Online Task-Free Continual Generative and Discriminative Learning via Dynamic Cluster Memory

Fei Ye[1,2] and Adrian G. Bors[1,2]
[1]Department of Computer Science, University of York, York YO10 5GH, UK
[2]Machine Learning Dept., Mohamed bin Zayed University of Artificial Intelligence, Abu Dhabi, UAE
122404504@qq.com, adrian.bors@york.ac.uk

Abstract

Online Task-Free Continual Learning (OTFCL) aims to learn novel concepts from streaming data without accessing task information. Most memory-based approaches used in OTFCL are not suitable for unsupervised learning because they require accessing supervised signals to implement their sample selection mechanisms. In this study, we address this issue by proposing a novel memory management approach, namely the Dynamic Cluster Memory (DCM), which builds new memory clusters to capture distribution shifts over time without accessing any supervised signals. DCM introduces a novel memory expansion mechanism based on the knowledge discrepancy criterion, which evaluates the novelty of the incoming data as the signal for the memory expansion, ensuring a compact memory capacity. We also propose a new sample selection approach that automatically stores incoming data samples with similar semantic information in the same memory cluster, while also facilitating the knowledge diversity among memory clusters. Furthermore, a novel memory pruning approach is proposed to automatically remove overlapping memory clusters through a graph relation evaluation, ensuring a fixed memory capacity while maintaining the diversity among the samples stored in the memory. The proposed DCM is model-free, plug-and-play, and can be used in both supervised and unsupervised learning without modifications. Empirical results on OTFCL experiments show that the proposed DCM outperforms the state-of-the-art while requiring fewer data samples to be stored. The source code is available at https://github.com/dtuzi123/DCM.

1. Introduction

Modern deep learning models have achieved remarkable results in a wide range of applications [9, 22, 52], but their success mainly relies on the accessibility of the large-scale dataset [26, 33]. However, when these advanced models are applied for learning a sequence of tasks, they will completely forget previously learned information, when learning new data. This results in degenerated performance on the tasks learnt in the past [76]. Such a learning paradigm is called continual learning, and catastrophic forgetting plays a major factor in deteriorating model's performance [45].

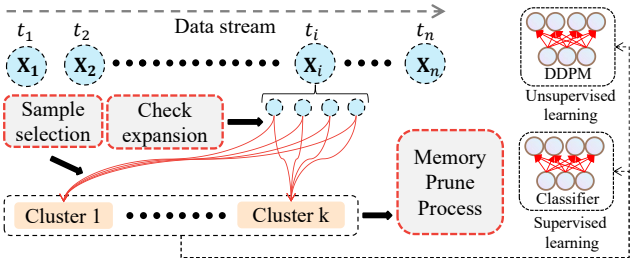


Figure 1. Illustration of the Dynamic Cluster Memory (DCM) allocation management. The novelty of each incoming sample is evaluated at each time t_i. Incoming data samples are selectively accumulated in corresponding memory clusters using a novelty criterion. To avoid overloading, a memory pruning process is used for automatically removing clusters representing overlapping information. The proposed DCM can be implemented in both supervised and unsupervised learning without any modifications.

Most existing works in continual learning [75] can be roughly classified into three branches: regularisation-based approaches [31], memory/experience replay [7] and dynamic network architectures [50]. Although some of these methods achieve promising results in continual learning, they still require accessing task information to implement their underlying methods that deal with new samples. Recently, the Online Task-Free Continual Learning (OTFCL) [4], in which a model can only access a small data batch at a time without knowing task information, was found to be a more challenging and realistic learning paradigm. Using a fixed-capacity memory buffer to store critical samples was shown to be effective in OTFCL [4, 27, 70]. However, most memory-based methods require supervised signals from the model trained on the labelled samples for implementing sample selection mechanisms [27, 70], limiting their applicability in unsupervised learning.

Image synthesis represents a fundamental application in unsupervised learning under OTFCL, and this has not yet been well studied in the context of continual learning, [71]. Image synthesis is challenging when using a fixed memory buffer due to the difficulty of preserving the entire category/domain information in the absence of supervised signals. In this study, we simultaneously address both classification and image synthesis under OTFCL by designing a novel memory approach from an entirely different angle. As illustrated in Fig. 1, we propose to accumulate incoming data samples with similar semantic information into the same memory clusters through a nonparametric data similarity evaluation, aiming to capture unique underlying data distributions while appropriately managing the memory resource allocation. For the data selection, we define a Knowledge Discrepancy Measure (KDM) criterion that estimates the knowledge discrepancy on a pair of samples in a nonparametric manner. In addition, we design each memory cluster as a fixed-capacity memory block, aiming to accumulate similar data samples over time in the same memory buffer. To adapt to the data distribution shifts over time, a novel memory dynamic expansion mechanism is proposed to evaluate the novelty of the incoming data using KDM. A large KDM measure indicates that the incoming data is novel enough and then we build a new memory cluster to preserve the novel samples during the subsequent learning. In addition, storing incoming samples in appropriate memory clusters can benefit memory optimization and management. To this end, we propose a novel sample selection approach to evaluate the relationship between the incoming data and each memory cluster using the KDM. This measure can guide storing incoming samples into corresponding memory clusters, ensuring the knowledge diversity among the memory clusters.

We have to consider that when using a resource-constrained machine, it is necessary to limit the memory capacity. To this end, we propose a novel Memory Pruning Process (MPP) to automatically remove the overlapping information from the memory, when the DCM is overloaded. Specifically, the MPP first identifies a pair of memory clusters that share significantly similar information through a graph relation evaluation. Then, one of these memory buffers is removed by using a diversity evaluation mechanism. Such an approach can avoid overloading the DCM while preserving diverse information as much as possible using a compact memory capacity. In a new direction of research, we explore the proposed DCM to train the DDPM model for implementing image generation under OTFCL, in the Denoising Diffusion Probabilistic Models (DDPM) [24], which represents a recently developed generative model showing remarkable image generation results.

We summarize our contributions as follows : (1) We propose the Dynamic Cluster Memory (DCM), a new memory management approach which can store diverse data samples without interacting with the model optimization and can thus be applied in both supervised and unsupervised learning; (2) The proposed DCM is plug-and-play and can be used in the context of training different models in OTFCL without modifications; (3) We propose a novel memory pruning approach to automatically remove overlapping memory clusters ensuring a fixed memory capacity, which can be used in a resource-constrained machine. (4) To our best knowledge, this paper is the first work to explore the potential advantage of DDPM in OTFCL by using a dynamic memory management allocation system.

2. Related Work

Continual Learning. Using a fixed-capacity memory buffer to preserve previous training samples was shown to reduce network forgetting in continual learning [7, 8, 11, 20, 44, 49, 55, 61, 62, 65]. Sample selection is usually considered for filtering out samples considered as unimportant preserving only those deemed critical. Memory-based approaches can also be combined with regularization [2, 5, 12, 13, 16–18, 21, 23, 25, 28, 29, 31, 39, 40, 42, 53, 59, 60, 73, 74] and knowledge distillation [10] based methods to improve model performance further. In addition to memory-based methods, several works [1, 47, 48, 54, 76] have explored training a generator such as a Variational Autoencoder (VAE) [30] or a Generative Adversarial Net (GAN) [19], to remember and replay past samples which can then be used to retrain the model in order to combat forgetting. However, the performance of these models is highly affected by the quality of generative replay samples, as shown in [67].

Online Task-Free CL. A real-time application system can deal with streaming data online without knowing task information. In the first such approach [4], a small memory buffer was used for storing some past samples for training later a classifier. This work was then extended to a new memory buffer based approach, called the Maximal Interfered Retrieval (MIR) [3], which can train classifiers through a retrieval mechanism. The Gradient Sample Selection (GSS), which is a constrained optimization problem, was proposed in [5] for selecting the samples to be preserved in the memory buffer. More recently, a memorization-based approach was implemented within a *learner-evaluator* framework, called the Continual Prototype Evolution (CoPE) that ensures the preservation of diverse samples for each task [15]. The Gradient-based Memory EDiting (GMED) [27] modifies the memorized samples, aiming to improving performance [27]. Recently, several attempts have been explored to develop a Dynamic Expansion Model (DEM) approach to address OTFCL, as in the Continual Unsupervised Representation Learning (CURL) [48] which builds new inference modules to capture new experiences from incoming data. Another DEM-

based approach considers a Dirichlet process-based expansion mechanism which automatically increases the model's capacity [37]. However, these approaches require more parameters and additional inference time during the testing phase compared to the memory-based methods [72].

Continual Generative Modelling. Training a model for image generation in continual learning was studied in recent studies [1, 47, 69]. The pioneering research study from [1] relied on a VAE-based framework which enables the generative model to produce images from past tasks without forgetting. Nevertheless, VAEs are rather poor data generation networks producing rather blurred images [35]. Consequently, the quality of generative replay samples is decreased, resulting in poor performance. This was addressed by using a more powerful generative model such as the Generative Adversarial Network (GAN) [47, 63, 68, 76] as a generative replay network. However, these models require knowing the task information, which is unavailable in OT-FCL. Recently, the Continual Generative Knowledge Distillation (CGKD) [71] uses a dynamic expansion model to sustain image generation under continual learning. CGKD employs a Fréchet Inception Distance (FID)-based expansion criterion to regulate model expansion, requiring the generation of a considerable number of samples. Such an approach requires substantial computational costs when implementing each component of CGKD as a DDPM. The proposed DCM can train the DDPM more efficiently since it neither requires model feedback nor sampling as in DDPM. We provide additional related information in **Appendix-A** from Supplemental Material (SM).

3. Method

3.1. Preliminary

Problem statement. In unsupervised learning we have an unlabelled training dataset (the c-th dataset from a sequence of datasets) $\mathcal{D}_c^S = \{\mathbf{x}_j\}_{j=1}^{N_c^S}$ and a testing dataset $\mathcal{D}_c^T = \{\mathbf{x}_j\}_{j=1}^{N_c^T}$, respectively, where N_c^S and N_c^T are the total number of training and testing samples respectively. We split the training dataset into C parts $\mathcal{D}_c^S = \{\mathcal{D}_{c,1}^S, \cdots, \mathcal{D}_{c,C}^S\}$ according to the online continual learning setting from [71]. A learning data stream $\mathcal{S}$ is formed by arranging these subsets in a sequence $\mathcal{S} = \{\mathcal{D}_{c,1}^S \bigcup \mathcal{D}_{c,2}^S \bigcup \cdots \mathcal{D}_{c,C}^S\}$. Let $\mathbf{t} = \{t_1, \cdots, t_n\}$ be a series of time intervals that split $\mathcal{S}$ into n non-overlapping data batches $\{\mathbf{X}_i, \cdots, \mathbf{X}_n\}$. At a certain learning time (t_i), the model can only access the data batch $\mathbf{X}_i$, consisting of b samples while all previously seen data batches $\{\mathbf{X}_1, \cdots, \mathbf{X}_{i-1}\}$ are unavailable. After the model finishes all n training times, its performance is evaluated on the testing dataset $\mathcal{D}_c^T$ using the image generation performance criterion. In addition to learning a single dataset/domain, this paper also studies a more challenging paradigm in which a data stream $\mathcal{S}$ is built by arranging together several different datasets in a sequence manner $\mathcal{S} = \{\mathcal{D}_1^S \bigcup \cdots, \bigcup \mathcal{D}_m^S\}$ where m is the number of datasets being considered. We also study supervised learning in which class labels are provided during training.

Diffusion model. The Denoising Diffusion Probabilistic Model (DDPM) emerged as the most popular image-generative model. DDPM is defined by two diffusion processes: a forward diffusion procedure that processes data and transfers it to the noise vector and a reverse diffusion algorithm that recovers the image from the noise vector, [24, 66]. In the forward diffusion process, a series of noise-processing operations is performed to add noise to the data:

$$q(\tilde{\mathbf{x}}_{1:T} \,|\, \tilde{\mathbf{x}}_0) = \prod_{t=1}^{T} q(\tilde{\mathbf{x}}_t \,|\, \tilde{\mathbf{x}}_{t-1}) \,, \tag{1}$$

where $\tilde{\mathbf{x}}_0$ is the original data sampled from an empirical data distribution $p(\tilde{\mathbf{x}}_0)$. T is a predefined number of diffusion steps and $q(\tilde{\mathbf{x}}_t \,|\, \tilde{\mathbf{x}}_{t-1}) = \mathcal{N}(\tilde{\mathbf{x}}_t; \sqrt{1-\beta_t}\tilde{\mathbf{x}}_{t-1}, \beta_t \mathbf{I})$. $\{\beta_t \in (0,1) \,|\, t = 1, \cdots, T\}$ is a variance schedule used to regulate the diffusion step size. By using Eq. (1), we can generate a series of noise images $\{\tilde{\mathbf{x}}_1, \cdots, \tilde{\mathbf{x}}_T\}$.

Estimating the backward diffusion process $q(\tilde{\mathbf{x}}_{t-1} \,|\, \tilde{\mathbf{x}}_t)$ remains challenging because it requires accessing the entire dataset. This intractable optimization can be solved by training a model $p_\theta(\tilde{\mathbf{x}}_{t-1} \,|\, \tilde{\mathbf{x}}_t)$ parameterized by θ and then the backward process is expressed as :

$$p_\theta(\tilde{\mathbf{x}}_{0:T}) = p(\tilde{\mathbf{x}}_T) \prod_{t=1}^{T} p_\theta(\tilde{\mathbf{x}}_{t-1} \,|\, \tilde{\mathbf{x}}_t) \,, \tag{2}$$

where $p_\theta(\tilde{\mathbf{x}}_{t-1} \,|\, \tilde{\mathbf{x}}_t) = \mathcal{N}(\tilde{\mathbf{x}}_{t-1}; \boldsymbol{\mu}_\theta(\tilde{\mathbf{x}}_t, t), \boldsymbol{\Sigma}_\theta(\tilde{\mathbf{x}}_t, t))$ and $p(\tilde{\mathbf{x}}_T) = \mathcal{N}(\mathbf{0}, \mathbf{I})$. $\boldsymbol{\Sigma}_\theta(\cdot,\cdot)$ and $\boldsymbol{\mu}_\theta(\cdot,\cdot)$ are trainable functions implemented by a deep learning model. Training a diffusion model involves a noise estimator model $\boldsymbol{\epsilon}_\theta(\cdot,\cdot)$ that predicts $\boldsymbol{\epsilon}$ from $\tilde{\mathbf{x}}_t$ and t, [24]. We consider the Improved DDPM (IDDPM) objective function [43] for training $\boldsymbol{\epsilon}_\theta$:

$$\begin{aligned} \mathcal{L}_{\text{IDDPM}} = \lambda \mathbb{E}_{q(\tilde{\mathbf{x}}_{0:T} \,|\, \tilde{\mathbf{x}}_0)} \Big[& -\log p_\theta(\tilde{\mathbf{x}}_0 \,|\, \tilde{\mathbf{x}}_1) \\ & + \sum_{t=1}^{T-2} \{\mathcal{L}_t\} + D_{KL}[q(\tilde{\mathbf{x}}_T \,|\, \tilde{\mathbf{x}}_0) \,||\, p_\theta(\tilde{\mathbf{x}}_T)] \Big] \\ & + \mathbb{E}_{t, \tilde{\mathbf{x}}_0, \boldsymbol{\epsilon}} \left[\|\boldsymbol{\epsilon} - \boldsymbol{\epsilon}_\theta(\sqrt{\widehat{\alpha}_t}\tilde{\mathbf{x}}_0 + \sqrt{1-\widehat{\alpha}_t}\boldsymbol{\epsilon}, t)\|^2 \right] , \end{aligned} \tag{3}$$

where $\mathcal{L}_t$ is defined as :

$$\mathcal{L}_t = D_{KL}\left[q(\tilde{\mathbf{x}}_t \,|\, \tilde{\mathbf{x}}_{t+1}, \tilde{\mathbf{x}}_0) \,||\, p_\theta(\tilde{\mathbf{x}}_t \,|\, \tilde{\mathbf{x}}_{t+1})\right] , \tag{4}$$

where $\tilde{\mathbf{x}}_T$ is a noise vector drawn from $\mathcal{N}(\mathbf{0}, \mathbf{I})$. $q(\tilde{\mathbf{x}}_T \,|\, \tilde{\mathbf{x}}_0)$ and $p_\theta(\tilde{\mathbf{x}}_0 \,|\, \tilde{\mathbf{x}}_1)$ are the distributions defined within the forward and backward diffusion process, respectively. $D_{KL}(\cdot,\cdot)$ is the Kullback–Leibler (KL) divergence and

$q(\tilde{\mathbf{x}}_t \,|\, \tilde{\mathbf{x}}_{t+1}, \tilde{\mathbf{x}}_0)$ is defined as, [24] :

$$q(\tilde{\mathbf{x}}_t \,|\, \tilde{\mathbf{x}}_{t+1}, \tilde{\mathbf{x}}_0) = \mathcal{N}(\tilde{\mathbf{x}}_t; \frac{\sqrt{\widehat{\alpha}_t}\beta_{t+1}}{1-\widehat{\alpha}_t}\tilde{\mathbf{x}}_0 + \frac{\sqrt{\widehat{\alpha}_{t+1}}(1-\widehat{\alpha}_t)}{1-\widehat{\alpha}_t}\tilde{\mathbf{x}}_{t+1}, \hat{\beta}_{t+1}\mathbf{I}) \,, \tag{5}$$

where $\hat{\beta}_{t+1} = \frac{(1-\widehat{\alpha}_t)}{1-\widehat{\alpha}_{t+1}}\beta_{t+1}$. $\widehat{\alpha}_t := \prod_{s=1}^{t}\{\alpha_s\}$ and $\alpha_t := 1-\beta_t$. $\boldsymbol{\epsilon}$ is a random vector drawn from $\mathcal{N}(\mathbf{0}, \mathbf{I})$. We consider $\lambda = 0.001$ in the experiments to ensure that the first term from (3) does not overwhelm the second term [43].

3.2. Dynamic Clustering Memory System

Most existing continuous learning memory buffer-based approaches usually require access to the class/task information [34, 46] or rely on the feedback from the classifier's optimization procedure [61], and this cannot be considered in unsupervised learning. Next, we introduce a new memory approach, which can train arbitrary models for either supervised or unsupervised learning under the Online Task-Free Continual Learning (OTFCL) assumption without modifications. Since both task and class information are absent during each training stage, storing diverse samples in a restricted memory buffer plays a crucial role in relieving forgetting. We propose to use a dynamic cluster memory system which dynamically builds new memory clusters to capture data distribution shifts over time.

Let $\mathcal{M}$ be a dynamic memory system that is assumed to have built k memory clusters $\mathcal{M} = \{\mathcal{M}^1, \cdots, \mathcal{M}^k\}$ at t_i, where each memory cluster $\mathcal{M}^j$ has a fixed memory size π (representing the number of stored samples). We can identify a prototype sample $\widehat{\mathbf{x}}^j$ that represents a compact knowledge representation for each memory cluster $\mathcal{M}^j$, thus significantly reducing computational costs when evaluating the distance between memory clusters. Before introducing the dynamic memory expansion, we first define a Knowledge Discrepancy Measure (KDM) to estimate the discrepancy score among memory clusters. For a given pair of data samples $\{\mathbf{x}_g, \mathbf{x}_h\}$, the knowledge discrepancy measure between $\mathbf{x}_g$ and $\mathbf{x}_h$ is defined as:

$$F_{\mathrm{KDM}}(\mathbf{x}_g, \mathbf{x}_h) = F_d(G_t(\mathbf{x}_g), G_t(\mathbf{x}_h)) \,, \tag{6}$$

where $G_t(\cdot)$ is an information representation function. $F_d(\cdot)$ is a distance measure function which can be implemented by means of a loss function or probability distance. By using Eq. (6), we can flexibly design different KDM criteria by only changing $G_t(\cdot)$ and $F_d(\cdot)$. In the following, we introduce two practical distance measures using the square loss and Jensen–Shannon divergence based on KDM.

Square Error (SE)-based KDM. The square error is a basic distance measure widely used as the objective function of the Variational Autoencoder (VAE [30]) and DDPM [24] when minimizing the distance between inputs and predictions. In this paper, we implement the function $G_t(\cdot)$ using an identity function that returns the original input $\mathbf{x}_g = G_t(\mathbf{x}_g)$ and $F_d(\cdot,\cdot)$ as the square loss function. Then the SE-based KDM is defined as:

$$F^{\mathrm{SE}}_{\mathrm{KDM}}(\mathbf{x}_g, \mathbf{x}_h) = \sum\nolimits_{j=1}^{d'} (\mathbf{x}_g[j] - \mathbf{x}_h[j])^2 \,, \tag{7}$$

where $\mathbf{x}_g[j]$ is the j-th dimension of $\mathbf{x}_g$ and d' is the data dimension. This approach represents the Dynamic Cluster Memory with Square Error selection (DCM-SE) model.

Jensen–Shannon divergence (JS)-based KDM. The JS divergence is a probability distance, which is used to evaluate the similarity between two probability distributions [30]. However, the data samples $\mathbf{x}_g$ and $\mathbf{x}_h$ are usually drawn from two empirical data distributions $p(\mathbf{x}_g)$ and $p(\mathbf{x}_h)$, estimating the JS divergence between $p(\mathbf{x}_g)$ and $p(\mathbf{x}_h)$ is intractable due to the lack of the explicit density function form. We address this by designing the information representation function $G_t(\cdot)$ as a forward diffusion process that transfers the data $\mathbf{x}_g$ to a Gaussian distribution $q(\tilde{\mathbf{x}}_g^{t^\star} \,|\, \mathbf{x}_g) = \mathcal{N}(\tilde{\mathbf{x}}_g^{t^\star}; \sqrt{\widehat{\alpha}_{t^\star}}\mathbf{x}_g, (1-\widehat{\alpha}_{t^\star})\mathbf{I})$, where $t^\star$ is a hyper-parameter to control the forward steps. In practice, we consider a small $t^\star$ to allow the transformed distribution $q(\tilde{\mathbf{x}}_g^{t^\star} \,|\, \mathbf{x}_g)$ to closely approximate the real data distribution $p(\mathbf{x}_g)$. Then we implement the distance measure function $F_d(\cdot,\cdot) = D_{\mathrm{JS}}(\cdot \,||\, \cdot)$ as the JS divergence :

$$F^{\mathrm{JS}}_{\mathrm{KDM}}(\mathbf{x}_g, \mathbf{x}_h) = D_{\mathrm{JS}}[q(\tilde{\mathbf{x}}_g^{t^\star} \,|\, \mathbf{x}_g) \,||\, q(\tilde{\mathbf{x}}_h^{t^\star} \,|\, \mathbf{x}_h)] \,. \tag{8}$$

In the following, we introduce a novel dynamic memory optimization approach to adaptively expand the memory based on the definition of KDM. When implementing this approach we have the Dynamic Cluster Memory with Jensen-Shannon selection (DCM-JS) model.

3.3. Dynamic Memory Optimization

In order to capture the data distribution shift during the training, it is crucial to develop a strategy to dynamically build new memory clusters into the memory buffer system. Each newly created cluster should be designed such that it represents information which is distinct enough from that represented by any of the other clusters, thus ensuring a compact memory system. In order to achieve this goal, we evaluate the distance between each incoming sample and every sample from each existing memory cluster using the KDM from Eq. (6), and such a measure is then used as the signal for memory expansion. This evaluation is computationally expensive considering that each memory cluster has a considerable number of samples. However, we can evaluate the distance between memory clusters by considering the prototypes (usually the centre) $\widehat{\mathbf{x}}^k$ for each memory cluster $\mathcal{M}^k$. Firstly, we consider the cluster prototype $\widehat{\mathbf{x}}^k$ as the nearest neighbour to each memory cluster $\mathcal{M}^k$ through :

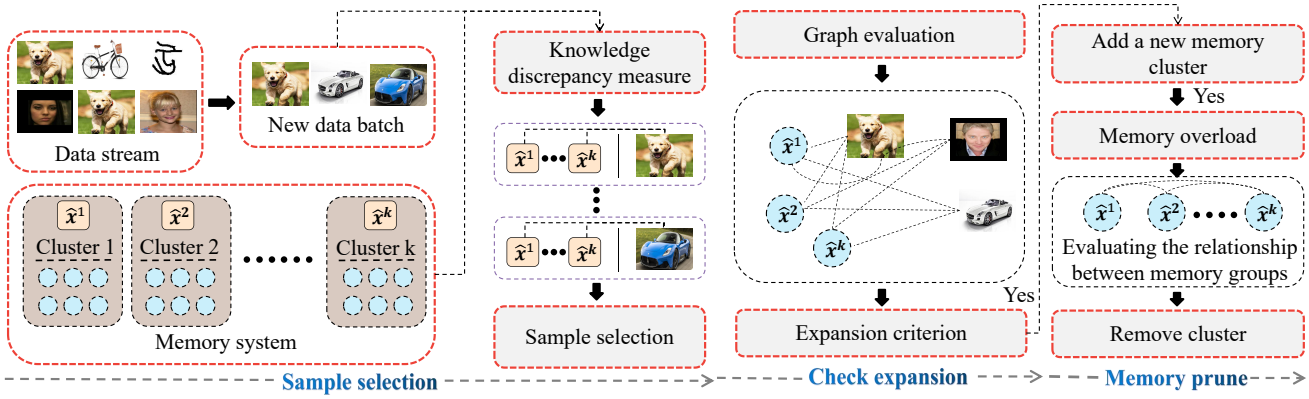


Figure 2. The optimization process of the proposed memory clustering system, consisting of three steps within a learning time. In the first step (sample selection), each incoming sample is compared with the central sample of each memory cluster using Eq. (6) to selectively store the new sample into an appropriate memory cluster. In the second step (memory expansion), if the expansion criterion, defined in Eq. (10) is satisfied, we build a new memory cluster. When $\mathcal{M}$ is overloaded, $|\mathcal{M}| > \rho$, the third step removes overlapping memory clusters.

$$c^\star = \underset{c=1,\cdots,|\mathcal{M}^k|}{\arg\min} \sum_{j=1, c\neq j}^{|\mathcal{M}^k|} F_{\text{KDM}}(\mathcal{M}^k[c], \mathcal{M}^k[j]), \quad (9)$$

where $|\mathcal{M}^k|$ is the number of samples for $\mathcal{M}^k$. We use $\widehat{\mathbf{x}}^k = \mathcal{M}^k[c^\star]$ to denote the cluster prototype as the nearest neighbour to the central sample that has the shortest distance to all other data from $\mathcal{M}^k$. We only identify a prototype once for each memory cluster using Eq. (9) to reduce the overall computational cost. By considering the prototype (central) sample for each memory cluster, the memory expansion criterion at the time t_i is efficiently evaluated by :

$$\max_{j=1,\cdots,b} \{F_{\text{KDM}}(\mathbf{x}_{i,j}, \widehat{\mathbf{x}}^1), \cdots, F_{\text{KDM}}(\mathbf{x}_{i,j}, \widehat{\mathbf{x}}^k)\} > \lambda, \quad (10)$$

where $\mathbf{x}_{i,j}$ is the j-th data sample of the data batch $\mathbf{X}_i$ drawn from S at t_i and λ is an expansion threshold (See the hyperparameter selection process in **Appendices-C10,C4**. By using Eq. (10), we aim to store a diversity of information in the memory clusters. A small λ encourages to frequently add new memory clusters during the training, while a large λ has an inverse effect. If the memory expansion criterion, defined in Eq. (10), is fulfilled at t_i, we build a new memory cluster $\mathcal{M}^{k+1}$ and use $\mathbf{x}_{i,j^\star}$ as the prototype for $\mathcal{M}^{k+1}$, where $j^\star = \arg\max_{j=1,\cdots,b}\{F_{\text{KDM}}(\mathbf{x}_{i,j}, \widehat{\mathbf{x}}^1), \cdots, F_{\text{KDM}}(\mathbf{x}_{i,j}, \widehat{\mathbf{x}}^k)\}$.

3.4. Sample Selection

Most existing memory-based continual learning approaches [34, 46] focus on supervised learning or require accessing the task information to implement sample selection strategies [8, 20, 21, 55]. Next, we introduce a new sample selection approach that does not need any supervised signals or model feedback. We assume that we have built a set of k memory clusters $\mathcal{M} = \{\mathcal{M}^1, \cdots, \mathcal{M}^k\}$ at time t_i. When

Algorithm 1: Algorithm for the proposed DCM.

for $t_i < t_n$ **do**
  if $|\mathcal{M}| == 0$ **then**
    $\mathcal{M}^1 = \mathbf{X}_1$ Build a new memory cluster ;
    Identity $\widehat{\mathbf{x}}^1$ using Eq. (9) ;
  else
    for $j < b$ **do**
      Sample Selection;
      $c^\star = \underset{c=1,\cdots,k}{\arg\max}\{F_{\text{KDM}}(\mathbf{x}_{i,j}, \widehat{\mathbf{x}}^c)\}$;
      $\mathcal{M}^{c^\star} = \mathcal{M}^{c^\star} \bigcup \mathbf{x}_{i,j}$;
      Remove redundant samples by Eq. (12) ;
    Check the memory expansion ;
    if *Eq.* (10) *is fulfilled* **then**
      Build a new memory cluster $\mathcal{M}^{k+1}$;
      Set $\mathbf{x}_{i,j}$ as the central sample $\widehat{\mathbf{x}}^{k+1}$;
      Memory Prune Process;
      if $|\mathcal{M}| > \rho$ **then**
        Remove overlapped clusters using Eq. (13) and Eq. (14) ;
  Training process;
  Update $\epsilon_\theta(\cdot,\cdot)$ on $\mathcal{M}$ using Eq. (3);

seeing an incoming data batch $\mathbf{X}_i$ at t_i, we evaluate the distance between each data sample $\mathbf{x}_{i,j} \in \mathbf{X}_i$, $j = 1, \ldots, |\mathbf{X}_i|$, where $|\mathbf{X}_i|$ represents the number of samples in the data batch, and each current memory cluster, as :

$$c^\star = \arg\max_{c=1,\cdots,k} F_{\text{KDM}}(\mathbf{x}_{i,j}, \widehat{\mathbf{x}}^c), \quad (11)$$

where $\mathcal{M}^{c^\star}$ is the selected memory cluster to store the sample $\mathbf{x}_{i,j}$. By using Eq. (11), we ensure that each memory cluster stores data samples defined by similar semantic information. When a certain memory cluster $\mathcal{M}^k$ is over-

Datasets	DCM-SE	DCM-JS	LTS	LGM	R-VAE	R-DDPM	CGKD-GAN	CNDPM	CGKD-WAE	MeRGANs
Split MNIST	28.57	30.63	71.67	66.31	55.67	63.26	54.34	65.19	47.98	49.96
Split Fashion	46.65	43.38	128.84	109.20	103.25	82.23	85.23	172.23	88.16	127.55
Split SVHN	61.52	62.61	87.25	72.60	65.18	87.22	101.26	150.18	100.15	81.35
Split CIFAR10	82.74	76.58	124.22	177.15	155.72	106.18	115.38	233.02	162.12	121.74
Average	54.87	**53.30**	102.99	106.31	94.95	84.72	89.05	155.15	99.54	95.15

Table 1. Image generation performance evaluated using the Fréchet Inception Distance (FID) score for class-incremental learning.

Datasets	DCM-SE	DCM-JS	LTS	LGM	R-VAE	R-DDPM	CGKD-GAN	CNDPM	CGKD-WAE	MeRGANs
CelebA-3DChair	**40.45**	82.18	186.25	241.14	210.18	183.72	132.12	340.25	154.45	166.99
CelebA-CACD	67.30	**48.38**	124.87	117.76	121.52	103.52	78.00	336.34	142.52	101.97
CelebA-ImageNet	**110.60**	115.25	255.94	265.23	225.12	217.98	178.72	186.76	170.07	236.81
Split MINIImageNet	**146.98**	154.83	179.78	216.06	205.12	181.15	176.18	302.58	241.11	169.26

Table 2. FID scores for assessing image generation performance for datasets with complex images.

loaded $|\mathcal{M}^k| > \pi$, we remove that sample which has the shortest distance to the prototype (central) sample $\widehat{\mathbf{x}}^k$:

$$e^\star = \arg\max_{e=1,\cdots,|\mathcal{M}^k|} \{F_{\text{KDM}}(\mathbf{x}'_{k,e}, \widehat{\mathbf{x}}^k)\}\,, \tag{12}$$

where $\mathbf{x}'_{k,e} \in \mathcal{M}^k$ is the e-th memorized sample of memory cluster $\mathcal{M}^k$, and $e^\star$ is the index of the removed sample from $\mathcal{M}^k$. This sample-removing process aims promoting knowledge diversity among the samples within each memory cluster while avoiding memory cluster overload.

3.5. Memory Pruning Process

When λ in the memory cluster system expansion criterion, defined in Eq. (10), is very small, then new memory clusters are frequently added, resulting in an overall memory overload. Moreover, when the proposed memory system is deployed on a resource-constrained device, it is necessary to manage the memory capacity while maintaining data diversity in the memory buffer system. In this paper, we address memory overload by introducing a novel Memory Pruning Process (MPP) to automatically remove statistically overlapping memory clusters. Let $\rho \in [1, 20]$ be a pre-defined maximum number of memory clusters according to the device's resource constraints. We assume that the memory $\mathcal{M}$ has added $k > \rho$ memory clusters during the training. To remove redundant memory clusters, we first define a relation matrix $\mathbf{B} \in \mathbf{R}^{k\times k}$ to describe the knowledge representation relationships among memory clusters, where $B[g, h]$ denotes the relation score between memory clusters $\mathcal{M}^g$ and $\mathcal{M}^h$, evaluated by $1/F_{\text{KDM}}(\widehat{\mathbf{x}}^g, \widehat{\mathbf{x}}^h)$ and $B[g, h] = B[h, g]$. Then, we identify a pair of memory clusters that share similar information through :

$$F_{\text{select}}(\mathbf{B}) = \underset{\{(g,h)\,|\,g,h=1,\cdots,k,g\neq h\}}{\arg\max} F_{\text{KDM}}(\mathcal{M}^g, \mathcal{M}^h)\,, \tag{13}$$

where $(g^\star, h^\star) = F_{\text{select}}(\mathbf{B})$ represents the index of two selected memory clusters $\{\mathcal{M}^{g^\star}, \mathcal{M}^{h^\star}\}$. In order to keep the knowledge diversity among the remaining clusters when removing one of them, we evaluate the discrepancy score between each selected cluster (either $\mathcal{M}^g$ or $\mathcal{M}^h$) and all other memory clusters using :

$$F_{\text{dis}}(g^\star, \mathbf{B}) = \sum\nolimits_{m=1,m\neq g^\star}^{k} \mathbf{B}[g^\star, m]\,. \tag{14}$$

$s_g = F_{\text{dis}}(g^\star, \mathbf{B})$ and $s_h = F_{\text{dis}}(h^\star, \mathbf{B})$ represent the discrepancy scores for $\mathcal{M}^{g^\star}$ and $\mathcal{M}^{h^\star}$. If $s_g > s_h$ then we remove $\mathcal{M}^h$ from $\mathcal{M}$, otherwise, we remove $\mathcal{M}^g$. Eq. (14) preserves the memory cluster that has a large distance with respect to the other clusters.

3.6. Algorithm Implementation

In the following we describe the algorithm to train a DDPM model under OTFCL using the proposed memory clustering system. The memory optimization process is illustrated in Fig. 2, and we provide the pseudo-code in **Algorithm 1**, which consists of four steps within a training time:

Step 1 (Sample selection). In the beginning, we build the first memory cluster $\mathcal{M}^1 = \mathbf{X}_1$ at t_1 and identify the prototype (central) sample $\widehat{\mathbf{x}}^1$ using Eq. (9). If the model is trained at a certain training time $t_i, i > 1$, we get an incoming data batch $\mathbf{X}_i$ at t_i and selectively store each sample $\mathbf{x}_{i,j} \in \mathcal{X}_i$ into $\mathcal{M}$ using the sample selection criterion defined in Eq. (11).

Step 2 (Check the model expansion). If the memory expansion criterion defined in Eq. (10) is fulfilled when checking the new sample $\mathbf{x}_{i,j}$ at t_i, we build a new memory cluster $\mathcal{M}^{k+1}$ and select $\mathbf{x}_{i,j}$ as the prototype sample for $\mathcal{M}^{k+1}$.

Step 3 (Memory pruning process). If the memory $\mathcal{M}$ is overloaded $|\mathcal{M}| > \rho$, then we remove overlapping memory clusters using Eq. (13) and Eq. (14).

Step 4 (Training process). We update the DDPM model $\boldsymbol{\epsilon}_\theta(\cdot, \cdot)$ on samples from the memory $\mathcal{M}$ using Eq. (3).

Methods	Resolution	CelebA-HQ	CACD	FFHQ
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-WVAE	$128 \times 128 \times 3$	139.96	158.32	179.59
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-WVAE	$256 \times 256 \times 3$	176.63	240.12	261.37

Table 3. FID scores for assessing the image generation performance for datasets containing high resolution images.

4. Experiment

4.1. Experiment Setting

Baselines and hyperparameters. Following the unsupervised learning setting from [71], we consider for comparison several baselines, including CGKD-GAN [71] and CGKD-WAE, where 'WAE' indicates that each component of CGKD is implemented by a Wasserstein auto-encoder [56], CNDPM [37]. Lifelong Teacher-Student (LTS) [68], Lifelong Generative Modelling (LGM) [47], Reservoir sampling [58] and MeRGANs [63]. Specifically, we employ Reservoir sampling [58] to train DDPM and VAE, respectively, resulting in Reservoir-DDPM (R-DDPM) and Reservoir-VAE (R-VAE). We set the batch size at each training time as $b = 64$, and the maximum memory size (the number of samples) as 2,000, for all models. We set the memory cluster size $\pi = 128$ and $t^\star = 100$ in Eq. (8). More details are shown in **Appendix-B1** from SM.

Datasets. We consider for training MNIST [36], Fashion [64], SVHN [41] and CIFAR10 [32], which are widely used in OTFCL [71]. Each of these is divided into five subsets according to the category information [4], resulting in Split MNIST, Split Fashion, Split SVHN and Split CIFAR10. Each image is resized to $32 \times 32 \times 3$ pixels. In addition, we also adopt several large-scale and complex-image datasets, including MINIImageNet [57], CACD [14], CelebA [38], 3DChair [6] and ImageNet [33].

4.2. Class-Incremental Generation

We train various models on Split MNIST, Split Fashion, Split SVHN and Split CIFAR10, and the generation performance results are reported in Table 1. These results show that the dynamic expansion models, such as CGKD-GAN and CGKD-VAE, provide better results than static models. R-DDPM does not achieve significant performance improvements despite employing a more powerful generative model (DDPM). In contrast, the proposed DCM-SE and DCM-JS outperform all baselines by a large margin, demonstrating that the proposed memory buffering management approach can explore the full potential ability of DDPM on image generation under OTFCL. Meanwhile, the proposed models use fewer memorized samples than other models according to the study from **Appendix-B2** from SM.

Methods	Split MNIST	Split CIFAR10	Split CIFAR100
finetune*	19.75 ± 0.05	18.55 ± 0.34	3.53 ± 0.04
GEM*	93.25 ± 0.36	24.13 ± 2.46	11.12 ± 2.48
iCARL*	83.95 ± 0.21	37.32 ± 2.66	10.80 ± 0.37
reservoir*	92.16 ± 0.75	42.48 ± 3.04	19.57 ± 1.79
MIR*	93.20 ± 0.36	42.80 ± 2.22	20.00 ± 0.57
GSS*	92.47 ± 0.92	38.45 ± 1.41	13.10 ± 0.94
CoPE-CE*	91.77 ± 0.87	39.73 ± 2.26	18.33 ± 1.52
CoPE*	93.94 ± 0.20	48.92 ± 1.32	21.62 ± 0.69
ER + GMED†	82.67 ± 1.90	34.84 ± 2.20	20.93 ± 1.60
ER_a + GMED†	82.21 ± 2.90	47.47 ± 3.20	19.60 ± 1.50
WGF-SVGD	-	47.90 ± 2.50	19.90 ± 2.30
CURL*	92.59 ± 0.66	-	-
CNDPM*	93.23 ± 0.09	45.21 ± 0.18	20.10 ± 0.12
Dynamic-OCM	94.02 ± 0.23	49.16 ± 1.52	21.79 ± 0.68
ORVAE	94.07 ± 0.13	50.43 ± 0.15	22.83 ± 0.25
DCM-SE	95.12 ± 0.32	50.02 ± 1.12	22.03 ± 0.79
DCM-JS	94.76 ± 0.29	49.89 ± 1.23	22.98 ± 0.74
Dynamic-JS	**98.52** ± 0.32	**55.94** ± 0.58	**26.27** ± 0.63

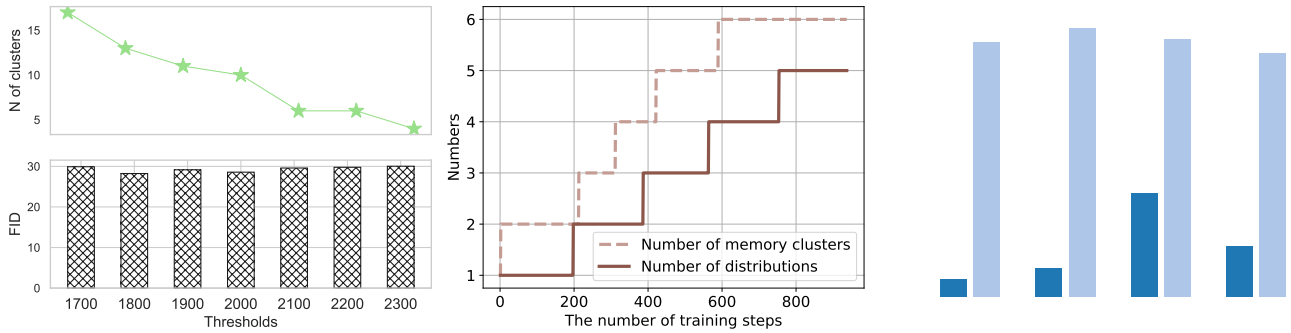
Table 4. Classification accuracy for five independent runs. * and † denote the results cited from [15] and [27], respectively.

4.3. Learning Multiple and Complex Domains

We investigate the performance of the proposed approach on datasets containing complex images. Specifically, we build a data stream consisting of CelebA and CACD, resulting in CelebA-CACD. Similarly, we create the data stream CelebA-3DChair and CelebA-ImageNet using pairs of different datasets, where all images are resized to $64 \times 64 \times 3$. The FID score evaluated on 5,000 testing data uniformly sampled from two testing datasets are reported in Table 2, while generated images after learning CelebA-3DChair are shown in Fig. 4. These results demonstrate that the proposed approach can achieve excellent performances on successions of complex datasets. The results for various models on datasets with high-resolution images such as CelebA-HQ [38], CACD and FFHQ are shown in Table 3 and these indicate that DCM-SE gives the best results.

4.4. Few-Shot Generation

We evaluate the performance of various models on a challenging dataset such as the MINIImageNet [57] that was used in few-shot learning experiments [51]. The MINI-ImageNet [57] consists of images from 100 categories, 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 build a data stream, namely Split MINIImageNet, by considering the



(a) Changing λ for DCM-SE. (b) The dynamic expansion of DCM-SE. (c) The time required for the memory optimization.

Figure 3. Ablation study results. (a) The number of memory clusters and model expansion of DCM-SE when changing λ in Eq. (10). (b) The number of memory clusters and data distribution shifts at each time. (3) The required time for the memory management optimization.

Figure 4. Image generation results for the proposed DCM-SE trained on CelebA-3DChair data stream.

meta-training and meta-validation datasets. Specifically, we divide the data stream into 16 parts, each consisting of samples from five successive categories. We report the image generation performance for various models in Table 2. We can observe that the proposed approach achieves the best results in the few-shot image generation tasks.

4.5. Classification Tasks

We consider the proposed memory management system approach in classification tasks by employing DCM-SE or DCM-JS to train a classifier in supervised learning. Following the Task Free Continual Learning (TFCL) setting from [15], the maximum memory size of the DCM is 2000, 1000, and 5000 for Split MNIST, Split CIFAR10, and Split CIFAR100, respectively. At each training time, we only access a batch of ten data samples. The network architecture for Split MNIST is a fully connected network with two layers of 400 units. For Split CIFAR10 and Split CIFAR100, we adopt a reduced version of ResNet18 [22]. The classification results from Table 4 show that the proposed memory management system approach outperforms other models even if its memory optimization does not interact with supervised signals such as class labels. In addition, we also extend DCM-JS with a network expansion mechanism, namely Dynamic-JS (See details in **Appendix-C9** from SM), which achieves the best performance.

4.6. Ablation Study

In this section, we perform a full ablation study to analyze the performance of the proposed approach under different configurations. More ablation study results are provided in **Appendix-C** from SM.

The effect when changing λ in Eq. (10). Changing λ affects the proposed memory expansion. We investigate this effect by training the DCM-SE on Split MNIST for different λ values and the results are shown in Fig. 3-a. A large λ leads to more memory clusters, while a small λ has an adverse effect. The results show that the proposed DCM-SE performs well even when using only six memory clusters.

The dynamic expansion process. We investigate whether the proposed memory expansion can appropriately create new memory clusters in order to adapt to the data distribution (task) shifts during the training. We train DCM-SE on Split MNIST and record the number of memory clusters and data distributions (tasks) each time, and the results are shown in Fig. 3-b. Note that the model does not access any task information during the training. We can observe that DCM-SE builds a new memory cluster to meet the change in the data distribution, showing that DCM-SE can provide appropriate signals for memory expansion.

Memory management processing time. In the following, given that the proposed memory management approach is nonparametric, we investigate the processing time required. We learn DCM-SE and DCM-JS on Split MNIST where we only optimize the memory without updating the model. From the results shown in Fig. 3-c we observe that the proposed DCM-SE requires less than one minute for its memory management optimization for most datasets.

5. Conclusion

This research study develops a memory management approach for training DDPM under the Online Task-Free Continual Learning (OTFCL) paradigm. The proposed Dynamic Cluster Memory (DCM) is a non-parametric efficient approach and can be applied in both supervised and unsupervised learning. To avoid memory overload, we propose a memory pruning process to automatically remove overlapping memory clusters. The results demonstrate the effectiveness of the proposed approach.

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, 3

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

[3] Rahaf Aljundi, Eugene Belilovsky, Tinne Tuytelaars, Laurent Charlin, Massimo Caccia, Min Lin, and Lucas Page-Caccia. Online continual learning with maximal interfered retrieval. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 11872–11883, 2019. 2

[4] 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. 1, 2, 7

[5] 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. 2

[6] 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

[7] 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, 2

[8] 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 IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 9275–9284, 2022. 2, 5

[9] Konstantinos Bousmalis, Nathan Silberman, David Dohan, Dumitru Erhan, and Dilip Krishnan. Unsupervised pixel-level domain adaptation with generative adversarial networks. In *Proc. of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3722–3731, 2017. 1

[10] Pietro Buzzega, Matteo Boschini, Angelo Porrello, Davide Abati, and Simone Calderara. Dark experience for general continual learning: a strong, simple baseline. *Advances in Neural Information Processing Systems (NeurIPS)*, 33: 15920–15930, 2020. 2

[11] Hyuntak Cha, Jaeho Lee, and Jinwoo Shin. Co^{2}L: Contrastive continual learning. In *Proc. of IEEE/CVF Int. Conf. on Computer Vision (ICCV)*, pages 9516–9525, 2021. 2

[12] Arslan Chaudhry, Marc'Aurelio Ranzato, Marcus Rohrbach, and Mohamed Elhoseiny. Efficient lifelong learning with A-GEM. In *Int. Conf. on Learning Representations (ICLR), arXiv preprint arXiv:1812.00420*, 2019. 2

[13] 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. 2

[14] 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

[15] Matthias De Lange and Tinne Tuytelaars. Continual prototype evolution: Learning online from non-stationary data streams. In *Proc. of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 8250–8259, 2021. 2, 7, 8

[16] 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

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

[18] Evgenii Egorov, Anna Kuzina, and Evgeny Burnaev. BooVAE: Boosting approach for continual learning of vae. *Advances in Neural Information Processing Systems (NeurIPS)*, 34:17889–17901, 2021. 2

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

[20] 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, 5

[21] Yiduo Guo, Bing Liu, and Dongyan Zhao. Online continual learning through mutual information maximization. In *Proc. International Conference on Machine Learning (ICML), vol. PMLR 162*, pages 8109–8126, 2022. 2, 5

[22] 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. 1, 8

[23] Christian Henning, Maria Cervera, Francesco D'Angelo, Johannes Von Oswald, Regina Traber, Benjamin Ehret, Seijin Kobayashi, Benjamin F Grewe, and João Sacramento. Posterior meta-replay for continual learning. *Advances in Neural Information Processing Systems (NeurIPS)*, 34:14135–14149, 2021. 2

[24] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in Neural Information Processing Systems (NeurIPS)*, 33:6840–6851, 2020. 2, 3, 4

[25] Julio Hurtado, Alain Raymond, and Alvaro Soto. Optimizing reusable knowledge for continual learning via metalearning. *Advances in Neural Information Processing Systems (NeurIPS)*, 34:14150–14162, 2021. 2

[26] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. Image-to-image translation with conditional adversarial networks. In *Proc. of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1125–1134, 2017. 1

[27] Xisen Jin, Arka Sadhu, Junyi Du, and Xiang Ren. Gradient-based editing of memory examples for online task-free continual learning. In *Advances in Neural Information Processing Systems (NeurIPS), arXiv preprint arXiv:2006.15294*, 2021. 1, 2, 7

[28] Ronald Kemker, Marc McClure, Angelina Abitino, Tyler Hayes, and Christopher Kanan. Measuring catastrophic forgetting in neural networks. In *Proc. of AAAI Conference on Artificial Intelligence*, pages 3390–3398, 2018. 2

[29] Sanghwan Kim, Lorenzo Noci, Antonio Orvieto, and Thomas Hofmann. Achieving a better stability-plasticity trade-off via auxiliary networks in continual learning. In *Proc. of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 11930–11939, 2023. 2

[30] D. P. Kingma and M. Welling. Auto-encoding variational Bayes. *arXiv preprint arXiv:1312.6114*, 2013. 2, 4

[31] J. Kirkpatrick, R. Pascanu, N. Rabinowitz, J. Veness, G. Desjardins, A. A. Rusu, K. Milan, J. Quan, T. Ramalho, A. Grabska-Barwinska, D. Hassabis, C. Clopath, D. Kumaran, and R. Hadsell. Overcoming catastrophic forgetting in neural networks. *Proc. of the National Academy of Sciences (PNAS)*, 114(13):3521–3526, 2017. 1, 2

[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. 1, 7

[34] Richard Kurle, Botond Cseke, Alexej Klushyn, Patrick van der Smagt, and Stephan Günnemann. Continual learning with bayesian neural networks for non-stationary data. In *International Conference on Learning Representations (ICLR), https://openreview.net/forum?id=SJlsFpVtDB*, 2020. 4, 5

[35] Anders Larsen, Søren Kaae Sønderby, Hugo Larochelle, and Ole Winther. Autoencoding beyond pixels using a learned similarity metric. In *Proc. International Conf. on Machine Learning (ICML), vol. PMLR 48*, pages 1558–1566, 2015. 3

[36] 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

[37] Soochan Lee, Junsoo Ha, Dongsu Zhang, and Gunhee Kim. A neural Dirichlet process mixture model for task-free continual learning. In *Int. Conf. on Learning Representations (ICLR), arXiv preprint arXiv:2001.00689*, 2020. 3, 7

[38] 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

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

[40] James Martens and Roger B. Grosse. Optimizing neural networks with kronecker-factored approximate curvature. In *Proc. of the 32nd Int. Conf. on Machine Learning (ICML), JMLR: WCP vol. 37.*, pages 2408–2417, 2015. 2

[41] 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

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

[43] Alexander Quinn Nichol and Prafulla Dhariwal. Improved denoising diffusion probabilistic models. In *Proc. of the International Conference on Machine Learning (ICML), vol. PMLR 139*, pages 8162–8171, 2021. 3, 4

[44] Xing Nie, Shixiong Xu, Xiyan Liu, Gaofeng Meng, Chunlei Huo, and Shiming Xiang. Bilateral memory consolidation for continual learning. In *Proc. of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 16026–16035, 2023. 2

[45] 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

[46] Krishnan Raghavan and Prasanna Balaprakash. Formalizing the generalization-forgetting trade-off in continual learning. *Advances in Neural Information Processing Systems (NeurIPS)*, 34:17284–17297, 2021. 4, 5

[47] J. Ramapuram, M. Gregorova, and A. Kalousis. Lifelong generative modeling. In *Proc. Int. Conf. on Learning Representations (ICLR), arXiv preprint arXiv:1705.09847*, 2017. 2, 3, 7

[48] Dushyant Rao, Francesco Visin, Andrei A. Rusu, Yee Whye Teh, Razvan Pascanu, and Raia Hadsell. Continual unsupervised representation learning. In *Proc. Neural Inf. Proc. Systems (NIPS)*, pages 7645–7655, 2019. 2

[49] 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

[50] 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. 1

[51] 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. 7

[52] Ali Sharif Razavian, Hossein Azizpour, Josephine Sullivan, and Stefan Carlsson. CNN features off-the-shelf: an astounding baseline for recognition. In *Proc. of the IEEE CVPR-workshops*, pages 806–813, 2014. 1

[53] 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

[54] 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

[55] 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, 5

[56] Ilya Tolstikhin, Olivier Bousquet, Sylvain Gelly, and Bernhard Schoelkopf. Wasserstein auto-encoders. In *International Conference on Learning Representations (ICLR), arXiv preprint arXiv:1711.01558*, 2018. 7

[57] 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

[58] Jeffrey S Vitter. Random sampling with a reservoir. *ACM Transactions on Mathematical Software (TOMS)*, 11(1):37–57, 1985. 7

[59] Liyuan Wang, Mingtian Zhang, Zhongfan Jia, Qian Li, Chenglong Bao, Kaisheng Ma, Jun Zhu, and Yi Zhong. Afec: Active forgetting of negative transfer in continual learning. *Advances in Neural Information Processing Systems (NeurIPS)*, 34:22379–22391, 2021. 2

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

[61] Zhenyi Wang, Li Shen, Le Fang, Qiuling Suo, Tiehang Duan, and Mingchen Gao. Improving task-free continual learning by distributionally robust memory evolution. In *Proc. International Conference on Machine Learning (ICML), vol. PMLR 162*, pages 22985–22998, 2022. 2, 4

[62] Zifeng Wang, Zizhao Zhang, Chen-Yu Lee, Han Zhang, Ruoxi Sun, Xiaoqi Ren, Guolong Su, Vincent Perot, Jennifer Dy, and Tomas Pfister. Learning to prompt for continual learning. In *Proc. of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 139–149, 2022. 2

[63] 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. 3, 7

[64] 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

[65] Qingsen Yan, Dong Gong, Yuhang Liu, Anton van den Hengel, and Javen Qinfeng Shi. Learning bayesian sparse networks with full experience replay for continual learning. In *Proc. of IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 109–118, 2022. 2

[66] Ling Yang, Zhilong Zhang, Yang Song, Shenda Hong, Runsheng Xu, Yue Zhao, Yingxia Shao, Wentao Zhang, Bin Cui, and Ming-Hsuan Yang. Diffusion models: A comprehensive survey of methods and applications. *ACM Computing Surveys*, 56(4):1–39, 2022. 3

[67] 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. 2

[68] Fei Ye and Adrian G. Bors. Lifelong teacher-student network learning. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 44:6280–6296, 2022. 3, 7

[69] Fei Ye and Adrian G. Bors. Continual variational autoencoder learning via online cooperative memorization. In *Proc. European Conference on Computer Vision (ECCV), vol. LNCS 13683*, pages 531–549, 2022. 3

[70] Fei Ye and Adrian G Bors. Task-free continual learning via online discrepancy distance learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 23675–23688, 2022. 1

[71] 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. 2, 3, 7

[72] Fei Ye and Adrian G Bors. Learning dynamic latent spaces for lifelong generative modelling. In *Proc. of the AAAI Conference on Artificial Intelligence*, pages 10891–10899, 2023. 3

[73] Jingwen Ye, Songhua Liu, and Xinchao Wang. Partial network cloning. In *Proc. of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 20137–20146, 2023. 2

[74] Haiyan Yin, Peng Yang, and Ping Li. Mitigating forgetting in online continual learning with neuron calibration. *Advances in Neural Information Processing Systems (NeurIPS)*, 34:10260–10272, 2021. 2

[75] Friedemann Zenke, Ben Poole, and Surya Ganguli. Continual learning through synaptic intelligence. In *Proc. of Int. Conf. on Machine Learning, vol. PLMR 70*, pages 3987–3995, 2017. 1

[76] Mengyao Zhai, Lei Chen, Fred Tung, Jiawei He, Megha Nawhal, and Greg Mori. Lifelong GAN: Continual learning for conditional image generation. In *Proc. of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 2759–2768, 2019. 1, 2, 3