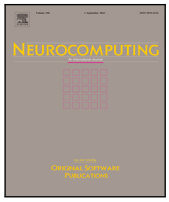
Neurocomputing 572 (2024) 127204

Contents lists available at ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom



AdaER: An adaptive experience replay approach for continual lifelong learning

Xingyu Li [a], Bo Tang [b,*], Haifeng Li [c]

[a] Department of Computer Science, Tulane University, New Orleans, 70118, LA, USA
[b] Department of Electrical and Computer Engineering, Worcester Polytechnic Institute, Worcester, 01609, MA, USA
[c] School of Geosciences and Info-Physics, Central South University, Changsha, 410083, Hunan, China

ARTICLE INFO

Communicated by L. Romeo

Keywords:
Continual lifelong learning
Contextual memory recall
Sequential learning
Experience replay

ABSTRACT

Continual lifelong learning is an machine learning framework inspired by human learning, where learners are trained to continuously acquire new knowledge in a sequential manner. However, the non-stationary nature of streaming training data poses a significant challenge known as catastrophic forgetting, which refers to the rapid forgetting of previously learned knowledge when new tasks are introduced. While some approaches, such as experience replay (ER), have been proposed to mitigate this issue, their performance remains limited, particularly in the class-incremental scenario which is considered natural and highly challenging. In this paper, we present a novel algorithm, called adaptive-experience replay (AdaER), to address the challenge of continual lifelong learning. AdaER consists of two stages: memory replay and memory update. In the memory replay stage, AdaER introduces a contextually-cued memory recall (C-CMR) strategy, which selectively replays memories that are most conflicting with the current input data in terms of both data and task. Additionally, AdaER incorporates an entropy-balanced reservoir sampling (E-BRS) strategy to enhance the performance of the memory buffer by maximizing information entropy. To evaluate the effectiveness of AdaER, we conduct experiments on established supervised continual lifelong learning benchmarks, specifically focusing on class-incremental learning scenarios. The results demonstrate that AdaER outperforms existing continual lifelong learning baselines, highlighting its efficacy in mitigating catastrophic forgetting and improving learning performance.

1. Introduction

State-of-the-art machine learning (ML) approaches have achieved remarkable performance in various tasks as image classification [1], distributed optimization [2,3], and security [4]. However, when trained with new tasks from non-stationary distributions, these models tend to rapidly forget previously learned information, a phenomenon known as "catastrophic forgetting" [5,6]. In contrast, human brains possess the ability to learn different concepts and perform conflicting tasks in a lifelong sequential manner, which is a desirable characteristic for artificial intelligent systems. As a result, there has been a growing interest in the field of continual lifelong learning [7,8], aiming to train artificial learners with non-stationary streaming training data, temporally correlated inputs, and minimal supervision.

One commonly used approach to address catastrophic forgetting is the utilization of previously trained experience through "memory replay", which involves rehearsing the memory of previously learned tasks along with new incoming tasks to reactivate relevant knowledge in the learning model, promoting knowledge consolidation [9,10]. Replay-based methods can be categorized into two groups based on how previous memories are used: (i) Experience replay, which stores raw training examples in a limited memory buffer [11]. (ii) Generative replay, which trains a separate generative model to generate synthetic samples for previously learned tasks [12]. Recent research has enhanced experience replay methodologies through diverse approaches. Specifically, [13] introduces a model-free λ-policy iteration using reinforcement learning techniques. [14] advances experience replay by focusing on interactions among distributed agents in a dynamic game setting. Furthermore, [15] showcases a Hamiltonian-driven adaptive dynamic programming strategy, emphasizing greater efficiency in experience replay.

Although there have been debates about the utilization of seen experiences in replay-based methods, recent studies suggest that these settings are necessary, especially in more challenging continual learning scenarios [16]. For instance, existing approaches struggle with the class-incremental (class-IL) scenario, where the learner needs to

* Corresponding author.
E-mail address: btang1@wpi.edu (B. Tang).

https://doi.org/10.1016/j.neucom.2023.127204
Received 22 August 2023; Received in revised form 25 October 2023; Accepted 26 December 2023
Available online 28 December 2023

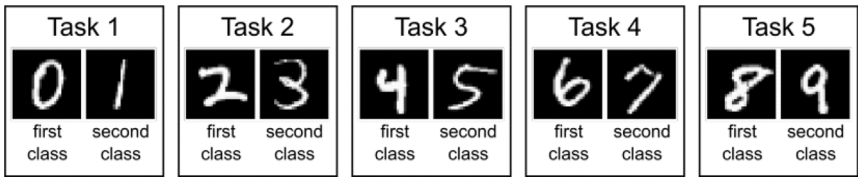


Fig. 1. Split-MNIST: continual lifelong learning example.

perform all learned tasks independently, as opposed to the simpler task-incremental (task-IL) problem where the learner makes decisions for a single task. The class-IL scenario, being more realistic and challenging, has gained popularity in recent years. In this context, the superiority of replay-based methods, such as experience replay (ER) [17], is evident compared to other approaches, especially in the class-IL scenario. Typically, ER and related methods involve two main stages: update and replay, which determine how unseen data is added to memory and which samples should be replayed, respectively. However, current ER methods have faced criticism for their random sampling strategies in both memory update and replay, and addressing this challenge remains an open problem in the field [18].

To overcome these limitations, we propose a novel continual learning algorithm called Adaptive Experience Replay (AdaER), which aims to enhance the efficiency of existing experience replay (ER) based approaches. In the replay stage, AdaER introduces the Contextually-Cued Memory Recall (C-CMR) method, which selects memories for replay based on contextual cues instead of random sampling. These contextual cues are derived from the most interfering examples (i.e., data-conflicting) and the associated forgetting tasks (i.e., task-conflicting) in the memory buffer. For example, as depicted in Fig. 1, when the learner encounters the fourth task, which involves classifying the digits 6 and 7, it is most prone to forgetting its previously acquired capability to classify digit 1 from task 1. This is primarily because the digit 7 shares significant similarities with digit 1, thereby affecting the model's retention. In such a scenario, replaying memories associated with digit 1 can effectively mitigate this forgetting. Concurrently, revisiting the learned classification boundaries for digit 1 in task 1 can also influence the model's performance on that task. Therefore, it would be advantageous to simultaneously replay memories related to digit 0 from task 1 to maintain a balanced learning landscape.

Additionally, to enhance the strategy for updating the memory buffer, AdaER introduces an innovative technique known as Entropy-Balanced Reservoir Sampling (E-BRS). This method is designed to optimize the information entropy within the replay memory buffer, thereby ensuring a more effective and balanced memory update strategy. In real-world scenarios, sequential streaming training data often exhibit imbalanced distributions, posing an additional challenge in continual learning. For example, the number of training examples for minority classes in the memory buffer may be limited, exacerbating the forgetting issue. The E-BRS method provides a balanced memory updating strategy that mitigates the bias caused by imbalanced training data. The contributions of this work can be summarized as follows:

- We propose the AdaER algorithm, a novel two-stages approach to overcome obstacles in ER-based continual lifelong learning.
- For the replay stage of AdaER, we introduce the C–CMR method, which selects memories for replay based on contextual cues related to interference and task performance. To further refine the update stage, AdaER reformulates the memory updating strategy via the E-BRS method, which optimizes the memory buffer's efficacy through the amplification of information entropy.
- Through extensive experiments on multiple benchmarks, the results demonstrate that AdaER outperforms existing continual lifelong learning baselines. For example, AdaER achieves 74.0% testing accuracy at the Split-FMNIST benchmark, 6.3% higher than ER.

The rest of this paper is organized as follows: the recent studies of continual lifelong learning are summarized in Section 2. The problem statement and some preliminaries are given in Section 3. The proposed AdaER algorithm with C–CMR replay and E-BRS update methods is demonstrated in detail in Section 4. Moreover, A comprehensive experiment study is presented in Section 5, followed by a conclusion in Section 6.

2. Related work

2.1. Continual learning approaches

2.1.1. Memory replay

Replay-based methods draw inspiration from the relationship between the mammalian hippocampus and neocortex in neuroscience, aiming to replicate the interleaving of current training tasks and previously learned memories. This interplay between real episodic memories and generalized experiences offers valuable insights into knowledge consolidation. Early examples of this concept include the use of a dual-memory learning system in [19] to mitigate forgetting. More recent approaches, such as the modified self-organizing map (SOM) with short-term memory (STM) in [20], have been developed as part of incremental learning frameworks.

Taking inspiration from the generative role of the hippocampus, [12] proposed a dual-model architecture consisting of a generative model and a continual learning solver, enabling the sampling and interleaving of trained examples, known as Generative Replay (GR). Gradient Episodic Memory (GEM) [21] stores a subset of seen examples as episodic memory that has a positive impact on previous tasks, while Averaged GEM (A-GEM) [11] improves the computational and memory efficiency of GEM through an averaging mechanism. Experience Replay (ER) [17] uses reservoir sampling [22] to update the memory buffer, thereby approximating the data distribution. Works such as [18] have further improved the memory update and replay processes to enhance performance, and FoCL [23] focuses on the feature space regularization instead of parameter space. [24] provides a multi-criteria subset selection strategy to overcome the unstable problem of ER. Other ML topics as Meta-learning [25] and segmentation [26] have also been applied to replay-based continual learning approaches. Despite being memory-intensive, replay-based methods have generally shown high performance.

2.1.2. Other continual learning approaches

Apart from memory replay-based approaches, other categories of continual learning approaches include regularization methods and dynamic architecture methods. While our focus in this paper is on memory replay-based approaches, we briefly discuss these other categories below.

Regularization methods aim to mitigate forgetting by retraining the lifelong learning model while balancing the knowledge of previous tasks and the current task. Learning without Forgetting (LwF) [27] achieves this by distilling knowledge from a large model to a smaller model, ensuring that the predictions for the current task align with those of previously learned tasks. Elastic Weight Consolidation (EWC) [7] identifies important weights for previously seen examples using Fisher Information and restricts their changes through quadratic terms

in the loss function. Synaptic Intelligence (SI) [28] penalizes parameters in the model's objective function in an unequal manner based on gradient information, identifying influential parameters. ISYANA [29] considers the relationship between tasks and the model, as well as the relationship between different concepts. [30] adopts variational auto-encoders to achieve exemplar-free continual lifelong learning. Regularization approaches are known for their ability to continually learn new tasks without storing seen examples or expanding the model's architecture. However, the trade-off in the loss function can lead to complex performance dynamics between seen and new tasks, especially when the task boundaries are unknown.

Dynamic architecture approaches modify the model's architecture to accommodate new tasks by adding new neural resources. Some works, such as [31], adopt a linear growth of the number of models in response to new tasks. Progressive Neural Networks (PNN) [32] preserves the previously trained network and allocates new sub-networks with fixed capacity to learn new information. Dynamically expanding network (DEN) [33] incrementally increases the number of trainable parameters to adapt to new examples, providing an online method for expanding network capacity. Dynamic architecture approaches offer the advantage of preserving knowledge of seen tasks with fixed model parameters. However, they face challenges in preventing parameter growth from becoming too rapid, which could lead to complexity and resource demands. Additionally, selecting appropriate parameters to target the test task is a significant challenge in these approaches.

2.2. Three scenarios of continual learning

The evaluation of continual lifelong learning approaches can be challenging due to differences in experimental protocols and access to task identity during testing. To facilitate meaningful comparisons, recent work by [16] has introduced three standardized evaluation scenarios of increasing difficulty for continual lifelong learning. These scenarios have been adopted by several subsequent studies [34].

We illustrate these scenarios using the popular continual learning benchmark, split MNIST [28], where the ten handwritten digits are learned sequentially in multiple tasks with limited classes, as shown in Fig. 1. The first scenario is task-incremental learning (**task-IL**), which is the easiest scenario where the learner always knows the target learning task. The second scenario is **domain-IL** [16,35], where the model is trained on tasks from different domains but with the same labels. For example, the learner needs to remember whether the testing digits come from the previous seen MNIST domain or the new learned hand-crafted digits from other domains. The most challenging scenario is **class-IL**, where the learner is required to distinguish all previously seen classes without knowing the task identity. Recent reports [36] indicate that the class-IL scenario is more realistic and poses a greater challenge for incrementally learning new classes. In this paper, we specifically focus on enhancing the performance of memory replay-based continual learning approaches in the class-IL scenario.

3. Problem statement

In conventional machine learning, the goal is to train a model f with parameters θ to predict outcomes of a stationary dataset $\mathcal{X}$, where the training sample $\{\boldsymbol{x}, y\} \in D$ that:

$$\theta^* = \underset{\theta}{\operatorname{argmin}}\, \mathbb{E}_{(\mathbf{x},y)\sim D}[l(f_\theta(\mathbf{x}), y)], \tag{1}$$

where $l(\cdot,\cdot)$ is the loss function that denotes the empirical risk of f_θ over D. However, in continual lifelong learning, the learning environment is typically non-static. We $\mathcal{X}$ divided into T tasks, which is indexed as $\mathcal{X}_t \sim D_t, t \in [1,\ldots,T]$. Particularly, when learning the tth task, the classifier has no access to the previous tasks. We denote the distribution of previously seen training examples in $\mathcal{X}_s = \{\mathcal{X}_1,\ldots,\mathcal{X}_{t-1}\}$ as $(\mathbf{x}_s, y_s) \sim D_s$, where the objective of a ML learner to achieve Eq. (1) can be described as:

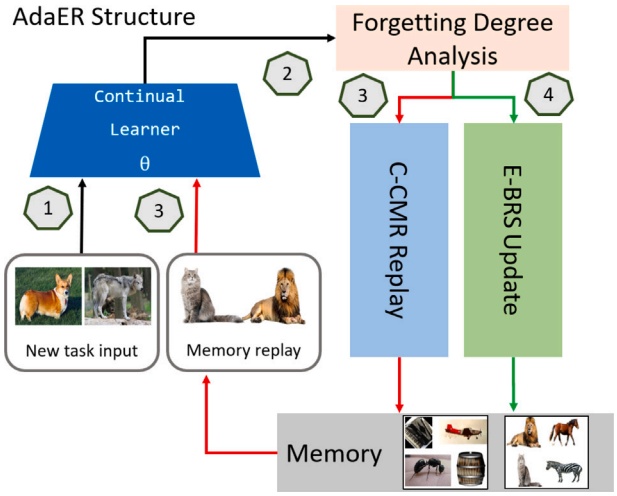


Fig. 2. System diagram of AdaER: an adaptive experience replay algorithm with the developed contextually-cued memory recall (C-CMR) method for the replay stage and the entropy-balanced reservoir sampling (E-BRS) strategy for the update stage: (1) the continual learner encounters the new task; (2) AdaER performs memory forgetting degree analysis; (3) AdaER uses C–CMR method to select memories for replay, learned with the new task inputs for model update; (4) the E-BRS method is applied to update the memory buffer with the forgetting analysis in parallel.

$$\theta^* = \underset{\theta}{\operatorname{argmin}}(\mathcal{L}_t + \mathcal{L}_s), \tag{2}$$

where the first part $\mathcal{L}_t = \mathbb{E}_{(\mathbf{x}_t,y_t)\sim D_t}[l(f_\theta(\mathbf{x}_t), y_t)]$ requires the learner to rapidly learn the current task, and the second part $\mathcal{L}_s = \mathbb{E}_{(\mathbf{x}_s,y_s)\sim D_s}[l(f_\theta(\mathbf{x}_s), y_s)]$ denotes the requirement of not forgetting the previous knowledge. Without the integration of $\mathcal{L}_s$, a typical ML learner will suffer from the catastrophic forgetting issue [6], e.g., due to the lack of stability in neural networks [7]. The requirement of minimizing both $\mathcal{L}_t$ and $\mathcal{L}_s$ is also known as the stability-plasticity dilemma in existing studies [21]. To better present this challenge, we summarize the following wide accepted assumptions in replay based continual learning researches [17,18] as follows: (i) The space of memory $\mathcal{M}$ is finite, which means only a subset of the experiences can be stored. (ii) The frequency of replaying experience is set to be the same as the frequency of learning new task batches. Meanwhile, the rehearsal of learned memory will impact the ability of learning new tasks. To balance the stability-plasticity dilemma, the size of replay buffer $\mathcal{R}$ is limited to be close to the size of training batch and much smaller than $\mathcal{M}$. (iii) All the tasks are assumed to be equally important.

To address this challenge, there have been several existing approaches in recent years, among which one popular method is known as experience replay (ER) [17]. The central feature of ER is to leverage a memory storage $\mathcal{M} \sim D_m$ for previously seen training samples, where $\mathcal{M} = \{(\mathbf{x}_1, y_1),\ldots,(\mathbf{x}_m, y_m)\}$ and $|\mathcal{M}| = M$. Obviously, it is not realistic to either store every seen training samples in $\mathcal{M}$ or replay each example in $\mathcal{M}$ at every current learning step. Typically, given a limited size of memory, the mechanism of ER could be summarized into the following two steps: (i). *Memory update:* $\mathcal{M}$ is updated when it learns more tasks, followed by a reservoir sampling method [22]. (ii). *Memory replay:* during the training of the current task t, a batch of examples $\mathcal{B}_m = \{\mathbf{x}_m, y_m\}$ are randomly sampled from $\mathcal{M}$ that is interleaved with the current batch $\mathcal{B}_t$ to improve the stability of the learner. Specifically, during the learning of task t, ER approach addresses the objective in Eq. (2) as $\theta = \operatorname{argmin}_\theta(\mathcal{L}_t + \mathcal{L}_m)$, where $\mathcal{L}_m = \mathbb{E}_{(\mathbf{x}_m,y_m)\sim D_m}[l(f_\theta(\mathbf{x}_m), y_m)]$.

However, the performance of ER is affected by the distribution discrepancy between the replayed batch $\mathcal{B}_m$ and the previously seen data

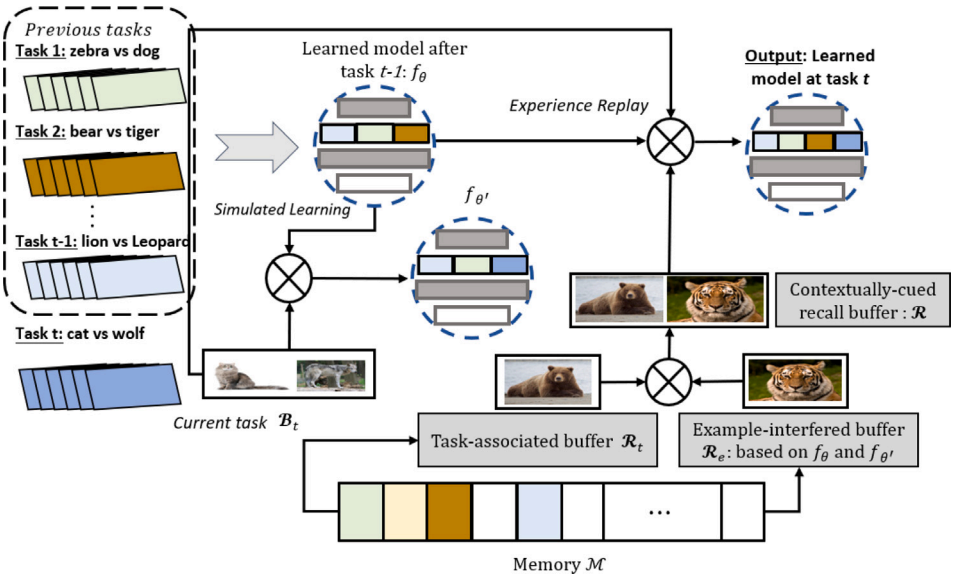


Fig. 3. Illustration of the developed C–CMR method: the most contextually-cued memories $\mathcal{R}$ are replayed to mitigate the forgetting with the combination of example-interfered buffer $\mathcal{R}_e$ and task-associated buffer $\mathcal{R}_t$.

$\mathcal{D}_s$, which becomes more pronounced as memory resources become limited. Additionally, random reservoir sampling used for memory update can lead to imbalanced datasets, exacerbating the issue of catastrophic forgetting, especially for classes with fewer training examples.

To address these limitations, our proposed algorithm focuses on improving both memory replay and update. We aim to mitigate the distribution discrepancy between $\mathcal{B}_m$ and $\mathcal{D}_s$ and alleviate the imbalance issue in real-world scenarios. By addressing these challenges, we aim to enhance the performance of ER-based continual lifelong learning approaches.

4. Methods

In this section, we present our novel continual lifelong learning algorithm, called Adaptive Experience Replay (AdaER). The structure of AdaER is depicted in Fig. 2, and it addresses the limitations of existing experience replay (ER) methods by enhancing both the replay and update stages. In AdaER, the memory tuples in $\mathcal{M}$ store not only the previously seen training data, but also the corresponding task IDs. Each memory content in $\mathcal{M}$ is denoted as $\mathbf{x}_m, y_m, t_m$, where $\mathbf{x}_m$ represents the input sample, y_m denotes the corresponding label, and t_m indicates the task ID.

To improve the replay stage, AdaER introduces a Contextually-Cued Memory Recall (C-CMR) method (Section 4.1). C–CMR determines which memories should be replayed based on contextual cues, considering the interference caused by the data-conflicting and task-conflicting examples in the memory buffer. The goal is to select the most relevant memories for effective knowledge consolidation.

For the update stage, AdaER enhances the memory updating strategy by maximizing the corresponding information entropy. This strategy is known as Entropy-Balanced Reservoir Sampling (E-BRS) (Section 4.2). By maximizing the information entropy, AdaER mitigates the imbalanced distribution issue that can arise in real-world scenarios, leading to improved performance and reduced forgetting.

The detailed design and implementation of AdaER are provided in the subsequent sections, which include the C–CMR method and the E-BRS strategy. These advancements aim to enhance the efficiency and effectiveness of ER-based continual lifelong learning methods, addressing the challenges associated with memory replay and update.

4.1. Contextually-cued memory recall

In the C–CMR method, we illustrate its functionality using a continuous animal classification scenario depicted in Fig. 3. The goal of C–CMR is to select appropriate memory examples for the seen animal classes. To achieve this, C–CMR employs two buffers: the example-interfered buffer $\mathcal{R}_e$ and the task-associated buffer $\mathcal{R}_t$. These buffers help determine the relevant memories from both the data-conflicting and task-conflicting perspectives. The selected memories are then stored in the contextually-cued buffer $\mathcal{R}$.

First, the C–CMR identifies the most interfered samples based on data conflicts and stores them in $\mathcal{R}_e$. These data-conflicting samples provide valuable information for knowledge consolidation. Simultaneously, C–CMR investigates the forgetting of associated tasks stored in $\mathcal{R}_e$ and identifies the task-related samples. These task-conflicting samples are then stored in $\mathcal{R}_t$. By considering both data conflicts and task conflicts, C–CMR leverages the information from both buffers, $\mathcal{R}_e$ and $\mathcal{R}_t$, to create the contextually-cued buffer $\mathcal{R}$.

The contextually-cued buffer $\mathcal{R}$ contains selected memory examples that are relevant for effective knowledge consolidation. By combining the information from the example-interfered buffer and the task-associated buffer, C–CMR ensures that the replayed memories are contextually appropriate and contribute to mitigating catastrophic forgetting in continual lifelong learning scenarios.

4.1.1. Example-interfered buffer

To identify memory examples that conflict with the current learning task, C–CMR introduces a virtual classifier f'_θ, which is trained on the current batch $\mathcal{B}_t$ without any memory replay. This approach is inspired by previous work [18] and addresses the stability-plasticity dilemma in lifelong learning. The learning of f'_θ at the tth task is formulated as a one-step stochastic gradient descent (SGD) optimization problem, where the model parameters θ' are updated using the gradients computed on $\mathcal{B}_t$ with a learning rate α. The update is performed according to the following equation:

$$\theta' = \theta - \alpha \nabla_\theta l(f_\theta; \mathcal{B}_t), \tag{3}$$

where $\nabla_\theta l(\cdot)$ represents the gradient of the loss function $l(\cdot)$ with respect to the model parameters θ. By updating θ using the gradients calculated on the current batch, the virtual classifier f'_θ is obtained.

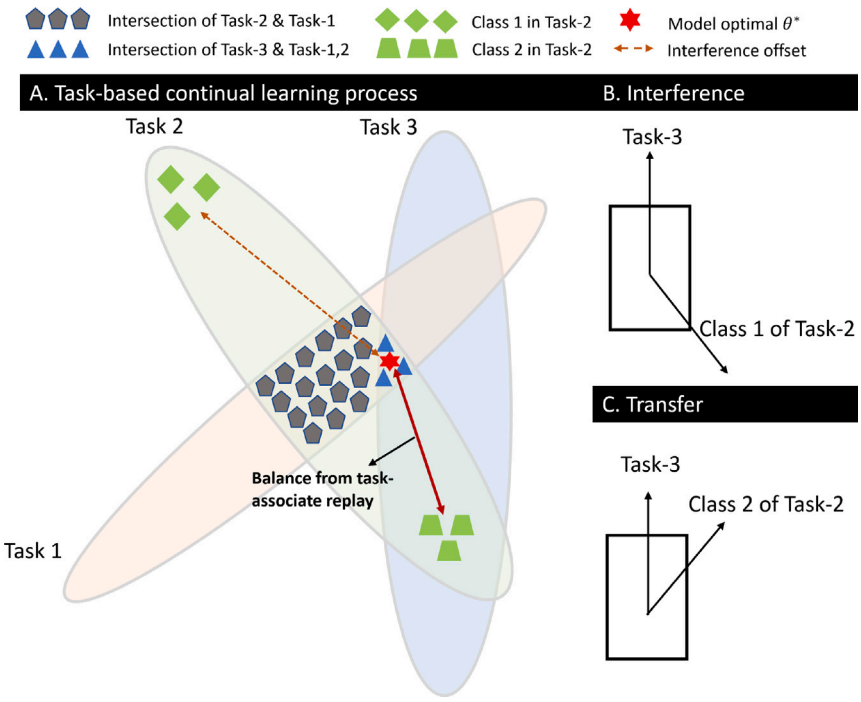


Fig. 4. Task-associated interference-transfer relationship with a three-task continual learning scenario, where Task 3 is interfered to Class 1 of Task 2 and transferred to Class 2 of Task 2.

The motivation behind using f'_θ is to assess the forgetting degree of each memory example with respect to the current learning task. Different memory examples exhibit varying degrees of forgetting, with some being transferable to the new task and others interfering or being forgotten. By comparing the performance of f'_θ and the original model f_θ on the memory examples, C–CMR quantifies the forgetting degree of each example, enabling the selection of conflicting memory examples for further analysis and handling. To quantize the forgetting degree of each memory example, we introduce a score vector $\mathbf{s}$ with a criterion, calculating $s(m)$ for the mth memory sample as:

$$s(m) = l(f_{\theta'}(\mathbf{x}_m), y_m) - l(f_\theta(\mathbf{x}_m), y_m), \tag{4}$$

where $\mathbf{s} \in \mathbb{R}^{M\times 1}$. Note that when the value of $s(m)$ is higher, the mth memory example is considered with a higher degree of forgetting against the learning of $\mathcal{B}_t$. As a result, C–CMR develops the example-interfered buffer $\mathcal{R}_e$ by choosing the top-p interfered memories instead of randomly sampling from $\mathcal{M}$, where $p = |\mathcal{R}_e|$.

4.1.2. Task-associated buffer

Although the example-interfered buffer $\mathcal{R}_e$ captures the most representative forgotten examples from a data-conflicting perspective, it does not consider the relationship between transfer and interference among the learned tasks. This relationship is illustrated in a three-task continual learning example shown in Fig. 4. The objective in this scenario is to find the optimal model $\theta^\star$ with the training task order from task 1 to task 3, which lies in the shown overlapping region. However, we can notice a surprising observation that the forgetting impacts from task 3 to the two classes of task 2 are different: (1) Class 1 of task 2 is situated outside the intersection of tasks 2 and 3. Consequently, it faces a heightened risk of interference and is more prone to forgetting. (2) Conversely, Class 2 of task 2 is nestled within the confines of the task 3 regions, suggesting a favorable transfer of knowledge. Given the framework of the example-interfered buffer, memory related to class 1 of task 2 emerges as a prime candidate for inclusion in $\mathcal{R}_e$. This is attributed to its pronounced distance from $\theta^\star$, leading to a magnified $s(m)$ value.

However, repeatedly revisiting $\mathcal{R}_e$ might inadvertently skew the model away from the optimal $\theta^\star$. This deviation could engender interference, potentially compromising the performance of task 2 due to the excessive rehearsal of class 1. To achieve the balance, it is imperative to sample not just the most forgotten examples but also their counterparts with the same task, like class 2 from task 2. With this objective in mind, the C–CMR introduces the task-associated buffer, $\mathcal{R}_t$. This buffer lays particular emphasis on classes whose vulnerability is not solely determined by the established forgetfulness metric, $\mathbf{s}$.

We denote the task IDs in $\mathcal{R}_e$ as $j = [1, \ldots, t_e]$, where $1 \le t_e < p$. If the number of jth task samples captured in $\mathcal{R}_e$ as p_j, $\mathcal{R}_t$ capture the number of training memories from the other classes in the task j as:

$$q_j = |\mathcal{R}_t| \times \frac{p_j}{p}, \tag{5}$$

where $|\mathcal{R}_t| + p = |\mathcal{R}|$. Note that the obtained q_j can be viewed as a weighted factor of the jth task in $\mathcal{R}_t$: when task j is more interfered with during the continual learning process, the chances of its training samples being selected into $\mathcal{R}_e$ is higher, whose data distribution needs to pay more attention for offset correction.

4.1.3. Contextually-cued recall

To address the challenge of transfer and interference in memory replay, our proposed C–CMR method combines both the example-interfered buffer $\mathcal{R}_e$ and the task-associated buffer $\mathcal{R}_t$ for replay. The final replay buffer obtained in C–CMR is denoted as $\mathcal{R} = \mathcal{R}_e + \mathcal{R}_t$. The memory replay process of C–CMR is summarized in Algorithm 1.

Meanwhile, we provide the convergence analysis of the proposed AdaER in Algorithm 1, based on two well-accepted assumptions in the field of continual lifelong learning [37]: (1) For any task t, the loss objective $l_t(f_\theta)$ is L-smooth with a constant $L > 0$. (2) The variance of each learning task and $\mathcal{L}_t + \mathcal{L}_s$ are all upper bounded by a corresponding constant. For clarity, we formulate the model update process in the provided C–CMR method as:

$$\theta = \theta - \alpha_1 \nabla_\theta l(f_\theta; \mathcal{B}_t) - \alpha_2 \nabla_\theta l(f_\theta; \mathcal{R}_e) - \alpha_3 \nabla_\theta l(f_\theta; \mathcal{R}_t), \tag{6}$$

Algorithm 1 C-CMR: A Contextually-Cued Memory Recall Approach for Continual Lifelong Learning

1: **Input** Initialized f_θ, Memory $\mathcal{M}$, learning rate α.
2: **Variables s** : The forgetting degree vector for each memory exemplar in $\mathcal{M}$; $\mathcal{R}_e$: the example-interfered buffer with the most representative forgotten examples; $\mathcal{R}_t$: the memory exemplar buffer from the most forgotten tasks; $\mathcal{R}$: the total reply buffer with size $|\mathcal{R}| = |\mathcal{R}_e| + |\mathcal{R}_t|$.
3: **for** $t = 1 : T$ **do**
4: Batch $\mathcal{B}_t$ for task t is drawn from D_t.
5: $\theta' = \theta - \alpha\nabla_\theta l(f_\theta; \mathcal{B}_t)$ as introduced from Eq. (3).
6: **if** replay is true **then**
7: Develop **s** from θ' and θ as in Eq. (4).
8: Search the most interfered examples $(\mathbf{x}_m, y_m)$ with highest values of $s(m)$ into $\mathcal{R}_e$.
9: Develop $\mathcal{R}_t$, where for the j-th task, the number of examples q_j follows Eq. (5).
10: $\mathcal{R} = \mathcal{R}_e + \mathcal{R}_t$.
11: $\theta = \theta - \alpha\nabla_\theta l(f_\theta; (\mathcal{R} + \mathcal{B}_t))$. ⇒ Update continual learner θ with the training of both current task batch and replay batch.
12: **end if**
13: **end for**
14: Memory update $\mathcal{M}$ as introduced in Algorithm 2.

where $\alpha_1 = \frac{|\mathcal{B}_t|}{|\mathcal{B}_t+\mathcal{R}|}$, $\alpha_2 = \frac{|\mathcal{R}_e|}{|\mathcal{R}_e+\mathcal{R}|}$, and $\alpha_3 = \frac{|\mathcal{R}_t|}{|\mathcal{R}_t+\mathcal{R}|}$. Note that the latter two terms of loss function from C–CMR are still bounded by the learned task objective, which proves that the proposed AdaER algorithm follows the same convergence rule of the Experience Replay algorithm [17,37]. Specifically, in this paper, to ensure that the continual learner can learn the current task while replaying memories, we set the size of $\mathcal{R}$ and $\mathcal{B}_t$ to be the same. To further investigate the contributions of $\mathcal{R}_e$ and $\mathcal{R}_t$ in C–CMR, we introduce a new hyperparameter $\tau = |\mathcal{R}_e|/(|\mathcal{R}_e + \mathcal{R}_t|)$, and its impact is studied in detail in Section 5.

4.2. Entropy-balanced reservoir sampling

Meanwhile, the AdaER algorithm improves the memory update stage of ER-based methods by introducing the entropy-balanced reservoir sampling (E-BRS) method. This method aims to increase the diversity of training samples within the memory buffer $\mathcal{M}$ by maximizing its information entropy, which also addresses the issue of imbalanced data distribution.

In existing ER-based methods, random reservoir sampling is commonly used to select which training examples are stored in the fixed-size memory buffer $\mathcal{M}$. The probability of each training exemplar being represented in reservoir sampling is M/N, where N is the number of seen samples. However, when the continual learning data is imbalanced, with some classes having fewer samples stored in $\mathcal{M}$, the risk of decreasing information entropy arises.

To address this, the E-BRS method encourages a balanced number of training samples for each class. This is achieved by approximating the information entropy through balancing the number of samples per class in $\mathcal{M}$, as shown in Algorithm 2. This approximation reduces the computational overhead compared to estimating the information entropy using kernel functions. Additionally, E-BRS takes the criterion score $s(m)$ obtained from C–CMR into consideration. In Line 7, E-BRS finds the class $\tilde{y}$ with the largest number of examples. Then, the least important memory examples in class $\tilde{y}$ is removed in Line 8, E-BRS prevents the most forgotten memory example from being replaced. This ensures that important information is retained in the memory buffer during the update stage.

Algorithm 2 Development of M in AdaER: E-BRS

1: **Input:** Data pair $(\mathbf{x}, y)$ from $\mathcal{B}_t$, memory buffer $\mathcal{M}$, seen examples N.
2: **if** $M > N$ **then**
3: $\mathcal{M}[N] \leftarrow (\mathbf{x}, y)$.
4: **else**
5: valid = RandInt($[0, N]$).
6: **if** valid $\leq M$ **then**
7: $\tilde{y} = \arg\max_y$ Count $(y \in \mathcal{M})$: find the class $\tilde{y}$ with the largest number of examples in $\mathcal{M}$.
8: $m = \arg\min_m(s(m)|y_m = \tilde{y})$: remove the least important memory example within $\tilde{y}$.
9: $\mathcal{M}[m] = (\mathbf{x}, y)$
10: **end if**
11: **end if**
12: Updated memory buffer $\mathcal{M}$.

4.3. Discussions

In this paper, we propose the AdaER algorithm to improve the efficiency of replay-based continual learning baselines. AdaER consists of two stages: C–CMR and E-BRS, which address the limitations of existing replay approaches from the replay and the update stage, respectively. The C–CMR method provides guidance for the replay strategy by considering both data-conflicting and task-conflicting examples. It combines the example-interfered buffer $\mathcal{R}_e$ and the task-associated buffer $\mathcal{R}_t$ to determine which memories to replay. The goal is to address catastrophic forgetting by replaying the most interfered memories. We compare our C–CMR method with the existing maximally interfered retrieval (MIR) approach and show that MIR fails in certain boundary scenarios where $\tau = 1$. We provide a detailed performance analysis of AdaER and MIR in Section 5.

The E-BRS method improves the random reservoir sampling strategy used in the memory buffer updating process. Instead of estimating information entropy using kernel functions, we balance the number of samples per class in $\mathcal{M}$. This simplification avoids computational complexity concerns.

Note that to perform the memory forgetting degree analysis, the proposed AdaER compares the gradient different in Eq. (4) between the current model f_θ and the virtual model $f_{\theta'}$, where $f_{\theta'}$ requires additional memory storage. However, as we observed, the extra memory size is typically much smaller than the model training, e.g., the ResNet-18 [38] in our experiments takes 7.5 GB for training, while the size of the "State_dict" of $f_{\theta'}$ is no more than 100 MB, which is negligible. Overall, AdaER combines the C–CMR and E-BRS methods to enhance the replay-based continual learning process. We show that AdaER outperforms existing baselines, including MIR, and provide detailed performance analysis in Section 5.

5. Experiments

1. Add results with multiple random seeds for robustness analysis, followed by a paired t-test to show the significance of the results.

2. Show the results with more recent method (after 2020) and compare under mini-imagenet/tiny-imagenet settings //

3. Exploring additional ablation studies, such as assessing the impact of using individual components instead of both, as well as the consequences of utilizing a single reservoir type instead of both (Re vs. Rs),

5.1. Experimental setup

To evaluate our proposed AdaER continual learning algorithm, we compared it with recently proposed baselines under the class-IL

Table 1
Numerical Details of introduced benchmarks in this paper. For each benchmark, N_{task} is the number of tasks, N_c denotes the number of classes per task, and N_{train} represents the training data size per each task respectively.

Dataset	N_{train}	N_{task}	N_c
Split-MNIST [39]	1000	5	2
Split-FMNIST [40,41]	1200	5	2
Split-CIFAR10 [42]	10,000	5	2
Split-CIFAR100 [42]	1000	50	2
Split-TinyImageNet [43]	10,000	10	20

continual learning settings over several supervised benchmarks in [16, 18]. Note that we develop our code on the opensource continual learning platform.[1] Meanwhile, we follow the source code[2] to build up the mainly compared MIR method.

5.1.1. Benchmarks

In this work, we introduce a suite of benchmarks specifically tailored for the field of continual lifelong learning. Detailed descriptions of these benchmarks are provided in Table 1. We have selected split-MNIST, split-FMNIST, and split-CIFAR10 as cornerstone benchmarks to serve as the primary basis for performance evaluation in continual learning scenarios. Additionally, the split-CIFAR100 benchmark is employed to investigate the efficacy of our proposed AdaER algorithm in handling tasks with extended sequences. Meanwhile, the split-TinyImageNet benchmark allows us to assess the comparative performance of various algorithms on more complex tasks, using paired t-tests for statistical significance analysis.

5.1.2. Baselines

We compare the proposed AdaER algorithm with the following existing baselines in recent literature: **oEWC** [45], **SI** [28], **GEM** [21], **AGEM** [11], **iCaRL** [46], **ER** [17], **MIR** [18], **GSS**, **HAL** [47], and **ER-ACE** [48]. Moreover, we also evaluate the performance of the following two different settings. **Online:** the learner is trained under the continual learning setting by simply applying SGD optimizer. **Joint:** all tasks are trained jointly as one complete dataset instead of in a continual manner, which usually gives us an upper bound of the learning performance of all tasks.

5.1.3. Training

To provide a fair comparison with existing baselines, we train all the neural networks in this paper with the Stochastic Gradient Descent (SGD) optimizer. Additionally, all compared benchmarks in the paper are executed with the same computational resources. For MNIST and FMNIST benchmarks, we use a two-layer MLP with 400 hidden nodes, which follows the settings in [21,49]. For CIFAR-10 and CIFAR-100, we use a standard Resnet-18 [38] which is introduced in [46]. Note that for the replay-based methods, the batch size for $\mathcal{B}_t$ and $\mathcal{R}$ are both set to 20, and the memory buffer is set to 100 by default.

5.1.4. Metrics

In this paper, we measure the performance of continual learning algorithms with the following four metrics, which are defined in the literature [11,21]. Note that for the T tasks in a continual learning benchmark, we evaluate the test performance after learning each task. As such, we conduct a result matrix $R \in \mathbb{R}^{T\times T}$, where $R_{i,j}$ denotes the testing accuracy of the continual learner on task t_j after learning task t_i. Let $\bar{b}_i$ be the testing accuracy for each task after the random initialization of the learning model and F_i be the best testing accuracy for task t_i, the introduced four metrics are as follows.

- **AverageAccuracy** : $\text{Acc} = \frac{1}{T}\sum_{i=1}^{T} R_{T,i}$.
- **AverageForgetting** : $\text{Forget} = \frac{1}{T-1}\sum_{i=1}^{T-1} R_{T,i} - F_i$.
- **BackwardTransfer** : $\text{Bwt} = \frac{1}{T-1}\sum_{i=1}^{T-1} R_{T,i} - R_{i,i}$.
- **ForwardTransfer** : $\text{Fwt} = \frac{1}{T-1}\sum_{i=1}^{T-1} R_{i,i} - \bar{b}_i$.

Specifically, for Acc, BWT, and FWT metrics, the higher value indicates better continual learning performance, and the lower the better for the Forget metrics.

5.2. Results

Performance analysis. Table 2 shows the performance of compared continual learning baselines against three benchmarks, where the results are averaged by three times of training with different seeds, demonstrated with the four evaluation metrics. Note that for all memory-based approaches, the size of the memory buffer is 100, and the continual learner trains each incoming training batch only once during the learning process of each benchmark. We can notice that the proposed AdaER algorithm achieves the best overall performance against every compared method at each introduced evaluation metric.

Specifically, for split-MNIST and split-FMNIST, AdaER achieves 89.6% and 74.0% testing accuracy, which is 3.7% and 6.3% higher than the ER method respectively. For the split-CIFAR10 benchmark, AdaER obtains a positive backward transfer result as 4.4, while the Bwt of ER is only −19.9. Additionally, compared to the MIR method, the forgetting of the proposed AdaER for split-CIFAR10 is only 18.0, which is 28.0% lower. And for the forward transfer, we can notice that compared to GSS, the proposed AdaER achieves −6.78 on the split-FMNIST benchmark which is 69.5% higher.

Furthermore, we also notice several interesting phenomenons: (i) through GEM obtains feasible performance on split-MNIST and split-FMNIST, it performs poorly on split-CIFAR10, which may indicate GEM has limited ability against complex continual learning tasks; (ii) though the averaged testing accuracy of GSS is very close to MIR method, it has worse performance on backward transfer, especially on split-FMNIST and split-CIFAR10 benchmarks.

Impacts of memory capacity. We then study the importance of the memory buffer size by evaluating the compared methods under different values of $M \in [50, 200]$. Firstly, we investigate the performance of compared baselines under different M in terms of the testing accuracy, where the results are shown in Fig. 5. It can be noticed that as M increases, the performance of compared continual learning methods becomes better, and the testing accuracy of the proposed AdaER algorithm outperforms other baselines on all benchmarks. Furthermore, in split-MNIST, the testing accuracy of GEM increases by 49.9% when M increases from 50 to 200, while only 2.5% in AdaER. This indicates that the proposed AdaER algorithm is more robust against the memory capacity M, compared to other existing continual learning approaches.

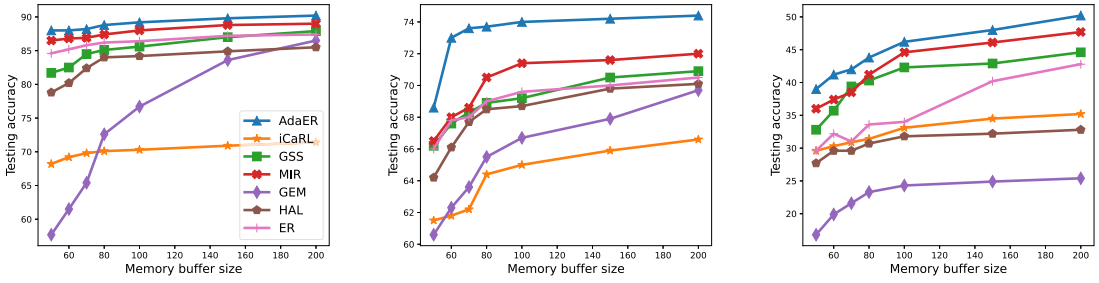
The results in Fig. 6 demonstrate the changes in backward transfer performance with different memory buffer sizes. Similar to the results in Fig. 5, as the size of $\mathcal{M}$ increases, the proposed AdaER consistently outperforms all compared continual learning baselines in terms of both backward transfer and robustness. For example, in split-CIFAR10, the backward transfer value of the proposed AdaER increases from −15.4 to 8.2, which outperforms other methods significantly.

Investigation of the first task. In Fig. 7, we study the change of the testing accuracy of the first task in each benchmark throughout the complete continual learning process, i.e., how the forgetting evolves with more tasks being learned. The results show that compared to all baselines, the proposed AdaER algorithm achieves the overall minimal forgetting of the performance of the first task for each benchmark. For example, the first task accuracy of AdaER decreases 3%, 10.6% and 23.4% corresponding to split-MNIST, split-FMNIST, and split-CIFAR10, which is 61.5%, 35.8% and 40% better than the results of MIR method in this work.

[1] {https://github.com/aimagelab/mammoth} [44].
[2] {https://github.com/aimagelab/mammoth} [18].

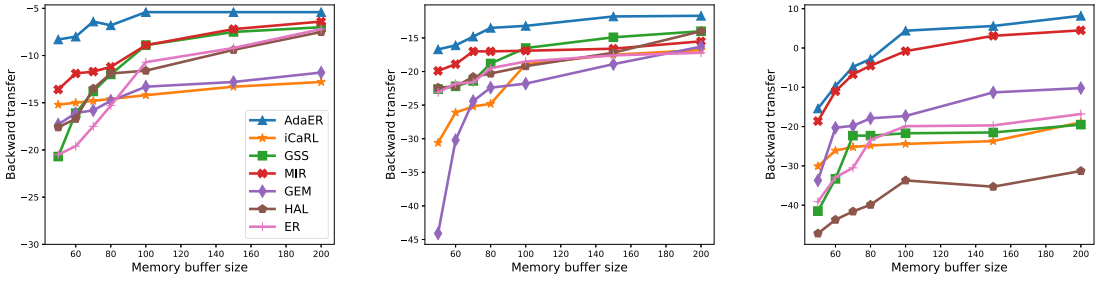
Table 2
Continual learning results for Split-MNIST, Split-FMNIST, and Split-CIFAR10. We report the mentioned four metrics: Acc (higher is better), Forget (lower is better), Bwt (higher is better), and Fwt (higher is better). Note that the results are spitted into different categories via the horizontal lines: the auxiliary joint and online baseline, the regularization-based methods, and the memory-based experience-replay baselines.

Method	Split-MNIST				Split-FMNIST				Split-CIFAR10			
	Acc ↑	Forget ↓	Bwt ↑	Fwt ↑	Acc ↑	Forget ↓	Bwt ↑	Fwt ↑	Acc ↑	Forget ↓	Bwt ↑	Fwt ↑
Online	18.4	98.4	−78.4	N/A	20.0	98.0	−78.6	N/A	13.0	83.6	−66.8	N/A
Joint	94.0	N/A	N/A	N/A	83.2	N/A	N/A	N/A	63.6	N/A	N/A	N/A
oEWC	19.8	98.9	−98.7	−14.4	20.0	98.6	−98.7	−14.5	17.7	78.4	−58.3	−12.9
SI	19.4	99.2	−98.8	−13.4	19.9	98.7	98.8	−13.3	15.2	83.8	−72.8	−12.7
ER	86.4	11.6	−10.7	−7.07	69.6	23.6	−18.5	−6.89	34.0	39.0	−19.9	−12.5
GEM	76.7	22.9	−13.3	−22.9	66.7	30.2	−21.8	−17.2	24.3	63.5	−17.3	−18.9
A-GEM	38.7	67.1	−57.0	−14.7	32.8	65.8	−70.5	−12.8	19.3	74.5	−43.7	−12.6
iCaRL	70.3	14.7	−14.2	N/A	65.0	29.5	−18.9	N/A	33.1	50.0	−24.4	N/A
HAL	84.2	19.4	−12.4	−11.6	68.7	19.2	−19.2	−19.3	31.8	43.8	−33.7	−12.9
GSS	85.6	14.3	−8.9	−10.7	69.2	22.7	−16.5	−22.3	42.3	29.6	−21.7	−13.3
MIR	88.0	9.0	−8.9	−7.07	71.4	22.4	−16.9	−6.89	44.6	25.0	−0.8	−12.5
C-CMR	88.4	7.2	−6.6	−7.01	73.0	18.5	−14.6	−6.87	45.4	22.6	3.5	−12.5
E-BRS	88.6	7.0	−6.0	−7.01	72.9	21.2	−14.8	−6.88	45.2	24.2	2.6	−12.5
AdaER	**89.6**	**6.6**	**−5.4**	**−6.90**	**74.0**	**18.0**	**−13.2**	**−6.78**	**46.2**	**18.0**	**4.4**	**−12.4**



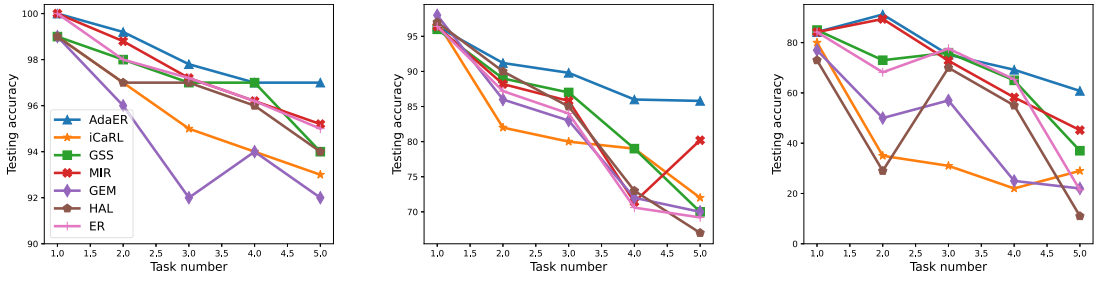
(a) split-MNIST (b) split-FMNIST (c) split-CIFAR10

Fig. 5. The impact of different memory size over averaged testing accuracy.



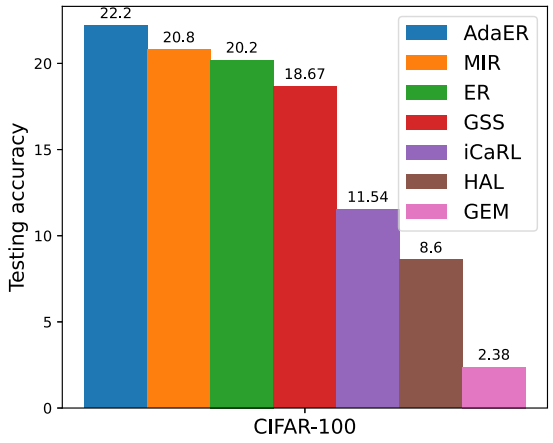
(a) split-MNIST (b) split-FMNIST (c) split-CIFAR10

Fig. 6. The impact of different memory size over backward transfer.

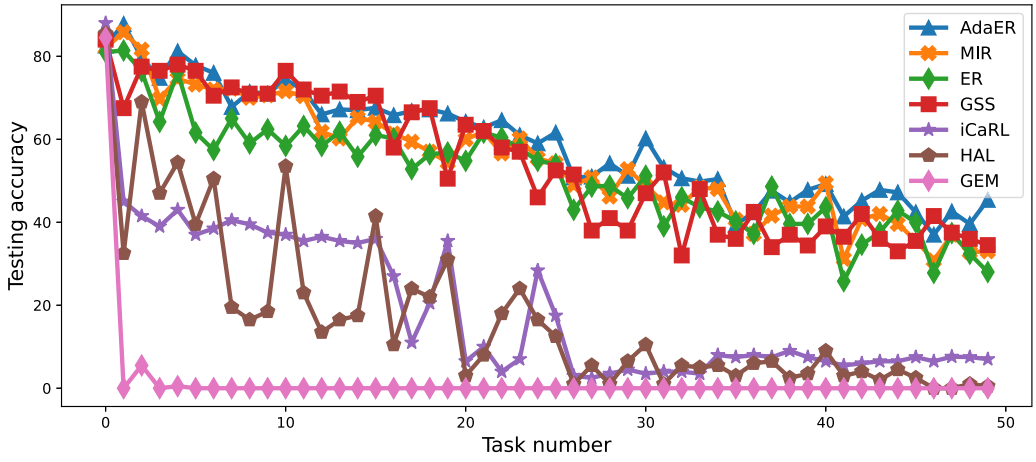


(a) split-MNIST (b) split-FMNIST (c) split-CIFAR10

Fig. 7. The testing accuracy of the first task as other tasks are learned.

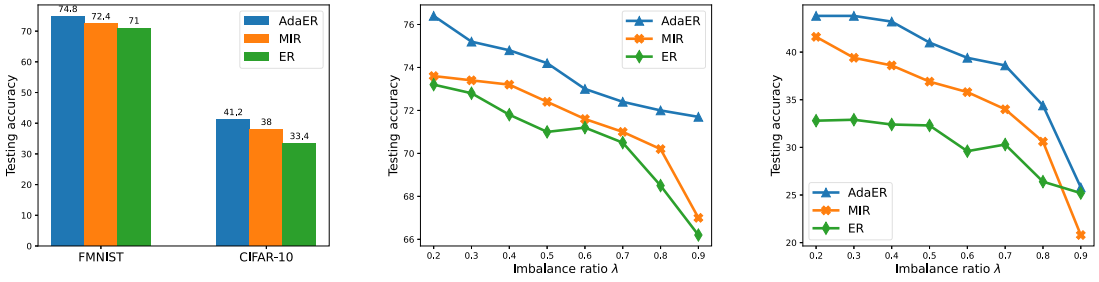


(a) Averaged testing accuracy



(b) Evolution of the first task

Fig. 8. Results of the compared baselines against split-CIFAR100 benchmark.



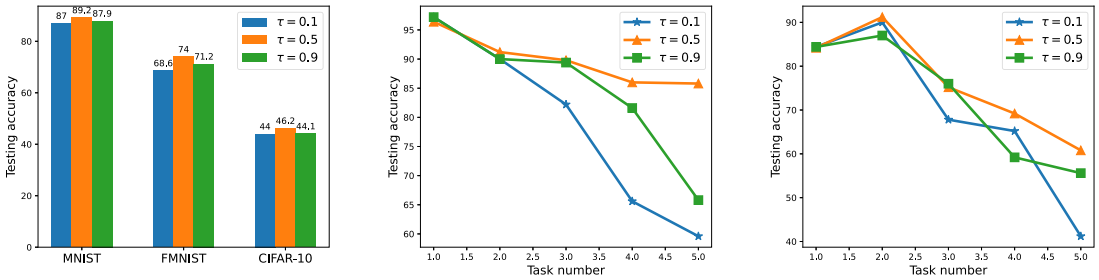
(a) Averaged testing accuracy (b) split-FMNIST (c) split-CIFAR10

Fig. 9. Performance of the compared approaches under different settings of imbalanced training dataset partition: when λ is larger, the imbalance degree is higher.

In addition, we also notice that the evolution of the first task is not monotonically decreasing. For example, the testing accuracy of AdaER after observing task 2 is significantly higher than task 1, which indicates that the proposed AdaER algorithm is with feasible transfer ability.

Study of long sequence task. We then test the performance of the proposed AdaER algorithm on continual learning tasks with longer sequences. To achieve this, we conduct the split-CIFAR100 benchmark, where the training data of 100 different labels are divided into 50 tasks, 2 classes per each task. Note that this benchmark is extremely challenging in our experimental settings. As such, different from other benchmarks that each training batch is seen by the learner only once during the learning process, we add the iterations per each batch to 5 for split-CIFAR100. The results in Fig. 8 show the performance of compared approaches against this benchmark, where the left one is the averaged testing accuracy and the right one illustrates the evaluation of testing accuracy of the first task.

It can be noticed from the results in Fig. 8(a) that GEM, iCaRL, and HAL methods perform poorly and the proposed AdaER outperforms ER,



(a) Averaged testing accuracy (b) split-FMNIST (c) split-CIFAR10

Fig. 10. Performance of the proposed AdaER algorithm under different settings of τ. (a): the averaged testing accuracy for each benchmark; (b): the evolution of first task testing accuracy on split-FMNIST; (c): the evolution of first task testing accuracy on split-CIFAR10.

Table 3
Summary statistics and paired t-test results.

Method	Mean±Std	t-Statistic	p-value
ER	17.52 ± 0.79	7.79	1.49e−05
MIR	17.58 ± 1.07	7.24	2.78e−05
ER-ACE	19.56 ± 0.77	5.31	3.41e−04
AdaER	$\mathbf{22.09 \pm 1.42}$	–	–

GSS, and MIR baselines against split-CIFAR100. Compared to ER with 20.2% testing accuracy, our AdaER algorithm achieves 22.2% which is 9.9% higher. Additionally, the first task evolution results in Fig. 8(b) also support our claims on GEM, iCaRL, and HAL methods, e.g., the testing accuracy of the first task in the GEM method drops rapidly to nearly zero after task 3. Specifically, while MIR and ER only have 33% and 29% final testing accuracy of the first task, the AdaER achieves 45.4%, which is 37.6% and 56.6% higher correspondingly. Besides, we can also notice that compared to ER, though the GSS method achieves a higher final first task testing accuracy, the averaged testing accuracy for every continual learning task in split-CIFAR100 is lower.

Study of large size task. We then evaluate the performance of the proposed AdaER algorithm with a more challenging benchmark, Split-TinyImageNet, where the training data of 200 labels are split into 10 tasks, 20 classes per each task without overlap. Specifically, the results in Table 3 are performed with 11 repetitions with different random seeds. In the context of this complex benchmark, the AdaER algorithm outperforms other methods on the averaged accuracy. For example, comparing to the ER method, which manages a mean accuracy of 17.52%, our AdaER achieves the best testing accuracy of 22.09%.

Meanwhile, the statistical validation through paired t-tests further solidifies our observations. When compared to the second best method ER-ACE, the t-statistic is 5.31 and the p-value is 3.41×10^{-4}, both of which indicate a statistically significant difference in performance. The same trend holds true when AdaER is compared with MIR and ER, reinforcing the algorithm's robustness and efficacy in continual learning tasks, particularly with larger sequences (see Fig. 12).

Impacts of λ. Then, we study the impact of imbalanced training data partition on the proposed AdaER algorithm. Note that for better evaluation, the performance of compared ER and MIR methods are also introduced for comparison. We setup the imbalanced data partition scenario on split-FMNIST and split-CIFAR10 benchmarks: for each task, the number of training sample difference between class 1 and class 2 over class 2 is set to λ.

We first provide the averaged testing accuracy results of compared methods in Fig. 9(a), where $\lambda = 0.5$. From the results, we can notice that the proposed AdaER algorithm outperforms both MIR and ER under the imbalanced scenario. Compared to the results in Table 2, all three introduced methods have a significant testing accuracy decrease against the split-CIFAR10 dataset. In this condition, we further evaluate this performance decrease and find that the proposed AdaER achieves minimal decline. The testing accuracy decline of AdaER is only 10.8% while 14.8% on MIR and 14.3% on ER, which supports our claims in Section 4.2.

Additionally, we investigate the performance changes of compared baselines with $\lambda \in [0.1, 0.9]$, of which the testing accuracy results are shown in Fig. 9(b) for split-FMNIST and 9(c) for split-CIFAR10. It can be noticed that as the value of λ increases, the performance of each compared method decreases significantly. However, the proposed AdaER algorithm also shows its robustness to the imbalanced data partition. For example, in split-FMNIST, the testing accuracy decline from $\lambda = 0.9$ to 0.1 is 6.2%, 9.0% and 9.6% for AdaER, MIR and ER respectively. And for split-CIFAR10, though the performance of AdaER drops faster than ER, it still outperforms ER over testing accuracy for each considered λ.

Impacts of τ. In this part, we evaluate the influence of the introduced hyper-parameter $\tau \in (0, 1)$ on the performance of the proposed AdaER algorithm. As illustrated in Section 4.1, τ denotes the weighted factor of task-associated buffer $\mathcal{R}_t$ and example-interfered buffer $\mathcal{R}_e$ that $\tau = p/|\mathcal{R}|$. Particularly, when $\tau = 0.5$, we consider q is equal to $0.5|\mathcal{R}|$. We first provide the averaged testing accuracy with different values of τ in Fig. 10(a), where the results show that when $\tau = 0.5$, the proposed AdaER algorithm achieves the best among other settings. For example, in split-FMNIST, when $\tau = 0.5$, the averaged testing accuracy of AdaER is 74.0%, which is 7.9% and 3.9% higher than $\tau = 0.1$ and $\tau = 0.9$ respectively.

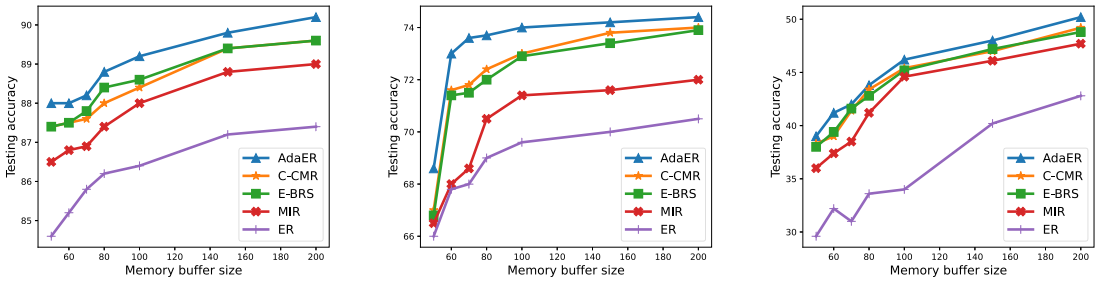
Additionally, we study the evolution of the first task testing accuracy with different values of τ on split-FMNIST and split-CIFAR10, whose corresponding results are shown in Figs. 10(b) and 10(c). Firstly, the results also support that when $\tau = 0.5$, the performance of AdaER is the best. Specifically, for example, in split-CIFAR10, the final testing accuracy of the first task when $\tau = 0.5$ is 60.8%, which is 47.6% higher against $\tau = 0.1$ and $9,4\%$ higher against $\tau = 0.9$.

Note that when $\tau = 0.1$ the AdaER algorithm is mainly impacted by the example-interfered buffer $\mathcal{R}_e$, while when $\tau = 0.9$ the AdaER algorithm is mainly impacted by the task-associated buffer $\mathcal{R}_t$. Hence, we can also obtain the observation that that compared to the performance of $\tau = 0.1$, the first task testing accuracy is more robust when $\tau = 0.9$, indicating that $\mathcal{R}_e$ is more powerful than $\mathcal{R}_t$.

5.3. Ablation study

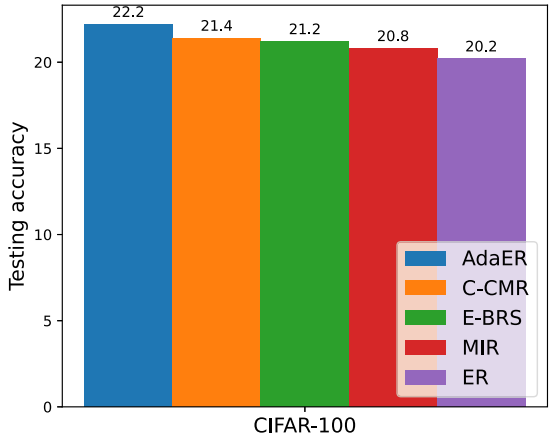
In this part, we provide the ablation study of the proposed AdaER algorithm, where the two methods C–CMR and E-BRS are evaluated independently. Particularly, we evaluate C–CMR via the proposed contextually cued recall method and the memory buffer is developed with a random reservoir sampling strategy. And for E-BRS we develop the memory buffer with entropy-balanced reservoir sampling with the MIR replay strategy. Note that for better presentation, the experimental results of ER and MIR methods are also introduced for comparison.

Study of memory buffer size M. We conduct the ablation study under the balanced training data partition setting as $\lambda = 0$. The

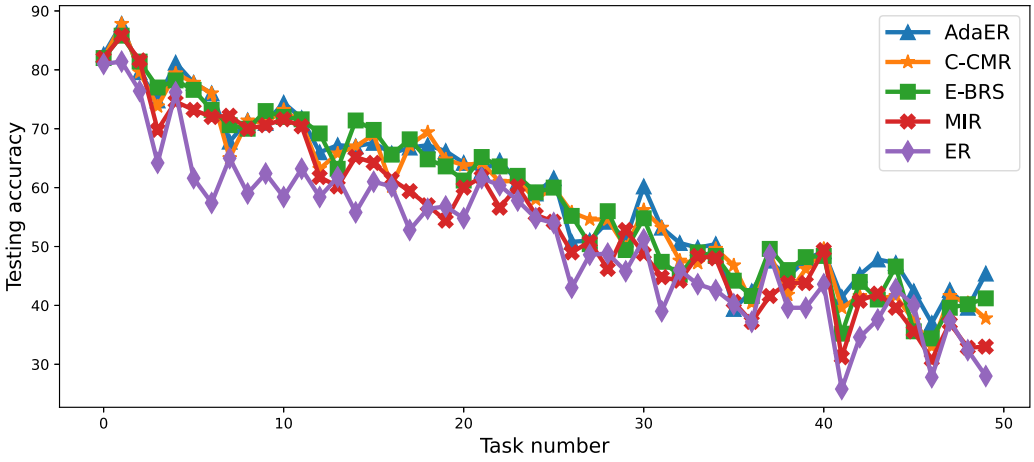


(a) split-MNIST (b) split-FMNIST (c) split-CIFAR10

Fig. 11. Ablation study of the proposed AdaER algorithm with the developed C–CMR and E-BRS methods: the performance of averaged testing accuracy as the increase of memory buffer size M.



(a) Averaged testing accuracy



(b) Evolution of the first task

Fig. 12. Ablation study of the proposed AdaER algorithm with the developed C–CMR and E-BRS methods over split-CIFAR100.

results of averaged testing accuracy for the compared methods under different memory buffer sizes are shown in Fig. 11. From the results, we can find that both C–CMR and E-BRS outperform MIR and ER at each benchmark. Interestingly, the performance of E-BRS is slightly better than C–CMR against split-MNIST while on the contrary against split-FMNIST. Particularly, as shown in Table 2, when $M = 100$, the averaged testing accuracy of C–CMR and E-BRS are 45.4% and 45.2, which are 33.5% and 32.9% higher than the ER method.

6. Conclusions and future work

The catastrophic forgetting problem is a long-standing challenge in the study of continual lifelong learning, especially in class-IL scenarios. While recently developed experience replay approaches have shown promising capability in mitigating this problem, their performance is still limited by its weakness of randomly sampling strategies on both the replay and update stages. As such, in this work, we propose the

adaptive experience replay (AdaER) algorithm, which improves the two stages of existing ER via two methods. For the replay stage, AdaER provides a novel contextually-cued recall strategy, which considered both the interfered examples and the associated tasks during continual learning process that guides which of the memory examples should be replayed. For the update stage, we develop entropy-balanced reservoir sampling (E-BRS), which improves the original reservoir sampling strategy by maximizing the information entropy of the memory buffer. The experimental results show that the proposed AdaER algorithm outperforms existing approaches against class-IL continual learning.

The AdaER algorithm presented in this paper offers considerable advancements in the catastrophic forgetting problem in Experience Replay based class-IL lifelong learning scenario. We believe it can be incorporated into existing machine learning models to enhance their ability to retain and utilize knowledge from earlier learning stages, thereby improving their overall performance and adaptability. Additionally, the contextually-cued recall strategy and entropy-balanced reservoir sampling can offer significant improvements to the process of continual machine learning model training.

There are several promising directions to explore in our future work. Firstly, the implicit relationships between the example-interfered buffer and the task-associated buffer could yield a more unified and effective approach, as opposed to treating them as mutually exclusive components in our current study. Secondly, although AdaER is currently tailored for class-IL continual learning, extending its efficacy to other paradigms such as task-IL and domain-IL warrants systematic evaluation. Thirdly, the entropy-balanced reservoir sampling strategy introduced in the paper also opens the door for the development of innovative entropy-centric strategies aimed at amplifying the computational efficiency of continual learning methodologies. Lastly, the study of memory forgetting degree needs additional exploration, aiming to reduce both computational burden and memory requirements, thereby obviating the need for supplementary virtual models.

CRediT authorship contribution statement

Xingyu Li: Methodology, Investigation, Software, Writing. **Bo Tang:** Conceptualization, Methodology, Supervision, Writing – review & editing. **Haifeng Li:** Conceptualization, Writing – review & editing.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

Acknowledgments

This research was partially funded by US National Science Foundation (NSF), Award IIS 2325863.

References

[1] A. Krizhevsky, I. Sutskever, G.E. Hinton, Imagenet classification with deep convolutional neural networks, Adv. Neural Inf. Process. Syst. 25 (2012) 1097–1105.
[2] Z. Qu, X. Li, R. Duan, Y. Liu, B. Tang, Z. Lu, Generalized federated learning via sharpness aware minimization, in: International Conference on Machine Learning, PMLR, 2022, pp. 18250–18280.
[3] Y. Zhou, X. Li, Y. Zhou, Y. Wang, Q. Hu, W. Wang, Deep collaborative multi-task network: A human decision process inspired model for hierarchical image classification, Pattern Recognit. 124 (2022) 108449.
[4] X. Li, Z. Qu, S. Zhao, B. Tang, Z. Lu, Y. Liu, Lomar: A local defense against poisoning attack on federated learning, IEEE Trans. Dependable Secure Comput. (2021).
[5] M. McCloskey, N.J. Cohen, Catastrophic interference in connectionist networks: The sequential learning problem, in: Psychology of Learning and Motivation, Vol. 24, Elsevier, 1989, pp. 109–165.
[6] I.J. Goodfellow, M. Mirza, D. Xiao, A. Courville, Y. Bengio, An empirical investigation of catastrophic forgetting in gradient-based neural networks, 2013, arXiv preprint arXiv:1312.6211.
[7] J. Kirkpatrick, R. Pascanu, N. Rabinowitz, J. Veness, G. Desjardins, A.A. Rusu, K. Milan, J. Quan, T. Ramalho, A. Grabska-Barwinska, et al., Overcoming catastrophic forgetting in neural networks, Proc. Natl. Acad. Sci. 114 (13) (2017) 3521–3526.
[8] C.V. Nguyen, Y. Li, T.D. Bui, R.E. Turner, Variational continual learning, in: International Conference on Learning Representations, 2018, URL https://openreview.net/forum?id=BkQqq0gRb.
[9] D. Oudiette, K.A. Paller, Upgrading the sleeping brain with targeted memory reactivation, Trends Cogn. Sci. 17 (3) (2013) 142–149.
[10] G.M. van de Ven, S. Trouche, C.G. McNamara, K. Allen, D. Dupret, Hippocampal offline reactivation consolidates recently formed cell assembly patterns during sharp wave-ripples, Neuron 92 (5) (2016) 968–974.
[11] A. Chaudhry, M. Rohrbach, M. Elhoseiny, T. Ajanthan, P.K. Dokania, P.H. Torr, M. Ranzato, On tiny episodic memories in continual learning, 2019, arXiv preprint arXiv:1902.10486.
[12] H. Shin, J.K. Lee, J. Kim, J. Kim, Continual learning with deep generative replay, Adv. Neural Inf. Process. Syst. 30 (2017).
[13] Y. Yang, B. Kiumarsi, H. Modares, C. Xu, Model-free λ-policy iteration for discrete-time linear quadratic regulation, IEEE Trans. Neural Netw. Learn. Syst. (2021).
[14] Y. Yang, H. Modares, K.G. Vamvoudakis, F.L. Lewis, Cooperative finitely excited learning for dynamical games, IEEE Trans. Cybern. (2023).
[15] Y. Yang, Y. Pan, C.-Z. Xu, D.C. Wunsch, Hamiltonian-driven adaptive dynamic programming with efficient experience replay, IEEE Trans. Neural Netw. Learn. Syst. (2022).
[16] G.M. Van de Ven, A.S. Tolias, Three scenarios for continual learning, 2019, arXiv preprint arXiv:1904.07734.
[17] D. Rolnick, A. Ahuja, J. Schwarz, T. Lillicrap, G. Wayne, Experience replay for continual learning, Adv. Neural Inf. Process. Syst. 32 (2019).
[18] R. Aljundi, E. Belilovsky, T. Tuytelaars, L. Charlin, M. Caccia, M. Lin, L. Page-Caccia, Online continual learning with maximal interfered retrieval, in: H. Wallach, H. Larochelle, A. Beygelzimer, F. d' Alché-Buc, E. Fox, R. Garnett (Eds.), Advances in Neural Information Processing Systems 32, Curran Associates, Inc., 2019, pp. 11849–11860.
[19] G.E. Hinton, D.C. Plaut, Using fast weights to deblur old memories, in: Proceedings of the Ninth Annual Conference of the Cognitive Science Society, 1987, pp. 177–186.
[20] A. Gepperth, B. Hammer, Incremental learning algorithms and applications, in: European Symposium on Artificial Neural Networks (ESANN), 2016.
[21] D. Lopez-Paz, M. Ranzato, Gradient episodic memory for continual learning, Adv. Neural Inf. Process. Syst. 30 (2017).
[22] J.S. Vitter, Random sampling with a reservoir, ACM Trans. Math. Softw. 11 (1) (1985) 37–57.
[23] Q. Lao, M. Mortazavi, M. Tahaei, F. Dutil, T. Fevens, M. Havaei, Focl: Feature-oriented continual learning for generative models, Pattern Recognit. 120 (2021) 108127.
[24] C. Zhuang, S. Huang, G. Cheng, J. Ning, Multi-criteria selection of rehearsal samples for continual learning, Pattern Recognit. 132 (2022) 108907.
[25] V.E. Martins, A. Cano, S.B. Junior, Meta-learning for dynamic tuning of active learning on stream classification, Pattern Recognit. 138 (2023) 109359.
[26] Y. Qiu, Y. Shen, Z. Sun, Y. Zheng, X. Chang, W. Zheng, R. Wang, SATS: Self-attention transfer for continual semantic segmentation, Pattern Recognit. 138 (2023) 109383.
[27] Z. Li, D. Hoiem, Learning without forgetting, IEEE Trans. Pattern Anal. Mach. Intell. 40 (12) (2017) 2935–2947.
[28] F. Zenke, B. Poole, S. Ganguli, Continual learning through synaptic intelligence, in: International Conference on Machine Learning, PMLR, 2017, pp. 3987–3995.
[29] F. Mao, W. Weng, M. Pratama, E.Y.K. Yee, Continual learning via inter-task synaptic mapping, Knowl.-Based Syst. 222 (2021) 106947.
[30] W. Sun, Q. Li, J. Zhang, D. Wang, W. Wang, Y.-a. Geng, Exemplar-free class incremental learning via discriminative and comparable parallel one-class classifiers, Pattern Recognit. 140 (2023) 109561.
[31] Y. Yao, G. Doretto, Boosting for transfer learning with multiple sources, in: 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, IEEE, 2010, pp. 1855–1862.
[32] A.A. Rusu, N.C. Rabinowitz, G. Desjardins, H. Soyer, J. Kirkpatrick, K. Kavukcuoglu, R. Pascanu, R. Hadsell, Progressive neural networks, 2016, arXiv preprint arXiv:1606.04671.
[33] J. Yoon, E. Yang, J. Lee, S.J. Hwang, Lifelong learning with dynamically expandable networks, in: International Conference on Learning Representations, 2018, URL https://openreview.net/forum?id=Sk7KsfW0-.
[34] P. Buzzega, M. Boschini, A. Porrello, D. Abati, S. Calderara, Dark experience for general continual learning: a strong, simple baseline, 2020, arXiv preprint arXiv:2004.07211.

[35] G.I. Parisi, R. Kemker, J.L. Part, C. Kanan, S. Wermter, Continual lifelong learning with neural networks: A review, Neural Netw. 113 (2019) 54–71.
[36] G.M. van de Ven, H.T. Siegelmann, A.S. Tolias, Brain-inspired replay for continual learning with artificial neural networks, Nat. Commun. 11 (1) (2020) 1–14.
[37] S. Han, Y. Kim, T. Cho, J. Lee, On the convergence of continual learning with adaptive methods, in: R.J. Evans, I. Shpitser (Eds.), Proceedings of the Thirty-Ninth Conference on Uncertainty in Artificial Intelligence, in: Proceedings of Machine Learning Research, Vol. 216, PMLR, 2023, pp. 809–818, URL https://proceedings.mlr.press/v216/han23a.html.
[38] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 770–778.
[39] Y. LeCun, L. Bottou, Y. Bengio, P. Haffner, et al., Gradient-based learning applied to document recognition, Proc. IEEE 86 (11) (1998) 2278–2324.
[40] H. Xiao, K. Rasul, R. Vollgraf, Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms, 2017, arXiv preprint arXiv:1708.07747.
[41] S. Farquhar, Y. Gal, Towards robust evaluations of continual learning, 2018, arXiv preprint arXiv:1805.09733.
[42] A. Krizhevsky, G. Hinton, et al., Learning multiple layers of features from tiny images, 2009.
[43] Y. Le, X.S. Yang, Tiny ImageNet visual recognition challenge, 2015, URL https://api.semanticscholar.org/CorpusID:16664790.
[44] M. Boschini, L. Bonicelli, P. Buzzega, A. Porrello, S. Calderara, Class-incremental continual learning into the extended DER-verse, IEEE Trans. Pattern Anal. Mach. Intell. (2022).
[45] J. Schwarz, W. Czarnecki, J. Luketina, A. Grabska-Barwinska, Y.W. Teh, R. Pascanu, R. Hadsell, Progress & compress: A scalable framework for continual learning, in: International Conference on Machine Learning, PMLR, 2018, pp. 4528–4537.
[46] S.-A. Rebuffi, A. Kolesnikov, G. Sperl, C.H. Lampert, Icarl: Incremental classifier and representation learning, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 2001–2010.
[47] A. Chaudhry, A. Gordo, P.K. Dokania, P. Torr, D. Lopez-Paz, Using hindsight to anchor past knowledge in continual learning, arXiv preprint arXiv:2002.08165 2 (7) (2020).
[48] L. Caccia, R. Aljundi, N. Asadi, T. Tuytelaars, J. Pineau, E. Belilovsky, New insights on reducing abrupt representation change in online continual learning, in: International Conference on Learning Representations, 2022, URL https://openreview.net/forum?id=N8MaByOzUfb.
[49] M. Riemer, I. Cases, R. Ajemian, M. Liu, I. Rish, Y. Tu, G. Tesauro, Learning to learn without forgetting by maximizing transfer and minimizing interference, 2018, arXiv preprint arXiv:1810.11910.

Xingyu Li received the B.S. degree from the School of Electronic Science and Engineering (National Model Microelectronics College), Xiamen University, Xiamen, China, in 2015, and the M.S. degree from the Department of Electrical and Computer Engineering, Stevens Institute of Technology, Hoboken, NJ, USA, in 2017. He is currently pursuing the Ph.D. degree with the Department of Electrical and Computer Engineering, Mississippi State University, MS, MS, USA. His current research interests include federated learning and continual learning.

Bo Tang is an Assistant Professor in the Department of Electrical and Computer Engineering at Mississippi State University. He received the Ph.D. degree in electrical engineering from University of Rhode Island (Kingstown, RI) in 2016. From 2016 to 2017, he worked as an Assistant Professor in the Department of Computer Science at Hofstra University, Hempstead, NY. His research interests lie in the general areas of statistical machine learning and data mining, as well as their various applications in cyber–physical systems, including robotics, autonomous driving and remote sensing.

Haifeng Li received his Master degree in Transportation Engineering from South China University of Technology, Guangzhou, China, in 2005, and Ph.D. degree in Photogrammetry and Remote Sensing from Wuhan University, Wuhan, China, in 2009. He was a Research Associate with the Department of Land Surveying and Geo-Informatics, The Hong Kong Polytechnic University, Hong Kong, in 2011, and a Visiting Scholar with the University of Illinois at Urbana-Champaign, Urbana, IL, USA, from 2013 to 2014. He is currently a Professor with the School of Geosciences and Info-Physics, Central South University, Changsha, China. He has authored over 30 journal articles. His current research interests include geo/remote sensing big data, machine/deep learning, and artificial/brain-inspired intelligence.