Learning Dynamic Latent Spaces for Lifelong Generative Modelling

Fei Ye and Adrian G. Bors

Department of Computer Science, University of York, York YO10 5GH, UK
fy689@york.ac.uk, adrian.bors@york.ac.uk

Abstract

Task Free Continual Learning (TFCL) aims to capture novel concepts from non-stationary data streams without forgetting previously learned knowledge. Mixture models, which add new components when certain conditions are met, have shown promising results in TFCL tasks. However, such approaches do not make use of the knowledge already accumulated for positive knowledge transfer. In this paper, we develop a new model, namely the Online Recursive Variational Autoencoder (ORVAE). ORVAE utilizes the prior knowledge by selectively incorporating the newly learnt information, by adding new components, according to the knowledge already known from the past learnt data. We introduce a new attention mechanism to regularize the structural latent space in which the most important information is reused while the information that interferes with novel samples is inactivated. The proposed attention mechanism can maximize the benefit from the forward transfer for learning novel information without forgetting previously learnt knowledge. We perform several experiments which show that ORVAE achieves state-of-the-art results under TFCL.

Introduction

The Variational Autoencoder (VAE) (Kingma and Welling 2013) is one of the most popular generative models, which has been widely applied for density estimation (Kim and Pavlovic 2020; Kingma et al. 2016; Maaløe et al. 2016), disentangled representations (Higgins et al. 2017; Ye and Bors 2021a,b) and in mixture models (Ye and Bors 2022c, 2020b). However, one challenge for VAEs is that they gradually lose performance on the previously learned probabilistic representations when learning a new task. This is caused by catastrophic forgetting (Parisi et al. 2019), which occurs when the network weights are over-written by network updating when training with a new task.

Episodic memory buffers, storing some of the past samples that will be incorporated together with samples from a newly given task (Chaudhry et al. 2019b), have been used for addressing catastrophic forgetting. Other memory-based methods usually train a generator that produces a set of generative samples as memorized instances (Shin et al. 2017). These approaches, however, require knowing the task identity and class label for each data sample (Derakhshani et al. 2021), which is not feasible in TFCL.

Recent works have proposed to learn a dynamic expansion model to address TFCL. For instance, the Continual Unsupervised Representation Learning (CURL) (Rao et al. 2019) dynamically builds the inference models to capture the data distribution shift while the Generative Replay Mechanism (GRM) is used to relieve forgetting. However, CURL still causes forgetting due to the frequent generative replay processes, theoretically explained in (Ye and Bors 2021c, 2022b,f). This issue is solved in (Ye and Bors 2022b), through an approach called the Online Cooperative Memorization (OCM), which optimizes a model structure by detecting the loss change. In addition, OCM employs two different memory buffers to store short- and long-term knowledge, further improving the performance. However, these approaches usually optimize a simple structure on the latent space where the previously learnt components are not fully utilized when training on new data. These components usually contain prior information which is useful for the knowledge transfer (Phuong and Lampert 2019).

In this paper, we address TFCL from two aspects. First, we address the forgetting problem in TFCL, by proposing a new model, namely the Online Recursive Variational Autoencoder (ORVAE) which automatically preserves the prior knowledge into the trained components while expanding its network architecture to adapt to the data distribution shift. Second, as novel samples in a data stream share similar characteristics with previously learnt ones, it is important to use prior knowledge to learn novel concepts. However, directly using all past samples in TFCL is impossible. Therefore, we introduce a new recursive expansion mechanism for ORVAE, incorporating all previously learnt information flow into the inference and decoding processes when learning novel samples. Unlike CURL and CNDPM, which do not fully utilize the latent variable information when learning incoming samples, ORVAE integrates all previously learnt and the currently updated variational distributions to form an augmented distribution used for decoding, ensuring the positive knowledge transfer. In addition, the proposed recursive expansion can also be helpful in learning a compact model because accumulating more knowledge can allow adapting to new samples fast while preventing the ex-

pansion. Additionally, inspired by the attention mechanism (Vaswani et al. 2017) that has been successfully used in computer vision (Parmar et al. 2018; Liu et al. 2021), language processing (Al-Rfou et al. 2019), and inductive learning (Veličković et al. 2018), we introduce a new attention mechanism, called the Expandable Graph Attention Mechanism (EGAM) which views each previously learnt variational distribution as a node and formulates the relevance among these latent distribution representations as a graph structure. EGAM generates attention weights to regularize the graph structure when learning novel data. Different from existing attention mechanisms (Vaswani et al. 2017; Veličković et al. 2018), the proposed EGAM can automatically expand its attention region to adapt to the expansion of ORVAE, and allows ORVAE to reuse the knowledge that contributes to learning novel samples selectively.

Further, our other contribution consists of deriving a new lower bound to the data likelihood in order to understand the forgetting behaviour of ORVAE under TFCL. The proposed theoretical analysis demonstrates that the objective function of ORVAE with several bounded negative error terms guarantees a lower bound to the data likelihood in each training instance during TFCL.

We summarise our contributions as : (1) A new model ORVAE, which optimizes a graph structure on the latent space in a recursive way, is proposed for utilizing prior information when learning novel data. (2) A new attention mechanism is proposed to regularize the prior information stored in the components for ORVAE, which maximizes the knowledge transfer when learning new concepts. (3) We provide new insights into the VAE's forgetting behaviour and theoretical guarantees for ORVAE under TFCL.

Supplementary materials (SM) and source code are available[1].

Related Work

Memory based Lifelong learning is a natural solution for CL, which uses a buffer to store some past samples, then replaying them when learning a new task (Chaudhry et al. 2019a; Lopez-Paz and Ranzato 2017; Pan et al. 2020; Rebuffi et al. 2017; Cha, Lee, and Shin 2021; Tiwari et al. 2022). However, using a fixed memory is not scalable to learning an infinite number of tasks. Other memory-based approaches focus on training a generator such as a VAE or a Generative Adversarial Net (GAN) (Goodfellow et al. 2014), as a generative replay network (Achille et al. 2018; Ramapuram, Gregorova, and Kalousis 2017; Rao et al. 2019; Shin et al. 2017; Wu et al. 2018; Zhai et al. 2019; Ye and Bors 2020a, 2022a) which produces past samples when learning new tasks. However, these approaches suffer from degenerated performance when learning a long sequence of data domains due to the frequent generating processes (Ye and Bors 2020a).

Dynamic Architecture Methods (DAMs) aim to expand its network architecture by adding new neural layers and hidden nodes in order to adapt the model for learning new tasks (Ye and Bors 2020b, 2023, 2021c; Rao et al. 2019; Ye and Bors 2022d,e,g). These approaches usually divide the whole network architecture into two components (Ye and Bors 2021c), the shared and the task-specific modules, where a new component is built based on a set of shared modules which do not update in the following tasks in order to relieve catastrophic forgetting (Wen, Tran, and Ba 2020). DAMs have been used in a task-incremental scenario (Ye and Bors 2021c) and TFCL (Rao et al. 2019; Lee et al. 2020), achieving promising results. The proposed ORVAE has two different features from existing expansion approaches (Ye and Bors 2022b; Lee et al. 2020). Firstly, ORVAE learns a graph structure in the latent space in which all previously learnt variables are utilized for knowledge transfer when learning new data. Secondly, the proposed attention mechanism can generate graph attention weights that regularize the graph structure in the latent space to maximize the benefit from the positive knowledge transfer.

Regularization based approaches alleviate catastrophic forgetting by incorporating an auxiliary term that penalizes changes in the network weights when the model learns a new task (Kirkpatrick et al. 2017; Li and Hoiem 2017; Nguyen et al. 2018) or store past samples into a small memory buffer to regulate the optimization (Guo et al. 2020; Chaudhry et al. 2019a). Recently, several works have proposed to regulate the representation that is robust to forgetting in continual learning by using adversarial training processes (Ebrahimi et al. 2020) and meta-learning (Javed and White 2019). However, these approaches still require both the task identity and the class label, which can not be applied in TFCL. Additionally, they have substantial computation requirements when learning a growing number of tasks (Lopez-Paz and Ranzato 2017). The proposed ORVAE is more efficient since we do not optimize the whole network architecture using past samples. Additionally, we utilize the learned information in an end-to-end learning manner without requiring any extra iterative training processes (Ebrahimi et al. 2020; Javed and White 2019).

Methodology

Problem Setups

In this paper, we mainly focus on unsupervised generative modelling in TFCL. Let $\mathcal{X} \in \mathbb{R}^{d_{\mathbf{x}}}$ represent the data space with the dimension of $d_{\mathbf{x}}$ and $\mathcal{D}_i$ be the training set of the i-th task/dataset. Let $f^i_{class}: \mathcal{X} \rightarrow \mathcal{C}$ be a function for $\mathcal{D}_i$, which infers the exact class label for each sample where $\mathcal{C}$ is the space of the data categories. Let us divide $\mathcal{D}_i$ into several disjoint parts $\{\mathbf{X}^i_1, \cdots, \mathbf{X}^i_{N^i}\}$ where each part $\mathbf{X}^i_k$ is made up by samples satisfying $f^i_{class}(\mathbf{x}) = k,\ \mathbf{x} \sim \mathbf{X}^i_k$. N^i is the number of data categories from $\mathcal{D}_i$. We then provide three TFCL scenarios, which represent a more challenging setting than the those from (van de Ven and Tolias 2019).

Class-Incremental for a single dataset (CIASD). A stream $\mathbf{X}^S$ consists of several data categories from a single training set, resulting in $\bigcup_{j=1}^{N^i} \mathbf{X}^i_j$.

Class-Incremental for multiple datasets (CIMD). We apply CIASD for multiple datasets, each consisting of several data categories, and then a data stream $\mathbf{X}^S$ is expressed as

[1]https://github.com/dtuzi123/ORVAE

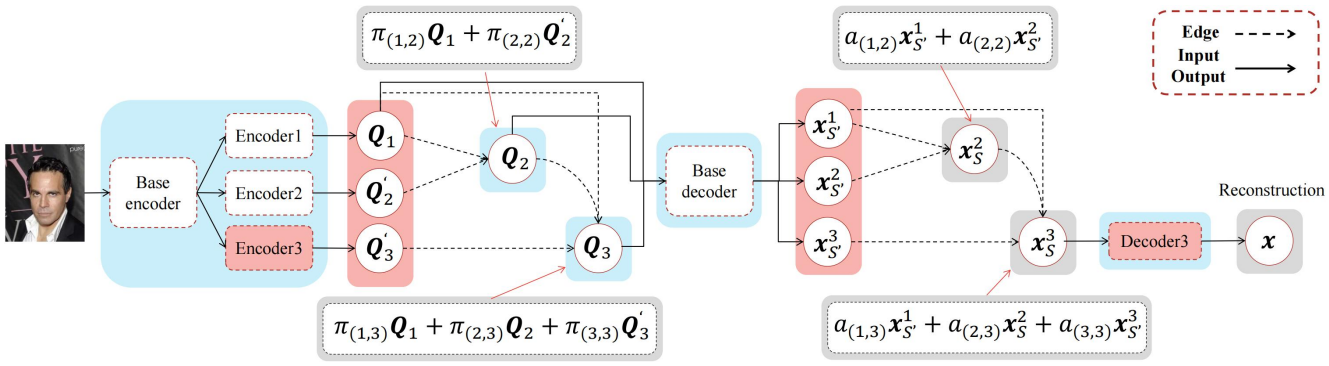

Figure 1: The network architecture of ORVAE when training the third component (only "Encoder3" and "Decoder3" are updated). Firstly, an image $\mathbf{x}$ passes three inference models ("Encoder1", "Encoder2", "Encoder3") to form three individual variational distributions $Q'_{\omega_1^*}(\mathbf{z}), Q'_{\omega_2^*}(\mathbf{z}), Q'_{\omega_3^*}(\mathbf{z})$, where $\mathbf{Q}'_2$ and $\mathbf{Q}_2$ in the figure denote $Q'_{\omega_2^*}(\mathbf{z})$ and $Q_{\omega_2^*}(\mathbf{z})$, respectively. Then we form two augmented variational distributions $Q_{\omega_2^*}(\mathbf{z})$ and $Q_{\omega_3^*}(\mathbf{z})$ in a recursive way. During the decoding, we take $\{\mathbf{z}_1, \mathbf{z}_2, \mathbf{z}_3\}$ drawn from $\{Q_{\omega_1^*}(\mathbf{z}), Q_{\omega_2^*}(\mathbf{z}), Q_{\omega_3^*}(\mathbf{z})\}$ as inputs of the base decoder which outputs the corresponding representations $\{\mathbf{x}^1_{S'}, \mathbf{x}^2_{S'}, \mathbf{x}^3_{S'}\}$, allowing us to form the augmented intermediate representation $\mathbf{x}^3_S$. Then we take $\mathbf{x}^3_S$ as the input of "Decoder3" for the image $\mathbf{x}$ reconstruction. Notice that $\{\mathbf{x}^1_{S'}, \mathbf{x}^2_{S'}, \mathbf{x}^3_{S'}\}$ are not considered as random variables.

$\bigcup_{i=1}^{M}\{\bigcup_{j=1}^{N^i}\mathbf{X}^i_j\}$, where M is the number of datasets.

General streaming setting for multiple datasets (GSSMD). A stream $\mathbf{X}^S$ consists of randomly chosen samples from multiple datasets without considering their categories, resulting in $\bigcup_{i=1}^{N}\mathcal{D}_i$.

In TFCL, we assume that there are a total of n training steps $\mathcal{T} = \{\mathcal{T}_1, \cdots, \mathcal{T}_n\}$ for $\mathbf{X}^S$ and learning each batch of images $\mathbf{X}^i_{batch} \sim \mathbf{X}^S, i = 1, \cdots, n$ is time dependent. During a certain training step $\mathcal{T}_i$, a model can only access $\mathbf{X}^i_{batch}$ and can not access the data corresponding to previous batches $\{\mathbf{X}^1_{batch}, \cdots, \mathbf{X}^{i-1}_{batch}\}$. After $\mathcal{T}_n$ is finished, we evaluate the performance on all testing data.

Preliminaries

We introduce a general generative latent variable model $p_{\theta^\star}(\mathbf{x}, \mathbf{z})$, where $\mathbf{x} \in \mathcal{X}$ and $\mathbf{z} \in \mathcal{Z}$ are the observed and the latent variables, where $\mathcal{Z} \in \mathrm{R}^{d_z}$, is the latent space with the dimension of d_z. Training this model is performed by maximizing the marginal log-likelihood $\log p_{\theta^\star}(\mathbf{x}) = \int p_{\theta^\star}(\mathbf{x}\,|\,\mathbf{z})p(\mathbf{z})\,\mathrm{d}\mathbf{z}$ with respect to the model's parameters $\theta^\star$, which is intractable due to the need of accessing all $\mathbf{z}$ in this integral. The VAE (Kingma and Welling 2013) introduces a variational distribution $Q_{\omega^\star}(\mathbf{z})$ to approximate the posterior $p_{\theta^\star}(\mathbf{z}\,|\,\mathbf{x})$, estimated using the objective function :

$$\begin{aligned}&\log p_{\theta^\star}(\mathbf{x}) \geq \mathcal{L}_{ELBO}(\mathbf{x}; \theta^\star, \omega^\star) := \\ &\mathbb{E}_{z\sim Q_{\omega^\star}(\mathbf{z})}\left[\log p_\theta(\mathbf{x}\,|\,\mathbf{z})\right] - D_{KL}\left[Q_{\omega^\star}(\mathbf{z})\,||\,p(\mathbf{z})\right],\end{aligned} \tag{1}$$

where the right-hand side (RHS) is called the Evidence Lower Bound (ELBO), and includes the negative reconstruction error and the Kullback–Leibler (KL) divergence between the variational distribution and the prior $p(\mathbf{z}) = \mathcal{N}(\mathbf{0}, \mathbf{I})$. By aiming to define a mixture model for the continuously learning framework, where some parameters can be reused, we decompose the latent representation into two parts: $\mathbf{z}_S$ over the space $\mathcal{Z}_S \in \mathbb{R}^{d_{\mathbf{zs}}}$, which is a shared latent representation, and the component-specific representation $\mathbf{z}$ over $\mathcal{Z}$, where $d_{\mathbf{zs}} > d_{\mathbf{z}}$. The inference process for $\mathbf{z}$ can be defined as $Q_{\omega^\star}(\mathbf{z}) = f_\omega \circ f_{\omega_S}(\mathbf{x})$ which is implemented by two independent networks : $f_{\omega_S}\colon \mathcal{X} \to \mathcal{Z}_S$ and $f_\omega\colon \mathcal{Z}_S \to \mathcal{Z}$. Notice that we use the superscript $\star$ to denote that the model's parameters $\omega^\star$ include the shared parts $\omega^\star = \{\omega, \omega_S\}$. The decoding process is also divided into two parts by introducing the intermediate representation $\mathbf{x}_S$ over the space $\mathcal{X}_S \in \mathbb{R}^{d_{\mathbf{xs}}}, d_{\mathbf{xs}} < d_{\mathbf{x}}$, which is inferred by a base decoder $G_{\theta_S}\colon \mathcal{Z} \to \mathcal{X}_S$. We also use a sub-decoder $G_\theta\colon \mathcal{X}_S \to \mathcal{X}$ with the decoding output implemented by $G_\theta(G_{\theta_S}(\mathbf{z}))$. $\{\omega_S, \theta_S\}$ are updated only when training for the first task and are frozen for the others. In practice, they can be extracted by a VAE trained on a large-scale dataset in order to provide fundamental feature information which can then be used for multiple tasks.

Online Recursive Variational Autoencoder

Initial network architecture. We begin with the construction of the initial network architecture for the Online Recursive Variational Autoencoder (ORVAE), which includes four sub-models $\{G_{\theta_1}, G_{\theta_S}, f_{\omega_1}, f_{\omega_S}\}$, forming the variational distribution $Q_{\omega_1^\star}(\mathbf{z}) = f_{\omega_1} \circ f_{\omega_S}(\mathbf{x})$ and the decoder $G_{\theta_1}(G_{\theta_S}(\mathbf{z}))$, respectively. The ELBO loss function for optimizing the initial architecture is defined as :

$$\begin{aligned}\mathcal{L}_{\mathrm{ORVAE}}(\mathbf{x}; \theta_1^\star, \omega_1^\star) := &\mathbb{E}_{z\sim Q_{\omega_1^\star}(\mathbf{z})}\left[\log p_{\theta_1^\star}(\mathbf{x}\,|\,\mathbf{z})\right] \\ &- D_{KL}\left[Q_{\omega_1^\star}(\mathbf{z})\,||\,p(\mathbf{z})\right],\end{aligned} \tag{2}$$

where $\theta_1^\star = \{\theta_1, \theta_S\}$ and $\omega_1^\star = \{\omega_1, \omega_S\}$ are the parameters of the first component. In TFCL, we desire to increase the model's capacity to learn the new knowledge by dynamically augmenting a mixture model in a recursive way. In the following, we describe how to build a new component based on the initial network.

Build the second component. The main function of ORVAE is to incorporate all previously learned latent representations

into the inference and decoding processes whenever learning novel information. We build a new inference model on the base encoder f_{ω_S}, represented by :

$$Q'_{\omega_2^\star}(\mathbf{z}) = f_{\omega_2} \circ f_{\omega_S}(\mathbf{x}), \tag{3}$$

where ω_2 represents the parameters of the inference model for the second component. We then incorporate the previously learned encoding distribution $Q_{\omega_1^*}(\mathbf{z})$ into the inference process, expressed by :

$$Q_{\omega_2^\star}(\mathbf{z}) = \pi_{(1,2)} Q_{\omega_1^*}(\mathbf{z}) + \pi_{(2,2)} Q'_{\omega_2^\star}(\mathbf{z}), \tag{4}$$

where $\pi_{(i,j)}$ represents the weight between the i-th component (previously learned) and the j-th component in the inference process. $Q_{\omega_2^\star}(\mathbf{z})$ is an augmented variational distribution by involving previously learned latent information. We use $Q_{\omega_i^\star}(\mathbf{z})$ and $Q'_{\omega_i^\star}(\mathbf{z})$ to represent the augmented variational distribution and the individual variational distribution, respectively, for $i > 1$.

For the decoding process, we build a sub-generator G_{θ_2} based on the base decoder G_{θ_S} and the decoding distribution $p_{\theta_2^\star}(\mathbf{x} \,|\, \mathbf{z})$ is represented by :

$$\begin{aligned} &p_{\theta_2^\star}(\mathbf{x} \,|\, \mathbf{z}) = \mathcal{N}(G_{\theta_2}(\mathbf{x}_S^2), \varepsilon \mathbf{I}), \\ &\mathbf{x}_S^2 = a_{(1,2)} G_{\theta_S}(\mathbf{z}_1) + a_{(2,2)} G_{\theta_S}(\mathbf{z_2}) \,, \end{aligned} \tag{5}$$

where $\mathbf{z}_1 \sim Q_{\omega_1^\star}(\mathbf{z}), \mathbf{z}_2 \sim Q_{\omega_2^\star}(\mathbf{z})$ and $a_{(i,j)}$ represents the importance of the intermediate representations $G_{\theta_S}(\mathbf{z}_i)$ when training the j-th component. ε and $\mathbf{I}$ are the noise variance and identity matrix. $\mathbf{x}_S^2$ is an augmented intermediate representation involving the information from all previously learned representations $\{\mathbf{z}_1, \mathbf{z}_2\}$.

Training the growing graph model. ORVAE trains a growing mixture model with an arbitrary number of components K in a recursive expansion manner in order to adapt to the data distribution shift and accumulate previously learnt information during TFCL. Firstly we define a variational distribution for the K-th component :

$$Q'_{\omega_K^\star}(\mathbf{z}) = f_{\omega_K} \circ f_{\omega_S}(\mathbf{x}) \,. \tag{6}$$

Then the augmented variational distribution is defined as :

$$Q_{\omega_K^\star}(\mathbf{z}) = \sum_{i=1}^{K-1} \left\{ \pi_{(i,K)} Q_{\omega_i^\star}(\mathbf{z}) \right\} + \pi_{(K,K)} Q'_{\omega_K^\star}(\mathbf{z}) \tag{7}$$

For the decoding process, we build a new sub-generator G_{θ_K} based on the base decoder G_{θ_S} and the decoding process is :

$$\begin{aligned} &p_{\theta_K^\star}(\mathbf{x} \,|\, \mathbf{z}) = \mathcal{N}(G_{\theta_K}(\mathbf{x}_S^K), \varepsilon \mathbf{I}), \\ &\mathbf{x}_S^K = \sum_{i=1}^{K-1} \left\{ a_{(i,K)} G_{\theta_S}(\mathbf{z}_i) \right\} + a_{(K,K)} G_{\theta_S}(\mathbf{z}_K), \end{aligned} \tag{8}$$

where $\theta_K^\star = \{\theta_K, \theta_S\}, \omega_K^\star = \{\omega_K, \omega_S\}$ are the parameter sets and $\mathbf{z}_K \sim Q_{\omega_K^*}(\mathbf{z})$. We present the network architecture of the proposed ORVAE when learning the third component in Fig. 1, where $Q_{\omega_2^\star}(\mathbf{z})$ and $Q_{\omega_3^\star}(\mathbf{z})$ are augmented by individual variational distributions $\{Q_{\omega_1^\star}(\mathbf{z}), Q'_{\omega_2^\star}(\mathbf{z}), Q'_{\omega_3^\star}(\mathbf{z})\}$, regularized by their component weights. For the decoding process, we augment $\mathbf{x}_S^3$ by using $\{\mathbf{x}_{S'}^1, \mathbf{x}_{S'}^2, \mathbf{x}_{S'}^3\}$ given by the base decoder G_{θ_S}, which is used as the input of the sub-generator G_{θ_3} for the reconstruction process. We use the reparameterization trick (Kingma and Welling 2013) for the sampling process of each variational distribution in order to ensure the differentiable optimization (See details in **Appendix-I** from the Supplementary Material (SM[1])).

Objective function. In order to embed the inference and decoding processes, described above, into the maximum likelihood framework, we propose a new loss function that guarantees a lower bound to the data likelihood (See Theorem 1), used to train the newly added K-th component, $K > 1$, in ORVAE :

$$\begin{aligned} \mathcal{L}_{\text{ORVAE}}\,(\mathbf{x}; \theta_K^\star, \omega_K^\star) := &\mathbb{E}_{\mathbf{z} \sim Q_{\omega_K}(\mathbf{z})} \left[\log p_{\theta_K^\star}(\mathbf{x} \,|\, \mathbf{z}) \right] \\ &- \sum_{i=1}^{K-1} \left\{ \pi_{(i,K)} D_{KL} \left[Q_{\omega_i^*}(\mathbf{z}) \,||\, p(\mathbf{z}) \right] \right\} \\ &- \pi_{(K,K)} D_{KL} \left[Q'_{\omega_K^*}(\mathbf{z}) \,||\, p(\mathbf{z}) \right] \,. \end{aligned} \tag{9}$$

We only update $\{\theta_K, \omega_K\}$ by maximizing Eq. (9) for training the K-th component to adapt to novel samples while freezing previously learned components to avoid forgetting.

Expandable Graph Attention Mechanism

Component weights $\{\pi_{(i,K)}, a_{(i,K)} \,|\, i = 1, \ldots, K\}$ in Eq. (9) play an important role for the knowledge transfer since they control the contribution of each component when optimizing Eq. (9). To search the optimal configuration for these component weights, we introduce the Expandable Graph Attention Mechanism (EGAM) which can dynamically optimize component (attention) weights for both the encoding and decoding processes during the continual learning. Firstly, let us describe how to generate attention weights when ORVAE builds the second component. As shown in Eq. (4) and Eq. (5), we have four component weights $\{\pi_{(1,2)}, \pi_{(2,2)}, \alpha_{(1,2)}, \alpha_{(2,2)}\}$, then we dynamically build two groups of attention parameters, denoted as $\{\zeta^e_{(1,1)}, \zeta^e_{(1,2)}\}$ and $\{\zeta^d_{(1,1)}, \zeta^d_{(1,2)}\}$, for encoders and decoders, respectively. Instead of the existing attention approaches that model the correlation between image/feature patches (Parmar et al. 2018; Shaw, Uszkoreit, and Vaswani 2018), the proposed EGAM models the correlation between learned representations when seeing a new instance $\mathbf{x}_{n+1}$ by dynamically updating the attention parameters $\{\zeta^e_{(1,2)}, \zeta^e_{(2,2)}, \zeta^d_{(1,2)}, \zeta^d_{(2,2)}\}$ with respect to the maximization of the objective function from Eq. (9) :

$$\zeta^e_{(j,2)} = \zeta^e_{(j,2)} + l_1 \nabla_{\zeta^e_{(j,2)}} \{ -\mathcal{L}_{\text{ORVAE}}(\mathbf{x}_n; \theta_2^\star, \omega_2^\star) \}, \tag{10}$$

where $j = 1, 2$ and l_1 is a the learning rate. $\mathbf{x}_n$ is the last given sample. We use Eq. (10) to update $\zeta^d_{(j,2)}, j = 1, 2$ in the same way. Then attention weights associated with $\mathbf{x}_{n+1}$ are generated by the softmax function :

$$\pi_{(j,2)} = \exp(\zeta^e_{(j,2)}) / \sum\nolimits_{i=1}^{2} \exp(\zeta^e_{(i,2)}) \,, j = 1, 2 \,. \tag{11}$$

We are also using Eq. (11) to generate $\{\alpha_{(1,2)}, \alpha_{(2,2)}\}$ based on $\{\zeta^d_{(1,2)}, \zeta^d_{(2,2)}\}$. These attention weights are then used

to regulate the variational distributions $\{Q_{\omega_1^\star}(\mathbf{z}), Q'_{\omega_2^\star}(\mathbf{z})\}$ and the representations $\{G_{\theta_S}(\mathbf{z}_1), G_{\theta_S}(\mathbf{z_2})\}$ during the inference and decoding processes. Then we update the attention parameters using Eq. (10) to adapt the new sample $\mathbf{x}_{n+1}$. When ORVAE expands the model, EGAM also adds new sets of attention parameters which are optimized using Eq. (10) to regularize the new latent structure (See details in **Appendix-J** from SM[1]). In practice, we jointly update the model and attention parameters by maximizing Eq. (9).

The Learning Algorithm

In this section, we introduce the detailed algorithm used for training ORVAE under TFCL. In order to avoid frequently building new components, we use a single memory buffer to replay a few past samples for training, denoted as $\mathcal{S}_t$ when it is updated at the training step $\mathcal{T}_t$, where $|\mathcal{S}_t|$ represents the number of its stored samples. Let $|\mathcal{S}_t|_{max}$ be the maximum size of the memory. We have the following steps for optimizing ORVAE from the training step $\mathcal{T}_t$ to $\mathcal{T}_{t+1}$ (See more details in **Appendix-A** from SM[1]). :

Step 1 (Updating the memory): We add a new data batch $\mathbf{X}_{batch}^{t+1}$ to the memory buffer $\mathcal{S}_t$, resulting in $\mathcal{S}_{(t+1)}$. If the memory buffer is overloaded, we randomly remove samples from $\mathcal{S}_{t+1}$ until its size becomes equal to $|\mathcal{S}_{(t+1)}|_{max}$.

Step 2 (Checking expansion): In this paper, we adapt a similar expansion process as in (Ye and Bors 2022b), described as follows. If $|\mathcal{S}_{(t+1)}| = |\mathcal{S}_{(t+1)}|_{max}$, then we evaluate the novelty of the incoming batch of samples $\mathbf{X}_{batch}^{t+1}$. We use a measure d^t, representing the absolute difference when evaluating the objective function from Eq. (9), calculated by the current model, on the memorized samples from $\mathcal{S}_t$ and the new batch of data $\mathbf{X}_{batch}^{t+1}$:

$$d^t = \big|\mathbb{E}_{\mathbf{x}\sim\mathcal{S}_t}[\mathcal{L}_{\text{ORVAE}}(\mathbf{x};\theta_K^t,\omega_K^t)] - \mathbb{E}_{\mathbf{x}\sim\mathbf{X}_{batch}^{t+1}}[\mathcal{L}_{\text{ORVAE}}(\mathbf{x};\theta_K^t,\omega_K^t)]\big|\,, \tag{12}$$

where $\{\theta_K^t, \omega_K^t\}$ are the parameters of the K-th mixture model trained at the training step $\mathcal{T}_t$.

Step 3 (Expansion): If $d^t > \lambda$, where λ is a threshold, ORVAE builds a new mixture component $\mathcal{M}^{(K+1)}$, in a recursive way, using Eq. (7) and (5), while $\mathcal{S}_{(t+1)}$ is the set to contain $\mathbf{X}_{batch}^{t+1}$ only at $\mathcal{T}_{(t+1)}$ in order to learn novel samples, otherwise, we randomly remove samples from $\mathcal{S}_{(t+1)}$ until its size is equal to $|\mathcal{S}_{(t+1)}|_{max}$.

Step 4 (Learning): We train ORVAE and attention parameters are updated based on $\mathcal{S}_{(t+1)}$ using Eq. (9).

Theoretical Analysis for TFCL

In this section, we introduce a new theoretical framework for analyzing the forgetting behaviour of TFCL. In the following, we consider the CIASD scenario.

Preliminaries

Notation. Let $\mathcal{D}_c$ represent a training set and we consider a data stream $\mathbf{X}^S = \bigcup_{j=1}^{N^c} \mathbf{X}_j^c$ from $\mathcal{D}_c$ according to the CIASD setting. We assume that there are n training steps for learning $\mathbf{X}^S$. Let $\mathcal{M}^t$ be a VAE model trained on $\mathcal{S}_t$ at $\mathcal{T}_t$ and τ_t' represents the distribution of $\mathcal{S}_t$. Let $\mathbb{P}_{\theta^t}$ represent the generator distribution of $\mathcal{M}^t$. We use M_k^t to represent the k-th component in ORVAE trained on $\mathcal{S}_t$ at $\mathcal{T}_t$ and $\mathbb{P}_{\theta_k^t}$ to represent the generator's distribution of M_k^t. Before we derive a lower bound on ELBO for CIASD, we firstly demonstrate that the objective function from Eq. (9) is a lower bound to the marginal log-likelihood when we can access all training samples.

Theorem 1 *For a given component M_K^t, the objective function $\mathcal{L}_{\text{ORVAE}}$ from Eq.* (9) *is a lower bound to the marginal log-likelihood:*

$$\log p_{\theta_K^*}(\mathbf{x}) = \mathcal{L}_{\text{ORVAE}}(\mathbf{x};\theta_K^*,\omega_K^*) + \mathcal{L}_{GAP}\,, \tag{13}$$

and $\mathcal{L}_{GAP} \geq 0$ is defined by :

$$D_{KL}[Q_{\omega_K^\star}(\mathbf{z})\,||\,p(\mathbf{z}\,|\,\mathbf{x})] + \pi_{(K,K)} D_{KL}\left[Q'_{\omega_K^\star}(\mathbf{z})\,||\,Q_{\omega_K^\star}(\mathbf{z})\right] + \sum_{i=1}^{K-1}\left\{\pi_{(i,K)} D_{KL}\left[Q_{\omega_i^\star}(\mathbf{z})\,||\,Q_{\omega_K^\star}(\mathbf{z})\right]\right\}. \tag{14}$$

The proof is provided in **Appendix-B** from SM[1]. From Eq. (14), we can observe that $\mathcal{L}_{GAP}$ also depends on the KL divergence between the previously learned variational distribution $Q_{\omega_i^\star}(\mathbf{z})$, $i < K$ and the current variational distribution $Q'_{\omega_K^\star}(\mathbf{z})$. However, Theorem 1 can not evaluate the generalization performance (the gap between ELBO and the data log-likelihood (Burda, Grosse, and Salakhutdinov 2015)) of ORVAE under the TFCL learning setting in which the source and target distributions are not identical. This motivates us to develop a new lower bound to ELBO, which can evaluate the generalization performance of ORVAE in each training step, described in the next section.

Online ELBO for a Single Model

Existing approaches (Burda, Grosse, and Salakhutdinov 2015; Rezende and Mohamed 2015) attempt to derive a tight ELBO to the data likelihood in order to improve the performance of the VAE. However, none of them considers TFCL. In this section, we introduce a new bound, called Online ELBO (OELBO) which assesses the model's generalization when the source distribution evolves over time. Let $\mathcal{D}_c^T$ be the testing dataset and we divide $\mathcal{D}_c^T$ into N^c parts $\{\mathbf{X}_1^{(T,c)}, \cdots, \mathbf{X}_{N^c}^{(T,c)}\}$ by using f_{class}^c, where the distribution of each $\mathbf{X}_i^{(T,c)}$ is represented by τ_i. In the following, we derive the online ELBO under this setting.

Theorem 2 ***(Online ELBO.)*** *Let $\mathcal{M}^t$ be a VAE model, which converged following training on $\mathcal{S}_t$ at $\mathcal{T}_t$. OELBO for a target set $\mathbf{X}_i^{(T,c)}$ is defined as:*

$$\mathbb{E}_{\mathbf{x}\sim\tau_i}\left[\log p_{\theta^t}(\mathbf{x})\right] \geq \mathbb{E}_{\mathbf{x}\sim\tau_t'}\left[\mathcal{L}_{ELBO}(\mathbf{x};\theta^t,\omega^t)\right] - C(\tau_i,\tau_t',\mathbb{P}_{\theta^t}) - D_{Log}(\tau_i,\tau_t')\,, \tag{15}$$

where $C(\tau_i,\tau_t',\mathbb{P}_{\theta^t})$ and $D_{Log}(\tau_i,\tau_t')$ are defined as :

$$C(\tau_i,\tau_t',\mathbb{P}_{\theta^t}) = D_{KL}(\tau_i\,||\,\tau_t') + |D_{KL}(\tau_t'\,||\,\mathbb{P}_{\theta^t}) - D_{KL}(\tau_i\,||\,\mathbb{P}_{\theta^t})|\,, \tag{16}$$

Methods	MNIST	N	Fashion	N	Split M-F	N	MFO	N
VAE-ELBO	-228.86	1	-341.63	1	-291.99	1	-1086.424	1
VAE-IW50	-211.64	1	-324.33	1	-278.42	1	-363.19	1
VAE-IW10	-214.45	1	358.12	1	-275.40	1	-348.15	1
BE-ELBO	-228.08	20	-348.86	20	-308.20	20	1171.74	20
BE-IW10	-209.09	20	-354.64	20	-295.00	20	1025.90	20
BE-IW50	-207.27	20	330.27	20	-290.93	10	-1069.29	20
CNDPM-IW50	113.15	29	-266.22	30	-242.54	30	-230.03	30
ORVAE-ELBO	**-108.21**	14	-281.76	18	**-241.40**	14	**-227.58**	13
ORVAE-IW10	-120.12	14	-278.07	19	-241.70	16	-241.70	16
ORVAE-IW50	-133.45	7	**-262.68**	13	-246.48	15	-246.48	15

Table 1: The sample log-likelihood estimation for Split MNIST, Split Fashion, Split MNIST-Fashion and MFO.

$$D_{Log}(\tau_i, \tau_t') = \mathbb{E}_{\mathbf{x}\sim\tau_t'}[p_{\tau_t'}(\mathbf{x}) \log p_{\tau_t'}(\mathbf{x})] - \mathbb{E}_{\mathbf{x}\sim\tau_i}[p_{\tau_i}(\mathbf{x}) \log p_{\tau_i}(\mathbf{x})], \quad (17)$$

*The proof is provided in **Appendix-C** from SM[1]. We call the RHS of Eq.* (15) *as $\mathcal{L}_{OELBO}(\mathbf{x}; \theta^t, \omega^t)$, which can be recovered up to the standard ELBO when the source and target distributions are equal, $\tau_i = \tau_t'$ (See details in **Appendix-C** from SM[1]). $p_{\tau_t'}(\mathbf{x})$ and $p_{\tau_i}(\mathbf{x})$ are the density functions of τ_t' and τ_i. $D_{Log}(\tau_i, \tau_t')$ is constant if τ_i and τ_t' are fixed, which can be bounded by $|D_{KL}(\tau_t' \,||\, \tau_i) - D_{KL}(\tau_i \,||\, \tau_t')|$.*

Limitations when using a single model. As shown in Eq. (15) for the Online ELBO, the marginal log-likelihood on the target distribution τ_i not only depends on ELBO estimated on the source distribution τ_t', but also relies on $C(\tau_i, \tau_t', \mathbb{P}_{\theta^t})$ where $D_{KL}(\tau_i \,||\, \tau_t')$ is crucial for the generalization performance of the model. If τ_t' is sufficiently different from τ_i, then a large $D_{KL}(\tau_i \,||\, \tau_t')$ would make ELBO on τ_t' to be far away from the data log-likelihood, $\mathbb{E}_{\mathbf{x}\sim\tau_i}\big[\log p_{\theta^t}(\mathbf{x})\big]$. In practice, a single model has limitations when trained on multiple tasks due to the fixed capacity of the memory (See **Lemma 1** from **Appendix-D** of SM).

In the following, we analyze the forgetting behaviour of ORVAE and show that ORVAE can address the limitations of a single VAE model.

Online ELBO for ORVAE

Let $\mathbf{M}^K = \{\mathrm{M}_1^{t_1}, \cdots, \mathrm{M}_K^{t_K}\}$ represent an ORVAE model which has trained K components, where each $\mathrm{M}_i^{t_i}$ converged on $\mathcal{S}_{t_i}$, at the training step $\mathcal{T}_{t_i}$.

Lemma 1 *The online ELBO of $\mathbf{M}^K$ for multiple target sets $\{\mathbf{X}_1^{(T,c)}, \cdots, \mathbf{X}_{N^c}^{(T,c)}\}$ is defined as :*

$$\sum_{i=1}^{N^c} \big\{\mathbb{E}_{\mathbf{x}\sim\tau_i}\big[\log p_\theta(\mathbf{x})\big]\big\} \geq \sum_{i=1}^{N^c} \Big\{ \max_{\mathcal{M}\in\mathbf{M}^K} \big\{ -C(\tau_i, \tau', \mathbb{P}_\theta) + \mathbb{E}_{\mathbf{x}\sim\tau'}\big[\mathcal{L}_{\mathrm{ORVAE}}(\mathbf{x}; \theta, \omega)\big] - D_{Log}(\tau_i, \tau')\big\}\Big\}. \quad (18)$$

where $\{\theta, \omega\}$ are the parameters of $\mathcal{M}$ and τ' is the distribution of the memorized samples that $\mathcal{M}$ was converged on. $\mathbb{P}_\theta$ denotes the generator distribution of $\mathcal{M}$.

	Inception Score (IS)				
Datasets	ORVAE	BE	VAE	CNDPM	ORVAE*
Split CIFAR10	3.01	2.92	2.86	2.88	2.91
Split TinyImageNet	2.92	2.53	2.82	2.46	2.67
	Fréchet Inception Distance (FID)				
Split CIFAR10	122.73	146.30	138.37	142.36	126.76
Split TinyImageNe	131.46	170.62	155.82	171.10	140.56

Table 2: Quality of the reconstruction results by various models when training with natural images under CIASD.

Remark. The proof is provided in **Appendix-E** from SM[1]. We have several key results for Lemma 1 as : 1) $\mathbf{M}^K$ achieves a tighter bound to the data log-likelihood than a single VAE model $\mathcal{M}$. 2) ORVAE can relieve the negative backward transfer by preserving prior knowledge into frozen components while performing the forward transfer by the proposed recursive expansion and attention mechanisms. 3) The objective function, Eq. (9) of ORVAE with other negative terms guarantees a lower bound to the data log-likelihood in each training step, as shown in the RHS of Eq. (18).

Experiments

The Experiment Setting

Datasets and evaluation criteria : We consider MNIST (LeCun et al. 1998), Fashion (Xiao, Rasul, and Vollgraf 2017) and OMNIGLOT (Lake, Salakhutdinov, and Tenenbaum 2015) datasets for the density estimation task. All datasets are binarized by using the setting from (Burda, Grosse, and Salakhutdinov 2015). We use the sample log-likelihood estimated from 5000 important samples (Burda, Grosse, and Salakhutdinov 2015) as the criterion for density estimation.
Baselines : We consider the following baselines: BE : We implement Batch Ensemble (BE) (Wen, Tran, and Ba 2020) as a mixture VAE model where the decoder shares parameters between components. BE also uses an episodic memory for training; VAE (VAE) : This baseline is a single VAE model with an episodic memory; CNDPM is a dynamic expansion model that uses the Dirichlet process for the expansion of components. The maximum number of components in CNDPM is restricted to 30 to avoid memory overload.

Density Estimation

Settings: We consider three settings, defined as in Problem Setups. **CIASD:** Split MNIST into ten parts according to the ten classes and create a data stream by connecting these parts orderly, one after another, denoted as Split MNIST. We repeat this for Fashion, denoted as Split Fashion; **CIMD:** We create a data stream by using samples from both Split MNIST and Split Fashion, denoted as Split MNIST-Fashion; **GSSMD:** We create a data stream by using ransom samples from MNIST, Fashion and OMNIGLOT, denoted as MFO.

The results for the density estimation task are shown in Table 1, where 'VAE-IW10' and 'VAE-ELBO' represent

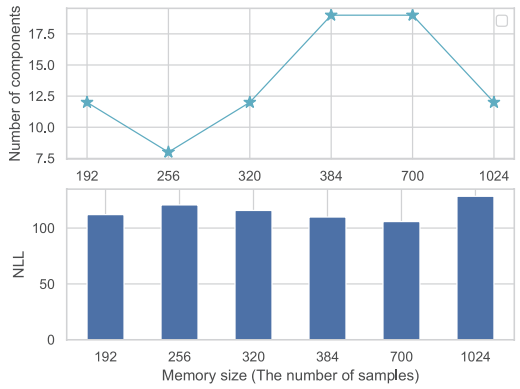

Figure 2: Performance when changing memory size.

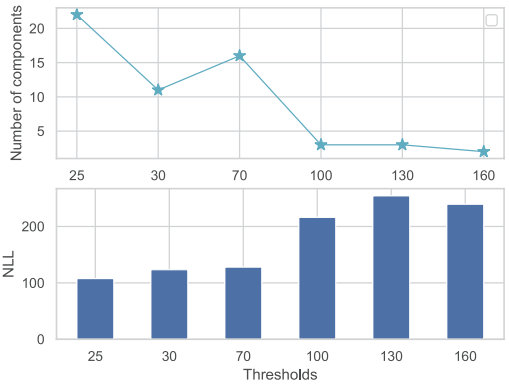

Figure 3: Performance when changing λ for Eq. (12).

Methods	Split MNIST	Split CIFAR10	Split CIFAR100
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
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
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
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
ORVAE	**94.07** ± 0.13	**50.43** ± **0.15**	**22.83** ± 0.25

Table 3: Classification accuracy for five independent runs for various models on three datasets.

VAE using Importance Weighted VAE (Burda, Grosse, and Salakhutdinov 2015) (using 10 importance samples) and ELBO as the objective functions.

The threshold λ, controlling the size of the architecture in Eq. (12), is set to 30 and 40 for Split MNIST and Split Fashion, respectively. The maximum number of samples in the memory is set to 512. We can observe that the importance sampling can not improve the performance since all models are trained under TFCL. From Table 1 we can see that the proposed ORVAE outperforms other baselines by a large margin for the density estimation tasks.

Since Split MNIST-Fashion and MFO require a large number of training steps, we set $\lambda = 500$ to control the total number of components. However, under such a challenging setting, the proposed ORVAE still achieves the state of the art results, outperforming CNDPM-IWELBO50, even for a smaller number of components, as shown in Table 1.

Evaluation of Generative Modelling Capability

We evaluate the generative ability of various models for CIFAR10 (Krizhevsky and Hinton 2009) and Tiny-ImageNet (Le and Yang 2015) datasets. For this experiment, we divide CIFAR10 and Tiny-ImageNet into ten parts, respectively, denoted as Split CIFAR10 (SC) and Split Tiny-ImageNet (STI). We consider $\lambda = 80$ for Eq. (12) and the maximum number of samples in the memory as 512. All models use ELBO with a small weight of 0.01 for the KL divergence term to avoid over-regularisation (Ye and Bors 2022c). From Table 2, the results for the Fréchet Inception Distance (FID) (Heusel et al. 2017) and Inception Score (IS) (Salimans et al. 2016) indicate that ORVAE outperforms other models in reconstruction quality. In Table 2, ORVAE* indicates that the shared module is pre-trained on other datasets, see more details in **Appendix-F** from SM[1].

Classification Task

This section considers supervised learning by training classifiers with ORVAE. We adapt the learning setting and network architecture from (De Lange and Tuytelaars 2021) (see details in **Appendix-G.5, G.6** from SM[1]). The results for all datasets are reported in Table 3, where the results from other baselines are cited from (De Lange and Tuytelaars 2021). These results show that the ORVAE outperforms CNDPM in every dataset using fewer parameters.

Ablation Study

We study the performance of ORVAE when changing the memory size. We train ORVAE when considering 192, 256, 320, 384, 700, and 1024 samples in the memory, on Split MNIST, and the results are reported in Fig. 2. A large memory does not ensure optimal performance, while the size of 320 or 256 sets the balance between performance and the model's complexity. In Fig. 3, we provide the results for ORVAE when varying the threshold λ. A large λ would lead to a compact network, while a small λ provides a growing number of components while improving the performance as well. More results can be seen in **Appendix-G** from SM[1].

Conclusion

A new model, the Online Recursive Variational Autoencoder (ORVAE)is proposed for addressing TFCL. ORVAE can accumulate knowledge by expanding its architecture without forgetting. Furthermore, a new attention mechanism is proposed to regulate the structural latent spaces to fully utilize previously learned representation information when learning novel samples. The empirical results demonstrate the performance and scalability of the proposed ORVAE in TFCL.

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 *Proc. Advances in Neural Inf. Proc. Systems (NIPS)*, 9873–9883.

Al-Rfou, R.; Choe, D.; Constant, N.; Guo, M.; and Jones, L. 2019. Character-level language modeling with deeper self-attention. In *Proc. of the AAAI Conference on Artificial Intelligence*, volume 33, 3159–3166.

Burda, Y.; Grosse, R.; and Salakhutdinov, R. 2015. Importance weighted autoencoders. *arXiv preprint arXiv:1509.00519*.

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

De Lange, M.; and Tuytelaars, T. 2021. Continual prototype evolution: Learning online from non-stationary data streams. In *Proceedings 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. International Conference on Machine Learning (ICML), vol. 139*, 2621–2631.

Ebrahimi, S.; Meier, F.; Calandra, R.; Darrell, T.; and Rohrbach, M. 2020. Adversarial Continual Learning. In *Proc. European Conf on Computer Vision (ECCV), vol. LNCS 12356*, 386–402.

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

Guo, Y.; Liu, M.; Yang, T.; and Rosing, T. 2020. Improved Schemes for Episodic Memory-based Lifelong Learning. In *Advances in Neural Information Processing Systems (NeurIPS)*.

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 Information Processing Systems (NIPS)*, 6626–6637.

Higgins, I.; Matthey, L.; Pal, A.; Burgess, C.; Glorot, X.; Botvinick, M.; Mohamed, S.; and Lerchner, A. 2017. β-VAE: Learning basic visual concepts with a constrained variational framework. In *Proc. Int. Conf. on Learning Representations (ICLR)*.

Javed, K.; and White, M. 2019. Meta-Learning Representations for Continual Learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, 1818–1828.

Kim, M.; and Pavlovic, V. 2020. Recursive Inference for Variational Autoencoders. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 33, 19632–19641.

Kingma, D. P.; Salimans, T.; Jozefowicz, R.; Chen, X.; Sutskever, I.; and Welling, M. 2016. Improved Variational Inference with Inverse Autoregressive Flow. In *Advances in Neural Information Processing Systems (NIPS)*, volume 29, 4743–4751.

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.

Lake, B. M.; Salakhutdinov, R.; and Tenenbaum, J. B. 2015. Human-level concept learning through probabilistic program induction. *Science*, 350(6266): 1332–1338.

Le, Y.; and Yang, X. 2015. Tiny imagenet visual recognition challenge. *CS 231N*, 7(7): 3.

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 *Proc. Int. Conference 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.; Lin, Y.; Cao, Y.; Hu, H.; Wei, Y.; Zhang, Z.; Lin, S.; and Guo, B. 2021. Swin Transformer: Hierarchical Vision Transformer using Shifted Windows. In *Proc. of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 10012–10022.

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

Maaløe, L.; Sønderby, C. K.; Sønderby, S. K.; and Winther, O. 2016. Auxiliary deep generative models. In *Proc. Int. Conf. on Machine Learning (ICML) vol. PMLR 48*, 1445–1453.

Nguyen, C. V.; Li, Y.; Bui, T. D.; and Turner, R. E. 2018. Variational continual learning. In *Proc. of Int. Conf. on Learning Representations (ICLR), arXiv preprint arXiv:1710.10628*.

Pan, P.; Swaroop, S.; Immer, A.; Eschenhagen, R.; Turner, R.; and Khan, M. E. 2020. Continual Deep Learning by Functional Regularisation of Memorable Past. In *Advances in Neural Information Processing Systems (NIPS)*, 4453–4464.

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.

Parmar, N.; Vaswani, A.; Uszkoreit, J.; Kaiser, L.; Shazeer, N.; Ku, A.; and Tran, D. 2018. Image transformer. In *Proc. International Conference on Machine Learning, vol. PMLR 80*, 4055–4064.

Phuong, M.; and Lampert, C. 2019. Towards Understanding Knowledge Distillation. In *Proc. International Conference on Machine Learning (ICML), vol. PMLR 97*, 5142–5151.

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 Inf. Proc. Systems (NeurIPS)*, 7645–7655.

Rebuffi, S.-A.; Kolesnikov, A.; Sperl, G.; and Lampert, C. H. 2017. icarl: Incremental classifier and representation learning. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, 2001–2010.

Rezende, D. J.; and Mohamed, S. 2015. Variational inference with normalizing flows. In *Proc. Int. Conf. on Machine Learning (ICML), vol. PMLR 37*, 1530–1538.

Salimans, T.; Goodfellow, I.; Zaremba, W.; Cheung, V.; Radford, A.; and Chen, X. 2016. Improved techniques for training GANs. In *Advances in Neural Inf. Proc. Systems (NIPS)*, 2234–2242.

Shaw, P.; Uszkoreit, J.; and Vaswani, A. 2018. Self-attention with relative position representations. In *Proc. Conf. of the Association for Computational Linguistics (NAACL)*, 464–468.

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

Tiwari, R.; Killamsetty, K.; Iyer, R.; and Shenoy, P. 2022. GCR: Gradient Coreset Based Replay Buffer Selection for Continual Learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 99–108.

van de Ven, G. M.; and Tolias, A. S. 2019. Three scenarios for continual learning. In *Proc. NIPS Continual Learning Workshop, arXiv preprint arXiv:1904.07734*.

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, Ł.; and Polosukhin, I. 2017. Attention is all you need. In *Advances in neural information processing systems*, 5998–6008.

Veličković, P.; Cucurull, G.; Casanova, A.; Romero, A.; Liò, P.; and Bengio, Y. 2018. Graph Attention Networks. In *International Conference on Learning Representations (ICLR), arXiv preprint arXiv:1710.10903*.

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

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

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

Ye, F.; and Bors, A. 2022a. Lifelong Teacher-Student Network Learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(10): 6280–6296.

Ye, F.; and Bors, A. G. 2020a. 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. 2020b. Mixtures of variational autoencoders. In *2020 Tenth International Conference on Image Processing Theory, Tools and Applications (IPTA)*, 1–6. IEEE.

Ye, F.; and Bors, A. G. 2021a. InfoVAEGAN: Learning Joint Interpretable Representations by Information Maximization and Maximum Likelihood. In *Proc. IEEE International Conference on Image Processing (ICIP)*, 749–753.

Ye, F.; and Bors, A. G. 2021b. Learning joint latent representations based on information maximization. *Information Sciences*, 567: 216–236.

Ye, F.; and Bors, A. G. 2021c. Lifelong infinite mixture model based on knowledge-driven Dirichlet process. In *Proc. of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 10695–10704.

Ye, F.; and Bors, A. G. 2022b. 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. 2022c. Deep Mixture Generative Autoencoders. *IEEE Transactions on Neural Networks and Learning Systems*, 33(10): 5789 –5803.

Ye, F.; and Bors, A. G. 2022d. Dynamic Self-Supervised Teacher-Student Network Learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

Ye, F.; and Bors, A. G. 2022e. Learning an evolved mixture model for task-free continual learning. In *Proc. IEEE Int. Conf. on Image Processing (ICIP)*, 1936–1940.

Ye, F.; and Bors, A. G. 2022f. Lifelong Generative Modelling Using Dynamic Expansion Graph Model. In *Proc. AAAI on Artificial Intelligence*, 8857–8865.

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

Ye, F.; and Bors, A. G. 2023. Lifelong Mixture of Variational Autoencoders. *IEEE Transactions on Neural Networks and Learning Systems*, 34(1): 461–474.

Zhai, M.; Chen, L.; Tung, F.; He, J.; Nawhal, M.; and Mori, G. 2019. Lifelong GAN: Continual Learning for Conditional Image Generation. In *Proc. IEEE/CVF International Conference on Computer Vision (ICCV)*, 2759–2768.