# Training a Dynamic Growing Mixture Model for Lifelong Learning

Fei Ye[1] and Adrian G. Bors[2] , *IEEE Senior Member*

***Abstract*—Lifelong learning defines a training paradigm that aims to continuously acquire and capture new concepts from a sequence of tasks without forgetting. Recently, Dynamic Expansion Models (DEM) have been proposed to address catastrophic forgetting under the lifelong learning paradigm. However, the efficiency of dynamic expansion models lacks a thorough explanation based on theoretical analysis. In this paper, we develop a new theoretical framework that interprets the forgetting process of the DEM as increasing the statistical discrepancy distance between the distribution of the probabilistic representation of the new data and the previously learnt knowledge. This analysis provides new insights into model's forgetting behavior. The theoretical analysis shows that adding new components to a mixture model represents a trade-off between model complexity and its performance. Inspired by the theoretical analysis, we introduce a new dynamic expansion model, called the Growing Mixture Model (GMM), where generative data components are added according to the novelty of the incoming task information compared to what is already known. A new component selection mechanism considering the model's already acquired knowledge is employed for updating new DEM's components, promoting efficient future task learning. We also train a compact Student model with samples drawn through the generative mechanisms of the GMM, aiming to accumulate cross-domain representations over time. By employing the Student model we can significantly reduce the number of parameters and make quick inferences during the testing phase.**

***Index Terms*—Lifelong generative modelling, Continual learning, Lifelong learning, Dynamic expansion model.**

## I. Introduction

Lifelong learning (LLL) describes a learning paradigm for training a model to successively learn a series of tasks without forgetting the information associated with any of them. Artificial intelligence models have been successfully used in a wide range of applications, such as image processing [1], object recognition [2], image translation [3] or image synthesis [4], [5] among others. However, existing deep learning models perform well when accessing all training data during a single learning stage, while they would fail after learning multiple tasks in succession. The collapse in performance in such cases is due to catastrophic forgetting [6].

A popular approach to relieve forgetting is the Generative Replay Mechanism (GRM) [7], which can be implemented by either a Generative Adversarial Net (GAN) [8] or a Variational Autoencoder (VAE) [9]. A generative model aims to provide pseudo-data through successive GRMs which are mixed with the data corresponding to a new task, forming a new training data set used for learning the model in order to avoid forgetting [10], [11]. GRM-based models have shown good results in continual classification tasks but they gradually lose their performance when learning many different tasks. Additionally, mode collapse [12] represents another drawback for GRMs, especially when these are repeatedly used for learning distinct data domains. Combining the dynamic model expansion and GRM into a unified optimization framework was proposed in [13], ensuring that model's capacity is progressively increased to adapt to significant changes in the data. Nevertheless, this approach still suffers from forgetting due to the degradation following repetitions of GRM processes. Other attempts have been focused on dynamic expansion mechanisms [14] or on using ensemble structures [15], in which each component is added on top of a joint network backbone. These approaches do not have to update the previously learnt network parameters when learning new data and thus can avoid forgetting [16]. However, such methods do not rely on theoretical guarantees, and the forgetting behavior behind these methods is not well understood.

This research study first proposes a novel theoretical framework for assessing the forgetting in GRM-based models, inspired by the domain adaptation theory [17]. Consequently, we derive a risk bound for assessing the generalization performance on the target domain achieved by a model trained on a source domain. Unlike the study from [17] which considers that the source domain is static and does not change over time, the theoretical framework from this paper evaluates the performance loss of a model trained on dynamically changing source domains over time. Moreover, we extend this framework in order to analyze the forgetting behavior for other continual learning models, including dynamic expansion and memory-based models.

Generative modeling is mainly used in unsupervised machine learning as a means to describe underlying structures in data. Generative artificial intelligence systems have been widely adopted in many applications, including for 3D object detection [18], image editing [19] and medical image segmentation [20]. Enabling generative models in continual learning without forgetting can lead to developing many new practical applications dealing with wide data variations. The core issue addressed in our paper is network forgetting when implementing image generative modeling in continual learning. Generative model forgetting, unlike forgetting in classification systems, has not been sufficiently explored before.

Most current AI methods, such as the Generative Replay Mechanism (GRM) [7], the memory replay [21] and the

[1]School of Information and Software Engineering, University of Electronic Science and Technology of China, Chengdu, E-mail: feiye@uestc.edu.cn

[2]Department of Computer Science, University of York, York YO10 5GH, UK, E-mail: adrian.bors@york.ac.uk

Dynamic Expansion Model (DEM) [22], [23] were shown to address catastrophic forgetting challenges. However, the GRM and memory replay methods are not scalable for learning a growing or infinite number of tasks. Although DEMs can deal with long sequences of tasks, most such models are primarily designed for classification tasks [22], [23], and are not suitable for image generative modeling. Therefore, we introduce a novel dynamic expansion framework, which can dynamically create new generative components while learning novel information. A new knowledge evaluation approach is proposed for assessing knowledge similarity between each previously learnt component and the data distribution corresponding to the incoming task. The proposed knowledge measure approach chooses a suitable component for learning a related task that shares visual concepts consistent with the selected component, accelerating future task learning. Furthermore, we introduce a new Knowledge Distillation (KD) based method for a Teacher-Student architecture, transferring the information learnt by each component into a lightweight Student module. Thus the entire learned knowledge is compressed into a unified latent space, accelerating the inference process and reducing the size of the model.

The proposed GMM framework has two primary advantages compared to current dynamic expansion models : (1) The proposed GMM framework, besides classification tasks also addresses image generation in continual learning; (2) Introduces a novel knowledge distillation approach that can transfer the knowledge learnt by a complex teacher model to a fixed-size student module, which can be deployed on resource-constrained devices; (3) The proposed GMM framework can effectively address a long sequence of tasks by selectively creating new experts when similar tasks reappear. We provide the code in https://github.com/dtuzi123/LifelongGMM.

This paper brings the following contributions :

- We analyze the forgetting process of continual learning models by formulating the model's risk from the learning and forgetting perspectives.
- We propose a dynamically expanding network architecture according to the novelty of the incoming tasks while reusing existing components whenever learning from data deemed similar to those acquired in the past.
- We propose a new knowledge distillation (KD) approach enabling a lightweight Student module to learn the information accumulated by the GMM together with that from new tasks. Furthermore, the Student module can capture cross-domain latent representations over time while enjoying fast inference during the testing phase.
- Extensive experiments show that the proposed GMM provides better performances than other methods in both continual generative modeling and classification tasks.

In Section II we review the related works. The theoretical framework is provided in Section III and the proposed methodology in Section IV. A series of experiments and ablation studies are provided in Section V, while the conclusions are drawn in Section VI.

## II. Related works

In the following we outline the main lifelong learning research directions : regularization, dynamic architecture and replay-based models.

### A. Regularization-based approaches

Regularization methods alleviate catastrophic forgetting by introducing an auxiliary term in the objective function, which penalizes changes in the network's weights when learning a new task [24], [25], [26]. Learning Without Forgetting (LwF) [27] is one of the most popular regularization approaches, which uses knowledge distillation to enforce the newly trained network for remembering previously learnt knowledge. Empirical results have shown that the Elastic Weight Consolidation (EWC) [28] is good at modeling images of random pixel permutations, while it performs worse when incrementally learning new data categories. However, a disadvantage of EWC is that it requires growing computational resources when learning many tasks. Such a problem was solved in [29], which considers model updating only for the latest learnt tasks, while EWC updates the corresponding Fisher matrix using the moving average [30]. Bayesian inference addressed forgetting for both continual learning of classification and generation tasks in the Variational Continual Learning (VCL) [31]. Works that regulate the learned representations were found to be robust to forgetting by using adversarial training [16] or meta-learning [32] in continual learning. However, these approaches are computationally intensive, given that they rely on inner iterative optimizations, especially when learning a growing number of tasks [33].

Regularization can be implemented using either knowledge distillation [27] or by adding penalty terms in the main objective function [31]. Specifically, knowledge distillation defines a loss for the teacher and its corresponding student module, that aligns the output patterns between them. Such an approach is related to the proposed GMM framework since we also implement a teacher-student framework which has associated a knowledge distillation loss function. Different from existing lifelong knowledge distillation models such as that from [27], which uses the same network architecture for the teacher and student module, the proposed knowledge distillation approach can transfer the knowledge learnt by a complex teacher module to a lightweight student module, reducing the storage requirements while accelerating the inference process.

### B. Memory replay

Memory replay approaches would either use a generative replay network or a memory buffer [21], [34], [35] as a replay mechanism which reproduces the previously learnt information through data generations when learning novel tasks. A typical memory replay approach [7] consists of a hybrid framework made up of a deep generative model and a classifier. The generator in this framework aims to replay the pseudo-data which are then merged with the newly given samples to be learnt, forming a joint dataset. This new dataset is then used for the lifelong training of the model. However, this

approach is only used for classification and is not able to learn new informative data representations that would benefit other downstream tasks [36]. Memory reply lifelong learning models would either use a Variational Autoencoder (VAE) [9], or a Generative Adversarial Network (GAN) [8] as the Generative Replay Mechanism (GRM). Achille *et al.* [37] proposed a new continual learning approach based on the VAE framework, aiming to capture meaningful representations across different data domains over time. The Minimum Description Length (MDL) principle is used in the Variational Assorted Surprise Exploration (VASE) [38] for encouraging learning disentangled representations. However, VAE-based generations often result in blurred images [1], [39] leading to degenerate performances when learning several datasets characterized by complex information. This problem is addressed by using better generative replay networks, such as GANs [36], [40]. However, also GANs suffer from mode collapse after learning several entirely different data domains [12].

Generative models used as generative replay networks are related to the proposed GMM framework. However, unlike the existing generative memory methods such as VASE [38] for example, which trains a unique generative model and thus is unable to address a long sequence of tasks, the proposed GMM framework explores and combines advantages from both GRM and dynamic expansion methods into a unified framework, achieving better performance when learning a long sequence of tasks.

### C. Lifelong generative modeling

Different from continual supervised learning, which usually learns classifiers to implement classification tasks, lifelong generative modeling aims to train a deep generative model capable of continually capturing underlying data structures and producing data reconstructions and generations [10], [41]. A new objective function was proposed in [37] for implementing lifelong generative modeling, based on the VAE [9], while learning disentangled representations for various tasks/domains without forgetting. Ramapuram *et al.* [10] introduces the Lifelong Generative Modeling (LGM) as a Teacher-Student continual learning model, where both the Teacher and Student are implemented by VAEs. During the training, LGM exchanges information between the Teacher and Student using successively GRMs. However, a VAE model usually produces lower quality generative replay samples that degenerates model's performance in continual learning. This issue was relieved by employing Generative Adversarial Networks (GANs) [11] as a generative replay network. A GAN can generate high-quality image generation outputs and thus the model trained on such pseudo-data can significantly relieve network forgetting. Moreover, a VAE has been combined with a GAN into a unified framework in the Lifelong VAEGAN [36], which is able to perform semi-supervised, supervised, unsupervised learning and disentangled representation learning. A dynamic expansion GAN mixture model was used as a Teacher, automatically adding new GAN-based experts when learning entirely different tasks, was explored in [42].

Compared to existing static models for lifelong generative modeling, the proposed GMM framework is able to learn a long sequence of tasks without forgetting. In addition, compared to other DEMs for lifelong generative modeling, our method can learn relevant and new data representations from a series of tasks, benefiting image reconstruction and interpolation.

### D. Dynamic expansion and ensemble models

Dynamic expansion models grow their capacity by increasing the number of parameters or by adding layers of hidden units for learning an increasing number of tasks [42], [43], [44]. Dirichlet processes have been used for expanding mixture models in the context of continual learning in [25], [26], [45]. Meanwhile, small memory buffers have been used for storing past relevant data, thus avoiding expanding frequently the network architecture in [14], while the Continual Unsupervised Representation Learning (CURL) [13] alternates dynamic model expansions with GRMs to avoid forgetting during continual learning. CURL expands only the architecture for the inference component while the decoder is continually updated based on the generative replay samples and novel data as well. Other methods, such as those from [44], [46], consider a shared module, updated only when learning the first task while afterwards is used as the backbone for other newly created tasks-specific modules. Feature BoOSTing and ComprEssion for class-incremental learning (FOSTER) [22] is an advanced dynamic expansion model, which adds new modules to fit the residuals between the target and the output of the original model which introduces a new distillation approach to remove redundant parameters. The Memory-efficient Expandable MOdel (MEMO) [23] was shown to achieve good generalization performance while maintaining a compact network backbone during continual learning. Recently, Large Language Models (LLM) have been explored in continual learning [47], where transformer block-based experts are continually added to mixture frameworks whenever learning new tasks.

This paper considers that the task label is given, but we do not know the total number of tasks, so we assume that an endless succession of tasks is provided for learning. Moreover, existing lifelong learning methods have not been analyzed theoretically so far, which inspires us to propose a theoretical framework for analyzing forgetting processes of continual learning models. Furthermore, compared to most existing dynamic expansion models [22], [23], which are specifically designed for learning classification tasks, the proposed GMM framework can support a variety of applications in continual learning, including image reconstruction, generation, interpolation and image-to-image translation among others.

## III. Theoretical analysis framework

This section introduces the theoretical framework for analyzing the forgetting behavior of the GRM and how this can be used for enabling the dynamic expansion mechanism under the lifelong learning.

### A. Preliminaries

**Learning setting.** This paper mainly focuses on a popular lifelong learning setting where task and boundaries information are always provided during the training. Let $\mathcal{X}$ and $\mathcal{Z}$ denote the input and feature space, respectively. Let $\mathcal{T} = \{\mathcal{T}_1, \cdots, \mathcal{T}_N\}$ denote a series of tasks, where each is associated with a training set $D_S^i = \{(\mathbf{x}_i, y_i)\}_{i=1}^{N_S^i}$, and a testing set $D_T^i = \{(\mathbf{x}_i, y_i)\}_{i=1}^{N_T^i}$. $\mathbf{x}_i$ and $y_i$ are an observed variable and its target variable, representing the class label. $N_S^i$ and $N_T^i$ represent the total number of samples for $D_S^i$ and $D_T^i$, respectively. The lifelong learning goal is to train a model that only accesses samples from the associated training set $D_S^i$ during the $i$-th task learning, while it cannot access the data associated with any of the previously learnt tasks $\{\mathcal{T}_1, \cdots, \mathcal{T}_{i-1}\}$. Once the learning of all tasks is completed, we evaluate the trained model on all testing sets $\{D_T^1, \cdots, D_T^N\}$.

*Definition 1:* (*Model.*) Let $\mathcal{M}(\theta, \varsigma, \varphi)$ be a single model that consists of a classifier $h_\varsigma \colon \mathcal{X} \to \mathcal{Y}$ and a generator $g_\theta \colon \mathcal{Z} \to \mathcal{X}$, where $\mathcal{Y}$ represents the space of label predictions, represented by $\{1, 2, \ldots, n'\}$, $n' > 2$ for multi-class classification and $\{-1, 1\}$ for binary classification, where $\theta$, $\varsigma$ , $\varphi$, represent the model components' parameters. For inferring the task label for a given input, we introduce a task-inference network $U_\varphi \colon \mathcal{X} \to \mathcal{T}$ in $\mathcal{M}$, where $\mathcal{T}$ is the task domain.

*Definition 2:* (*Generative replay processes*). Generative replay processes represent an approach for successfully relieving forgetting when training a single model under multiple tasks [7]. We suppose that a single model $\mathcal{M}(\theta^t, \varsigma^t, \varphi^t)$, where $\theta^t$, $\varsigma^t$, $\varphi^t$, represent model components' parameters at time $t$, was trained on a sequence of training sets $\{D_S^1, \ldots, D_S^t\}$ in which the generator and classifier are represented by $g_{\theta^t}$ and $h_{\varsigma^t}$, respectively. When learning a new task ($\mathcal{T}_{t+1}$) using the model $\mathcal{M}$, the GRM first generates a set of pseudo-paired samples through the following process :

$$\{\mathbf{X}', \mathbf{Y}'\} = \{\mathbf{x}'_j \sim \mathbb{P}_{\theta^t}, y'_j = h_{\varsigma^t}(\mathbf{x}_j) \,|\, j = 1, \ldots, n\}, \quad (1)$$

where $n$ is the number of pseudo-paired samples. $\mathbb{P}_{\theta^t}$ is the distribution of generative replay samples produced by $g_{\theta^t}$. Then we incorporate the pseudo set $\{\mathbf{X}', \mathbf{Y}'\}$ with the new training set $D_S^{t+1}$ in the $\mathcal{T}_{t+1}$-th task learning. Let $\tilde{S}^t$ represent the distribution of the pseudo set $\{\mathbf{X}', \mathbf{Y}'\}$. We assume that $U_\varphi$ in $\mathcal{M}(\theta^t, \varsigma^t, \varphi^t)$ is an optimal task-inference function which returns the exact task label for an input $\mathbf{x}'_j$. Then we can define the distribution $\tilde{S}_i^{(t-i)}$ of the pseudo set $\{\{\mathbf{x}'_j, y'_j\} \,|\, j = 1, \ldots, n\}$, where each $\mathbf{x}'_j$ satisfies $U_\varphi(\mathbf{x}'_j) = i$. The superscript $(t-i)$ in $\tilde{S}_i^{(t-i)}$ denotes the number of GRM processes required for the $(i)$-th task after learning the $(t)$-th task. For instance, when the model is trained on the second task ($t = 2$), the number of GRM processes for learning the first task ($i = 1$) is $t - i = 1$. Therefore, we can have several distributions $\{\tilde{S}_i^0, \tilde{S}_i^1, \ldots, \tilde{S}_i^{t-i}\}$, where $\tilde{S}_i^0 = \tilde{S}_i$ represents the distribution of the training set $D_S^i$. When the number of tasks grows ($t$ is increased), the distribution $\tilde{S}_i^{t-i}$ gradually becomes more different from $\tilde{S}_i^0$ because repeated GRM processes lead to gradually forgetting the information about the original $(i)$-th task. We also consider $\tilde{S}_{i,\mathcal{X}}^{(t-i)}$ for representing the marginal distribution of $\tilde{S}_i^{(t-i)}$.

*Definition 3: (Data distribution).* We define $S_i$ as the statistical representation for the testing dataset $D_T^i$ from the $i$-th task. Let $S_{i,\mathcal{X}}$ represent the marginal distribution for $S_i$ along $\mathcal{X}$.

*Assumption 1:* We consider that $\mathcal{L} \colon \mathcal{Y} \times \mathcal{Y} \to [0, 1]$ is symmetric and bounded, satisfying $\forall (y, y') \in \mathcal{Y}^2, \mathcal{L}(y, y') \leq Q'$, where $Q'$ is a positive number. We also assume that $\mathcal{L}(\cdot, \cdot)$ is fulfilling the triangle inequality.

*Definition 4: (Discrepancy distance [17]).* Let $S_i$ and $\tilde{S}_i^{(t-i)}$ be two distributions over the space $\mathcal{X} \times \mathcal{Y}$. Let $\mathcal{L} : \mathcal{Y} \times \mathcal{Y} \to [0, 1]$ be a loss function that satisfies Assumption 1. Let $h, h' \in \mathcal{H}$ represent two classifiers, where $\mathcal{H}$ is the space of all classifiers. The discrepancy distance $\mathcal{L}_d$ between the two marginals $\tilde{S}_{i,\mathcal{X}}^{(t-i)}$ and $S_{i,\mathcal{X}}$ is defined as :

$$\mathcal{L}_d\left(\tilde{S}_{i,X}^{(t-i)}, S_{i,X}\right) = \sup_{(h,h') \in \mathcal{H}^2} \Big| \mathop{\mathbb{E}}_{\tilde{S}_{i,\mathcal{X}}^{(t-i)}} \left[\mathcal{L}\left(h'\left(\mathbf{x}\right), h\left(\mathbf{x}\right)\right)\right] - \mathop{\mathbb{E}}_{S_{i,\mathcal{X}}} \left[\mathcal{L}\left(h'\left(\mathbf{x}\right), h(\mathbf{x})\right)\right] \Big|. \quad (2)$$

*Definition 5:* (*Measuring the loss for a given distribution [17].*) For a given loss function $\mathcal{L} \colon \mathcal{Y} \times \mathcal{Y} \to [0, 1]$ and a joint distribution $S_i$, we can measure the loss between two classifiers $\{h, h'\}$ using samples drawn from $S_i$ as :

$$\mathcal{L}_{S_i}\left(h, h'\right) = \mathbb{E}_{\mathbf{x} \sim S_i}[\mathcal{L}(h(\mathbf{x}), h'(\mathbf{x}))]\,. \quad (3)$$

### B. Risk bounds for a model with a single component

The discrepancy distance, defined in Eq. (2), has been successfully used for deriving generalization bounds in various domain adaptation applications [17], [48], [49], as well as for matching real and generated data distributions when training GANs [50], [51], [52]. The main idea explored in this section is that of using the risk bound for analyzing the forgetting behavior, by defining the gap between the target risk and the source risk evaluated by the model as a forgetting process. As the model is trained on the dynamically changing source data distributions, its generalization performance on the target data is found to rely on the discrepancy distance between the source and target domains. Firstly, we derive the risk bound for analyzing the forgetting behavior of a model with a single component, trained on a certain task.

*Theorem 1:* Let $S_i$ and $\tilde{S}_i^{(t-i)}$ be two joint distributions over the space $\mathcal{X} \times \mathcal{Y}$. We define $h_{S_i} = \arg\min_{h \in \mathcal{H}} \mathcal{L}_{S_i}(h, f_{S_i})$ and $h_{\tilde{S}_i^{(t-i)}} = \arg\min_{h \in \mathcal{H}} \mathcal{L}_{\tilde{S}_i^{(t-i)}}(h, f_{\tilde{S}_i^{(t-i)}})$ as the optimal classifiers for $S_i$ and $\tilde{S}_i^{(t-i)}$, respectively, where $f_{S_i}$ and $f_{\tilde{S}_i^{(t-i)}}$ are the target functions for $S_i$ and $\tilde{S}_i^{(t-i)}$, respectively. By satisfying Definition 1 and considering $S_i$ and $\tilde{S}_i^{(t-i)}$ be the source and target distributions, we derive a risk bound according to the domain adaptation theory [17], as :

$$\begin{aligned} \mathcal{L}_{S_i}\left(h, f_{S_i}\right) \leq \mathcal{L}_{\tilde{S}_i^{(t-i)}}\left(h, f_{\tilde{S}_i^{(t-i)}}\right) + f'\left(S_i, \tilde{S}_i^{(t-i)}\right) \\ + \mathcal{L}_d\left(S_{i,\mathcal{X}}, \tilde{S}_{i,\mathcal{X}}^{(t-i)}\right), \end{aligned} \quad (4)$$

where $f'\big(S_i, \tilde{S}_i^{(t-i)}\big)$ is defined as :

$$f'\big(S_i, \tilde{S}_i^{(t-i)}\big) = \mathcal{L}_{\tilde{S}_i^{(t-i)}}\big(h, h_{\tilde{S}_i^{(t-i)}}\big) - \mathcal{L}_{\tilde{S}_i^{(t-i)}}\big(h, f_{\tilde{S}_i^{(t-i)}}\big) + \mathcal{L}_{S_i}(f_{S_i}, h_{S_i}) + \mathcal{L}_{\tilde{S}_i^{(t-i)}}(h_{S_i}, h_{\tilde{S}_i^{(t-i)}}), \quad (5)$$

where $f_{S_i}$ is the true labelling function for $S_i$.

The proof of Theorem 1 is provided in Appendix-A from the supplemental material[1]. From Theorem 1, we can estimate the gap on the risk bound of a certain task $\mathcal{T}_i$, achieved by a single model that is currently trained on the $t$-th task. However, this bound cannot provide an explicit way of measuring how the learning of each task can affect the performance of the model. Next, we evaluate the errors that accumulate during the continual learning.

*Theorem 2:* Let $\tilde{S}_i^{(t-i)}$ represent the joint distribution over the space $\mathcal{X} \times \mathcal{Y}$ and $\mathcal{L}$ be a loss function satisfying Assumption 1. The information loss of a single model on the task $\mathcal{T}_i$ when learning a given $(t)$-th task is given by :

$$\mathcal{L}_{S_i}\big(h, f_{S_i}\big) \leq \mathcal{L}_{S_i^{(t-i)}}\big(h, f_{\tilde{S}_i^{(t-i)}}\big) + \sum_{k=-1}^{t-i-1}\Big(\mathcal{L}_d(\tilde{S}_{i,\mathcal{X}}^{(k)}, \tilde{S}_{i,\mathcal{X}}^{(k+1)}) + f'(\tilde{S}_i^{(k)}, \tilde{S}_i^{(k+1)})\Big), \quad (6)$$

where the last term of the right hand side (RHS) is :

$$f'(\tilde{S}_i^{(k)}, \tilde{S}_i^{(k+1)}) = \mathcal{L}_{\tilde{S}_i^{(k+1)}}\big(h, h_{\tilde{S}_i^{(k+1)}}\big) - \mathcal{L}_{\tilde{S}_i^{(k+1)}}\big(h, f_{\tilde{S}_i^{(k+1)}}\big) + \mathcal{L}_{\tilde{S}_i^{(k)}}\big(f_{\tilde{S}_i^{(k)}}, h_{\tilde{S}_i^{(k)}}\big) + \mathcal{L}_{\tilde{S}_i^{(k+1)}}\big(h_{\tilde{S}_i^{(k)}}, h_{\tilde{S}_i^{(k+1)}}\big), \quad (7)$$

where we use $\tilde{S}_i^{(0)}$ to represent $S_i$.

The proof for Theorem 2 is provided in Appendix-B from the supplemental material[1].

**Remarks.** We have several observations from Theorem 2 :

- If a task was learnt during an early training stage ($i$ is small), then Eq. (7) indicates that there are more accumulated errors $\sum_{k=-1}^{t-i-1}\Big(\mathcal{L}_d(\tilde{S}_{i,\mathcal{X}}^{(k)}, \tilde{S}_{i,\mathcal{X}}^{(k+1)}) + f'(\tilde{S}_i^{(k)}, \tilde{S}_i^{(k+1)})\Big)$. These accumulated errors are mainly caused by the repetitive GRM processes. This also indicates that the model $\mathcal{M}(\theta^t, \varsigma^t, \varphi^t)$ would tend to forget more about the earlier tasks than those learnt more recently.
- The discrepancy between the source and target distributions when learning new tasks is crucial for a tight risk bound. If the distance measure in Eq. (7) is small, then the risk bound becomes tight, resulting in a better performance on the target distribution.
- A robust generative replay network leads to a tight risk bound in Eq. (7) since it can better approximate the target distribution at each training step. This conclusion was also theoretically and empirically proved in [36].

[1]The file 'Supplemental material.pdf' in https://github.com/dtuzi123/GMM

*Lemma 1:* Based on the Assumption 1, we have the risk bound of a single model on all its known tasks when learning the $(t)$-th task, as :

$$\sum_{i=1}^{t}\Big\{\mathcal{L}_{S_i}\big(h, f_{S_i}\big)\Big\} \leq \sum_{i=1}^{t}\Big\{\mathcal{L}_{S_i^{(t-i)}}\big(h, f_{\tilde{S}_i^{(t-i)}}\big) + \sum_{k=-1}^{t-i-1}\Big(\mathcal{L}_d(\tilde{S}_{i,\mathcal{X}}^{(k)}, \tilde{S}_{i,\mathcal{X}}^{(k+1)}) + f'(\tilde{S}_i^{(k)}, \tilde{S}_i^{(k+1)})\Big)\Big\}. \quad (8)$$

The proof of Lemma 1 is based on the results from Theorem 2, where we sum the accumulated error terms from Eq. (6) for learning $(t)$ tasks leading to Eq. (8). It follows from Lemma 1 that a GMM model with a single component, by minimizing the discrepancy distance term $\mathcal{L}_d(\tilde{S}_{i,\mathcal{X}^{(k)}}, \tilde{S}_{i,\mathcal{X}^{(k+1)}})$ between the distribution approximated by the model and the target distribution when learning each task, can achieve optimal performance.

In practice, a model with a single component suffers a great loss of information when learning an increasing number of tasks on its own, especially when each task is associated with an entirely different data probabilistic representation. This is because by repeatedly using GRM processes leads to stacking accumulated errors in the risk bound, as shown by Eq. (8).

### C. Risk bounds for the dynamic expansion model

In the following we extend the theoretical analysis from addressing a single static component to that of a dynamically expanding model.

*Definition 6:* (*Dynamic expansion model.*) Let $\mathbf{M} = \{\mathcal{M}_1, \cdots, \mathcal{M}_K\}$ be a mixture model made up of $K$ components, where each $\mathcal{M}_j$ represents the $j$-th component consisting of a generator $g_{\theta_j} : \mathcal{Z} \to \mathcal{X}$ and a classifier $h_{\varsigma_j} : \mathcal{X} \to \mathcal{Y}$. We also allow each component to be reused for learning new tasks through GRM processes to relieve forgetting.

We propose a risk bound for an optimal mixture model that dynamically builds new components when learning novel information ensuring that the resulting mixture model would not suffer from forgetting.

*Lemma 2:* Let us consider an optimal mixture model with $K$ components trained at the $t$-th task learning, where the number of tasks is the same as that of components, $K = t$. We define the risk bound of the optimal dynamic model as :

$$\sum_{i=1}^{t}\Big\{\mathcal{L}_{S_i}\left(h_{\varsigma_i}, f_{S_i}\right)\Big\} \leq \sum_{i=1}^{t}\Big\{\mathcal{L}_{\tilde{S}_i^{(0)}}\big(h_{\varsigma_i}, f_{\tilde{S}_i^{(0)}}\big) + f'\big(S_i, \tilde{S}_i^{(0)}\big) + \mathcal{L}_d\big(S_{i,\mathcal{X}}, \tilde{S}_{i,\mathcal{X}}^{(0)}\big)\Big\}. \quad (9)$$

The detailed proof is provided in Appendix-C from SM[1].

Lemma 2 provides an optimal solution for a dynamic expansion model in which lifelong learning can be seen as the generalization model under multiple target-source domain matching pairings. The prediction of a dynamic expansion model is made by the output of a certain selected component (classifier) from $\{h_{\zeta_i} \in \mathcal{H} \,|\, i = 1, \ldots, t\}$ in the mixture system. Therefore, the performance of each target dataset relies on the generalization ability of the associated classifier.

In order to ensure an efficient data representation, with minimal computational and memory resources, we aim to train a dynamic expansion model by assuming that the number of given tasks is eventually larger than the number of components. In this case, a flexible risk bound, considering the information loss of a dynamic expansion model, under different component configurations, is proposed in the following.

*Definition 7:* (*Retraining through repeated GRM processes.*) Let $B = \{b_1, \ldots, b_j\}$ denote a set of labels where each $b_i$ represents the task that is only trained once. The corresponding distribution for $b_i$ is defined as $\tilde{S}^{(0)}_{b_i}$. Let $B' = \{b'_1, \ldots, b'_n\}$ represent a set of labels, where each $b'_j$ indicates that the $b'_j$-th task was used for re-training through repeated GRM processes. We also define a set $\hat{B} = \{\hat{b}_1, \ldots, \hat{b}_n\}$, where each $\hat{b}_i > 1$ denotes that the $b'_i$-th task was trained using GRM processes for $(\hat{b}_i)$ times $\tilde{S}^{(0)}_{b'_i} \rightarrow \tilde{S}^{(\hat{b}_i)}_{b'_i}$, where $\tilde{S}^{(\hat{b}_i)}_{b'_i}$ represents the corresponding probabilistic representation.

*Lemma 3:* Let us consider the risk bounds for the dynamic expansion model, when re-training with previously seen tasks. Let $f_t(b_i)$ be a function that receives the task ID/label and returns the index of the component $h_{\zeta_{f_t(b_i)}}$ which was trained with the given task (the $b_i$-th task) and $\zeta$ are the parameters. The risk bound for a dynamic expansion model $\mathbf{M}$ training at the $(t)$-th task can be derived by :

$$\begin{aligned}
&\sum_{i=1}^{t}\Big\{\mathcal{L}_{S_i}\left(h_{\zeta_i}, S_i\right)\Big\} \leq \sum_{i=1}^{\mathrm{card}(B)}\Big\{\mathcal{L}_{\tilde{S}^{(0)}_{b_i}}\big(h_{\zeta_{f_t(b_i)}}, f_{\tilde{S}^{(0)}_{b_i}}\big) \\
&+ f'\big(S_{b_i}, \tilde{S}^{(0)}_{b_i}\big) + \mathcal{L}_d\big(S_{b_i,\mathcal{X}}, \tilde{S}^{(0)}_{b_i,\mathcal{X}}\big)\Big\} \\
&+ \sum_{i=1}^{\mathrm{card}(B')}\Big\{\mathcal{L}_{\tilde{S}^{(t-\hat{b}_i)}_{b'_i}}\big(h_{\zeta_{f_t(b'_i)}}, f_{\tilde{S}^{(t-\hat{b}_i)}_{b'_i}}\big) \\
&+ \sum_{k=-1}^{\hat{b}_i-1}\Big(\mathcal{L}_d(\tilde{S}^{k}_{b'_i,\mathcal{X}}, \tilde{S}^{(k+1)}_{b'_i,\mathcal{X}}) + f'(\tilde{S}^{k}_{b'_i}, \tilde{S}^{(k+1)}_{b'_i})\Big)\Big\},
\end{aligned} \tag{10}$$

where $f_t(b_i)$ in Eq. (10) ensures that the classifier after learning $b_i$ tasks is selected to evaluate the risk bound for $S_{b_i}$ in Eq. (10). We consider $t > 1$ to represent the number of tasks to be learned and $\mathrm{card}(\cdot)$ represents the cardinal number in a set. We omit the component index in Eq. (10) for the sake of simplification. We also know that $0 \leq \mathrm{card}(B) \leq K$, $0 \leq \mathrm{card}(B') \leq K$, $\mathrm{card}(B) + \mathrm{card}(B') > K$ and $\mathrm{card}(B') = \mathrm{card}(\hat{B})$, where $K$ represents the total number of mixture components that were built for learning $\mathbf{M}$. The risk for training a single component model $M(\theta^t, \omega^t, \psi^t)$ is defined as $\mathrm{R}_{\mathrm{single}}$ in Eq. (8), while $\mathrm{R}_{\mathrm{mixture}}$ is the RHS of Eq. (10), represents the risk for the dynamic expansion model :

$$\mathrm{R}_{\mathrm{single}} \geq \mathrm{R}_{\mathrm{mixture}} \,. \tag{11}$$

The proof of Lemma 3 is provided in Appendix-D from the supplemental material[1].

**Remark.** There are several observations from Lemma 3.

- Lemma 3 provides the risk-bound analysis for the dynamic expansion model in a realistic environment. We consider a number of $(K)$ components resulted after training the mixture model $\mathbf{M}$ according to certain conditions such us memory constraints or task complexity.
- We observe that $B' = \varnothing$ leads to Eq. (10) becoming identical to Eq. (9), resulting in a tight risk bound while also requiring additional network parameters. On the other hand, when we have $\mathrm{card}(B) = 1$ then $B = \{t\}$, the right-hand-side of Eq. (10) becomes as in Eq. (8), implying a large gap in the risk bound. This happens when the dynamic expansion model does not result in an architecture expansion.
- The trade-off between generalization performance and model complexity can be explained by considering the ratio $v = (K - \mathrm{card}(B))/(K - \mathrm{card}(B'))$. When $v$ increases, the model adds more parameters while also improving its generalization performance. Conversely, when $v$ decreases, the number of parameters is gradually reduced.

### D. Theoretical Analysis for the Memory-based Methods

In this section, we extend the proposed theoretical framework to analyze the forgetting in various memory-based methods.

*Definition 8:* (The memory buffer). Let $\mathcal{C}_i$ be a fixed-size memory buffer updated at the $i$-th task and $S_{\mathcal{C}_i}$ its distribution. The memory buffer $\mathcal{M}_i$ is used to relieve the forgetting at the new task learning $(\mathcal{T}_{i+1})$.

*Theorem 3:* Let $\tilde{S}_1 \otimes \cdots \tilde{S}_i$ be a joint distribution of all previously seen training datasets over $\mathcal{X} \times \mathcal{Y}$ at the $i$-th task learning. Let $\tilde{S}_i \otimes S_{\mathcal{M}_{i-1}}$ be a joint distribution of the dataset that combines $\mathcal{M}_{i-1}$ and $D^i_S$. Let $\mathcal{L}$ be a loss function satisfying Assumption 1. Then we can derive a generalization bound to describe the performance of a single classifier trained with the memory buffer $\mathcal{M}_{i-1}$ at the $i$-th task learning :

$$\begin{aligned}
&\mathcal{L}_{\tilde{S}_1\otimes\cdots\tilde{S}_i}\left(h, f_{\tilde{S}_1\otimes\cdots\tilde{S}_i}\right) \leq \mathcal{L}_{\tilde{S}_i\otimes S_{\mathcal{M}_{i-1}}}\left(h, f_{\tilde{S}_i\otimes S_{\mathcal{M}_{i-1}}}\right) \\
&+ f'\big(\tilde{S}_1 \otimes \cdots \tilde{S}_i, \tilde{S}_i \otimes S_{\mathcal{M}_{i-1}}\big) \\
&+ \mathcal{L}_d\big(\tilde{S}_{1,\mathcal{X}} \otimes \cdots \tilde{S}_{i,\mathcal{X}}, \tilde{S}_{i,\mathcal{X}} \otimes S_{\mathcal{M}_{i-1,\mathcal{X}}}\big)\,,
\end{aligned} \tag{12}$$

where $\tilde{S}_{1,\mathcal{X}} \otimes \cdots \tilde{S}_{i,\mathcal{X}}$ and $\tilde{S}_{i,\mathcal{X}} \otimes S_{\mathcal{M}_{i-1,\mathcal{X}}}$ denote the marginal distribution of $\tilde{S}_1 \otimes \cdots \tilde{S}_i$ and $\tilde{S}_i \otimes S_{\mathcal{M}_{i-1}}$, respectively. $f_{\tilde{S}_1\otimes\cdots\tilde{S}_i}$ and $f_{\tilde{S}_i\otimes S_{\mathcal{M}_{i-1}}}$ are the true labeling function for the data samples drawn from $\tilde{S}_1 \otimes \cdots \tilde{S}_i$ and $\tilde{S}_i \otimes S_{\mathcal{M}_{i-1}}$, respectively.

**Remark.** There are several observations from Theorem 3.

- Theorem 3 provides the risk-bound analysis for the memory-based methods in continual learning. Specifically, the memory buffer $\mathcal{M}_{i-1}$ stores data samples from all previous tasks $\{\mathcal{T}_1, \cdots, \mathcal{T}_{i-1}\}$ which are then replayed during the $i$-th task learning.
- Ensuring the selection of high-quality data while preserving them into the memory buffer plays an important role in achieving a good performance. For example, if the memory buffer stores more important data, that can statistically represent all previous datasets $\{D^1_S, \cdots, D^{i-1}_S\}$, the distance term $\mathcal{L}_d\big(\tilde{S}_{1,\mathcal{X}} \otimes \cdots \tilde{S}_{i,\mathcal{X}}, \tilde{S}_{i,\mathcal{X}} \otimes S_{\mathcal{M}_{i-1,\mathcal{X}}}\big)$ is reduced, leading to a smaller target risk.
- The results of Theorem 3 can be applied to describe the performance for a board range of memory-based con-

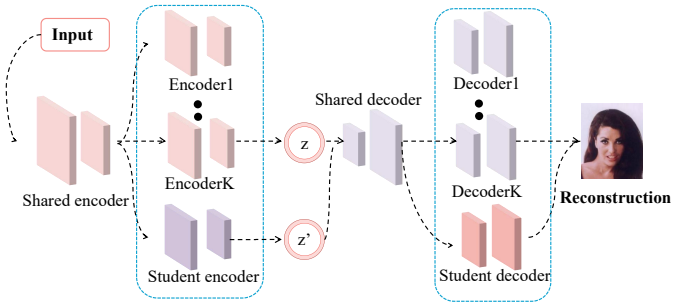



Fig. 1. The Generative Mixture Model (GMM) diagram. A few components ('Encoder K', 'Student Encoder', 'Decoder K', 'Student Decoder') update their parameters during lifelong learning. We train the Student module when learning each task by using the loss function from Eq. (25).

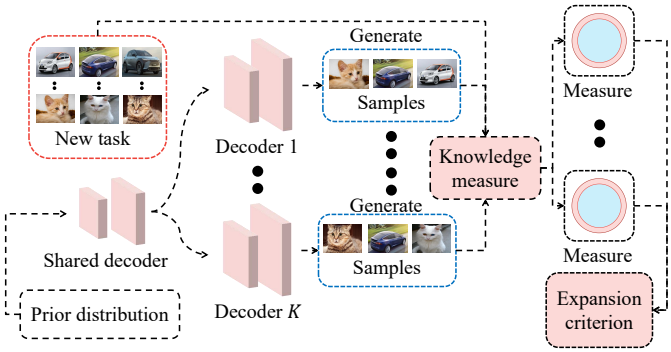



Fig. 2. Diagram illustrating the knowledge novelty decision approach used for the mixture model expansion. The probabilistic representations of samples generated by each component of $\mathbf{M}$ and that of real samples characterizing a new task are used for checking the model expansion, according to Eq. (15).

tinual learning methods, including the Dark Experience Replay (DER) [53], CarM [54], and others.

## IV. Methodology

Theorem 2 shows that a single VAE suffers from more forgetting when learning an increasing number of tasks. In order to address this drawback, the theoretical analysis from Lemma 3 shows that a dynamic expansion mixture model requires a trade-off between the resulting model complexity and its performance. Inspired by this theoretical framework, we develop a new Growing Mixture Model (GMM) that satisfies two aspects. First, it addresses forgetting in lifelong learning by preserving the learned knowledge in the frozen network structure while dynamically adding new components to a growing mixture, when learning new tasks. Then, in order to reduce the complexity of the model while preserving its performance on previously learnt data, it shares some of its parameters between these components by employing a new knowledge assessment mechanism.

### A. The Growing Mixture Model (GMM)

In the following we detail the proposed GMM's network architecture. In order to reduce the whole model's parameters, we consider a shared module, represented by two networks $\mathrm{f}^e_{\omega_s}: \mathcal{X} \to \mathcal{Z}'$ and $\mathrm{f}^d_{\theta_s}: \mathcal{Z} \to \mathcal{X}'$ for encoding and decoding data, $\mathcal{Z}'$ and $\mathcal{X}'$ are the feature space and the reconstructed data, respectively. $\{\omega_s, \theta_s\}$ are the parameters of the shared module. During the GMM expansion, we also dynamically build component-specific modules, each consisting of two neural networks $\mathrm{f}^e_{\tilde{\omega}_i}: \mathcal{Z}' \to \mathcal{Z}$ and $\mathrm{f}^d_{\tilde{\theta}_i}: \mathcal{X}' \to \mathcal{X}$ where $i$ is the component index. The encoder for the $i$-th component is implemented by $q_{\omega_{s,i}}(\mathbf{z}\,|\,\mathbf{x}) = \mathrm{f}^e_{\omega_i} \odot \mathrm{f}^e_{\tilde{\omega}_s}(\mathbf{x})$, where $\odot$ indicates that the shared model function $\mathrm{f}^e_{\omega_s}$ is nested within the component-specific function $\mathrm{f}^e_{\omega_i}$. Similarly with the encoder, we implement the decoder for the $i$-th component as $p_{\theta_{s,i}}(\mathbf{x}\,|\,\mathbf{z}) = \mathrm{f}^e_{\theta_i} \odot \mathrm{f}^e_{\tilde{\theta}_s}(\mathbf{z})$, where $\odot$ indicates that the specific subdecoder $\mathrm{f}^e_{\theta_i}(\mathbf{z})$ is built on top of the shared decoder $\mathrm{f}^e_{\theta_s}$. Since each component has an independent encoder and decoder, the optimization for the $i$-th component corresponds to maximizing the Evidence Lower Bound (ELBO) [9] :

$$\begin{aligned}\mathcal{L}^i_{ELBO}(\mathbf{x};\theta,\omega) := \mathbb{E}_{\mathbf{z}\sim q_{\omega_{s,i}}(\mathbf{z}\,|\,\mathbf{x})}\left[\log p_{\theta_{s,i}}(\mathbf{x}\,|\,\mathbf{z})\right] \\ - KL\left[q_{\omega_{s,i}}(\mathbf{z}\,|\,\mathbf{x})\,\|\,p(\mathbf{z})\right]\,.\end{aligned} \tag{13}$$

The shared module is only updated at the first iteration. The structure of the nested decoder and encoder is shown in Fig. 1.

### B. Selection mechanism and expansion criterion

Lemma 3 shows that the optimal performance is achieved when the number of components is equal to that of tasks. In this case, the accumulated errors are minimal. However, training a task-specific component for a mixture system would require substantial memory budgets as the number of tasks grows. To address this problem, we introduce a knowledge-evaluation criterion that guides component selection and mixture model expansion. The primary motivation lies in the observation from Lemma 3, which indicates that the accumulated errors depend on the discrepancy distance between the target and source distribution. To reduce the accumulated errors, a reasonable solution is for a given component to learn multiple tasks of a similar nature, indicated by a small discrepancy distance between the data corresponding to these tasks.

First, we consider estimating the discrepancy distance between the new task and the data representation learnt by each previously trained component. A high discrepancy distance indicates that the new task is novel enough and in this case we add a new component for learning this task. However, such an approach requires a classifier that is assumed to be trained on the labeled dataset, which is not available in unsupervised learning. We assume that at the $t$-th task learning, we have already trained $K$ components. When a new task $\mathcal{T}_{t+1}$ is provided for training, we propose to compare the data log-likelihood between the probabilistic representations of each learned component and that corresponding to the new task :

$$\begin{aligned}\mathrm{F}_{\mathcal{K}}(\mathcal{M}_i, \mathcal{T}_{t+1}) = \frac{1}{m}\sum_{k=1}^{m}\Big|\mathcal{L}^i_{ELBO}(\mathbf{x}'_{i,k};\theta,\omega) \\ - \mathcal{L}^i_{ELBO}(\mathbf{x}_{(t+1,k)};\theta,\omega)\Big|,\end{aligned} \tag{14}$$

for $i = 1, \cdots, K$, where $m$ is the number of samples that are used for the evaluation and $\mathcal{M}_i$ is the $i$-th trained component. In practice, we set $m = 5000$ samples in all experiments. Since we can not access all past samples, we use each VAE component to generate pseudo samples $\mathbf{x}'_{i,k}$, where $i$ is the component index, while $\mathbf{x}_{(t+i,k)}$ represents the $k$-th real

training sample drawn from the training set of $\mathcal{T}_{t+1}$. Eq. (14) measures the similarity of the probabilistic representation of each component with that corresponding to the new task which is used for component selection as :

$$\min_{i=1}^{K} \mathrm{F}_{\mathcal{K}}(\mathcal{M}_i, \mathcal{T}_{t+1}) \geq \lambda\,, \tag{15}$$

where $\lambda$ is a parameter defining the GMM size. The detailed component expansion and selection process is illustrated in Fig. 2. If the condition from Eq. (15) is satisfied, the GMM will add and train a new component $\mathcal{M}_{K+1}$ to the mixture $\mathbf{M}$; otherwise, it selects a component according to the criterion :

$$s = \arg\min_{i=1}^{K} \mathrm{F}_{\mathcal{K}}(\mathcal{M}_i, \mathcal{T}_{t+1})\,, \tag{16}$$

where $s$ represents the selected component index, which is then updated using the GRM.

The choice of $\lambda$ in the expansion criterion from Eq. (15), represents a trade-off between model complexity and performance. For example, if $\lambda$ is large, GMM tends to frequently reuse its components for learning new tasks while accumulating more errors due to repetitive GRM processes. On the other hand, when $\lambda$ is small, GMM tends to create more components, increasing the complexity of the model while reducing the accumulated errors.

### C. The unsupervised algorithm implementation

In the following, we summarize the unsupervised GMM algorithm implementation, while the pseudocode is provided in Algorithm 1 from the Appendix-E of the SM[1] :

- **Step 1. Training phase :** If the GMM has no components yet, we create a new component for learning the first task $\mathcal{T}_1$. Otherwise, we check whether to add or not a new component, using Eq. (15). If GMM adds a new component, then we only train the last component on $D_i^S$ at the $i$-th task learning by using Eq. (13). If GMM does not perform the expansion we use the $s$-th selected component to generate a dataset denoted as $D'$ and form a joint dataset $D_i^S = D_i^S \bigcup D'$ which is used to train that $s$-th component.
- **Step 2. The knowledge measure evaluation :** After the current task learning ($\mathcal{T}_i$) is finished, we evaluate the knowledge measure between each component and the dataset $D_{i+1}^S$ of the next task ($\mathcal{T}_{i+1}$) by using Eq. (14).
- **Step 3. The expansion and selection process :** We employ the knowledge measure to decide either the expansion or the selection process, controlled by $\lambda$. If Eq. (15) is satisfied, then we add a new component, otherwise, we choose an appropriate component using Eq. (16) for learning task $\mathcal{T}_i$. We then proceed to learning the next task $\mathcal{T}_{i+1}$ (**Step 1**).

### D. The supervised learning task

Although the proposed GMM is mainly used for unsupervised generative modeling tasks, it can be extended to supervised applications such as image-to-image translation and image classification. Therefore, we only modify the individual components to implement these applications, and the training process is otherwise similar to that for unsupervised learning.

**Image-to-Image Translation task (IoIT).** The conditional VAE (CVAE) is a well-known variant of VAEs [55], which is able to make prediction tasks, defined by :

$$\begin{aligned}\log p_{\theta_i}(\mathbf{y} \mid \mathbf{x}) \geq\ & \mathbb{E}_{q_{\omega_i}(\mathbf{z}|\mathbf{x},y)}\left[\log p_{\theta_i}(\mathbf{y} \mid \mathbf{x}, \mathbf{z})\right] \\ & - D_{KL}\left(q_{\omega_i}(\mathbf{z} \mid \mathbf{x}, \mathbf{y}) \,||\, p_{\theta_i}(\mathbf{z} \mid \mathbf{x})\right).\end{aligned} \tag{17}$$

where $p_{\theta_i}(\mathbf{z} \mid \mathbf{x})$ represents the prior network which receives $\mathbf{x}$ and returns $\mathbf{z}$. $p_{\theta_i}(\mathbf{y} \mid \mathbf{x}, \mathbf{z})$ is a recognition network, of parameters $\theta_i$, which is used for the prediction task. In this paper, we consider a reduced version of the CVAE for implementing each component in order to reduce the total number of parameters. Firstly, we replace $q_{\omega_i}(\mathbf{z} \mid \mathbf{x}, \mathbf{y})$ by employing an encoder $q_{\omega_i}(\mathbf{z} \mid \mathbf{x})$, of parameters $\omega_i$, which processes each image as input. A simple Gaussian distribution $\mathcal{N}(\mathbf{0}, \mathbf{I})$ can be used for replacing the prior $p_{\theta_i}(\mathbf{z} \mid \mathbf{x})$ reducing the model's size. Eq. (17) is reformulated as the main objective function for the Image-to-Image translation task, $\mathcal{L}_{\mathrm{IoIT}}(\mathbf{x}, \mathbf{y}; \omega_i, \theta_i)$ :

$$\begin{aligned}\mathcal{L}_{\mathrm{IoIT}}(\mathbf{x}, \mathbf{y}; \omega_i, \theta_i) =\ & \mathbb{E}_{q_{\omega_i}(\mathbf{z} \mid \mathbf{y})}\left[\log p_{\theta_i}(\mathbf{y} \mid \mathbf{x}, \mathbf{z})\right] \\ & - D_{KL}(q_{\omega_{c^t}}(\mathbf{z} \mid \mathbf{y}) \,||\, p_{\theta_i}(\mathbf{z})),\end{aligned} \tag{18}$$

where $\mathbf{y}$ in Eq. (18) belongs to the image domain and $i$ is the component index. We can also use an efficient inference procedure at the testing phase, $\mathbf{y} \sim p_{\theta_i}(\mathbf{y} \mid \mathbf{x}, \mathbf{z})$ and $\mathbf{z} \sim p_{\theta_i}(\mathbf{z})$ since each component is a reduced version of the CVAE. As we do not attempt to generate past samples by using $p_{\theta_i}(\mathbf{y} \mid \mathbf{x}, \mathbf{z})$, $\forall i = 1, \ldots, t-1$ after the task switch, the model adds a new component when seeing a new task in order to avoid forgetting.

**Classification task.** For the classification task, we allow $\mathbf{y}$ as a discrete variable (one-hot vector), representing the class label. Therefore, we implement $p_{\varsigma_i}(\mathbf{y} \mid \mathbf{x}, \mathbf{z})$ as a classifier. The inference model $p_{\varsigma_i}(\mathbf{z} \mid \mathbf{x})$ is replaced by using a simple normal distribution $p(\mathbf{z}) = \mathcal{N}(\mathbf{0}, \mathbf{I})$ in Eq. (17), and then we have the following loss function for training the classifier :

$$\begin{aligned}\mathcal{L}_{\mathrm{C}}(\mathbf{x}, \mathbf{y}; \omega_i, \varsigma_i) =\ & \mathbb{E}_{q_{\omega_i}(\mathbf{z}|\mathbf{x},\mathbf{y})}\left[\log p_{\varsigma_i}(\mathbf{y} \mid \mathbf{x}, \mathbf{z})\right] \\ & - D_{KL}(q_{\omega_i}(\mathbf{z} \mid \mathbf{x}, \mathbf{y}) \,||\, p(\mathbf{z})).\end{aligned} \tag{19}$$

Eq. (19) still requires training the generato in order to implement GRMs. We employ the following objective function for the generator :

$$\begin{aligned}\mathcal{L}_{\mathrm{Gen}}(\mathbf{x}, \mathbf{y}; \omega_i, \theta_i) =\ & \mathbb{E}_{q_{\omega_i}(\mathbf{z}|\mathbf{x},\mathbf{y})}\left[\log p_{\theta_i}(\mathbf{x} \mid \mathbf{y}, \mathbf{z})\right] \\ & - D_{KL}(q_{\omega_i}(\mathbf{z} \mid \mathbf{x}, \mathbf{y}) \,||\, p(\mathbf{z})).\end{aligned} \tag{20}$$

In practice, we optimize $\mathcal{L}_{\mathrm{Gen}}$ and $\mathcal{L}_{\mathrm{C}}$ by using data mini-batches. In addition, the loss function $\mathcal{L}_{\mathrm{Gen}}$ is also used during the testing phase for selecting the suitable component.

### E. Training a compressed Student module

We also propose a Teacher-Student architecture [11], for mapping multiple tasks into a unique latent space, where the expanding nested encoder-decoder model, described above, is considered as a Teacher which is used for training a compressed Student model [42] under unsupervised learning. The advantage of training a Student is that of being able

TABLE I
CLASSIFICATION RESULTS BY VARIOUS MODELS FOR MSFIR LEARNING.

| | MSE | | | | | SSMI | | | | | PSNR | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **Datasets** | LGM | CURL | BE | GMM | Stud | LGM | CURL | BE | GMM | Stud | LGM | CURL | BE | GMM | Stud |
| MNIST | 129.93 | 211.21 | 19.24 | 26.64 | 176.82 | 0.45 | 0.46 | 0.92 | 0.88 | 0.42 | 14.52 | 13.27 | 22.57 | 21.02 | 13.72 |
| Fashion | 89.28 | 110.60 | 38.81 | 33.67 | 178.04 | 0.51 | 0.44 | 0.61 | 0.75 | 0.37 | 15.82 | 14.89 | 14.46 | 19.68 | 8.81 |
| SVHN | 169.55 | 102.06 | 39.57 | 30.27 | 146.70 | 0.24 | 0.26 | 0.66 | 0.64 | 0.47 | 8.11 | 10.86 | 18.90 | 15.55 | 13.58 |
| IFashion | 432.90 | 115.29 | 36.52 | 35.03 | 158.18 | 0.26 | 0.54 | 0.75 | 0.77 | 0.43 | 9.04 | 15.51 | 19.32 | 19.47 | 14.17 |
| RMNIST | 130.28 | 279.47 | 25.41 | 22.97 | 157.55 | 0.45 | 0.29 | 0.88 | 0.90 | 0.43 | 14.51 | 10.84 | 21.31 | 21.71 | 14.18 |
| **Average** | 190.38 | 163.72 | 31.91 | **29.71** | 163.45 | 0.38 | 0.39 | 0.76 | **0.78** | 0.42 | 12.40 | 13.07 | 19.31 | **19.48** | 12.89 |

to compress the information learnt by a complex GMM Teacher module, implemented by an expanding VAE model as described in Section IV-A, by transferring the essence of its knowledge into a lightweight model which can be used for fast inference during the testing phase. Furthermore, the Student module enables many applications, including image reconstruction and image interpolation across multiple domains. The Teacher-Student framework structure is shown in Fig. 1, where the GMM is used as the Teacher module, and the Student module is based on a shared encoder and a shared decoder. We propose a new Knowledge Distillation (KD) loss based on the Kullback-Leibler (KL) divergence between the probabilistic representation of the Student and that of the Teacher, while also considering the information from the new task, in the context of lifelong learning. The training enables the Student to acquire knowledge from both the new task as well as from the data generated by the Teacher module :

$$D_{KL}\left(\mathrm{P}_{\theta_1}, \mathrm{P}_{\theta_2}, \ldots, \mathrm{P}_{\theta_K} \,||\, \mathrm{P}_{\theta_{stu}}\right), \tag{21}$$

where each $\mathrm{P}_{\theta_i}$ is the distribution approximated by the generator of the $i$-th component of the Teacher. $\mathrm{P}_{\theta_{stu}}$ is the distribution representing the data generated by the Student. However, directly optimizing Eq. (21) is intractable because the first KL component from Eq. (21) involves multiple distributions. Then we approximate Eq. (21) by :

$$D_{KL}\left(\mathrm{P}_{\theta_1}, \mathrm{P}_{\theta_2}, \ldots, \mathrm{P}_{\theta_K} \,||\, \mathrm{P}_{\theta_{stu}}\right) \approx \sum_{i=1}^{K} D_{KL}\left(\mathrm{P}_{\theta_i} \,||\, \mathrm{P}_{\theta_{stu}}\right), \tag{22}$$

where each $D_{KL}(\mathrm{P}_{\theta_i} \,||\, \mathrm{P}_{\theta_{stu}})$ can be expressed as :

$$D_{KL}\left(\mathrm{P}_{\theta_i} \,||\, \mathrm{P}_{\theta_{stu}}\right) = \mathbb{E}_{\mathrm{P}_{\theta_i}}\left[\log p_{\theta_i}(\mathbf{x}) - \log p_{\theta_{stu}}(\mathbf{x})\right], \tag{23}$$

where $p_{\theta_i}(\mathbf{x})$ and $p_{\theta_{stu}}(\mathbf{x})$ represent the functions for $\mathrm{P}_{\theta_i}$ and $\mathrm{P}_{\theta_{stu}}$. The first term in Eq. (23) can be omitted during the optimization because we are only required to update the parameters $\theta_{stu}$ of the Student module during the knowledge distillation procedure.

Then, the knowledge distillation loss for training the Student module, after considering the optimization problem from Eq. (23) for all components in the Teacher, is represented as a maximization problem :

$$\max\left\{\sum_{i=1}^{K} \mathbb{E}_{\mathrm{P}_{\theta_i}}\left[\log p_{\theta_{stu}}(\mathbf{x})\right]\right\}. \tag{24}$$

By considering the $(t)$-th task learning, we construct a loss function for the Student :

$$\mathcal{L}_{stu} = \mathbb{E}_{\mathbf{x}\sim S_{t,\mathcal{X}}}\left[\log p_{\theta_{stu}}(\mathbf{x})\right] + \sum_{i=1}^{K}\left\{\mathbb{E}_{\mathbb{P}_{\theta_i}}\left[\log p_{\theta_{stu}}(\mathbf{x})\right]\right\}, \tag{25}$$

where $S_{t,\mathcal{X}}$ is the distribution of the training dataset $D_t^S$. $\log p_{\theta_{stu}}(\mathbf{x}) = \log \int p_{\theta_{stu}}(\mathbf{x}\,|\,\mathbf{z})\; p(\mathbf{z})\mathrm{d}\mathbf{z}$ is intractable due to having to integrate over all latent variables $\mathbf{z}$. Then we estimate each log-likelihood function using ELBO :

$$\underbrace{\mathbb{E}_{q_{\omega_{Stu}}(\mathbf{z}|\mathbf{x})}\left[\log p_{\theta_{Stu}}(\mathbf{x}\mid\mathbf{z})\right] - D_{KL}\left[q_{\omega_{Stu}}(\mathbf{z}\mid\mathbf{x}) \,||\, p(\mathbf{z})\right]}_{\text{ELBO on the t-th task}}$$
$$+ \underbrace{\sum_{i=1}^{K} \mathbb{E}_{\mathbf{x}'\sim\mathbb{P}_{\theta_i}}\left(\mathbb{E}_{q_{\omega_{Stu}}(\mathbf{z}\,|\,\mathbf{x})}\left[\log p_{\theta_{Stu}}(\mathbf{x}\mid\mathbf{z})\right] - D'_{KL}\right)}_{\text{Knowledge distillation optimization}}, \tag{26}$$

where $D'_{KL}$ is defined as :

$$D_{KL}\left[q_{\omega_{Stu}}(\mathbf{z}\mid\mathbf{x}) \,||\, p(\mathbf{z})\right]. \tag{27}$$

In Fig. 1, we illustrate the Teacher-Student network architecture, where a GMM is the Teacher while the Student is characterized by the specific parameters $\{\theta_{stu}, \omega_{stu}\}$ while sharing the other parameters with the joint network. This approach reduces the complexity of the whole system while enabling the training of a lightweight Student module in the testing phase. Furthermore, during training, the Student module is always activated to compress the knowledge from the new task as well as from all trained GMM components. We provide the pseudo-code of the teacher-student framework in Algorithm 2 of Appendix-E from the supplemental material[1]. We also provide the theoretical analysis of the knowledge assimilation by the Student module in Appendix-F from SM[1].

## V. EXPERIMENTS

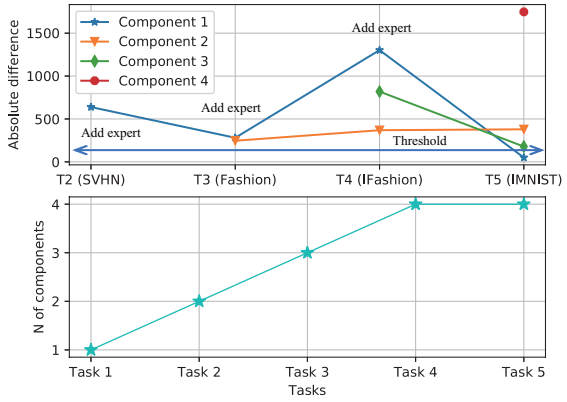
In the following, we provide the experiments and discuss the results of the proposed methodology.

### A. Implementation settings

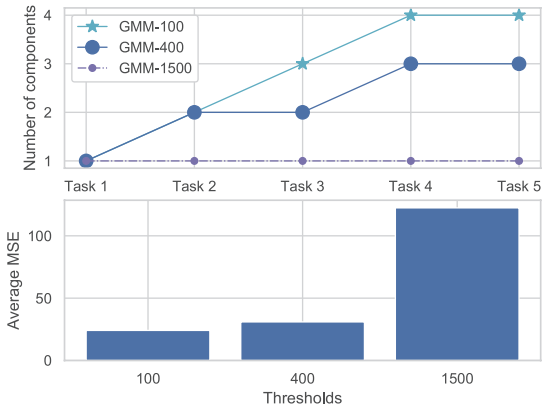
We use TensorFlow for the implementation of all models and adopt Adam optimization algorithm [56], where we consider a learning rate of $2 \cdot 10^{-4}$ and $\beta$ is 0.5. We implement the shared encoder by using a CNN with four layers of $\{128, 256, 512, 1024\}$ units. We implement each sub-encoder $\tilde{\omega}_i$ by two fully-connected layers $\{1024, 100\}$. For the

TABLE II
RESULTS AFTER THE CONTINUAL LEARNING OF CCCOS DATASETS.

| | MSE | | | | | SSMI | | | | | PSNR | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **Datasets** | LGM | CURL | BE | GMM | Stud | LGM | CURL | BE | GMM | Stud | LGM | CURL | BE | GMM | Stud |
| CelebA | 1536.06 | 1446.86 | 209.93 | 214.55 | 646.95 | 0.33 | 0.34 | 0.69 | 0.69 | 0.49 | 15.14 | 15.42 | 23.61 | 23.52 | 18.71 |
| CACD | 2348.35 | 2202.88 | 459.93 | 363.17 | 1394.11 | 0.26 | 0.27 | 0.55 | 0.62 | 0.38 | 13.16 | 13.40 | 20.21 | 21.28 | 15.38 |
| 3D-Chair | 1430.87 | 1258.02 | 629.55 | 483.29 | 1527.70 | 0.43 | 0.47 | 0.73 | 0.80 | 0.47 | 15.68 | 16.18 | 19.26 | 20.72 | 15.60 |
| Omniglot | 3356.40 | 2464.04 | 753.30 | 361.33 | 4258.15 | 0.20 | 0.26 | 0.78 | 0.89 | 0.28 | 11.76 | 13.13 | 18.55 | 21.99 | 10.75 |
| Sub-IM | 1147.64 | 1336.58 | 773.89 | 783.21 | 1064.51 | 0.30 | 0.32 | 0.37 | 0.37 | 0.32 | 15.80 | 16.07 | 18.47 | 18.44 | 17.06 |
| Average | 1963.86 | 1741.68 | 565.30 | **441.11** | 1778.29 | 0.30 | 0.33 | 0.62 | **0.67** | 0.39 | 14.31 | 14.84 | 20.02 | **21.19** | 15.50 |




(a) Knowledge evaluation (top) and the number of components (bottom).




(b) No. of components (top) and the performance vs threshold $\lambda$ (bottom).

Fig. 3. Evaluating the model's complexity considering its number of components for the MSFIR's lifelong learning.

decoding process, we implement the shared part $\theta_S$ by using a CNN that has 3 fully connected layers with $\{256, 8, 8\}$ units and 2 layers with $\{256, 256\}$ units. Then we implement the sub-encoder $\tilde{\theta}_i$ by using a CNN that consists of three layers of $\{256, 256, 3\}$ units. For an input image of size of $64 \times 64 \times 3$, the shared encoder $\omega_S$ is implemented by a CNN with 3 layers with $\{64, 128, 256\}$ units. We also implement the sub-encoder $\tilde{\omega}_i$ by using a network that consists of a convolution layer with 256 units and two fully connected layers of $\{1024, 256\}$ units. For the decoding process, we implement the shared decoder $\theta_S$ by using a network that consists of three convolution layers $\{256, 256, 256\}$ and then three layers of $\{256, 8, 8\}$ fully connected units. We then implement the sub-decoder $\theta_i$ by using a CNN that consists of $\{256, 128, 3\}$ processing units.

### B. Datasets and evaluation criteria

This section provides information about the datasets used for training and the evaluation criteria.

- For unsupervised learning, we consider a sequence of tasks, where we resize all images to $32 \times 32 \times 3$ pixels. The datasets used for experiments are MNIST [57], SVHN [58], Fashion [59], InverseFashion (IFashion) and Rotated MNIST (RMNIST), which contains images which are rotations of those from MNIST (MSFIR sequence).
- For supervised classification, we incorporate CIFAR10 [60] after MSFIR, resulting in the MSFIRC sequence.

**Evaluation criteria:** For supervised learning, we evaluate the average classification accuracy on all tasks as the performance criterion. Meanwhile, for the unsupervised learning, we consider the structural similarity index measure (SSIM) [61], the MSE and PSNR [61] in order to evaluate the reconstruction quality.

**Baselines:** We compare the results of the proposed methodology with those from three popular continual learning models : Continual Unsupervised Representation Learning (CURL) [13], Lifelong Generative Modeling (LGM) [10] and BatchEnsemble (BE) [15]. BE was originally used in supervised learning while here is used for unsupervised learning after being implemented as a mixture model where each component is a VAE having some trainable vectors onto a joint network. We only update this joint neural network at the first task learning while freezing it afterwards in order to avoid forgetting.

### C. Generative modeling tasks

We provide the classification results for the lifelong learning of MSFIR, with 20 epochs for learning each task, in Table I, where 'Stud' represents the results for the Student model which accesses the generated information from the GMM considered as a Teacher within a Student-Teacher architecture. From Table I, we observe that the proposed GMM outperforms the baseline in each task, demonstrating the advantage of the proposed approach.

### D. Continual learning of datasets containing complex images

GMM is evaluated when considering tasks involving the learning of databases consisting of complex images, described

TABLE III
CLASSIFICATION ACCURACY ACHIEVED BY VARIOUS MODELS AFTER THE CONTINUAL LEARNING OF MSFIRC.

| **Dataset** | LGM [10] | CURL [13] | BE[15] | GMM | MRGANs [66] |
|---|---|---|---|---|---|
| MNIST | 90.54% | 91.30% | 99.40% | 99.44% | 91.24% |
| SVHN | 22.56% | 62.05% | 74.46% | 85.13% | 64.12% |
| Fashion | 68.29% | 79.18% | 88.95% | 91.49% | 80.10% |
| IFashion | 73.70% | 82.51% | 86.45% | 68.75% | 82.19% |
| RMNIST | 90.52% | 98.56% | 99.10% | 98.50% | 98.30% |
| CIFAR10 | 57.43% | 67.34% | 52.48% | 65.27% | 67.19% |
| **Average** | 67.17% | 80.16% | 83.47% | **84.76**% | 80.52% |

TABLE IV
CLASSIFICATION RESULTS FOR THE LIFELONG LEARNING OF MSFIRRC.

| **Dataset** | GMM | BE [15] |
|---|---|---|
| MNIST | 86.01% | 99.28% |
| SVHN | 86.91% | 74.84% |
| Fashion | 90.68% | 87.60% |
| IFashion | 91.02% | 86.03% |
| RMNIST | 99.01% | 98.77% |
| RFashion | 91.43% | 86.60% |
| CIFAR10 | 64.61% | 54.79% |
| **Average** | **87.10**% | 83.99% |

in the following :

**Sub-ImageNet (Sub-IM).** This is a subset of ImageNet where the number of images associated with each task is balanced by randomly collecting 60,000 samples from the ImageNet [62].

**CelebA** is a large-scale face attributes dataset which has more than 200K face images of celebrities [63].

**CACD** is another human face dataset consisting of 163,446 images from 2,000 persons collected from the internet [64].

**3D-Chair** contains rendered images of around 1,000 different three-dimensional chair models [65].

For Sub-IM we consider 50,000 samples as the training set while the other 10,000 are the testing set. For CelebA, CACD and 3D-Chair, we randomly select ninety percent of all data from each dataset as the training set while all remaining samples are used for testing. We build a sequence of tasks for learning complex images from CelebA, CADS, 3D-Chair, Omniglot and Sub-IM databases, namely CCCOS, where we resize all images to $64 \times 64 \times 3$. Various models are trained under CCCOS and we report the results in Table II. We can observe that GMM outperforms other baselines in both CCCOS and MSFIR settings.

### *E. Classification tasks*

In the following we compare GMM with the state-of-the-art in image classification. LGM [10] is designed for the unsupervised learning, but it can be adapted for supervised classification, by training a classifier on the joint dataset consisting of real samples and the generated samples from LGM. We report the classification accuracy in Table III, which shows that the proposed GMM outperforms other models. Meanwhile, BE also achieves good performance, where we consider the same number of components as that of tasks, ensuring optimal performance (further detail in Lemma 3).

TABLE V
RESULTS FOR SPLIT CIFAR WHERE 'C' DENOTES THE NUMBER OF COMPONENTS FOR THE PROPOSED GMM MODEL.

| **Methods** | Split CIFAR |
|---|---|
| FROMP-$L_2$ [67] | $75.6\% \pm 0.4$ |
| FROMP [67] | $76.2\% \pm 0.4$ |
| SI [68] | $73.5\% \pm 0.5$ |
| VCL + random coreset [31] | $67.4\% \pm 1.4$ |
| EWC [28] | $71.6\% \pm 0.9$ |
| GMM | $76.40\% \pm 0.3$ (6 C) |
| GMM | 65.70% (5 C) |

TABLE VI
RESULTS FOR THE CONTINUOUS LEARNING BENCHMARK WHERE 'C' DENOTES THE NUMBER OF COMPONENTS FOR THE PROPOSED GMM.

| **Methods** | Permuted MNIST | Split MNIST |
|---|---|---|
| Improved VCL* [69] | $93.1\% \pm 1$ | $98.4\% \pm 0.4$ |
| EWC* [28] | 84% | 63.1% |
| DLP* [70] | 82% | 61.2% |
| SI* [68] | 86% | 98.9% |
| FROMP* [67] | $94.9\% \pm 0.1$ | $99.0\% \pm 0.1$ |
| FRCL-TR* [71] | $94.3\% \pm 0.2$ | $97.8\% \pm 0.7$ |
| FRCL-RND* [71] | $94.2\% \pm 0.1$ | $97.1\% \pm 0.7$ |
| GMM | **96.46**%$\pm$0.03 (10 C) | **99.21**%$\pm$0.04 (5 C) |
| GMM | 88.78% (7 C) | 96.77% (4 C) |
| GMM | 95.25% (8 C) | 91.37% (3 C) |

We also test various models when learning many tasks, by considering a sequence consisting of MNIST, SVHN, Fashion, IFashion, RMNIST, rotated Fashion (RFashion) and CIFAR10 (MSFIRRC). We consider $\lambda = 180$ in Eq. (15) for the proposed GMM. After all tasks are learnt, GMM has five components, reusing the first component which learns MNIST and learning afterwards a similar task (RMNIST), and also reusing the third component which firstly learns Fashion and then RFashion. According to the classification results from Table IV GMM outperforms BE in terms of the average classification accuracy as well as on each dataset, except for MNIST.

Despite the fact that the GMM is used for task-incremental learning, we also investigate its performance in the class-incremental setting. Following the setting from [67], we create a new dataset, namely Permuted MNIST, where MNIST is divided into ten tasks, and each task would process images created following a certain pixel permutation in the images from MNIST. Split MNIST [68] splits MNIST into five tasks, where each of these contains samples belonging to two successive classes of digits. For Permuted MNIST, the classifier in each expert is implemented by using a Multilayer Perceptron (MLP) with 2 hidden layers with 100 units each. The classifier of each expert is a neural network comprised of 2 layers with 256 units on each layer when training on Split MNIST. We vary the threshold $\lambda$ from Eq. (15), between 80 to 120 for Permuted MNIST and between 30 and 60 for Split MNIST. We also consider a more challenging dataset, Split CIFAR [68] using the same procedure as in [67], where CIFAR10 is considered as the initial task followed by five tasks, where the training samples for each task are selected from 10 data

TABLE VII
CLASSIFICATION RESULTS, WHERE THE RESULTS FOR ALL OTHER METHDSO THAN GMM ARE CITED FROM [72].

| **Methods** | Split CIFAR10 | R-MNIST |
|---|---|---|
| DER [53] | 70.51% ± 1.67 | 97.57% ± 1.47 |
| ER [73] | 57.74% ± 0.27 | 94.89% ± 0.95 |
| DER++ [53] | 72.70% ± 1.36 | 97.54% ± 0.43 |
| iCaRL [74] | 47.55% ± 3.95 | - |
| A-GEM [75] | 22.67% ± 0.57 | 89.04% ± 7.01 |
| HAL [76] | 41.79% ± 4.46 | 92.35% ± 0.81 |
| GEM [33] | 26.20% ± 1.26 | 92.55% ± 0.85 |
| FDR [77] | 28.71% ± 3.23 | 95.48% ± 0.68 |
| GSS [34] | 49.73% ± 4.78 | 89.38% ± 3.12 |
| $Co^2L$ [72] | 74.26% ± 0.77 | 98.65% ± 0.31 |
| GMM | 76.12% ± 1.35 (4C) | 98.78% ± 1.12 (18C) |

categories in their class order from CIFAR100. Following from [67], we adopt a network architecture that consists of four convolution layers and two fully connected layers. The shared classifier is implemented by four convolution layers and we build a sub-classifier during the expansion process by using two fully connected layers added on the top layer of the shared classifier. Therefore, each sub-classifier reuses parameters from this shared classifier, thus reducing model's size. The parameters of the shared classifier are only updated when learning the first task to relieve forgetting. We vary the threshold $\lambda$ from 80 to 100 and perform five independent runs for GMM. The average classification accuracy for GMM in Permuted MNIST, Split MNIST and Split CIFAR, are provided in Tables V and VI, with all other results cited from [67]. The comparison models Functional Regularisation of Memorable Past (FROMP) and FROMP-$L_2$ models were proposed in [67], while Synaptic Intelligence (SI) was proposed in [68]. We denote by '6 C' that GMM trains six components. These results show that optimal performance is ensured when GMM has a number of components equal to that of the learnt tasks, which is consistent with Lemma 2.

We also compare with several recently proposed continual learning models on Split CIFAR10 and R-MNIST dataset [33] that consists of 20 tasks, where each task is constructed by rotating MNIST images by a random angle between $[0, 180)$. The classification results on Split CIFAR10 and R-MNIST are reported in Table VII, which shows that the proposed GMM outperforms other baselines.

### *F. Comparison with State-of-the-Art*

We compare the proposed framework with the state-of-the-art. Given that the proposed GMM framework has a flexible network design, we can implement the shared model for all experts using a large-size pre-trained vision transformer [78]. Each new expert is then implemented using a simple fully connected layer reducing the overall model size. In the class-incremental learning setting, each task usually contains non-overlapped category information. As a result, we dynamically build a new expert when seeing a new task. At the testing phase, we consider the VAE of each expert to evaluate the sample log-likelihood in order to choose the appropriate expert for the prediction of the given testing samples. We compare with the following recent and popular continual learning methods :

- Dark Experience Replay (DER) [53] is a popular memory-based approach, which employs a reservoir sampling to update the memory buffer. DER++ is an advanced verson of the DER, which adds an additional loss term to enforce the higher conditional likelihood with respect to the ground truth labels [53].
- Gradient Episodic Memory (GEM) [33] is a memory-based approach, which calculates the gradient information to guide the model's optimization. Averaged-GEM (AGEM) [75] is an advanced version of the GEM, which reduces computational costs.
- ICL w Pure-MM [79] is a recent continual learning approach consisting of two systems, relying on a Visual Transformer (ViT) [80], which cooperatively deals with network forgetting in continual learning.
- Incremental Mixture of Experts (IMOE) [47] is a new dynamic expansion model, which is built based on the CLIP framework [81]. For a fair comparison with the proposed GMM framework, we employ the pre-trained ViT as the backbone for IMOE instead of CLIP. In addition, we employ the same network architecture for implementing both GMM and IMOE, respectively.

The results for Split CIFAR10, Split CIFAR100 and Split TinyImageNet are reported in Table VIII. The proposed GMM framework outperforms most baselines by a large margin on these datasets. In addition, by dynamically building lightweight experts, when learning new tasks, onto the pre-trained ViT as the backbone leads to significant performance improvements, as shown by the results of ICL w Pure-MM and IMOE. In addition, the proposed GMM framework outperforms the current state-of-the-art method IMOE, demonstrating its effectiveness. We also provide the number of parameters and the training times for the baselines in Table IX. We observe that the dynamic expansion models such as IMOE and GMM have more parameters than other baselines. This is because both IMOE and GMM dynamically create new experts based on the large-scale pre-trained ViT backbone. In addition, compared to the static model, dynamic expansion models only update the parameters of one expert, when learning each task. Furthermore, the proposed GMM framework requires fewer parameters and enjoys a faster training process than the current state-of-the-art, IMOE.

### *G. Ablation study*

We investigate the significance of each module from the proposed GMM. In Fig. 3a we evaluate the knowledge in the mixture model, showing the evaluation of Eq. (15) in the top plot while the bottom one evaluates the number of components, where GMM expands with four more components through the lifelong learning of the MSFIR sequence. The first component which initially has learnt MNIST, is afterwards trained on a similar dataset, namely RMNIST, which consists of rotated images from MNIST. The proposed

TABLE VIII
THE AVERAGE ACCURACY EVALUATED ON STANDARD CONTINUAL LEARNING BENCHMARKS, CONSIDERING 10 RUNS. THE RESULTS OF BASELINES ARE TAKEN FROM [53] AND [82], [79].

| Methods | Split CIFAR10 | Split TinyImageNet | Split CIFAR100 |
|---|---|---|---|
| ER [73] | 93.61 ± 0.27 | 48.64 ± 0.46 | 73.37 ± 0.43 |
| GEM [33] | 92.16 ± 0.69 | - | - |
| A-GEM [75] | 89.48 ± 1.45 | 25.33 ± 0.49 | 48.06 ± 0.57 |
| iCaRL [74] | 88.22 ± 2.62 | 31.55 ± 3.27 | - |
| FDR [77] | 93.29 ± 0.59 | 49.88 ± 0.71 | - |
| GSS [34] | 91.02 ± 1.57 | - | 57.50 ± 1.93 |
| HAL [76] | 84.54 ± 2.36 | - | 42.94 ± 1.80 |
| DER [53] | 93.40 ± 0.39 | 51.78 ± 0.88 | - |
| DER++ [53] | 93.88 ± 0.50 | 51.91 ± 0.68 | 75.64 ± 0.60 |
| DER+++refresh [82] | 94.64 ± 0.38 | 54.06 ± 0.79 | 77.71 ± 0.85 |
| ICL w Pure-MM [79] | **99.68** | - | 96.35 |
| IMOE [47] | 96.27 ± 0.51 | 90.02 ± 0.65 | 94.52 ± 0.85 |
| GMM | 98.65 ± 0.46 | **91.62** ± 0.86 | **96.56** ± 0.73 |

TABLE IX
THE NUMBER OF PARAMETERS AND THE TRAINING TIMES OF VARIOUS MODELS. 'PARAMS' AND 'TIMES' DENOTE THE NUMBER OF PARAMETERS AND THE TRAINING TIMES (HOURS), RESPECTIVELY.

| | Split CIFAR10 | | Split TinyImageNet | | Split CIFAR100 | |
|---|---|---|---|---|---|---|
| Methods | Params | Times | Params | Times | Params | Times |
| DER++ [53] | 11M | 3.10 | 11M | 26.23 | 11M | 5.35 |
| DER+++refresh [82] | 11M | 10.25 | 11M | 30.62 | 11M | 12.98 |
| IMOE [47] | 108M | 10.42 | 114M | 24.26 | 135M | 13.62 |
| GMM | 98M | 8.62 | 107M | 20.82 | 116M | 10.58 |

GMM is tested by changing $\lambda$ from Eq. (15), and the results are shown in Fig. 3b. A large $\lambda$ allows for the GMM to reuse existing components more frequently, resulting in a lower performance. In contrast, a small $\lambda$ leads to GMM expanding with more components during the training, increasing the model's complexity while improving its performance. We provide additional ablation results in Appendix-G from SM[1].

## VI. CONCLUSIONS AND FUTURE WORK

This paper develops a theoretical analysis framework for lifelong learning models by assessing their forgetfulness when learning multiple tasks. This analysis results in the derivation of risk bounds for evaluating the ability to learn new information by considering the already assimilated knowledge. A Growing Mixture Model (GMM) is then proposed based on a knowledge measure that evaluates the information novelty during the lifelong learning. The analysis shows that by dynamically adding and training new components in a mixture model, we can significantly relieve forgetting and improve the generalization performance. This approach allows the proposed GMM to gradually increase its capacity while reusing existing components when learning related tasks. The experimental results indicate that GMM outperforms other baselines in both supervised and unsupervised lifelong learning.

In the following we discuss some limitations and developments for future work. The primary limitation of the proposed GMM framework is its incompatibility with the deployment on extremely resource-limited devices while attempting to learn an infinite array of tasks, as the GMM framework dynamically expands its size throughout the training phase. To mitigate this challenge, one viable solution is to establish a knowledge distillation method that effectively consolidates multiple akin teacher components into a single one, thereby facilitating a reduction in the overall model size. This strategy allows the GMM framework to progressively accumulate its knowledge while retaining a constant model size. Alternatively, component pruning, by systematically identifying and eliminating knowledge-redundant components from the GMM framework, can be adopted. Furthermore, an additional challenge confronting the proposed GMM framework is the degradation of the model's performance when addressing limited data scenarios, such as in the continual few-shot learning [83], where the number of data instances per task is minimal. This issue can be addressed by using a generative model for each expert.

In our proposed GMM framework, we incorporate a shared backbone network to streamline model complexity. A significant challenge arises as the shared backbone network only updates its parameters during the initial task and remains static for subsequent tasks. The results indicate that the proposed GMM framework consistently delivers superior performance across various data domains, despite the shared backbone being static after learning the first task. Conversely, when experts possess independent parameters and are not reliant on a shared backbone, the GMM framework's performance is enhanced due to the increased number of trainable parameters. However, parameter sharing among experts can result in performance instability with changing task sequences, as shown in the results from Fig. 3 from the Appendix-G in the supplementary material. The reason behind this result is that the shared backbone captures and learns different information for different learning orders of tasks. In a future study we will aim to develop a novel regularization method that incrementally optimizes the shared backbone network, facilitating future task learning while preserving the entire previously acquired knowledge.

## REFERENCES


[1] F. Ye and A. G. Bors, "Learning joint latent representations based on information maximization," *Inforation Sciences*, vol. 567, pp. 216–236, 2021.

[2] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proc. of IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 779–788.

[3] M.-Y. Liu, T. Breuel, and J. Kautz, "Unsupervised image-to-image translation networks," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2017, pp. 700–708.

[4] A. Odena, C. Olah, and J. Shlens, "Conditional image synthesis with auxiliary classifier GANs," in *Proc. of Int. Conf. on Machine Learning (ICML), vol. PMLR 70*, 2017, pp. 2642–2651.

[5] F. Ye and A. G. Bors, "Mixtures of variational autoencoders," in *Proc. Int. Conf. on Image Proc. Theory, Tools and Applic. (IPTA)*, 2020, pp. 1–6.

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

[7] H. Shin, J. K. Lee, J. Kim, and J. Kim, "Continual learning with deep generative replay," in *Advances in Neural Inf. Proc. Systems (NeurIPS)*, 2017, pp. 2990–2999.

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

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

[10] J. Ramapuram, M. Gregorova, and A. Kalousis, "Lifelong generative modeling," *Neurocomputing*, vol. 404, pp. 381–400, 2020.

[11] F. Ye and A. G. Bors, "Lifelong teacher-student network learning," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 44, no. 10, pp. 6280–6296, 2022.

[12] A. Srivastava, L. Valkov, C. Russell, M. U. Gutmann, and C. Sutton, "VEEGAN: Reducing mode collapse in GANs using implicit variational learning," in *Adv. in Neural Inf. Proc. Systems (NeurIPS)*, 2017, pp. 3308–3318.

[13] D. Rao, F. Visin, A. A. Rusu, Y. W. Teh, R. Pascanu, and R. Hadsell, "Continual unsupervised representation learning," in *Advances in Neural Information Proc. Systems (NeurIPS)*, 2019, pp. 7645–7655.

[14] S. Lee, J. Ha, D. Zhang, and G. Kim, "A neural Dirichlet process mixture model for task-free continual learning," *Int. Conf. of Learning Representations (ICLR), arXiv preprint arXiv:2001.00689*, 2020.

[15] Y. Wen, D. Tran, and J. Ba, "BatchEnsemble: an alternative approach to efficient ensemble and lifelong learning," in *Int. Conf. on Learning Representations (ICLR), arXiv preprint arXiv:2002.06715*, 2020.

[16] S. Ebrahimi, F. Meier, R. Calandra, T. Darrell, and M. Rohrbach, "Adversarial continual learning," in *Proc. European Conf. on Computer Vision (ECCV), vol. LNCS 12356*, 2020, pp. 386–402.

[17] Y. Mansour, M. Mohri, and A. Rostamizadeh, "Domain adaptation: Learning bounds and algorithms," in *Proc. Conf. on Learning Theory (COLT), arXiv preprint arXiv:2002.06715*, 2009.

[18] Z. Wu, Y. Wu, X. Wang, Y. Gan, and J. Pu, "A robust diffusion modeling framework for radar camera 3d object detection," in *Proc. of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, 2024, pp. 3282–3292.

[19] H. Ling, K. Kreis, D. Li, S. W. Kim, A. Torralba, and S. Fidler, "Editgan: High-precision semantic image editing," *Adv. in Neural Inf. Proc. Systems (NeurIPS)*, vol. 34, pp. 16 331–16 345, 2021.

[20] J. Wu, R. Fu, H. Fang, Y. Zhang, Y. Yang, H. Xiong, H. Liu, and Y. Xu, "MedSegDiff: Medical image segmentation with diffusion probabilistic model," in *Medical Imaging with Deep Learning*. PMLR 227, 2024, pp. 1623–1639.

[21] J. Bang, H. Kim, Y. Yoo, J.-W. Ha, and J. Choi, "Rainbow memory: Continual learning with a memory of diverse samples," in *Proc. IEEE/CVF Conf. on Computer Vision and Pattern Recog.*, 2021, pp. 8218–8227.

[22] F.-Y. Wang, D.-W. Zhou, H.-J. Ye, and D.-C. Zhan, "Foster: Feature boosting and compression for class-incremental learning," in *Proc. of the European Conference on Computer Vision (ECCV), vol. LNCS 13685*, 2022, pp. 398–414.

[23] D. Zhou, Q. Wang, H. Ye, and D. Zhan, "A model or 603 exemplars: Towards memory-efficient class-incremental learning," in *Int. Conf. on Learning Representations (ICLR), arXiv preprint arXiv:2205.13218*, 2023.

[24] J. Zhang, D. Zhou, and M. Chen, "Adaptive cointegration analysis and modified RPCA with continual learning ability for monitoring multimode nonstationary processes," *IEEE Trans. on Cybernetics*, vol. 53, no. 8, pp. 4841–4854, 2023.

[25] W. Chen, B. Chen, Y. Liu, X. Cao, A. Zhao, H. Zhang, and L. Tian, "Max-margin deep diverse latent Dirichlet allocation with continual learning," *IEEE Trans. on Cyber.*, vol. 52, no. 7, pp. 5639 – 5653, 2022.

[26] Z. Wang, C. Chen, and D. Dong, "A Dirichlet process mixture of robust task models for scalable lifelong reinforcement learning," *IEEE Trans. on Cybernetics*, vol. 53, no. 12, pp. 7509–7520, 2023.

[27] Z. Li and D. Hoiem, "Learning without forgetting," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 40, no. 12, pp. 2935–2947, 2017.

[28] J. Kirkpatrick, R. Pascanu, N. Rabinowitz, J. Veness, G. Desjardins, A. A. Rusu, K. Milan, J. Quan, T. Ramalho, A. Grabska-Barwinska, D. Hassabis, C. Clopath, D. Kumaran, and R. Hadsell, "Overcoming catastrophic forgetting in neural networks," *Proc. of the National Academy of Sciences (PNAS)*, vol. 114, no. 13, pp. 3521–3526, 2017.

[29] J. Schwarz, W. Czarnecki, J. Luketina, A. Grabska-Barwinska, Y. W. Teh, R. Pascanu, and R. Hadsell, "Progress & compress: A scalable framework for continual learning," in *Proc. of Int. Conf. on Machine Learning (ICML), vol. PMLR 80*, 2018, pp. 4535–4544.

[30] J. Martens and R. B. Grosse, "Optimizing neural networks with Kronecker-factored approximate curvature," in *Proc. of Int. Conf. on Machine Learning (ICML)*, vol. 37, 2015, pp. 2408–2417.

[31] C. V. Nguyen, Y. Li, T. D. Bui, and R. E. Turner, "Variational continual learning," in *International Conference on Learning Representations (ICLR), arXiv preprint arXiv:1710.10628*, 2018.

[32] K. Javed and M. White, "Meta-learning representations for continual learning," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2019, p. 1820–1830.

[33] D. Lopez-Paz and M. Ranzato, "Gradient episodic memory for continual learning," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2017, pp. 6467–6476.

[34] R. Aljundi, M. Lin, B. Goujaud, and Y. Bengio, "Gradient based sample selection for online continual learning," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2019, pp. 11 817–11 826.

[35] J. Bang, H. Koh, S. Park, H. Song, J.-W. Ha, and J. Choi, "Online continual learning on a contaminated data stream with blurry task boundaries," in *Proc. of IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022, pp. 9275–9284.

[36] F. Ye and A. G. Bors, "Learning latent representations across multiple data domains using lifelong VAEGAN," in *Proc. European Conf. on Computer Vision (ECCV), vol. LNCS 12365*, 2020, pp. 777–795.

[37] A. Achille, T. Eccles, L. Matthey, C. Burgess, N. Watters, A. Lerchner, and I. Higgins, "Life-long disentangled representation learning with cross-domain latent homologies," in *Advances in Neural Inf. Proc. Systems (NeurIPS)*, 2018, pp. 9873–9883.

[38] H. Xu, L. Szymanski, and B. McCane, "VASE: Variational assorted surprise exploration for reinforcement learning," *IEEE Trans. on Neural Networks and Learning Systems*, vol. 34, no. 3, pp. 1243–1252, 2023.

[39] F. Ye and A. G. Bors, "InfoVAEGAN: learning joint interpretable representations by information maximization and maximum likelihood," in *Proc. of IEEE Int. Conf. on Image Processing*, 2021, pp. 749–753.

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

[41] E. Egorov, A. Kuzina, and E. Burnaev, "BooVAE: Boosting approach for continual learning of VAE," *Advances in Neural Information Processing Systems (NeurIPS)*, pp. 17 889 – 17 901, 2021.

[42] F. Ye and A. G. Bors, "Dynamic self-supervised teacher-student network learning," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 45, no. 5, pp. 5731–5748, 2023.

[43] ——, "Lifelong generative modelling using dynamic expansion graph model," in *Proc. of the AAAI Conference on Artificial Intelligence*, vol. 36, no. 8, 2022, pp. 8857–8865.

[44] ——, "Lifelong mixture of variational autoencoders," *IEEE Trans. on Neural Networks and Learning Syst.*, vol. 34, no. 1, pp. 461–474, 2023.

[45] ——, "Lifelong infinite mixture model based on knowledge-driven dirichlet process," in *Proc. of IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2021, pp. 10 695–10 704.

[46] ——, "Deep mixture generative autoencoders," *IEEE Trans. on Neural Networks and Learning Systems*, vol. 33, no. 10, pp. 5789–5803, 2022.

[47] J. Yu, Y. Zhuge, L. Zhang, P. Hu, D. Wang, H. Lu, and Y. He, "Boosting continual learning of vision-language models via mixture-of-experts adapters," in *Proc. of IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024, pp. 23 219–23 230.

[48] S. Ben-David, J. Blitzer, K. Crammer, and F. Pereira, "Analysis of representations for domain adaptation," in *Advances in Neural Inf. Proc Systems (NIPS)*, 2007, pp. 137–144.

[49] C. Cortes and M. Mohri, "Domain adaptation and sample bias correction theory and algorithm for regression," *Theoretical Computer Science*, vol. 519, pp. 103–126, 2014.

[50] G. K. Dziugaite, D. M. Roy, and Z. Ghahramani, "Training generative neural networks via maximum mean discrepancy optimization," in *Proc. of Conf. on Uncertainty in Artificial Intell. (UAI)*, 2015, pp. 258–267.

[51] C.-L. Li, W.-C. Chang, Y. Cheng, Y. Yang, and B. Póczos, "MMD GAN: Towards deeper understanding of moment matching network," in *Advances in Neural Inf. Proc. Systems (NeurIPS)*, 2017, pp. 2203–2213.

[52] Y. Li, K. Swersky, and R. Zemel, "Generative moment matching networks," in *Proc. Int. Conf. on Machine Learnning (ICML)*, 2015, pp. 1718–1727.

[53] P. Buzzega, M. Boschini, A. Porrello, D. Abati, and S. Calderara, "Dark experience for general continual learning: a strong, simple baseline," *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 33, pp. 15 920–15 930, 2020.

[54] S. Lee, M. Weerakoon, J. Choi, M. Zhang, D. Wang, and M. Jeon, "CarM: Hierarchical episodic memory for continual learning," in *Proc. of ACM/IEEE Design Automation Conference*, 2022, pp. 1147–1152.

[55] K. Sohn, H. Lee, and X. Yan, "Learning structured output representation using deep conditional generative models," in *Advances in Neural Inf. Proc. Systems (NIPS)*, 2015, pp. 3483–3491.

[56] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Int. Conf. on Learning Representations (ICLR), arXiv preprint arXiv:1412.6980*, 2015.

[57] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proc. of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

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

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

[60] A. Krizhevsky and G. Hinton, "Learning multiple layers of features from tiny images," Tech. Rep., 2009.

[61] A. Hore and D. Ziou, "Image quality metrics: PSNR vs. SSIM," in *Proc. Int. Conf. on Pattern Recognition (ICPR)*, 2010, pp. 2366–2369.

[62] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Advances in Neural Inf. Proc. Systems (NIPS)*, 2012, pp. 1097–1105.

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

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

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

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

[67] P. Pan, S. Swaroop, A. Immer, R. Eschenhagen, R. Turner, and M. E. E. Khan, "Continual deep learning by functional regularisation of memorable past," in *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 33, 2020, pp. 4453–4464.

[68] F. Zenke, B. Poole, and S. Ganguli, "Continual learning through synaptic intelligence," in *Proc. of Int. Conf. on Machine Learning, vol. PLMR 70*, 2017, pp. 3987–3995.

[69] S. Swaroop, C. V. Nguyen, T. D. Bui, and R. E. Turner, "Improving and understanding variational continual learning," in *Proc. NIPS-workshops Continual Learning, arXiv preprint arXiv:1905.02099*, 2018.

[70] A. J. Smola, S. Vishwanathan, and E. Eskin, "Laplace propagation," in *Advances in Neural Inf. Proc. Systems (NIPS)*, 2004, pp. 441–448.

[71] M. K. Titsias, J. Schwarz, A. G. d. G. Matthews, R. Pascanu, and Y. W. Teh, "Functional regularisation for continual learning with Gaussian processes," in *International Conf. on Learning Represenations (ICLR), arXiv preprint arXiv:1901.11356*, 2019.

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

[73] M. Riemer, I. Cases, R. Ajemian, M. Liu, I. Rish, Y. Tu, and G. Tesauro, "Learning to learn without forgetting by maximizing transfer and minimizing interference," in *Int. Conf. on Learning Representations (ICLR), arXiv preprint arXiv:1810.11910*, 2019.

[74] S.-A. Rebuffi, A. Kolesnikov, G. Sperl, and C. H. Lampert, "iCaRL: Incremental classifier and representation learning," in *Proc. of IEEE Conf. on Computer Vision and Pattern Recog.*, 2017, pp. 2001–2010.

[75] A. Chaudhry, M. Ranzato, M. Rohrbach, and M. , "Efficient lifelong learning with A-GEM," in *International Conf. on Learning Representations (ICLR), arXiv preprint arXiv:1812.00420*, 2019.

[76] A. Chaudhry, A. Gordo, P. Dokania, P. Torr, and D. Lopez-Paz, "Using hindsight to anchor past knowledge in continual learning," in *Proc. of AAAI Conf. on Artificial Intelligence*, 2021, pp. 6993–7001.

[77] A. S. Benjamin, D. Rolnick, and K. Kording, "Measuring and regularizing networks in function space," in *Int. Conf. on Learning Representations (ICLR), arXiv preprint arXiv:1805.08289*, 2019.

[78] Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin, and B. Guo, "Swin transformer: Hierarchical vision transformer using shifted windows," in *Proc. of IEEE/CVF Int. Conf. on Computer Vision (ICCV)*, 2021, pp. 10 012–10 022.

[79] B. Qi, X. Chen, J. Gao, D. Li, J. Liu, L. Wu, and B. Zhou, "Interactive continual learning: Fast and slow thinking," in *Proc. of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024, pp. 12 882–12 892.

[80] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby, "An image is worth 16x16 words: Transformers for image recognition at scale," in *Int. Conf. on Learning Represent (ICLR), arXiv preprint arXiv:2010.11929*, 2020.

[81] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark *et al.*, "Learning transferable visual models from natural language supervision," in *Proc. Int. Conf. on Machine Learning (ICML)*. PMLR 139, 2021, pp. 8748–8763.

[82] Z. Wang, Y. Li, L. Shen, and H. Huang, "A unified and general framework for continual learning," in *International Conference on Learning Representations (ICLR), arXiv preprint arXiv:2403.13249*, 2024.

[83] E. Lee, C.-H. Huang, and C.-Y. Lee, "Few-shot and continual learning with attentive independent mechanisms," in *Proc. of IEEE/CVF Int. Conf. on Computer Vision (ICCV)*, 2021, pp. 9455–9464.


**Fei Ye** is currently working at the University of Electronic Science and Technology of China. Fei Ye obtained the PhD degree in computer science from the University of York, UK. He received the bachelor degree from Chengdu University of Technology, China, in 2014 and the master degree in computer science and technology from Southwest Jiaotong University, China, in 2018. His research topics includes deep generative image models, lifelong learning and mixture models.

**Adrian G. Bors** (Senior Member, IEEE) received the M.Sc. degree in electronics engineering from the Polytechnic University of Bucharest, Bucharest, Romania, in 1992, and the Ph.D. degree in informatics from the University of Thessaloniki, Thessaloniki, Greece, in 1999. In 1999 he joined the Department of Computer Science, Univ. of York, UK, where he is currently an Associate Professor. Dr. Bors was a Research Scientist at University of Tampere, Finland, and held visiting positions at the Univ. of California at San Diego (UCSD), the Univ. of Montpellier, France and at the MBZ University of Artificial Intelligence, Abu Dhabi, UAE. He was an Associate Editor of the IEEE TRANSACTIONS ON IMAGE PROCESSING from 2010 to 2014 and the IEEE TRANSACTIONS ON NEURAL NETWORKS from 2001 to 2009. He.. was also a Co-Guest Editor for special issues for the International Journal for Computer Vision in 2018 and the Journal of Pattern Recognition in 2015. Dr. Bors has authored and co-authored more than 180 research papers, including 50 in journals. His research interests include machine learning, computer vision, pattern recognition and image processing.