The Thirty-Seventh AAAI Conference on Artificial Intelligence (AAAI-23)

Lifelong Compression Mixture Model via Knowledge Relationship Graph

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) represents a challenging scenario for lifelong learning because the model, under this paradigm, does not access any task information. The Dynamic Expansion Model (DEM) has shown promising results in this scenario due to its scalability and generalisation power. However, DEM focuses only on addressing forgetting and ignores minimizing the model size, which limits its deployment in practical systems. In this work, we aim to simultaneously address network forgetting and model size optimization by developing the Lifelong Compression Mixture Model (LGMM) equipped with the Maximum Mean Discrepancy (MMD) based expansion criterion for model expansion. A diversity-aware sample selection approach is proposed to selectively store a variety of samples to promote information diversity among the components of the LGMM, which allows more knowledge to be captured with an appropriate model size. In order to avoid having multiple components with similar knowledge in the LGMM, we propose a data-free component discarding mechanism that evaluates a knowledge relation graph matrix describing the relevance between each pair of components. A greedy selection procedure is proposed to identify and remove the redundant components from the LGMM. The proposed discarding mechanism can be performed during or after the training. Experiments on different datasets show that LGMM achieves the best performance for TFCL.

Introduction

Continual learning (CL) represents an innate ability of humans that enables them to adapt to an evolving environment. Due to its natural property of acquiring new knowledge and skills without forgetting CL has become an important direction of research in deep learning. CL enable tackling real-world online problems such as autonomous driving and streaming services. However, existing Deep Learning models do not fulfil CL requirements as they tend to quickly forget the previously learnt knowledge when trying to learn a new task. Therefore, conventional models suffer from massive degeneration on previous tasks and this phenomenon is called catastrophic forgetting, (Parisi et al. 2019).

Most existing studies assume a sequence of multiple tasks, where each task is assigned by sampling multiple incremental classes and the model can access the task descriptor during training (Chaudhry et al. 2019). In this study, we consider a more realistic scenario in CL, called Task-Free Continual Learning (TFCL) (Aljundi, Kelchtermans, and Tuytelaars 2019; Ye and Bors 2022f), where task identities and descriptors are not available during training. An efficient and simple approach to reduce forgetting in TFCL is to manage a small fixed capacity memory buffer (De Lange and Tuytelaars 2021; Jin et al. 2021; Bang et al. 2021) that aims to store and replay past samples during training, called the memory-based approach. Such an approach is suitable for learning a fixed-length data stream and cannot handle infinite data streams. Recently, the Dynamic Expansion Model (DEM) (Lee et al. 2020; Rao et al. 2019; Ye and Bors 2022f,a) has shown impressive results on TFCL due to its scalability and generalisation performance. DEM is enabled with an architecture expansion mechanism, which increases the model's capacity to address the data distribution shift. However, these approaches only focus on addressing forgetting and ignore the model's architecture optimization, limiting their applicability in practical systems.

In this paper, we aim to simultaneously address catastrophic forgetting and model's architecture minimization for DEM through two goals : (1) Firstly, we want to promote the learnt knowledge diversity among trained components to avoid missing previously seen information; (2) Secondly, to compress DEM's architecture by removing superfluous components from DEM in order to ensure good performance with a minimal architecture. To realise these objectives, we propose a diversity-aware sample selection approach that selectively stores the different samples with respect to the knowledge stored in each learned component of DEM. Then, we introduce a new expansion criterion that evaluates the difference between the current memory buffer and the previously learned knowledge. We propose to estimate the Maximum Mean Discrepancy (MMD) on the feature space between the memory buffer and each trained component as the expansion signal. By choosing MMD in the expansion criterion can avoid substantial computational costs and enables unsupervised learning. Second, to compress DEM's architecture, we introduce a new mixture component discarding mechanism that selectively removes components from DEM without compromising its performance. The proposed discarding mechanism identifies those components that have

learnt similar knowledge. We first construct the relationship graph between the trained components, where the similarity between two components is evaluated using the Fréchet Inception Distance (FID) (Heusel et al. 2017) estimated on the pseudo samples generated by themselves. We then select a pair of components with the highest learnt knowledge similarity. We also propose a diversity selection criterion which decides to remove one of the selected components without reducing the diversity of the knowledge of the remaining components. Finally, we implement these mechanisms under a greedy algorithm framework where we iteratively remove the redundant components from DEM while retaining only those essential. The proposed discarding mechanism has several advantages over other compression methods, including Pruning (Zhu and Gupta 2017) and Knowledge Distillation (KD) (Ye and Bors 2022e) : (1) It does not require samples or the task information compared with KD, and consequently is more suitable for TFCL; (2) It can reduce the number of components, leading to a reduction in the necessary inference time. However, Pruning (Hung et al. 2019) only reduces the number of parameters and can not reduce the number of components for the DEM, which still requires significant inference times due to the component selection process.

We summarize our contributions as follows : (1) We propose a novel expansion criterion that evaluates the discrepancy between the learned knowledge and the incoming samples using MMD, aiming to learn non-overlapping data distributions by the DEM model. (2) We propose a diversity-aware sample selection approach that selectively stores samples that are different with respect to the already learned knowledge, which further promotes knowledge diversity among the trained components. (3) We develop a new mixture component discarding mechanism that optimizes DEM architecture by selectively removing unnecessary components. The proposed mechanism does not require any supervision signals and thus can be used in both supervised and unsupervised learning. (4) We evaluate our model against the standard TFCL benchmarks and the results show the effectiveness of the proposed model. Supplementary materials (SM) and source code are available[1].

Related Works

Memory-based continual learning methods have been shown to mitigate catastrophic forgetting. They can be divided into those using a small memory buffer and those using a generative replay network. Methods from the former category usually preserve data from the previous tasks and use them to regularise the model optimisation (Hinton, Vinyals, and Dean 2014; Jung et al. 2018; Kirkpatrick et al. 2017; Kurle et al. 2020; Li and Hoiem 2017; Nguyen et al. 2018; Polikar et al. 2001; Ren et al. 2017; Ritter, Botev, and Barber 2018; Rebuffi et al. 2017; Cha, Lee, and Shin 2021; Yan et al. 2022; Bang et al. 2022; Gu et al. 2022; Tiwari et al. 2022). Approaches from the latter category aim to train a generator such as a Variational Autoencoder (VAE) (Kingma and Welling 2013) or a Generative Adversarial Nets (GANs) (Goodfellow et al. 2014) that generate data similar to the past samples (Achille et al. 2018; Ramapuram, Gregorova, and Kalousis 2017; Shin et al. 2017; Ye and Bors 2020a,b, 2022e; Zhai et al. 2019; Ye and Bors 2021b). These approaches have a fixed capacity and are not scalable when learning multiple tasks.

The Dynamic Expansion Model (DEM) would expand its learning capacity by increasing the number of network layers and/or processing nodes to handle new knowledge (Cortes et al. 2017; Ye and Bors 2022b; Li and Hoiem 2017; Ye and Bors 2022c; Rao et al. 2019; Rusu et al. 2016; Wen, Tran, and Ba 2020; Xiao et al. 2014; Ye and Bors 2020c, 2021a, 2023). Such models prevent catastrophic forgetting by freezing the weights of the previously learned network (Rusu et al. 2016) or by splitting the whole model into general and task-specific components, the latter of which can be extended for learning a growing number of tasks, (Ye and Bors 2021a). The DEM has several advantages over a single model, such as scalability and performance and can guarantee optimal performance for each learned task if the number of components matches the number of tasks, as shown theoretically and empirically in (Ye and Bors 2021a, 2022d).

Recent work has drawn attention to a more complex scenario in CL, where task identities are not available. One of the attempts for addressing TFCL introduces the use of a slight memory buffer to store and replay some past samples during training. To avoid overloading the memory, a suitable memory management method is required. This approach was first explored in (Aljundi, Kelchtermans, and Tuytelaars 2019) for training a classifier under TFCL. This was then extended to train both the classifier and VAEs (Aljundi et al. 2019a) by using a new retrieval mechanism that selectively stores the most distinct samples, called Maximal Interfered Retrieval (MIR). Then, the Gradient Sample Selection (GSS) was introduced in (Aljundi et al. 2019b), where sample selection is formulated in the memory as a constrained optimisation reduction. 'Sample selection is also performed in a *learner-evaluator* framework called the Continual Prototype Evolution (CoPE) (De Lange and Tuytelaars 2021), which maintains the same number of samples for each class in memory, ensuring the balance of data for each class. Another approach to TFCL, called Gradient-based Memory EDiting (GMED) (Jin et al. 2021), modifies stored data such that edited samples would increase the loss in the upcoming model updates. However, these approaches depend on a single memory system that cannot capture the entire information from the data stream. This inspires several attempts to apply DEM to TFCL. The first attempt of applying DEM to TFCL was the Continual Unsupervised Representation Learning (CURL) (Rao et al. 2019), where new inference models are inserted into a VAE framework to capture new knowledge when a certain expansion criterion is met. Then, similar ideas are used in the Continual Neural Dirichlet Process Mixture (CN-DPM) (Lee et al. 2020), which dynamically builds a VAE model as an expert in a mixture system using the Dirichlet processes. More recently, the Online Cooperative Memorization (OCM) (Ye and Bors 2022a) employs two memory buffers to store the short- and long-term information from the data stream in a novel DEM approach

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

to TFCL. However, these approaches lead to non-optimal architectures, which do not consider ensuring the learning of a diversity of information.

Methodology

Dynamic Expansion Model (DEM)

Let $\mathcal{X}$ and $\mathcal{Y}$ be the space of observation and target label, respectively, and $\mathcal{Z}$ denote the feature space. Let $\mathcal{T} = \{\mathcal{T}_1, \cdots, \mathcal{T}_n\}$ be a set of training steps and $\mathcal{D}$ be a data stream. We assume that learning $\mathcal{D}$ requires n training steps, $\mathcal{D} = \bigcup_{i=1}^{n} \mathcal{D}_i$, proceeding continuously as $\mathcal{D}_i = \{\mathbf{x}_j, y_j\}_{j=1}^{b}$ representing each time the i-th batch of samples and $b = |\mathcal{D}_i|$ is the i-th batch size. In TFCL, a model can only access the observation $\mathbf{x}_j$ and the target y_j from $\mathcal{D}_i$ at a certain training step ($\mathcal{T}_i$) and can not access all previously learnt batches $\{\mathcal{D}_1, \cdots, \mathcal{D}_{i-1}\}$. We evaluate the performance of the model against the test dataset, before completing n training steps.

A simple and effective approach to reduce forgetting in TFCL is to use a small memory buffer of real training samples, denoted as $\mathcal{M}_i$ updated at $\mathcal{T}_i$. Following from (Ye and Bors 2022a,f), we define a dynamic expansion model $\mathcal{G} = \{S_1, \cdots, S_t\}$ that has already learnt t components, we only update the current component S_t on $\mathcal{M}_i$, while freezing the other $t-1$ trained components at $\mathcal{T}_i$ to ensure their learnt information preservation. Each component S_t consists of a classifier C_t and a Variational Autoencoder (VAE) (Kingma and Welling 2013) V_t that is used for the component selection at the testing phase. We implement C_t through a fully connected or a convolutional network $f_{\xi_t} : \mathcal{X} \rightarrow \mathcal{Y}$, parameterized by ξ_t, used for the classification task. We implement V_t using two networks $f^v_{\omega_t} : \mathcal{X} \rightarrow \mathcal{Z}$ and $f^v_{\theta_t} : \mathcal{Z} \rightarrow \mathcal{X}$, where the former models an encoding distribution $q_{\omega_t}(\mathbf{z} \,|\, \mathbf{x})$ and the latter models a decoding distribution $p_{\theta_t}(\mathbf{x} \,|\, \mathbf{z})$, assumed to be Gaussian. We define the loss functions used to train S_t on $\mathcal{M}_i$ at $\mathcal{T}_i$ as :

$$\mathcal{L}_C(S_t, \mathcal{M}_i) \triangleq \frac{1}{|\mathcal{M}_i|} \sum\nolimits_{j=1}^{|\mathcal{M}_i|} \left\{ \mathcal{L}_{ce}\big(f_{\xi_t}(\mathbf{x}_j), y_j\big) \right\}, \quad (1)$$

$$\begin{aligned} \mathcal{L}_V(S_t, \mathcal{M}_i) \triangleq\ & \mathbb{E}_{q_{\omega_t}(\mathbf{z} \,|\, \mathbf{x})} \left[\log p_{\theta_t}(\mathbf{x}_t \,|\, \mathbf{z}) \right] \\ & - D_{KL} \left[q_{\omega_t}(\mathbf{z} \,|\, \mathbf{x}_t) \,||\, p(\mathbf{z}) \right], \end{aligned} \quad (2)$$

where $\mathcal{L}_{ce}$ is the cross-entropy loss and $|\mathcal{M}_i|$ is the number of samples stored in $\mathcal{M}_i$. $\mathcal{L}_C$ is the classification loss used for training C_t and $\mathcal{L}_V$ is the VAE loss for training V_t. $D_{KL}(\cdot||\cdot)$ and $p(\mathbf{z}) = \mathcal{N}(0, 1)$ are the Kullback–Leibler (KL) divergence and the prior distribution. In the following section, we introduce a model expansion mechanism based on the Maximum Mean Discrepancy (MMD) criterion.

Model Expansion Mechanism

Maximum Mean Discrepancy : MMD is a distance on the space of probability measures, which is based on the notion of embedding probabilities in a Reproducing Kernel Hilbert space (RKHS) (Tolstikhin, Sriperumbudur, and Schölkopf 2016). Let P and Q represent two Borel probability measures while X and U represent random variables over a topological space $\mathcal{X}$. Let $f \colon \mathcal{X} \rightarrow \mathbf{R} \in \mathcal{F}$ be a function, where $\mathcal{F}$ is a class of functions. The Maximum Mean Discrepancy (MMD) between P and Q, represents an integral probability metric, defined as :

$$\mathcal{L}_{\mathrm{M}}(P, Q) \triangleq \sup_{f \in \mathcal{F}} \left(\mathbb{E}_{x \sim P} \left[f(x) \right] - \mathbb{E}_{u \sim Q} \left[f(u) \right] \right). \quad (3)$$

We have $\mathcal{L}_{\mathrm{M}}(P, Q) = 0$, if $P = Q$. In this paper, we consider the function class $\mathcal{F}$ to be a unit ball in a RKHS with a positive definite kernel $k(x, x')$ and RKHS has the reproducing property $f \in \mathcal{F}, f(x) = \langle f\ , k(x, \cdot) \rangle$. This induces the squared population MMD based on the kernel functions :

$$\begin{aligned} \mathcal{L}^2_{\mathrm{M}}(P, Q) = & \mathbb{E}_{x, x' \sim P}[k(x, x')] - 2\mathbb{E}_{x \sim P, u \sim Q}[k(x, u)] \\ & + \mathbb{E}_{u, u' \sim Q}[k(u, u')], \end{aligned} \quad (4)$$

where x' and u' are independent copies of x and u, respectively. Usually, we draw the same number of samples from P and Q ($N^P = N^Q$), where N^P and N^Q denotes the number of samples for P and Q, respectively. Then an unbiased empirical estimate for Eq. (4) is defined as :

$$\mathcal{L}^e_{\mathrm{M}}(P, Q) = \frac{1}{N^P(N^p - 1)} \sum\nolimits_{i \neq j}^{N^P} \Big\{ h(i, j) \Big\}, \quad (5)$$

where $h(i, j) = k(x_i, x_j) + k(u_i, u_j) - k(x_i, u_j) - k(x_j, u_i)$.

MMD-based model expansion : In the following, we explain how the MMD measure from Eq. (5) can be used as an expansion criterion. We assume that a dynamic expansion model $\mathcal{G} = \{S_1, \cdots, S_t\}$ has already trained t components while the t-th component S_t, is currently trained on $\mathcal{M}_i$ at $\mathcal{T}_i$. Since we want to promote probabilistic diversity between the trained components, we maximize the distance between each trained component and the distribution of the current memory as :

$$c^\star = \arg \max_{c=i, i+1, \cdots, n} \sum\nolimits_{j=1}^{t-1} \Big\{ \mathcal{L}^e_{\mathrm{M}}\big(\mathbb{P}_{\tilde{\mathbf{z}}_j}, \mathbb{P}_{\mathbf{z}_{\mathcal{M}_c}}\big) \Big\}, \quad (6)$$

where $\mathbb{P}_{\tilde{\mathbf{z}}_j}$ is the distribution formed by the latent variable $\{\tilde{\mathbf{z}}_{j,1}, \cdots, \tilde{\mathbf{z}}_{j,m}\}$, where each $\tilde{\mathbf{z}}_{j,s}$ is given by the inference model of V_j that receives the sample $\tilde{\mathbf{x}}_{j,s}$ generated by itself. $\mathbb{P}_{\mathbf{z}_{\mathcal{M}_c}}$ is the distribution formed by the latent variables $\{\mathbf{z}_1, \cdots, \mathbf{z}_m\}$ and each $\mathbf{z}_s$ is given by the inference model of the current component V_t that receives the stored sample $\mathbf{x}_s$ drawn from the memory $\mathcal{M}_c$. $c^\star$ is the optimal index of the training step in Eq. (6) which can be seen as a recursive optimization problem when $\mathcal{G}$ is expanded (t is increased). However, searching for the optimal $c^\star$ needs to access all training steps, which would lead to severe forgetting since we can not revisit all past samples. Therefore, we introduce a practical solution to implement the optimization problem (Eq. (6)) by involving the expansion criterion expressed as :

$$\min \Big\{ \mathcal{L}^e_{\mathrm{M}}\big(\mathbb{P}_{\tilde{\mathbf{z}}_1}, \mathbb{P}_{\mathbf{z}_{\mathcal{M}_c}}\big), \cdots, \mathcal{L}^e_{\mathrm{M}}\big(\mathbb{P}_{\tilde{\mathbf{z}}_{t-1}}, \mathbb{P}_{\mathbf{z}_{\mathcal{M}_c}}\big) \Big\} \geq \lambda \quad (7)$$

where λ is an expansion threshold that controls the expansion of the model. If the data statistics of the current memory $\mathcal{M}_c$ is largely different from the probabilistic representations of all previously trained components at $\mathcal{T}_c$ (satisfying Eq. (7)), then we freeze the current component S_t to preserve the information of the current memory while building a new component S_{t+1} to be trained next. In the next section, we introduce a new sample selection approach from the memory $\mathcal{M}_i$ to further increase the statistical diversity among the trained components.

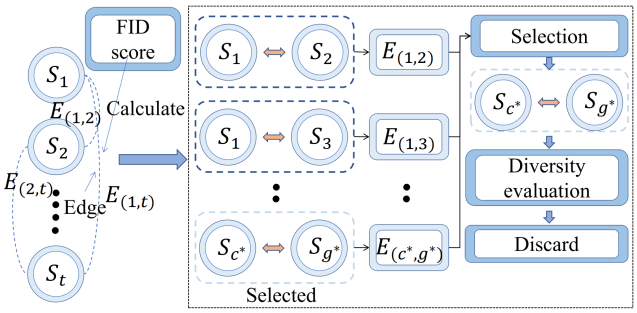

Figure 1: The proposed mixture component discarding mechanism.

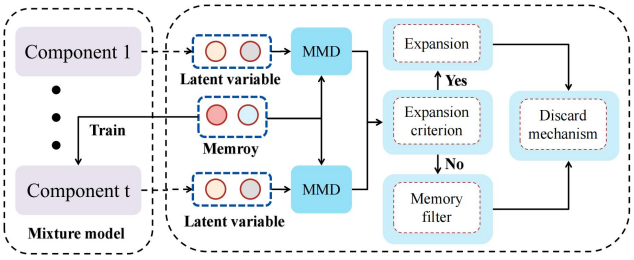

Figure 2: The overview of the proposed optimization framework that includes four steps.

Diversity-Aware Sample Selection

First, we consider a simple selection approach in which the earliest stored samples are always removed and the newly seen samples are continuously added, while the current memory is kept at a size no larger than $|\mathcal{M}_i|^{Max}$, which is the maximum memory size. The sample selection approach is akin to a Sliding Window (SW) recording memory :

$$\mathcal{M}_i = \mathcal{M}_{i-1} \bigcup \mathcal{D}_i \rightarrow \mathcal{M}_i = \bigcup_{j=b}^{|\mathcal{M}_i|^{Max}+b} \mathcal{M}_i[j]\,, \quad (8)$$

where $\mathcal{M}_i[j]$ represents the j-th stored samples of $\mathcal{M}_i$ and b is the batch size, considered as $b = 10$ in the experiments. However, according Eq. (8), $\mathcal{M}_i$ always stores only the recently given samples and thus would not record the previously learnt information. To solve this problem, we introduce a new sample selection approach that selectively stores the more statistically diverse samples with respect to the already learnt knowledge, stored in each model component.

Since VAE is a natural estimator of the sample log-likelihood, it can be used to detect the novelty of new samples (Rao et al. 2019). On this basis, we propose to sum up the negative VAE loss (negative sample log-likelihood) of all previously learned components as a discrepancy score for each stored sample, expressed as :

$$\mathcal{L}_d(\mathbf{x}_s) = \frac{1}{t-1} \sum\nolimits_{j=1}^{t-1} \Big\{ -\mathcal{L}_V(S_j, \mathbf{x}_s) \Big\}\,, \quad (9)$$

where $\mathbf{x}_s$ represents the s-th stored sample. If the discrepancy score (Eq. (9)) is low means that each trained component knows more about $\mathbf{x}_s$ and then we want to remove $\mathbf{x}_s$ from the current memory in order to promote the statistical diversity between the current memory and the existing trained components. Eq. (9) is used for the sample selection in a procedure called Diverse Sampling Selection (DivSS) :

$$\mathbf{x}'_s = \arg \min_{\mathbf{x}_s \in \mathcal{M}_i} \{\mathcal{L}_d(\mathbf{x}_s)\}\,, \quad (10)$$

then $\mathbf{x}'_s \notin \mathcal{M}_i$. We repeat recursively the selection and exclusion of $\mathbf{x}'_s$'es until $|\mathcal{M}_i| \leq |\mathcal{M}_i|^{Max}$.

The Component Discarding Mechanism

In this section, we develop a new component discarding mechanism that selectively removes the unnecessary components to reduce the overall size of the model while promoting the probabilistic diversity of generated data among the remaining components. The main idea behind the proposed discard mechanism is to identify the components which are characterized by overlapping probabilistic representations and remove those that are not necessary. However, searching for components directly in the DEM (Lee et al. 2020; Rao et al. 2019) would incur significant computational costs. In this paper, we propose to create a relationship graph matrix that describes the relevance between each pair of mixture components and we discard the components by analysing the relationships in this graph matrix.

The proposed component discarding mechanism is illustrated in Fig. 1, considering that we have already trained t components in the DEM mixture. First, we generate 1,000 samples using the VAE of each component, describing the previously learned knowledge, denoted as $\mathbb{P}_{\tilde{x}_j}$ for the j-th component. Then we form the relationship graph between these trained components using the FID score (Heusel et al. 2017) estimated from the generated samples and calculated between pairs of components. FID is appropriate for measuring the discrepancy between two empirical data distributions (Gulrajani et al. 2017) and thus can be used to measure the knowledge similarity between two components. Let $\mathbf{E} \in \mathbb{R}^{t \times t}$ represent a relationship matrix in which $E_{(c,g)}$ represents the edge value between S_c and S_g, estimated by $1/f_{\text{FID}}(\mathbb{P}_{\tilde{x}_c}, \mathbb{P}_{\tilde{x}_g})$, where $f_{\text{FID}}(\cdot,\cdot)$ is the FID estimator. Since the FID criterion is symmetric, $\mathbf{E}$ is a symmetric matrix. The selected components are decided by:

$$\begin{aligned} E^\star &= \max_{c,g=1,\cdots,t} E_{(c,g)}, \\ c^\star, g^\star &= \arg \max_{c,g=1,\cdots,t} E_{(c,g)}\,, \end{aligned} \quad (11)$$

where $c^\star$ and $g^\star$ are the indices of the selected components. We then remove one of the selected components from $\mathcal{G}$ according to the statistical diversity assessment. The component to be removed during the training is chosen according to $E^\star > \lambda_2$ from Eq. (11), where $\lambda_2 \in [0, 10]$ is a threshold. We also define a threshold $n \in [1, 20]$ to represent the remaining number of components after performing the discarding mechanism (See details in Appendix-A from SM[1]).

As shown in Fig. 1, after Eq. (11), we evaluate the diversity score for each of the selected component $S_{c^\star}$ or $S_{g^\star}$ using :

$$\mathcal{L}_{diversity}(S_a) = \frac{1}{t-1} \sum\nolimits_{j=1}^{t} \left\{ \frac{1}{E_{(j,a)}} \right\}\,,\ j \neq a\,, \quad (12)$$

Methods	Split MNIST	Split CIFAR10	Split CIFAR100
finetune*	19.75 ± 0.05	18.55 ± 0.34	3.53 ± 0.04
GEM*	93.25 ± 0.36	24.13 ± 2.46	11.12 ± 2.48
iCARL*	83.95 ± 0.21	37.32 ± 2.66	10.80 ± 0.37
reservoir*	92.16 ± 0.75	42.48 ± 3.04	19.57 ± 1.79
MIR*	93.20 ± 0.36	42.80 ± 2.22	20.00 ± 0.57
GSS*	92.47 ± 0.92	38.45 ± 1.41	13.10 ± 0.94
CoPE-CE*	91.77 ± 0.87	39.73 ± 2.26	18.33 ± 1.52
CoPE*	93.94 ± 0.20	48.92 ± 1.32	21.62 ± 0.69
ER + GMED†	82.67 ± 1.90	34.84 ± 2.20	20.93 ± 1.60
ER_a + GMED†	82.21 ± 2.90	47.47 ± 3.20	19.60 ± 1.50
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
DivSS + Discard	96.16 ± 0.11	50.12 ± 0.23	25.24 ± 0.17
DivSS	$\mathbf{96.95} \pm 0.13$	$\mathbf{53.71} \pm 0.19$	$\mathbf{26.03} \pm 0.16$
SW	96.81 ± 0.12	50.91 ± 0.25	25.65 ± 0.16
SW + Discard	95.93 ± 0.15	49.93 ± 0.23	24.18 ± 0.18

Table 1: Classification accuracy of five independent runs for various models on three datasets. * and † denote the results cited from (De Lange and Tuytelaars 2021) and (Jin et al. 2021), respectively.

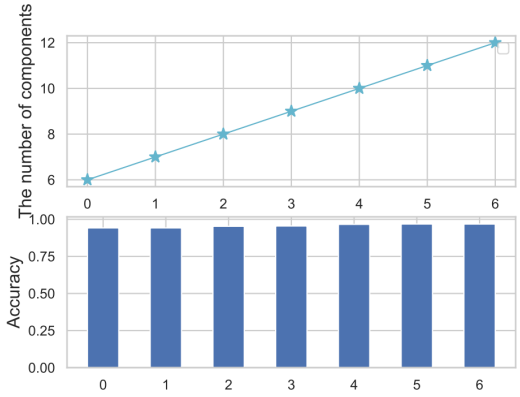

Figure 3: Assessing the performance of the model when changing the number of remaining components.

$$\mathcal{G} = \{S_j \in \mathcal{G} \mid j = 1, \cdots, t, j \neq c\}$$
$$c = \arg\min_{c^\star, g^\star} \left\{ \mathcal{L}_{diversity}(S_{c^\star}), \mathcal{L}_{diversity}(S_{g^\star}) \right\}. \quad (13)$$

The component with the minimum diversity score, according to Eq. (13) is discarded. To avoid reselection at the next discarding step, we also remove all edge values associated with the deleted component from $\mathbf{E}$. We can repeat Eq. (13) and Eq. (12) in order to discard more components from $\mathcal{G}$ (See Appendix-A from SM[1]).

The Unified Optimization Framework

In this section, we propose a unified optimisation framework for the mixture model that integrates the proposed expansion mechanism, sample selection approach and discard mechanism for training a dynamic expansion model. The overview of the proposed framework is shown in Fig. 2 and is summarized in four steps :

Methods	Split MNIST	Split CIFAR10	Split CIFAR100
DivSS + Discard	10	10	6
DivSS	29	31	9
SW	30	35	10
SW + Discard	10	10	6

Table 2: Number of components for the proposed model when learning various datasets.

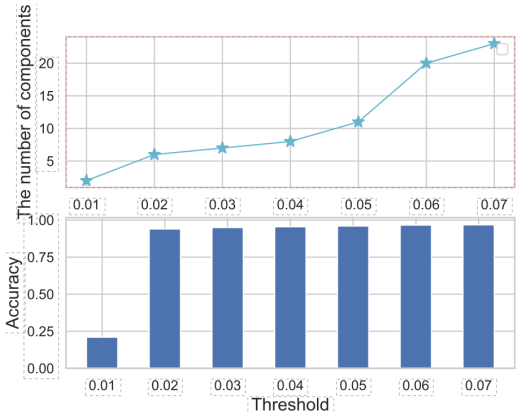

Figure 4: Assessing the performance of the model when changing λ_2.

Step 1 (The training of the current component.) We assume that a dynamic expansion model has trained t components. Then we train the current component S_t on $\mathcal{M}_i$ at $\mathcal{T}_i$ using $\mathcal{L}_C(S_t, \mathcal{M}_i)$ and $\mathcal{L}_V(S_t, \mathcal{M}_i)$ using (1) and (2).

Step 2 (Check the expansion of the model.) If the current memory $\mathcal{M}_i$ is full $|\mathcal{M}_i| \geq |\mathcal{M}_i|^{Max}$, then we check the expansion of the model using Eq. (7). If the expansion criterion is satisfied, we create a new component S_{t+1} for $\mathcal{G}$ and clean up $\mathcal{M}_i$ in order to learn statistically non-overlapping samples, otherwise, we perform **Step 3**.

Step 3 (Sample selection.) If $|\mathcal{M}_i| \geq |\mathcal{M}_i|^{Max}$, then we estimate the sample log-likelihood for each stored sample using Eq. (9). We perform the sample selection for the current memory $\mathcal{M}_i$ using Eq. (10).

Step 4 (Discarding the components.) Once all the training steps are completed, we repeatedly perform the component discarding procedure to remove the unnecessary components from the dynamic expansion model $\mathcal{G}$.

Experiments

Datasets : Split MNIST divides MNIST (LeCun et al. 1998) containing 60k training samples, into five tasks according to pairs of digits in increasing order (De Lange and Tuytelaars 2021). **Split CIFAR10** splits CIFAR10 (Krizhevsky and Hinton 2009) into five tasks where each task consists images from two different classes (De Lange and Tuytelaars 2021). **Split CIFAR100** divides CIFAR100 into 20 tasks where each task has 2500 samples from 5 different classes (Lopez-Paz and Ranzato 2017).

Methods	Split MImageNet
ER$_a$	25.92 ± 1.2
ER + GMED	27.27 ± 1.8
MIR+GMED	26.50 ± 1.3
MIR	25.21 ± 2.2
DivSS	**29.63** ± 1.5
DivSS + Discard	27.58 ± 2.7

Table 3: Classification accuracy for 20 runs when testing various models on Split MImageNet.

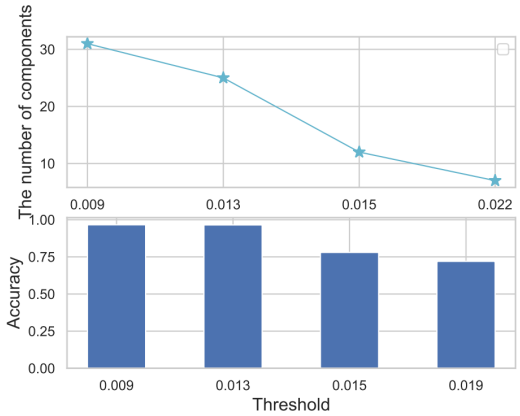

Figure 5: The number of components and the classification accuracy of the proposed model on Split MNIST when changing λ.

Methods	Split MNIST	Split CIFAR10	Split MImageNet
Vanilla	21.53 ± 0.1	20.69 ± 2.4	3.05 ± 0.6
ER	79.74 ± 4.0	37.15 ± 1.6	26.47 ± 2.3
MIR	84.80 ± 1.9	38.70 ± 1.7	25.83 ± 1.5
ER + GMED	82.73 ± 2.6	40.57 ± 1.7	28.20 ± 0.6
MIR+GMED	86.17 ± 1.7	41.22 ± 1.1	26.86 ± 0.7
DivSS + Discard	95.80 ± 2.1	43.57 ± 1.6	28.27 ± 1.5
DivSS	**96.51** ± 1.8	**44.23** ± 1.4	**29.35** ± 1.2

Table 4: Classification accuracy of five independent runs for various models over streams with fuzzy task boundaries.

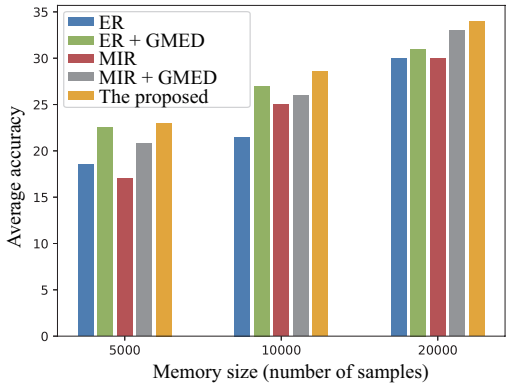

Figure 6: The performance change of various models on Split MINI-ImageNet when changing the memory size. More results are reported in Appendix-C.1 from SM[1].

Network architecture and hyperparameters : In the experiments we adapt the setting from (De Lange and Tuytelaars 2021), where ResNet-18 (He et al. 2016) is used as the classifier for Split CIFAR10 and Split CIFAR100. We also use an MLP network with 2 hidden layers of 400 units (De Lange and Tuytelaars 2021) as the classifier for Split MNIST. The maximum memory size for Split MNIST, Split CIFAR10, and Split CIFAR100 is 2000, 1000 and 5000, respectively. In each training step $\mathcal{T}_i$, we only access a small batch ($b = 10$) of training samples. The number of epochs for each training step/time is 10. The model expansion threshold λ from Eq. (7) for Split MNIST, Split CIFAR10, Split CIFAR100, and Split MImageNet is 0.009, 0.04, 0.03 and 0.055, respectively.

Baselines : We train a DEM with the MMD-based expansion mechanism, the sample selection approach (DivSS) and the component discarding mechanism, called "DivSS + Discard". We also replace DivSS by a Sliding Window (SW) and the resulting approach is called "SW + Discard". In this paper, we mainly compare with the two most popular dynamic expansion models, CURL (Rao et al. 2019) and CN-DPM (Lee et al. 2020). Additionally, we also compare with a series of recent TFCL approaches, including : fine-tune that trains a classifier on the data stream, GSS (Aljundi et al. 2019b), Dynamic-Online Cooperative Memorization (OCM) (Ye and Bors 2022a), MIR (Aljundi et al. 2019a), Gradient Episodic Memory (GEM) (Lopez-Paz and Ranzato 2017), Reservoir (Vitter 1985), Incremental Classifier and Representation Learning (iCARL) (Rebuffi et al. 2017), CoPE, ER + GMED and ER$_a$ + GMED (Jin et al. 2021) where ER is the Experience Replay (Rolnick et al. 2019) and ER$_a$ is ER with data augmentation.

Classification Task

In this section, we evaluate the performance of the proposed approach in classification and we adapt the standard TFCL experiment setting of (De Lange and Tuytelaars 2021). We report the classification accuracy of Split MNIST, Split CIFAR10 and Split CIFAR100 in Table 1, where "DivSS" means that the proposed approach does not use the component discarding mechanism. The number of components of the proposed model is provided in Table 2. The classification accuracy shows that the proposed "DivSS" outperforms the other baselines in the three datasets. Moreover, the proposed mixture component discarding mechanism can significantly compress the trained model without much performance loss, according to the results from Table 1.

We also investigate the effectiveness of the proposed approach in learning a large dataset, such as MINI-ImageNet (Le and Yang 2015). The Split MINI-ImageNet (S-MImageNet) contains 20 disjoint tasks, where each task consists of learning the images of five classes (Aljundi et al. 2019a). We adapt the settings of (Aljundi et al. 2019a), with a maximum memory size of $10K$ using a lean version of ResNet-18 (He et al. 2016) as the classifier. The classification accuracy for Split MImageNet is given in Table 3, where

Methods	Split MNIST-SVHN Accuracy	No	Parameters	Speed (seconds)
DivSS	**65.29**	18	175M	0.0064
DivSS + Discard	63.78	13	126M	0.0048
DivSS + Pruning	60.63	18	131M	0.0052
DivSS + KD	58.82	13	126M	0.0048
CNDPM	60.26	31	301M	0.0104
Methods	**Split MNIST-CIFAR10**			
DivSS	**64.36**	16	155M	0.0050
DivSS + Discard	62.15	11	106M	0.0032
DivSS + Pruning	58.97	16	112M	0.0041
DivSS + KD	56.64	11	106M	0.0032
CNDPM	59.92	27	262M	0.0106

Table 5: Classification accuracy under Split MNIST-SVHN and Split MNIST-CIFAR10

we compare with several state-of-the-art methods, quoting the results from (Jin et al. 2021). The number of components in the proposed model is 10 and we keep 8 components while removing two components for "DivSS + Discard" using the proposed component discarding mechanism. These results show that the proposed approach performs better than other baselines under challenging learning conditions.

Results on Fuzzy Task Boundaries

In a more realistic learning environment, a model is provided with a data stream with fuzzy task boundaries (Lee et al. 2020). To investigate the performance of the proposed approach under these conditions, we swap randomly samples between two tasks as in the study from (Lee et al. 2020). We train the proposed approach under Split MNIST, Split CIFAR10 and Split MImageNet under this setting of corrupting the data and the results are shown in Table 4. In Appendix-C4 from SM[1] we also give the number of components for the proposed approach under fuzzy task boundaries. These results demonstrate that the proposed approach is robust and provides better performance under this setting when compared to other approaches.

Ablation Study

We perform a full ablation study to evaluate the effectiveness of each component used in our approach. Additional ablation results are provided in Appendix-C from SM[1].

Expansion mechanism : First, we study the mixture expansion process of the proposed approach when we change the threshold λ in Eq. (7). We train the proposed approach under Split MNIST and plot the number of components and classification accuracy in Fig. 5. A large λ encourages the model to use fewer components, resulting in poorer performance. As λ decreases, the model improves its performance while also adding more components to the mixture.

Discarding mechanism : We investigate the performance of the proposed approach when we discard some redundant components. We train the proposed "DivSS" under Split MNIST and use the proposed component discarding mechanism to remove unnecessary components. We present the results in Fig. 3, where we can observe that the performance of the proposed approach does not degrade too much even when considering fewer components, such as 6 in this case. Moreover, we use λ_2 for $E^\star < \lambda_2$ from Eq. (11), to select the components to be removed during the training (see details in Appendix-B of SM[1]). We plot the results with different λ_2 in Fig. 4. When increasing λ_2, the model tends to discard fewer components. If λ_2 is very small, the model is left with only two components, resulting in degenerating performance.

The size of memory buffer : We train the proposed approach "DivSS + Discard" under Split MNIST, Split CIFAR10, Split CIFAR100 and Split MImageNet with different buffer memory sizes. The average accuracy for each dataset is provided in Fig. 6, where we compare with MIR + GMED and ER + GMED. These results show that reducing the memory buffer size leads to a degenerated performance for all methods, but the proposed approach performs better than all other baselines with different memory configurations.

Model complexity and processing time : We create Split MNIST-SVHN which consists of Split MNIST and Split SVHN and similarly Split MNIST-CIFAR10. The results on Split MNIST-SVHN and Split MNIST-CIFAR10 are provided in Table 5, where "No" and "Parameters" are the number of components and parameters. "Speed" denotes the testing time for each testing sample. The proposed model outperforms CNDPM by using fewer parameters and requiring less testing time under TFCL. These results demonstrate that CNDPM ends up with statistically overlapping experts, which does not improve the performance while requiring additional computational costs.

Compressing methods : We compare with other compressing methods such as Pruning (Zhu and Gupta 2017) and Knowledge Distillation (KD) (Ye and Bors 2022e). Since Pruning is not designed for discarding components, we apply it to reduce the number of parameters for each component. We also apply KD to compress many components into one by using the KD process (Ye and Bors 2022e). In Table 5, where "No", "Parameters", and "Speed" denotes the number of components, we provide the number of parameters and the required inference time for each sample. These results show that the proposed compression method achieves the best performance while employing fewer parameters.

Conclusion

In this paper, we introduce a novel DEM model enabled by the MMD-based expansion mechanism for TFCL. A diversity-aware sample selection approach is proposed to promote knowledge diversity among components, which can further improve performance. Then, we propose a novel component discarding mechanism to reduce a significant number of parameters without sacrificing much performance for the proposed model. The proposed discarding mechanism can be performed during or after the training process, which provides a flexible learning manner. We perform a series of TFCL experiments, and the empirical results demonstrate that the proposed approach achieves best performance than other baselines.

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.; Caccia, L.; Belilovsky, E.; Caccia, M.; Lin, M.; Charlin, L.; and Tuytelaars, T. 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), arXiv preprint arXiv:1903.08671*.

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.

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

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

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

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.

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.

Gu, Y.; Yang, X.; Wei, K.; and Deng, C. 2022. Not Just Selection, but Exploration: Online Class-Incremental Continual Learning via Dual View Consistency. In *Proc. of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 7442–7451.

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

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

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.

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

Hung, C.-Y.; Tu, C.-H.; Wu, C.-E.; Chen, C.-H.; Chan, Y.-M.; and Chen, C.-S. 2019. Compacting, Picking and Growing for Unforgetting Continual Learning. In *Advances in Neural Information Processing Systems*, 13647–13657.

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)*, 29193–29205.

Jung, H.; Ju, J.; Jung, M.; and Kim, J. 2018. Less-forgetting learning in deep neural networks. In *Proc. AAAI Conf. on Artificial Intelligence*, volume 32, 3358–3365.

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.

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)*.

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.

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

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.

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.

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

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 *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2001–2010.

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

Ritter, H.; Botev, A.; and Barber, D. 2018. Online Structured Laplace Approximations for Overcoming Catastrophic Forgetting. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 31, 3742–3752.

Rolnick, D.; Ahuja, A.; Schwarz, J.; Lillicrap, T. P.; and Wayne, G. 2019. Experience Replay for Continual Learning. In *Advances in Neural Information Processing Systems 34 (NeurIPS)*, 348–358.

Rusu, A. A.; Rabinowitz, N. C.; Desjardins, G.; Soyer, H.; Kirkpatrick, J.; Kavukcuoglu, K.; Pascanu, R.; and Hadsell, R. 2016. Progressive neural networks. *arXiv preprint arXiv:1606.04671*.

Shin, H.; Lee, J. K.; Kim, J.; and Kim, J. 2017. Continual learning with deep generative replay. In *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.

Tolstikhin, I. O.; Sriperumbudur, B. K.; and Schölkopf, B. 2016. Minimax estimation of maximum mean discrepancy with radial kernels. *Advances in Neural Information Processing Systems*, 29: 1930–1938.

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

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*.

Xiao, T.; Zhang, J.; Yang, K.; Peng, Y.; and Zhang, Z. 2014. Error-driven incremental learning in deep convolutional neural network for large-scale image classification. In *Proc. of ACM Int. Conf. on Multimedia*, 177–186.

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. 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. Lifelong learning of interpretable image representations. In *Proc. Int. Conf. on Image Processing Theory, Tools and Applications (IPTA)*, 1–6.

Ye, F.; and Bors, A. G. 2020c. Mixtures of variational autoencoders. In *Proc. Int. Conf. on Image Processing Theory, Tools and Applications (IPTA)*, 1–6.

Ye, F.; and Bors, A. G. 2021a. 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. 2021b. Lifelong Twin Generative Adversarial Networks. In *Proc. IEEE Int. Conf. on Image Processing (ICIP)*, 1289–1293.

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. Dynamic Self-Supervised Teacher-Student Network Learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(10): 6280–6296.

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

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

Ye, F.; and Bors, A. G. 2022f. Task-Free Continual Learning via Online Discrepancy Distance Learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, 1–16.

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. of the IEEE/CVF Int. Conf. on Computer Vision (ICCV)*, 2759–2768.

Zhu, M.; and Gupta, S. 2017. To prune, or not to prune: exploring the efficacy of pruning for model compression. *arXiv preprint arXiv:1710.01878*.