
Article:
Ye, Fei and Bors, Adrian Gheorghe orcid.org/0000-0001-7838-0021 (2023) Lifelong Mixture of Variational Autoencoders. IEEE Transactions on Neural Networks and Learning Systems. pp. 461-474. ISSN: 2162-237X

https://doi.org/10.1109/TNNLS.2021.3096457

Lifelong Mixture of Variational Autoencoders

Fei Ye and Adrian G. Bors
Department of Computer Science, University of York, York YO10 5GH, UK

***Abstract*—In this paper, we propose an end-to-end lifelong learning mixture of experts. Each expert is implemented by a Variational Autoencoder (VAE). The experts in the mixture system are jointly trained by maximizing a mixture of individual component evidence lower bounds (MELBO) on the log-likelihood of the given training samples. The mixing coefficients in the mixture model, control the contributions of each expert in the global representation. These are sampled from a Dirichlet distribution whose parameters are determined through non-parametric estimation during the lifelong learning. The model can learn new tasks fast when these are similar to those previously learnt. The proposed Lifelong mixture of VAE (L-MVAE) expands its architecture with new components when learning a completely new task. After the training, our model can automatically determine the relevant expert to be used when fed with new data samples. This mechanism benefits both the memory efficiency and the required computational cost as only one expert model is used during the inference. The L-MVAE inference model is able to perform interpolations in the joint latent space across the data domains associated with different tasks and is shown to be efficient for disentangled learning representation.**

***Index Terms*—Lifelong learning, Mixture of Variational Autoencoders, Multi-task learning, Mixture of Evidence Lower Bounds, Disentangled representations.**

I. Introduction

Deep learning models suffer from catastrophic forgetting [1] when training on multiple databases in a sequential manner, indicating that a model quickly forgets the characteristics of the previously learned experiences while adjusting to learning new information. The ability of artificial learning systems of continuously acquiring, preserving, transferring skills and knowledge throughout their lifespan is called lifelong learning [1]. Existing related approaches adopt three different methodologies: using dynamic architectures, embedding regularization during the training, and employing generative replay mechanisms. Dynamic architecture approaches [2], [3], [4], [5], [6] would increase the network capacity by adding new layers and new processing units on each layer in order to adapt the network's architecture to acquiring new information. However, such approaches would require a specific architecture design while the number of parameters would increase progressively with the number of tasks. Regularization approaches [7], [8], [9], [10], [11] aim to impose a penalty when updating the network' parameters in order to preserve the knowledge associated with previously learned tasks. These approaches, in practice, suffer from performance degradation when learning a series of tasks where the datasets are entirely different from the previously learned ones. Memory-based methods use a buffer in order to upload previously learned data samples [12], [13], or utilize powerful generative networks such as a Variational Autoencoders (VAEs) [14], [15], [16], [17] or a Generative Adversarial Networks (GANs) [18], [19] as a memory-based replay network that reproduces and generates data which is consistent with what has seen and learned before. These approaches would need additional memory storage space in order to store the generated data while the performance on the previously learned tasks is heavily dependent on the generator's ability to realistically replicate data.

Recently, the state of the art methods show promising results on prediction tasks [6], [7], [20], [21], [22], [23] but they do not capture the underlying structure behind the data, which prevents them from being applied in a wide range of applications. There are only very few attempts addressing representation learning under the lifelong setting [16], [15]. The performance of these methods degrades significantly when engaging in the lifelong training on datasets containing complex images or on a long sequence of tasks. The reason is that these approaches require to retrain their generators on artificially generated data. Meanwhile, the performance loss on each dataset is accumulated during the lifelong learning of a sequence of several tasks. To address this problem, we propose a probabilistic mixture of experts model, where each expert infers a probabilistic representation of a given task. A Dirichlet sampling process defines the likelihood of a certain expert to be activated when presented with a new task.

This paper has the following contributions :

- We propose a novel mixture learning model, namely the Lifelong Mixture of VAEs (L-MVAE). Instead of capturing different characteristics of a database as in other mixture models [24], [25], [26], [27], the proposed mixture model enables to automatically embed the knowledge associated with each database into a distinct latent space modelled by one of the mixture's experts during the lifelong learning.
- A mixing-coefficient sampling process is introduced in L-MVAE in order to activate or drop out experts. Besides defining an adaptive architecture for the model, this procedure accelerates the learning process when acquiring new tasks while overcoming forgetting of the previously learned tasks.

The remainder of the paper contains a detailed overview of the existing state of the art in Section II, while the proposed L-MVAE model is discussed in Section III. In Section IV we discuss the theory behind the proposed L-MAE model, while in Section V we explain how the proposed methodology can be used in unsupervised, supervised and semi-supervised learning applications. The expansion mechanism for the model's architecture is presented in Section VI. The experimental results are analyzed in Section VII while the conclusions are drawn in Section VIII.

II. Related research studies

A variational autoencoder (VAE) [28] is made up of two networks, an encoder and a decoder. Given a data set, the encoder extracts a latent vector $\mathbf{z}$, and the decoder aims to reconstruct the given data from the latent vectors. A number of research works have been developed for capturing meaningful and disentangled data representations by using the VAE framework [29], [30], [31], [32], [33]. These approaches show promising results on achieving disentanglement between latent variables as well as interpretable visual results, where specific properties of the scene can be manipulated through changing the relevant latent variables. However, these models work well only on data samples drawn from a single domain, corresponding to a specific database used for training. When they are re-trained on a different database, their parameters are updated and then they fail to perform on the tasks learned previously. This happens because they do not have appropriate objective functions to deal with catastrophic forgetting [9], [34], [35].

Recently, there have been some attempts to learn cross-domain representations under the lifelong learning by introducing an environment-dependent mask that specifies a subset of generative factors [16], or by proposing a teacher-student lifelong learning framework [15] and a hybrid model [36] based on Generative Adversarial Nets (GANs) [37] and VAEs. The models proposed in [15], [16], [36] are based on the Generative Replay Mechanisms (GRM) aiming to overcome forgetting. However, these methods suffer from poor performance when considering complex data.

Aljundi *et al.* [38] proposed a lifelong learning system named the Expert Gate model, where new experts are added to a network of experts. The most relevant autoencoder from the given set of experts is chosen during the testing stage, according to the reconstruction error of the data. This may not necessarily correspond to the best log-likelihood estimate for the best data. Moreover, the Expert Gate model was used only for supervised classification tasks.

Regularization based approaches alleviate catastrophic forgetting by adding an auxiliary term that penalizes changes in the weights when the model is trained on a new task [6], [7], [8], [9], [10], [11], [35], [39], [40], [41] or store past samples to regulate the optimization [20], [42]. However, regularization based approaches have huge computation requirements when the number of tasks increases [43].

In another direction of research, mixtures of VAEs have been employed for continuous learning [24], [25], [26], [27]. These models are able to capture underlying complex structures behind data and therefore perform well on many downstream tasks including clustering and semi-supervised classification. However, these mixture models would only capture characteristics of a single database, which had been split into batches, and tend to forget previously learned data characteristics when attempting to learn a sequence of distinct tasks. In contrast to the above mentioned methods, our model is able to capture underlying generative latent variable representations across multiple data domains during the lifelong learning.

III. The Lifelong Mixture of VAEs

A. Problem formulation

In this paper we consider a model made up of a mixture of networks which is able to deal with three different learning scenarios: supervised, semi-supervised and unsupervised, under the lifelong learning setting. Let us consider a sequence of tasks and denote $\mathcal{D}^{(k)} = \{\mathbf{x}_i^{(k)}, \mathbf{y}_i^{(k)}\}_i^{N_k}$ as a dataset characterizing the $k-th$ task, where $\mathbf{x}_i^{(k)} \in \mathcal{X}^{(k)}$ is the source domain and $\mathbf{y}_i^{(k)} \in \mathcal{Y}^{(k)}$ is the target domain which is usually defined by class labels, while each domain $\{\mathcal{D}^{(i)} | i = 1, \ldots, K\}$ is associated to a given task. We aim to learn a model which not only generates or reconstructs data but which can also generate meaningful representations useful for various tasks during a lifelong learning process.

B. Mixture objective function

Traditional mixture models [44], [45] normally capture different characteristics of a dataset by learning several latent variable vectors, with distinct sets of variables associated to each mixture' component. In this paper, we implement each expert by using a generative latent variable model, such as a VAE, $p_\theta(\mathbf{x}, \mathbf{z}) = p_\theta(\mathbf{x}|\mathbf{z})p(\mathbf{z})$, where $\mathbf{z} \in I\!R^d$ is the latent variable and θ represents the decoder's parameters. The learning goal of the generative model is to maximize the log-likelihood of the data distribution, which is actually a difficult problem due to the intractability of the marginal distribution $p(\mathbf{x}) = \int p_\theta(\mathbf{x}|\mathbf{z})p(\mathbf{z})d\mathbf{z}$, requiring access to all latent variables. Instead, we optimize the evidence lower bound (ELBO) on the data log-likelihood, [28] :

$$\begin{aligned} \log p(\mathbf{x}) &\geq \mathbb{E}_{\mathbf{z}\sim q_\varepsilon(\mathbf{z}|\mathbf{x})}[\log p_\theta(\mathbf{x}|\mathbf{z})] - D_{KL}[q_\varepsilon(\mathbf{z}|\mathbf{x})||p(\mathbf{z})] \\ &= \mathcal{L}_{\mathrm{VAE},\theta,\varepsilon}(\mathbf{x}), \end{aligned} \tag{1}$$

where $q_\varepsilon(\mathbf{z}|\mathbf{x})$ is called the variational distribution, and ε represents the parameters of the encoder. We use the Gaussian distribution for both the prior $p(\mathbf{z})$ as well as for the variational distribution $q_\varsigma(\mathbf{z}|\mathbf{x})$. The latent variable $\mathbf{z}$ is sampled using the reparametrization trick [28] $\mathbf{z}_i = \mathbf{u}_i + \delta \otimes \sigma_i$, where $\mathbf{u}_i$ and σ_i are inferred by the encoder, and δ is sampled from $\mathcal{N}(0, \mathbf{I})$. $p_\theta(\mathbf{x}|\mathbf{z})$ is implemented by a decoder with trainable parameters θ, receiving the latent variables $\mathbf{z}$ and producing data reconstructions $\mathbf{x}'$.

When considering that we have K experts in the mixture model, we introduce the loss function named MELBO, as the Mixture of individual ELBOs $\mathcal{L}^i_{VAE}(\mathbf{x})$, defined through (1) :

$$\mathcal{L}_{L-MVAE}(\mathbf{x}) = \frac{\sum_{i=1}^{K} w_i \mathcal{L}^i_{VAE}(\mathbf{x})}{\sum_{i=1}^{K} w_i}, \tag{2}$$

where w_i is the mixing coefficient, which controls the significance of the i-th expert. We model all mixing coefficients by using a Dirichlet distribution $\{w_1, \ldots, w_K\} \sim \mathrm{Dir}(\mathbf{a})$, of parameters $\mathbf{a} = \{a_1, \ldots, a_K\}$. In the following we describe a mechanism for selecting appropriate L-MVAE components during the training.

C. The selection of L-MVAE mixture's components during training

Certain research studies [24], [25] have considered equal contributions for the components of deep learning mixture systems. However, in this paper we consider that each mixture component is specialized for a specific task. The selection of a specific mixture component is performed through the mixing weights w_i, $i = 1, \ldots, K$. We assume that the weighting probability for each mixture's component is drawn from a Multinomial distribution, such as the Bernoulli distribution, defined by a Dirichlet prior.

Assignment vector. In the following, we introduce an assignment vector $\mathbf{c}$, with each of its entries $c_i \in \{0, 1\}$, $i = 1, \ldots, K$, representing the probability of including or not the i-th expert in the mixture. c_i is sampled from as Bernoulli distribution. Before starting the training, we set all entries as $c_i = 0$, $i = 1, \ldots, K$. The assignment probability for each mixing component is calculated considering the sample log-likelihood of each expert after learning each task, as :

$$p(c_j) = 1 - \frac{\exp(-\mathcal{L}^j_{VAE}(\mathbf{x}_b)) + u\, c'_j}{\sum\limits_{i=1}^{K} \exp(-\mathcal{L}^i_{VAE}(\mathbf{x}_b)) + u\, c'_i}, \tag{3}$$

where $\mathbf{x}_b$ is a data sample sampled from the given data batch, drawn from the database corresponding to the current task learning. c'_j denotes the assignment variable for j-th expert, before evaluating Eq. (3), and represents the value resulted when learning the previous task. $u\, c'_j$ is used to ensure that $p(c_j)$ is outside the range of possible values for $c'_j = 1$, when evaluating (3), and therefore we consider u as a large value. Then we find the maximum probability for a mixing component :

$$p(c_{j^*}) = \max(p(c_1), \ldots, p(c_K)), \tag{4}$$

where j^* represents the index corresponding to the selected VAE component according to the parameters learnt during the previous tasks. We then normalize the other assignment variables, except for j^* by :

$$p(c_i) = \begin{cases} 1 & c'_i = 1 \\ 0 & c'_i = 0 \end{cases}, \; i = 1, 2, \ldots, K, \; i \neq j^*. \tag{5}$$

Since c'_i is an assignment corresponding to the learning process of the previous task, before evaluating Eq. (3), in order to determine the dropout status of i-th expert during the current task learning, we use Eq. (5) to recover the dropout status of all experts except for j^*-th expert which is actually dropped out from the future training because it is going to be used for recording and reproducing the information associated with the current task being learnt. When learning the first task, all mixture's components will be trained and then when leaning the second task, only $K - 1$ components are trained, while one component is no longer trained because it is considered as a depository of the information associated with the first task. This component will consequently be used to generate information consistent with the probabilistic representation associated with the first task. This process is continued until the last task is being learnt when at least one VAE is available for training. In consequence the number of mixing components K considered initially should be at least equal to the number of tasks assumed to be learned during the lifelong learning process. In Section VI we describe a mechanism for expanding the mixture.

The sampling of mixing weights. Suppose that L-MVAE finished learning the t-th task. We collect several batches of samples $\{\mathbf{x}_i, \ldots, \mathbf{x}_N\}$ from the $(t+1)$-th task where each $\mathbf{x}_i$ represents the i-th batch of samples, which are used to evaluate the assignment vector $\mathbf{c}$ by using Eq. (3). We calculate the average probability $p(c_j) = \sum_{i=1}^{N} p(c^i_j)/N$ where each $p(c^i_j)$ represents the probability for the assignment of the i-th batch of sample, $\mathbf{x}_i$. Then we find $p(c_{j*})$ by using Eq. (4) and we recover the previous assignments except for c_{j*} by using Eq. (5). Then, the Dirichlet parameters are calculated in order to fix the mixture components containing the information corresponding to the previously learnt tasks while making the other mixture's components available for training with the future tasks. For the mixing components that have been used for learning the previous tasks, we consider

$$a_i = \begin{cases} e & c_i = 1 \\ \frac{1 - e * K'}{K - K'} & c_i = 0, i = 1, \ldots, K' \end{cases} \tag{6}$$

where e is a very small positive value. For $i = 1, \ldots, K'$, where K' represents the number of tasks learnt so far out of a total of K given tasks, during the lifelong learning. A small value for the Dirichlet parameters implies that the corresponding mixture components are no longer trained. Then mixing weights $w_1, \ldots, w_K$ are sampled from Dirichlet distribution with parameters $a_1, \ldots, a_K$. In the final, we train the mixture model with $w_1, \ldots, w_K$ by using Eq. (2) at the $(t+1)$-th task learning.

Testing phase. Suppose that after the lifelong learning process, we have trained K components. In the testing phase, we perform a selection of a single component to be used for the given data samples. We firstly calculate the selection probability $\{v_1, \ldots, v_K\}$ by calculating the log-likelihood of the data sample for each component :

$$v_i = \frac{\exp(-\frac{1}{\mathcal{L}^j_{VAE}(\mathbf{x})})}{\sum\limits_{i=1}^{K} \exp(-\frac{1}{\mathcal{L}^i_{VAE}(\mathbf{x})})}, \;\; i = 1, \ldots, K. \tag{7}$$

Then we select a component by sampling the mixing weight vector $\mathbf{w}$ from Categorical distribution $Cat(v_1, \ldots, v_K)$.

The structure of the proposed L-MVAE model is shown in Figure 1. In the following section we evaluate the convergence properties of the proposed L-MVAE model during the lifelong learning.

IV. Theoretical analysis of L-MVAE

In this section, we evaluate the convergence properties of the proposed L-MVAE model during the lifelong learning. We evaluate the evolution of the objective function $\mathcal{L}_{L-MVAE}(\mathbf{x})$ during the training and how we can define a lower bound on the data's log-likelihood. We also show how L-MVAE model infers across several tasks during the lifelong learning.

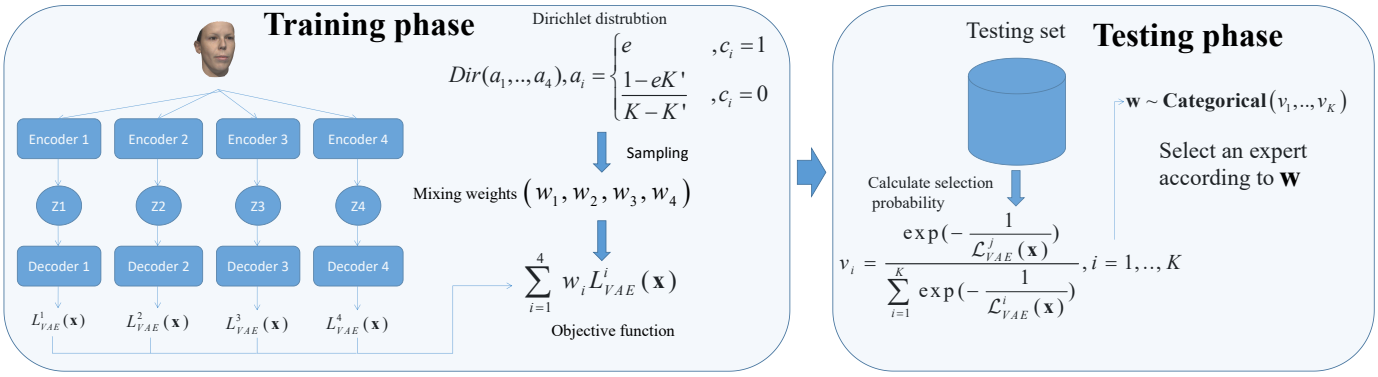


Fig. 1. The structure of the proposed Lifelong Mixture of VAEs learning system with $K = 4$ components. Each expert has an independent inference and generation process and therefore we can calculate $\mathcal{L}^i_{VAE}(\mathbf{x})$, the ELBO for each expert. Each c_i represents the probability of the assignment for the i-th component, which is used to determine each a_i by using Eq. (6). Then the mixing weights $\{w_1, \ldots, w_4\}$ are sampled from the Dirichlet distribution and are used in Eq. (2). During the testing phase, for given data samples we select an appropriate mixture component to be used.

Definition. Let us define the following function :

$$\mathcal{L}_{EMIX}(\mathbf{x}) = \sum_{i=1}^{K} w_i \exp(\mathcal{L}^i_{VAE}(\mathbf{x})) \tag{8}$$

where $\mathcal{L}^i_{VAE}(\mathbf{x})$ is defined for i-th mixture component by considering the objective function (1) and where we consider $\sum_{i=1}^{K} w_i = 1$. We also define the likelihood function for the mixture model, denoted as $\mathcal{L}_{L-Log}(\mathbf{x})$.

Lemma. By considering $\mathcal{L}_{L-Log}(\mathbf{x})$, defining the likelihood function

$$\mathcal{L}_{L-Log}(\mathbf{x}) = \sum\nolimits_{i=1}^{K} w_i \int p_{\theta_i}(\mathbf{x}|\mathbf{z}) p_{\theta_i}(\mathbf{z}) d\mathbf{z}, \tag{9}$$

and the previous *Definition*, we have :

$$\log \mathcal{L}_{L-Log}(\mathbf{x}) > \log \mathcal{L}_{EMIX}(\mathbf{x}). \tag{10}$$

Proof. After considering the latent variables $\mathbf{z}$ for each VAE component, the marginal log-likelihood of the mixture is given by :

$$\begin{aligned} \log \mathcal{L}_{L-Log}(\mathbf{x}) &= \log \left(\int \sum\nolimits_{i=1}^{K} w_i p_{\theta_i}(\mathbf{x}, \mathbf{z}) d\mathbf{z} \right) \\ &= \log \left(\sum\nolimits_{i=1}^{K} w_i \int p_{\theta_i}(\mathbf{x}|\mathbf{z}) p_{\theta_i}(\mathbf{z}) d\mathbf{z} \right), \end{aligned} \tag{11}$$

We know that $\log p_{\theta_i}(\mathbf{x}) = \log \int p_{\theta_i}(\mathbf{x}|\mathbf{z}) p_{\theta_i}(\mathbf{z}) d\mathbf{z}$ is bounded by the local ELBO objective function $\mathcal{L}^i_{VAE,\theta_i,\varepsilon_i}(\mathbf{x})$, according to (1), and we have

$$\int p_{\theta_i}(\mathbf{x}|\mathbf{z}) p_{\theta_i}(\mathbf{z}) d\mathbf{z} \geq \exp(\mathcal{L}^i_{VAE,\theta_i,\varepsilon_i}(\mathbf{x})), \tag{12}$$

where θ_i represents the parameters for the i-th mixture component.

Since the log function is a monotone increasing function, then we have:

$$\begin{aligned} &\log \left(\sum\nolimits_{i=1}^{K} w_i \int p_{\theta_i}(\mathbf{x}|\mathbf{z}) p_{\theta_i}(\mathbf{z}) d\mathbf{z} \right) \geq \\ &\log \left(\sum\nolimits_{i=1}^{k} w_i \exp(\mathbb{E}_{q_{\varepsilon_i}(\mathbf{z}|\mathbf{x})}[\log p_{\theta_i}(\mathbf{x}|\mathbf{z}) - \log q_{\varepsilon_i}(\mathbf{z}|\mathbf{x})]) \right), \end{aligned} \tag{13}$$

which proves the *Lemma*.

Theorem 1: Optimizing the mixture's objective function, $\mathcal{L}_{L-MVAE}(\mathbf{x})$, corresponds to finding a lower bound on the log-likelihood of the data, $\log \mathcal{L}_{L-Log}(\mathbf{x})$.

Proof 1: From the *Lemma*, we have :

$$\begin{aligned} \log \mathcal{L}_{L-Log}(\mathbf{x}) &= \log \left(\sum\nolimits_{i=1}^{K} w_i \int p_{\theta_i}(\mathbf{x}|\mathbf{z}) p_{\theta_i}(\mathbf{z}) d\mathbf{z} \right) \\ &\geq \log \left(\sum\nolimits_{i=1}^{K} w_i \exp(\mathcal{L}^i_{VAE,\theta_i,\varepsilon_i}(\mathbf{x})) \right), \end{aligned} \tag{14}$$

When we optimize the mixture's objective function $\mathcal{L}_{L-MVAE}(\mathbf{x})$, the loss $\mathcal{L}^i_{VAE,\theta_i,\varepsilon_i}(\mathbf{x})$ corresponding to each component is increased. Then, the right hand side from (14) will be increased to approximate $\log \mathcal{L}_{L-Log}(\mathbf{x})$, given that the $\log \mathbf{u}$ is monotonically increasing for $\mathbf{u} \in [0, +\infty)$. However, any increase has an upper limit in $\mathcal{L}_{L-Log}(\mathbf{x})$, according to (14)

□

Theorem 2: Let us define $\log \mathcal{L}^{\star}_{L-Log}(\mathbf{x})$ as the log-likelihood of the objective function $\mathcal{L}_{L-MVAE}(\mathbf{x})$. Then we have $\log \mathcal{L}^{\star}_{L-Log}(\mathbf{x}) \leq \max\{\log p_{\theta_i}(\mathbf{x})\}$, $\forall i \in \{1, \ldots, K\}$ during the inference, where $\log p_{\theta_i}(\mathbf{x})$ represents the log-likelihood of a single VAE model, characterized by parameters θ_i.

Proof 2: Estimating the log-likelihood $\log \mathcal{L}^{\star}_{L-Log}(\mathbf{x})$ during the inference is intractable because the generation process of the mixture model involves an implicit component selection procedure. By considering (2), the log-likelihood $\log \mathcal{L}^{\star}_{L-Log}(\mathbf{x})$ is given by :

$$\log \mathcal{L}^{\star}_{L-Log}(\mathbf{x}) = \log \left(\sum_{i=1}^{K} w_i \mathcal{L}^i_{VAE,\theta_i,\varepsilon_i}(\mathbf{x}) \right), \tag{15}$$

where $\sum_{i=1}^{K} w_i = 1$, where the mixing parameters w_i are sampled from $Cat(\tau)$, $\tau = (\tau_1 \, \tau_2 \ldots \tau_K)^T$ where $\tau_i = \log p_{\theta_i}(\mathbf{x}) / \sum_{j=1}^{K} \log p_{\theta_j}(\mathbf{x})$. The marginal log-likelihood $\log p_{\theta_i}(\mathbf{x})$ for each VAE component is given by its approximation $\mathcal{L}^i_{VAE,\theta_i,\varepsilon_i}(\mathbf{x})$. The proposed model selects only the

most suitable expert VAE, indexed as h, which has the highest likelihood for the given data samples used during the training :

$$\begin{aligned}&\log \mathcal{L}^{\star}_{L-Log}(\mathbf{x}) \leq \mathcal{L}^{h}_{VAE_{\theta_h,\varepsilon_h}}(\mathbf{x}),\\&\log p_{\theta_h}(\mathbf{x}) \geq \mathcal{L}^{h}_{VAE_{\theta_i,\varepsilon_i}}(\mathbf{x}), i=1,\ldots,K,\end{aligned} \tag{16}$$

□

where $\log p_{\theta_h}(\mathbf{x}) = \max\{\log p_{\theta_i}(\mathbf{x})\},\ i = 1,\ldots,K$. This shows that during the testing stage, we can evaluate the data's log-likelihood, by using the proposed L-MVAE model. Unlike in the approach from [38], the proposed mixture system not only that can perform generation tasks but it also learns meaningful data representations across the domains assimilated during the lifelong learning process.

V. Defining the Lifelong MVAE for supervised, semi-supervised and unsupervised learning

In this section, we extend the mixture model for being used under various types of learning paradigms, such as : unsupervised, supervised, and semi-supervised.

Unsupervised disentangled representation learning. In order to encourage the latent representations to capture meaningful variations of data under unsupervised learning assumptions, we extend the disentangled representation learning approach from [32], which was built by using a similar concept to the β-VAE [29] used for modelling disentangled representations in single VAE models. We extend [32] to be used for the mixture objective function by replacing (2) with the following loss function :

$$\begin{aligned}L^{US}_{L-MVAE,\tilde{\varepsilon},\tilde{\theta}}(\mathbf{x}) = \sum_{i=1}^{K} w_i \Big(-\gamma|D_{KL}(q_{\varepsilon_i}(\mathbf{z}_i|\mathbf{x})||p(\mathbf{z}_i)) - C| \\ +\mathbb{E}_{\mathbf{z}_i\sim q_{\tilde{\varepsilon_i}}(\mathbf{z}_i|\mathbf{x})}\log p_{\tilde{\theta_i}}(\mathbf{x}|\mathbf{z}_i)\Big)\end{aligned} \tag{17}$$

where we have $\sum_{i=1}^{K} w_i$. The first term represents the Kullback-Leibler (KL) divergence associated with the output of each VAE decoder, by considering the disentanglement among the latent space variables, weighted by w_i, while the last term is associated with the log-likelihood of the data reconstruction by each mixture's encoder. The parameters associated with the disentanglement are set similarly to those from [29]: C is linearly increasing during the training, starting from a low value, while γ defines the contribution of this modified KL term to the objective function. $\tilde{\varepsilon} = \{\tilde{\varepsilon}_i|i = 1,\ldots,K\}$ and $\tilde{\theta} = \{\tilde{\theta}_i|i = 1,\ldots,K\}$ represent the parameters for all encoders and decoders of the mixture and of the individual components, respectively.

Lifelong supervised learning. We consider that the given data $\{\mathcal{X}|\mathbf{x}_i \in \mathcal{X}, i = 1,\ldots,N\}$ is labelled $\{\mathcal{Y}|\mathbf{y}_i \in \mathcal{Y}, i = 1,\ldots,N\}$, within a supervised learning framework. When considering a single VAE component we define a latent generative variable model $p_\theta(\mathbf{x},\mathbf{z},\mathbf{d}) = p_\theta(\mathbf{x}|\mathbf{z},\mathbf{d})p(\mathbf{z},\mathbf{d})$, where $\mathbf{z}$ is the continuous latent variable and $\mathbf{d}$ is the latent variable associated with the discrete information, labels for example. Then we derive its corresponding ELBO, considering two distinct encoders, characterized by the parameters ε and ς for the discrete $\mathbf{z}$ and continuous $\mathbf{d}$ latent variables, respectively, as follows:

$$\begin{aligned}&\mathcal{L}^{S}_{\theta,\varepsilon,\varsigma}(\mathbf{x}) = \mathbb{E}_{q_{\varepsilon,\varsigma}(\mathbf{z},\mathbf{d}|\mathbf{x})}\log\left[\frac{p_\theta(\mathbf{x},\mathbf{z},\mathbf{d})}{q_{\varepsilon,\varsigma}(\mathbf{z},\mathbf{d}|\mathbf{x})}\right]\\&= \mathbb{E}_{q_{\varepsilon,\varsigma}(\mathbf{z},\mathbf{d}|\mathbf{x})}\log\left[\frac{p_\theta(\mathbf{x}|\mathbf{z},\mathbf{d})p(\mathbf{z})p(\mathbf{d})}{q_\varepsilon(\mathbf{z}|\mathbf{x})q_\varsigma(\mathbf{d}|\mathbf{x})}\right]\\&= \mathbb{E}_{q_{\varepsilon,\varsigma}(\mathbf{z},\mathbf{d}|\mathbf{x})}\log[p_\theta(\mathbf{x}|\mathbf{z},\mathbf{d})] + \mathbb{E}_{q_{\varsigma,\varepsilon,\delta}(\mathbf{z},\mathbf{d}|\mathbf{x})}\log\left[\frac{p(\mathbf{z})}{q_\varepsilon(\mathbf{z}|\mathbf{x})}\right]\\&+ \mathbb{E}_{q_{\varepsilon,\varsigma}(\mathbf{z},\mathbf{d}|\mathbf{x})}\log\left[\frac{p(\mathbf{d})}{q_\varsigma(\mathbf{d}|\mathbf{x})}\right] = \mathbb{E}_{q_{\varepsilon,\varsigma}(\mathbf{z},\mathbf{d}|\mathbf{x})}\log[p_\theta(\mathbf{x}|\mathbf{z},\mathbf{d})]\\&- D_{KL}[q_\varepsilon(\mathbf{z}|\mathbf{x})||p(\mathbf{z})] - D_{KL}[q_\varsigma(\mathbf{d}|\mathbf{x})||p(\mathbf{d})].\end{aligned} \tag{18}$$

This equation uses the assumption that $\mathbf{z}$ is independent from $\mathbf{d}$, which is guaranteed by using two separate inference models $q_\varepsilon(\mathbf{z}|\mathbf{x})$ and $q_\zeta(\mathbf{d}|\mathbf{x})$ for modelling $\mathbf{z}$ and $\mathbf{d}$. Eq. (18) corresponds to the ELBO for one of the components in our mixture model. We then define the mixture's objective function by evaluating a sum over all individual components ELBO's, each multiplied by its associated mixing coefficient, resulting in :

$$\begin{aligned}\mathcal{L}^{S}_{L-MVAE}(\mathbf{x}) = \sum_{i=1}^{K} w_i(\mathbb{E}_{q_{\varepsilon_i,\varsigma_i}(\mathbf{z},\mathbf{d}|\mathbf{x})}\left[\log p_\theta\left(\mathbf{x}|\mathbf{z},\mathbf{d}\right)\right] \\ -D_{KL}[q_{\varepsilon_i}(\mathbf{z}|\mathbf{x})||p(\mathbf{z})] - D_{KL}[q_{\varsigma_i}(\mathbf{d}|\mathbf{x})||p(\mathbf{d})]),\end{aligned} \tag{19}$$

where $\varepsilon = \{\varepsilon_i|i = 1,\ldots,K\}$ and $\varsigma = \{\varsigma_i|i = 1,\ldots,K\}$, represent the parameters for the encoders modelling continuous $\mathbf{z}$, and discrete $\mathbf{d}$, latent variables, for each mixture' component. We call each $q_{\xi_i}(\mathbf{d}|\mathbf{x})$ as the class-specific encoder. The last two terms from (19) represent the KL divergences between the posterior and prior distributions for the variables $\mathbf{z}$ and $\mathbf{d}$, associated to the continuous and discrete latent spaces, respectively.

For the discrete variables we consider sampling using the Gumbel-Max trick for $q_\varsigma(\mathbf{d}|\mathbf{x})$, as in [46], [47], in order to produce differentiable discrete variables. We implement $q_\varsigma(\mathbf{d}|\mathbf{x})$ by using a neural network of parameters ς in which the last layer implements the softmax function producing the probability vector $\mathbf{d}' = (d'_1, d'_2, \ldots, d'_K)$, while the sampling process is defined by :

$$d_k = \frac{\exp((\log d'_k + g_k)/T)}{\sum_i^K \exp((\log d'_i + g_i)/T)} \tag{20}$$

where d_k is the sampled value and d'_k is its probability. g_k is sampled from the $\mathrm{Gumbel}(0,1)$ distribution. The sample vector $\mathbf{d}$ is treated as a continuous approximation of the categorical representation (one-hot vector). This sampling process is incorporated into both generation and inference stages. For enforcing the discrete latent variables $\mathbf{d}$ to capture discriminative information such as the data type, we introduce a mixture of cross-entropy loss $\mathcal{L}_{S-Mix,\tilde{\varsigma}}(\mathbf{x})$:

$$\mathcal{L}_{S-Mix,\varsigma}(\mathbf{x}) = E_{(\mathbf{x},\mathbf{y})\sim(\mathcal{X},\mathcal{Y})}\sum_{i=1}^{K} w_i\eta(q_{\varsigma_i}(\mathbf{d}|\mathbf{x}),\mathbf{y}), \tag{21}$$

where we incorporate the individual VAE components cross-entropy loss $\eta(\cdot,\cdot)$ weighted by the associated mixing coefficients, characterizing the encoders specific to learning the

discrete variables, into a single objective function for the mixture system and $\varsigma = \{\varsigma_1, \ldots, \varsigma_k\}$. The pseudocode for the supervised learning is provided in Algorithm 1 where we firstly optimize the parameters of the model by using Eq. (19) and Eq. (21) at each iteration.

Lifelong semi-supervised learning. We also consider the semi-supervised learning context [48] for the proposed L-MVAE model. Under the semi-supervised setting, we only have a small subset of labeled observations $\mathbf{x}$, with labels $\mathbf{y}$ with and a much larger number of unlabeled data samples $\hat{\mathbf{x}}$ for each learning task, where $\hat{\mathbf{x}}, \mathbf{x} \in \mathcal{X}$. In semi-supervised learning the unlabelled data samples are then associated based on their statistical consistency with the labelled data. Labels $\mathbf{y}$ replace the discrete variables $\mathbf{d}$, used for supervised learning, during the decoding process, and the objective function is:

$$\mathcal{L}^{SemS}_{Mix,\tilde{\theta},\tilde{\varepsilon},\tilde{\varsigma}}(\hat{\mathbf{x}}) = \sum_{i=1}^{K} w_i \Big(\mathbb{E}_{q_{\varepsilon_i,\varsigma_i}(\mathbf{z},\mathbf{y}|\mathbf{x}')} \left[\log p_\theta\left(\mathbf{x}'|\mathbf{z},\mathbf{y}\right)\right] - D_{KL}[q_{\varepsilon_i}(\mathbf{z}|\hat{\mathbf{x}})||p(\mathbf{z})]\Big) \tag{22}$$

where $\mathcal{L}^{SemS}_{Mix,\tilde{\theta},\tilde{\varepsilon},\tilde{\varsigma}}(\hat{\mathbf{x}})$ is the loss function for the semi-supervised learning of the L-MVAE model, $\sum_{i=1}^{K} w_i = 1$, while $\tilde{\theta}$, $\tilde{\varepsilon}$ and $\tilde{\varsigma}$ represent the mixture's model parameters characterizing the decoders and the encoders specific to the continuous $\mathbf{z}$ and to the labels $\mathbf{y}$, respectively.

In addition to $\mathcal{L}^{SemS}_{Mix}(\hat{\mathbf{x}})$ from (22), we also optimize the parameters $\tilde{\varsigma}$ using the mixture cross-entropy $\mathcal{L}_{S-Mix}(\mathbf{x})$, similar to equation (21), used for supervised learning. For the unlabeled samples, missing labels are inferred by using Gumble-softmax based sampling in which the probability vector $\mathbf{d}'$ is sampled from the encoder, defined by $q_\varsigma(\mathbf{d}|\mathbf{x}')$. These resulting discrete variables are then used during the decoding. The final objective function for semi-supervised learning tasks is defined as:

$$\mathcal{L}^{SemS}_{L-MVAE}(\mathbf{x}) = \mathcal{L}^{SemS}_{Mix}(\hat{\mathbf{x}}) + \beta \mathcal{L}^{S}_{L-MVAE}(\mathbf{x}), \tag{23}$$

where the first term is given in (22), and β controls the importance of the loss associated to the supervised learning $\mathcal{L}^{S}_{L-MVAE}(\mathbf{x})$, which is defined in (19). We separately optimize the parameters of the model by using (23) and (21) during each iteration, similar to the supervised learning setting.

VI. Mixture Expansion Mechanism

A given mixture architecture has limits in its modelling capabilities. Such limits are especially exposed during the lifelong learning, when the model has to learn new tasks. In this section, we introduce a procedure for expanding the L-MVAE architecture in order to enhance the architecture ability to deal successfully with a growing number of tasks. Meanwhile we aim to use a minimal number of model parameters and optimize the training time for efficiently learning all tasks. We introduce a joint network by adding to the existing VAE component structure consisting in an encoder and a decoder, defined by the parameters θ'_1 and ε'_1, respectively, a sub-encoder and a sub-decoder, with parameters θ_S and ε_S, respectively. During the first task learning, we build the first mixture component based on this joint network. We use

Algorithm 1: Supervised training for the L-MVAE model.

Training phase :

While $T < taskCount^{\max}$ **do**
  Sample $X^T = \{x_1^T, x_2^T, ..., x_N^T\}$ from the T-th task
  Sample $Y^T = \{y_1^T, y_2^T, ..., y_N^T\}$ from the T-th task
  While $epoch < epoch^{\max}$ **do**
    While $batch < batch^{\max}$ **do** minibatch procedure
      $\mathbf{w} \sim \mathrm{Dir}(a_1, .., a_K)$ Sampling mixing weights
      Train all expert networks by optimizing $\mathbb{L}^S_{L-MVAE}$
      Train all class-specific encoders by optimizing $L_{S-Mix,\varsigma}$
    End
  End
$p(c_j) = 1 - \dfrac{\exp(-\mathbb{L}^j_{VAE}(\mathbf{x}_b)) + u\, c'_j}{\sum_{i=1}^{K} \exp(-\mathbb{L}^i_{VAE}(\mathbf{x}_b)) + u\, c'_i}$
$p(c_t = 1) = \max(p(c_t = 1), .., p(c_K = 1))$ find the maximum probability
$p(c_t = 1) = 1, p(c_t = 0) = 0$ normalize the maximum probability
$p(c_i = 1) = \begin{cases} 1 & c'_i = 1 \\ 0 & c'_i = 0 \end{cases}, i = 1, 2.., K, i \neq t$ normalize other probabilities
$a_i = \begin{cases} e & c_i = 1 \\ \dfrac{1 - eK'}{K - K'} & c_i = 0 \end{cases}$
End

Testing phase :

$v_i = \dfrac{\exp(-\frac{1}{\mathbb{L}^j_{VAE}(\mathbf{x})})}{\sum_{i=1}^{K} \exp(-\frac{1}{\mathbb{L}^i_{VAE}(\mathbf{x})})}, i = 1, .., K$
$\mathbf{w} \sim \textbf{Categorical}(v_1, .., v_K)$
Select an expert according to $\mathbf{w}$

1

$p_{\theta_1}(\mathbf{x}|\mathbf{z})$ and $q_{\varepsilon_1}(\mathbf{z}|\mathbf{x})$ to represent the decoder and encoder, respectively, where $\theta_1 = \{\theta_S, \theta'_1\}$ and $\varepsilon_1 = \{\varepsilon_S, \varepsilon'_1\}$. During the training we update both the shared parameter set $\{\theta_S, \varepsilon_S\}$ and the specific parameter set $\{\theta'_1, \varepsilon'_1\}$ when learning the first task. For the following task learning, the $\{\theta_S, \varepsilon_S\}$ parameters are fixed, while when a new component would be added, then only its corresponding specific parameter set $\{\theta'_2, \varepsilon'_2\}$ is updated using Eq. (1) considering the new task for training. In the following, we introduce a new mechanism for acquiring the knowledge corresponding to a new task during the lifelong learning, by either updating an existing mixture component, or adding a new component and training its parameters. We show the process of the proposed expansion mechanism in Fig. 2.

In order to allow a single component to learn several similar tasks, we introduce a similarity measure between the probabilistic representation associated with a new task and the information recorded by each learnt mixture component. If the new task is novel enough relative to the already learnt knowledge, the mixture model will add a new component in order to learn the new task. Otherwise the training algorithm will select and update the most appropriate component. Let us consider that the mixture model has K components after learning the $j-1$-th task. We would like to evaluate the novelty of the j-th task by comparing the knowledge acquired by each of the

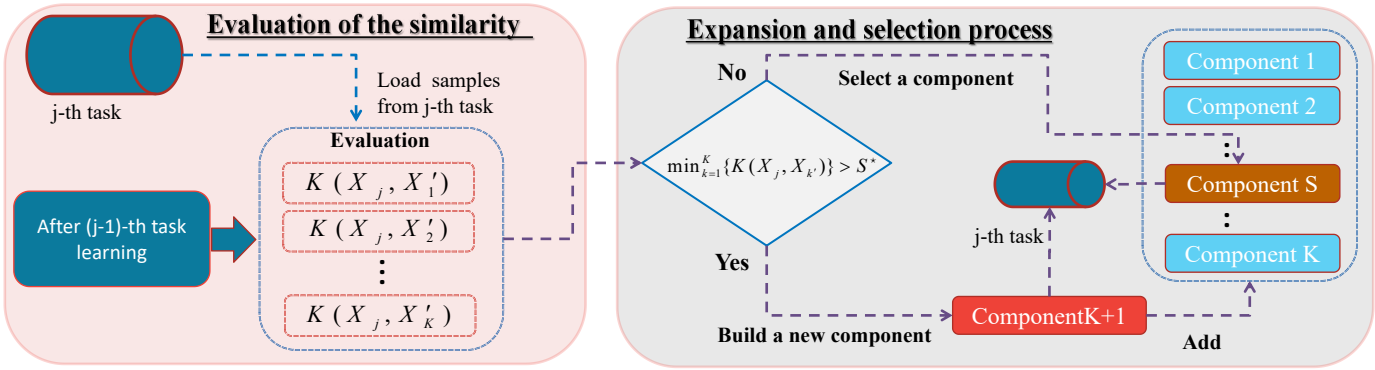


Fig. 2. The illustration of the component expansion mechanism. Once the $(j-1)$-th task was finished, we collect samples from the j-th database and then evaluate the compatibility between the data from j-th database and the probabilistic representation of each component by using Eq. (24). Then Eq. (25) is used to either select an existing component of the mixture or to expand the network by adding a new component. The added component during the j-th task learning is marked in a red rectangle. The testing phase for the expansion mechanism is identical to the one shown in Fig. 1.

K components and the probabilistic representation of the j-th task. We consider a probabilistic representation of the j-th task by randomly selecting a set $\{\mathcal{X}_j|\mathbf{x}_{j,i} \in \mathcal{X}_j; i = 1,\dots,N_j\}$ where in the experiments we consider $N_j = 1000$ samples. The probabilistic representation of the knowledge acquired by each expert is represented by its ability to generate specific data. Thus, we generate for each expert $k = 1,\dots,K$, a dataset $\{\mathcal{X}'_k|f_{\theta_k}(\mathbf{x}_{j,l}) \in \mathcal{X}'_k; l = 1,\dots,N'_k\}$, where in the experiments we consider $N'_l = 1000$, for $l = 1,\dots,K$ and j represents the database used for sampling the original data $\mathbf{x}_{j,l}$. We define as statistical similarity the following L2 distance between all data from each two datasets :

$$\mathcal{K}(\mathcal{X}_j,\mathcal{X}'_k) = \frac{1}{N_j}\frac{1}{N'_k}\sum_{i=1}^{N_j}\sum_{l=1}^{N'_k}\|\mathbf{x}_{j,i} - f_{\theta_k}(\mathbf{x}_{j,l})\|\,, \qquad (24)$$

for all $k = 1,\dots,K$. A new expert $K+1$ is built in the mixture model when none of the experts is able to generate data similar to those from the new dataset, according to the following criterion :

$$\min_{k=1}^{K}\{\mathcal{K}(\mathcal{X}_j,\mathcal{X}'_k)\} > S^\star\,, \qquad (25)$$

where $S^\star$ is a threshold defining the level of novelty in the knowledge acquired by each expert. The parameter set of the new expert is $\{\varepsilon_S,\varepsilon'_{K+1},\theta_S,\theta'_{K+1}\}$ and where only the parameters $\varepsilon'_{K+1},\theta'_{K+1}$ are trained according to the objective defined in (1). If (25) is not fulfilled then the most suitable component is chosen :

$$L = \arg\min_{k=1}^{K}\mathcal{K}(\mathcal{X}_j,\mathcal{X}'_k) \qquad (26)$$

and its encoder and decoder parameters $\{\theta'_L,\varepsilon'_L\}$ are updated. We call the proposed expansion mechanism with the mixture model as L-MVAE dynamic (L-MVAE-Dyn). By considering a fixed component of the model, made up of the sub-decoder and sub-encoder of parameters $\{\varepsilon_S,\theta_S\}$ we ensure a common heritage knowledge for all tasks, corresponding to a set of features shared by the data from several databases. When learning each task, we add an additional set of parameters corresponding to characteristic information for each database. This procedure ensures a fast and efficient learning procedure, while maintaining the required set of parameters to a minimum, when learning a sequence of tasks.

VII. Experiments

We evaluate the performance of the proposed L-MVAE system when learning several tasks, and in several applications including classification, reconstruction and disentangled representation learning. Afterwards, we assess how L-MVAE is used for semi-supervised and unsupervised learning tasks in the context of lifelong learning. The implementations are done using the TensorFlow framework.

A. Supervised learning

We select four datasets for the lifelong supervised training of L-MVAE: MNIST [49], Fashion [50], SVHN [51] and CIFAR10 [52]), called MFSC sequence. We estimate the average classification accuracy on all testing data samples across different domains during the lifelong training, and the results are provided in Fig. 3, where each task is trained for 10 epochs using Stochastic Gradient Descent (SGD). From these results we observe that each time when training with a new dataset, L-MVAE maintains almost its full performance on the previously learned tasks. For comparison in the same plot we show the results obtained by Deep Generative Replay (DGR) [14] which has a significant performance drop on the previously learnt tasks, when training with a new dataset, as it can be observed from Fig. 3.

In Table I we provide the classification accuracy for the lifelong learning of the MFSC sequence of databases. When all these databases are used jointly for training, within an approach named "JVAE", we achieve good accuracy results on simple datasets such as MNIST and Fashion, but the performance drops on the datasets containing more complex images. "Transfer" represents training a single classifier on a sequence of tasks without using the generative replay mechanism. We can observe that the "Transfer" approach only achieves good results on the latest task and completely forgets any previously learnt knowledge. L-MVAE-S is the mixture model sharing

Methods	MNIST	Fashion	SVHN	Cifar10
L-MVAE	**97.97**	90.02	**87.00**	**69.32**
L-MVAE-S	96.18	**91.64**	86.20	66.94
JVAEs	97.72	88.47	61.87	52.69
Transfer	5.28	5.23	13.82	68.67
DGR [14]	90.20	72.64	62.44	56.43
LGM [15]	61.06	63.57	64.21	56.84
CURL [17]	91.46	74.29	66.78	59.46

TABLE I
CLASSIFICATION ACCURACY WHEN CONSIDERING THE LIFELONG LEARNING OF MNIST, FASHION, SVHN AND CIFAR10 DATABASES. MFSC AND CSFM DENOTE THE ORDER OF THE DATABASES USED FOR THE LIFELONG TRAINING.

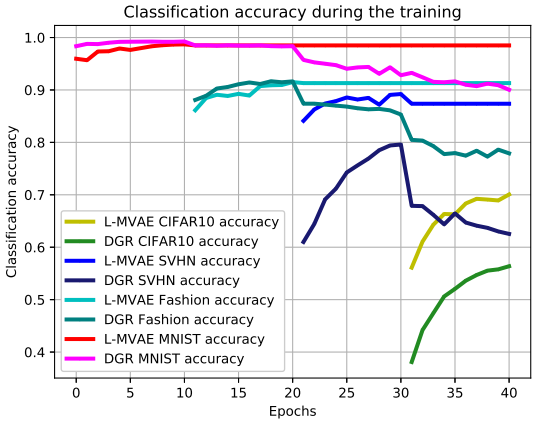


Fig. 3. Classification accuracy on all testing data samples across several domains during the lifelong learning.

the parameters of the decoder with all experts. Although L-MVAE-S uses fewer parameters than L-MVAE, it still provides very good results. The generative replay based methods used for comparison, Lifelong generative modeling (LGM) [15], DGR [14] and Continual Unsupervised Representation Learning (CURL) [17] display a performance fall on all tasks, which is mainly caused by the quality of generative replay samples. Since we evaluate a sequence of different domains, the generative replay based methods tend to forget the initial learnt tasks.

B. Semi-supervised learning

We investigate the performance of the L-MVAE system in semi-supervised tasks. We randomly select 1,000 training images from the MNIST as the labelled dataset, and 10,000 labelled images from each of the datasets: Fashion, SVHN and Cifar10. The remaining unlabelled samples are used as the training set. We train the L-MVAE system on both the labelled and unlabelled samples under the MNIST, Fashion, SVHN and Cifar10 lifelong learning, according to Eq. (23) where we set β= 0.5. The results are shown in Table II, where we use '*' to denote the model to be learned under the lifelong setting. It can be observed that the proposed model almost achieves better results than CURL [17] in each task learning and even achieves competitive results when comparing with the current state of the art semi-supervised methods trained only on a single dataset, such as CAE [53], M1 [53], M1+M2 [53] and Semi-VAE [48].

C. Unsupervised lifelong reconstruction and interpolation

In the following, L-MVAE model is used in unsupervised applications, where there are no data labels. We train the proposed mixture system with four components ($K = 4$) under the MNIST, Fashion, SVHN and Cifar10 (MFSC) as well as when using CelebA, CACD, 3D-chairs and Omniglot (CCDO) lifelong learning settings. The original images for MFSC and for CCDO databases are provided in Figures 4 a-d and 5 a-d, respectively. The image reconstruction results corresponding to these images, following the lifelong learning, are shown in Figures 4 e-h, and Figures 5 e-h, respectively. These results show that the proposed L-MVAE mixture system is able to make accurate inference across several different domains. We also explore performing interpolations in the latent space of multiple domains. When interpolating between two latent vectors, we initially select the most relevant expert, according to the selection strategy from Section III-C, and then infer the latent variables using the selected inference model. The selected decoder will then recover images from the interpolated latent variable space. We present the interpolation results in Figures 6 a-d, for images from CelebA, CACD, 3D-chairs and Omniglot databases. The proposed model achieves continuity in the latent space as reflected in the generated images derived by each expert, according to these results.

D. Disentangled representation learning

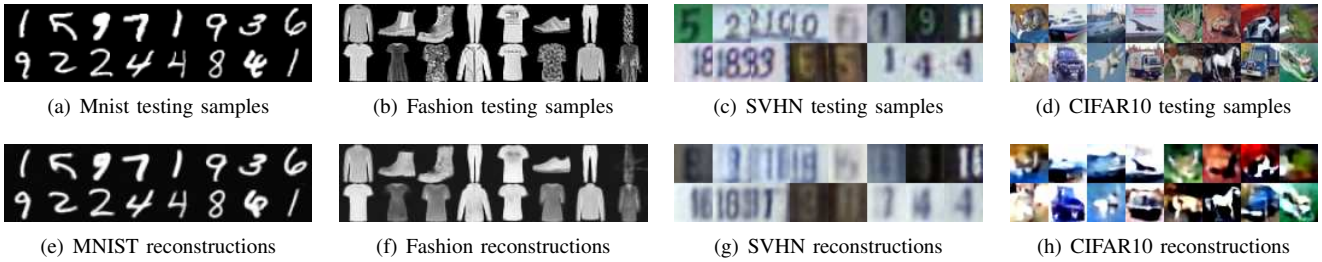
We train L-MVAE system under the CelebA, CACD, 3D-chairs and Omniglot lifelong learning by using the disentangled loss function from Eq. (17) where C is increased from a very small value to 25.0 during the training and we set γ= 4. After the training, the L-MVAE system firstly chooses the most relevant expert and then a single latent variable inferred by the selected expert is changed from -3 to 3 while fixing the other latent variables. The results are shown in Figures 7 and 8. From Figures 7 a-d we observe that the proposed L-MVAE approach can discover four disentangled representations for CelebA by changing: age, hair style, illumination and face orientation. From Figures 8 a-c we can observe that we can change chair size, style and orientation.

E. Visual quality evaluation for the generated images

For assessing the representation learning ability under the lifelong setting, we evaluate the negative log-likelihood (NLL), representing the reconstruction error plus the KL divergence term, as well as the inception score (IS) [54] for the reconstructed images from the testing set. First, we train various models under the MNIST, Fashion, SVHN and CIFAR10 (MFSC) lifelong learning setting, by considering 100 epochs for learning each task. The results for MFSC and when considering the learning of the databases in reversed order as CSFM, are provided in Tables III and IV for the average NLL and the average reconstruction error, respectively, when comparing against CURL [17], LGM [15] and with JVAE

Dataset	L-MVAE*	CURL* [17]	CAE [53]	M1 [53]	M1+M2 [53]	Semi-VAE [48]
MNIST	4.95	14.67	4.77	4.24	2.40	2.88
Fashion	**16.93**	64.28	/	/	/	/
SVHN	**23.00**	66.39	/	/	/	/
CIFAR10	**48.32**	43.57	/	/	/	/

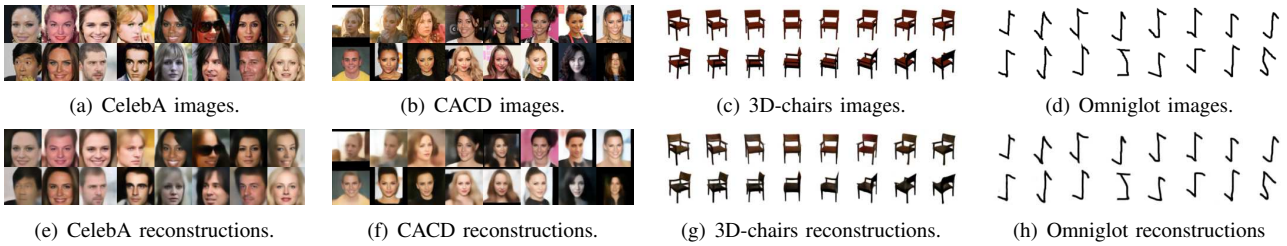
TABLE II
SEMI-SUPERVISED CLASSIFICATION ERRORS ON MNIST UNDER THE LIFELONG LEARNING FOR MNIST, FASHION, SVHN, AND CIFAR10 DATABASES.



(a) Mnist testing samples (b) Fashion testing samples (c) SVHN testing samples (d) CIFAR10 testing samples

(e) MNIST reconstructions (f) Fashion reconstructions (g) SVHN reconstructions (h) CIFAR10 reconstructions

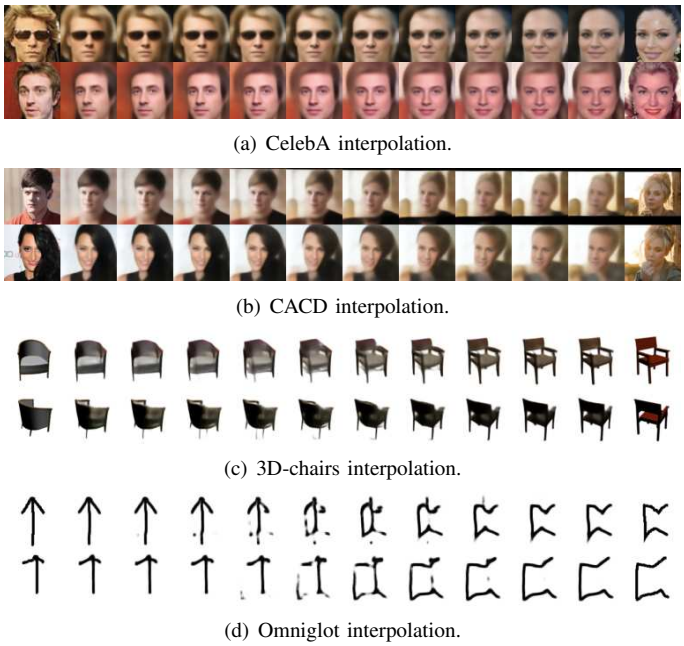
Fig. 4. Reconstruction results by L-MVAE after the lifelong learning of MNIST, Fashion, SVHN and CIFAR10 (MFSC).



(a) CelebA images. (b) CACD images. (c) 3D-chairs images. (d) Omniglot images.

(e) CelebA reconstructions. (f) CACD reconstructions. (g) 3D-chairs reconstructions. (h) Omniglot reconstructions

Fig. 5. Reconstruction results by L-MVAE after the lifelong learning of the CelebA, CACD, 3D-chairs and Omniglot (CCDO).



(a) CelebA interpolation.

(b) CACD interpolation.

(c) 3D-chairs interpolation.

(d) Omniglot interpolation.

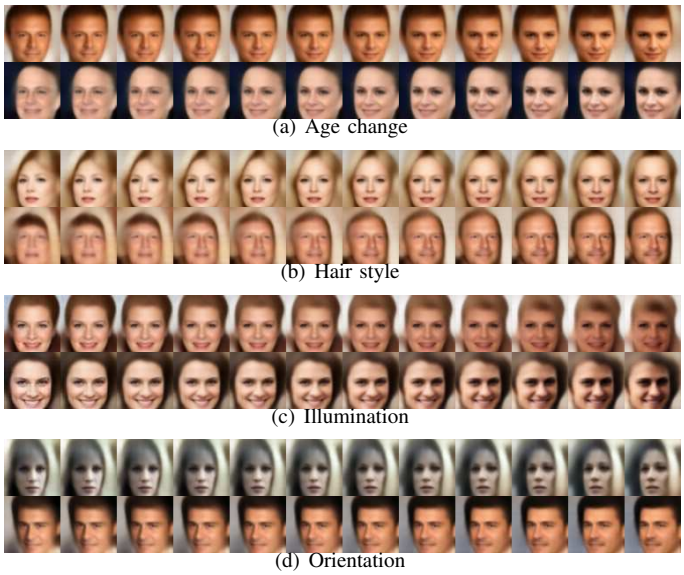
Fig. 6. Interpolation results after the lifelong learning of CelebA, CACD, 3D-chairs and Omniglot databases. The extreme images on each row are real, while those in between are generated by L-MVAE as interpolations exploring the latent space.

Dataset	L-MVAE	CURL [17]	JVAE	Lifelong
MNIST	**48.66**	272.95	195.79	MFSC
Fashion	**53.02**	190.36	173.64	MFSC
SVHN	**40.39**	127.64	208.19	MFSC
Cifar10	**752.91**	1409.74	1840.40	MFSC
MNIST	**43.26**	64.99	/	CSFM
Fashion	**44.15**	131.13	/	CSFM
SVHN	**39.46**	278.01	/	CSFM
Cifar10	**778.17**	2406.13	/	CSFM

TABLE III
NEGATIVE LOG-LIKELIHOOD (NLL) ESTIMATION FOR ALL TESTING IMAGES FOR THE LIFELONG LEARNING OF THE PROBABILISTIC REPRESENTATIONS FOR MNIST, FASHION, SVHN AND CIFAR10 DATABASES.

Dataset	L-MVAE	CURL [17]	JVAE	Lifelong
MNIST	**25.83**	167.09	68.59	MFSC
Fashion	**34.09**	139.91	141.16	MFSC
SVHN	**27.20**	84.45	295.94	MFSC
Cifar10	**631.14**	1225.41	1792.08	MFSC
MNIST	**20.09**	33.55	/	CSFM
Fashion	**26.46**	252.53	/	CSFM
SVHN	**25.81**	110.21	/	CSFM
Cifar10	**653.39**	2340.37	/	CSFM

TABLE IV
IMAGE AVERAGE RECONSTRUCTION ERROR FOR AFTER THE LIFELONG LEARNING OF MNIST, FASHION, SVHN AND CIFAR10 DATABASES.



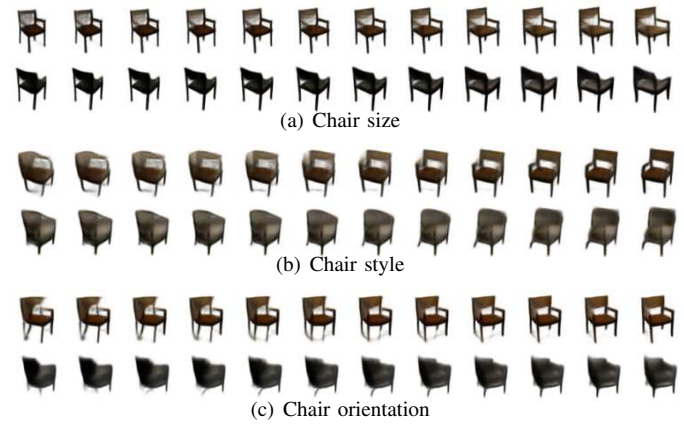
(a) Age change

(b) Hair style

(c) Illumination

(d) Orientation

Fig. 7. Disentangled results after the Lifelong training with CelebA, CACD, 3D-chairs and Omniglot databases.



(a) Chair size

(b) Chair style

(c) Chair orientation

Fig. 8. Disentangled results after the Lifelong training with CelebA, CACD, 3D-chairs and Omniglot databases.

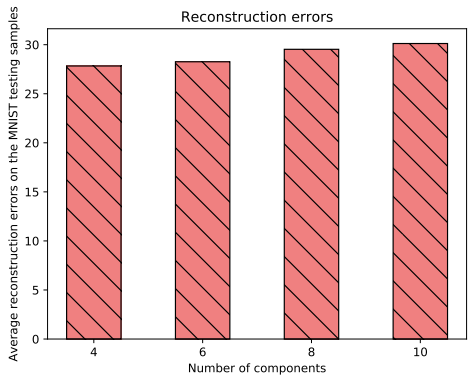


Fig. 9. Reconstruction errors when changing the number of experts.

Dataset	L-MVAE	CURL [17]	LGM [15]
CIFAR10	**4.82**	3.85	3.23
CIFAR100	**4.78**	3.56	3.64
ImageNet	**5.01**	3.72	3.47

TABLE V
THE IS SCORE FOR 5,000 TESTING IMAGES UNDER THE LIFELONG LEARNING OF IMAGENET, CIFAR100, CIFAR10 AND MNIST DATABASES.

Dataset	L-MVAE	CURL [17]
CIFAR10	**4.73**	3.59
CIFAR100	**4.13**	3.47
ImageNet	**5.52**	3.56

TABLE VI
THE IS SCORE FOR GENERATED IMAGES AFTER THE LIFELONG LEARNING OF CIFAR100, CIFAR10 AND IMAGENET DATABASES.

(when training with all databases at once). These results show that the proposed approach achieves the best results.

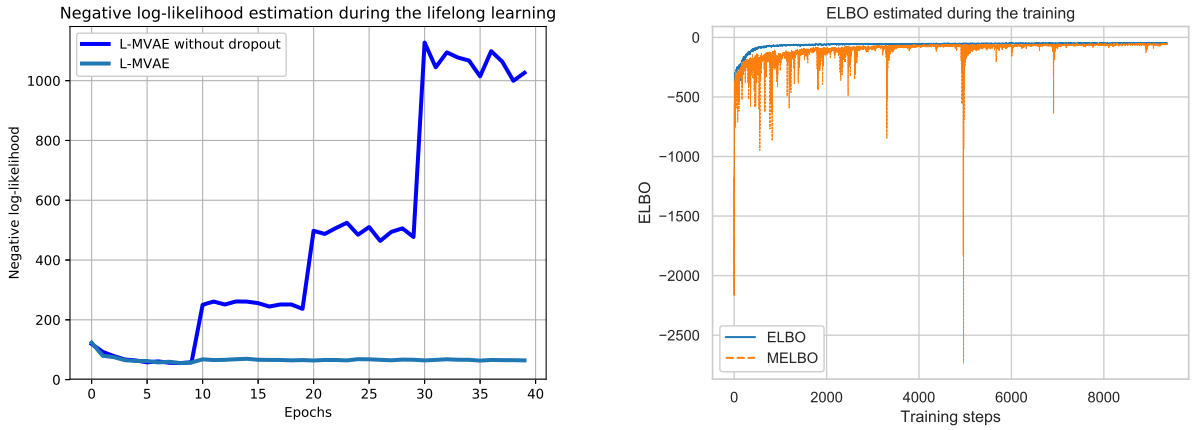
We also consider the lifelong training for ImageNet, CIFAR100, CIFAR10 and MNIST. After the training, we choose 5,000 images for testing from CIFAR10, CIFAR100 and ImageNet, respectively, and we the IS score of the reconstructed images is provided in Table V when comparing with CURL [17] and LGM [15]. Then we train various models under the CIFAR100, CIFAR10 and ImageNet lifelong learning and we provide the results in Table VI. These results show that the proposed model still provide the best performance even when learning a sequence of several databases containing complex and diverse images.

F. Ablation study

We perform an ablation study to investigate the performance when we change the configuration of the mixture model. We train L-MVAE with 4, 6, 8, 10 components under the MNIST, Fashion, SVHN and Fashion lifelong learning setting. We plot the average reconstruction errors on all MNIST testing samples in Figure 9. The results show that the number of components does not affect the performance too much and this is why we use $K = 4$ components in the experiments.

We also investigate the performance of the proposed model when not properly estimating the Dirichlet parameters, where the weights w_i, $i = 1, \ldots, 4$ are sampled from the same distribution. We call the model that does not have a component selection as "L-MVAE without dropout". We train this model under the same lifelong task learning as above and then we plot the NLL results on the first task (MNIST) in Fig. 10 a, where it can be observed that this model would lose its performance during the following tasks when not following the proposed dropout approach described in Section III-C. The reason for such results is that all experts are activated during the learning of the following tasks if the Dirichlet parameters are not changed accordingly.

In the following experiments we provide empirical evidence for the theory analysis results. We train the L-MVAE model under the lifelong learning of MNIST, Fashion, SVHN and CIFAR10 where we evaluate MELBO, from Eq. (2), for each



(a) NLL estimation when selecting the L-MVAE mixture components during the training and without component dropout.

(b) MELBO and ELBO estimation.

Fig. 10. Analysis results for the L-MVAE framework.

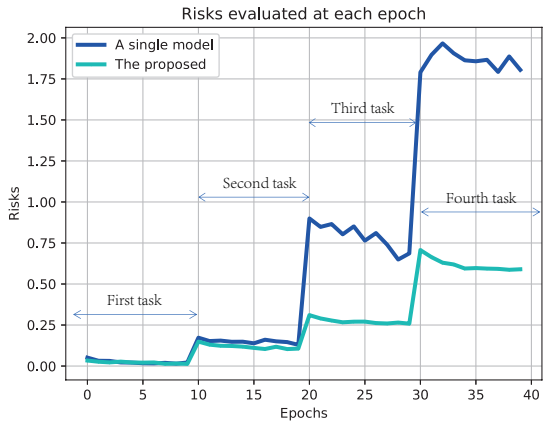


Fig. 11. The risks evaluated at each epoch under the MNIST, Fashion, SVHN and CIFAR10 lifelong learning.

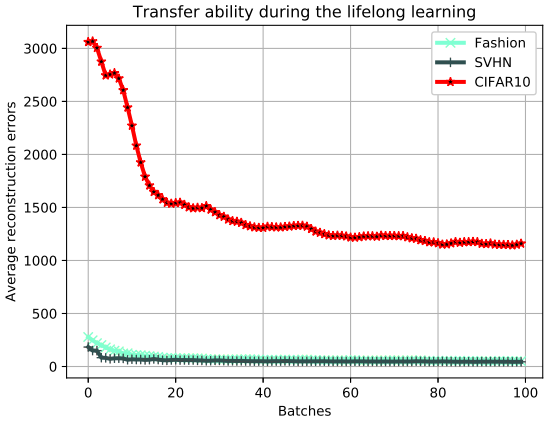


Fig. 12. The transfer learning ability $s(\theta_k, j)$, defined in Eq. (27), for the proposed model under the MNIST, Fashion, SVHN and CIFAR10 lifelong learning.

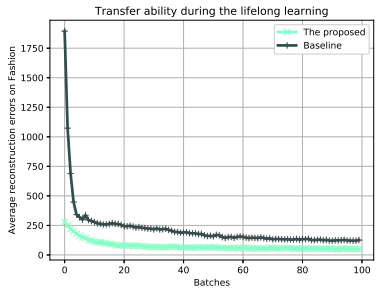
training step in the first task and the results are shown in Fig. 10 b where we also consider a single VAE model with optimal ELBO when training on MNIST (MELBO and ELBO are estimated by using the negative reconstruction errors and the KL divergence). From these results, MELBO is always bounded by this optimal ELBO and still represents a lower bound on the sample log-likelihood since $\log p(\mathbf{x}) \geq$ ELBO, according to *Theorem 2* from Section IV. In addition, we also train a single expert with GRM and a mixture model with 4 experts under MNIST, Fashion, SVHN and CIFAR10 lifelong learning. We consider the classification error rate as the risk of a model evaluated on the testing set and the accumulated errors are calculated by summing up the risk on the testing sets of all learnt tasks. We consider 10 epochs for each task training and plot the results in Fig. 11. We observe that when considering a single model tends to have a large risk while increasing the learning of additional tasks. The proposed L-MVAE mixture model always has a lower risk than a single VAE.

Model	Lifelong	*IS*
MIX+Wasserstein GAN in [55]	No	4.04
DCGAN [56] in [57]	No	4.89
ALI [58] in [57]	No	4.97
PixelCNN++ [59] in [60]	No	5.51
WGAN in [55]	No	3.82
L-MVAE-MFSC	Yes	5.77
L-MVAE-CSFM	Yes	5.322

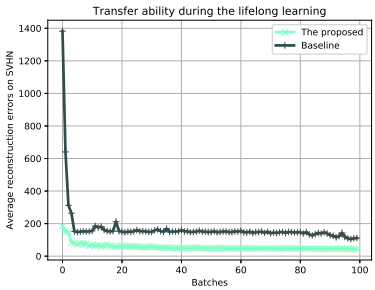
TABLE VII
INCEPTION SCORE (IS) EVALUATED ON CIFAR10.

G. Transfer metric and transfer learning

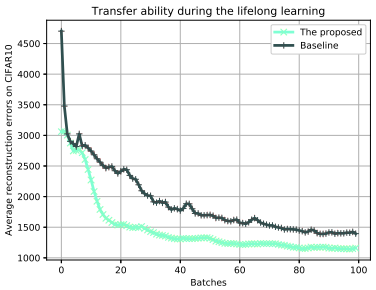
In this section, we evaluate how quickly the proposed L-MVAE approach learns a new task when presented with a new database for training. We can interpret the learning of the probabilistic representation of a new dataset by L-MVAE, as q knowledge transfer process from one domain to another. This



(a) Fashion



(b) SVHN

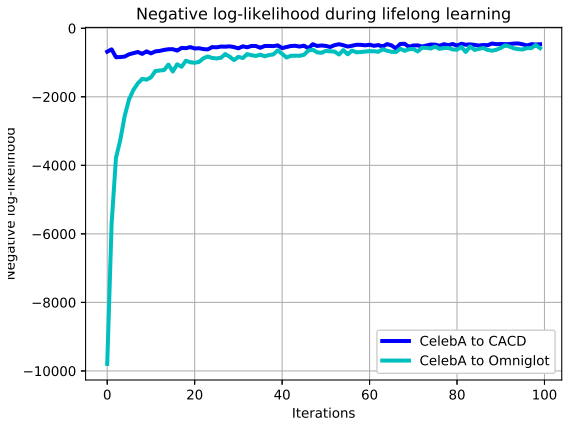


(c) Cifar10

Fig. 13. Average reconstructions errors during the lifelong learning.

	MSE		**SSMI**		**PSNR**	
Datasets	L-MVAE-Dynamic	BatchEnsemble [61]	L-MVAE-Dynamic	BatchEnsemble [61]	L-MVAE-Dynamic	BatchEnsemble [61]
MNIST	20.45	19.06	0.91	0.92	22.24	22.64
Fashion	35.55	179.72	0.74	0.27	19.45	12.23
SVHN	31.78	130.63	0.63	0.35	15.37	9.05
CIFAR10	853.42	846.52	0.34	0.36	17.28	17.34
Average	**235.30**	293.98	**0.66**	0.47	**18.59**	15.31

TABLE VIII
THE RECONSTRUCTION PERFORMANCE OF VARIOUS MODELS UNDER THE MFSC LIFELONG LEARNING.



(a) Fashion

Fig. 14. The negative log-likelihood (NLL) evaluated when learning the second task.

Dataset	L-MVAE-Dynamic	BatchEnsemble [61]
MNIST	99.18	99.34
Fashion	90.46	88.52
SVHN	78.63	74.85
CIFAR10	63.71	54.80
Average	**82.99**	79.38

TABLE IX
THE CLASSIFICATION ACCURACY OF VARIOUS MODELS UNDER MFSC LIFELONG LEARNING.

Dataset	L-MVAE-Dynamic	BatchEnsemble [61]
MNIST	89.40	99.17
SVHN	86.54	70.59
Fashion	90.44	85.72
IFashion	89.35	85.47
IMNIST	99.27	98.84
RFashion	91.74	87.39
CIFAR10	67.91	54.69
Average	**87.81**	83.13

TABLE X
THE CLASSIFICATION ACCURACY OF VARIOUS MODELS UNDER MSFIIRC LIFELONG LEARNING.

results in mixing the information being learnt by the expert from the new database with the information already stored in the networks' parameters, corresponding to the previously learnt tasks. In this paper, we propose a new metric, assessing the ability for transferring information during the lifelong learning when learning each new task. Considering a batch of N_j images from the given j-th database, we have the following measure:

$$s(\theta_k, j) = \frac{1}{N_j} \sum_{i=1}^{N_j} \delta(\mathbf{x}_{i,j}, f_{\theta_k}(\mathbf{x}_{i,j})) \tag{27}$$

where $s(\theta_k, j)$ is the performance score of the k-th mixture component of parameters θ_k for the j-th task, and $\{\mathbf{x}_{i,j} \in \mathcal{X}_j | i = 1, \ldots, N_j\}$ represents a given batch of images sampled from the j-th database, and $\delta(\cdot, \cdot)$ is the performance metric, considered as either the Mean Square Error (MSE), or it can be the classification accuracy, depending on the application of each task. $f_{\theta_k}(\mathbf{x}_{i,j})$ represents the image reconstructed by the L-MVAE model considering the given batch of images

$\mathbf{x}_{i,j}$ corresponding to the j-th task. The proposed metric can measure the training efficiency when a model is trained with a new task, representing the information transfer ability of the model when learning new tasks.

We train the L-MVAE model under the MNIST, Fashion, SVHN and CIFAR10 lifelong learning setting. The transfer ability during the lifelong learning is evaluated in Fig. 12, by considering the MSE as $\delta(\cdot,\cdot)$ in equation (27). These results show that the proposed approach converges quickly when learning the probabilistic representation of a new database. The baseline is considered to be our model trained on a single dataset, MNIST. The average reconstruction errors, calculated using equation (27) are provided in Figures 13 a-c for Fashion, SVHN and CIFAR10 databases, respectively. We observe that the proposed approach adapts quickly to a new task when compared to the baseline. We further investigate the difference of the transfer ability when considering learning the tasks in a different order. We train our model under the CelebA to CACD and CelebA to Omniglot, respectively. Then we measure the negative log-likelihood of the model for the second task and the results are presented in Fig. 14. It can be observed that learning CACD as the prior task can significantly accelerate the convergence when the future task shares similar visual concepts to the prior task.

H. Studying the over-regularization factors during training

In this section, we discuss the over-regularization problem in the proposed L-MVAE mixture model. A strong penalty on the KL divergence term in the VAE framework [28] can allow the variational distribution to match the prior distributions exactly, so $D_{KL}(q(\mathbf{z}|\mathbf{x})||p(\mathbf{z}))=0$. However, this may lead to a poor representation of the underlying data structure for $q(\mathbf{z}|\mathbf{x}) = p(\mathbf{z})$. To solve this problem, we implement each expert by using β-VAE [29], which includes a penalty term $\beta^\star$ on KL divergence, expressed as :

$$\log p(\mathbf{x}) \geq \mathbb{E}_{\mathbf{z}\sim q_\varepsilon(\mathbf{z}|\mathbf{x})}[\log p_\theta(\mathbf{x}|\mathbf{z})] - \beta^\star D_{KL}[q_\varepsilon(\mathbf{z}|\mathbf{x})||p(\mathbf{z})]. \tag{28}$$

In the beginning of the training, we use a small $\beta^\star$ which we then gradually increase $\beta^\star$ up to 1.0, during each task training, in the mixture objective function L_{L-MVAE} from Eq. (2), after replacing $\mathcal{L}^i_{VAE}(\mathbf{x})$ by using Eq. (28). We train the mixture model L-MVAE under the MNIST, Fashion, SVHN and CIFAR10 lifelong setting (MFSC sequence) as well as when considering learning these databases in reversed order, denoted as CSFM. We evaluate the Inception Score (IS) on 5000 testing samples from CIFAR10 and the corresponding reconstructions obtained by L-MVAE-MFSC and L-MVAE-CSFM, respectively. L-MVAE-MFSC and L-MVAE-CSFM represent L-MVAE to be trained on the order "MFSC" and "CSFM", respectively. The reconstruction results measured by Mean Squared Error (MSE), the structural similarity index measure (SSIM) [62] and Peak-Signal-to-Noise Ratio (PSNR) [62] are provided in Table VII show that L-MVAE achieves competitive results when comparing with other generative models, such as BatchEnsemble [61], that are only trained on CIFAR10. Additionally, the results also show that the order of learning the four databases does not have a significant impact on the L-MVAE training.

I. The results for the expandable mixture model

In this section, we evaluate the performance of the proposed expansion mechanism. We also compare to a state of the art ensemble model, called BatchEnsemble [61]. In order to allow BatchEnsemble to do unsupervised learning tasks, we implement each ensemble member as a VAE. We use the MSE, SSIM PSNR for the evaluation of reconstruction quality. We train L-MVAE and BatchEnsemble under MNIST, Fashion, SVHN and CIFAR10 lifelong learning. We set threshold $S^\star = 600$. After the training, L-MVAE has added three new components in the mixture model. We report the performance of the reconstruction in Table VIII where L-MVAE-Dynamic outperforms BatchEnsemble on three criteria. In the following, we perform the classification tasks under MNIST, Fashion, SVHN, and CIFAR10 lifelong learning. After the training, L-MVAE-Dynamic has four components. We report the results in Table IX. It observes that L-MVAE-Dynamic outperforms BatchEnsemble. We also perform a long sequence of tasks : MNIST, SVHN, Fashion, InverseFashion (IFashion), InverseMNIST (IMNIST), RatedFashion (RFashion), CIFAR10 (MSFIIRC). We set the threshold $S^\star = 400$ in Eq.(25) for MSFIIRC and provide the results in Tabel X where L-MVAE-Dynamic has five components after the lifelong learning. The first component is reused to learn RMNIST and the third component is reused to learn RFashion. This demonstrates that the proposed selection process can choose an appropriate expert that shares similar knowledge with a future task. Under this challenging learning setting, L-MVAE-Dynamic almost achieves the best results in each task when compared to BatchEnsemble.

VIII. Conclusion

This paper proposes a novel mixture system, called Lifelong Mixtures of VAEs (L-MVAE) model which is enabled for lifelong representation learning. Each time when a new task is available, the L-MVAE model adapts its weights in order to learn its corresponding probabilistic representation, without forgetting the information learnt from the previous tasks. A mixing-coefficient is used to determine which experts are activated or inactivated during the lifelong learning, preventing catastrophic forgetting. The proposed lifelong learning framework is applied for supervised, unsupervised and in semi-supervised learning. The L-MVAE model is also enabled with an expanding component mechanism. When presented with learning a completely new database, the mixture adds a component, while otherwise it updates the most suitable existing component. Experiments on various databases show the abilities of the proposed model in representing latent spaces inferred from learning sequentially from various databases. The representation capabilities of the model are shown by its ability to infer disentangled representations and interpolations in multiple domains learnt during the lifelong learning.

REFERENCES

[1] G. Parisi, R. Kemker, J. Part, C. Kanan, and S. Wermter, "Continual lifelong learning with neural networks: A review," in *Proc. of the ACM India Joint Int. Conf. on Data Science and Management of Data*, 2019, pp. 362–365.

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

[3] A. Rusu, N. Rabinowitz, G. Desjardins, H. Soyer, J. Kirkpatrick, K. Kavukcuoglu, R. Pascanu, and R. Hadsell, "Progressive neural networks," 2016. [Online]. Available: https://arxiv.org/abs/1606.04671

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

[5] G. Zhou, K. Sohn, and H. Lee, "Online incremental feature learning with denoising autoencoders," in *Proc. Int. Conf. on Artificial Intelligence and Statistics (AISTATS), vol. PMLR 22*, 2012, pp. 1453–1461.

[6] R. Polikar, L. Upda, S. Upda, and V. Honavar, "Learn++: An incremental learning algorithm for supervised neural networks," *IEEE Trans. on Systems Man and Cybernetics, Part C*, vol. 31, no. 4, pp. 497–508, 2001.

[7] G. Hinton, O. Vinyals, and J. Dean, "Distilling the knowledge in a neural network," in *Proc. NIPS Deep Learning Workshop*, 2014. [Online]. Available: https://arxiv.org/abs/1503.02531

[8] H. Jung, J. Ju, M. Jung, and J. Kim, "Less-forgetting learning in deep neural networks," in *Proc. AAAI Conf. on Artif. Intel.*, 2016, pp. 3358–3365.

[9] J. Kirkpatrick, R. Pascanu, N. Rabinowitz, J. Veness, G. Desjardins, 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.

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

[11] B. Ren, H. Wang, J. Li, and H. Gao, "Life-long learning based on dynamic combination model," *Applied Soft Computing*, vol. 56, pp. 398–404, 2017.

[12] A. Chaudhry, M. Rohrbach, M. Elhoseiny, T. Ajanthan, P. Dokania, P. H. S. Torr, and M. Ranzato, "On tiny episodic memories in continual learning," in *Proc. ICML Workshop on Multi-Task and Lifelong Reinforcement Learning*, 2019. [Online]. Available: https://arxiv.org/abs/1902.10486

[13] R. Aljundi, M. Lin, B. Goujaud, and Y. Bengio, "Gradient based sample selection for online continual learning," in *Advances in Neural Inf. Proc. Systems (NeurIPS)*, 2019, pp. 11 816–11 825.

[14] 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.

[15] J. Ramapuram, M. Gregorova, and A. Kalousis, "Lifelong generative modeling," in *Proc. Int. Conf. on Learning Representations (ICLR)*, 2018. [Online]. Available: https://arxiv.org/pdf/1705.09847.pdf

[16] 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.

[17] D. Rao, F. Visin, A. Rusu, R. Teh, Y. W. Pascanu, and R. Hadsell, "Continual unsupervised representation learning," in *Advances in Neural Inf. Proc. Systems (NeurIPS)*, 2019. [Online]. Available: https://arxiv.org/abs/1910.14481

[18] M. Zhai, L. Chen, F. Tung, J. He, M. Nawhal, and G. Mori, "Lifelong GAN: Continual learning for conditional image generation," in *Proc. IEEE Int. Conf. on Computer Vision (ICCV)*, 2019, pp. 2759–2768.

[19] 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.

[20] Y. Guo, M. Liu, T. Yang, and T. Rosing, "Improved schemes for episodic memory-based lifelong learning," 2020. [Online]. Available: https://arxiv.org/abs/1909.11763

[21] P. Singh, V. K. Verma, P. Mazumder, L. Carin, and P. Rai, "Calibrating CNNs for lifelong learning," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

[22] G. van de Ven and A. Tolias, "Three scenarios for continual learning," 2019. [Online]. Available: https://arxiv.org/abs/1904.07734

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

[24] Y. Shi, N. Siddharth, B. Paige, and P. Torr, "Variational mixture-of-experts autoencoders for multi-modal deep generative models," in *Advances in Neural Inf. Proc. Systems (NeurIPS)*, 2019, pp. 15 718–15 729.

[25] R. Kurle, S. Günnemann, and P. van der Smagt, "Multi-source neural variational inference," in *Proc. AAAI Conf. on Artificial Intelligence*, 2019, pp. 4114–4121.

[26] N. Dilokthanakul, P. Mediano, M. Garnelo, M. Lee, H. Salimbeni, K. Arulkumaran, and M. Shanahan, "Deep unsupervised clustering with Gaussian mixture variational autoencoders," in *Proc. Int. Conf. on Learning Representations (ICLR)*, 2018. [Online]. Available: https://arxiv.org/abs/1611.02648

[27] M. Wu and N. Goodman, "Multimodal generative models for scalable weakly-supervised learning," in *Advances in Neural Information Processing Systems*, 2018, pp. 5575–5585.

[28] D. P. Kingma and M. Welling, "Auto-encoding variational Bayes," 2013. [Online]. Available: https://arxiv.org/abs/1312.6114

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

[30] H. Kim and A. Mnih, "Learning disentangled joint continuous and discrete representations," in *Proc. Int. Conf. on Machine Learning (ICML), PMLR 80*, 2018, pp. 2649–2658.

[31] S. Gao, R. Brekelmans, G. V. Steeg, and A. Galstyan, "Auto-encoding total correlation explanation," in *Proc. Int. Conf. on Art. Intel. and Stat. (AISTATS), vol. PMLR 89*, 2019, pp. 1157–1166.

[32] C. P. Burgess, I. Higgins, A. Pal, L. Matthey, N. Watters, G. Desjardins, and A. Lerchner, "Understanding disentangling in β-VAE," in *Proc. NIPS Workshop on Learning Disentangled Representations*, 2018. [Online]. Available: https://arxiv.org/abs/1804.03599

[33] E. Mathieu, T. Rainforth, N. Siddharth, and Y. W. Teh, "Disentangling disentanglement in variational autoencoders," in *Proc. Int. Conf. on Machine Learning (ICML), PMLR 97*, 2019, pp. 4402–4412.

[34] R. M. French, "Catastrophic forgetting in connectionist networks," *Trends in cognitive sciences*, vol. 3, no. 4, pp. 128–135, 1999.

[35] H. Ritter, A. Botev, and D. Barber, "Online structured Laplace approximations for overcoming catastrophic forgetting," in *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 31, 2018, pp. 3742–3752.

[36] F. Ye and A. G. Bors, "Learning latent representations across multiple data domains using lifelong vaegan," in *Computer Vision – ECCV 2020*, A. Vedaldi, H. Bischof, T. Brox, and J.-M. Frahm, Eds. Cham: Springer International Publishing, 2020, pp. 777–795.

[37] I. 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.

[38] R. Aljundi, P. Chakravarty, and T. Tuytelaars, "Expert gate: Lifelong learning with a network of experts," in *Proc. of IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 3366–3375.

[39] W. Dai, Q. Yang, G. R. Xue, and Y. Yu, "Boosting for transfer learning," in *Proc. Int. Conf. on Machine Learning (ICML)*, 2007, pp. 193–200.

[40] C. V. Nguyen, Y. Li, T. D. Bui, and R. E. Turner, "Variational continual learning," in *Proc. Int. Conf. on Learning Representations (ICLR)*, 2018. [Online]. Available: https://arxiv.org/abs/1710.10628

[41] R. Kurle, B. Cseke, A. Klushyn, P. van der Smagt, and S. Günnemann, "Continual learning with bayesian neural networks for non-stationary data," in *Proc. Int. Conf. on Learning Representations (ICLR)*, 2020.

[42] A. Chaudhry, M. Ranzato, M. Rohrbach, and M. Elhoseiny, "Efficient lifelong learning with A-GEM," in *Proc. Int. Conf. on Learning Representations (ICLR)*, 2019. [Online]. Available: https://arxiv.org/abs/1812.00420

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

[44] N. Nasios and A. G. Bors, "Variational learning for Gaussian mixture models," *IEEE Trans. on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 36, no. 4, pp. 849–862, 2006.

[45] L. Weruaga and J. Via, "Sparse multivariate Gaussian mixture regression," *IEEE Trans. on Neural Networks and Learning Systems*, vol. 26, no. 5, pp. 1098–1108, 2015.

[46] E. Jang, S. Gu, and B. Poole, "Categorical reparameterization with Gumbel-Softmax," in *Proc. Int. Conf. on Learning Representations (ICLR)*, 2017. [Online]. Available: https://arxiv.org/abs/1611.01144

[47] E. J. Gumbel, "Statistical theory of extreme values and some practical applications," *NBS Applied Mathematics Series*, vol. 33, 1954.

[48] S. Narayanaswamy, B. Paige, J.-W. Van de Meent, A. Desmaison, N. Goodman, P. Kohli, F. Wood, and P. Torr, "Learning disentangled representations with semi-supervised deep generative models," in *Advances in Neural Inf. Proc. Systems (NeurIPS)*, 2017, pp. 5925–5935.

[49] 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.

[50] H. Xiao, K. Rasul, and R. Vollgraf, "Fashion-MNIST: a novel image dataset for benchmarking machine learning algorithms," 2017. [Online]. Available: https://arxiv.org/abs/1708.07747

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

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

[53] D. P. Kingma, S. Mohamed, D. J. Rezende, and M. Welling, "Semi-supervised learning with deep generative models," in *Advances in Neural Inf. Proc. Systems (NIPS)*, 2014, pp. 3581–3589.

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

[55] S. Arora, R. Ge, Y. Liang, T. Ma, and Y. Zhang, "Generalization and equilibrium in generative adversarial nets (GANs)," in *Proc. Int. Conf. on Machine Learning (ICML), vol. PMLR 70*, 2017, pp. 224–232. [Online]. Available: arXiv preprint arXiv:1703.00573

[56] A. Radford, L. Metz, and S. Chintala, "Unsupervised representation learning with deep convolutional generative adversarial networks," in *Proc. Int. Conf. on Learning Representations (ICLR)*, 2016. [Online]. Available: arXiv preprint arXiv:1511.06434

[57] L. Chen, S. Dai, Y. Pu, C. Li, Q. Su, and L. Carin, "Symmetric variational autoencoder and connections to adversarial learning," *arXiv preprint arXiv:1709.01846*, 2017.

[58] V. Dumoulin, I. Belghazi, B. Poole, O. Mastropietro, A. Lamb, M. Arjovsky, and A. Courville, "Adversarially learned inference," in *Proc. Int. Conf. on Learning Representations (ICLR)*, 2017. [Online]. Available: arXiv preprint arXiv:1606.00704

[59] T. Salimans, A. Karpathy, X. Chen, and D. P. Kingma, "PixelCNN++: Improving the PixelCNN with discretized logistic mixture likelihood and other modifications," in *Proc. Int. Conf. on Learning Representations (ICLR)*, 2017. [Online]. Available: https://arxiv.org/abs/1701.05517

[60] Y. Pu, W. Wang, R. Henao, C. L., Z. Gan, C. Li, and L. Carin, "Adversarial symmetric variational autoencoder," in *Advances in Neural Inf. Proc. Systems (NeurIPS)*, 2017, pp. 4333–4342.

[61] Y. Wen, D. Tran, and J. Ba, "BatchEnsemble: an alternative approach to efficient ensemble and lifelong learning," in *Proc. Int. Conf. on Learning Representations (ICLR)*, 2020. [Online]. Available: https://arxiv.org/abs/2002.06715

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