Online Task-Free Continual Learning via Dynamic Expansionable Memory Distribution

Fei Ye[1] and Adrian G. Bors[2]
[1]School of Information and Software Engineering,
University of Electronic Science and Technology of China, Chengdu
[2]Department of Computer Science, University of York, York YO10 5GH, UK
feiye@uestc.edu.cn, adrian.bors@york.ac.uk

Abstract

Recent continuous learning (CL) research primarily addresses catastrophic forgetting within a straightforward learning framework where class and task information are predefined. However, in the Task-Free Continual Learning (TFCL), representing a more realistic and challenging CL scenarios, such information is typically absent. In this paper, we address the online TFCL by introducing an innovative memory management approach, by incorporating a dynamic memory system for storing selected data representatives from evolving distributions while a dynamically expandable memory system enables the retention of essential long-term knowledge. The proposed dynamic expandable memory system manages a series of memory distributions, each designed to represent the information from a distinct data category. A new memory expansion mechanism that assesses the proximity between incoming samples and existing memory distributions is proposed for evaluating when to add new memory distributions into the system. Additionally, a novel memory distribution augmentation technique is proposed for selectively gathering suitable samples for each memory distribution, enhancing the statistical robustness over time. To prevent memory saturation before the training phase, we introduce a memory distribution reduction strategy that automatically eliminates overlapping memory distributions, ensuring adequate capacity for accommodating new information in subsequent learning episodes. We conduct a series of experiments demonstrating that our proposed approach attains state-of-the-art performance in both supervised and unsupervised learning contexts. The source code is available at https://github.com/dtuzi123/DEMD.

1. Introduction

Lifelong/Continual learning aims to train a model that is able to continually capture novel information over time without forgetting any of the previously learnt knowledge. Different from the traditional deep learning, which would consider large dataset for training altogether, continual learning paradigm aims to learn successively a sequence of tasks, each defined by using a certain amount of data for training. When considering training successively a machine learning model on a sequence of datasets, the model suffers from significant performance degeneration, caused by catastrophic forgetting, [42].

Existing models aiming to mitigate catastrophic forgetting in continual learning can be summarized into three different branches : rehearsal-based methods [6, 9], which employ and manage a small size memory buffer for storing and replaying samples from previous tasks; regularization-based methods [29, 38], which modify the primary objective function by adding regularization terms that penalize significant changes of certain important network parameters; and expansion-based methods [11, 24], which dynamically creates and adds new sub-models and hidden layers into an expandable framework in order to learn new tasks. Among these approaches, the rehearsal-based method is a straight forward approach that balances the usage of memory with processing the requirements to address network forgetting in continual learning [17]. However, most rehearsal-based methods require access to the class and task information for implementing memory updating mechanisms [4, 53]. Consequently, such methods can not be applied in the Task-Free Unsupervised Continual Learning (TFUCL), which represents a more realistic continual learning scenario where both task and class labels are unavailable.

Numerous studies have addressed mitigating the issue of classifier forgetting during continual learning, whereas continual data generative models have received comparatively less attention [62]. Enabling generative models into continual learning frameworks can facilitate the implementation of artificial intelligence generation systems in dynamic, real-time learning environments, which holds substantial

practical significance. In this paper, we tackle the forgetting problem associated with training generative models under the more challenging TFUCL setting. Biological research concludes that information is continually processed in the brain through dual slow and fast memory mechanisms [39]. Inspired by these results we propose the Dynamic Expanding Memory Distribution (DEMD) model that integrates a Dynamic Memory System (DMS) for retaining the dynamic information alongside a Dynamic Scalable Memory System (DSMS) aimed at safeguarding essential long-term knowledge for the continual learning model. The DSMS orchestrates an array of continually evolving sub-memory buffers, each modeling a distinct memory distribution made up of a low-dimensional feature space aiming to encapsulate a compact probabilistic representation. To continuously accommodate novel information over time, we propose a novel dynamic memory expansion mechanism that incrementally generates and incorporates new memory distributions into the DSMS upon detecting shifts in the given data distributions. This expansion mechanism ensures a suitable discrepancy among memory distributions, thereby enabling to capture diverse knowledge while maintaining a compact memory capacity. Furthermore, we propose a novel memory augmentation strategy that enhances each memory distribution by preserving relevant data samples into their corresponding sub-memory buffers.

The proposed DEMD method initially allocates higher memory capacities for the dynamic memory system, maximizing the storage of data samples to furnish ample training data during the model's early learning stages. As the training progresses, the DEMD progressively aims to increase the capacity of the DSMS to retain essential long-term information while systematically discarding previously memorized samples from the DMS, thereby maintaining a streamlined memory system. Moreover, once the overall memory capacity reaches its limit, this memory strategy ceases to accommodate new data samples in subsequent learning sessions. To solve this challenge, we introduce an innovative memory distribution reduction technique that autonomously eliminates those memories recording overlapping knowledge, following a discrepancy assessment. This reduction mechanism consistently ensures sufficient memory capacity to retain novel information over time while maintaining the overall discrepancy among the memory distributions from DEMD, facilitating the preservation of a diversity of knowledge even when faced with a severely constrained memory capacity. Furthermore, we present a novel theoretical framework that offers theoretical insights and guarantees for the proposed memory-based management methodology.

We summarize our contributions as follows : (1) We propose a novel memory approach, namely the Dynamic Expansionable Memory Distribution (DEMD), that processes data through a dual fast-slow memory optimization mechanism; (2) We propose a novel dynamic memory expansion mechanism that automatically adds new memory distributions into the dynamic expansionable memory system without requiring any supervised signals; (3) We propose a novel memory distribution augmentation approach that evolves and augments each memory distribution by selectively accumulating appropriate data samples over time; (4) A novel memory reduction approach that automatically removes redundant memory distributions, continually providing sufficient memory capacities for preserving the novel information at the end of the training process; (5) We develop a novel theory framework to analyze and provide theoretical guarantees for the proposed memory approach.

2. Related Work

Rehearsal-based methods represent a popular way to address model forgetting by managing a small-size memory buffer for preserving past data samples. Such methods have shown good results in continual learning [7, 8, 19, 20, 25, 43, 46, 47, 54]. Other studies have improved the performance of the rehearsal-based methods by regularizing the objective function [2, 9, 12–14, 26, 37, 38, 41, 51, 57]. Moreover, rehearsal-based methods can be implemented by training a generative model such as a Variational Autoencoder (VAE) or a Generative Adversarial Network (GAN) [18], aiming to preserve and replay past data samples when learning new information. Such approaches are able to reproduce memorized data samples without requiring memory buffers [1, 30, 44, 52, 67].

Knowledge Distillation (KD) based methods train a dual teacher-student network, by aligning the outputs of its teacher and student components in order to compress the model structure [23]. One typical KD-based approach, called Learning Without Forgetting (LWF) [35] specifically encourages a student module to remember the input pattern of its corresponding teacher, thus relieving network forgetting. A well known KD method combined with an rehearsal-like approach is the Incremental classifier and Representation Learning (iCaRL) [45]. Specifically, iCaRL employs a new nearest-mean-of-exemplars classification strategy to learn a robust classifier. Another study proposes a self-supervised distillation mechanism to maintain all previously learnt features and representations, minimizing network forgetting [8]. However the KD models' performance decreases when significantly increasing the number of tasks to learnt during the continual learning.

Dynamic Network Architectures. Dynamic expansion models would increase the model's capacity to capture new information with new network layers or modules, resulting in mixture models, addressing the training scaling problems when increasing the number of tasks, [11, 24, 27, 43, 48, 55, 58, 68]. The Continual Generative Knowledge Distil-

lation (CGKD) [64] is a dynamic expansion framework replying data through generation, which dynamically creates and adds new experts into a mixture system. One of the advantages of the dynamic expansion models is that they can chronologically preserve the best performances in the given tasks, sometimes by freezing previously learnt subnetworks/experts [27]. In addition, recent studies have proposed to build experts based on the new backbones such as the Vision Transformer (ViT) [15]. The dynamic expansion model [16, 61] was shown to improve the model's performance on any individual task during the continual learning. However, the primary weakness of such dynamic expansion models is that it requires considerable storage space as well as computational costs. We provide additional discussions about the related works in **Appendix-A** from the Supplementary Material (SM).

3. Methodology

3.1. Preliminary

Continual learning, unlike the conventional learning which would utilize a huge training dataset at once for training a model [21], specifically seeks to enable the development of the model across a succession of learning tasks [66]. In this paper, we concentrate on a more intricate scenario where we consider a model trained in an online batch-to-batch learning without having access to any task or class information. Let $\mathcal{D}^j = \{\mathbf{x}_i\}_{i=1}^{n^{\mathcal{D}_i}}$ and $\tilde{\mathcal{D}}^j = \{\mathbf{x}_i\}_{i=1}^{n^{\tilde{\mathcal{D}}_i}}$ denote the j-th unlabeled training and testing sets within the data space $\mathcal{X} \in \mathbb{R}^{W\times H\times C}$, where W, H, and C represent the image width, height, and number of channels, respectively. $n^{\mathcal{D}_i}$ and $n^{\tilde{\mathcal{D}}_i}$ denote the total number of training and testing samples, respectively. In a class-incremental learning scenario, the training dataset $\mathcal{D}^j$ is typically made available for training by being partitioned into C' subsets $\{\mathcal{D}_1^j, \cdots, \mathcal{D}_{C'}^j\}$, with each subset $\mathcal{D}_c^j$ containing data samples from one or more adjacent categories. A data stream $S = \{\mathcal{D}_1^j, \cdots, \mathcal{D}_{C'}^j\}$ can be sequentially constructed from these subsets successively made available for training after being divided into n data batches within the batch-to-batch learning paradigm, represented as $S = \bigcup_{c=1}^{n}\{\mathbf{X}_c\}$. During a specific training duration ($\mathcal{T}_i$), only the associated data batch $\mathbf{X}_i = \{\mathbf{x}_{(1,i)}, \cdots, \mathbf{x}_{(b,i)}\}$ is accessible, while all previously seen data batches $\{\mathbf{X}_1, \cdots, \mathbf{X}_{i-1}\}$ remain inaccessible, where b indicates the batch size. The objective of a generative model $\mathcal{G}_{\theta_i}$ in the TFUCL setting is to identify the optimum parameter set θ_i that effectively minimizes the distance between the distribution of the generated images and the empirical distribution formed by all previously seen data batches at the learning stage $\mathcal{T}_i$:

$$\theta_i^* = \underset{\theta_i\in\Theta}{\operatorname{argmax}}\{F_{\text{dis}}(\mathcal{P}_{\mathbf{X}_{0:i}}, \mathcal{P}_{\mathbf{X}_{\theta_i}})\}\,, \tag{1}$$

where $F_{\text{dis}}(\cdot,\cdot)$ is a probability distance to evaluate the difference between two distributions. $\mathcal{P}_{\mathbf{X}_{0:i}}$ and $\mathcal{P}_{\mathbf{X}_{\theta_i}}$, denoting the empirical data distribution of all previously seen data batches $\{\mathbf{X}_1, \cdots, \mathbf{X}_i\}$ and the generated images $\mathbf{X}_{\theta_i} \sim \mathcal{G}_{\theta_i}$ produced by the generator $\mathcal{G}_{\theta_i}$, respectively. θ_i^* is the optimal parameter set estimated after the learning at $\mathcal{T}_i$, which minimizes the distance between $\mathcal{P}_{\mathbf{X}_{0:i}}$ and $\mathcal{P}_{\mathbf{X}_{\theta_i}}$. Once the model finishes the final training time ($\mathcal{T}_n$), we evaluate the model's performance on the whole testing dataset $\tilde{\mathcal{D}}^j = \{\mathbf{x}_i\}_{i=1}^{n^{\tilde{\mathcal{D}}_i}}$ using the model's generative performance criterion. In the experiments we also extend this unsupervised continual learning model to the case when class labels are available during the training, such is the case in the supervised continual learning.

3.2. The Memory System Structure

To effectively capture both short-term and long-term information in continual learning, we introduce an innovative memory approach consisting of a Dynamic Memory System (DMS), designated as $\mathcal{M}_i^D$, and a Dynamic Scalable Memory System (DSMS), denoted as $\mathcal{M}_i^{DS}$, where the subscript i corresponds to the memory buffers refreshed at $\mathcal{T}_i$. Specifically, the updating mechanism for the DMS employs a Last In First Out (LIFO) strategy, which is designed to retain recent information. The DSMS focuses on safeguarding essential and enduring information related to all previously learned categories, thereby effectively mitigating the issue of network forgetting. More precisely, we characterize $\mathcal{M}_i^{DS}$ as an expanding collection of t memory distributions, represented as $\{\mathcal{P}_{\mathcal{M}(1)}, \cdots, \mathcal{P}_{\mathcal{M}(t)}\}$, where t is adjustable over time to accommodate environmental changes. Each memory distribution $\mathcal{P}_{\mathcal{M}(j)}$ is constructed from a collection of retained samples stored within a corresponding fixed-size sub-memory buffer $\mathcal{M}(j)$. Notably, evaluating the probability distances between these distributions presents challenges due to the lack of having explicit probability density functions. Drawing inspiration from existing research that summarizes data samples through a defined multidimensional Gaussian distribution within a low-dimensional latent space [22, 49], we propose to formulate the memory distribution in the latent space. Specifically, we propose a mapping function $f\colon \mathcal{X} \to \mathcal{Z}$ that transforms a data sample into a low-dimensional feature space $\mathcal{Z} \in \mathbb{R}^{d'}$, where d' represents the dimensionality of the feature space. Utilizing $f(\cdot)$ enables us to derive a set of feature vectors for the sub-memory buffer $\mathcal{M}(j)$ as follows :

$$\mathbf{Z}_{\mathcal{M}(j)} = \{\mathbf{z}\,|\,\mathbf{z} = f(\mathbf{x}_c), c = 1, 2, \cdots, |\mathcal{M}(j)|\}\,, \tag{2}$$

where $\mathbf{x}_c \in \mathcal{M}(j)$ is the c-th memorized sample from $\mathcal{M}(j)$ and $|\mathcal{M}(j)|$ denotes the number of samples for $\mathcal{M}(j)$. By using $\mathbf{Z}_{\mathcal{M}(j)}$, we calculate the mean vector and

covariance matrix as :

$$\begin{aligned}\mu_{\mathcal{M}(j)}(s) =& \frac{1}{d'}\sum_{c=1}^{d'}\{\mathbf{z}_c(s)\}\,,\\ \Sigma_{\mathcal{M}(j)}(s,s') =& \mathbb{E}[(\mathbf{z}_c(s)-\mu_{\mathcal{M}(j)}(s))\\ & (\mathbf{z}_c(s')-\mu_{\mathcal{M}(j)}(s'))]\,,\end{aligned} \tag{3}$$

where $\mu_{\mathcal{M}(j)}(s)$ denotes the s-th dimension of the mean vector $\boldsymbol{\mu}_{\mathcal{M}(j)} = \{\mu_{\mathcal{M}(j)}(1)\cdots,\mu_{\mathcal{M}(j)}(d')\}$ and $\Sigma_{\mathcal{M}(j)}(s,s')$ denotes the variance between $\mathbf{z}_c(s)$ and $\mathbf{z}_c(s')$ that are the s-th and s'-th dimension of the feature vector $\mathbf{z}_c$ obtained by $\mathbf{x}_c$ using $f(\cdot)$. Let us denote $\boldsymbol{\Sigma}_{\mathcal{M}(j)} = \{\Sigma_{\mathcal{M}(j)}(1,1),\cdots,\Sigma_{\mathcal{M}(j)}(d,d')\}$ as a covariance matrix. Then we form the memory distribution to represent a compact representation for the associated sub-memory buffer ($\mathcal{M}(j)$) by considering :

$$\mathcal{P}_{\mathcal{M}(j)} = \mathcal{N}(\boldsymbol{\mu}_{\mathcal{M}(j)}, \boldsymbol{\Sigma}_{\mathcal{M}(j)})\,. \tag{4}$$

By using the explicit probability distribution, we can easily evaluate the discrepancy among memory distributions using probabilistic distances.

3.3. Discrepancy-based Memory Optimization

An optimal memory system $\mathcal{M}_i^{DS}$ should satisfy two aspects : (1) Each memory component representation should capture information which is distinct from the others; (2) The memory system $\mathcal{M}_i^{DS}$ should be deployed on a small number of memory components, thus limiting the number of required parameters. To achieve these two goals, we formulate the memory system updating as a min-max constrained optimization problem, expressed as :

$$\begin{aligned}&\min_{t=1,\cdots,n}\Big\{\max\Big\{\sum_{c=1}^{t}\Big\{\sum_{j=1}^{t-c-1}F_d(\mathcal{P}_{\mathcal{M}(j)},\mathcal{P}_{\mathcal{M}(c)})\Big\}\Big\}\Big\}\,,\\ &\text{s.t}\quad F_d(\mathcal{P}_{\mathcal{M}(j)},\mathcal{P}_{\mathcal{M}(c)})\leq\lambda\,,\end{aligned} \tag{5}$$

where $F_d(\cdot,\cdot)$ is the distance between two probability densities and λ is a pre-defined hyperparameter ensuring an appropriate discrepancy between two memory distributions. However, finding the optimal solutions from Eq. (5) using the gradient-based approaches is intractable because t is a discrete variable. Instead, we implement the goals from Eq. (5) by introducing a novel memory distribution expansion approach that appropriately adds a new sub-memory buffers into $\mathcal{M}_i^{DS}$ at a certain training time ($\mathcal{T}_i$) :

$$\min_{c=1,\cdots,b,j=1,\cdots,t}\{F'_d(\mathbf{x}_{(c,i)},\mathcal{P}_{\mathcal{M}(j)})\}>\lambda\,, \tag{6}$$

where $\mathbf{x}_{(c,i)}$ is the c-th sample from the data batch $\mathbf{X}_i$ at $\mathcal{T}_i$ and $F'_d(\cdot,\cdot)$ is a distance measure defined as :

$$\begin{aligned}&F'_d(\mathbf{x}_{(c,i)},\mathcal{P}_{\mathcal{M}(j)}) = 1-\\ &\frac{\sum_{s=1}^{d'}f(\mathbf{x}_{(c,i)})[s]\mu_{\mathcal{M}(j)}(s)}{\sqrt{\sum_{s=1}^{d'}(f(\mathbf{x}_{(c,i)})(s))^2}\sqrt{\sum_{s=1}^{d'}(\mu_{\mathcal{M}(j)}(s))^2}}\,,\end{aligned} \tag{7}$$

Algorithm 1 The training of the DSRF framework.

Input: The total number of training steps n; The model's parameters and the total number of memory distributions;
Output: The model's parameters

for $i < n$ **do**
- **Step 1 (Check the model expansion).**
- **if** $t = 0$ **then**
  - $\mathcal{M}(1) = \{\mathbf{x}_{(1,1)},\cdots,\mathbf{x}_{(b,1)}\}, t = t+1$
- **if** $|\mathcal{M}(1)| = \rho$ **then**
  - Form $\mathcal{P}_{\mathcal{M}(1)}$ using Eq. (4)
- **if** $\min_{c=1,\cdots,b}\{F'_d(\mathbf{x}_{(c,i)},\mathcal{P}_{\mathcal{M}(j)})\} > \lambda$ **then**
  - Build $\mathcal{M}(t+1)$ and add it into $\mathcal{M}_i^{DS}$
  - $t = t+1$
- **else**
  - **Step 2 (The memory distribution augmentation)**
  - **for** $c = 1, c \leq b$ **do**
    - $j^* = \underset{j=1,\cdots,t}{\operatorname{argmin}}\{F'_d(f(\mathbf{x}_{(c,i)}),\mathcal{P}_{\mathcal{M}(j)})\}$
    - **if** $|\mathcal{M}(j^*)| = \rho$ **then**
      - **if** $|\mathcal{M}_i^D| + |\mathcal{M}_i^{DS}| = \rho_{\text{all}}$ *and* $|\mathcal{M}_i^D| \neq 0$ **then**
        - Randomly remove one data from $\mathcal{M}_i^D$
        - Add $\mathbf{x}_{(c,i)}$ into $\mathcal{M}(j^*)$
      - **else**
        - Add $\mathbf{x}_{(c,i)}$ into $\mathcal{M}(j^*)$
  - **Step 3 (The memory distribution reduction)**
  - **if** $|\mathcal{M}_i^D| + |\mathcal{M}_i^{DS}| = \rho_{\text{all}}$ *and* $|\mathcal{M}_i^D| = 0$ **then**
    - $j^*, c^* = \underset{j,c=1,\cdots,t,j\neq c}{\operatorname{argmin}}\{F_d(\mathcal{P}_{\mathcal{M}(j)},\mathcal{P}_{\mathcal{M}(c)})$
    - **if** $F_{\text{dp}}(\mathcal{P}_{\mathcal{M}(j^*)}) < F_{\text{dp}}(\mathcal{P}_{\mathcal{M}(c^*)})$ **then**
      - Remove $\mathcal{P}_{\mathcal{M}(j^*)}$ from $\mathcal{M}_i^{DS}$
    - **else**
      - Remove $\mathcal{P}_{\mathcal{M}(c^*)}$ from $\mathcal{M}_i^{DS}$

where $f(\mathbf{x}_{(c,i)})(s)$ and $\boldsymbol{\mu}_{\mathcal{M}(j)}(s)$ denote the s-th dimension of the feature vector of $\mathbf{x}_{(c,i)}$ and the s-th dimension of $\boldsymbol{\mu}_{\mathcal{M}(j)}$. Given that $F_d(\cdot,\cdot)$ is a probability distance that cannot be utilized for a pair of samples, we propose the adoption of cosine similarity distance $F'_d(\cdot,\cdot)$ for two primary reasons: (1) It has a low storage requirement while having a high computational efficiency, particularly for low-dimensional feature spaces; (2) It is constrained within a range of -1 to 1, being easier to determine the threshold λ. Upon satisfying the criterion defined in Eq. (6) at $\mathcal{T}_i$, we utilize the novel sample $\mathbf{x}_{(c,i)}$ to initialize a new memory distribution (the sub-memory buffer $\mathcal{M}(t+1)$) and incorporate it into $\mathcal{M}_i^{DS}$ as :

$$\begin{aligned}&\mathcal{M}(t+1) = \{\mathbf{x}_{(c,i)}\},\\ &\mathcal{P}_{\mathcal{M}(t+1)} = \mathcal{N}(\boldsymbol{\mu}_{\mathcal{M}(t+1)} = f(\mathbf{x}_{(c,i)}), \boldsymbol{\Sigma}_{\mathcal{M}(t+1)} = \mathbf{I})\,,\end{aligned} \tag{8}$$

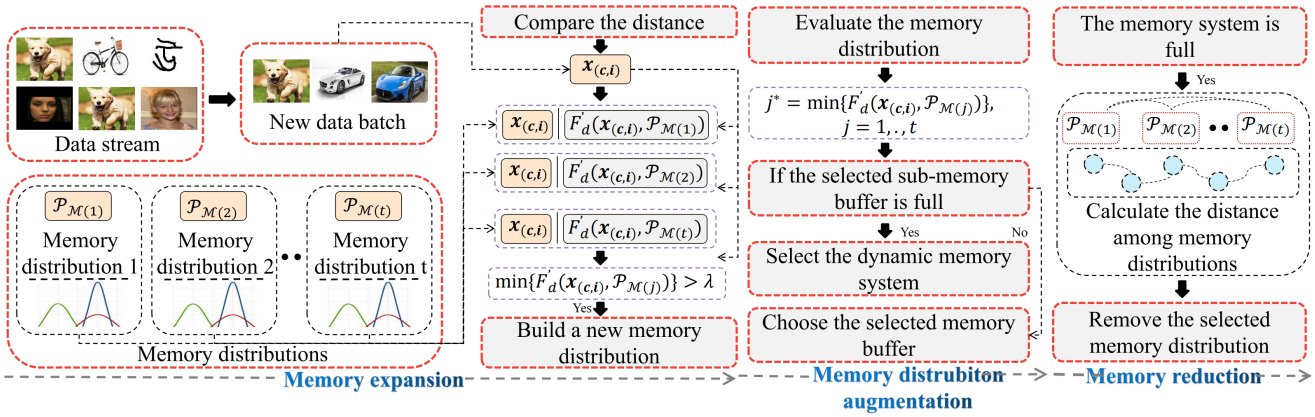


Figure 1. The optimization process of the proposed Dynamic Expanding Memory Distribution (DEMD) model, consisting of three steps at a training time $(\mathcal{T}_i), i = 1, \cdots, n$. In the first step, we dynamically create the first memory distribution $\mathcal{M}(1)$. In the subsequent learning, if Eq. (6) is satisfied at $\mathcal{T}_i$, we create a new memory distribution and add it into the memory system $\mathcal{M}_i^{DS}$. In the second step, we get the incoming data batch $\mathbf{X}_i$ at $\mathcal{T}_i$. For each sample $\mathbf{x}_{(c,i)}$ from $\mathbf{X}_i$, where Eq. (6) is not fulfilled, we choose an appropriate memory distribution using Eq. (9). If the selected memory distribution is full, we add $\mathbf{x}_{(c,i)}$ into the DMS ($\mathcal{M}_i^D$), otherwise, we add $\mathbf{x}_{(c,i)}$ into the selected memory distribution. In the second step, if the total memory capacity is full, we first determine a pair of the knowledge-overlapping memory distributions using Eq. (10) and then choose and remove the redundant memory using Eq. (11).

where we employ the identity matrix $\mathbf{I} \in \mathbb{R}^{d' \times d'}$ as the initial covariance matrix. In contrast, when the criterion defined in Eq. (6) is not satisfied, we perform the memory distribution selection for storing incoming data samples, described in the following section.

3.4. Memory Representation Augmentation

One of the objectives in Eq. (5) is to enhance the discrepancy among memory distributions in order to increase the memory representation by the DEMD model. To accomplish this, we propose to incentivize each memory distribution to encapsulate distinct information. This is realized through a novel memory distribution augmentation strategy that incrementally enriches the memorized statistical data representation over time. Specifically, if for an incoming data sample $\mathbf{x}_{(c,i)}$ from the data batch $\mathbf{X}_i$ at time $\mathcal{T}_i$, Eq. (6) is not fulfilled, we selectively assign that to an appropriate memory distribution according to :

$$j^* = \underset{j=1,\cdots,t}{\operatorname{argmin}} \{F_d'(f(\mathbf{x}_{(c,i)}), \mathcal{P}_{\mathcal{M}(j)})\}, \tag{9}$$

where j^* is the index of the selected memory distribution $\mathcal{P}_{\mathcal{M}(j^*)}$ (the sub-memory buffer $\mathcal{M}(j^*)$) used to store $\mathbf{x}_{(c,i)}$. If the selected sub-memory buffer $\mathcal{M}(j^*)$ is full $|\mathcal{M}(j^*)| = \rho$, where ρ represents its maximum capacity, we add $\mathbf{x}_{(c,i)}$ into the dynamic memory system $\mathcal{M}_i^D$. When the overall memory capacity is full $|\mathcal{M}_i^D| + |\mathcal{M}_i^{DS}| = \rho_{\text{all}}$, we automatically remove the earliest memorized samples from $\mathcal{M}_i^D$, which enables an efficient memory allocation for the DSMS in subsequent learning.

3.5. The Redundant Memory Reduction

When the DMS is empty, $|\mathcal{M}_i^D| = \emptyset$, and the DSMS is full, $|\mathcal{M}_i^D| = \rho_{\text{all}}$, the memory system can fail to safeguard essential long-term information leading to potential forgetfulness issues. In such situations, the system should eliminate extraneous memory distributions to ensure adequate capacity for new critical data sample retention. Drawing upon the results provided by Eq. (5), we introduce an innovative methodology for memory distribution reduction that systematically discards redundant memory distributions while preserving the discrepancy among the remaining distributions. Specifically, we begin by identifying a pair of memory distributions characterized by the minimal probability distance, defined as :

$$\{j^*, c^*\} = \underset{j,c=1,\cdots,t, j\neq c}{\operatorname{argmin}} \{F_d(\mathcal{P}_{\mathcal{M}(j)}, \mathcal{P}_{\mathcal{M}(c)})\}, \tag{10}$$

where j^* and c^* are the indices of the selected memory distributions. Given that both $\mathcal{P}_{\mathcal{M}(j)}$ and $\mathcal{P}_{\mathcal{M}(c)}$ are defined as explicit probability distributions, there are various measures which can be used for implementing the probability distance function $F_d(\cdot,\cdot)$. These include the Jensen–Shannon (JS) divergence, Kullback–Leibler (KL) divergence, or the Wasserstein metric. For our analysis, we opted for the symmetrical KL divergence due to two key considerations: (1) it possesses an analytical solution applicable to two explicit probability distributions; (2) it is substantiated by the findings outlined in **Theorem 1** from Section 4 and is underpinned by theoretical assurances. Consequently, in the alignment with Eq. (5), it is imperative that each remaining memory distribution exhibits a substantial difference from one another, after the memory distribution reduction phase. Therefore, we propose employing a discrepancy measure in order to eliminate one of the memory distributions, articulated as follows :

$$F_{\text{dp}}(\mathcal{P}_{\mathcal{M}(j^*)}) = \sum_{j=1}^{t} \{F_d(\mathcal{P}_{\mathcal{M}(j)}, \mathcal{P}_{\mathcal{M}(j^*)})\}. \tag{11}$$

Table 1. The Fréchet Inception Distance (FID) score evaluated on 5,000 testing data samples for the class-incremental setting.

Datasets	DEMD	DCM-SE	DCM-JS	LTS	LGM	CGKD-GAN	CGKD-WAE	MeRGANs
Split MNIST	**23.62**	28.57	30.63	71.67	66.31	54.34	47.98	49.96
Split Fashion	**40.16**	46.65	43.38	128.84	109.20	85.23	88.16	127.55
Split SVHN	**59.42**	61.52	62.61	87.25	72.60	101.2	100.15	81.35
Split CIFAR10	**73.27**	82.74	76.58	124.22	177.15	115.38	162.12	121.74
Average	**49.11**	54.87	53.30	102.99	106.31	89.05	99.54	95.15

Table 2. FID score for the imbalanced class setting, evaluated on 5,000 testing data.

Datasets	DEMD	DCM-SE	DCM-JS
Split MNIST	**26.95**	29.16	30.62
Split Fashion	**43.27**	46.91	48.49
Split SVHN	**57.92**	60.58	65.00
Split CIFAR10	**79.13**	82.28	90.44
Average	**51.81**	54.73	58.63

If $F_{\rm dp}(\mathcal{P}_{\mathcal{M}(j^*)}) < F_{\rm dp}(\mathcal{P}_{\mathcal{M}(c^*)})$, then we remove the memory distribution $\mathcal{M}_{j^*}$ from the DSMS ($\mathcal{M}_i^{DS}$), otherwise, we remove $\mathcal{M}_{c^*}$.

3.6. Algorithm Framework

In the following we provide the algorithm steps for the learning procedure, which is also illustrated in the diagram from Fig. 1, while the corresponding pseudocode is in **Algorithm 1** for the proposed DEMD framework, which is summarized into three steps at a certain training time $\mathcal{T}_i$:

Step 1 (Check the memory expansion). In the initial learning procedure, we dynamically create the first sub-memory buffer $\mathcal{M}(1)$ and continually store incoming data samples into $\mathcal{M}(1)$ until this buffer is full. Then, we form the memory distribution $\mathcal{P}_{\mathcal{M}(1)}$ by calculating the mean vector and covariance matrix using Eq. (3). In the subsequent learning, if the expansion criterion defined in Eq. (6) is satisfied, we create a new memory distribution, $\mathcal{M}_i^D$.

Step 2 (Memory representation augmentation). When the expansion criterion defined in Eq. (6) is not satisfied, we use Eq. (9) to update the appropriate memory distribution $\mathcal{P}_{\mathcal{M}(j^*)}$ for a given incoming sample $\mathbf{x}_{(c,i)} \in \mathcal{X}_i$ at $\mathcal{T}_i$. If $\mathcal{P}_{\mathcal{M}(j^*)}$ is full $|\mathcal{M}(j^*)| = \rho$, we store $\mathbf{x}_{(c,i)}$ into the DMS ($\mathcal{M}_i^D$), otherwise, we store $\mathbf{x}_{(c,i)}$ into $\mathcal{M}(j^*)$.

Step 3 (Redundant memory reduction). If the memory capacity is full $|\mathcal{M}_i^D| + |\mathcal{M}_i^{DS}| = \rho_{\rm all}$ and the DMS ($\mathcal{M}_i^D$) is empty, we perform the memory distribution reduction process, based on checking the redundancy using Eq. (10) and Eq. (11).

4. Theoretical Analysis and Guarantees

The Variational Autoencoder (VAE) [31] is the most popular model for unsupervised generative modeling. A VAE consists of an encoder and a decoder and is trained based on achieving a lower bound to the sample log-likelihood, called the Evidence Lower Bound (ELBO), as the primary objective function. The ELBO can also be used as the performance criterion for unsupervised generative modeling. In this paper, we formulate the decreasing in the ELBO as the forgetting assessment process and develop a novel theory framework to analyze the forgetting behavior of the proposed memory-based approach under the TFUCL scenario. First we provide some important definitions.

Definition 1 ***(The empirical memory distribution.)*** *$\mathcal{P}_{\mathcal{M}(j)}$ represents the embedding memory distribution for $\mathcal{M}(j)$ and we define $\mathcal{P}_{\widehat{\mathcal{M}}(j)}$ as the empirical data distribution of $\mathcal{M}(j)$ in the data space. Let $\mathcal{P}_{\mathcal{M}_i^{DS}}$ denote the empirical distribution of $\mathcal{M}_i^{DS}$ at $\mathcal{T}_i$. Let $\mathcal{P}_{\mathcal{M}_i^{DS},\widehat{\mathcal{M}}(1):\widehat{\mathcal{M}}(t)}$ be the empirical distribution of the memory systems $\mathcal{M}_i^{DS}$ and $\mathcal{M}_i^D$.*

Theorem 1 *Let S denote a data stream while the DSMS already contains t memory distributions at $\mathcal{T}_i$. We derive a lower bound for a VAE model at T_i, expressed as :*

$$\begin{aligned}\mathbb{E}_{\mathcal{P}_{\mathbf{x}_{0:i}}}\big[\log p_{\theta^i}(\mathbf{x})\big] &\geq \mathbb{E}_{\mathcal{P}_{\mathcal{M}_i^{DS},\widehat{\mathcal{M}}(1):\widehat{\mathcal{M}}(t)}}\big[\log p_{\theta^i}(\mathbf{x})\big] \\ &- F_d\big(\mathcal{P}_{\mathbf{x}_{0:i}}, \mathcal{P}_{\mathcal{M}_i^{DS},\widehat{\mathcal{M}}(1):\widehat{\mathcal{M}}(t)}\big) \\ &- \mathcal{F}_{\rm A}\big(\mathcal{P}_{\mathcal{M}_i^{DS},\widehat{\mathcal{M}}(1):\widehat{\mathcal{M}}(t)}, \mathcal{P}_{\mathbf{x}_{0:i}}, \mathcal{P}_{\theta^i}\big)\,,\end{aligned} \tag{12}$$

where the last term $\mathcal{F}_{\rm A}\big(\mathcal{P}_{\mathcal{M}_i^{DS},\widehat{\mathcal{M}}(1):\widehat{\mathcal{M}}(t)}, \mathcal{P}_{\mathbf{x}_{0:i}}, \mathcal{P}_{\theta^i}\big)$ from the Right-Hand-Side (RHS) of Eq. (12) is defined as :

$$\begin{aligned}&\big|F_d\big(\mathcal{P}_{\mathcal{M}_i^{DS},\widehat{\mathcal{M}}(1):\widehat{\mathcal{M}}(t)}, \mathcal{P}_{\theta^i}\big) - D_{\rm KL}\big(\mathcal{P}_{\mathbf{x}_{0:i}} \,||\, \mathcal{P}_{\theta^i}\big)\big| \\ &+ \mathbb{E}_{\mathcal{P}_i}\big[p_{W^i}(\mathbf{x}) \log p_{W^i}(\mathbf{x})\big] - \mathbb{E}_{\mathcal{P}_{\mathcal{M}_i^{DS},\widehat{\mathcal{M}}(1):\widehat{\mathcal{M}}(t)}}\big[\\ &p_{\mathcal{M}_i^{DS},\widehat{\mathcal{M}}(1):\widehat{\mathcal{M}}(t)}(\mathbf{x}) \log p_{\mathcal{M}_i^{DS},\widehat{\mathcal{M}}(1):\widehat{\mathcal{M}}(t)}(\mathbf{x})\big]\,,\end{aligned} \tag{13}$$

where $p_{W^i}(\mathbf{x})$ and $p_{\mathcal{M}_i^{DS},\widehat{\mathcal{M}}(1):\widehat{\mathcal{M}}(t)}(\mathbf{x})$ are the density functions for $\mathcal{P}_{\mathbf{x}_{0:i}}$ and $\mathcal{P}_{\mathcal{M}_i^{DS},\widehat{\mathcal{M}}(1):\widehat{\mathcal{M}}(t)}$, respectively. $F_d\big(\mathcal{P}_{\mathbf{x}_{0:i}}, \mathcal{P}_{\mathcal{M}_i^{DS},\widehat{\mathcal{M}}(1):\widehat{\mathcal{M}}(t)}\big)$ is the symmetrical KL divergence, defined as :

$$\begin{aligned}&D_{\rm KL}\big(\mathcal{P}_{\mathbf{x}_{0:i}} \,||\, \mathcal{P}_{\mathcal{M}_i^{DS},\widehat{\mathcal{M}}(1):\widehat{\mathcal{M}}(t)}\big) \\ &\quad + D_{\rm KL}\big(\mathcal{P}_{\mathbf{x}_{0:i}} \,||\, \mathcal{P}_{\mathcal{M}_i^{DS},\widehat{\mathcal{M}}(1):\widehat{\mathcal{M}}(t)}\big)\,,\end{aligned} \tag{14}$$

Table 3. FID results when considering datasets with complex images as well as when learning two different domains.

Datasets	DEMD	DCM-SE	DCM-JS	LTS	LGM	CGKD-GAN	CGKD-WAE	MeRGANs
CelebA-3DChair	**38.16**	40.45	82.18	186.25	241.14	132.12	154.45	166.99
CelebA-CACD	**45.28**	67.30	48.38	124.87	117.76	78.00	142.52	101.97
S-MINIImageNet	**140.02**	146.98	154.83	179.78	216.06	176.18	241.11	169.26

Table 4. FID scores for assessing the image generation performance for datasets containing high-resolution images.

Methods	Resolution	CelebA-HQ	CACD	FFHQ
DEMD	$128 \times 128 \times 3$	**86.71**	**55.69**	**89.72**
DCM-SE	$128 \times 128 \times 3$	89.23	69.11	95.02
DCM-JS	$128 \times 128 \times 3$	96.03	57.19	90.80
CGKD-GAN	$128 \times 128 \times 3$	132.65	142.66	157.03
CGKD-WAE	$128 \times 128 \times 3$	139.96	158.32	179.59
DEMD	$256 \times 256 \times 3$	**74.17**	**106.34**	**121.27**
DCM-SE	$256 \times 256 \times 3$	87.39	110.21	123.95
DCM-JS	$256 \times 256 \times 3$	75.18	123.96	129.38
CGKD-GAN	$256 \times 256 \times 3$	168.52	236.98	254.32
CGKD-WAE	$256 \times 256 \times 3$	176.63	240.12	261.37

where $D_{\mathrm{KL}}(\cdot,\cdot)$ is the KL divergence. According to Eq. (12) we can define the ELBO of $\mathcal{P}_{\mathbf{x}_{0:i}}$ at T_i as :

$$\begin{aligned}\mathbb{E}_{\mathcal{P}_{\mathbf{x}_{0:i}}}\big[\log p_{\theta^i}(\mathbf{x})\big] \geq -F_d\big(\mathcal{P}_{\mathbf{x}_{0:i}}, \mathcal{P}_{\mathcal{M}_i^{DS},\widehat{\mathcal{M}}(1):\widehat{\mathcal{M}}(t)}\big) \\ + \mathbb{E}_{\mathcal{P}_{\mathcal{M}_i^{DS},\widehat{\mathcal{M}}(1):\widehat{\mathcal{M}}(t)}}\big[\log \mathcal{L}_{ELBO}(\mathbf{x};\theta^i,\omega^i)\big] \\ - \mathcal{F}_{\mathrm{A}}\big(\mathcal{P}_{\mathcal{M}_i^{DS},\widehat{\mathcal{M}}(1):\widehat{\mathcal{M}}(t)}, \mathcal{P}_{\mathbf{x}_{0:i}}, \mathcal{P}_{\theta^i}\big)\,,\end{aligned} \tag{15}$$

*If the distribution $\mathcal{P}_{\mathcal{M}_i^{DS},\widehat{\mathcal{M}}(1):\widehat{\mathcal{M}}(t)}$ is equal to the data distribution $\mathcal{P}_{\mathbf{x}_{0:i}}$, then Eq. (15) becomes the standard ELBO. The detailed proof is provided in **Appendix-C** from SM.*

Remarks : Observations from **Theorem 1** : (1) The term $F_d(\cdot,\cdot)$ plays a critical role. If $\mathcal{P}_{\mathcal{M}_i^{DS},\widehat{\mathcal{M}}(1):\widehat{\mathcal{M}}(t)}$ approximates $\mathcal{P}_{\mathbf{x}_{0:i}}$, then this term is small, resulting in a high sample log-likelihood in Eq. (15). (2) By increasing the discrepancy among empirical memory distributions $\{\mathcal{P}_{\widehat{\mathcal{M}}(1)}, \cdots, \mathcal{P}_{\widehat{\mathcal{M}}(t)}\}$ helps $\mathcal{P}_{\mathcal{M}_i^{DS},\widehat{\mathcal{M}}(1):\widehat{\mathcal{M}}(t)}$ to preserve more information and therefore decrease the distance term $F_d\big(\mathcal{P}_{\mathbf{x}_{0:i}}, \mathcal{P}_{\mathcal{M}_i^{DS},\widehat{\mathcal{M}}(1):\widehat{\mathcal{M}}(t)}\big)$, resulting in better performance. (3) From Eq. (15), an optimal DSMS should have a small number t of empirical memory distributions characterized by high discrepancies between each other. To find the optimal DSMS in practice, we first define $\mathcal{P}_{\mathcal{M}(j)}$ as the explicit memory distribution (Gaussian) for each $\mathcal{P}_{\widehat{\mathcal{M}}(j)}$ using Eq. (4) and then formulate the memory expansion as a min-max constrained optimization problem, as in Eq. (5).

5. Experiments

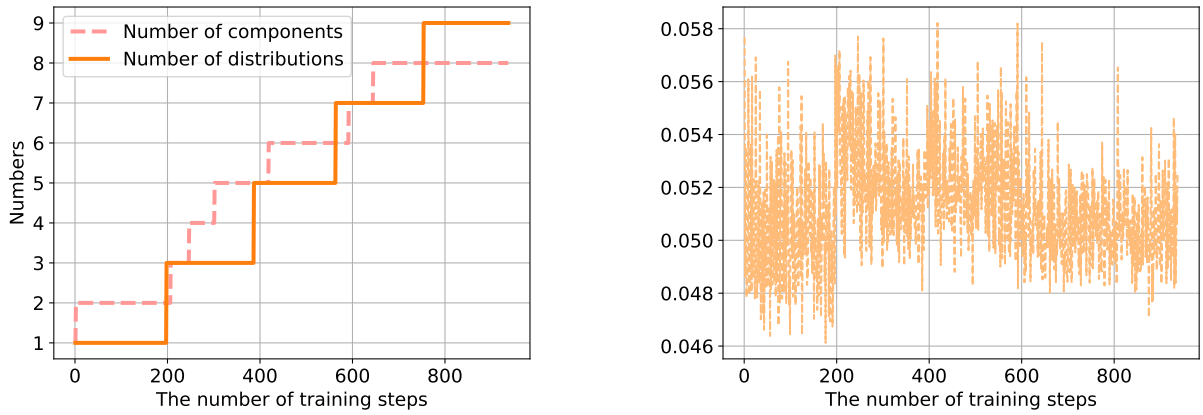
Datasets. We consider several datasets used for evaluating the performance in unsupervised continual learning [64], including MNIST [34], Fashion [60], SVHN [40] and CIFAR10 [32]. Each dataset is divided into five subsets by grouping two successive data categories [3], resulting in Split MNIST, Split Fashion, Split SVHN and Split CIFAR10. We resize each image from all datasets to $32 \times 32 \times 3$ pixels. In addition to the simple datasets, this paper also evaluates the model's performance using large-scale and complex-image datasets, including CACD [10], MINIImageNet [56], CelebA [36], 3DChair [5] and ImageNet [33]. We provide the additional information about the experiment setting in the **Appendix-B** from SM.

5.1. Class-Incremental Learning

In the class-incremental we train various models on the Split MNIST, Split Fashion, Split SVHN and Split CIFAR10, respectively, learning data from 2 categories (classes) during each task. By following the setting from [65], the maximum memory size for all models is set to 2,000 samples and the batch size as b=64. The results from Tab. 1, compare the proposed DEDM with Memory Replay GANs (MeRGANs) [59], Lifelong Teacher-Student (LTS), [63] and the Lifelong Generative Modeling (LGM), [44]. The Dynamic Cluster Memory (DCM) [65] uses a memory buffer for storing data, employing either the square error or the Jensen–Shannon (JS) divergence, resulting in the DCM-SE and DCM-JS models, respectively. The Continual Generative Knowledge Distillation (CGKD) [64] using GAN and Wasserstein Autoencoder (WAE) for replay data generation, are denoted as CGKD-GAN and CGKD-WAE, respectively. Although the dynamic expansion models such as CGKD-GAN can dynamically create new experts and freeze all previously learnt ones in order to maintain the performance on previous samples, their performance is lower than the state-of-the-art memory-based methods such as DCM-SE and DCM-JS. Such a performance gap is caused by two aspects : (1) DCM-SE and DCM-JS employ the Denoising Diffusion Probabilistic Model (DDPM) as generative model and can produce better generation results than CGKD-GAN; (2) Previously created experts in a dynamic expansion framework could not learn from new training samples before its parameters are frozen. The proposed DEMD outperforms other baselines on all datasets, as shown in Tab. 1.

5.2. The Imbalanced Class Setting

Class imbalance is a challenging problem in machine learning where there are sharp variations in the number of training data for various classes. Most models when trained on a



(a) The expansion process.

(b) The expansion signal.

Figure 2. Ablation study results. (a) The number of memory distributions and the data distributions (task ID) at each training time $\mathcal{T}_i$. (b) The expansion signal evaluated by the left hand side of Eq. (6).

class-imbalanced setting would lead to performance degeneration. In the following we consider the continual learning for the class-imbalanced setting. Specifically, the even-numbered categories are considered as classes with very few training data, assumed to be only 200 for each class. In addition, we compare with the current state-of-the-art such as DCM-SE and DCM-JS on Split MNIST, Split Fashion, Split SVHN and Split CIFAR10 datasets, respectively. The maximum memory size for all models is 2,000 and the results of the class-imbalance setting are shown in Tab. 2.

5.3. Results on Datasets with Complex images

We also evaluate the performance of the proposed approach on datasets containing complex images considering the domain-incremental scenario. Thus, we consider learning CelebA [36], CACD [10] and 3DChair [5] datasets. Following the setting from [65], we create two domain-incremental data by combining CelebA and 3DChair as CelebA-3DChair, while also considering CelebA and CACD as CelebA-CACD. The maximum memory size for all models is 2,000 and the results of various models are reported in Tab. 3, which shows that the proposed DEMD outperforms all baselines on complex datasets.

Following the setting from [65], we also consider evaluating the performance of various models on a challenging dataset such as the MINIImageNet [56] that can be used for few-shot learning [50]. The MINIImageNet [56] dataset which consists of images from 100 classes, which are divided into 64, 16, and 20 classes, respectively, corresponding to meta-training, meta-validation, and meta-testing in few-shot learning tasks. We consider to build a data stream, called the Split MINIImageNet (S-MINIImageNet) that combines the meta-training and meta-validation datasets. Specifically, the data stream is divided into 16 subsets and each part consists of samples from five successive classes. We train various models on the few-shot learning datasets and the results are reported in Tab. 3, which shows that the proposed approach outperforms all other baselines in the few-shot continual learning setting. Furthermore, we evaluate the performance on high-resolution image datasets, including CelebA-HQ [36], CACD [10] and FFHQ [28], respectively, and the results are provided in Tab. 4. According to these results, the proposed approach achieves the best performance on these high-resolution datasets.

5.4. Ablation Study

We perform a series of experiments to analyze the performance of the proposed approach under different configurations. More results can be found in **Appendix-C** from SM.

The memory expansion process. We investigate the dynamic expansion process of the DSMS by recording the number of memory buffers and the learned tasks at each training time. In Fig. 2a we provide the results after training the proposed approach on Split MNIST, considering the threshold for defining new memory buffers as $\lambda = 0.053$ in Eq. (6). The proposed approach adds, almost consistently, a new memory distribution for each new data category.

The dynamic signals. In Fig. 2b we evaluate the expansion signals using the expression from the left hand side of Eq. (6). The results show that a large expansion signal corresponds to the learning of a new data category, indicating that the proposed approach provides valid signals to guide the memory expansion process.

6. Conclusion

This paper addresses forgetting under the TFUC setting by proposing the Dynamic Expanding Memory Distribution (DEMD) framework that consists of memory systems DSMS for long-term information and a DMS for short-term memory. A novel memory expansion mechanism is proposed to incrementally increase the capacity of the DSMS for capturing critical new information. Empirical results demonstrate that the proposed approach achieves state-of-the-art performance.

References

[1] A. Achille, T. Eccles, L. Matthey, C. Burgess, N. Watters, A. Lerchner, and I. Higgins. Life-long disentangled representation learning with cross-domain latent homologies. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 9873–9883, 2018. 2

[2] Hongjoon Ahn, Sungmin Cha, Donggyu Lee, and Taesup Moon. Uncertainty-based continual learning with adaptive regularization. In *Advances in Neural Information Processing Systems*, pages 4394–4404, 2019. 2

[3] Rahaf Aljundi, Klaas Kelchtermans, and Tinne Tuytelaars. Task-free continual learning. In *Proc. of IEEE/CVF Conf. on Computer Vision and Pattern Recognition*, pages 11254–11263, 2019. 7

[4] R. Aljundi, M. Lin, B. Goujaud, and Y. Bengio. Gradient based sample selection for online continual learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 11817–11826, 2019. 1

[5] M. Aubry, D. Maturana, A. A. Efros, B. C. Russell, and J. Sivic. Seeing 3D chairs: exemplar part-based 2D-3D alignment using a large dataset of CAD models. In *Proc. of IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 3762–3769, 2014. 7, 8

[6] Jihwan Bang, Heesu Kim, YoungJoon Yoo, Jung-Woo Ha, and Jonghyun Choi. Rainbow memory: Continual learning with a memory of diverse samples. In *Proc. of IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8218–8227, 2021. 1

[7] Jihwan Bang, Hyunseo Koh, Seulki Park, Hwanjun Song, Jung-Woo Ha, and Jonghyun Choi. Online continual learning on a contaminated data stream with blurry task boundaries. In *Proc. of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9275–9284, 2022. 2

[8] Hyuntak Cha, Jaeho Lee, and Jinwoo Shin. Co2l: Contrastive continual learning. In *Proc. of the IEEE/CVF International Conference on Computer Vision (CVPR)*, pages 9516–9525, 2021. 2

[9] A. Chaudhry, M. Rohrbach, M. Elhoseiny, T. Ajanthan, P. Dokania, P. H. S. Torr, and M.'A. Ranzato. On tiny episodic memories in continual learning. *arXiv preprint arXiv:1902.10486*, 2019. 1, 2

[10] B.-C. Chen, C.-S. Chen, and W. H. Hsu. Cross-age reference coding for age-invariant face recognition and retrieval. In *Proc. European Conf on Computer Vision (ECCV), vol. LNCS 8694*, pages 768–783, 2014. 7, 8

[11] C. Cortes, X. Gonzalvo, V. Kuznetsov, M. Mohri, and S. Yang. Adanet: Adaptive structural learning of artificial neural networks. In *Proc. of Int. Conf. on Machine Learning (ICML), vol. PMLR 70*, pages 874–883, 2017. 1, 2

[12] Danruo Deng, Guangyong Chen, Jianye Hao, Qiong Wang, and Pheng-Ann Heng. Flattening sharpness for dynamic gradient projection memory benefits continual learning. *Advances in Neural Information Processing Systems (NeurIPS)*, 34:18710–18721, 2021. 2

[13] Mohammad Mahdi Derakhshani, Xiantong Zhen, Ling Shao, and Cees Snoek. Kernel continual learning. In *Proc. of International Conference on Machine Learning (ICLR), vol. PMLR 139*, pages 2621–2631, 2021.

[14] Shibhansh Dohare, J Fernando Hernandez-Garcia, Qingfeng Lan, Parash Rahman, A Rupam Mahmood, and Richard S Sutton. Loss of plasticity in deep continual learning. *Nature*, 632(8026):768–774, 2024. 2

[15] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representions (ICLR), arXiv preprint arXiv:2010.11929*, 2021. 3

[16] Arthur Douillard, Alexandre Ramé, Guillaume Couairon, and Matthieu Cord. Dytox: Transformers for continual learning with dynamic token expansion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9285–9295, 2022. 3

[17] Sebastian Farquhar and Yarin Gal. Towards robust evaluations of continual learning. *arXiv preprint arXiv:1805.09733*, 2018. 1

[18] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial nets. In *Advances in Neural Information Processing Systems (NIPS)*, pages 2672–2680, 2014. 2

[19] Yanan Gu, Xu Yang, Kun Wei, and Cheng Deng. Not just selection, but exploration: Online class-incremental continual learning via dual view consistency. In *Proc. of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7442–7451, 2022. 2

[20] Yiduo Guo, Bing Liu, and Dongyan Zhao. Online continual learning through mutual information maximization. In *International Conference on Machine Learning (ICLR), vol. PMLR 162*, pages 8109–8126, 2022. 2

[21] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proc. of IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016. 3

[22] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. GANs trained by a two time-scale update rule converge to a local Nash equilibrium. In *Advances in Neural Information Processing Systems (NIPS)*, pages 6626–6637, 2017. 3

[23] G. Hinton, O. Vinyals, and J. Dean. Distilling the knowledge in a neural network. In *Proc. NIPS Deep Learning Workshop, arXiv preprint arXiv:1503.02531*, 2014. 2

[24] Ching-Yi Hung, Cheng-Hao Tu, Cheng-En Wu, Chien-Hung Chen, Yi-Ming Chan, and Chu-Song Chen. Compacting, picking and growing for unforgetting continual learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 13647–13657, 2019. 1, 2

[25] Fushuo Huo, Wenchao Xu, Jingcai Guo, Haozhao Wang, and Yunfeng Fan. Non-exemplar online class-incremental continual learning via dual-prototype self-augment and refinement. In *Proc. of the AAAI Conference on Artificial Intelligence*, pages 12698–12707, 2024. 2

[26] Saurav Jha, Dong Gong, He Zhao, and Lina Yao. NPCL: Neural processes for uncertainty-aware continual learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 34329–34353, 2023. 2

[27] Haeyong Kang, Rusty John Lloyd Mina, Sultan Rizky Hikmawan Madjid, Jaehong Yoon, Mark Hasegawa-Johnson, Sung Ju Hwang, and Chang D Yoo. Forget-free continual learning with winning subnetworks. In *Proc. of International Conference on Machine Learning (ICLR), vol. PMLR 162*, pages 10734–10750, 2022. 2, 3

[28] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *Proc. of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4401–4410, 2019. 8

[29] Ronald Kemker, Marc McClure, Angelina Abitino, Tyler Hayes, and Christopher Kanan. Measuring catastrophic forgetting in neural networks. In *Proc. of the AAAI Conference on Artificial Intelligence*, pages 3390–3398, 2018. 1

[30] Junsu Kim, Hoseong Cho, Jihyeon Kim, Yihalem Yimolal Tiruneh, and Seungryul Baek. SDDGR: Stable diffusion-based deep generative replay for class incremental object detection. In *Proc. of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 28772–28781, 2024. 2

[31] D. P. Kingma and M. Welling. Auto-encoding variational Bayes. *arXiv preprint arXiv:1312.6114*, 2013. 6

[32] Alex Krizhevsky and Geoffrey Hinton. Learning multiple layers of features from tiny images. Technical report, Univ. of Toronto, 2009. 7

[33] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in Neural Inf. Proc. Systems (NIPS)*, pages 1097–1105, 2012. 7

[34] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proc. of the IEEE*, 86(11):2278–2324, 1998. 7

[35] Z. Li and D. Hoiem. Learning without forgetting. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 40(12): 2935–2947, 2017. 2

[36] Z. Liu, P. Luo, X. Wang, and X. Tang. Deep learning face attributes in the wild. In *Proc. of IEEE Int. Conf. on Computer Vision (ICCV)*, pages 3730–3738, 2015. 7, 8

[37] David Lopez-Paz and Marc'Aurelio Ranzato. Gradient episodic memory for continual learning. In *Advances in Neural Information Processing Systems (NIPS)*, pages 6467–6476, 2017. 2

[38] James Martens and Roger B. Grosse. Optimizing neural networks with kronecker-factored approximate curvature. In *Proc. International Conference on Machine Learning (ICML), vol. JMLR: W&CP 37*, pages 2408 – 2417, 2015. 1, 2

[39] James L McClelland, Bruce L McNaughton, and Randall C O'Reilly. Why there are complementary learning systems in the hippocampus and neocortex: insights from the successes and failures of connectionist models of learning and memory. *Psychological review*, 102(3):419, 1995. 2

[40] Y. Netzer, T. Wang, A. Coates, A. Bissacco, B. Wu, and A. Y. Ng. Reading digits in natural images with unsupervised feature learning. In *NIPS Workshop on Deep Learning and Unsupervised Feature Learning*, 2011. 7

[41] Cuong V Nguyen, Yingzhen Li, Thang D Bui, and Richard E Turner. Variational continual learning. In *International Conference on Learning Representations (ICLR), arXiv preprint arXiv:1710.10628*, 2018. 2

[42] G. I. Parisi, R. Kemker, J. L. Part, C. Kanan, and S. Wermter. Continual lifelong learning with neural networks: A review. *Neural Networks*, 113:54–71, 2019. 1

[43] R. Polikar, L. Upda, S. S. Upda, and Vasant Honavar. Learn++: An incremental learning algorithm for supervised neural networks. *IEEE Trans. on Systems Man and Cybernetics, Part C*, 31(4):497–508, 2001. 2

[44] J. Ramapuram, M. Gregorova, and A. Kalousis. Lifelong generative modeling. In *International Conference on Learning Representations (ICLR), arXiv preprint arXiv:1705.09847*, 2017. 2, 7

[45] Sylvestre-Alvise Rebuffi, Alexander Kolesnikov, Georg Sperl, and Christoph H Lampert. iCaRL: Incremental classifier and representation learning. In *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 2001–2010, 2017. 2

[46] B. Ren, H. Wang, J. Li, and H. Gao. Life-long learning based on dynamic combination model. *Applied Soft Computing*, 56:398–404, 2017. 2

[47] Hippolyt Ritter, Aleksandar Botev, and David Barber. Online structured Laplace approximations for overcoming catastrophic forgetting. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 3742–3752, 2018. 2

[48] Andrei A Rusu, Neil C Rabinowitz, Guillaume Desjardins, Hubert Soyer, James Kirkpatrick, Koray Kavukcuoglu, Razvan Pascanu, and Raia Hadsell. Progressive neural networks. *arXiv preprint arXiv:1606.04671*, 2016. 2

[49] T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Radford, and X. Chen. Improved techniques for training GANs. In *Advances in Neural Information Processing Systems (NIPS)*, pages 2234–2242, 2016. 3

[50] Edgar Schönfeld, Sayna Ebrahimi, Samarth Sinha, Trevor Darrell, and Zeynep Akata. Generalized zero- and few-shot learning via aligned variational autoencoders. In *Proc. of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8247–8255, 2019. 8

[51] Yujun Shi, Li Yuan, Yunpeng Chen, and Jiashi Feng. Continual learning via bit-level information preserving. In *Proc. of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 16674–16683, 2021. 2

[52] H. Shin, J. K. Lee, J. Kim, and J. Kim. Continual learning with deep generative replay. In *Advances in Neural Inf. Proc. Systems (NIPS)*, pages 2990–2999, 2017. 2

[53] Shengyang Sun, Daniele Calandriello, Huiyi Hu, Ang Li, and Michalis K. Titsias. Information-theoretic online memory selection for continual learning. In *International Conference on Learning Represenations (ICLR), arXiv preprint arXiv:2204.04763*, 2022. 1

[54] Rishabh Tiwari, Krishnateja Killamsetty, Rishabh Iyer, and Pradeep Shenoy. GCR: Gradient coreset based replay buffer

selection for continual learning. In *Proc. of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 99–108, 2022. 2

[55] Vinay Kumar Verma, Kevin J Liang, Nikhil Mehta, Piyush Rai, and Lawrence Carin. Efficient feature transformations for discriminative and generative continual learning. In *Proc. of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 13865–13875, 2021. 2

[56] Oriol Vinyals, Charles Blundell, Timothy Lillicrap, Koray Kavukcuoglu, and Daan Wierstra. Matching networks for one shot learning. *Advances in Neural Information Processing Systems (NIPS)*, 29:3637–3645, 2016. 7, 8

[57] Shipeng Wang, Xiaorong Li, Jian Sun, and Zongben Xu. Training networks in null space of feature covariance for continual learning. In *Proc. of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 184–193, 2021. 2

[58] Yeming Wen, Dustin Tran, and Jimmy Ba. BatchEnsemble: an alternative approach to efficient ensemble and lifelong learning. In *International Conference on Learning Representations (ICLR), arXiv preprint arXiv:2002.06715*, 2020. 2

[59] C. Wu, L. Herranz, X. Liu, J. van de Weijer, and B. Raducanu. Memory replay GANs: Learning to generate new categories without forgetting. In *Advances In Neural Inf. Proc. Systems (NeurIPS)*, pages 5962–5972, 2018. 7

[60] H. Xiao, K. Rasul, and R. Vollgraf. Fashion-MNIST: a novel image dataset for benchmarking machine learning algorithms. *arXiv preprint arXiv:1708.07747*, 2017. 7

[61] Mengqi Xue, Haofei Zhang, Jie Song, and Mingli Song. Meta-attention for vit-backed continual learning. In *Proc. of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 150–159, 2022. 3

[62] Fei Ye and Adrian G. Bors. Learning latent representations across multiple data domains using lifelong VAEGAN. In *Proc. European Conf. on Computer Vision (ECCV), vol. LNCS 12365*, pages 777–795, 2020. 1

[63] Fei Ye and Adrian G. Bors. Lifelong teacher-student network learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(19):6280–6296, 2022. 7

[64] Fei Ye and Adrian G. Bors. Continual variational autoencoder via continual generative knowledge distillation. In *Proc. AAAI Conference on Artificial Intelligence*, pages 10918–10926, 2023. 3, 7

[65] Fei Ye and Adrian G Bors. Online task-free continual generative and discriminative learning via dynamic cluster memory. In *Proc. of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 26202–26212, 2024. 7, 8

[66] F. Zenke, B. Poole, and S. Ganguli. Continual learning through synaptic intelligence. In *Proc. of International Conference on Machine Learning, vol. PLMR 70*, pages 3987–3995, 2017. 3

[67] M. Zhai, L. Chen, F. Tung, J He, M. Nawhal, and G. Mori. Lifelong GAN: Continual learning for conditional image generation. In *Proc. of the IEEE/CVF Int. Conf. on Computer Vision (ICCV)*, pages 2759–2768, 2019. 2

[68] Guanyu Zhou, Kihyuk Sohn, and Honglak Lee. Online incremental feature learning with denoising autoencoders. In *Proc. Int. Conf. on Artificial Intelligence and Statistics (AISTATS), vol. PMLR 22*, pages 1453–1461, 2012. 2