Proceedings Paper:
Ye, Fei and Bors, Adrian Gheorghe (2024) Task-Free Continual Generation and Representation Learning via Dynamic Expansionable Memory Cluster. In: AAAI Conference on Artificial Intelligence. Proceedings of the ... AAAI Conference on Artificial Intelligence. AAAI Press, pp. 16451-16459.

https://doi.org/10.1609/aaai.v38i15.29582

Task-Free Continual Generation and Representation Learning via Dynamic Expansionable Memory Cluster

Fei Ye[1,2], **Adrian G. Bors**[1,2]

[1]Department of Computer Science, University of York, York YO10 5GH, UK
[2]Machine Learning Department, Mohamed bin Zayed University of Artificial Intelligence, Abu Dhabi, UAE
fy689@york.ac.uk, adrian.bors@york.ac.uk

Abstract

Human brains can continually acquire and learn new skills and knowledge over time from a dynamically changing environment without forgetting previously learnt information. Such a capacity can selectively transfer some important and recently seen information to the persistent knowledge regions of the brain. Inspired by this intuition, we propose a new memory-based approach for image reconstruction and generation in continual learning, consisting of a temporary and evolving memory, with two different storage strategies, corresponding to the temporary and permanent memorisation. The temporary memory aims to preserve up-to-date information while the evolving memory can dynamically increase its capacity in order to preserve permanent knowledge information. This is achieved by the proposed memory expansion mechanism that selectively transfers those data samples deemed as important from the temporary memory to new clusters defined within the evolved memory according to an information novelty criterion. Such a mechanism promotes the knowledge diversity among clusters in the evolved memory, resulting in capturing more diverse information by using a compact memory capacity. Furthermore, we propose a two-step optimization strategy for training a Variational Autoencoder (VAE) to implement generation and representation learning tasks, which updates the generator and inference models separately using two optimisation paths. This approach leads to a better trade-off between generation and reconstruction performance. We show empirically and theoretically that the proposed approach can learn meaningful latent representations while generating diverse images from different domains. The source code and supplementary material (SM) are available at https://github.com/dtuzi123/DEMC.

Introduction

Generative Adversarial Networks (GANs) (Goodfellow et al. 2014a) and Variational Autoencoders (VAEs) (Kingma and Welling 2013) represent two of the most popular deep generative models used in many applications, including image synthesis (Karras et al. 2020), video generation (Tulyakov et al. 2018) and image-to-image translation (Isola et al. 2017). However, despite their impressive performance, most generative models can only be applied to static data distributions. When trained in a continual learning fashion, these models suffer a significant performance degradation when tested on past tasks, and such a performance loss is called catastrophic forgetting (Parisi et al. 2019; Ye and Bors 2020). A simple and natural way to mitigate forgetting is to use deep generative models such as GANs and VAEs to remember and generate past knowledge that is then incorporated together with new data, and used for retraining the model (Seff et al. 2017). However, the performance of such an approach is highly dependent on the quality of the generative replay samples, and this degrades significantly after learning several different data domains (Ye and Bors 2023a). Another approach proposes using the dynamic expansion mechanism to gradually add new parameters to learn new tasks, while freezing all previously learned parameters to preserve past knowledge (Varshney et al. 2021; Zhai et al. 2020). However, these approaches require access to the explicit task boundaries, which is impossible in real scenarios (Aljundi, Kelchtermans, and Tuytelaars 2019).

In this paper, we study a new and more realistic learning scenario called the Task-Free Continual Learning (TFCL) (Aljundi, Kelchtermans, and Tuytelaars 2019), in which a model is trained on a data stream without explicit task boundaries. Storing diverse samples in a memory buffer was shown to relieve forgetting in TFCL (Aljundi, Kelchtermans, and Tuytelaars 2019). However, most existing memory-based methods can only be applied to supervised learning (Aljundi, Kelchtermans, and Tuytelaars 2019; Jin et al. 2021), while unsupervised representation learning and generation in TFCL has not been studied before. Here we aim to learn a generative model capable of generating diverse images while also capturing meaningful latent representation across different data domains without forgetting. Biological studies show that the human brain can store short- and long-term information in different regions (Berns et al. 2013), where the short-term information can be persistent when necessary. Inspired by this intuition, we propose a new memory system consisting of temporary and evolved memory buffers, each with different storage strategies. The first type of memory aims to temporarily preserve the up-to-date information about a data stream, while the second type permanently preserves necessary samples from the temporary memory during the training. The evolved memory can dynamically build a sequence of memory clusters, each having a fixed size, expected to capture different informa-

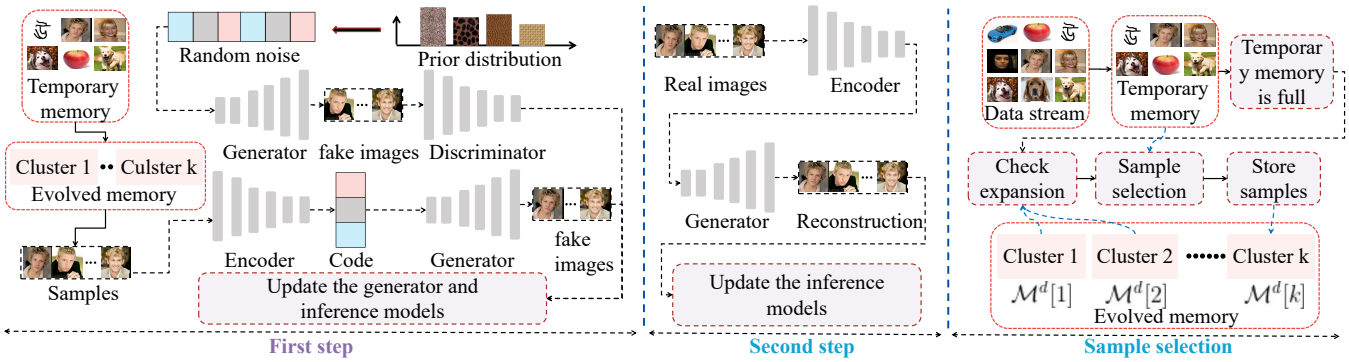

Figure 1: The updating procedure of the proposed Continual Variational Autoencoder (CAA), at each training time, can be summarized in three steps. In the first step, we update all modules using $\mathcal{L}_{g'}$ and $\mathcal{L}_d$ from Eq. (5) and (3). In the second step, we only update the inference model by minimizing $\mathcal{L}_{\text{inf}}$, Eq. (6), with the frozen generator and discriminator. In the sample selection step, if the memory expansion criterion (Eq. (8)) is satisfied, we transfer the temporary memory buffer to a new memory cluster $\mathcal{M}^d[k]$, clearing up $\mathcal{M}_i^t$. More details are provided in Appendix-A from Supplementary Material (SM).

tion. In order to gradually store important information in the evolved memory, we propose a new memory expansion mechanism that evaluates the probabilistic distance between each existing memory cluster and the current temporary memory's probabilistic representation. A large distance indicates that the temporary memory is new to the already preserved knowledge and then many critical samples from the temporary memory are stored into a newly created cluster of the evolved memory following a sample selection process. Furthermore, we perform this expansion mechanism on the aggregated posterior achieved by the inference model, which is computationally efficient.

To relieve poor generation results caused when using the VAE loss, some research studies proposed to explore the generation ability of GANs, within a hybrid VAE-GAN framework (Huang et al. 2018; Larsen et al. 2015; Ulyanov, Vedaldi, and Lempitsky 2018; Xian et al. 2019). However, these methods adopt a unified objective function, which can not guarantee the learning of an optimal inverse mapping for the generator. In this paper, we develop a new model called the Continual Variational Autoencoder (CAA) that formulates the inference model and the generator (decoder) learning as two separate optimisation paths, providing a better trade-off between generation and reconstruction performance. In particular, we introduce a two-step optimization strategy. In the first step, we want the generator to approximate the data distribution as closely as possible while the model can also learn latent representations. In the second step, we freeze the generator and optimise the inference model only by maximising the sample log-likelihood in order to learn an optimal inverse mapping of the generator. Such a training procedure, called the Two-Step Optimization Strategy (TSOS), is a good compromise between optimizing the generator and the inference model. Moreover, unlike existing hybrid VAE-GAN approaches (Huang et al. 2018; Larsen et al. 2015; Ulyanov, Vedaldi, and Lempitsky 2018; Xian et al. 2019), which can only learn a static data distribution, the proposed CAA is trained on a dynamically changing data stream without explicit task boundaries.

Extensive experiments show that the proposed Continual Variational Autoencoder (CAA) can produce diverse images across all learnt data domains over time, which is consistent with our theoretical results. The contributions are summarised as follows: (1) We propose a bio-inspired memory approach to deal with the generation and unsupervised representation learning in TFCL; (2) We propose a new training strategy which ensures a good trade-off between generation and reconstruction performance; (3) We provide the theoretical analysis and understanding of the proposed memory approach. (4) We establish a new evaluation criterion for the generation and unsupervised representation learning.

Related Work

Lately, Continual Learning (CL) has become a hot topic in machine learning since it enables machines to deal with tasks without forgetting. Many efforts are devoted to mitigating forgetting in CL by developing a memory buffering system that stores past examples to relieve forgetting (Bang et al. 2022; Cha, Lee, and Shin 2021; Yan et al. 2022; Lopez-Paz and Ranzato 2017; Derakhshani et al. 2021; Shi et al. 2021; Wang et al. 2021; Egorov, Kuzina, and Burnaev 2021). Regularization and knowledge distillation methods have recently been considered to further improve performance (Aljundi et al. 2019b; Chaudhry et al. 2019b,a; Bang et al. 2021, 2022; Cha, Lee, and Shin 2021; Kirkpatrick et al. 2017; Kurle et al. 2020; Li and Hoiem 2017). Although the memory-based approaches perform well, they suffer from the negative backward transfer when learning new samples (Ye and Bors 2022a). Such a drawback is solved by employing a dynamic network architecture which preserves prior knowledge into frozen parameters of specific units while building new hidden layers and units to learn novel tasks (Wen, Tran, and Ba 2020; Ye and Bors 2023b,a).

Using a memory buffer to store past samples was shown to perform well in the Task Free Continual Learning (TFCL) (Aljundi, Kelchtermans, and Tuytelaars 2019; Aljundi et al. 2019a,a; De Lange and Tuytelaars 2021; Jin et al. 2021; Ye and Bors 2023e,d, 2022b,a). The Reservoir sampling (Vitter

1985) randomly stores data samples over time and can be used in unsupervised learning. However, these methods perform worse when applied to learning a long-term data stream due to their fixed model capacity. Meanwhile, dynamic architecture models have achieved promising results (Lee et al. 2020; Rao et al. 2019; Ye and Bors 2022a). The first dynamic expansion model for TFCL, proposed in (Rao et al. 2019), dynamically adds new inference models to capture data changes while using Generative Replay Mechanisms (GRMs) to alleviate forgetting. Then the Continual Neural Dirichlet Process Mixture (CN-DPM) (Lee et al. 2020), proposes to use Dirichlet processes to augment the mixture model while freezing the trained components to preserve past knowledge. More recently, the dynamic expansion model was implemented by using a mixture of VAEs as in the Online Cooperative Memorization (OCM) model (Ye and Bors 2022a), which employs a kernel-based sample selection approach to manage the memory buffer for TFCL.

Most attempts in TFCL are devoted to the classification task, while data generation was less explored in this context. The pioneering work in continual generation tasks employs a VAE-based framework (Achille et al. 2018), which learns shared and task-specific representations over time. More recently, continual generation was implemented using a teacher-student (Ramapuram, Gregorova, and Kalousis 2017; Ye and Bors 2023a) or a hybrid VAE-GAN framework (Ye and Bors 2020). Despite achieving promising results, these methods rely highly on the task information, which is intractable in TFCL. The recently developed dynamic expansion models can solve this drawback by dynamically increasing the model's capacity (Ye and Bors 2022a; Rao et al. 2019), but lead to the creation of many sub-models requiring significant inference times during testing.

Methodology

Problem Definition and Network Architecture

Under the TFCL, a model is trained on a data stream without accessing the knowledge about task boundaries. Let $\mathcal{D}_t^T = \{\mathbf{x}_i^T\}_{i=1}^{n_t^T}$ and $\mathcal{D}_t^S = \{\mathbf{x}_i^S\}_{i=1}^{n_t^S}$ be the t-th unlabelled testing and training datasets, respectively, where n_t^T and n_t^S represent the number of samples for $\mathcal{D}_t^T$ and $\mathcal{D}_t^S$, respectively. In a class-incremental learning paradigm, each training set $\mathcal{D}_t^S$ is divided into C_t^S parts $\{\mathcal{D}_{t,j}^S \,|\, j = 1, \cdots, C_t^S\}$ according to the category information. A data stream S in such a learning paradigm is defined as:

$$S = \bigcup\nolimits_{j=1}^{C_t^S} \mathcal{D}_{t,j}^S \,. \tag{1}$$

During the training, a model is able to access a small batch of b samples $\mathbf{X}_i = \{\mathbf{x}_j^S\}_{j=1}^b$ from S at a training time ($\mathcal{T}_i$), while all previous data batches $\{\mathbf{X}_1, \cdots, \mathbf{X}_{i-1}\}$ are unavailable. We assume that there are a total of n data batches for S. The model is evaluated on $\mathcal{D}_t^T$ after finishing training.

Network architecture. The structure of the proposed Continual Variational Autoencoder (CAA) is shown in Fig. 1; it contains three modules: the discriminator, generator and inference model (encoder). We implement the discriminator using a convolutional neural network (CNN) $f_\psi \colon \mathcal{X} \to \mathrm{R}$ which receives an image $\mathbf{x} \in \mathcal{X}$ and returns a scalar, where $\mathcal{X}$ is the data space. The generator is implemented by a deconvolution neural network $f_\theta \colon \mathcal{Z} \to \mathcal{X}$ which receives a low-dimensional latent variable vector $\mathbf{z} \in \mathcal{Z}$ and produces a generated image $\mathbf{x}' \in \mathcal{X}$, where $\mathcal{Z}$ is the latent variable space. We can form a decoding distribution using f_θ, expressed as $p_\theta(\mathbf{x} \,|\, \mathbf{z})$ which is usually a Gaussian distribution. For learning meaningful latent variables and enabling the decoding process, the proposed CAA has an inference model, implemented by a CNN f_ϵ which takes an image $\mathbf{x}$ as input and returns the Gaussian hyperparameters $\{\boldsymbol{\mu}_\epsilon, \boldsymbol{\sigma}_\epsilon\}$, which forms an encoding distribution $q_\epsilon(\mathbf{z} \,|\, \mathbf{x}) = \mathcal{N}(\boldsymbol{\mu}_\epsilon, \boldsymbol{\sigma}_\epsilon^2 \mathbf{I})$. The latent variable $\mathbf{z} = \boldsymbol{\mu}_\epsilon + \boldsymbol{\sigma}_\epsilon \odot \boldsymbol{\tau}$ is drawn from $\mathcal{N}(\boldsymbol{\mu}_\epsilon, \boldsymbol{\sigma}_\epsilon^2 \mathbf{I})$ using the reparameterization trick (Kingma and Welling 2013) to ensure end-to-end training, $\boldsymbol{\tau} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$, and $\odot$ is the element-wise product.

The Two-Step Optimization Strategy (TSOS)

In this section, we address the drawbacks of VAEs in generating blurred images or of GANs which cannot learn meaningful latent representations, by proposing a two-step optimization strategy, described as follows.

First step: We aim to approximate the memory distribution by updating the generator. To implement this goal, we firstly employ a WGAN-GP loss (Gulrajani et al. 2017) for updating the generator and discriminator at $\mathcal{T}_i$:

$$\mathcal{L}_g = -\mathbb{E}_{\mathbf{x}' \sim \mathbb{P}_{\theta_i}} [f_{\psi_i}(\mathbf{x}')] \,, \tag{2}$$

$$\begin{aligned} \mathcal{L}_d = {} & \mathbb{E}_{\mathbf{x}' \sim \mathbb{P}_{\theta_i}} [f_{\psi_i}(\mathbf{x}')] - \mathbb{E}_{\mathbf{x}_j \sim \mathbb{P}_{\widetilde{\mathcal{M}}_i}} [f_{\psi_i}(\mathbf{x}_j)] \\ & + \gamma \mathbb{E}_{\hat{\mathbf{x}} \sim \mathbb{P}_{\hat{\mathbf{x}}}} \left[\left(\|\nabla_{\hat{\mathbf{x}}} f_{\psi_i}(\hat{\mathbf{x}})\|_2 - 1 \right)^2 \right] , \end{aligned} \tag{3}$$

where $\hat{\mathbf{x}}$ is an interpolated image produced by $\hat{\mathbf{x}} = u\mathbf{x}_i + (1-u)\mathbf{x}'$ where u is drawn from a uniform distribution $U(0,1)$ and $\mathbb{P}_{\hat{\mathbf{x}}}$ is the distribution of the interpolated images. $\widetilde{\mathcal{M}}_i$ is a joint memory buffer consisting of $\mathcal{M}^d$ and $\mathcal{M}_i^t$ and $\mathbb{P}_{\widetilde{\mathcal{M}}_i}$ is the memory distribution. $\mathbb{P}_{\theta_i}$ is the generator distribution updated at $\mathcal{T}_i$. $\mathcal{L}_g$ and $\mathcal{L}_d$ are two loss functions which are minimized using the stochastic gradient method for optimizing the generator and discriminator, respectively. In order to learn an inverse mapping of the generator, we minimize the negative maximum likelihood function at $\mathcal{T}_i$:

$$\begin{aligned} \mathcal{L}_v = {} & \mathbb{E}_{\mathbf{x} \sim \mathbb{P}_{\widetilde{\mathcal{M}}_i}} \big[- \mathbb{E}_{q_{\epsilon_i}(\mathbf{z}|\mathbf{x})} [\log p_{\theta_i}(\mathbf{x} \,|\, \mathbf{z})] \\ & + D_{KL} (q_{\epsilon_i}(\mathbf{z} \,|\, \mathbf{x}) \,||\, p(\mathbf{z})) \big] , \end{aligned} \tag{4}$$

where ϵ_i is the parameter set of the inference model updated at $\mathcal{T}_i$. The first term in Eq. (4) is implemented by the reconstruction error of the decoder $p_{\theta_i}(\mathbf{x} \,|\, \mathbf{z})$ and the second term is the Kullback–Leibler divergence between the variational distribution $q_{\epsilon_i}(\mathbf{z} \,|\, \mathbf{x})$ and the prior distribution $p(\mathbf{z})$. We update the generator and inference model by using a unified loss function, as in VAEGAN (Larsen et al. 2015):

$$\mathcal{L}_{g'} = \mathcal{L}_g + \lambda_2 \mathcal{L}_v \,, \tag{5}$$

where $\lambda_2 \in [0.1, 1]$ is a hyperparameter which balances the generator and inference model learning.

Second step: Since the loss function from Eq. (5), trades off the optimization between the generator and inference model,

it cannot ensure learning a good inverse mapping of the generator. Then we optimize the inference model $q_{\epsilon_i}(\mathbf{z}\,|\,\mathbf{x})$ in order to learn meaningful latent representations improving the image reconstruction performance. We freeze the generator and only update the parameters of the inference model by minimizing the negative sample log-likelihood using the memorized and generative replay samples at $\mathcal{T}_i$:

$$\begin{aligned}\mathcal{L}_{\text{inf}} = \mathbb{E}_{\mathbf{x}\sim\mathbb{P}_{\widetilde{\mathcal{M}}_i\otimes\hat{\mathcal{M}}_i}}\big[-\mathbb{E}_{q_{\epsilon_i}(\mathbf{z}|\mathbf{x})}\left[\log p_{\theta_i}(\mathbf{x}\,|\,\mathbf{z})\right] \\ + D_{KL}\left(q_{\epsilon_i}(\mathbf{z}\,|\,\mathbf{x})\,||\,p(\mathbf{z})\right)\big]\,,\end{aligned} \tag{6}$$

where $\mathbb{P}_{\widetilde{\mathcal{M}}_i\otimes\hat{\mathcal{M}}_i}$ is the distribution of $\widetilde{\mathcal{M}}_i$ and $\hat{\mathcal{M}}_i$, with the latter representing a set of generative replay samples drawn from the generator at $\mathcal{T}_i$, and $\otimes$ denotes the joint dataset.

The Dynamic Expansionable Memory Cluster

Current memory-based methods usually consider storing the entire past information within a single restricted memory buffer, which is not scalable to a dynamically changing data stream. In this paper, we propose a bio-inspired memory approach which dynamically manages two memory buffers: the temporary buffer $\mathcal{M}^t$ with a fixed length $|\mathcal{M}^t|^{max}$ aiming to store more recent information and an evolving memory buffer $\mathcal{M}^d$ that can dynamically add novel samples over time to preserve the long-term required information. $\mathcal{M}^d$ continually adds a sequence of memory clusters $\{\mathcal{M}^d[1],\cdots,\mathcal{M}^d[k]\}$ during the training, where each cluster has a small fixed size $|\mathcal{M}^d[i]|^{max}$, and is expected to preserve a diverse information. $\mathcal{M}^d$ can add a new memory cluster $\mathcal{M}^d[k+1]$ when the temporary buffer $\mathcal{M}_i^t$ is largely different from the existing clusters of $\mathcal{M}^d$. We can interpret this memory expansion process as an optimization problem:

$$\bar{\mathcal{M}}^t = \arg\max_{\{\mathcal{M}_i^t\,|\,i=c+1,\cdots,n\}} \sum_{j=1}^{k} \mathcal{L}_d(\mathcal{M}^d[j],\mathcal{M}_i^t)\,, \tag{7}$$

where c is the index of the training time and $\mathcal{M}^d[k]$ was added to $\mathcal{M}^d$ at $\mathcal{T}_c$. k is the number of existing memory clusters in $\mathcal{M}^d$. $\mathcal{L}_d(\cdot,\cdot)$ is an arbitrary measuring function that evaluates the distance between each memory cluster and the temporary memory, while the maximum distance is achieved by the optimal memory $\bar{\mathcal{M}}^t$. However, we can not evaluate Eq. (7) in the TFCL because it requires passing all training times. Instead, we implement the goal of Eq. (7) by proposing a novel memory expansion criterion, which evaluates the difference between the temporary buffer $\mathcal{M}_i^t$ and each memory cluster $\{\mathcal{M}^d[j]\,|\,j=1,\cdots,k\}$, as the expansion signal at each training time $\mathcal{T}_i$:

$$\min\Big\{\mathcal{L}_d(\mathcal{M}^d[1],\mathcal{M}_i^t),\cdots,\mathcal{L}_d(\mathcal{M}^d[k],\mathcal{M}_i^t)\Big\} \geq \lambda, \tag{8}$$

where $\lambda\in[0,40]$ is an expansion detection threshold controlling the growing process of $\mathcal{M}^d$. Instead of evaluating the distance between memory buffers in the image space, which would require a higher computational complexity, we propose to estimate the distance on the aggregate posterior of each memory set using the inference model f_ϵ:

$$\begin{aligned}q_{\mathcal{M}^d[j]}(\mathbf{z}\,|\,\mathbf{x}) &= \mathbb{E}_{\mathbf{x}\sim\mathcal{M}^d[j]}[q_\epsilon(\mathbf{z}\,|\,\mathbf{x})]\,,\\ q_{\mathcal{M}_i^t}(\mathbf{z}\,|\,\mathbf{x}) &= \mathbb{E}_{\mathbf{x}\sim\mathcal{M}_i^t}[q_\epsilon(\mathbf{z}\,|\,\mathbf{x})]\,.\end{aligned} \tag{9}$$

The aggregate posterior (Gaussian distribution) has been successfully used in zero-shot learning (Chen et al. 2020), as the optimal prior in VAE learning (Tomczak and Welling 2018), and in the VAEGAN model (Makhzani et al. 2016). However, none of the current studies considers measuring the distance on the aggregate posterior in continual learning. In this paper, we propose to implement $\mathcal{L}_d(\mathcal{M}^d[j],\mathcal{M}_i^t)$ by using the Jensen–Shannon divergence which is a symmetrical statistical measure with an analytical solution:

$$\mathcal{L}_d(\mathcal{M}^d[j],\mathcal{M}_i^t) = D_{JS}\big(q_{\mathcal{M}^d[j]}(\mathbf{z}\,|\,\mathbf{x})\,||\,q_{\mathcal{M}_i^t}(\mathbf{z}\,|\,\mathbf{x})\big). \tag{10}$$

When satisfying the criterion from Eq. (8), we transfer some samples from the temporary buffer $\mathcal{M}_i^t$ to a new memory cluster $\mathcal{M}^d[k+1]$ by performing sample selection:

$$\mathcal{M}^d[k+1] = \bigcup_{j=1}^{|\mathcal{M}^d[k+1]|^{max}} \widehat{\mathcal{M}}_i^t[j]\,, \tag{11}$$

where $\widehat{\mathcal{M}}_i^t$ is a memory buffer sorted as $\mathcal{L}_{\text{inf}}(\widehat{\mathcal{M}}_i^t[a]) > \mathcal{L}_{\text{inf}}(\widehat{\mathcal{M}}_i^t[b])$ for $a<b$, where $\mathcal{L}_{\text{inf}}$ is defined in (6). $\widehat{\mathcal{M}}_i^t[b]$ is the b-th sample from $\widehat{\mathcal{M}}_i^t$ and $|\mathcal{M}^d[k+1]|^{max}$ is the maximum memory cluster size, $|\mathcal{M}^d[k+1]|^{max} < |\mathcal{M}_i^t|$. $\mathcal{L}_v(\cdot)$ is the loss function of the inference model from Eq. (4). Eq. (11) selects the data that have large loss values and tend to be forgotten. The temporary memory buffer $\mathcal{M}_i^t$ is then emptied to avoid storing again the same data.

Algorithm Implementation

We provide the algorithm for learning the proposed CAA, which can be summarized into three steps in a training time:

- **Step 1.** At a training time $\mathcal{T}_i$, the memory buffer is updated using the Reservoir approach (Vitter 1985). The generator and inference model are updated on the memory buffer using $\mathcal{L}_{g'}$ from Eq. (5) and then the discriminator using $\mathcal{L}_d$ from Eq. (3).
- **Step 2.** The inference model is updated using $\mathcal{L}_{\text{inf}}$ from Eq. (6).
- **Step 3 (Memory expansion).** If the temporary memory buffer is full, $|\mathcal{M}_i^t| = |\mathcal{M}_i^t|^{max}$ and $\mathcal{M}^d$ is empty, the first memory cluster $\mathcal{M}^d[1]$ is created using $\mathcal{M}_i^t$. This is used for checking the memory expansion, and if Eq. (8) is satisfied, we perform the sample selection and transfer the temporary memory buffer to a new memory cluster $\mathcal{M}^d[k+1]$ using Eq. (11) while clearing up the temporary memory buffer to avoid learning samples which are statistically similar to those already stored.

Theoretical Framework

In this section, we extend the framework from (Ye and Bors 2023c) to analyze the forgetting behaviour and provide the theoretical understanding for the Continual Variational Autoencoder (CAA).

Definition 1. Let $\mathbb{P}_{\mathcal{M}_i^t}$ represent the distribution of the memory buffer $\mathcal{M}_i^t$ at the training time ($\mathcal{T}_i$). If $\mathcal{M}^d$ has several memory clusters $\{\mathcal{M}^d[1],\cdots,\mathcal{M}^d[k]\}$, we define a probabilistic representation $\mathbb{P}_{\mathcal{M}^d[j]}$ for each cluster $\mathcal{M}^d[j]$.

Datasets	Reconstruction					Generation				
	CAA	OCM	OCM-Dynamic	Reservoir	CNDPM	CAA	OCM	OCM-Dynamic	Reservoir	CNDPM
Split MNIST	30.57	34.68	33.56	35.02	36.78	21.46	90.10	76.13	45.88	77.42
Split Fashion	72.33	75.87	73.12	102.70	75.23	67.28	123.58	121.60	99.56	123.62
Split SVHN	52.70	68.96	67.08	55.26	64.29	57.14	175.59	168.92	61.29	172.42
Split CIFAR10	119.58	120.83	122.52	123.13	124.26	74.97	173.73	170.23	93.57	175.27
Average	**68.29**	75.05	74.07	79.02	75.14	**55.21**	140.75	134.22	75.07	137.18

Table 1: FID results for the class-incremental learning paradigm, achieved by various models.

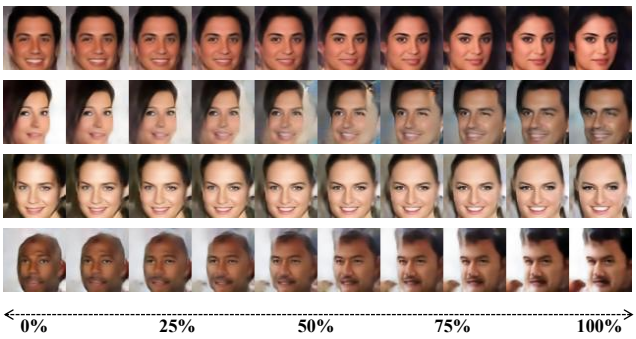

Figure 2: The image interpolation results of the proposed approach under CelebA.

Definition 2. For a given S consisting of samples from a dataset $\mathcal{D}_t^S$, let $\mathbb{P}_{\mathcal{D}_i}$ be a distribution of all visited data batches $\{\mathbf{X}_1, \cdots, \mathbf{X}_i\}$ drawn from S at $(\mathcal{T}_i)$.

In the following we analyze the forgetting behaviour of the proposed CAA by deriving a new lower bound to the sample log-likelihood.

Theorem 1. For a given data stream S, we derive a lower bound to the sample log-likelihood at $(\mathcal{T}_i)$:

$$\mathbb{E}_{\mathbb{P}_{\mathcal{D}_i}}\big[\log p_{\theta_i}(\mathbf{x})\big] \geq \mathbb{E}_{\mathbb{P}_{\mathcal{M}_i^t \otimes \mathcal{M}^d}}\big[\mathcal{L}_{ELBO}(\mathbf{x}; \theta_i, \epsilon_i)\big] - D_{JS}\big(\mathbb{P}_{\mathcal{D}_i} \,||\, \mathbb{P}_{\mathcal{M}_i^t \otimes \mathcal{M}^d}\big) - \mathcal{F}_{\mathrm{A}}\big(\mathbb{P}_{\mathcal{M}_i^t \otimes \mathcal{M}^d}, \mathbb{P}_{\mathcal{D}_i}, \mathbb{P}_{\theta_i}\big) + \mathcal{F}_{\mathrm{B}}\big(\mathbb{P}_{\mathcal{D}_i}, \mathbb{P}_{\mathcal{M}_i^t \otimes \mathcal{M}^d}\big), \tag{12}$$

where $\mathcal{F}_{\mathrm{A}}$ and $\mathcal{F}_{\mathrm{B}}$ are defined as:

$$\mathcal{F}_{\mathrm{A}}\big(\mathbb{P}_{\mathcal{M}_i^t \otimes \mathcal{M}^d}, \mathbb{P}_{\mathcal{D}_{i+1}}, \mathbb{P}_{\theta_i}\big) \triangleq \big|D_{KL}\big(\mathbb{P}_{\mathcal{M}_i^t \otimes \mathcal{M}^d} \,||\, \mathbb{P}_{\theta_i}\big) - D_{KL}\big(\mathbb{P}_{\mathcal{D}_i} \,||\, \mathbb{P}_{\theta_i}\big)\big| \tag{13}$$

$$\mathcal{F}_{\mathrm{B}}\big(\mathbb{P}_{\mathcal{D}_i}, \mathbb{P}_{\mathcal{M}_i^t \otimes \mathcal{M}^d}\big) \triangleq \mathbb{E}_{\mathbb{P}_{\mathcal{D}_i}}\big[p_{\mathcal{D}_i}(\mathbf{x}) \log p_{\mathcal{D}_i}(\mathbf{x})\big] - \mathbb{E}_{\mathbb{P}_{\mathcal{M}_i^t \otimes \mathcal{M}^d}}\big[p_{\mathcal{M}_i^t \otimes \mathcal{M}^d}(\mathbf{x}) \log p_{\mathcal{M}_i^t \otimes \mathcal{M}^d}(\mathbf{x})\big], \tag{14}$$

where $\mathbb{P}_{\mathcal{M}_i^t \otimes \mathcal{M}^d}$ represents the distribution of all memorized samples. $p_{\mathcal{M}_i^t \otimes \mathcal{M}^d}$ and $p_{\mathcal{D}_i}$ are the density functions for $\mathbb{P}_{\mathcal{M}_i^t \otimes \mathcal{M}^d}$ and $\mathbb{P}_{\mathcal{D}_i}$, respectively.

The detailed proof can be found in **Appendix-B** from SM. From Theorem 1, we observe that the memory buffers play an essential role in the performance of the proposed CAA during the training. If $\mathbb{P}_{\mathcal{M}_i^t \otimes \mathcal{M}^d}$ preserves more information about $\mathbb{P}_{\mathcal{D}_i}$, the JS divergence term $D_{JS}\big(\mathbb{P}_{\mathcal{D}_i} \,||\, \mathbb{P}_{\mathcal{M}_i^t \otimes \mathcal{M}^d}\big)$ in Eq. (12) is small, leading to a better performance for the proposed CAA.

Proposition 1. We assume that the previously seen data batches $\{\mathbf{X}_1, \cdots, \mathbf{X}_i\}$ can be divided into $\widetilde{C}_i$ sets $\{\mathcal{D}_{t,1}^S, \cdots, \mathcal{D}_{t,\widetilde{C}_i}^S\}$ at $(\mathcal{T}_i)$, with each one belonging to a data category or a task. Let $\mathbb{P}_{\mathcal{D}_{t,j}^S : \mathcal{D}_{t,\widetilde{C}_i}^S}$ represent a joint distribution of $\{\mathcal{D}_{t,1}^S, \cdots, \mathcal{D}_{t,\widetilde{C}_i}^S\}$. Then, we can derive a lower bound to the sample log-likelihood at $(\mathcal{T}_i)$:

$$\mathbb{E}_{\mathbb{P}_{\mathcal{D}_{t,j}^S : \mathcal{D}_{t,\widetilde{C}_i}^S}}\big[\log p_{\theta_i}(\mathbf{x})\big] \geq -D_{JS}\big(\mathbb{P}_{\mathcal{D}_{t,j}^S : \mathcal{D}_{t,\widetilde{C}_i}^S} \,||\, \mathbb{P}_{\mathcal{M}_i^t \otimes \mathcal{M}^d}\big) + \mathbb{E}_{\mathbb{P}_{\mathcal{M}_i^t \otimes \mathcal{M}^d}}\big[\mathcal{L}_{ELBO}(\mathbf{x}; \theta_i, \epsilon_i)\big] - \mathcal{F}_{\mathrm{A}}\big(\mathbb{P}_{\mathcal{M}_i^t \otimes \mathcal{M}^d}, \mathbb{P}_{\mathcal{D}_{t,j}^S : \mathcal{D}_{t,\widetilde{C}_i}^S}, \mathbb{P}_{\theta_i}\big) + \mathcal{F}_{\mathrm{B}}\big(\mathbb{P}_{\mathcal{D}_{t,j}^S : \mathcal{D}_{t,\widetilde{C}_i}^S}, \mathbb{P}_{\mathcal{M}_i^t \otimes \mathcal{M}^d}\big). \tag{15}$$

We provide the detailed proof in **Appendix-C** from SM. From Eq. (15), we have several observations: (1) As the training time $\mathcal{T}_i$ is growing, the number of target distributions $(\widetilde{C}_i)$ also increases, raising a challenge for the model's training. (2) The forgetting happens when the memory buffer $\mathcal{M}^d$ does not add the necessary information from the previously learnt target distributions $\{\mathbb{P}_{\mathcal{D}_{t,1}^S}, \cdots, \mathbb{P}_{\mathcal{D}_{t,\widetilde{C}_i-1}^S}\}$. (3) The knowledge diversity among the clusters of $\mathcal{M}^d$ can help capture sufficient information for all previously seen distributions and therefore reduce all JS divergence terms in Eq. (15), resulting in better performance; the memory expansion mechanism (Eq. (8)) implements this goal by adding a new memory cluster that has sufficient novel information when compared to the existing memory clusters.

Experiments

Baselines and Performance Criterion. Since this paper focuses on lifelong generative modelling, we adopt the Fréchet Inception Distance (FID) (Heusel et al. 2017) for the image generation and reconstruction evaluation, as in (Ye and Bors 2022a). We adopt Reservoir sampling (Vitter 1985) to train CAA as a baseline, named Reservoir. Other baselines are described in **Appendix-D5** from SM.

Datasets and Hyperparameters: For the class-incremental learning paradigm, we adopt the standard datasets, including Split MNIST (LeCun et al. 1998), Split SVHN (Netzer et al. 2011), Split Fashion MNIST (Fashion) (Xiao, Rasul, and Vollgraf 2017) and Split CIFAR10 (Krizhevsky and Hinton 2009) (See details in **Appendix-D2** from SM). We test the expansion threshold λ in Eq. (8) for values between

	Reconstruction					Generation				
Datasets	CAA	OCM	OCM-Dynamic	Reservoir	CNDPM	CAA	OCM	OCM-Dynamic	Reservoir	CNDPM
MNIST	26.96	29.52	32.84	28.23	33.43	245.32	239.44	168.50	298.24	217.83
SVHN	53.75	55.86	54.81	55.26	57,38	137.43	234.74	185.17	155.06	219.28
CIFAR10	120.42	121.50	124.09	123.64	124.74	87.17	283.14	242.70	65.05	277.96
Average	**67.04**	68.96	70.58	69.04	71.85	**156.64**	252.44	198.7	172.78	238.36

Table 2: FID results for image reconstruction and generation tasks under the domain-incremental learning paradigm.

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
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
Dynamic-CAA	**95.23** ± 0.05	**50.28** ± 1.16	**23.58** ± 0.57

Table 3: The classification accuracy of five independent runs for various models on three datasets.

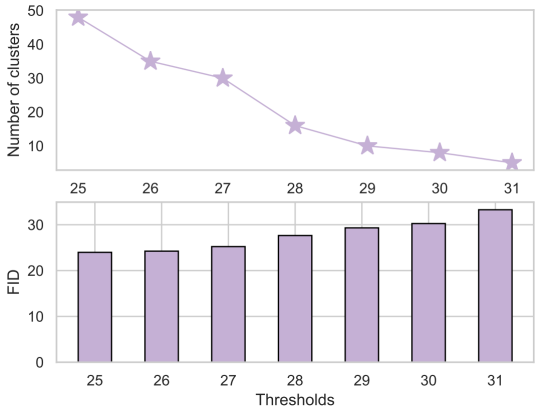

Figure 3: The performance and the number of memory clusters when changing λ in Eq. (8).

5 to 30. Then, we consider λ for Split MNIST, Split Fashion, Split SVHN and Split CIFAR10, Split MSC, CelebA and ImageNet under the generation task, as 28, 30, 29, 29, 21, 27 and 26, respectively. We also empirically find that employing $\lambda_2 = 1$ in Eq. (5) performs well. For the classification task, the final λ for Split MNIST, Split CIFAR10, Split CIFAR100, Split MImageNet and Permuted MNIS is 20, 22, 21, 25 and 25, respectively. Each memory cluster in the evolved memory buffer $\mathcal{M}^d$ can store up to 64 samples and the batch size used during each training time is 64.

Class-Incremental Generation

We evaluate the performance of various models in the class-incremental learning paradigm. We create a data stream for each dataset, namely Split MNIST, Split Fashion, Split SVHN and Split CIFAR10, as described above, where all images are resized to the resolution $32 \times 32 \times 3$. For this setting, we consider that a model can only see a batch of 64 samples from the data stream S at a training time. We restrict the maximum memory buffer size (the number of memorized samples) to 2500 for all models. Specifically, the maximum memory buffer size for the temporary buffer $\mathcal{M}^t$ and the evolved memory buffer $\mathcal{M}^d$ is 1000 and 1500, respectively. All methods use the same maximum memory capacity restriction. The results for the class-incremental learning paradigm are reported in Table 1, where the generation performance is evaluated using 5000 generated samples from each model and real testing examples from each dataset, respectively. The results from Table 1, show that the proposed CAA outperforms other baselines, including the VAE-based models, on the image reconstruction task. Moreover, the proposed CAA significantly outperforms other baselines on the image generation task, especially on the more complex CIFAR10 dataset. We provide the generation results of various models in the **Appendix-E1** from SM. We can observe that the proposed CAA achieves high-quality and diverse digit image generation when compared to other methods. These results show that CAA generates sharp and diverse images achieving a better trade-off between image reconstruction and generation performance under TFCL, than other baselines.

Domain-Incremental Generation

In this section, we evaluate the performance of various models on a more challenging learning paradigm, the domain-incremental generation. We create a data stream by collecting samples from three datasets, including MNIST, SVHN and CIFAR10. The maximum memory buffer size for $\mathcal{M}^t$ and $\mathcal{M}^d$ is 1000 and 2000, respectively. All models use a similar network architecture and memory configuration for a fair comparison. The performance of various models for the domain-incremental learning paradigm is reported in Table 2 where we can observe that the proposed CAA outperforms the other methods.

In the following, we evaluate the performance of various models on datasets containing complex images. First, we create a data stream for CelebA (Liu et al. 2015) and ImageNet (Krizhevsky, Sutskever, and Hinton 2012), respectively. The maximum memory buffer size for all models is

	Reconstruction					Generation				
Datasets	CAA	OCM	OCM-Dynamic	Reservoir	CNDPM	CAA	OCM	OCM-Dynamic	Reservoir	CNDPM
CelebA	20.41	25.23	24.26	25.72	26.27	23.64	91.88	90.30	24.70	93.29
ImageNet	84.97	88.44	82.16	86.94	89.85	37.49	225.41	210.52	86.31	224.89
Average	**52.62**	56.83	53.21	56.33	58.06	**30.56**	158.64	150.41	55.50	159.14

Table 4: FID results for complex datasets, achieved by various models.

Methods	**Split MImageNet**	**Permuted MNIST**
ER_a	25.92 ± 1.2	78.11 ± 0.7
ER + GMED	27.27 ± 1.8	78.86 ± 0.7
MIR+GMED	26.50 ± 1.3	79.25 ± 0.8
MIR	25.21 ± 2.2	79.13 ± 0.7
CNDPM	27.12 ± 1.5	80.68 ± 0.7
ODDL	27.45 ± 0.9	82.33 ± 0.6
ODDL-S	28.68 ± 1.5	83.56 ± 0.5
OCM-Dynamic	28.74 ± 1.6	84.56 ± 0.7
CAA (proposed)	**30.27** ± 1.4	**86.89** ± 0.4

Table 5: Classification accuracy for 20 runs when testing various models on Split MImageNet and Permuted MNIST. The results of all baselines, except for OCM-Dynamic, are taken from (Ye and Bors 2022b).

3000, and the results are reported in Table 4. We also provide interpolation results of the proposed CAA for CelebA in Fig. 2, which show that the proposed CAA can smoothly transform one image into another without forgetting.

Classification Task Results

Although the proposed CAA is mainly applied to the task-free continual generation task, it can be extended to be applied to the classification task. Following from (Ye and Bors 2022a), we train a classifier along with CAA, which can be seen as a mixture component in a dynamic expansion framework. We call this approach as Dynamic-CAA, employing the dynamic expansion mechanism (Ye and Bors 2022a), while expanding its capacity during the training. For the classification task, we adapt the standard TFCL experiment setting of (De Lange and Tuytelaars 2021), where the maximum memory size for Split MNIST, Split CIFAR10 and Split CIFAR100 is of 2000, 1000 and 5000, respectively. During the training, a model can only see ten samples at a training time.

The classification accuracy on Split MNIST, Split CIFAR10 and Split CIFAR100 is reported in Table 3. We can observe that the proposed approach outperforms other baselines on all datasets. We also consider Split MImageNet, which divides MINI-ImageNet (Le and Yang 2015) into 20 tasks and Permuted MNIST (Goodfellow et al. 2014b) consisting of 10 tasks, where each task assigns a random pixel permutation for all images. The maximum memory for $\mathcal{M}^t$ and $\mathcal{M}^d$ for the complex datasets, including Split MImageNet and Permuted MNIST, is of 0.1K and 0.9K, respectively. The classification accuracy is reported in Table 5, which shows that the proposed approach still performs better than other baselines on these complex datasets.

Settings			**Split MNIST**	
TSOS	$\mathcal{L}_g$	$\mathcal{L}_{g'}$	Reconstruction	Generation
✓	×	✓	**30.57**	**21.46**
✓	✓	×	33.84	34.04
×	×	✓	33.62	21.78
×	✓	×	45.72	47.56

Table 6: The performance (FID) of the proposed CAA with different configurations on Split MNIST.

Ablation Study

In this section, we investigate the effectiveness of various components of the proposed methodology.

The two-step strategy and loss functions. We consider various configurations to evaluate the performance of the proposed CAA. According to the results from Table 6, the best performance is achieved when using the loss function $\mathcal{L}_{g'}$ from Eq. (5) and the Two-Step Optimization Strategy (TSOS). On the other hand, without using TSOS, the proposed CAA still achieves good generation results. This shows that the proposed TSOS can further improve the reconstruction performance without sacrificing the quality of generated images. Furthermore, the loss $\mathcal{L}_{g'}$ can lead to better results in terms of the reconstruction and generation performance when compared with $\mathcal{L}_g$ from Eq. (2)..

Changing the threshold λ in Eq. (8)**.** We investigate the performance and the number of memory clusters when changing the threshold λ, and the results are shown in Fig. 3. We observe that when decreasing λ, the performance of CAA gradually improves while the number of memory clusters also increases. In contrast, a large threshold would reduce the number of memory clusters while the performance decreases. The parameter λ maintains a trade-off between the memory buffer size and the performance of the CAA.

Conclusion

In this paper, we propose a new model (CAA) for task-free continual generation. To avoid forgetting, we propose a new dual memory system consisting of a temporary and an evolved memory buffer for training the CAA. The proposed memory system can dynamically store diverse samples without accessing any supervised or task information. Empirical and theoretical results demonstrate the performance of the proposed CAA framework.

References

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

Aljundi, R.; Belilovsky, E.; Tuytelaars, T.; Charlin, L.; Caccia, M.; Lin, M.; and Page-Caccia, L. 2019a. Online Continual Learning with Maximal Interfered Retrieval. In *Advances in Neural Information Processing Systems (NeurIPS)*, 11872–11883.

Aljundi, R.; Kelchtermans, K.; and Tuytelaars, T. 2019. Task-free continual learning. In *Proc. of IEEE/CVF Conf. on Computer Vision and Pattern Recognition*, 11254–11263.

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

Bang, J.; Kim, H.; Yoo, Y.; Ha, J.-W.; and Choi, J. 2021. Rainbow Memory: Continual Learning with a Memory of Diverse Samples. In *Proc. of IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, 8218–8227.

Bang, J.; Koh, H.; Park, S.; Song, H.; Ha, J.-W.; and Choi, J. 2022. 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)*, 9275–9284.

Berns, G. S.; Blaine, K.; Prietula, M. J.; and Pye, B. E. 2013. Short-and long-term effects of a novel on connectivity in the brain. *Brain connectivity*, 3(6): 590–600.

Cha, H.; Lee, J.; and Shin, J. 2021. Co2l: Contrastive continual learning. In *Proc. of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 9516–9525.

Chaudhry, A.; Ranzato, M.; Rohrbach, M.; and Elhoseiny, M. 2019a. Efficient lifelong learning with A-GEM. In *Int. Conf. on Learning Representations (ICLR), arXiv preprint arXiv:1812.00420*.

Chaudhry, A.; Rohrbach, M.; Elhoseiny, M.; Ajanthan, T.; Dokania, P.; Torr, P. H. S.; and Ranzato, M. 2019b. On Tiny Episodic Memories in Continual Learning. *arXiv preprint arXiv:1902.10486*.

Chen, X.; Li, J.; Lan, X.; and Zheng, N. 2020. Generalized zero-shot learning via multi-modal aggregated posterior aligning neural network. *IEEE Trans. on Multimedia*, 24: 177–187.

De Lange, M.; and Tuytelaars, T. 2021. Continual prototype evolution: Learning online from non-stationary data streams. In *Proc. of the IEEE/CVF International Conference on Computer Vision*, 8250–8259.

Derakhshani, M. M.; Zhen, X.; Shao, L.; and Snoek, C. 2021. Kernel Continual Learning. In *Proc. Int. Conference on Machine Learning (ICML), vol. PMLR 139*, 2621–2631.

Egorov, E.; Kuzina, A.; and Burnaev, E. 2021. BooVAE: Boosting Approach for Continual Learning of VAE. *Advances in Neural Information Proc. Systems*, 17889–17901.

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

Goodfellow, I. J.; Mirza, M.; Xiao, D.; Courville, A.; and Bengio, Y. 2014b. An empirical investigation of catastrophic forgetting in gradient-based neural networks. In *arXiv preprint arXiv:1312.6211*.

Gulrajani, I.; Ahmed, F.; Arjovsky, M.; Dumoulin, V.; and Courville, A. C. 2017. Improved training of Wasserstein GANs. In *Advances in Neural Inf. Proc. Syst.*, 5767–5777.

Heusel, M.; Ramsauer, H.; Unterthiner, T.; Nessler, B.; and Hochreiter, S. 2017. GANs trained by a two time-scale update rule converge to a local Nash equilibrium. In *Advances in Neural Inf. Proc. Systems (NIPS)*, 6626–6637.

Huang, H.; He, R.; Sun, Z.; and Tan, T. 2018. IntroVAE: Introspective variational autoencoders for photographic image synthesis. In *Advances in Neural Inf. Proc. Systems*, 52–63.

Isola, P.; Zhu, J.-Y.; Zhou, T.; and Efros, A. A. 2017. Image-to-image translation with conditional adversarial networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 1125–1134.

Jin, X.; Sadhu, A.; Du, J.; and Ren, X. 2021. 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*.

Karras, T.; Laine, S.; Aittala, M.; Hellsten, J.; Lehtinen, J.; and Aila, T. 2020. Analyzing and improving the image quality of stylegan. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 8110–8119.

Kingma, D. P.; and Welling, M. 2013. Auto-encoding variational Bayes. *arXiv preprint arXiv:1312.6114*.

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

Krizhevsky, A.; and Hinton, G. 2009. Learning multiple layers of features from tiny images. Technical report, Univ. of Toronto.

Krizhevsky, A.; Sutskever, I.; and Hinton, G. E. 2012. Imagenet classification with deep convolutional neural networks. In *Advances in Neural Inf. Proc. Syst.*, 1097–1105.

Kurle, R.; Cseke, B.; Klushyn, A.; van der Smagt, P.; and Günnemann, S. 2020. Continual Learning with Bayesian Neural Networks for Non-Stationary Data. In *Proc. Int. Conf. on Learning Representations (ICLR)*.

Larsen, A. B. L.; Sønderby, S. K.; Larochelle, H.; and Winther, O. 2015. Autoencoding beyond pixels using a learned similarity metric. *arXiv preprint arXiv:1512.09300*.

Le, Y.; and Yang, X. 2015. Tiny imageNet visual recognition challenge. Technical report, Univ. of Stanford.

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

Lee, S.; Ha, J.; Zhang, D.; and Kim, G. 2020. A Neural Dirichlet Process Mixture Model for Task-Free Continual Learning. In *Int. Conf. on Learning Representations (ICLR), arXiv preprint arXiv:2001.00689*.

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

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

Lopez-Paz, D.; and Ranzato, M. 2017. Gradient episodic memory for continual learning. In *Advances in Neural Information Processing Systems (NIPS)*, 6467–6476.

Makhzani, A.; Shlens, J.; Jaitly, N.; Goodfellow, I.; and Frey, B. 2016. Adversarial autoencoders. In *International Conference on Learning Representations (ICLR), arXiv preprint arXiv:1511.05644*.

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

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

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

Rao, D.; Visin, F.; Rusu, A. A.; Teh, Y. W.; Pascanu, R.; and Hadsell, R. 2019. Continual Unsupervised Representation Learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, 7645–7655.

Seff, A.; Beatson, A.; Suo, D.; and Liu, H. 2017. Continual learning in generative adversarial nets. *arXiv preprint arXiv:1705.08395*.

Shi, Y.; Yuan, L.; Chen, Y.; and Feng, J. 2021. Continual learning via bit-level information preserving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 16674–16683.

Tomczak, J.; and Welling, M. 2018. VAE with a VampPrior. In *Proc. International Conference on Artificial Intelligence and Statistics (AISTATS), vol. PMLR 84*, 1214–1223.

Tulyakov, S.; Liu, M.-Y.; Yang, X.; and Kautz, J. 2018. MoCoGAN: Decomposing motion and content for video generation. In *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 1526–1535.

Ulyanov, D.; Vedaldi, A.; and Lempitsky, V. 2018. It takes (only) two: Adversarial generator-encoder networks. In *Proc. AAAI Conf. on Artificial Intelligence*, 1250–1257.

Varshney, S.; Verma, V. K.; Srijith, P.; Carin, L.; and Rai, P. 2021. CAM-GAN: Continual Adaptation Modules for Generative Adversarial Networks. *Advances in Neural Information Processing Systems*, 34: 15175–15187.

Vitter, J. S. 1985. Random sampling with a reservoir. *ACM Trans. on Mathematical Software (TOMS)*, 11(1): 37–57.

Wang, S.; Li, X.; Sun, J.; and Xu, Z. 2021. Training networks in null space of feature covariance for continual learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 184–193.

Wen, Y.; Tran, D.; and Ba, J. 2020. BatchEnsemble: an Alternative Approach to Efficient Ensemble and Lifelong Learning. In *Int. Conf. on Learning Representations (ICLR), arXiv preprint arXiv:2002.06715*.

Xian, Y.; Sharma, S.; Schiele, B.; and Akata, Z. 2019. f-VAEGAN-D2: A feature generating framework for any-shot learning. In *Proc. of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, 10275–10284.

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

Yan, Q.; Gong, D.; Liu, Y.; van den Hengel, A.; and Shi, J. Q. 2022. Learning Bayesian Sparse Networks with Full Experience Replay for Continual Learning. In *Proc. of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 109–118.

Ye, F.; and Bors, A. G. 2020. Learning Latent Representations Across Multiple Data Domains Using Lifelong VAEGAN. In *Proc. European Conf. on Computer Vision (ECCV), vol. LNCS 12365*, 777–795.

Ye, F.; and Bors, A. G. 2022a. Continual Variational Autoencoder Learning via Online Cooperative Memorization. In *Proc. European Conference on Computer Vision (ECCV), vol. LNCS 13683*, 531–549.

Ye, F.; and Bors, A. G. 2022b. Task-Free Continual Learning via Online Discrepancy Distance Learning. In *Advances in Neural Information Proc. Systems (NeurIPS)*, 23675–23688.

Ye, F.; and Bors, A. G. 2023a. Dynamic Self-Supervised Teacher-Student Network Learning. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 45(5): 5731–5748.

Ye, F.; and Bors, A. G. 2023b. Learning dynamic latent spaces for lifelong generative modelling. In *Proc. of the AAAI Conference on Artificial Intelligence*, 10891–10899.

Ye, F.; and Bors, A. G. 2023c. Lifelong Variational Autoencoder via Online Adversarial Expansion Strategy. In *AAAI Conference on Artificial Intelligence*, 10909–10917.

Ye, F.; and Bors, A. G. 2023d. Self-Evolved Dynamic Expansion Model for Task-Free Continual Learning. In *Proc. of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 22102–22112.

Ye, F.; and Bors, A. G. 2023e. Wasserstein Expansible Variational Autoencoder for Discriminative and Generative Continual Learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 18665–18675.

Zhai, M.; Chen, L.; He, J.; Nawhal, M.; Tung, F.; and Mori, G. 2020. Piggyback GAN: Efficient lifelong learning for image conditioned generation. In *Proc. European Conference on Computer Vision, vol. LNCS 12366*, 397–413.