Image and Vision Computing 151 (2024) 105288

Contents lists available at ScienceDirect

Image and Vision Computing

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

Utilizing Inherent Bias for Memory Efficient Continual Learning: A Simple and Robust Baseline

Neela Rahimi *, Ming Shao

College of Engineering, University of Massachusetts, Dartmouth, USA

ARTICLE INFO

Keywords:
Online continual learning
Bias-robust model
Feature similarity
Memory footprint
Forgetting
Exemplar-free variation
Scalable learning framework
Knowledge retention

ABSTRACT

Learning from continuously evolving data is critical in real-world applications. This type of learning, known as Continual Learning (CL), aims to assimilate new information without compromising performance on prior knowledge. However, learning new information leads to a bias in the network towards recent observations, resulting in a phenomenon known as catastrophic forgetting. The complexity increases in Online Continual Learning (OCL) scenarios where models are allowed only a single pass over data. Existing OCL approaches that rely on replaying exemplar sets are not only memory-intensive when it comes to large-scale datasets but also raise security concerns. While recent dynamic network models address memory concerns, they often present computationally demanding, over-parameterized solutions with limited generalizability. To address this long-standing problem, we propose a novel OCL approach termed "Bias Robust online Continual Learning (BRCL)." BRCL retains all intermediate models generated. These models inherently exhibit a preference for recently learned classes. To leverage this property for enhanced performance, we devise a strategy we describe as 'utilizing bias to counteract bias.' This method involves the development of an Inference function that capitalizes on the inherent biases of each model towards the recent tasks. Furthermore, we integrate a model consolidation technique that aligns the first layers of these models, particularly focusing on similar feature representations. This process effectively reduces the memory requirement, ensuring a low memory footprint. Despite the simplicity of the methodology to guarantee expandability to various frameworks, extensive experiments reveal a notable performance edge over leading methods on key benchmarks, getting continual learning closer to matching offline training. (Source code will be made publicly available upon the publication of this paper.)

1. Introduction

Continual Learning (CL) has emerged as a pivotal paradigm in machine learning, aiming to enable models to learn from data over time without forgetting previously acquired knowledge [1,2]. However, a persistent challenge in CL is a phenomenon known as *catastrophic forgetting* [3], where models tend to overwrite old knowledge as they acquire new information. To address this, recent efforts, including softmax separation [4], bias correction [5], and knowledge distillation [1,2,6–8] have been developed to recalibrate the bias dynamics between older and newer classes. More competitive performance is shown by the approaches that consider updating the model by a balanced set of old and new tasks [9,10]. The landscape of CL has been evolving towards towards online settings [9,11,12] where models learn all information in a single pass through the data, referred to as **Online Continual Learning (OCL)** [13]. In OCL, it is imperative that models assimilate new information efficiently in one pass without the luxury of revisiting old data. Conventional CL approaches are not suitable in online scenarios, as they typically involve multiple passes over the data for model update [14] or buffer update to incorporate more representative exemplars, ensuring distinct decision boundaries after successive updates [4,15,16]. Such practices inherently demand the retention of prior training datasets, making them memory-draining and impractical in contexts with data privacy constraints [17].

The challenge intensifies as models encounter new training data. When the influx from newer tasks significantly outnumbers that of previous tasks [18], the model's knowledge tends to skew towards recent information [2,19,20]. This phenomenon is particularly pronounced when the data from newer tasks significantly outnumbers that of previous tasks [5], causing the model's knowledge to skew towards recent information [2,19,20]. One of the primary reasons is as models evolve to handle new tasks, implicit modifications in model weights can

* Corresponding author.
E-mail addresses: nrahimi@umassd.edu (N. Rahimi), mshao@umassd.edu (M. Shao).

https://doi.org/10.1016/j.imavis.2024.105288
Received 8 April 2024; Received in revised form 18 September 2024; Accepted 23 September 2024
Available online 27 September 2024

compromise the knowledge of older tasks [21], introducing a bias towards newer classes and exacerbating the forgetting issue [22]. A deeper investigation by [13] indicates that the **fully connected layers**, not the feature extractor, exhibit this bias. With this insight, we hypothesize **(i)** the bias in penultimate layers can direct the classification process; while most efforts focused on alleviating the bias, we propose to take advantage of existing bias which we call *bias recency* **(ii)** In this direction, retaining information from fully connected layers might offer a promising avenue for preserving knowledge.

Recent feature replay-based CL including DER [23], FOSTER [24], and MEMO [25] leveraging backbone extension have offered strategies that significantly improve performance at the expense of computational efficiency. Generative feature replay, on the other hand, produces data of old classes in CL procedure [26,27]. However, they present solutions with limited generalizability, especially on large-scale datasets [17]. We argue that constantly adapting a single network for generative replays fails to sustain OCL in the long run. This limitation is highlighted by recent research demonstrating that deep learning models, gradually lose their ability to adapt to new tasks, when applied in continual learning settings, As a result, the network's plasticity deteriorates over time, causing it to perform no better than shallow networks, unless variability is continually injected into the network [28]. This insight suggests that instead of relying on a single backbone network, employing dynamic networks could offer a more robust solution, especially in large-scale continual learning scenarios.

We argue that the bias inherited from each network indicates its preference for recent tasks. We sought to develop a method that draws on these insights and presents the *Bias Robust online Continual Learning* (BRCL) framework in this paper. In our exploration, we first highlight how the inherent bias in a network's backbone can serve as a unique metric for model selection (Fig.1). We term this strategy "using bias against bias" enabling the inference function to pinpoint the most discriminative network. Second, since storing all networks along the CL procedure can strain memory constraints, we propose a model consolidation approach to store a subset of networks. Our evaluations of BRCL on popular CL benchmarks in an online setting have yielded impressive performance, particularly with large-scale datasets such as ImageNet-1 K. The contribution of this paper can be summarized as:

- We propose a novel BRCL framework to address the pervasive catastrophic forgetting problem in OCL.
- We develop an inference function leveraging the inherent bias in CL networks.
- The integration of an efficient consolidator allows the framework to further reduce the memory footprint.
- In contrast to SoTA works in dynamic networks, our purpose is to introduce a simple framework that maintains expandability to various dataset sizes and models.

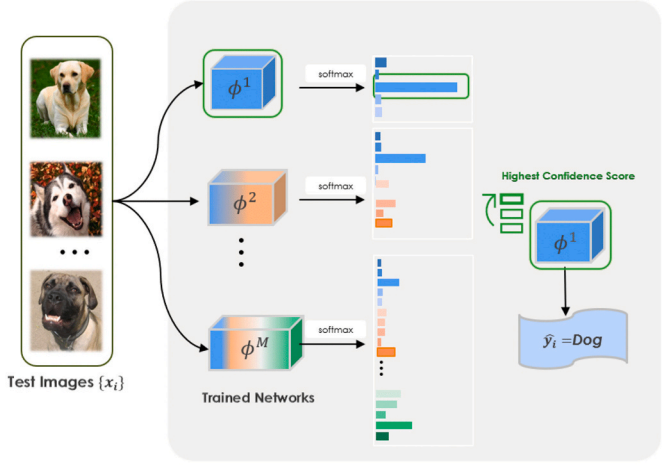

Fig. 1. Overview of the proposed BRCL in runtime. BRCL reserves a set of trained networks $\phi^1, \phi^2, \ldots, \phi^M$ obtained in the incremental training stage. When test data arrives, the Inference function evaluates the confidence score to identify the competent feature extractor (ϕ^1 in this example). In our method, feature representations rather than raw data are stored, which reduces memory footprint substantially, especially for large-scale datasets.

2. Related work

Rehearsal-based strategy. Methods in this category use a memory buffer for exemplar storage during training. However, a *fixed-size memory* buffer quickly fills with large-scale data, leaving fewer samples for older classes and causing class imbalance [1,11,12,29–32]. Some newer methods allow memory update but may still exclude older classes [9,20]. These methods often neglect task specificity, especially with random sampling updates [11,12,20]. Zhang et al. [33] proposed a Bayesian framework that incorporates full experience replay, and utilizes a sparse network approach, to manage memory usage. Additionally, adjusting model parameters or stored exemplars [1,15,34] requires revisiting older data, by continuous resampling [35] making this strategy non-optimal for online data processing.

Regularization-based methods. These algorithms impose restrictions when updating weights in a neural network by approaches including penalizing existing loss functions [36–38] or introducing new ones to the algorithm, such as *cross-distilled loss* in [1], and *less-forgetting loss* in [7]. Other existing methods apply gradient modification during optimization [39,40], and employ additional layers, e.g., another softmax for old classes [4] or a linear layer for bias removal [5]. Some studies have incorporated knowledge distillation [1,8,15] and attention-based distillation [41] to tackle catastrophic forgetting issues and imbalance problems in CL. Studies by [42,43] argued an inherent limitation in regularization approaches and proved that they could not learn the correct solution without task label inferences.

Feature replay. Feature replay methods [44–48] have made significant strides in enhancing performance with limited memory buffers. These methods, while innovative, often skirt the strict protocols of online learning. For instance, FOSTER [24] and MEMO [25] rely on retaining input data, which can be infeasible in real-time or privacy-sensitive scenarios. Generative feature replay [26,27] leverages generative models to simulate features, thus reducing memory needs. However, its efficacy hinges on the generative model's quality. Although several follow-up works [49–52] persist in advancing this domain, they confront enduring challenges. Among these, the representation drift and the generation of high-fidelity samples across a broad spectrum of tasks stand out as particularly persisting obstacle, specially when it comes to scaling to large-scale datasets.

3. Methodology

3.1. Background

The OCL problem discussed in this paper is restricted to one single pass over the data. The model shall be trained continually in a dataset $D = \{(x,y)|x \in X, y \in Y\}$, where $x \in X$ and $y \in Y$ is the image and its label. Training occurs post-completion of each task, this involves several mini-batches of data $\{(x,y)\}$, the model waits for task to conclude before updating. We strictly follow the online class incremental learning setup [13] where and task labels or any task-indicating strategy is not provided, i.e., *single-head* setting is applied. More specifically, given a data batch $\mathscr{D} = \{D_1, D_2, \ldots, D_M\}$ under M unknown distributions, our aim is to incrementally learn and optimize the network $\phi(x;\theta)$ parameterized by θ for data seen so far. We follow the same local i.i.d. assumption that the i-th task distribution for D_i is stationary. Therefore, given a mini-batch $\{(x^i, y^i)\} \in D^i$, OCL is defined as:

$$\phi^i \leftarrow \phi^{i-1}, D^i \cup \varepsilon^{1:i-1}; \tag{1}$$

where ε^i is a small exemplar set sampled from D^i. The utilization of exemplar set in the memory does not circumvent the one-pass requirement as long as this exemplar set is not updated after the model has seen the task. Exemplars are selected and retained during the initial pass through the data for a given task, and stored in memory, for future training stages. This approach avoids additional passes over the training data and ensures compliance with the OCL protocol [13].

3.2. Exemplar selection

The exemplar selection employs the herding algorithm [53], as utilized in iCaRL [15], while adapting it for an online continual learning setting. Unlike the original iCaRL method where exemplar sets can be updated after training on each class, in our approach, once an exemplar set for a class is determined, it remains fixed. Hence, $h(.)$ is the function that selects η *exemplars* from the set of feature vectors $\mathbf{F}$ based on their proximity to the *mean* of feature vectors f, formulated as $\mu_j = \frac{1}{|\mathbf{F}_j|}\sum_{f\in\mathbf{F}_j} f$ for each class j.

$$\varepsilon_j = h\left(\mathbf{F}_j, \mu_j, \eta\right). \tag{2}$$

Note that we use superscripts to index different tasks, and subscript to denote classes.

3.3. Network generation

If we store an instance of the frozen model ϕ^i after training on D^i, and we repeat this step every time we learn a new task, we can incrementally learn M networks $\Phi = \{\phi^i\}, i \in [1, M]$, where each ϕ^i rely solely on ϕ^{i-1} and exemplars seen so far $\varepsilon^{1:i-1}$. Algorithm 1 shows the procedure to produce the network set Φ.

Algorithm 1. Training and validation of BRCL.

Input: Data batch $\mathcal{D}$, an initial model $\phi^0(\cdot;\theta)$.
Output: Network set $\Phi = \{\phi^i\}$.

```
for i = 1 to M do
    ⟨φ^i⟩ ← ⟨φ^{i−1}, D^i ∪ ε^{1:i−1}⟩
    Update network set Φ ← Φ ∪ {φ^i}
    for each testing batch x ∈ X do
        Compute softmax scores s_c^i by Eq. (3)
        Select the model φ* ← I_f(x, Φ) using Eq. (4)
        Compute accuracy by selected φ*(x)
    end for
end for
```

3.4. Bias-robust framework

The proposed BRCL is motivated by two well-recognized phenomena in OCL, i.e., **knowledge drift** and **weak generalization** due to depending on a single model.

Knowledge Drift. In OCL, the imbalance data problem occurs at each training stage when the majority class from D^i has abundant data points to shift the weights towards the new class, while old classes have the minority data samples from $D^{1:i-1}$. This phenomenon is highlighted in Fig. 2 which shows the softmax scores of four networks after the model was incrementally trained on four tasks and tested on samples from Task 1. We can see that the softmax scores drop by a large margin after only four tasks. In particular, ϕ^1 presents the highest score over all the other models. The reason is the imbalance problem arises during training: when data from new classes significantly overshadows that of older ones, resulting in (1) penalizing logits associated with older classes, including their *bias* term in the softmax layer [4] (2) overwhelming decision boundary of minority classes by majority classes with more discriminative margins [18]. Fig. 3 visualize this problem by showing the transformation of the feature space for Task 1 as the model undergoes training on subsequent tasks. The progressive blurring of class boundaries in the feature space is a manifestation of the knowledge drift issue, highlighting the bias recency in the continual learning framework, where the model's updates are dominated by newer classes, leading to a degradation of the feature representations for earlier classes. Please note that Figs. 2 and 3 are independent visualizations as they have different numbers of classes in each task.

Weak generability. While Φ is trained sequentially with the incoming data, existing OCL approaches often rely on a single feature embedding or network for all tasks. A single model may struggle to capture complex representations or adequately replay features across diverse classes, leading to compromised performance. Fig. 3 illustrates how the compact and well-separated clusters associated with robust class representations generated by ϕ^1 become increasingly interspersed and less defined in subsequent models. The overlap of features between different classes can result in increased confusion for the model, suggesting a weakening of the model's generalization ability for previously learned classes as new information is incorporated. This becomes problematic when applied to large-scale datasets.

To handle the mentioned complications, we propose utilizing all networks trained incrementally during runtime. Given that each network is inherently biased towards its most recent task, examining these biased scores could lead us to an optimal network for a given test batch. As evidenced by Fig. 2, we hypothesize that the score corresponding to the correct label could be used as an inference, particularly if that label is associated with the most recent task. We can formulate the inference function framework to optimize the network selection, which consists of:

- A set of networks $\Phi = \{\phi^i\}$, where each ϕ^i produces a distinct feature map tailored to its corresponding dataset D^i.
- The inference function $\mathcal{I}_m$ designed to select the most suitable networks from Φ based on our proposed metric.

As outlined in Section 3.1, the first module can be constructed by storing the set of networks, Φ Fig. 1. The following section will delve into the architecture of the inference framework and the metrics used for model selection. As indicated in Fig. 2, while a bias towards recent tasks can cause performance deterioration of older tasks in a single network scenario, we argue that it can make a reliable metric for task inference using the associated network. If a test sample's correct label is part of a recent task, its softmax score is likely to be the highest by the network that was just trained on it. We exploit this behavior to gauge the confidence level of each model, thereby selecting the most reliable one for a given task. To ensure that the confidence levels indicated by the softmax scores accurately reflect the certainty of the predictions, we employ temperature scaling, a post-hoc calibrature technique that adjusts the softmax scores without altering the models' accuracy. Temperature scaling, as outlined by Guo et al. [54], involves dividing the logits by a temperature parameter T before applying the softmax function. This approach effectively calibrates the confidence levels of the model's predictions, making them more representative of the true likelihood of the correctness. The modified softmax score vector for a class $c \in C$ in network ϕ^i is thus defined as:

$$s_c^i = \text{Softmax}\left(\phi^i(x;\theta)/T\right); \tag{3}$$

where $\text{Softmax}(.) : \mathscr{R}^{d^i} \rightarrow [0,1]^{d^i}$ and $T > 1$ is the temperature parameter, with $T = 1$ recovering the original softmax probabilities. Parameter

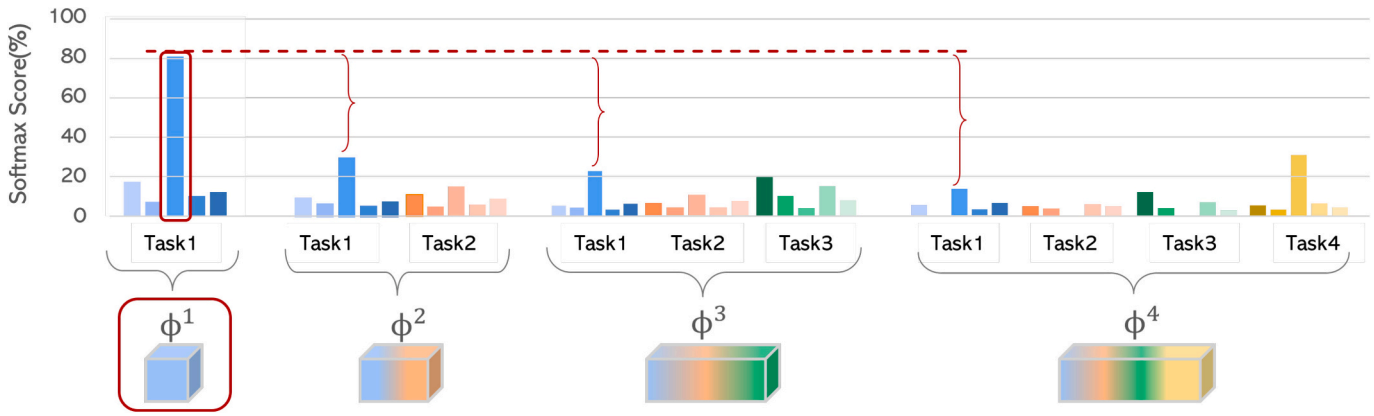

Fig. 2. Illustration of bias in $\phi^1 : \phi^4$. x-axis shows networks and tasks addressed by each network, and y-axis is the average **softmax score** by $\phi^1 : \phi^4$, given a batch of n test samples. The 3rd class in Task 1 is the ground truth. In the test, when ϕ^1 is selected, the highest softmax score comes from the 3rd class as expected, which will be selected for classification purposes. This advantage has been amplified due to the bias of ϕ^1 towards Task 1. In $\phi^2 : \phi^4$ the bias towards other tasks will diminish the score and mislead classifiers.

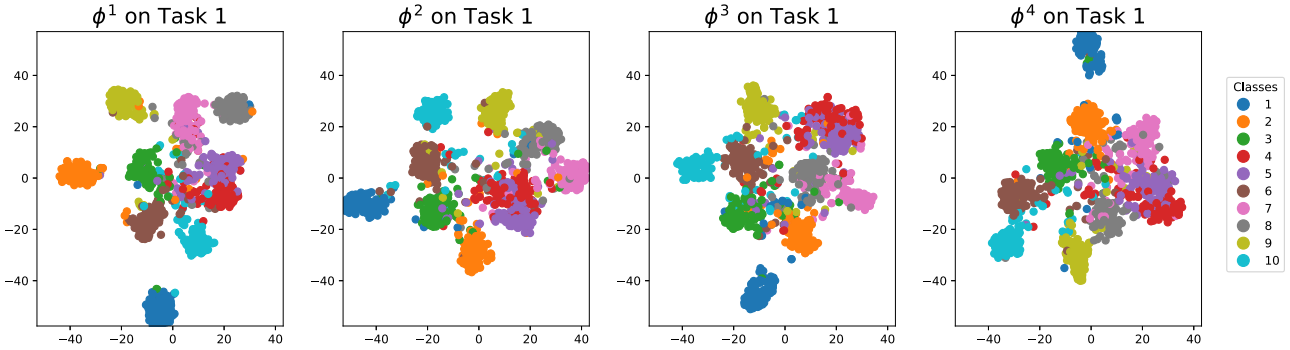

Fig. 3. Visualization of saturation of feature space related to *Task*1 across continual learning steps. Each subplot illustrates the 2D-tSNE embedding of *Task*1 features extracted from the models ϕ^i after training on Task i incrementally, $i \in \{1, 2, 3, 4\}$. It is observed that as the model is evolving, the originall distinct class clusters become progressively less discernible.

T softens the softmax output by increasing its entropy aiming to distribute the confidence levels more evenly across the classes. As T increases beyond 1, the model's output probabilities edge towards a state of uniform uncertainty. As T approaches infinity ($T \to \infty$), the output probabilities trend towards maximum uncertainty wherein each class's probability moves closer to being $1/|C|$, where $|C|$ is the number of classes. On the opposite end, lowering T closer to 0 would theoretically sharpen the distribution, converging on absolute certainty for a single class, although such a scenario is avoided by keeping T greater than 1. Value of T is found empirically. This approach ensures that while the confidence across classes becomes more balanced, the model's predictive accuracy remains unaffected since the adjustment does not alter the class with the highest softmax score.

We define the inference function $\mathscr{I}_f$ to select the network $\phi^i \in \Phi$ that maximizes the temperature-scaled softmax score for a test sample $x \in X_{test}$ as:

$$\mathscr{I}_f(x, \Phi) = \underset{\phi^i \in \Phi}{arg\,max} \left(\underset{c \in C}{max}(s_c^i) \right). \tag{4}$$

Incorporating temperature scaling as a calibration technique is especially crucial in our context, where accurate representation of confidence is essential for selecting the most competent network for a given ask.

The entire training and validation procedure of BRCL is explained in Algorithm 1.

3.5. Model consolidation

The proposed BRCL framework maintains multiple networks (M in total), which could lead to significant memory overhead compared to using a single network. We aim to decrease memory consumption by eliminating the need to store one backbone network for each task, a strategy utilized by DER [23] and critiqued in works like MEMO [25]. MEMO's analysis challenges the necessity of a backbone per task, particularly from a memory-efficiency standpoint, suggesting omitting layers that undergo lower gradient changes could be a more efficient approach. However, MEMO's findings also highlight certain limitations. Firstly, their approach yields significant benefits primarily when a vast number of classes are introduced in the initial task, which may not be feasible or practical in all scenarios. On the other hand, the generalizability of the gradient shifting measurement could vary across different network architectures. Additionally, the process of determining which layers to retain or remove, as suggested by their method, could indeed be time-consuming and computationally intensive. Finally the proposed memory saving approach amounts to 22% and ends up having 78% of DER's memory consumption. BRCL tries to overcome these limitations, by proposing an approach that is not only more generalized across different network architectures but also avoids the intensive process of layer-wise evaluation for memory efficiency. By doing so, we aim to provide a more universally applicable solution in OCL, particularly for scenarios with diverse and varying datasets. Hence, we introduce a

consolidation algorithm to remove redundant networks and reduce memory consumption. The redundancy can be quantified through the similarities between feature spaces of $\phi^i(D)$ and $\phi^j(D)$. Intuitively, if feature maps learned from ϕ^i and ϕ^j are statistically similar, both networks will yield comparable performance on the same dataset D; hence, the newer model can retain the knowledge from the previous task and it can be used to substitute the older model. Our consolidation strategy focuses on removing older networks associated with previous tasks to conserve memory. Fig. 4 illustrates Maximum Mean Discrepancy (MMD) [55] values between features learned by ϕ^i and ϕ^j on CIFAR100 (normalized), which can be computed through:

$$\text{MMD}\left[\phi^i\left(\bigcup_{k\le i} D^k;\theta^{fc}\right),\phi^j\left(\bigcup_{k\le i} D^k;\theta^{fc}\right)\right]; \tag{5}$$

where θ^{fc} indicates the last fully connected layer of the network, and $j \geq i$ indicates the values in the upper diagonal region of the matrix.

In Fig. 4, the MMD values along the same row indicate the divergence between feature representations generated from the same underlying data. We observe that as we move from left to right along a row, the MMD distance between feature representations tends to increase. This increase is more pronounced when more tasks are added to the model; which can be attributed to the networks specializing on the current task at hand with abundant available data. It seems since the network was initialized with the first task, its distribution stays aligned with the first task after training on multiple tasks, that's why we don't see much diversion in the first row in comparison to last rows. This inspires us to keep fewer networks at the beginning but more at a later stage. This observation motivates us to develop a consolidation algorithm, detailed Algorithm 2. Briefly, the algorithm compares the current model ϕ^i and its r preceding models. If their MMD distance falls below a threshold δ, we remove the preceding models and retain the current one. Both r and δ are tunable parameters, allowing for different subsets of models to be selected.

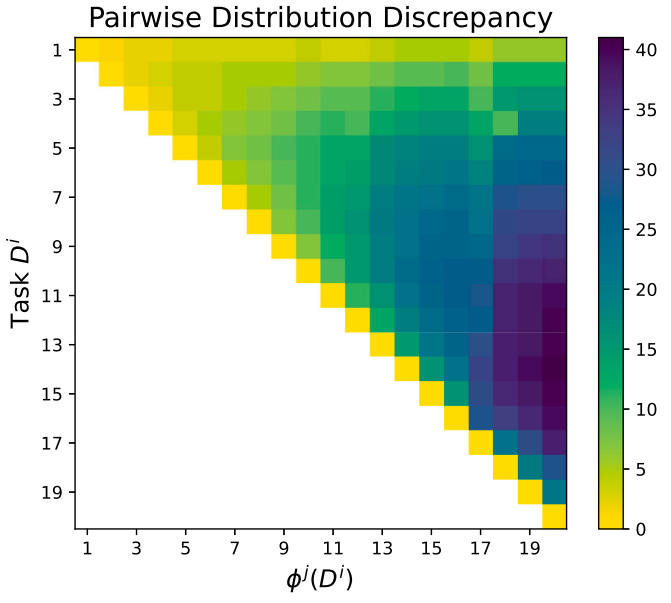

Fig. 4. Feature discrepancy distribution generated by MMD. Where each row represents a task D^i of CIFAR100 dataset and each column denotes the ϕ^j used for extracting features of D^i. Therefore each cell shows $\phi^j\left(D^i;\theta^{fc}\right)$ where $j \geq i$. Since data is trained incrementally, ϕ^j has not seen $D^{i>j}$, that's why the lower diagonal section is empty. Darker color represents higher discrepancy, while lighter color shows closer feature distribution.

Algorithm 2. Model Consolidation.

Input: Network set Φ, exemplar sets E, threshold for discrepancy δ, and backtracking parameter $r = 2$.
Output: Selected model set $\hat{\Phi}$.

for $i = 1$ to $M - r$ **do**
  $max = i + r$
  while $i <= max$ **do**
    Add $\phi^i(\varepsilon^{max})$ to the set $\hat{\Phi}$
    if $\text{MMD}(\phi^{max}(\varepsilon^i), \phi^i(\varepsilon^i)) \leq \delta$ **then**
      eliminate ϕ^i from $\hat{\Phi}$
      retain ϕ^{max} in $\hat{\Phi}$.
    else if $\text{MMD}(\phi^{max-1}(\varepsilon^i), \phi^i(\varepsilon^i)) \leq \delta$ **then**
      eliminate ϕ^i from $\hat{\Phi}$
      retain both ϕ^{max} and ϕ^{max-1} in $\hat{\Phi}$.
    else
      retain all
    end if
  end while
end for
return subset set: $\hat{\Phi}$

4. Experiments and results

This section explains the datasets, metrics, experimental settings, implementation details, results, and analysis.

4.1. Datasets

We evaluate our method and recent SOTA methods on three popular CL visual datasets, including:

- **CIFAR100** [56]. It contains 100 classes. For each class, 500 images are used for training and 100 for testing. The image size is 32×32.
- **mini-ImageNet** [57]. As a mini version of ImageNet, it contains its first 100 classes. For each class, 1200 images are used for training and 100 for testing. The image size is 256×256 with a center crop of 224×224.
- **ImageNet-1 K** [57]. This dataset contains 1000 classes. For each class, 1200 images are used for training and 100 for testing. The image size is 256×256 with a center crop of 224×224.

Both mini-ImageNet and ImageNet-1 k are divided to 10 Incremental tasks of 10 and 100 classes respectively.

4.2. Metrics

In addition to the classification accuracy, we use four CL evaluation metrics [13,14,17,58], as explained below:

End Incremental Accuracy (EIA). Let $a_j^i \in [0,1]$ denote the top-1 test accuracy on the j-th task after the training concludes on the i-th task ($j \leq i \leq M$). EIA is then defined as a_j^M, which provides a snapshot of the model's performance after all tasks have been learned.

Average Incremental Accuracy (AIA). Since the accuracy is continually updated in the OCL setting, to capture the historical variation of all tasks, we define AIA as the average of incremental accuracies up to the final task M:

$$AIA = \frac{1}{M}\sum_{j=1}^{M} a_j^M. \tag{6}$$

Average Final Forgetting (AFF). Forgetting provides a measure of

the model's ability to retain knowledge from previous tasks. For the j-th task, final forgetting evaluates the decline from the peak accuracy to its final accuracy after learning the last task. Hence, AFF aggregates the model's forgetfulness as:

$$AFF = \frac{1}{M-1}\sum_{i=1}^{M-1} \max_{j}\left(a_j^i - a_j^M\right), j \leq i. \tag{7}$$

A lower AFF indicates stronger retention of knowledge across tasks.

Average Learning Accuracy (ALA). OCL involves learning tasks sequentially, where learning plasticity can be compromised by model's stability. We define the *plasticity measure* based on ALA, which evaluates the capability of the model to learn new information [58] and Forward Transfer (FWT) [14] to quantify how continual learning is helpful in learning a task. ALA, therefore, is defined as:

$$ALA = \frac{1}{M}\sum_{i=1}^{M} a_i^i. \tag{8}$$

Backward Transfer. To gauge the impact of learning new tasks on the performance of older tasks, we adopt the *BackWard Transfer (BWT)* metric in this experiment following the setup in [59]. Given the top-1 test accuracy a_j^i on the j-th task after concluding the training on the i-th task, BWT is defined as follows:

$$\text{BWT} = \frac{1}{M-1}\sum_{i=2}^{M}\frac{1}{i}\sum_{j=1}^{i}\left(a_i^i - a_j^j\right). \tag{9}$$

A larger value for BWT indicates better retention of performance on earlier tasks when learning new tasks, providing a comprehensive view of a model's adaptability in learning continually.

4.3. Baselines

We compare BRCL with leading online, dynamic networks, and offline methods explained in the following. For *iCaRL* [15], following [13], we modified the original MemoryUpdate technique to use reservoir sampling [60], making it apt for online scenarios. The modified iCaRL is denoted as iCaRL*. *MIR* introduces a new strategy for retrieving samples from buffer to optimize information transfer [30]. *A-GEM* [20] (a more efficient version of *GEM* [19]) aims to prevent forgetting by constraining the parameter update using samples in the memory buffer. *ER* [12] is a replay-based effective method that leverages reservoir sampling to update and random sampling to retrieve from the memory. *GDumb* [9] estimates parameter importance from gradients computed for the current task and adjusts learning rates accordingly.

Regarding feature replay-based methods, *DER* [23] expands a new backbone for facing new tasks and introduces effective auxiliary loss. *FOSTER* [24] optimizes the memory consumption by limiting to only one backbone. By combining gradient boosting with the teacher-student model, *FOSTER* aims to leverage the strengths of both approaches. *MEMO* [25] decouples the feature maps from the middle layers of the network, where specialized blocks assimilate deep-layer features and generalized blocks learn common task features.

We also compare against various baselines, including: *LUCIR* [7] and *BiC* [5] which tackle bias very effectively; *Mn-T* with Feedback-based Exemplar Selection that prefers samples in boundary of feature map [34]; *iTAML* [61] and *DML* [62] as two strong meta-learning baselines. In particular, DML uses a distillation loss to encourage the network to learn representations in align with older tasks. For a fair comparison, with offline methods that revisit historical data, we keep our online framework but use an image-set priori. Conferring from Eq. (10), a higher n translates to more samples of the same label in the test batch and thus results in enhanced accuracy. We set $n = 5$ only in this experiment.

4.4. Implementation details

We use two *A100* GPUs and PyTorch libraries for model training and evaluations. Moreover, ResNet-32 was applied as the backbone for CIFAR100, and ResNet-18 for both mini-ImageNet and ImageNet-1 k. In addition, temperature value is set to $T = 1.8$ for all experiments on CIFAR100, since this value resulted in highest confidence score in softmax scores; while, for ImageNet datasets T is set to 1.6. The experiments on all three datasets start with an initial learning rate of 0.1 and weight decay of 0.1, every 30 epochs. CIFAR100 is trained with 160 epochs and both ImageNet datasets 70 epochs. All experiments use a fixed value of 20 exemplars per class in the batch unless otherwise specified. For CIFAR100, we apply three task divisions: 5, 10 and 20. For mini-ImageNet and ImageNet-1 k datasets, we divide the dataset into 10 tasks, with 10 and 100 classes per task, respectively. We start the incremental process from the beginning throughout all experiments ($base = 0$) to simulate the online data processing scenario in an online fashion.

4.5. Evaluation results

Comparisons with online Baselines. Evaluation on the CIFAR100 dataset (Table 1) shows that BRCL and its exemplar-free variation BRCL ($\varepsilon = 0$) consistently outperform online baselines in all metrics and task settings. The largest performance gap is seen in the 5-incremental-tasks setting in EIA and AIA. Our framework successfully balances stability and plasticity, as indicated by reduced forgetting (AFF) and improved learning ability. For mini-ImageNet dataset with 10 incremental tasks (Table 2), BRCL and BRCL ($\varepsilon = 0$) outperform the strongest baseline, Gdumb, by margins of 19.4% and 13.9% in end incremental accuracy. On the large-scale ImageNet-1 K dataset (Table 3), our methods indicate gain over all baselines. Though the accuracy in terms of EIA and AIA remains competitive, forgetting and learning ability sets our proposed methods apart. Even our exemplar-free variation, demonstrates an acceptable performance throughout all metrics.

Comparisons with Feature Replay Baselines. Table 2 shows that feature replay methods consistently perform closely across all evaluation metrics. BRCL is successful in mitigating average forgetting in all CIFAR100 task divisions. While DER takes the edge on 10 and 20 incremental tasks, BRCL outperforms all baselines in 5 tasks setting. In the case of mini-ImageNet, BRCL outperforms DER, FOSTER and MEMO in all evaluation metrics. On the other hand, BRCL ($\varepsilon = 0$) indicates the highest plasticity of 77.9%. Evaluation of the large-scale ImageNet-1 k dataset reveals a robust performance for our proposed methods. It achieves a 0.5% increase in average incremental accuracy (AIA) and minimizes forgetting to 5.1%, while maintaining a high plasticity. Notably, BRCL ($\varepsilon = 0$) surpasses all online baselines and FOSTER while operating in an exemplar-free setting.

Comparisons with Offline Baselines. For a more comprehensive analysis we decided to include comparisons with offline and meta-learning CL methods. Though the proposed inference model is not trained on any data, its function is similar to that of a meta-learner in meta-learning CL methods. Fig. 5 summarizes our experiments against selected offline and meta-learning approaches. As indicated before, in this experiment only, we use an image-set priori of $n = 5$ same-label samples. As indicated in 5a and 5c, BRCL consistently outperforms other baselines on mini-ImageNet. In the case of ImageNet-1 K, depicted in 5b and 5d, all methods initially perform well. However, as the number of tasks increases, our proposed methods widen the performance gap with the baselines, showing a robust performance in large-scale data. Conventional methods experience performance degradation due to the accumulation of bias. In contrast, our proposed methods effectively manage this bias, thereby maintaining a consistently higher performance as the number of tasks increases.

Table 1
Results (%) on **CIFAR100** dataset with three settings 5,10 and 20 incremental tasks. The performance is reported based on EIA, AIA, AFF, and ALA. [↑] indicates higher is better and [↓] indicates lower is better. **Bold** and underline highlight the best and second best, respectively. The horizontal line separates online and feature replay baselines and our proposed methods. BRCL ($\varepsilon = 0$) indicates the exemplar-free variabtion of BRCL where no exemplars are stored from previous classes. Model consolidation strategy is NOT used for our methods in this experiment.

Method	5 Inc. Tasks				10 Inc. Tasks				20 Inc. Tasks			
	EIA [↑]	**AIA** [↑]	**AFF** [↓]	**ALA** [↑]	**EIA** [↑]	**AIA** [↑]	**AFF** [↓]	**ALA** [↑]	**EIA** [↑]	**AIA** [↑]	**AFF** [↓]	**ALA** [↑]
iCaRL*(CVPR'17)	28.4	41.0	15.0	44.9	25.8	38.8	15.2	48.8	15.2	30.3	38.4	44.5
MIR (NIPS'19)	15.8	40.6	45.1	44.1	14.9	31.7	45.0	41.2	12.1	17.3	58.6	27.4
AGEM (ICLR'19)	12.9	26.9	59.1	38.3	7.58	19.5	66.8	33.5	4.1	13.6	73.6	27.5
ER (arXiv'19)	15.6	27.3	46.3	36.8	24.3	32.8	59.4	41.7	12.8	18.2	59.3	30.7
Gdumb (ECCV'20)	36.3	49.2	10.3	55.8	29.7	42.4	14.2	52.7	16.5	33.3	16.1	42.9
DER (CVPR'21)	58.4	67.8	11.4	73.9	**54.1**	62.2	26.3	**68.9**	**44.0**	**55.1**	30.8	63.8
FOSTER (ECCV'22)	52.7	64.6	34.1	70.1	46.4	53.7	35.9	64.3	36.4	53.4	38.7	**64.0**
MEMO (ICLR'23)	53.4	65.4	21.3	71.0	48.1	58	29.7	64.5	39.2	52.8	34.1	60.1
BRCL	**58.8**$^{+0.04}$	**68.1**$^{+0.3}$	**7.3**$^{-3}$	**74.5**$^{+0.6}$	53.7$^{-0.4}$	**63.5**$^{+1.3}$	**6.3**$^{-7.9}$	67.4$^{-1.5}$	43.4$^{-0.6}$	54.0$^{-1.1}$	**4.3**$^{-11.8}$	61.9$^{-2.1}$
BRCL ($\varepsilon = 0$)	54.3	65.7	13.0	71.1	44.4	57.3	14.5	68.1$^{-0.8}$	28.6	40.4	15.4$^{-0.7}$	53.4

Table 2
Results (%) on **mini-ImageNet** (10 Inc. Tasks). The performance is reported based on EIA, AIA, AFF, and ALA. [↑] indicates higher is better and [↓] indicates lower is better. Model consolidation strategy is NOT used for our methods in this experiment.

Method	**EIA** [↑]	**AIA** [↑]	**AFF** [↓]	**ALA** [↑]
iCaRL*	42.4	57.2	13.4	60.1
MIR	32.5	46.1	32.0	55.5
AGEM	25.0	30.3	56.8	47.8
ER	41.3	58.1	50.4	61.0
Gdumb	44.6	59.9	17.6	62.9
DER	63.2	72.0	6.8	76.2
FOSTER	60.4	69.4	8.8	76.0
MEMO	60.8	71.1	8.5	77.2
BRCL	**64.0** $^{+0.8}$	**72.2** $^{+0.2}$	**6.5** $^{-0.3}$	76.6 $^{-0.6}$
BRCL ($\varepsilon = 0$)	58.5	67.1	12.4	**77.9** $^{+0.7}$

Table 3
Results (%) on **ImageNet-1 K** (10 Inc. Tasks). The performance is reported based on EIA, AIA, AFF, and ALA. [↑] indicates higher is better and [↓] indicates lower is better. Model consolidation strategy is NOT used for our methods in this experiment.

Method	**EIA** [↑]	**AIA** [↑]	**AFF** [↓]	**ALA** [↑]
iCaRL*	11.9	26.3	18.7	35.4
MIR	16.5	45	26.1	49.5
AGEM	8.2	15.4	70.5	20.1
ER	11.6	22.7	36.6	33.9
Gdumb	12.5	24.5	20.2	36.6
DER	58.2	68.5	7.4	74.8
FOSTER	43.7	60.9	10.5	70.6
MEMO	57.2	67.8	7.3	74.4
BRCL	**58.7** $^{+0.5}$	**68.7** $^{+0.2}$	**5.1** $^{-2.2}$	**76.0** $^{+1.2}$
BRCL ($\varepsilon = 0$)	55.9	65.9	12.8	72.8

4.6. Memory footprint

Continual learning inherently involves a trade-off between *memory consumption* and *performance*, a concept echoed by recent works including [17]. An inclusive assessment needs to take memory footprint into account to avoid skewed evaluations based on different resource utilization. Table 4 outlines the memory requirements for the CIFAR100, mini-ImageNet, and ImageNet-1 K datasets, configured with 10 incremental tasks. To avoid redudancy we report only best performing method in each category of benchmarks; for example, DML [62] is the select method in *Meta-learning* benchmarks. The memory size (MB) is detailed in terms of the resources needed for training sets, exemplars, and model parameters, with a summarization in the Total Storage column. The number of parameters (**#P**), is also provided to understand the size of parameters networks(s) store in total. The networks used for training CIFAR100 and ImageNet datasets are ResNet-32 and ResNet-18, respectively, for all methods ensuring consistency in the evaluation framework. Finally, we report the model's accuracy using Average End Accuracy (AIA) to give an indication of performance alongside the memory footprint.

For our proposed method and it's variation we are using 10 incremental tasks, therefore, model consolidation is not used. In *Online* methods, the model doesn't require accessing "train set" after training is completed on a particular task, hence doesn't need storing the train set; therefore, Memory Size is noted as *Not Required (NR)*. Memory Size (**MS**) in megabytes is divided into *Train set*, *Exemplar* set and *Model Size* (network parameters required storage) and finally the column *Total Storage* shows the summation. In addition number of parameters, *#P* (in million), based on each method's architecture is shown. ResNet-32 is utilized for training all methods on CIFAR100, and ResNet-18 for training on the two ImageNet datasets. Last, the accuracy in terms of average end accuracy (AIA) is reported in last column. When it comes to memory consumption, online methods consume 10 MB less memory than our BRCL ($\varepsilon = 0$) on CIFAR100 but consume 57 MB and 2641 MB more on mini-ImageNet and ImageNet-1 K datasets.

In our analysis, we investigate the efficacy of methods using a radar plot (Fig. 6) indicating accuracy via EIA and AIA, and their retention capabilities by AFF and BWT. Additionally, we take into account the overall memory usage, which includes the combined requirements of exemplar storage and network parameters. To effectively visualize and compare the performance of various methods in radar plot, a series of data transformations were required. These transformations were essential to address the distinct ranges and negative values and to ensure a balanced representation of each metric. Accuracy metrics, EIA and AIA remains unchanged. Backward transfer values (BWT) are negative for all methods, the value range is shifted by subtracting each value from the absolute value of most negative one. For metrics like memory storage and forgetting index (AFF), where lower values are better, we inverted the values to ensure a coherent interpretation where higher values uniformly indicate superior performance. Finally the values are normalized within each metric. The area encompassed by each method's plot serves as an indicator of its relative efficacy, with larger areas denoting enhanced performance.

BRCL is implemented in two configurations: (i) maintaining 20 exemplars per class from preceding tasks ($\varepsilon = 20$) and (ii) an exemplar-free variant, where no exemplars are retained for training subsequent tasks ($\varepsilon = 0$), allowing to explore the trade-offs inherent to exemplar usage. All methods use 20 exemplars per class across CIFAR100, mini-ImageNet, and ImageNet-1 K datasets with 10 incremental tasks. From the visual clarity standpoint, the radar plot cannot show all the baseline methods discussed in this paper;therefore, Gdumb is selected as the best performing method among online methods; in addition, DER and MEMO

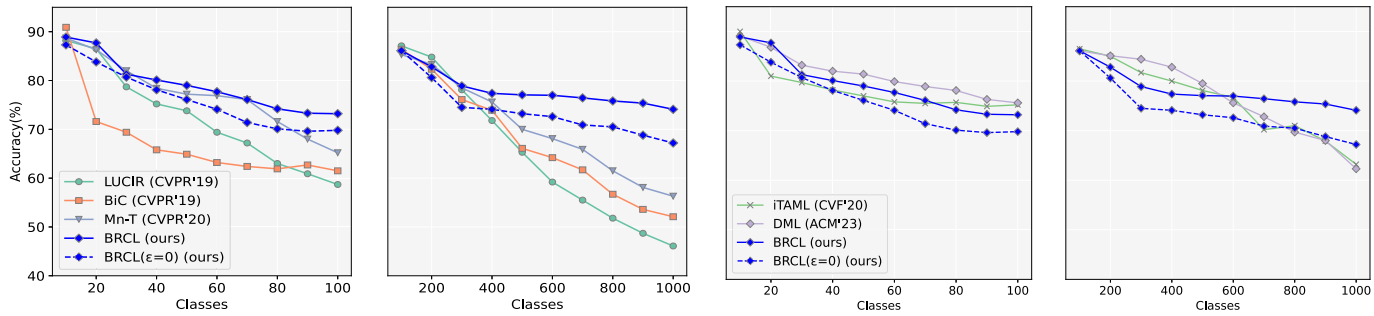

Fig. 5. Performance comparison in terms of accuracy % (AIA). Comparison of BRCL and BRCL($\varepsilon = 0$) with **offline** methods(a and b) and **meta-learning** methods(c and d).

Table 4
Memory consumption comparison of selected online, feature replay, and meta-learning methods. For performance report, we show the best-performing method in each category, which is *Gdumb*, *DML* and *DER*. "**MS**" indicates **M**emory **S**ize in MB. "*#P*" represents parameters (million) used in inference. Required storage is reported for *Train Set*, 20 *Exemplars* per class, *Model Size*, and their *Total*. ResNet-32 network is used for training on CIFAR100 and ResNet-18 for ImageNet datasets. All methods are training with 10 incremental tasks across all datasets. There are two configurations for BRCL in the last two rows, (1) 20 exemplars per class similar to all baselines, (2) exemplar-free variation. Model consolidation is not used in this experiment.

Method	Dataset	MS (Train Set)	MS (Exemplar)	Model Size	Total Storage	#P(M)	AIA(%)
Gdumb(Online)	CIFAR100	NR	5.85 MB	1.8 MB	7.6 MB	0.46	42.4
	mini-ImgNet	NR	287 MB	25.5 MB	312 MB	11.2	59.9
	ImageNet-1 k	NR	2871 MB	25.5 MB	2896 MB	11.2	44.1
DER (Feature replay)	CIFAR100	146 MB	5.85 MB	17.6 MB	23.5 MB	4.60	62.2
	mini-ImgNet	17,227 MB	287 MB	255 MB	542 MB	111.7	72.0
	ImageNet-1 k	172,265 MB	2871 MB	255 MB	175,391 MB	111.7	68.5
DML(Meta-learning)	CIFAR100	146 MB	5.85 MB	1.8 MB	154 MB	0.46	71.3
	mini-ImgNet	17,227 MB	287 MB	25.5 MB	17,539 MB	11.2	76.3
	ImageNet-1 k	172,265 MB	2871 MB	25.5 MB	175,151 MB	11.2	72.2
BRCL ($\varepsilon = 20$)	CIFAR100	NR	5.85 MB	17.6 MB	23.5 MB	4.60	60.7
	mini-ImgNet	NR	287 MB	255 MB	542 MB	111.7	71.7
	ImageNet-1 k	NR	2871 MB	255 MB	3126 MB	111.7	70.4
BRCL ($\varepsilon = 0$)	CIFAR100	NR	NR	17.6 MB	17.6 MB	4.60	57.3
	mini-ImgNet	NR	NR	255 MB	255 MB	111.7	67.1
	ImageNet-1 k	NR	NR	255 MB	255 MB	111.7	65.9

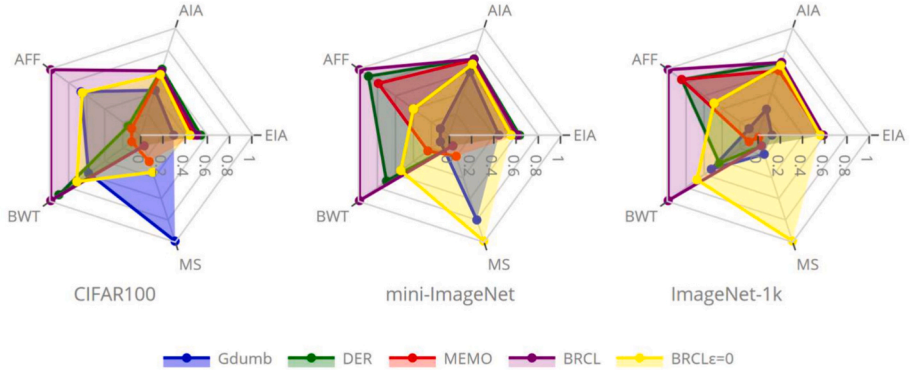

Fig. 6. Mutlidimensional Evaluation of Continual Learning Methods. Comparative analysis of different approaches, evaluating them based on five key metrics: End Incremental Accuracy (EIA), Average Incremental Accuracy (AIA), Average Final Forgetting (AFF), BackWard Transfer (BWT) and Memory Size (MS). MS accounts for the total storage as indicated in Table 4. For consistency all metrics are normalized. EIA and AIA measures are reported without any transformation. BWT values are offset; AFF and MS are inverted to show higher is better. Finally BWT, AFF and MS values are normalized. A method's efficacy is inferred from the surface area it occupies within the plot. It is important to note that the depicted surface areas represent relative, rather than absolute, performance magnitudes.

as the best benchmarks in feature replay methods.

The radar plot offers a multi-faceted visualization of performance trade-offs. a few observations can be made: Gdumb the select single-network candidate, shows the least memory usage for CIFAR100. However, its memory advantage decreases with larger datasets, suggesting that maintaining compressed network information could be more effective for knowledge retention than using raw exemplars alone. While accuracy (AIA & EIA) remains competitive, forgetting (AFF) and backward transfer (BWT) highlight the differences between the benchmarks. In MEMO's performance, we can see while the approach attempts to economize on memory usage by sharing shallow feature blocks, it suffers significantly in backward transfer. This trade-off results in a reduction in memory consumption but at a considerable cost to the model's ability to retain knowledge.

Moreover, the exemplar-free configuration of BRCL ($\varepsilon = 0$) remains competitive, especially in terms of memory efficiency, making it a strong candidate for resource-constrained settings. When BRCL operates without exemplars, it still shows notable performance on larger datasets. This indicates that the network parameters are capable of keeping features information, this is particularly noteworthy for large-scale datasets like ImageNet-1 K where it shows comparable accuracy and retention metrics. Overall, the performance of our proposed methods across different metrics indicates its adaptability and extendability. While there is no one-size-fits-all solution in CL, BRCL stands out for a remarkable trade-off between memory efficiency and performance.

This type of visualization aids in understanding the comprehensive capability of each method, especially when considering the deployment of continual learning systems in real-world settings where a balance between memory consumption, accuracy, and adaptability is crucial. Each method exhibits a distinct trade-off profile between the metrics, indicating that the choice of method should be tailored to the specific constraints and goals of the deployment environment, such as available memory and the importance of minimizing forgetting.

4.7. Time overhead

The process of traversing all stored models during inference increases computational demands. However, this increase is minimal in practical terms. For instance, using a ResNet18 model to classify an image of size 128×128 on an A100 GPU takes approximately 0.0001 s (100 microseconds) per image. Even if we multiply this by 10 for traversing multiple models, the total inference time would be around 1 s for 1000 images. By using the Consolidation Algorithm 2 to reduce the number of networks by half, the time overhead decreases further to approximately 0.5 s. This negligible overhead, especially when compared to the benefits BRCL framework, naming the minimal forgetting, highlights the trade-off between memory footprint and performance. As discussed by Zhou et al. [17], there is no "free lunch" in continual learning, and any performance gains typically come with increased memory or computation requirements. Our approach, while slightly increasing the inference time, ensures superior model performance in terms of accuracy, plasticity and knowledge retention in large-scale data settings, making it suitable for applications where these factors are critical.

4.8. Expandability of BRCL

BRCL introduces a robust yet simple approach for continual learning in large-scale dataset setting. Our approach innovatively leverages the inherent bias in a novel way: rather than reducing or correcting it; we use it to guide the selection of the most appropriate model during inference. This strategy enhances the system's ability to retain previously learned knowledge while adapting to new tasks. As a result, this method demonstrates significantly less forgetting, in comparison to baseline approaches.

Moreover, the consolidation strategy (Alg. 2) efficiently manages memory usage with minimal performance trade-offs. This algorithm sets a new standard by **eliminating the need for a new backbone per task in Dynamic Networks**, leveraging distribution similarity within networks. For example, MEMO [25] also uses feature similarly to create a multi-branch network that decouples general blocks and specializes network heads for each task. However, the generalizability and transferability of the generalized blocks heavily hinge on using an expansive set of base classes to capture feature representations. Another strong baseline, FOSTER [24] introduces several modules to address the memory footprint issue: (1) Feature Boosting, which involves freezing the old model and dynamically expanding new modules, to fit the residuals between the target and the old model's output (2) Logits Alignment, for scaling the logits to reduce classification bias between old and new classes; (3) Feature Compression, for removing redundant parameters; and (4) Feature Enhancement uses a distillation strategy to balance the learning of old and new categories. While this method is effective, it incorporates significant complexity and overhead. In contrast, BRCL proposes an effective yet straightforward methodology and further addresses the memory footprint, based solely on feature similarity without any priories. It is expandable to any backbone network, including transformers, and scalable across a variety of dataset sizes.

5. Ablation study and analysis

In this section, we perform experiments to validate the effectiveness of components of BRCL and their robustness to changes in hyperparameters. For this purpose, the CIFAR100 dataset with 20 incremental tasks is used. Table 5 summarizes the result of ablative experiments.

Inference Model $\mathscr{I}_m$. We hypothesize that the inference module $\mathscr{I}_m$ is an important component in the success of BRCL. We evaluate our hypothesis by ablating $\mathscr{I}_m$ from the model (second row). Therefore the last network which was trained on all tasks, will be used, similar to conventional OCL. In addition, there is a significant drop of 11.5% in AIA and an increase of 28.0% in forgetting. It seems that the learning ability of the model is also impacted significantly as the ALA has dropped 8.3%.

Similarity threshold δ. We evaluate the performance of BRCL against different numbers of preserved networks ϕ^i by adjusting the similarity threshold δ. Intuitively, more preserved models help BRCL retain previous knowledge better, yielding higher accuracy. We follow Algorithm 2, and start from a small value for δ and gradually increase δ. This will limit the number of networks reserved as the similarity tolerance grows. Changing the value of δ in this manner resulted in 20, 12, 10, 8, and 5 networks reserved in Φ. Details of this experiment can be seen in Fig. 7. A higher δ discarded 15 out of 20 networks and impacted the performance based on the *Average Incremental Accuracy (AIA)* by 3%. Nonetheless, the BRCL with 5 networks reserved, is still better than the leading OCL approaches.

Test batch size n. In the context of our defined methodology, particularly Eq.3, we explored the impact of varying the number of text samples n, all belonging to the same class. The value of n same-label batches in a validation batch is a decisive metric in the performance of the model regarding both model selection and class prediction. This is important because larger samples size can effectively mitigate the influence of noisy predictions. To quantify this effect, we extend Eq. 10 by computing an average softmax score over a batch of n samples. For each network ϕ^i in the set Φ, and for each data batch $\{x_1, x_2, \ldots, x_n\}$ of same class, we calculate the average softmax score as follows:

$$\widehat{s}_i = \frac{1}{n}\sum_{j=1}^{n} \mathrm{Softmax}\big(\big(\phi^i(x_j;\theta)/T\big)\big). \tag{10}$$

The above equation, allows us to determine the consensus decision across multiple samples, reducing the randomness inherent in individual predictions. Subsequently, the inference function $\mathscr{I}_m$ is hypothesized to provide a more reliable network selection mechanism, especially in scenarios where the input samples are prone to variability and noise. Fig. 8 shows the impact of increasing n in performance of CIFAR100 and mini-ImageNet datasets in terms of AIA(%).

Both graphs show a marked improvement when the number of

Table 5
Ablation study on CIFAR100(20 Inc. Tasks) with 20 exemplars per class.

Method	EIA	AIA	AFF	ALA
$n = 1, \mathscr{I}_m, \lvert\Phi\rvert = 20$	43.4	54.0	4.3	61.9
$n = 1, \lvert\Phi\rvert = 1$	33.8	42.5	32.3	53.6
$n = 1, \mathscr{I}_m, \lvert\Phi\rvert = 10$	36.1	41.4	35.2	50.1
$n = 20, \mathscr{I}_m, \lvert\Phi\rvert = 20$	72.6	77.2	2.9	80.6

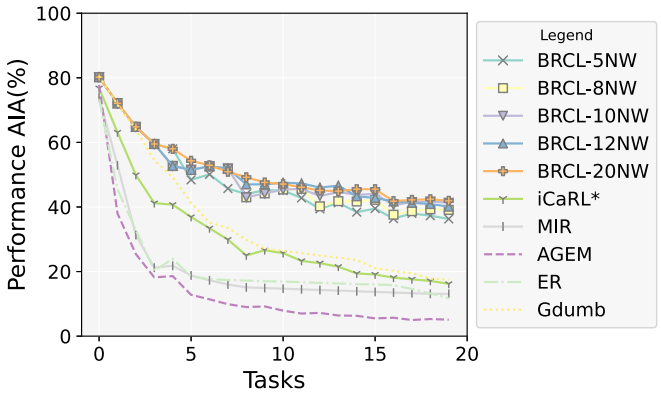

Fig. 7. Performance of BRCL with different number of reserved networks on CIFAR100 (20 Inc. Tasks). For BRCL, 20, 12, 10, 8, and 5 networks are utilized by tuning threshold δ.

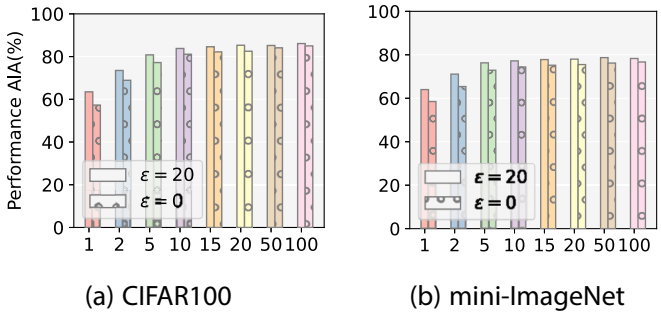

Fig. 8. Impact of test batch size n with the same label on (a) CIFAR100 (10 Inc. Tasks) and (b) mini-ImageNet (10 Inc. Tasks) datasets.

samples with the same label increases from 1 to 2, indicating that even a small increase in batch size can significantly enhance the reliability of the classification. For a fair comparison with baselines, we set $n = 1$ **in all previous experiments** including Table 1, Table 2 and Table 3, except for *Comparisons with Offline Baselines* IV-E, as explicitly noted in that section.

The performance could be boosted when $n > 1$, as shown in Table 5 (last row). Therefore, it is of theoretical interest to determine the lower bound on the number of successful predictions, denoted by k within a set of n batch size for the expected predictive performance. Given n samples in a test batch, we are interested in the number of successful predictions, as this will augment the softmax score, as shown in Eq. 10. Assume the random variable X indicates the number of successful predictions in a batch of n samples, then $P(X)$ follows a binomial distribution:

$$P(X = k) = \binom{n}{k} \cdot p_y^k \cdot \left(1 - p_y\right)^{n-k}; \tag{11}$$

where p_y is the (expected) prediction accuracy of an individual prediction, p_y is assumed to be constant across all trials within the batch. We expect a larger k to result the average softmax score for a test batch to facilitate the proposed inferencing. The expectation $\mathscr{E}(X) = np_y$ implies the expected value of X is linear in n and p_y. Since p_y is mainly determined by the network performance, adjusting the batch size n offers a lever to affect $\mathscr{E}(X)$. Empirical observations from CIFAR100 and mini-ImageNet datasets exhibit a pronounced increase in predictions performance at smaller batch sizes (e.g., $n = 1, 2, 5, 10$) with a tendency towards plateauing as n reaches and exceeds 10. As n becomes larger, the ratio kn begins to converge to p_y, signifying a stabilization in the amplification effect provided by the larger batch sizes. Drawing from the aforementioned discussion, it is evident that the value of k is contingent upon the batch size n and the inherent success probability p_y of the network's predictions. An empirical lower bound for k, which is crucial for obtaining dependable predictions, can be approximated as $k \geq 20p_y$. It is important to note that this lower bound may vary with different datasets and the network's performance, as represented by p_y.

5.1. Discussion

This paper introduces a framework for OCL that takes performance, memory consumption and knowledge retention into account. The offered model is simple yet generalizable to other datasets and network architectures. This method, leverages an existing property in all continual learning models, known as bias to guide the inference model. The consolidation strategy enables our framework to dynamically allocate memory resources and prioritize crucial information retention from earlier tasks, practically achieving a harmonious balance between learning new information and retaining previously acquired knowledge. This approach contributes to the scalability of BRCL, offering a practical solution for handling large datasets. Extensive experiments using variety of metrics are implemented to evaluate the proposed framework. The multi-faced radar plot is recommended as a tool to evaluate different approaches, using a set of metrics to show the dominant strength of each method. By demonstrating the effectiveness of dynamic and memory-aware strategies, our work paves the way for OCL models to handle large-scale data with limited resources. In future work, we can explore the potential of fusing multiple metrics, as discussed in the multi-sensor fusion [63], to further improve the model consolidation performance. While we currently rely on a single MMD distance, incorporating multiple metrics could provide a more robust solution by capturing different aspects of the data.

6. Conclusions

In this work, we introduced Bias-Robust class Continual Learning (BRCL), a simple yet effective framework in Online Continual Learning (OCL). BRCL employs a two-module strategy to maintain a set of incrementally learned networks, and to utilize the inherent bias towards recent tasks to selects the competent network, at inference time. A standout feature of our proposed approach is its consolidation strategy to scale memory consumption with minimal computational overhead, by eliminating networks with closely convergent feature distributions. This strategy manages available resources and helps maintain an equilibrium, crucial in OCL environments where the model must adapt to new data. Additionally, the exemplar-free variation of BRCL shows competitive performance and is distinguished by its raw-data-free strategy, well-suited for applications with privacy concerns. Extensive experiments using a comprehensive array of metrics was conducted to benchmark our proposed framework against current state-of-the-art methods.

CRediT authorship contribution statement

Neela Rahimi: Writing – original draft, Visualization, Validation, Methodology, Investigation, Formal analysis, Conceptualization. **Ming Shao:** Writing – original draft, Supervision, Resources, Project administration, Funding acquisition.

Declaration of generative AI and AI-assisted technologies in the writing process

During the preparation of this work the author(s) used *Consensus* in order to enhance clarity and readability of manuscript. After using this tool/service, the author(s) reviewed and edited the content as needed and take(s) full responsibility for the content of the publication.

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.

Acknowledgement

We gratefully acknowledge the support for this research from various sources. This work has been partially supported by the Marine and Undersea Technology (MUST) Research Program at the University of Massachusetts Dartmouth, funded by the Office of Naval Research (ONR) under Grant No. N00014-20-1-2170. Additionally, we acknowledge the support from the National Science Foundation under Grant No. 2144772. Our thanks also extend to the UMass Dartmouth Cybersecurity Center and the Center for Scientific Computing and Data Science Research (CSCDR) at UMass Dartmouth for their invaluable contributions to this work.

References

[1] F.M. Castro, M.J. Marín-Jiménez, N. Guil, C. Schmid, K. Alahari, End-to-end incremental learning, in: Proceedings of the European conference on computer vision (ECCV), 2018, pp. 233–248, 1, 2.
[2] Z. Li, D. Hoiem, Learning without forgetting, IEEE Trans. Pattern Anal. Mach. Intell. 40 (12) (2017) 2935–2947, 1.
[3] M. McCloskey, N.J. Cohen, Catastrophic interference in connectionist networks: The sequential learning problem, in: Psychology of Learning and Motivation 24, Elsevier, 1989, pp. 109–165, 1.
[4] H. Ahn, J. Kwak, S. Lim, H. Bang, H. Kim, T. Moon, Ss-il: Separated softmax for incremental learning, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2021, pp. 844–853, 1, 2, 3.
[5] Y. Wu, Y. Chen, L. Wang, Y. Ye, Z. Liu, Y. Guo, Y. Fu, Large scale incremental learning, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019, pp. 374–382, 1, 2, 7.
[6] A. Rannen, R. Aljundi, M.B. Blaschko, T. Tuytelaars, Encoder based lifelong learning, in: Proceedings of the IEEE International Conference on Computer Vision, 2017, pp. 1320–1328, 1.
[7] S. Hou, X. Pan, C.C. Loy, Z. Wang, D. Lin, Learning a unified classifier incrementally via rebalancing, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019, pp. 831–839, 1, 2, 7.
[8] P. Liu, H. Zhao, M. Chen, L. Qi, Class-incremental learning via deep model consolidation, in: The IEEE Winter Conference on Applications of Computer Vision, IEEE, 2020, pp. 1719–1728, 1, 2.
[9] A. Prabhu, P.H. Torr, P.K. Dokania, Gdumb: A simple approach that questions our progress in continual learning, in: European Conference on Computer Vision, Springer, 2020, pp. 524–540, 1, 2, 7.
[10] T.L. Hayes, N.D. Cahill, C. Kanan, Memory efficient experience replay for streaming learning, CoRR abs/1809.05922 (2018) 1.
[11] R. Aljundi, M. Lin, B. Goujaud, Y. Bengio, Gradient based sample selection for online continual learning, Adv. Neural Inf. Proces. Syst. 32 (2019), 1, 2.
[12] A. Chaudhry, M. Rohrbach, M. Elhoseiny, T. Ajanthan, P.K. Dokania, P.H. Torr, M. Ranzato, On tiny episodic memories in continual learning, arXiv preprint arXiv: 1902.10486, 2019, 1, 2, 7.
[13] Z. Mai, R. Li, J. Jeong, D. Quispe, H. Kim, S. Sanner, Online continual learning in image classification: An empirical survey, Neurocomputing 469 (2022) 28–51, 1, 2, 3, 6, 7.
[14] L. Wang, X. Zhang, H. Su, J. Zhu, A comprehensive survey of continual learning: Theory, method and application, arXiv preprint arXiv:2302.00487, 2023, 1, 6, 7.
[15] 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, 1, 2, 3, 7.
[16] J. Xu, Z. Zhu, Reinforced continual learning, in: Advances in Neural Information Processing Systems, 2018, pp. 899–908, 1.
[17] D.-W. Zhou, Q.-W. Wang, Z.-H. Qi, H.-J. Ye, D.-C. Zhan, Z. Liu, Deep Class-Incremental Learning: A Survey, Feb. 2023 arXiv:2302.03648 [cs]. [Online]. Available: http://arxiv.org/abs/2302.03648 1, 2, 6, 9, 10.
[18] Q. Dong, S. Gong, X. Zhu, Imbalanced deep learning by minority class incremental rectification, IEEE Trans. Pattern Anal. Mach. Intell. 41 (6) (2018) 1367–1381, 1, 3.
[19] D. Lopez-Paz, M. Ranzato, Gradient episodic memory for continual learning, Adv. Neural Inf. Proces. Syst. 30 (2017), 1, 7.
[20] A. Chaudhry, M. Ranzato, M. Rohrbach, M. Elhoseiny, Efficient lifelong learning with a-gem, arXiv preprint arXiv:1812.00420, 2018, 1, 2, 7.
[21] C. Zeno, I. Golan, E. Hoffer, D. Soudry, Task agnostic continual learning using online variational bayes, *arXiv preprint arXiv:1803.10123*, 2018, 2.
[22] S. Lin, L. Yang, D. Fan, J. Zhang, Beyond not-forgetting: Continual learning with backward knowledge transfer, in: Advances in Neural Information Processing Systems 35, 2022, pp. 16 165–16 177. 2.
[23] S. Yan, J. Xie, and X. He, "DER: Dynamically Expandable Representation for Class Incremental Learning," pp. 3014–3023. 2, 5, 7.
[24] F.-Y. Wang, D.-W. Zhou, H.-J. Ye, and D.-C. Zhan, "FOSTER: Feature boosting and compression for class-incremental learning." [Online]. Available: http://arxiv.org/abs/2204.04662 2, 7, 10.
[25] D.-W. Zhou, Q.-W. Wang, H.-J. Ye, D.-C. Zhan, A Model or 603 Exemplars: Towards Memory-Efficient Class-Incremental Learning, Feb. 2023 [Online]. Available: http://arxiv.org/abs/2205.13218 2, 5, 7, 10.
[26] X. Liu, C. Wu, M. Menta, L. Herranz, B. Raducanu, A.D. Bagdanov, S. Jui, J.V. de Weijer, Generative feature replay for class-incremental learning, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops, 2020, pp. 226–227, 2.
[27] G. Shen, S. Zhang, X. Chen, Z.-H. Deng, Generative feature replay with orthogonal weight modification for continual learning, in: In 2021 International Joint Conference on Neural Networks (IJCNN), IEEE, 2021, pp. 1–8, 2.
[28] S. Dohare, J.F. Hernandez-Garcia, Q. Lan, P. Rahman, A.R. Mahmood, R.S. Sutton, Loss of plasticity in deep continual learning, Nature 632 (8026) (2024) 768–774, 2.
[29] M. Acharya, T.L. Hayes, C. Kanan, Rodeo: Replay for online object detection, arXiv preprint arXiv:2008.06439, 2020, 2.
[30] R. Aljundi, L. Caccia, E. Belilovsky, M. Caccia, M. Lin, L. Charlin, T. Tuytelaars, Online continual learning with maximally interfered retrieval, in: CoRR abs/1908.04742, 2019, 2, 7.
[31] P. Buzzega, M. Boschini, A. Porrello, D. Abati, S. Calderara, Dark experience for general continual learning: a strong, simple baseline, Adv. Neural Inf. Proces. Syst. 33 (2020), pp. 15 920–15 930. 2.
[32] C.V. Nguyen, Y. Li, T.D. Bui, R.E. Turner, Variational continual learning, arXiv preprint arXiv:1710.10628, 2017, 2.
[33] D. Gong, Q. Yan, Y. Liu, A.V.D. Hengel, J.Q. Shi, Learning bayesian sparse networks with full experience replay for continual learning, arXiv preprint arXiv: 2202.10203, 2022, 2.
[34] Y. Liu, Y. Su, A.-A. Liu, B. Schiele, Q. Sun, Mnemonics training: Multi-class incremental learning without forgetting, in: Proceedings of the IEEE/CVF conference on Computer Vision and Pattern Recognition, 2020, pp. 12 245–12 254, 2, 7.
[35] J.S. Smith, L. Valkov, S. Halbe, V. Gutta, R. Feris, Z. Kira, L. Karlinsky, Adaptive memory replay for continual learning, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2024, pp. 3605–3615, 2.
[36] S.-W. Lee, J.-H. Kim, J. Jun, J.-W. Ha, B.-T. Zhang, Overcoming catastrophic forgetting by incremental moment matching, Adv. Neural Inf. Proces. Syst. 30 (2017) 2.
[37] F. Zenke, B. Poole, S. Ganguli, Continual learning through synaptic intelligence, in: International Conference on Machine Learning, PMLR, 2017, pp. 3987–3995, 2.
[38] R. Aljundi, F. Babiloni, M. Elhoseiny, M. Rohrbach, T. Tuytelaars, Memory aware synapses: Learning what (not) to forget, in: Proceedings of the European Conference on Computer Vision (ECCV), 2018, pp. 139–154, 2.
[39] D. Lopez-Paz, M. Ranzato, Gradient episodic memory for continual learning, Adv. Neural Inf. Proces. Syst. 30 (2017) 2.
[40] X. He, H. Jaeger, Overcoming catastrophic interference using conceptor-aided backpropagation, in: International Conference on Learning Representations, 2018, 2.
[41] Q. Yan, S. Liu, X. Zhang, Y. Zhu, J. Sun, Y. Zhang, An internal-external constrained distillation framework for continual semantic segmentation, in: Chinese Conference on Pattern Recognition and Computer Vision (PRCV), Springer, 2023, pp. 325–336, 2.
[42] T. Lesort, A. Stoian, D. Filliat, Regularization shortcomings for continual learning, *arXiv preprint arXiv:1912.03049*, 2019, 2.
[43] J. Knoblauch, H. Husain, T. Diethe, Optimal continual learning has perfect memory and is np-hard, in: International Conference on Machine Learning, PMLR, 2020, pp. 5327–5337, 2.
[44] L. Pellegrini, G. Graffieti, V. Lomonaco, D. Maltoni, Latent replay for real-time continual learning, in: In 2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), IEEE, 2020, pp. 10 203–10 209. 2.
[45] T.L. Hayes, C. Kanan, Lifelong machine learning with deep streaming linear discriminant analysis, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops, 2020, pp. 220–221, 2.
[46] T.L. Hayes, K. Kafle, R. Shrestha, M. Acharya, C. Kanan, Remind your neural network to prevent catastrophic forgetting, in: European Conference on Computer Vision, Springer, 2020, pp. 466–483, 2.
[47] A. Iscen, J. Zhang, S. Lazebnik, C. Schmid, Memory-efficient incremental learning through feature adaptation, in: European Conference on Computer Vision, Springer, 2020, pp. 699–715, 2.
[48] L. Yu, B. Twardowski, X. Liu, L. Herranz, K. Wang, Y. Cheng, S. Jui, J.V.D. Weijer, Semantic drift compensation for class-incremental learning, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020, pp. 6982–6991, 2.
[49] 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) 4069, 2.
[50] K. Binici, S. Aggarwal, N.T. Pham, K. Leman, T. Mitra, Robust and resource-efficient data-free knowledge distillation by generative pseudo replay, Proc. AAAI Conf. Artif. Intell. 36 (6) (2022) 6089–6096, 2.

[51] G. Graffieti, D. Maltoni, L. Pellegrini, V. Lomonaco, Generative negative replay for continual learning, Neural Netw. 162 (2023) 369–383, 2.
[52] A. Agarwal, B. Banerjee, F. Cuzzolin, S. Chaudhuri, Semantics-driven generative replay for few-shot class incremental learning, in: Proceedings of the 30th ACM International Conference on Multimedia, 2022, pp. 5246–5254, 2.
[53] M. Welling, Herding dynamical weights to learn, in: Proceedings of the 26th Annual International Conference on Machine Learning, 2009, pp. 1121–1128, 3.
[54] C. Guo, G. Pleiss, Y. Sun, K.Q. Weinberger, On Calibration of Modern Neural Networks, 2017, 4.
[55] I.O. Tolstikhin, B.K. Sriperumbudur, B. Schölkopf, Minimax estimation of maximum mean discrepancy with radial kernels, in: D. Lee, M. Sugiyama, U. Luxburg, I. Guyon, R. Garnett (Eds.), Advances in Neural Information Processing Systems vol. 29, Curran Associates, Inc, 2016 [Online]. Available: https://proceedings.neurips.cc/paper/2016/file/5055cbf43fac3f7e2336b27310f0b9ef-Paper.pdf 5.
[56] A. Krizhevsky, Learning Multiple Layers of Features from Tiny Images, Master's thesis,, University of Tront, 2009, p. 5.
[57] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, L. Fei-Fei, Imagenet: A large-scale hierarchical image database, in: 2009 IEEE Conference on Computer Vision and Pattern Recognition, Ieee, 2009, pp. 248–255, 5, 6.
[58] Q. Gu, D. Shim, F. Shkurti, Preserving linear separability in continual learning by backward feature projection, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2023, pp. 24 286–24 295, 6, 7.
[59] S. Liu, T. Pan, C. Wang, X. Ma, W. Dong, T. Hu, S. Zhang, Y. Zhang, Q. Yan, A comprehensive review of continual learning with machine learning models, in: International Conference on Image, Vision and Intelligent Systems, Springer, 2023, pp. 504–512 (7).
[60] J.S. Vitter, Random Sampling with a Reservoir, 1985, pp. 37–57, https://doi.org/10.1145/3147.3165 7 [Online]. Available.
[61] J. Rajasegaran, S. Khan, M. Hayat, F.S. Khan, M. Shah, itaml: An incremental task-agnostic meta-learning approach, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020, pp. 13 588–13 597, 7.
[62] H. Liu, Z. Yan, B. Liu, J. Zhao, Y. Zhou, A.E. Saddik, Distilled meta-learning for multi-class incremental learning, ACM Trans. Multimed. Comput. Commun. Appl. 7 (2023) 9.
[63] Y. Zhang, H. Zhang, N.M. Nasrabadi, T.S. Huang, Multi-metric learning for multi-sensor fusion based classification, Inform. Fusion 14 (4) (2013) 431–440, 12.