceedings of the National Academy of Sciences*, 114(13):3521–3526, 2017.

[17] Arslan Chaudhry, Puneet K. Dokania, Thalaiyasingam Ajanthan, and Philip H. S. Torr. Riemannian walk for incremental learning: Understanding forgetting and intransigence. In *Proceedings of the European Conference on Computer Vision (ECCV)*, September 2018.

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

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

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

[21] Grégoire Petit, Adrian Popescu, Hugo Schindler, David Picard, and Bertrand Delezoide. FeTrIL: Feature translation for exemplar-free class-incremental learning. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 3911–3920, January 2023.

[22] Ping Guo and Michael R. Lyu. A pseudoinverse learning algorithm for feedforward neural networks with stacked generalization applications to software reliability growth data. *Neurocomputing*, 56:101–121, 2004.

[23] Huiping Zhuang, Zhiping Lin, and Kar-Ann Toh. Blockwise recursive moore–penrose inverse for network learning. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 52(5):3237–3250, 2022.

[24] Huiping Zhuang, Zhiping Lin, Yimin Yang, and Kar-Ann Toh. Analytic learning of convolutional neural network for pattern recognition, 2022.

[25] Mark D. McDonnell, Dong Gong, Amin Parvaneh, Ehsan Abbasnejad, and Anton van den Hengel. RanPAC: Random projections and pre-trained models for continual learning. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine, editors, *Advances in Neural Information Processing Systems*, volume 36, pages 12022–12053. Curran Associates, Inc., 2023.

[26] Huiping Zhuang, Run He, Kai Tong, Ziqian Zeng, Cen Chen, and Zhiping Lin. 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, Mar. 2024.

[27] Run He, Huiping Zhuang, Di Fang, Yizhu Chen, Kai Tong, and Cen Chen. REAL: Representation enhanced analytic learning for exemplar-free class-incremental learning, 2024.

[28] Huiping Zhuang, Run He, Kai Tong, Di Fang, Han Sun, Haoran Li, Tianyi Chen, and Ziqian Zeng. Analytic federated learning, 2024.

[29] Zichen Liu, Chao Du, Wee Sun Lee, and Min Lin. Locality sensitive sparse encoding for learning world models online. In *The Twelfth International Conference on Learning Representations*, 2024.

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

[31] Hyunseo Koh, Dahyun Kim, Jung-Woo Ha, and Jonghyun Choi. Online continual learning on class incremental blurry task configuration with anytime inference. In *International Conference on Learning Representations*, 2022.

[32] Jihwan Bang, Heesu Kim, YoungJoon Yoo, Jung-Woo Ha, and Jonghyun Choi. Rainbow memory: Continual learning with a memory of diverse samples. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8218–8227, June 2021.

[33] Zifeng Wang, Zizhao Zhang, Sayna Ebrahimi, Ruoxi Sun, Han Zhang, Chen-Yu Lee, Xiaoqi Ren, Guolong Su, Vincent Perot, Jennifer Dy, and Tomas Pfister. DualPrompt: Complementary prompting for rehearsal-free continual learning. In Shai Avidan, Gabriel Brostow, Moustapha Cissé, Giovanni Maria Farinella, and Tal Hassner, editors, *Computer Vision – ECCV 2022*, pages 631–648, Cham, 2022. Springer Nature Switzerland.

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

[35] David Rolnick, Arun Ahuja, Jonathan Schwarz, Timothy Lillicrap, and Gregory Wayne. Experience replay for continual learning. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019.

[36] Zifeng Wang, Zizhao Zhang, Chen-Yu Lee, Han Zhang, Ruoxi Sun, Xiaoqi Ren, Guolong Su, Vincent Perot, Jennifer Dy, and Tomas Pfister. Learning to prompt for continual learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 139–149, June 2022.

[37] Tyler L. Hayes and Christopher Kanan. Lifelong machine learning with deep streaming linear discriminant analysis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, June 2020.

[38] Alex Krizhevsky, Vinod Nair, and Geoffrey Hinton. Learning multiple layers of features from tiny images. Technical report, Toronto, ON, Canada, 2009.

[39] Dan Hendrycks, Steven Basart, Norman Mu, Saurav Kadavath, Frank Wang, Evan Dorundo, Rahul Desai, Tyler Zhu, Samyak Parajuli, Mike Guo, Dawn Song, Jacob Steinhardt, and Justin Gilmer. 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)*, pages 8340–8349, October 2021.

[40] Ya Le and Xuan Yang. Tiny imagenet visual recognition challenge. *CS 231N*, 7(7):3, 2015.

[41] Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Herve Jegou. Training data-efficient image transformers & distillation through attention. In Marina Meila and Tong Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 10347–10357. PMLR, 18–24 Jul 2021.

[42] Gyuhak Kim, Changnan Xiao, Tatsuya Konishi, and Bing Liu. Learnability and algorithm for continual learning. In Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett, editors, *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 16877–16896. PMLR, 23–29 Jul 2023.

[43] Gyuhak Kim, Bing Liu, and Zixuan Ke. A multi-head model for continual learning via out-of-distribution replay. In Sarath Chandar, Razvan Pascanu, and Doina Precup, editors, *Proceedings of The 1st Conference on Lifelong Learning Agents*, volume 199 of *Proceedings of Machine Learning Research*, pages 548–563. PMLR, 22–24 Aug 2022.

[44] Huiping Zhuang, Zhiping Lin, and Kar-Ann Toh. Correlation projection for analytic learning of a classification network. *Neural Processing Letters*, 53(6):3893–3914, dec 2021.

A Proof of Theorem 3.1

Proof. in task $k-1$, we have

$$\hat{\boldsymbol{W}}_{\text{FCN}}^{(k-1)} = (\boldsymbol{X}_{1:k-2}^{\text{total}\top}\boldsymbol{X}_{1:k-2}^{\text{total}} + \boldsymbol{X}_{k-1}^{(\text{B})\top}\boldsymbol{X}_{1:k-1}^{(\text{B})} + \gamma\boldsymbol{I})^{-1}\left[\boldsymbol{X}_{1:k-2}^{\text{total}\top}\boldsymbol{Y}_{1:k-2}^{\text{total}} + \boldsymbol{X}_{k-1}^{(\text{B})\top}\bar{\boldsymbol{Y}}_{k-1}^{\text{train}} \quad \boldsymbol{X}_{k-1}^{(\text{B})\top}\tilde{\boldsymbol{Y}}_{k}^{\text{train}}\right]. \tag{13}$$

Hence, in task k, we have

$$\hat{\boldsymbol{W}}_{\text{FCN}}^{(k)} = (\boldsymbol{X}_{1:k-1}^{\text{total}\top}\boldsymbol{X}_{1:k-1}^{\text{total}} + \boldsymbol{X}_{k}^{(\text{B})\top}\boldsymbol{X}_{k}^{(\text{B})} + \gamma\boldsymbol{I})^{-1}\left[\boldsymbol{X}_{1:k-1}^{\text{total}\top}\boldsymbol{Y}_{1:k-1}^{\text{total}} + \boldsymbol{X}_{k}^{(\text{B})\top}\bar{\boldsymbol{Y}}_{k}^{\text{train}} \quad \boldsymbol{X}_{k}^{(\text{B})\top}\tilde{\boldsymbol{Y}}_{k}^{\text{train}}\right]. \tag{14}$$

We have defined the autocorrelation memory matrix $\boldsymbol{R}_{k-1}$ in the paper via

$$\boldsymbol{R}_{k-1} = (\boldsymbol{X}_{1:k-2}^{\text{total}\top}\boldsymbol{X}_{1:k-2}^{\text{total}} + \boldsymbol{X}_{k-1}^{(\text{B})\top}\boldsymbol{X}_{k-1}^{(\text{B})} + \gamma\boldsymbol{I})^{-1}. \tag{15}$$

To facilitate subsequent calculations, here we also define a cross-correlation matrix $\boldsymbol{Q}_{k-1}$, i.e.,

$$\boldsymbol{Q}_{k-1} = \left[\boldsymbol{X}_{1:k-2}^{\text{total}\top}\boldsymbol{Y}_{1:k-2}^{\text{total}} + \boldsymbol{X}_{k-1}^{(\text{B})\top}\bar{\boldsymbol{Y}}_{k-1}^{\text{train}} \quad \boldsymbol{X}_{k-1}^{(\text{B})\top}\tilde{\boldsymbol{Y}}_{k}^{\text{train}}\right]. \tag{16}$$

Thus we can rewrite (13) as

$$\hat{\boldsymbol{W}}_{\text{FCN}}^{(k-1)} = \boldsymbol{R}_{k-1}\boldsymbol{Q}_{k-1}. \tag{17}$$

Therefore, in task k we have

$$\hat{\boldsymbol{W}}_{\text{FCN}}^{(k)} = \boldsymbol{R}_{k}\boldsymbol{Q}_{k}. \tag{18}$$

From (15), we can recursively calculate $\boldsymbol{R}_k$ from $\boldsymbol{R}_{k-1}$, i.e.,

$$\boldsymbol{R}_{k} = \left(\boldsymbol{R}_{k-1}^{-1} + \boldsymbol{X}_{k}^{(\text{B})\top}\boldsymbol{X}_{k}^{(\text{B})}\right)^{-1}. \tag{19}$$

According to the Woodbury matrix identity, we have

$$(\boldsymbol{A}+\boldsymbol{U}\boldsymbol{C}\boldsymbol{V})^{-1} = \boldsymbol{A}^{-1} - \boldsymbol{A}^{-1}\boldsymbol{U}(\boldsymbol{C}^{-1}+\boldsymbol{V}\boldsymbol{A}^{-1}\boldsymbol{U})^{-1}\boldsymbol{V}\boldsymbol{A}^{-1}.$$

Let $\boldsymbol{A} = \boldsymbol{R}_{k-1}^{-1}$, $\boldsymbol{U} = \boldsymbol{X}_{k}^{(\text{B})\top}$, $\boldsymbol{C} = \boldsymbol{I}$, and $\boldsymbol{V} = \boldsymbol{X}_{k}^{(\text{B})}$ in (19), we have

$$\boldsymbol{R}_{k} = \boldsymbol{R}_{k-1} - \boldsymbol{R}_{k-1}\boldsymbol{X}_{k}^{(\text{B})\top}(\boldsymbol{I}+\boldsymbol{X}_{k}^{(\text{B})}\boldsymbol{R}_{k-1}\boldsymbol{X}_{k}^{(\text{B})\top})^{-1}\boldsymbol{X}_{k}^{(\text{B})}\boldsymbol{R}_{k-1}. \tag{20}$$

Hence, $\boldsymbol{R}_k$ can be recursively updated using its last-task counterpart $\boldsymbol{R}_{k-1}$ and data from the current task (i.e., $\boldsymbol{X}_{k}^{(\text{B})}$). This proves the recursive calculation of the autocorrelation memory matrix.

Next, we derive the recursive formulation of $\hat{\boldsymbol{W}}_{\text{FCN}}^{(k)}$. To this end, we also recurse the cross-correlation matrix $\boldsymbol{Q}_k$ in task k, i.e.,

$$\boldsymbol{Q}_{k} = \left[\boldsymbol{X}_{1:k-1}^{\text{total}\top}\boldsymbol{Y}_{1:k-1}^{\text{total}} + \boldsymbol{X}_{k}^{(\text{B})\top}\bar{\boldsymbol{Y}}_{k}^{\text{train}} \quad \boldsymbol{X}_{k}^{(\text{B})\top}\tilde{\boldsymbol{Y}}_{k}^{\text{train}}\right] = \boldsymbol{Q}'_{k-1} + \left[\boldsymbol{X}_{k}^{(\text{B})\top}\bar{\boldsymbol{Y}}_{k}^{\text{train}} \quad \boldsymbol{X}_{k}^{(\text{B})\top}\tilde{\boldsymbol{Y}}_{k}^{\text{train}}\right], \tag{21}$$

where

$$\boldsymbol{Q}'_{k-1} = \begin{cases} \begin{bmatrix}\boldsymbol{Q}_{k-1} & \boldsymbol{0}_{d_{(\text{B})}\times(d_{y_k}-d_{y_{k-1}})}\end{bmatrix}, & d_{y_k} > d_{y_{k-1}} \\ \boldsymbol{Q}_{k-1}, & d_{y_k} = d_{y_{k-1}} \end{cases}. \tag{22}$$

Note that the concatenation in (22) is due to the assumption that $\boldsymbol{Y}_{1:k}^{\text{train}}$ in task k contains more data classes (hence more columns) than $\boldsymbol{Y}_{1:k-1}^{\text{train}}$. It is possible that there are no new classes appear in task k, then $\tilde{\boldsymbol{Y}}_{k}^{\text{train}}$ should be $\boldsymbol{0}$.

Similar to what (22) does,

$$\hat{\boldsymbol{W}}_{\text{FCN}}^{(k-1)\prime} = \begin{cases} \left[\hat{\boldsymbol{W}}_{\text{FCN}}^{(k-1)} \quad \boldsymbol{0}_{d_{(\text{B})} \times (d_{y_k} - d_{y_{k-1}})}\right], & d_{y_k} > d_{y_{k-1}} \\ \hat{\boldsymbol{W}}_{\text{FCN}}^{(k-1)}, & d_{y_k} = d_{y_{k-1}} \end{cases} \tag{23}$$

We have

$$\hat{\boldsymbol{W}}_{\text{FCN}}^{(k-1)\prime} = \boldsymbol{R}_{k-1}\boldsymbol{Q}'_{k-1}. \tag{24}$$

Hence, $\hat{\boldsymbol{W}}_{\text{FCN}}^{(k)}$ can be rewritten as

$$\begin{aligned} \hat{\boldsymbol{W}}_{\text{FCN}}^{(k)} &= \boldsymbol{R}_k\boldsymbol{Q}_k \\ &= \boldsymbol{R}_k(\boldsymbol{Q}'_{k-1} + \left[\boldsymbol{X}_k^{(\text{B})\top}\bar{\boldsymbol{Y}}_k^{\text{train}} \quad \boldsymbol{X}_k^{(\text{B})\top}\tilde{\boldsymbol{Y}}_k^{\text{train}}\right]) \\ &= \boldsymbol{R}_k\boldsymbol{Q}'_{k-1} + \boldsymbol{R}_k\boldsymbol{X}_k^{(\text{B})\top}\left[\bar{\boldsymbol{Y}}_k^{\text{train}} \quad \tilde{\boldsymbol{Y}}_k^{\text{train}}\right]. \end{aligned} \tag{25}$$

By substituting (20) into $\boldsymbol{R}_k\boldsymbol{Q}'_{k-1}$, we have

$$\begin{aligned} \boldsymbol{R}_k\boldsymbol{Q}'_{k-1} &= \boldsymbol{R}_{k-1}\boldsymbol{Q}'_{k-1} - \boldsymbol{R}_{k-1}\boldsymbol{X}_k^{(\text{B})\top}(\boldsymbol{I} + \boldsymbol{X}_k^{(\text{B})}\boldsymbol{R}_{k-1}\boldsymbol{X}_k^{(\text{B})\top})^{-1}\boldsymbol{X}_k^{(\text{B})}\boldsymbol{R}_{k-1}\boldsymbol{Q}'_{k-1} \\ &= \hat{\boldsymbol{W}}_{\text{FCN}}^{(k-1)\prime} - \boldsymbol{R}_{k-1}\boldsymbol{X}_k^{(\text{B})\top}(\boldsymbol{I} + \boldsymbol{X}_k^{(\text{B})}\boldsymbol{R}_{k-1}\boldsymbol{X}_k^{(\text{B})\top})^{-1}\boldsymbol{X}_k^{(\text{B})}\hat{\boldsymbol{W}}_{\text{FCN}}^{(k-1)\prime}. \end{aligned} \tag{26}$$

To simplify this equation, let $\boldsymbol{K}_k = (\boldsymbol{I} + \boldsymbol{X}_k^{(\text{B})}\boldsymbol{R}_{k-1}\boldsymbol{X}_k^{(\text{B})\top})^{-1}$. Since

$$\boldsymbol{I} = \boldsymbol{K}_k\boldsymbol{K}_k^{-1} = \boldsymbol{K}_k(\boldsymbol{I} + \boldsymbol{X}_k^{(\text{B})}\boldsymbol{R}_{k-1}\boldsymbol{X}_k^{(\text{B})\top}),$$

we have $\boldsymbol{K}_k = \boldsymbol{I} - \boldsymbol{K}_k\boldsymbol{X}_k^{(\text{B})}\boldsymbol{R}_{k-1}\boldsymbol{X}_k^{(\text{B})\top}$. Therefore,

$$\begin{aligned} &\boldsymbol{R}_{k-1}\boldsymbol{X}_k^{(\text{B})\top}(\boldsymbol{I} + \boldsymbol{X}_k^{(\text{B})}\boldsymbol{R}_{k-1}\boldsymbol{X}_k^{(\text{B})\top})^{-1} \\ &= \boldsymbol{R}_{k-1}\boldsymbol{X}_k^{(\text{B})\top}\boldsymbol{K}_k \\ &= \boldsymbol{R}_{k-1}\boldsymbol{X}_k^{(\text{B})\top}(\boldsymbol{I} - \boldsymbol{K}_k\boldsymbol{X}_k^{(\text{B})}\boldsymbol{R}_{k-1}\boldsymbol{X}_k^{(\text{B})\top}) \\ &= (\boldsymbol{R}_{k-1} - \boldsymbol{R}_{k-1}\boldsymbol{X}_k^{(\text{B})\top}\boldsymbol{K}_k\boldsymbol{X}_k^{(\text{B})}\boldsymbol{R}_{k-1})\boldsymbol{X}_k^{(\text{B})\top} \\ &= \boldsymbol{R}_k\boldsymbol{X}_k^{(\text{B})\top}. \end{aligned} \tag{27}$$

Substituting (27) into (26), $\boldsymbol{R}_k\boldsymbol{Q}'_{k-1}$ can be written as

$$\boldsymbol{R}_k\boldsymbol{Q}'_{k-1} = \hat{\boldsymbol{W}}_{\text{FCN}}^{(k-1)\prime} - \boldsymbol{R}_k\boldsymbol{X}_k^{(\text{B})\top}\boldsymbol{X}_k^{(\text{B})}\hat{\boldsymbol{W}}_{\text{FCN}}^{(k-1)\prime}. \tag{28}$$

Substituting (28) into (25) implies that

$$\begin{aligned} \hat{\boldsymbol{W}}_{\text{FCN}}^{(k)} &= \hat{\boldsymbol{W}}_{\text{FCN}}^{(k-1)\prime} - \boldsymbol{R}_k\boldsymbol{X}_k^{(\text{B})\top}\boldsymbol{X}_k^{(\text{B})}\hat{\boldsymbol{W}}_{\text{FCN}}^{(k-1)\prime} + \boldsymbol{R}_k\boldsymbol{X}_k^{(\text{B})\top}\left[\bar{\boldsymbol{Y}}_k^{\text{train}} \quad \tilde{\boldsymbol{Y}}_k^{\text{train}}\right] \\ &= \left[\hat{\boldsymbol{W}}_{\text{FCN}}^{(k-1)} - \boldsymbol{R}_k\boldsymbol{X}_k^{(\text{B})\top}\boldsymbol{X}_k^{(\text{B})}\hat{\boldsymbol{W}}_{\text{FCN}}^{(k-1)} + \boldsymbol{R}_k\boldsymbol{X}_k^{(\text{B})\top}\bar{\boldsymbol{Y}}_k^{\text{train}} \quad \boldsymbol{R}_k\boldsymbol{X}_k^{(\text{B})\top}\tilde{\boldsymbol{Y}}_k^{\text{train}}\right]. \end{aligned} \tag{29}$$

which completes the proof.

□

B GCIL Properties

The GCIL scenario [6] is a recent CIL focus. Given task-wise learning tasks, we can involve all class labels in a set $\mathcal{S}$ with the number of classes N. The sample sizes, such as the numbers of input images of different classes appearing in task k, are modeled as a random vector $\boldsymbol{c}_k \in \mathbb{R}^N$. Each entry $\boldsymbol{c}_{k,i}$ is a random variable denoting the sample size of class i in task k. In the generalized form, $\boldsymbol{c}_k$ is sampled from a task-dependent distribution. The GCIL scenario can be summarized as the following three key properties.

Property B.1. *The number of classes in a task is not fixed. Suppose m_k is the number of classes in task k, we have:*

$$M_k = |\{i \in \mathcal{S} : \boldsymbol{c}_{k,i} > 0\}| \sim \mathcal{M}_k, \tag{30}$$

where $\mathcal{M}_k$ is a task-dependent distribution.

Property B.2. *Classes appearing in different tasks could overlap. For two tasks k and k', $k \neq k'$, we have:*

$$P(\boldsymbol{c}_k \odot \boldsymbol{c}_{k'} \neq 0) > 0, \tag{31}$$

where $\odot$ denotes element-wise multiplication of two vectors and $P(\cdot)$ is the probability.

Property B.3. *Sample sizes of different classes at the same task could be different. That is, for task k, we have*

$$i, j \in \mathcal{S}, i \neq j, P(\boldsymbol{c}_{k,i} \neq \boldsymbol{c}_{k,j} \mid \boldsymbol{c}_{k,i} \neq 0, \boldsymbol{c}_{k,j} \neq 0) > 0. \tag{32}$$

In short, the number of classes and samples could vary throughout the continual learning.

C Si-Blurry Setting

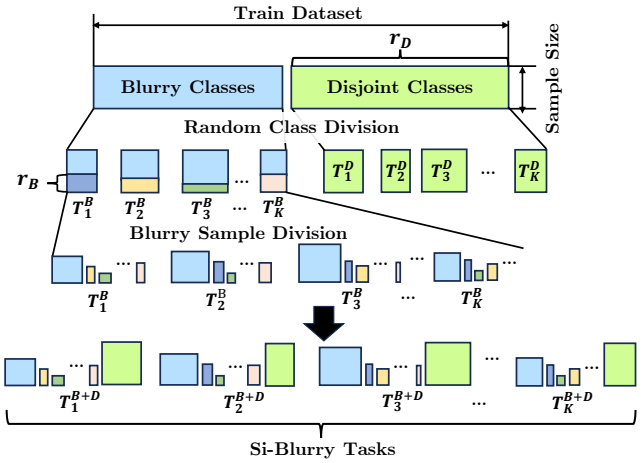

Figure 3: A configuration example of Si-Blurry setting.

The Si-Blurry setting [30] satisfies all the three properties of GCIL mentioned in Appendix B and can be treated as its good realization. As shown in Figure 3, for a K-task learning, the Si-Blurry first randomly partitions all classes 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* $\boldsymbol{r}_{\mathrm{D}}$, which is defined as the ratio of the number of disjoint classes to the number of all classes. Then disjoint classes and blurry classes are randomly assigned to disjoint tasks (T^{D}) and blurry tasks (T^{B}) respectively. Next, 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* $\boldsymbol{r}_{\mathrm{B}}$, which is defined as the ratio of the extracted sample within samples in all blurry tasks. Finally, each Si-Blurry task $T^{\mathrm{B+D}}$ with a stochastic blurry task boundary consists of a disjoint and blurry task. We adopt Si-Blurry with different combinations of $\boldsymbol{r}_{\mathrm{D}}$ and $\boldsymbol{r}_{\mathrm{B}}$ for reliable empirical validations.

D Compute Resources

GPU Usage. We conduct experiments in PyTorch on one Nvidia Geforce RTX 4090 GPU with a batch size of 64 for training and 128 for inference. Figure 4 shows that the GACL uses minimal GPU memory. Our GACL significantly reduces GPU memory usage since it requires no back-propagation, thereby detaching gradients from tensors during calculations. This characteristic allows our approach to be applied with a larger batch size without memory leaks.

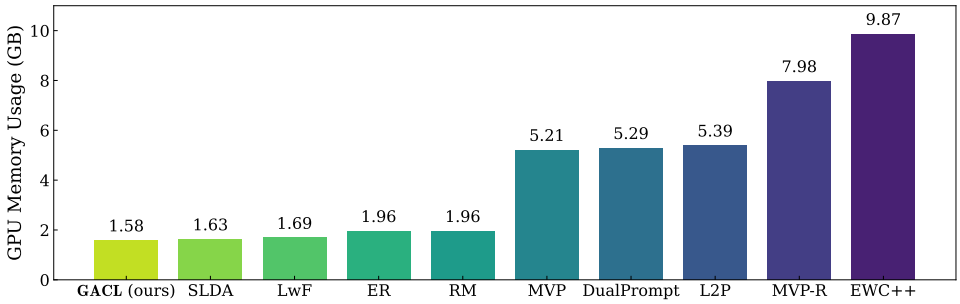

Figure 4: GPU memory consumption in GB with a batch size of 64 where replay-based methods are with 2000 memory size.

Training Time. Table 5 further illustrates the GACL's training time compared to others on one Nvidia Geforce RTX 4090 GPU, highlighting its efficiency. The GACL is faster than any other baselines except SLDA on three datasets because only the classifier and autocorrelation memory matrix $\boldsymbol{R}$ are updated, leading to small numbers of trainable parameters compared to those baselines in a back-propagation manner.

Table 5: Average Training time of 5 independent seeds in seconds (s) where replay-based methods are with 2000 memory size.

Method	EFCIL	CIFAR-100 (s)	ImageNet-R (s)	Tiny-ImageNet (s)
RM [32]	✗	>2 days	>2 days	>2 days
MVP-R [30]	✗	717	527	1597
ER [35]	✗	369	330	715
EWC++ [16]	✗	650	391	1356
LwF [14]	✓	334	229	862
L2P [36]	✓	651	285	1246
DualPrompt [33]	✓	656	332	1294
MVP [30]	✓	628	300	1345
SLDA [37]	✓	401	284	915
GACL (ours)	✓	**611**	**321**	**1246**

E Hyperparameter Analysis for Regularization Term

Table 6: $\mathcal{A}_{\text{AUC}}$, $\mathcal{A}_{\text{Avg}}$, and $\mathcal{A}_{\text{Last}}$ of the GACL on all benchmark datasets with various values of the regularization term γ.

γ	CIFAR-100 (%)			ImageNet-R (%)			Tiny-ImageNet (%)		
	$\mathcal{A}_{\text{AUC}}$	$\mathcal{A}_{\text{Avg}}$	$\mathcal{A}_{\text{Last}}$	$\mathcal{A}_{\text{AUC}}$	$\mathcal{A}_{\text{Avg}}$	$\mathcal{A}_{\text{Last}}$	$\mathcal{A}_{\text{AUC}}$	$\mathcal{A}_{\text{Avg}}$	$\mathcal{A}_{\text{Last}}$
0	$8.87_{\pm 4.96}$	$9.83_{\pm 5.82}$	$8.65_{\pm 6.47}$	$2.03_{\pm 0.36}$	$2.85_{\pm 0.86}$	$0.71_{\pm 0.09}$	$4.38_{\pm 2.17}$	$6.14_{\pm 4.01}$	$0.62_{\pm 0.11}$
10	$57.57_{\pm 2.35}$	$55.97_{\pm 3.22}$	$\mathbf{70.45_{\pm 0.08}}$	$38.65_{\pm 0.69}$	$44.38_{\pm 0.83}$	$41.96_{\pm 0.10}$	$62.74_{\pm 0.64}$	$69.24_{\pm 0.79}$	$62.73_{\pm 0.09}$
100	$\mathbf{57.99_{\pm 2.46}}$	$\mathbf{56.24_{\pm 3.12}}$	$70.31_{\pm 0.06}$	$41.68_{\pm 0.78}$	$47.30_{\pm 0.84}$	$42.22_{\pm 0.10}$	$\mathbf{63.14_{\pm 0.66}}$	$\mathbf{69.32_{\pm 0.87}}$	$\mathbf{62.68_{\pm 0.08}}$
500	$56.98_{\pm 2.61}$	$55.46_{\pm 3.23}$	$70.00_{\pm 0.02}$	$\mathbf{42.92_{\pm 0.79}}$	$\mathbf{49.01_{\pm 0.85}}$	$\mathbf{42.70_{\pm 0.14}}$	$62.90_{\pm 0.67}$	$68.95_{\pm 0.88}$	$62.41_{\pm 0.09}$
1000	$56.03_{\pm 2.70}$	$54.76_{\pm 3.31}$	$69.61_{\pm 0.08}$	$42.69_{\pm 0.80}$	$48.90_{\pm 0.90}$	$42.67_{\pm 0.16}$	$61.96_{\pm 0.67}$	$68.48_{\pm 0.83}$	$62.10_{\pm 0.07}$
10000	$51.01_{\pm 3.04}$	$50.92_{\pm 3.62}$	$66.38_{\pm 0.07}$	$38.55_{\pm 0.85}$	$45.16_{\pm 0.84}$	$40.10_{\pm 0.19}$	$57.54_{\pm 0.74}$	$65.21_{\pm 0.70}$	$59.55_{\pm 0.07}$

The regularization term γ plays a crucial role and demonstrates robust behavior throughout our experiments. We assess the impact of the regularization term γ in Table 6 and visualize the real-time accuracy of the GACL as it learns from training samples in Figure 5. Table 6 reveals the GACL's consistent performance across a broad range of γ values, spanning from 10 to 10000. This highlights the versatility and robustness of our proposed GACL. However, as indicated in Figure 5, γ of 10000 leads to slightly poorer performance because the ACL is prone to underfitting due to simple linear regression [44].

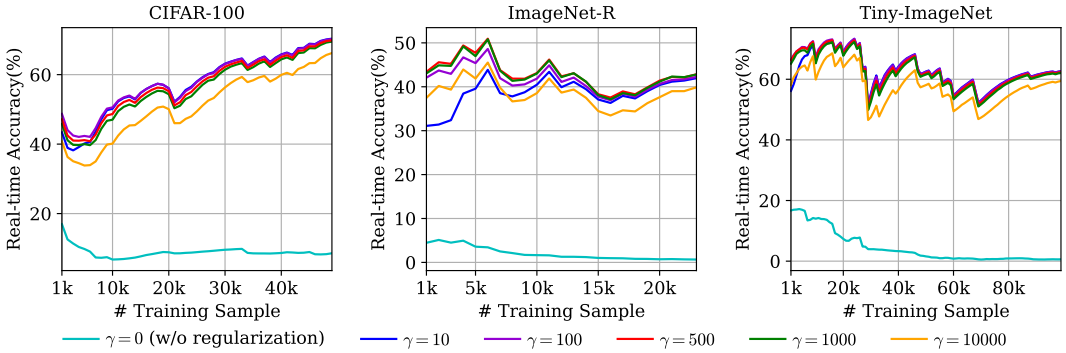

Figure 5: Real-time accuracy of the GACL on all benchmark datasets with various values of the regularization term γ.

Notably, both Table 6 and Figure 5 demonstrate that the absence of regularization results in a significant decline in performance. This underscores the crucial importance of incorporating γ in the model. As indicated in (7), if we eliminate regularization by setting γ to 0, the initial autocorrelation memory matrix $\boldsymbol{R}_0$ becomes zero. Subsequently, the computation of the autocorrelation memory matrix in task 1, denoted as $\boldsymbol{R}_1$, is expressed as:

$$\boldsymbol{R}_1 = (\boldsymbol{X}_1^{\text{total}\top}\boldsymbol{X}_1^{\text{total}})^{-1} = (\boldsymbol{X}_1^{(\text{B})\top}\boldsymbol{X}_1^{(\text{B})})^{-1}.$$

However, it's crucial to emphasize that $\boldsymbol{X}_1^{(\text{B})\top}\boldsymbol{X}_1^{(\text{B})}$ might result in a singular matrix, rendering it non-invertible. This potential singularity introduces an error in calculating $\boldsymbol{R}_1$, leading to a decrease in accuracy.

F Analysis of task-wise Accuracy Trends of the GACL

As depicted in Figure 2 (a), the task-wise accuracy of the GACL on CIFAR-100 demonstrates an increase. Notably, in the initial two tasks, the accuracy is lower compared to other EFCIL methods. However, on the other datasets, the GACL remains relatively stable. Upon a more detailed examination of the dataset split, we infer that the observed variations in trends are attributed to the specific dataset settings.

For a dataset with N classes, the class number ratio r_c after training on i-th samples is defined as $r_c = d_i/N$, where d_i is the number of classes that have been seen observed at that point. As Figure 6 indicates, by examining the real-time accuracy and the class number ratio r_c across the three sets of figures, a notable observation is made: when the sample size is small, the class number ratio r_c on CIFAR-100 always surpasses that of the other two datasets on 5 seeds. This suggests that tasks on CIFAR-100 are notably more complex and intricate, resembling a few-shot learning scenario.

Consequently, the GACL exhibits lower task-wise accuracy compared to other gradient-based EFCIL methods, particularly in the initial stages. However, as more training samples are acquired, its accuracy progressively improves.

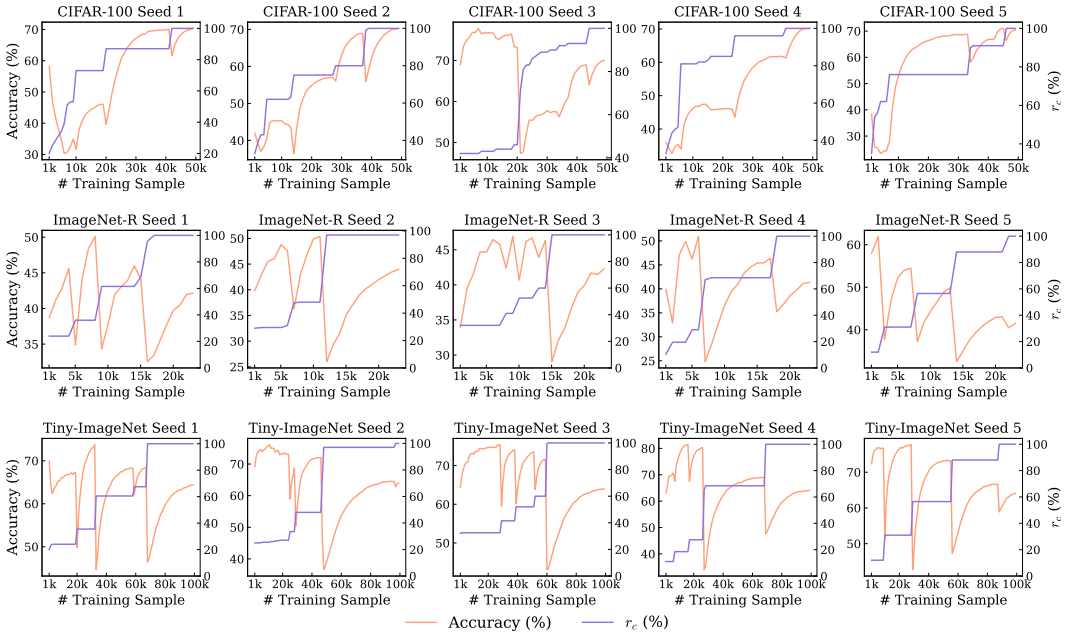

Figure 6: Real-time accuracy and class number ratio r_c on 5 independent random seeds.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: All the claims are clearly clarified, including the contributions made in the paper and important assumptions and limitations.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: In Section 4.5, a discussion of the limitations and the future work of the GACL is conducted.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.

- The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
- The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
- The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
- If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
- While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory Assumptions and Proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

   Justification: The proof of the Theorem 3.1 is listed in Appendix A.

   Guidelines:

   - The answer NA means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.
   - The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
   - Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
   - Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental Result Reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: Our paper comprehensively outlines both the experimental implementation and algorithmic details, ensuring transparency in our method.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example

(a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
(b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
(c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
(d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: All datasets are publicly accessible, and we have provided the source code.

   Guidelines:

   - The answer NA means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental Setting/Details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: We have reported all the necessary details of our experiment.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment Statistical Significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: Results in this paper are reported by the average of 5 different seeds with standard error.

   Guidelines:

   - The answer NA means that the paper does not include experiments.

- The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments Compute Resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: The information on the computer resources for our GACL is listed in Appendix D.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code Of Ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: The paper fully complies with the NeurIPS Code of Ethics.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader Impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [NA]

    Justification: There is no societal impact of the work performed.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.

- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: The models and benchmark datasets mentioned in the paper are all openly accessible with no personally identifiable information or offensive content.

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: We have verified that this paper cites all the datasets and models we used.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
    - If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New Assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

Answer: [Yes]

Justification: We have made our source code publicly available at `https://github.com/CHEN-YIZHU/GACL`.

Guidelines:

- The answer NA means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and Research with Human Subjects**

Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

Answer: [NA]

Justification: The paper does not involve crowdsourcing nor research with human subjects.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional Review Board (IRB) Approvals or Equivalent for Research with Human Subjects**

Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

Answer: [NA]

Justification: The paper does not involve crowdsourcing nor research with human subjects.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.