Enhanced Experience Replay for Class Incremental Continual Learning

Jiafu Hao[1], Son Lam Phung[1], Yang Di[1], Hoang Thanh Le[1], Abdesselam Bouzerdoum[1,2]
[1]University of Wollongong, Wollongong, Australia
[2]Hamad Bin Khalifa University, Doha, Qatar

***Abstract*—Continual learning aims to construct a machine model that learns multiple tasks sequentially. However, it may face the significant challenge of catastrophic forgetting, where the model minimizes the loss on the current data and forgets the previously learned tasks. A common approach to mitigate catastrophic forgetting is replay-based continual learning, which stores data points from previous tasks in a buffer and revisits them periodically. However, not all data points in a task contribute the same significance for learning. Hence, coreset selection is crucial for continual learning with imbalanced and noisy datasets. In this paper, we introduce Enhanced Experience Replay (EER), a simple yet effective method that selects the most representative and informative coreset for training at each iteration. EER not only optimizes the model's adaptability to the current dataset but also prioritizes samples exhibiting a high affinity with previous tasks. Evaluated on CIFAR-10 and CIFAR-100 datasets, our coreset selection mechanism significantly enhances task adaptability and prevent catastrophic forgetting. The proposed method achieves state-of-the-art performance on the two benchmark datasets.**

***Index Terms*—Continual learning, coreset selection, replay-based learning, gradient-based training**

I. Introduction

Humans can acquire new knowledge while retaining mastery over previously learned skills. Achieving human-like capability of lifelong learning is a major goal in machine learning. Toward this goal, Continual Learning (CL) [1] represents the ability to sequentially acquire new tasks and use newly gained knowledge to augment the existing competencies. However, the CL models may suffer from catastrophic forgetting. The model's performance on previously-learned tasks degrades significantly when it performs new learning tasks [2].

Recently, several CL algorithms have been proposed to enhance task diversity and mitigate catastrophic forgetting [3]. Based on the learning strategies, these algorithms can be divided into three categories: regularization methods [5], [9], parameter isolation methods [6], [7], and memory-based experience replay (ER) methods [8], [10]. Among them, the ER algorithms have shown the state-of-the-art results. They employ a compact buffer to retain a selected subset of training instances from prior tasks, which can be reused in the new tasks. These methods demonstrate significant performance enhancements when memory resources are sufficient [11].

In the ER algorithms, employing random sampling is a cost-effective approach for buffer storage [11]. However, low-quality, noisy, or imbalanced data selected by random sampling may severely reduce the model performance. Therefore, it is crucial to mitigate the interference from ineffective instances on model parameters for the previous learning tasks, see Figure 1.

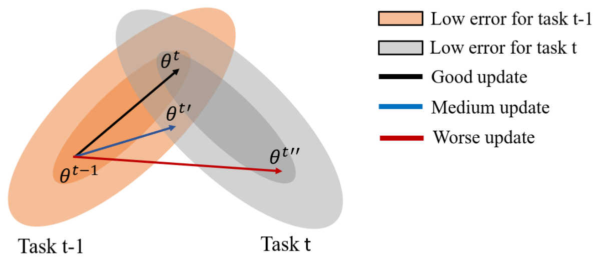

Fig. 1: Replay-based continual learning aims to acquire knowledge relevant to current task t and retain the representations learned from past task $t-1$. The black arrow represents the update of the minimum error for both $t-1$ and t. However, low-quality data points in the dataset can trigger parameter updates in the model that deviate from this objective (blue arrow). There is even a catastrophic forgetting of task $t-1$ (red arrow).

In this paper, we propose an Enhanced Experience Replay (EER) method to remove less representative instances during the learning phase, allowing for selecting a coreset of highly representative instances. Specifically, we introduce Enhanced Gradient Selection (EGS) algorithm to evaluate and rank each data point during the training phase based on its gradient relevance and difference. After eliminating low-quality and noise instances, we combine the most informative ones into a coreset. This aims to enhance the robustness of training across both current and previous tasks, thereby mitigating catastrophic forgetting.

The proposed EGS algorithm utilizes the gradient scores of each data point to determine its significance. However, in practice, some classes inherently possess a higher level of complexity, and excessive scores from these classes may lead to a imbalanced coreset. This issue is more evident in the replay buffer. To address this problem, we propose a Category Fairness (CF) selection algorithm to reduce dominant classes in the buffer memory. This ensures a balanced representation of all classes in the buffer, thus preventing representational drift from previous tasks.

Our main contributions are summarized as follows:

- We address the challenges in continual learning with high-noise and imbalanced data by utilizing a coreset of highly representative instances.

- We propose the EGS algorithm, a simple yet effective method for selecting the representative training subsets in each minibatch, thereby mitigating catastrophic forgetting.
- We introduce the CF selection algorithm for balancing the selected classes, thereby maintaining a class-balanced dataset and optimizing resource allocation in the buffer.

II. Related Work

In this section, we provide a review of continual learning methods. The existing coreset selection algorithms and continual learning settings are then discussed.

Regularization. The regularization-based methods utilize regularization terms or loss constraints during the learning process to reduce catastrophic forgetting. For instance, Elastic Weight Consolidation (EWC) [5] applies a penalty to the loss function based on the relevance of parameters for previous learning tasks. Variational Auto-Regressive Gaussian Processes (VAR-GPs) [12] constructs an auto-regressive variational distribution using sparse inducing point approximations for posterior distributions. In contrast, Synaptic Intelligence (SI) [13] estimates the importance of individual weights instead of the entire parameters. Learning without Forting (LWF) [4] utilizes knowledge distillation to transfer knowledge from a pre-trained model to a new model while learning new tasks.

Parameter isolation. The parameter isolation methods mainly focus on identifying and preserving major parameters that are crucial for previously-learned tasks. The remaining parameters are updated to adapt to new tasks. In [14], Xu *et al.* combined reinforcement learning to reward the preservation of essential knowledge from previous tasks during the learning of new tasks. In [15], Cossu *et al.* presented a method that integrates Gated Incremental Memories (GIMs) with recurrent neural networks. In [16], Temeem *et al.* introduced Continual Learning with Adaptive Weights (CLAW) that employs the adaptive weight mechanism and Bayesian inference for more principled weight adaptation.

Experience replay. The replay-based methods retain knowledge from previous tasks by periodically revisiting and retraining on past data, thereby diminishing catastrophic forgetting. For example, Chaudhry *et al.* [11] utilized a small episodic memory buffer to store and selectively replay data samples from previous tasks. The replay-based CL methods can be used as a robust module in other networks to achieve performance improvement. For example, iCaRL [17] combines a feature extractor with a nearest-mean-of-exemplars classifier to retain knowledge from previous classes while learning new classes. DER [18] uses a mixture of current task data and a dark dataset for knowledge distillation. Gradient Episodic Memory (GEM) [19] and Averaged GEM (A-GEM) [8] employ the projected gradient descent algorithm to maintain the model's capabilities when learning new tasks.

Coreset selection. The coreset-selection-based methods aim to maintain compact and informative training datasets. These coresets enable scalable and efficient Bayesian inference procedures by approximating the model's logit sum [20], [21]. Several studies have utilized coresets to approximate gradient sum of samples [22], [23]. For example, in [24], Johnson *et al.* proposed Importance Sampling (IS) to amplify the loss and gradients of influential samples. This method utilizes influence functions as the foundation in the amplification process. In [25], Yoon *et al.* proposed Onine Coreset Selection (OCS), a hybrid gradient algorithm for selecting coresets. In [26], Tiwari *et al.* introduced Gradient Coreset Replay (GCR) that maintains a representative set of data approximating the gradient of all previously seen data.

Continual learning setting. Beyond various learning approaches, CL can be applied to diverse scenarios depending on the availability of specific information and data at different stages. Based on the evolution of data aspects and their correlation with the learnable function, Gido *et al.* [27] classified CL into three categories, including Task Incremental Learning (Task-IL), Class Incremental Learning (Class-IL), and Domain Incremental Learning (Domain-IL). Task-IL associates each task with a unique task identifier, allowing for the design of distinct output layers or separate networks for each task. In contrast, Class-IL ignores the task identifiers and only requires a single model to classify different classes in each task. Domain-IL assumes that the tasks have the same structure, but the inputs may change incrementally.

III. Proposed Methodology

In this section, we briefly provide an overview of the training process for replay-based CL methods, then present the two proposed algorithms, including Enhanced Gradient Selection (EGS) and Class Fair (CF).

A. Overview

We focus on the problem of image classification with Class-IL. This can be characterized as sequential learning from a data stream that evolves over time. We conceptualize the situation of CL as a sequence of tasks $\mathcal{T} = \{\mathcal{T}_1, \mathcal{T}_2, ..., \mathcal{T}_T\}$, where each task consists of a set of data points with their labels, so that the task $\mathcal{T}_t$ includes $D_t = \{x_{t,n}, y_{t,n} \mid n = 1, ..., N_t\}$ [28]. Here, $x_{t,n}$ denotes the n^{th} image instance in task t, and the $y_{t,n}$ is its label. We assume that the sets of labels for any two different tasks, referred to as $y_{t,n}$ and $y_{k,n}$, do not have any classes in common (i.e., $\mathcal{T}_t, \mathcal{T}_k \in \mathcal{T}, \forall t \neq k, y_t \cap y_k = \phi$). For each task t, a CL algorithm learns the model parameters Θ to minimize the following cross-entropy loss function:

$$\mathcal{L} = \sum_{t=1}^{T} \sum_{n=1}^{N_t} \ell(y_{t,n}, f_\Theta(x_{t,n})), \tag{1}$$

where $\ell(\cdot)$ is the cross entropy loss (or any standard loss functions), and $f_\Theta(x_{t,n})$ is the predicted label of the input $x_{t,n}$. In practice, the model tends to excessively focus on the available data points of the current task and overwrite the learned representations from the previous tasks (a.k.a., catastrophic forgetting).

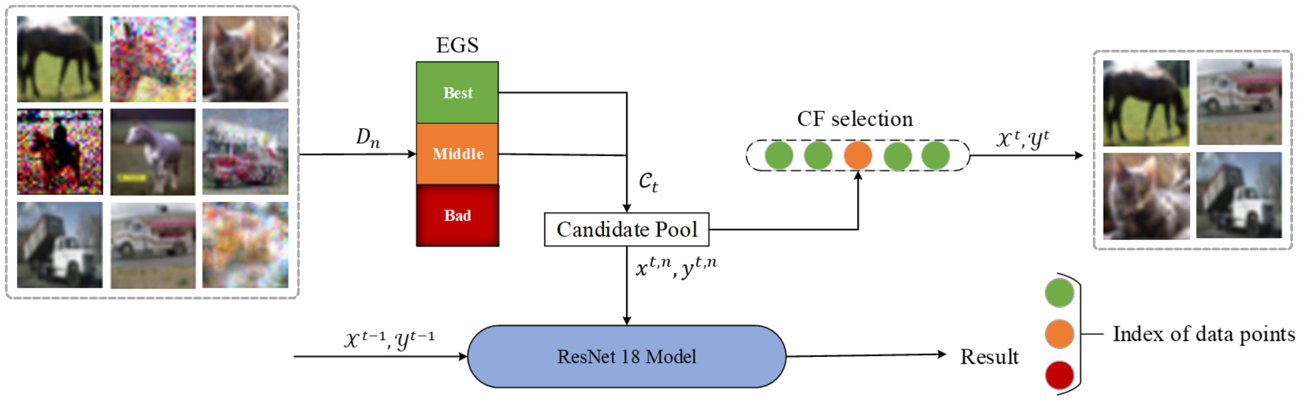

Fig. 2: Overview of the proposed continual learning framework. The proposed system employs Enhanced Gradient Selection (EGS) and Category Fairness (CF) selection methods to select the coreset and buffer. Inputs constitute an imbalanced minibatch, peppered with noise. Each instance earns a score or rating (green, yellow, and red) following the EGS algorithm application. Instances tagged with green or yellow indices are selected for the coreset and used in training. The buffer's candidate pool is then updated from the coreset using the EGS algorithm in conjunction with the CF algorithm.

To encourage the model to allocate attention to the previous tasks, ER-based methods maintain a memory buffer $\mathcal{M}$ as an episodic memory that stores the seen data points and their labels. The newly arrived data and samples retrieved from the buffer are then combined to train the model by minimizing the sum of two losses:

$$\mathcal{L} + \mathcal{L}' = \mathcal{L} + \sum_{t=1}^{T} \lambda \ell(y_t, f_\Theta(x_t)), \tag{2}$$

where λ is a hyperparameter representing the replay loss coefficient. Here, x_t and y_t denote data points stored in the buffer, i.e. $\{x_t, y_t\} \in \mathcal{M}_t$.

The replay buffer plays a key role in preserving the knowledge acquired from previous tasks. It presents selected excerpts of experiences that serve as reminders to reinforce the learned concepts. Replaying previous data points can partially maintain model performance on those tasks. However, most replay methods employ the standard reservoir sampling [10], often overlooking the quality of training datasets.

To maximize the retention of previous data, the memory buffer initially extracts as many samples as possible from the several tasks. Upon depletion of buffer capacity, data points in previous tasks are randomly discarded and replaced with newer data from the current task. However, this strategy may lead to significant model forgetting when a substantial part of the current data is composed of noise samples.

To overcome this problem, we propose the construction of a coreset, facilitating the models to learn both current and past data in the most sustainable manner. As depicted in Figure 2, we accomplish this by constructing a coreset under Enhanced Experience Replay (EER). First, Enhanced Gradient Selection (EGS) removes data points that minimally contribute to model gradient updates, such as low-quality, noisy, and class-imbalanced samples. It retains a small yet highly representative subset of data for model training. Second, to further enhance model generalization and diminish bias, we propose a Category Fairness (CF) selection algorithm to balance the number of classes in the buffer.

B. Enhanced Gradient Selection

We employ the gradient descent method during the training process, where the model's parameters are progressively adjusted in the direction of the negative gradients to minimize the loss function. Note that, low-quality samples may not effectively manifest changes in the model's gradients. Therefore, at the beginning of training process, we score and rank input data points after processing their gradients to filter low-quality samples.

However, calculating gradients for the entire dataset is costly and inefficient, with the cost rising exponentially as the dataset size increases. Inspired by OCS [25], we employ minibatches as computation units. Within each minibatch iteration, valuable samples are selected, ranked, and extracted, then subsequently added to a candidate pool $\mathcal{C}_t$. The strategy strikes a balance between maintaining computational efficiency and ensuring the extraction of high-quality samples for continued model training.

The gradient is a vector that represents the rate of increase in specific direction. A large magnitude signifies a steep increase, while a smaller one suggests a gradual incline. The direction of the gradient vector reveals where the function increases most rapidly and the magnitude shows the speed at which the function increases in the specific direction. In a situation where a minibatch has two samples with similar but opposite gradients, a problem can occur. Their combined gradient, which is used for model updates, could be reduced to zero. This scenario could lead to ineffective model parameter updates, thereby decelerating the learning process or possibly causing model stagnation. Furthermore, the simple multiplication of two gradient sets may misrepresent the concept of positive and negative gradients.

We perform an exhaustive evaluation of the gradient's direction $\mathcal{O}$ and magnitude $\mathcal{M}$. This approach provides a better understanding of the gradient, ensuring the robustness and stability of the model training process. To this end, we propose a criterion grounded in Relevance ($\mathbb{C}_{\mathcal{M}}^{\mathcal{R}}$ and $\mathbb{C}_{\mathcal{O}}^{\mathcal{R}}$) and Difference ($\mathbb{C}_{\mathcal{M}}^{\mathcal{D}}$ and $\mathbb{C}_{\mathcal{O}}^{\mathcal{D}}$). This method adaptively selects valuable instances, unswayed by outlier influences.

Relevance. Given n^{th} pair of data point and its labels $d_{t,n} = \{x_{t,n}, y_{x,n} \in D_t\}$, with corresponding gradient $\nabla g(d_{t,n})$ at task $\mathcal{T}_t$. The relevance in gradient magnitude $\mathbb{C}_{\mathcal{M}}^{\mathcal{R}}$ is defined as:

$$\mathbb{C}_{\mathcal{M}}^{\mathcal{R}}(d_{t,n}|D_t) = \frac{\nabla g(d_{t,n}) \cdot \nabla g(D_t)^T}{\|d_{t,n}\| \cdot \|D_t\|}, \tag{3}$$

where $\nabla g(D_t)$ is the mean gradient of all data points in task $\mathcal{T}_t$. To mitigate the impact of outliers on the results, we impose a norm $\|d_{t,n}\| \cdot \|D_t\|$. The results are normalized to ensure a consistent impact on the output.

The relevance in gradient direction $\mathbb{C}_{\mathcal{O}}^{\mathcal{R}}$ is defined by:

$$\mathbb{C}_{\mathcal{O}}^{\mathcal{R}}(d_{t,n}|D_t) = \frac{(\nabla g_c(d_{t,n}) \cdot B) \cdot B^T \cdot \nabla g(D_t)}{\|d_{t,n}\| \cdot \|D_t\|}, \tag{4}$$

where B is the orthogonal basis by the Gram-Schmidt process, and $\nabla g_c(d_{t,n}) = \nabla g(d_{t,n}) - \nabla g(D_t)$. The cosine similarity calculation aims to focus on the direction of each gradient with respect to the mean, rather than its absolute direction in high-dimensional space.

Difference. The computation approach for Difference is essentially the same as the aforementioned formula with slight variations in the input variables. Given n^{th} pair of data point and its labels $d_{t,n} = \{x_{t,n}, y_{x,n} \in D_t\}$, with corresponding gradient $\nabla g(d_{t,n})$ at task $\mathcal{T}_t$. The difference in gradient magnitude $\mathbb{C}_{\mathcal{M}}^{\mathcal{D}}$ is defined as:

$$\mathbb{C}_{\mathcal{M}}^{\mathcal{D}}(d_{t,n}|d_t) = \frac{\lambda}{N_t} \sum_{k=1}^{N_t} \frac{\nabla g(d_{t,n}) \cdot \nabla g(d_{t,k})^T}{\|d_{t,n}\| \cdot \|d_{t,k}\|}, \tag{5}$$

where $d_{t,k}$ are all samples except the data point n in task $\mathcal{T}_t$. i.e. $n \neq k$, $d_{t,k} \in D_t$. The total data points number is N_t and λ is a coefficient of Difference. A coefficient adapted to the current task can further improve the performance.

The difference in gradient direction $\mathbb{C}_{\mathcal{O}}^{\mathcal{D}}$ is defined as:

$$\mathbb{C}_{\mathcal{O}}^{\mathcal{D}}(d_{t,n}|d_t) = \frac{\lambda}{N_t} \sum_{k=1}^{N_t} \frac{(\nabla g_c(d_{t,n}) \cdot B) \cdot B^T \cdot \nabla g_c(d_{t,k})}{\|d_{t,n}\| \cdot \|d_{t,k}\|}. \tag{6}$$

The relevance and difference metrics perceive the minibatch as a rough representation of the current task dataset. This similarity measures the degree to which a given data instance describes the current task in each training step, with its value ranging from 0 to 1. In contrast, We represent the difference of each data point as the average deviation of that data point from other data points, with its value spanning from -1 to 0.

C. Enhanced Gradient Selection for CL

During the training process, the model is exposed to a continual stream of data. In practice, this data might contain noisy or redundant instances. These incoming data instances have the potential to disrupt and undermine the model's performance. To address this issue, we propose a strategy that leverages enhanced gradient vectors to identify the most beneficial instances for current and previous task training. Our approach of EGS for current task adaptation can thus be formally defined as follows:

$$\mathbb{C}_t = \arg\max \left(\mathbb{C}_{\mathcal{O}}^{\mathcal{R}} + \mathbb{C}_{\mathcal{M}}^{\mathcal{R}} + \mathbb{C}_{\mathcal{O}}^{\mathcal{D}} + \mathbb{C}_{\mathcal{M}}^{\mathcal{D}} \right). \tag{7}$$

Note that, from Eq. (7), we can obtain the top-k instances most valuable to the current task $\mathcal{T}_t$ and store them into the coreset $\mathcal{C}_t$. Upon the selection of this representative $\mathcal{C}_t$, we optimize the subsequent objective for current task at each iteration with the loss fuction described in Eq. (1).

At each iteration, we consider the selected coresets as candidates for the replay buffer. Once training for task $\mathcal{T}_t$ is complete, we combine CF to pick a buffer $\mathcal{M}_t$ from the collected candidates, or we may update coreset $\mathcal{C}_t$ iteratively to maintain the bounded buffer size required for continual learning. The pseudocode of EGS is shown in Algorithm 1.

Algorithm 1 The pseudocode of the proposed EGS strategy.

```
Require: Dataset: $D_t$.
         Prams: Candidate size=$k$, buffer size=$m$.
 1: Initialize: $\mathcal{C}_t \leftarrow \{\}$, $\mathcal{M}_t \leftarrow \{\}$.
 2: while Training do
 3:     Sample minibatch from task stream, $d_{t,n} \sim D_t$
 4:     Evaluate the gradient of each data point $d_{t,n}$
 5:     Evaluate the Relevance and the Difference
 6:     Combine and rank
 7:     Update candidate pool, $\mathcal{C}_t \leftarrow \left(d_{t,n}^{top-k}, \mathcal{D}_t\right)$
 8:     Compute overall loss $\mathcal{L} + \mathcal{L}'$
 9:     Update the gradient of $\mathcal{C}_t$
10:     Store buffer, $\mathcal{M}_t \leftarrow \left(d_{t,n}^{top-m}, \mathcal{C}_t\right)$
11:     CF $\left(\mathcal{M}_t, \mathcal{C}_t, m\right)$
12: end while
13: return $\mathcal{C}_t, \mathcal{M}_t$
```

D. Category Fairness selection

In the previous section, we have selected a coreset $\mathcal{C}_t$. However, our preliminary experiments show that certain classes in some tasks display an imbalance, which is particularly severe in the buffer. This phenomenon occurs because the feature distribution of certain classes can be more complex. It may overlap significantly with other classes, which makes classification more difficult. In such situations, the model tries to optimize the loss for these challenging classes. Consequently, the small gradient values may lead to a failure in selecting the coreset.

We propose a buffer selection strategy based on the coreset. We define m_t as the buffer capacity of task $\mathcal{T}_t$, ξ_t^r as the number of classes in the current classification task. After the completion of task $\mathcal{T}_t$, we perform another enhanced gradient selection on the coreset. We select the top-a data points for storage in the buffer candidate pool, where $a = \frac{m_t}{t}$. The ideal scenario is when $\xi_t^1 = \xi_t^2 ... = \xi_t^r = \frac{m_t}{tr}$, allowing the model to recall any past class ξ_t^r fairly and attain maximum theoretical robustness.

To achieve this objective, we randomly remove data points from the dominant classes in the buffer candidate pool until the count is $\frac{m_t}{tr}$. The memory vacancies are then filled with data points from vulnerable classes. The new buffer candidate pool can be joint trained with data points from the coreset without worrying about the imbalance problem. The pseudocode is shown in Algorithm 2.

Algorithm 2 The pseudocode of the proposed CF strategy.

```
Require: Candidate Pool: $\mathcal{C}_t$, Buffer: $\mathcal{M}_t$.
         buffer size=$m_t$.
 1: Initialize: $\mathcal{C}_t' \leftarrow \{\}$,
    Check the labels of $\mathcal{C}_t$ to find total number of classes $r$,
    Detect index of each class $(\xi_t^1, \xi_t^2 ..., \xi_t^r)$,
    Split buffer, $a = \frac{m_t}{tr}$.
 2: for $r$ do
 3:   if $a < \xi_t^r$: then
 4:     Random delete $(len(\xi_t^r) - a)$ index from $\xi_t^r$
 5:     Update buffer, $\mathcal{M}_t \leftarrow (\xi_t^r)$
 6:   else if $a > \xi_n^r$: then
 7:     Select data points of class r from $\mathcal{C}_t$
 8:     Store to the temporary cache $\mathcal{C}_t'$
 9:     Do EGS and pick top-$(a - \xi_n^r)$ data points
10:     Update buffer $\mathcal{M}_t$
11:   end if
12: end for
13: return $\mathcal{M}_t$
```

IV. Experiments and Results

In this section, we conduct performance comparisons with state-of-the-art continual learning baselines across various datasets to assess the efficacy of the EER method.

A. Datasets and Protocol

We conduct experiments on two benchmark datasets: CIFAR-10 and CIFAR-100 [19]. These datasets have an image resolution of 32×32 pixels with 10 and 100 classes of interest, respectively. To meet the requirement of sequential arrival of tasks, we use Sequential CIFAR-10 (S-CIFAR-10) and Sequential CIFAR-100 (S-CIFAR-10) to validate the proposed method. S-CIFAR-10 is divided into 5 tasks, and each task has two unique classes. In contrast, S-CIFAR-100 is divided into 20 tasks, each task has 5 non-overlapping classes. Figure 3 presents a minibatch in S-CIFAR-10 and S-CIFAR-100 datasets.

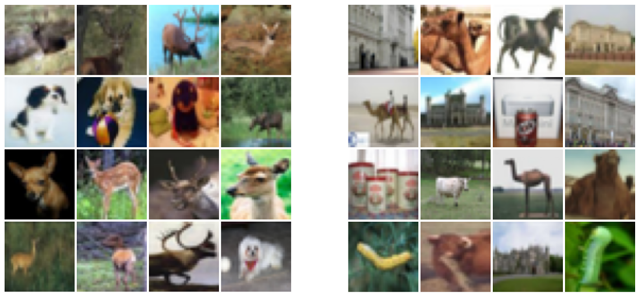

Fig. 3: Examples from S-CIFAR-10 and S-CIFAR-100.

B. Baselines

The proposed algorithm EER is compared with *six* state-of-the-art experience replay-based continual learning algorithms, including Experience Replay (ER) [11], Online Coreset Selection (OCS) [25], Average Gradient Episodic Memory (A-GEM) [8], Gradient-Based Sample Selection (GSS) [10], Incremental Classifier and Representation Learning (iCaRL) [17], and Dark Experience Replay (DER) [18]. In addition, we include two non-continual learning baselines SGD and JOINT to establish the upper and lower performance bounds, respectively. SGD trains the tasks sequentially without any continual learning strategy to prevent forgetting, whereas JOINT trains all tasks simultaneously.

C. Experimental Setting

We employ a standard ResNet-18 without pretraining as the backbone network for all CL methods across all datasets. We focus on the Class-IL scenarios, while all CL methods train with a buffer size of K=500 for S-CIFAR-10 and a buffer size of K=2000 for S-CIFAR-100. For the fairness of experimental results, we utilize the SGD optimizer for training all networks. Other CL methods are trained using the same network configurations.

D. Experimental Results

Table I shows the performance of continual learning upper bound (SGD) and lower bound (JOINT). The SDG produces an accuracy of 5.7% on S-CIFAR-100, which shows a significant decrement of 13.9% as compared to CIFAR-10. The JOINT achieves the average accuracy of 92.2% and 71.0% on S-CIFAR-10 and S-CIFAR-100, respectively.

TABLE I: AVERAGE TEST ACCURACY (%) OF THE LOWEST AND THE HIGHEST BOUNDS.

Method	S-CIFAR-10	S-CIFAR-100
SGD	19.6 (±0.1)	5.7 (±1.4)
JOINT	92.2 (±0.2)	71.0 (±0.2)

Table II summarizes the average accuracy of all tasks with the Class-IL setting. EER demonstrates superior performance over OCS by achieving 4.1% and 2.5% improvements on S-CIFAR-10 and S-CIFAR-100, respectively. This is attributed to the proposed coreset selection algorithm using gradient information. Interestingly, despite the absence of noisy or imbalanced instances in public datasets, both EER and OCS outperform other methods for all datasets. This shows their efficiency in filtering out less representative instances. Furthermore, the use of soft targets (logits) over hard targets (labels) is beneficial, as demonstrated by DER which achieves significant improvement as compared to ER. Our finding suggests that soft targets could potentially convey more valuable information for learning.

TABLE II: AVERAGE TEST ACCURACY (%) OF ALL TASKS ON THE TWO BENCHMARK DATASETS.

Method	**S-CIFAR-10** ***K=500***	**S-CIFAR-100** ***K=2000***
A-GEM [8]	22.7 (±0.6)	50.7 (±2.3)
iCaRL [17]	47.6 (±0.1)	60.3 (±0.9)
GSS [10]	49.7 (±3.9)	27.1 (±1.2)
ER [11]	62.0 (±1.7)	42.8 (±0.5)
DER [18]	70.5 (±1.7)	55.5 (±0.9)
OCS [25]	72.9 (±1.8)	60.5 (±0.6)
EER (Ours)	**74.1** (±1.4)	**61.0** (±0.6)

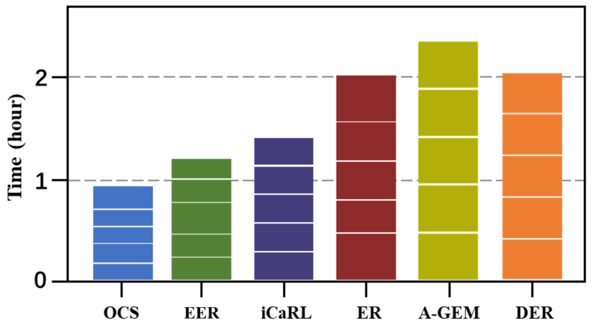

Fig. 4: Running time on S-CIFAR-10 when buffer=500.

For time complexity, a key consideration is to minimize the overall processing time in the setting of data stream processing. Preventing that the pace of training cannot keep up with the rate at which new data becomes available. To this end, we measure the running time of *seven* replay-based methods, upon completion of the final task. To maintain a level playing field, all tests were conducted under uniform conditions on a desktop computer, equipped with an NVIDIA 4090 GPU and an Intel i5-13500K CPU.

Figure 4 exhibits the execution time measured on S-CIFAR-10. Different blocks represent different tasks, and the sum of the five blocks is the overall completion time. The results show that the proposed EER is more time-efficient than most replay methods. Its running time is slightly higher than that of OCS due to a higher computational complexity. The non-coreset replay methods (ER, A-GEM, and DER) demonstrate similar time performance. We do not include the running time of GSS because it is far exceeding the average (over 14 hours), suggesting that GSS may be unsuitable for practical scenarios.

TABLE III: ABLATION STUDY OF CATEGORY FAIRNESS (CF) SELECTION AND ENHANCED GRADIENT SELECTION (EGS).

Components	**S-CIFAR-10** ***K=500***	**S-CIFAR-100** ***K=2000***
+ Relevance	70.8	57.7
+ Difference	73.5	59.9
+ CF selection	74.1	61.0

E. Ablation Study

In this section, we ablate important design elements in the proposed method, including the relevance, difference, and CF selection. The ablation study is presented in Table III. Although the relevance measure show conceptual merits in selecting highly-representative data points, it may suffer from selecting redundant samples, which negatively impact its effectiveness. Similarly, the difference itself is insufficient because it may select non-representative instances. Therefore, the integration of relevance and difference enhances the average accuracy and significantly mitigate the problem of catastrophic forgetting. A robust intervention in the form of category fairness selection forces the model to maintain an equal representation of classes in memory, bolstering the overall robustness of the model.

V. Conclusion

This paper proposes Enhanced Experience Replay (EER), a gradient-based coreset selection method for replay-based continual learning. Our approach utilizes the model's gradients as a criterion for coreset selection. We integrate this objective within the continual learning workflow to strategically update the replay buffer. The extensive experiments on the two benchmark datasets CIFAR-10 and CIFAR-100 shows that EER outperforms the existing methods with the same Class-IL setting. The results also demonstrate the scalibility of the proposed method as the number of tasks increases. However, we acknowledge that this current study can be further extended by developing parameter regularization mechanisms to manage the memory effectively. We also leave the exploration of applying EER for other computer vision tasks (e.g., object detection and semantic image segmentation [29], [30]) to a future study.

Acknowledgments

This work is supported by a research grant from the Australian Research Council. The authors also thank the NSW Space Research Network and the NSW State Government for financial support via SRN Research Pilot Project grant 2022-03.

References

[1] S. Thrun, "A Lifelong Learning Perspective for Mobile Robot Control," *Intelligent Robots and Systems*, pp. 201–214, 1995.

[2] M. McCloskey and N. J. Cohen, "Catastrophic Interference in Connectionist Networks: The Sequential Learning Problem," *Psychology of Learning and Motivation*, vol. 24, pp. 109–165, 1989.

[3] Z. Mai, R. Li, J. Jeong, D. Quispe, H. Kim and S. Sanner, "Online Continual Learning in Image Classification: An Empirical Survey," *arXiv preprint arXiv:2101.10423*, 2021.

[4] Z. Li and D. Hoiem, "Learning without Forgetting," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 40, no. 12, pp. 2934-2947, 2018.

[5] J. Kirkpatrick, R. Pascanu, N. Rabinowitz, et al.,"Overcoming Catastrophic Forgetting in Neural Networks," in *Proceedings of the National Academy of Sciences*, 2017, pp. 3521—3526.

[6] A. A. Rusu, N. C. Rabinowitz, and G. Desjardins, et al., "Progressive Neural Networks," *arXiv preprint arXiv:1606.04671*, 2016.

[7] S. Golkar, M. Kagan, and K. Cho, "Continual Learning via Neural Pruning," *arXiv preprint arXiv:1903.04476*, 2019.

[8] A. Chaudhry, M. A. Ranzato, M. Rohrbach, and M. Elhoseiny, "Efficient Lifelong Learning with A-GEM," *arXiv preprint arXiv:1812.00420*, 2018.

[9] M. H. Phan, T. A. Ta, S. L. Phung, L. T. Tran, A. Bouzerdoum, "Class Similarity Weighted Knowledge Distillation for Continual Semantic Segmentation," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022, pp. 16845-16854.

[10] R. Aljundi, M. Lin, B. Goujaud, and Y. Bengio, "Gradient Based Sample Selection for Online Continual Learning," in *Conference on Advances in Neural Information Processing Systems (NeurIPS)*, 2019, pp. 1–10.

[11] A. Chaudhry, M. Rohrbach, and M. Elhoseiny, et al., "On Tiny Episodic Memories in Continual Learning," *arXiv preprint arXiv:1902.10486*, 2019.

[12] S. Kapoor, T. Karaletsos, and T. Bui, "Variational Auto-Regressive Gaussian Processes for Continual Learning," in *International Conference on Machine Learning (ICML)*,2021 ,pp. 1-11.

[13] F. Zenke, B. Poole, and S. Ganguli, "Continual Learning through Synaptic Intelligence," in *International Conference on Machine Learning (ICML)*, 2017, pp. 3987–3995.

[14] J. Xu and Z. Zhu, "Reinforced Continual Learning," in *International Conference on Neural Information Processing Systems*, 2018, pp. 907—916.

[15] A. Cossu, A. Carta, and D. Bacciu, "Continual Learning with Gated Incremental Memories for Sequential Data Processing," in *International Joint Conference on Neural Networks (IJCNN)*, 2020, pp. 1–8.

[16] A. Tameem, H. Zhao, and R. E. Turner, "Continual Learning with Adaptive Weights (Claw)," *arXiv preprint arXiv:1911.09514*, 2019.

[17] S. A. Rebuffi, A. Kolesnikov, G. Sperl, and C. H. Lampert, "iCaRL: Incremental Classifier and Representation Learning," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017 pp. 2001–2010.

[18] P. Buzzega, M. Boschini, A. Porrello, D. Abati, and S. Calderara, "Dark Experience for General Continual Learning: A Strong, Simple Baseline," in *Conference on Advances in Neural Information Processing Systems (NeurIPS)*, 2020, pp. 15920–15930.

[19] D. Lopez-Paz, and M. A. Ranzato, "Gradient Episodic Memory for Continual Learning," in *Conference on Advances in Neural Information Processing Systems (NeurIPS)*, 2017, pp. 1–10.

[20] J. Huggins, T. Campbell, and T. Broderick, "Coresets for Scalable Bayesian Logistic Regression," in *Conference on Advances in Neural Information Processing Systems (NeurIPS)*, 2016, pp. 1–9.

[21] J. Zhang, R. Khanna, A. Kyrillidis, and S. Koyejo, "Bayesian Coresets: Revisiting the Nonconvex Optimization Perspective," in *International Conference on Artificial Intelligence and Statistics*, 2021, pp. 2782–2790.

[22] B. Mirzasoleiman, J. A. Bilmes, and J. Leskovec, "Coresets for Data-efficient Training of Machine Learning Models," *International Conference on Machine Learning (ICML)*, 2020, pp. 1–11.

[23] B. Mirzasoleiman, K. Cao, and J. Leskovec, "Coresets for Robust Training of Deep Neural Networks Against Noisy Labels," *Conference on Advances in Neural Information Processing Systems (NeurIPS)*, 2020, pp. 11465–11477.

[24] T. B. Johnson, and C. Guestrin, "Training Deep Models Faster with Robust, Approximate Importance Sampling," in *Conference on Advances in Neural Information Processing Systems (NeurIPS)*, 2018, pp. 1–11.

[25] J. Yoon, and D. Madaan, E. Yang, and S. J. Hwang,"Online Coreset Selection for Rehearsal-based Continual Learning," in *International Conference on Learning Representations (ICLR)*, 2022, pp. 1–12.

[26] R. Tiwari, K. Killamsetty, R. Iyer, and P. Shenoy, "GCR: Gradient Coreset Based Replay Buffer Selection for Continual Learning," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022, pp. 99–108.

[27] G.M. Ven, T. Tuytelaars and S. Tolias, "Three Types of Incremental Learning," *Nature Machine Intelligence*, vol. 4, pp. 1185-1197, 2022.

[28] Q. Yan, D. Gong, Y. Liu, A. van den Hengel, J. Shi. "Learning Bayesian Sparse Networks with Full Experience Replay for Continual Learning," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022, pp. 109–118.

[29] Y. Di, S. L. Phung, J. van den Berg, J. Clissold, and A. Bouzerdoum, "TP-YOLO: A Lightweight Attention-based Architecture for Tiny Pest Detection," in *IEEE International Conference on Image Processing (ICIP)*, 2023, pp. 3394-3398.

[30] H. T. Le, S. L. Phung, and A. Bouzerdoum, "Bayesian Gabor network with uncertainty estimation for pedestrian lane detection in assistive navigation," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 32, no. 8, pp. 5331-5345, 2022.