Article:
Ye, Fei and Bors, Adrian Gheorghe orcid.org/0000-0001-7838-0021 (2024) Lifelong Generative Adversarial Autoencoder. IEEE Transactions on Neural Networks and Learning Systems. pp. 14684-14698. ISSN 2162-237X

https://doi.org/10.1109/TNNLS.2023.3281091

Lifelong Generative Adversarial Autoencoder

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

***Abstract*—Lifelong learning describes an ability that enables humans to continually acquire and learn new information without forgetting. This capability, common to humans and animals, has lately been identified as an essential function for an artificial intelligence system aiming to learn continuously from a stream of data during a certain period of time. However, modern neural networks suffer from degenerated performance when learning multiple domains sequentially, and fail to recognize past learnt tasks after being retrained. This corresponds to catastrophic forgetting and is ultimately induced by replacing the parameters associated with previously learnt tasks with new values. One approach in lifelong learning is the Generative Replay Mechanism (GRM) that trains a powerful generator as the generative replay network, implemented by a Variational Autoencoder (VAE) or a Generative Adversarial Networks (GANs). In this paper, we study the forgetting behaviour of GRM-based learning systems by developing a new theoretical framework in which the forgetting process is expressed as an increase in the model's risk during the training. Although many recent attempts have provided high-quality generative replay samples by using GANs, they are limited to mainly downstream tasks due to the lack of inference. Inspired by the theoretical analysis while aiming to address the drawbacks of existing approaches, we propose the Lifelong Generative Adversarial Autoencoder (LGAA). LGAA consists of a generative replay network and three inference models, each addressing the inference of a different type of latent variable. The experimental results show that LGAA learns novel visual concepts without forgetting and can be applied to a wide range of downstream tasks.**

***Index Terms*—Lifelong learning, Generative Adversarial Autoencoders, Generative Replay Mechanism.**

I. Introduction

LIFELONG learning is becoming a requirement in real-time applications of artificial intelligence systems. This underlines the capability of remembering previously learned knowledge from multiple sources, during several training stages [1]. Such abilities are genetically inherited in humans and animals, enabling them to adapt to the environment during their entire life. However, neural network systems are far from matching such capabilities. The current state-of-the-art deep learning approaches would perform well on individual databases [2], [3], but suffer from degenerated performance when attempting to learn a sequence of tasks, where each task is associated with a different database [4], [5], [6], [7], [8], [9]. After having previously learnt the information associated with a certain dataset, a deep neural network updates its parameters when training for a new task. Consequently, its performance on the previous dataset degenerates due to the significant changes in the model's parameters, resulting in catastrophic forgetting when testing on the data learnt in the past.

In order to alleviate catastrophic forgetting, memory-based approaches employ additional buffers to store a small subset of previously seen data samples [10], [11], [12]. However, such an approach requires to design the criteria that would dynamically remove or add data samples in the buffer. Additionally, as the number of tasks increases, memory-based approaches require large buffers, which is unsuitable in practical applications. Another solution, called the Generative Replay Mechanism (GRM), consists of enabling a generator as the generative replay network for reproducing past samples when learning new tasks.

Many lifelong learning approaches employ Generative Adversarial Networks (GANs) as GRMs. Such an approach was firstly proposed in [13], where a classifier was used with the GRM framework, learning from the samples associated with a new task while generated samples are drawn from the outputs of the generator. However, such an approach requires generating a large number of samples after each task switch, which results in significant memory requirements. More recently, GRM was combined with the Knowledge Distillation for Conditional Image Generation, in an approach called the Lifelong GAN [14], which is built upon the BicycleGAN framework [15]. Lifelong GAN mainly focuses on the conditional image generation task while requiring the storage of past samples when learning new databases, which is not applicable for learning an infinite number of tasks. Additionally, such approaches require designing a specific network architecture for the classifier [13] and an encoding-decoding framework [14] to support the learning of many downstream tasks. Moreover, they do not learn meaningful latent representations within a single latent space. This represents a challenge for many tasks including image interpolation [16] and disentangled representation learning [17].

Learning meaningful and disentangled data representations has been shown to benefit many tasks [18]. Recently, learning disentangled representations under the lifelong learning was explored by introducing a framework based on the Variational Autoencoder (VAE), called VAE with Shared Embeddings (VASE) [19], which uses an environment-dependent mask to learn domain-specific latent representations. Additionally, VASE also uses the GRM to relieve forgetting. However, VAE-based GRM methods usually yield blurred generative replay images when compared with using GANs, leading to degenerated performance on the past tasks. Furthermore, existing lifelong learning literature does not provide the theoretical analysis for GRM-based approaches. This inspires us to develop a new theoretical framework in order to understand the forgetting behaviour of GRM-based models during lifelong learning. The main idea of the proposed theoretical analysis is to treat the lifelong learning problem as a dynamic domain adaptation problem in which the source domain is evolved over time. We then derive the risk bound based on the results for the dynamic domain adaptation problem in which we formulate

the variance of the model's risk as a learning or forgetting process, providing new insights into the forgetting behaviour of GRM-based models. Our other contribution consists in the development of Lifelong Generative Adversarial Autoencoders (LGAA), representing a new approach to lifelong learning which combines the advantages of both GANs and VAEs. We propose to train a powerful generative replay network by using adversarial learning while also training the inference models on the joint data samples corresponding to the new task combined with those produced by the generator, through a new optimization approach. The trained inference models can be used in a variety of applications, such as classification, image interpolation and for learning disentangled representations. The advantage of the proposed LGAA over existing GRM-based methods is that LGAA can train a robust generative replay network compared to VAE-based methods, while it can capture meaningful latent representations across domains compared to GAN-based approaches.

Our contributions are as follows :

1) We propose a novel lifelong learning model, called the Lifelong Generative Adversarial Autoencoder (LGAA), which not only it trains a robust generative replay network but also induces accurate inference models.
2) We develop a new theoretical framework for GRM-based models, in which the forgetting process is expressed by an increase in the model's risk during the training. The proposed theoretical analysis provides new insights into the forgetting behaviour of GRM-based models.
3) We extend LGAA by using adversarial learning to be used in a self-supervised manner while enforcing the inference models to capture data generative factors across domains.
4) Experiments show that the proposed LGAA can accumulate latent information from multiple domains without forgetting, which benefits many downstream tasks such as classification, reconstruction, and interpolation.

The rest of the paper is organized as in the following. Section II outlines the main approaches in the area of lifelong learning, while Section III outlines the background of Generative Reply Mechanisms (GRMs) and Section IV provides their theoretical analysis. The proposed Lifelong Generative Adversarial Autoencoder (LGAA) model is introduced in Section V and its training in Section VI. The experimental results are provided and discussed in Section VII, while the conclusions of this study are drawn in Section VIII.

II. Related works

Lifelong learning can be branched into three different perspectives : regularization, dynamic architectures, and memory replay based methods. Regularization methods introduce an additional term in the loss function that penalizes changes in the weights when the model is trained on a new task [13], [20], [21], [22], [23], [24], [25], [26]. This can prevent forgetting by preserving the weights considered important for storing the knowledge about previous tasks. Meanwhile, dynamic architectures increase the number of neurons and network layers in order to adapt to learning novel information [27]. Most memory replay approaches are using either Generative Adversarial Networks (GANs) [28] or Variational Autoencoders (VAEs) [29] to replay the previously learnt knowledge [30], [31], [32], [33]. For instance, Wu *et al.* [34] proposed the Memory Replay GANs (MeRGANs), which generates images from new categories under the lifelong learning setting. Meanwhile, Lifelong GAN [14] enables image to image translation under lifelong learning. However, the approaches from [34] and [14] lack a data inference procedure and Lifelong GAN requires to load all previously learnt data in order to be able to appropriately reproduce the corresponding information. Approaches employing both generative and inference mechanisms are based on the VAE framework [19], [35]. However, the VAE-based framework can not provide high-quality generative replay samples resulting in blurred images when trained on image databases. Also knowledge distillation was explored for lifelong learning in [36], [21]. The Lifelong Teacher-Student (LTS) [37], is a knowledge distillation approach which not only that transfers the discriminative information but also the data generative factors when continuously learning various tasks. However, LTS has an extra module (Student), which requires more parameters and additional computational cost to train when compared with the proposed LGAA.

Hybrid VAE-GAN methods adopt the inference mechanism from a VAE model which can capture data representations [38] enabling then adversarial learning to match either the data distribution [39], the latent variables distribution [40], or their joint distributions [38], [41], [42], [43], [44], [45], [46], [47]. Adversarial Autoencoder (AAE) [40] is the first method based on a VAE-GAN architecture, which replaces the Kullback-Leibler (KL) divergence with adversarial loss, encouraging the output distribution of the encoder to match the prior distribution. BiGANs [42] is another hybrid model which trains a discriminator network to learn the inverse mapping by projecting data back into the latent space. More recently, the Introspective Variational Autoencoder (IntroVAE) was applied for photographic image synthesis [48]. Unlike most other hybrid methods, which require an additional discriminator network for adversarial learning, IntroVAE employs the Inference network to distinguish between fake and real data. In addition, the hybrid model can be used to prevent the mode collapse from the GAN model [47] in which a reconstructing network is used for improving GAN's optimization. Although these hybrid models have shown promising performance in both generation as well as in inference mechanisms, they perform well only when trained on a single dataset and their performance degenerates when learning additional new tasks.

This paper is the first research study to propose a novel hybrid lifelong learning model, which not only that it addresses the drawbacks of existing hybrid methods but also provides inference mechanisms for GRM, benefiting on many downstream tasks across domains under the lifelong learning framework. Furthermore, the approach proposed in this paper also addresses disentangled representation learning [49] in the context of lifelong learning. Many recent approaches aim to modify the VAE framework in order to learn a meaningful representation of data by imposing a large penalty on the Kullback-Leibler (KL) divergence term [50], [51], [52], or

on the total correlation latent variables [17], [53], [54], [55]. These methods perform well on independent and identically distributed data samples from a single domain and cannot learn the information when changing the probabilistic representations of the data associated with multiple databases.

III. The Generative Replay Mechanism (GRM)

In this section, we introduce the background of generative replay approaches including GAN-based and VAE-based GRMs.

A. GAN-based GRM

A GAN model [28] consists of a discriminator $f_\omega \colon \mathcal{X} \to \mathbb{R}$ and a generator $G_\theta \colon \mathcal{Z} \to \mathcal{X}$ where $\mathcal{X}$ and $\mathcal{Z}$ represent the input data and latent spaces, respectively; $\{\theta, \omega\}$ are the parameters for the generator and discriminator. GANs enjoy an efficient sampling process where a random vector $\mathbf{z}$ is drawn from a simple prior distribution $p(\mathbf{z}) = \mathcal{N}(0, \mathbf{I})$, and is then used as input to $G_\theta(\cdot)$ for producing data, considered in the following as images $\mathbf{x}'$. The learning goal of GANs is that of trying to train an optimal discriminator that distinguishes between a real image $\mathbf{x}$ drawn from the empirical data distribution $\mathbb{P}_\mathbf{x}$, and a fake one $\mathbf{x}'$, output of the generator. Meanwhile the generator is encouraged to produce samples $\mathbf{x}'$ that can best cheat the discriminator. This learning process can be summarized as a two-player minimax game with the loss function :

$$\min_G \max_f V(G, f) = \mathbb{E}_{\mathbf{x}\sim\mathbb{P}_\mathbf{x}}[\log f(\mathbf{x})] + \mathbb{E}_{\mathbf{z}\sim p(\mathbf{z})}[1 - \log(f(G(\mathbf{z})))]. \tag{1}$$

By considering the properties of GANs, forgetfulness could be relieved by training a GAN model on a joint dataset consisting of generative samples produced by using the generator $G_\theta(\cdot)$ and real data sampled from a database corresponding to a given task.

B. VAE-based GRM

The VAE [29] is a generative latent variable model $p_\theta(\mathbf{x}, \mathbf{z})$ which is enabled with an inference mechanism. VAEs consist of a decoder, used as generator $G_\theta \colon \mathcal{Z} \to \mathcal{X}$, similar to that found in GANs, and an encoder, used as the inference model $F_\xi \colon \mathcal{X} \to \mathcal{Z}$, where ξ are the parameters of the inference network. Typically, training a VAE model requires to maximize the sample log-likelihood $\log p_\theta(\mathbf{x}) = \log \int p_\theta(\mathbf{x} \,|\, \mathbf{z}) p(\mathbf{z}) \, \mathrm{d}\mathbf{z}$ which is intractable during the optimization because it requires to get all latent variables $\mathbf{z}$. To address this, VAEs use variational inference which decomposes $\log p_\theta(\mathbf{x})$ as :

$$\log p_\theta(\mathbf{x}) - D_{KL}(q_\xi(\mathbf{z} \,|\, \mathbf{x}) \,||\, p(\mathbf{z} \,|\, \mathbf{x})) = \mathbb{E}_{z\sim q_\xi(\mathbf{z}\,|\,\mathbf{x})}\left[\log p_\theta(\mathbf{x} \,|\, \mathbf{z})\right] - D_{KL}\left[q_\xi(\mathbf{z} \,|\, \mathbf{x}) \,||\, p(\mathbf{z})\right] \tag{2}$$

where $D_{KL}(\cdot)$ is the Kullback–Leibler (KL) divergence and $p(\mathbf{z} \,|\, \mathbf{x})$ is the posterior. $q_\xi(\mathbf{z} \,|\, \mathbf{x})$ is the variational distribution, implemented by the inference model $F_\xi(\cdot)$. Since $D_{KL}(\cdot) \geq 0$, the sample log-likelihood $\log p_\theta(\mathbf{x})$ can be approximated by a lower bound, called the Evidence Lower Bound (ELBO) :

$$\log p_\theta(\mathbf{x}) \geq \mathcal{L}_{\mathrm{VAE}}(\mathbf{x}) := \mathbb{E}_{z\sim q_\xi(\mathbf{z}|\mathbf{x})}\left[\log p_\theta(\mathbf{x} \mid \mathbf{z})\right] - D_{KL}\left[q_\xi(\mathbf{z} \,|\, \mathbf{x}) \,||\, p(\mathbf{z})\right]. \tag{3}$$

The last term from the right-hand side of Eq. (3) can be seen as the regularization term penalizing the deviation of the variational distribution $q_\xi(\mathbf{z} \,|\, \mathbf{x})$ from the prior $p(\mathbf{z})$. This can allow VAEs to generate images from the random noise vector $\mathbf{z} \sim p(\mathbf{z})$. Similar to GANs, a VAE model can be trained on its generations to relieve the forgetting when learning new tasks.

IV. Theoretical Analysis of the GRM

In Section III, we have introduced two different generative models, GAN and VAE, which can be used as generative replay mechanisms. In this section, we provide the theoretical framework in which we analyze the forgetting behaviour of GRM-based systems. This framework represents the motivation for the proposed Lifelong Generative Adversarial Autoencoder method, detailed in Section V. We firstly introduce some definitions of key concepts used in this framework.

Definition 1: We consider the data $\mathbf{x}^t$ and their t-th task probabilistic representation, $\mathbf{x}^t \sim \mathbb{P}_{\mathbf{x}^t}$, $t \geq 1$. Let us consider a model $\mathcal{M}_t$ (implemented by a generative model, GAN or VAE), which has been trained for the t-th given task, under the lifelong learning assumption of training sequentially with multiple tasks. Let $\mathbb{P}_{\tilde{\mathbf{x}}^t}$ be the distribution of generative replay samples drawn from the generator of $\mathcal{M}_t$ and $\tilde{\mathbf{x}}^t$ the random variable over $\mathbb{P}_{\tilde{\mathbf{x}}^t}$.

Then we have the following conditional probability.

Definition 2: Let us define

$$p(\tilde{\mathbf{x}}^t \mid \tilde{\mathbf{x}}^{t-1}, \mathbf{x}^t) = \exp(-\Psi(\mathbb{P}_{(\tilde{\mathbf{x}}^{t-1}, \mathbf{x}^t)}, \mathbb{P}_{\tilde{\mathbf{x}}^t})), \tag{4}$$

as the probability of generated data $\tilde{\mathbf{x}}^t$ when observing $\tilde{\mathbf{x}}^{t-1}$ and $\mathbf{x}^t$ of the joint probability $\mathbb{P}_{(\tilde{\mathbf{x}}^{t-1}, \mathbf{x}^t)}$, where $\Psi(\cdot)$ is a probabilistic measure of comparison between two distributions, which can be the f-divergence [56], or the Wasserstein distance [57] (Earth-Mover distance).

Theorem 1: By marginalizing over $\tilde{\mathbf{x}}^{t-1}$ and $\mathbf{x}^t$, on $p(\tilde{\mathbf{x}}^t \mid \tilde{\mathbf{x}}^{t-1}, \mathbf{x}^t)$, the resulting marginal distribution $p(\tilde{\mathbf{x}}^t)$ encodes the statistical correlations from all previously learnt distributions :

$$p(\tilde{\mathbf{x}}^t) = \int \dots \int p(\tilde{\mathbf{x}}^1) \prod_{i=0}^{t-2} p(\tilde{\mathbf{x}}^{t-i} \mid \tilde{\mathbf{x}}^{t-i-1}, \mathbf{x}^{t-i}) \prod_{i=0}^{t-2} p(\mathbf{x}^{t-i}) \, \mathrm{d}\tilde{\mathbf{x}}^1 \dots \mathrm{d}\tilde{\mathbf{x}}^{t-1} \, \mathrm{d}\mathbf{x}^2 \dots \mathrm{d}\mathbf{x}^t \tag{5}$$

The proof is provided in Appendix-A. In the following, we describe how a GRM-based model can avoid forgetting when achieving the optimal solution.

Lemma 1: As shown in Definition 2, the conditional probability $p(\tilde{\mathbf{x}}^t \,|\, \tilde{\mathbf{x}}^{t-1}, \mathbf{x}^t)$ can evaluate the knowledge loss when learning the t-th task. As $p(\tilde{\mathbf{x}}^t \,|\, \tilde{\mathbf{x}}^{t-1}, \mathbf{x}^t) \approx 1$, the model would approximate $\mathbb{P}_{(\tilde{\mathbf{x}}^{t-1}, \mathbf{x}^t)}$. Then, we conclude that $p(\tilde{\mathbf{x}}^t)$ approximates the true joint distribution $\prod_{i=1}^{t} p(\mathbf{x}^i)$ when all previously learnt distributions are the exact approximations to their target distributions while learning every task from a sequence of tasks.

The detailed proof for *Lemma 1* is provided in Appendix-B. In order to analyse how a GRM-based model would lose knowledge when learning new tasks, we propose to adopt the domain theoretical analysis from [58] (*Theorem 2*) in order to

evaluate the model's risk. Firstly, we define the model's risk as :

$$\mathcal{R}(h, h_{(\nu_t)}) = \mathbb{E}_{\mathbf{x}\sim\nu_t}[\mathcal{L}(h(\mathbf{x}), h_{(\nu_t)}(\mathbf{x}))] \tag{6}$$

where $h\colon \mathcal{X} \to \mathcal{X}$ is the given model (usually representing the encoding-decoding process) and $\mathcal{L}(\cdot)$ is the loss function (usually the classification loss). $h_{(\nu_t)}$ is the identity function $\mathbf{x} = h_{(\nu_t)}(\mathbf{x})$ for the dataset ν_t. Then we can derive the risk bound between the target and source distribution as follows.

Theorem 2: Let us consider two data population samples, one corresponding to the generated data $\{\nu_{t'} \in \mathbb{R}^s | \nu_{t'} \sim \mathbb{P}_{\tilde{\mathbf{x}}^t}\}$ and another corresponding to the real data $\{\nu_t \in \mathbb{R}^s | \nu_t \sim \mathbb{P}_{\mathbf{x}^t}\}$ of sizes n_t and $n_{t'}$. For any $s' > s$ and $a' < \sqrt{2}$, there is a constant n_0, depending on s', satisfying that for any $\delta > 0$ and $\min(\nu_t, n_{t'}) \geq n_0 \max(\delta^{-(s'+2)}, 1)$. Then with the probability of at least $1-\delta$ for all $h \in \mathcal{H}$, where $\mathcal{H}$ is a family of models, we have:

$$\begin{aligned}\mathcal{R}\left(h, h_{(\nu_t)}\right) \leq \mathcal{R}\left(h, h_{(\nu_{t'})}\right) + W\left(\nu_t, \nu_{t'}\right) \\ + \sqrt{2\log\left(\frac{1}{\delta}\right)/a'}\left(\sqrt{\frac{1}{n_t}} + \sqrt{\frac{1}{n_{t'}}}\right) \\ + D(\mathcal{R}(h, h_{(\nu_t)}) + \mathcal{R}(h, h_{(\nu_{t'})})),\end{aligned} \tag{7}$$

where $\mathcal{R}(h, h_{(\nu_t)})$ and $\mathcal{R}(h, h_{(\nu_{t'})})$ denote the observed risks for ν_t and $\nu_{t'}$, respectively, and $W(\nu_t, \nu_{t'})$ is the Wasserstein distance between ν_t and $\nu_{t'}$. $D(\cdot)$ is the combined error when we find the optimal model, h' :

$$h' := \arg\min_{h\in\mathcal{H}} (\mathcal{R}(h, h_{(\nu_t)}) + \mathcal{R}(h, h_{(\nu_{t'})}). \tag{8}$$

Remark 1: We have the following observations :

- The risk $\mathcal{R}\left(h, h_{(\nu_t)}\right)$ of the model h on real training data samples is bounded by the model's risk on the generative replay samples plus the Wasserstein distance between ν_t and $\nu_{t'}$, and the combined error, according to Eq. (7).
- The bound is tight when the Wasserstein distance between the target and source distribution is small.

Theorem 2 only provides a risk bound for a single task and does not explain how a GRM-based model loses the knowledge when learning new tasks. In the following, we propose a risk bound for multi-task learning analysis, which assesses how the knowledge learnt from a task j is forgotten after the model is trained with other tasks.

Since the generator of a GRM-based model reproduces data consistent with all previously learnt data probabilistic representations of the given tasks, we assume that we have an optimal task-inference model $f_{task}\colon \mathcal{X} \to \mathbb{R}$ which can provide the exact task label for each data sample. With $f_{task}(\cdot)$, we can form several distributions for the generative samples in different training phases. Let $\mathbb{P}_{\tilde{\mathbf{x}}}^{(t,j)}$ be the distribution of the generative samples drawn and selected from the model $\mathcal{M}^t$, where if $f_{task}(\mathbf{x}') = j$ then sample $\mathbf{x}' \sim \mathbb{P}_{\tilde{\mathbf{x}}^t}$ is selected. Therefore, $\mathbb{P}_{\tilde{\mathbf{x}}}^{(t,j)}$ represents the distribution of generative samples corresponding to the j-th task and we have $\{\mathbb{P}_{\tilde{\mathbf{x}}}^{(j-1,j)}, \mathbb{P}_{\tilde{\mathbf{x}}}^{(j,j)}, \ldots, \mathbb{P}_{\tilde{\mathbf{x}}}^{(t,j)}\}$ at different training phases, where $\mathbb{P}_{\tilde{\mathbf{x}}}^{(j-1,j)}$ and $\mathbb{P}_{\tilde{\mathbf{x}}}^{(k,j)}, k \geq j$ represent the distributions of the training set of the j-th task and the distribution of generative samples drawn and selected from $\mathbb{P}_{\tilde{\mathbf{x}}^k}$ and $f_{task}(\cdot)$ at the k-th task learning. Considering these assumptions and notations, we derive the risk bound which measures the forgetting behaviour of a model after learning each task during lifelong learning.

Lemma 2: We derive the risk bound for learning the j-th task, at the t-th task learning, $t \geq j$, as :

$$\begin{aligned}\mathcal{R}\left(h, h_{(\nu_j)}\right) \leq \mathcal{R}\left(h, h_{(\nu_{(t,j)})}\right) \\ + \sum_{k=j-1}^{t-1}\Big\{W\left(\nu_{(k,j)}, \nu_{(k+1,j)}\right) \\ + \sqrt{2\log\left(\frac{1}{\delta}\right)/a'}\left(\sqrt{\frac{1}{n_{\nu_{(k,j)}}}} + \sqrt{\frac{1}{n_{\nu_{(k+1,j)}}}}\right) \\ + D(\mathcal{R}(h, h_{(\nu_{(k,j)})}) + \mathcal{R}(h, h_{(\nu_{(k+1,j)})})\Big\},\end{aligned} \tag{9}$$

where $\nu_{(j-1,j)}$ represents ν_j, and $\{\nu_{(t,j)} \in \mathbb{R}^s | \nu_{(t,j)} \sim \mathbb{P}_{\tilde{\mathbf{x}}}^{(t,j)}\}$ and $n_{\nu_{(t,j)}}$ is the corresponding sample size. We provide the detailed proof in Appendix-C.

From Eq. (9), we observe that the difference between the target and source distributions, evaluated by the Wasserstein distance, during the learning of each task plays an important role in the performance. Suppose the model achieves the optimal solution (*Lemma 1*), then the risk bound from Eq. (9) is tight since the tightness of this bound depends only on the Wasserstein distance between the target and generator distributions and a constant. As we discussed in Sections III-A and III-B, GANs and VAEs exhibit different forgetting behaviours according to Eq. (9). For instance, a GAN usually produces better data generations when compared to the VAE, and, therefore, can provide high-quality generative replay samples when learning a new task. This can minimize the Wasserstein distance between the target and source distribution in each task learning, leading to a tight risk bound.

In the following, we extend Lemma 2 to the situation when learning multiple tasks.

Lemma 3: For a given sequence of t tasks, we derive the risk bound at the t-th task learning :

$$\begin{aligned}\sum_{c=1}^{t}\mathcal{R}\left(h, h_{(\nu_c)}\right) \leq \sum_{c=1}^{t}\Big\{\mathcal{R}\left(h, h_{(\nu_{(t,c)})}\right) \\ + \sum_{k=c-1}^{t-1}\Big\{W\left(\nu_{(k,c)}, \nu_{(k+1,c)}\right) + \\ \sqrt{2\log\left(\frac{1}{\delta}\right)/a'}\left(\sqrt{\frac{1}{n_{\nu_{(k,j)}}}} + \sqrt{\frac{1}{n_{\nu_{(k+1,j)}}}}\right) \\ + D(\mathcal{R}(h, h_{(\nu_{(k,c)})}) + \mathcal{R}(h, h_{(\nu_{(k+1,c)})}))\Big\}\Big\}.\end{aligned} \tag{10}$$

For the proof, we sum up the risks of the model, defined according to *Lemma 2*, for all tasks.

Remark 2: The following observations are consequences of *Lemma 3*.

- From Eq. (10), we observe that the error terms are accumulated when learning an increasing number of tasks.
- The errors for the earlier learnt tasks are larger than those corresponding to those learnt more recently due to the

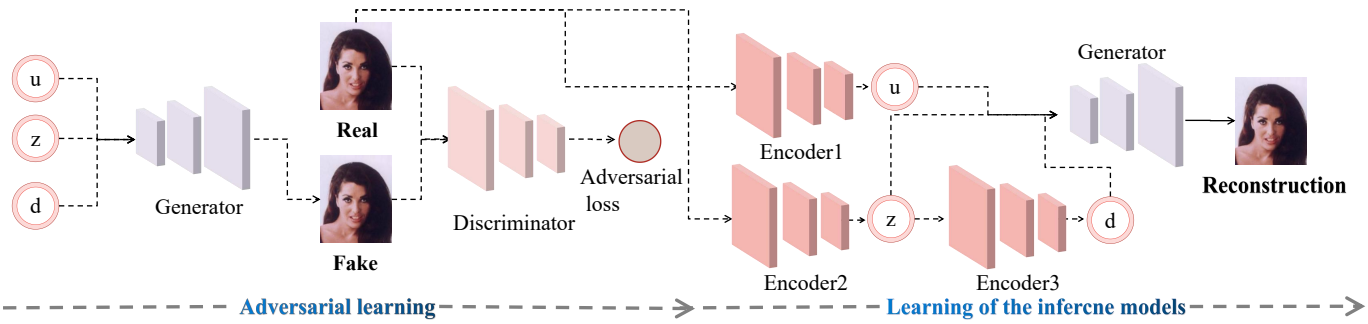


Fig. 1. The network structure of the proposed LGAA model. The whole learning procedure is divided into two steps. At the first step, we draw random vectors $\{\mathbf{u}, \mathbf{z}, \mathbf{d}\}$ from the prior distributions and then consider them as input for the generator for producing the fake image. The adversarial loss, defined by Eq. (12), is used for both the generator and discriminator. At the second step, we use the inference model (Encoder 1, Encoder 2, Encoder 3) to infer the latent variables $\{\mathbf{u}, \mathbf{z}, \mathbf{d}\}$ which are used for the reconstruction through the decoding process. The loss function Eq. (14), is used to jointly train all modules.

degeneration in the previously learned knowledge and the generator's retraining limitations.

- The optimal performance of the model h can be achieved if the generator distribution approximates the target distribution in each task learning (*Lemma 1*). This requires training a good generative replay network that generates high-quality data consistent with the learnt knowledge.
- GANs, when compared to VAE models, provide high-quality generative replay samples which can lead to better performance, generating sharper images for example. This phenomenon is theoretically explained in Lemma 3 where the Wasserstein distance between the generator's distribution and the target distribution in each task learning is crucial for performance.

V. Lifelong Generative Adversarial Autoencoder

From the theoretical analysis section, we observe that the quality of generative replay samples is the key to the performance of the GRM-based models during lifelong learning. Additionally, most GRM-based approaches do not have an inference model to train, which prevents them from extracting meaningful representations for downstream tasks. This inspires us to propose the Lifelong Generative Adversarial Autoencoder (LGAA), which not only learns a powerful generative replay network but also trains accurate inference models for representation learning.

A. Problem formulation

For a given sequence of t tasks, each characterized by a distinct data set, corresponding to the data distributions $\{\mathbb{P}_{\mathbf{x}^1}, \mathbb{P}_{\mathbf{x}^2}, \ldots, \mathbb{P}_{\mathbf{x}^t}\}$, our learning goal is to learn an inference model which can infer and accumulate the representation information from novel concepts without forgetting previously learnt knowledge. This can allow us to perform many downstream tasks, such as classification, reconstruction and interpolation, by using the inference model.

B. Training a powerful generative replay network

We consider that the underlying information from the observed data samples is defined by three latent variables $\{\mathbf{u}, \mathbf{d}, \mathbf{z}\}$ where $\mathbf{d} = \{d_i | i = 1, \ldots, K\}$ is the discrete variable

Algorithm 1: The supervised learning for LGAA

Input: All training databases
Output: The model's parameters

for $i < taskCount$ **do**
  $\mathbb{P}_{\tilde{\mathbf{x}}^{(i-1)}} \leftarrow G_{\theta_{(i-1)}}(\mathbf{z})$ Form the generator distribution ;
  $\tilde{\mathbf{Y}}^{(i-1)} \leftarrow F_{\delta_{(i-1)}}(\mathbf{u} \mid \mathbf{x})$ Generate class labels ;
  $\tilde{\mathbf{A}}^{(i-1)}$ Generate domain labels using the sample process $\mathbf{d} \sim q_{\varepsilon_{(i-1)}}(\mathbf{d} \mid \mathbf{z}), \mathbf{z} \sim q_{\varsigma_{(i-1)}}(\mathbf{z} \mid \mathbf{x}), \mathbf{x} \sim \mathbb{P}_{\tilde{\mathbf{x}}(i-1)}$;
  Combine the data;
  $\mathbb{P}_{(\mathbf{x}^i, \tilde{\mathbf{x}}^{(i-1)})} = \mathbb{P}_{\mathbf{x}^i} \bigcup \mathbb{P}_{\tilde{\mathbf{x}}^{(i-1)}}$;
  $\mathbf{A} = \mathbf{A}^i \bigcup \tilde{\mathbf{A}}^{(i-1)}$;
  $\mathbf{Y} = \mathbf{Y}^i \bigcup \tilde{\mathbf{Y}}^{(i-1)}$;
  for $j < batchCount$ **do**
    $(\mathbf{x}, \mathbf{y}) \sim (\mathbb{P}_{(\mathbf{x}^i, \tilde{\mathbf{x}}^{(i-1)})}, \mathbf{Y})$. ;
    Adversarial learning ;
    Update the generator and discriminator using $\min_G \max_D \mathcal{L}^G_{GAN}(\theta_i, \omega_i)$;
    Learning by the VAE loss ;
    Update all components using $\mathcal{L}^{Sup}_{VAE}(\theta_i, \varsigma_i, \varepsilon_i, \delta_i)$;
    Update the inference models ;
    Update $q_{\varepsilon_i}(\mathbf{d} \mid \mathbf{z})$ using $\mathcal{L}_{\mathbf{d}}(\varepsilon_i)$;
    Update $q_{\delta_i}(\mathbf{u} \mid \mathbf{x})$ using $\mathcal{L}_{\mathbf{u}}(\delta_i)$;
  end
end

(one-hot vector) representing the domain information, where K is the number of domains; $\mathbf{u} = \{u_i | i = 1, \ldots, M\}$ is a discrete variable of dimension M, which represents the discriminative information; $\mathbf{z}$ is the continuous variable. The generation process is defined by :

$$\begin{aligned} \mathbf{d} &= f_{\text{OneHot}}(d), d \ \sim \text{Cat}\,(K, 1/K)\,, \\ \mathbf{u} &= f_{\text{OneHot}}(u), u \sim \text{Cat}\,(M, 1/M)\,, \\ \mathbf{z} &\sim \mathcal{N}(0, \mathrm{I}), \\ \tilde{\mathbf{x}} &\sim p_\theta(\mathbf{x} \mid \mathbf{z}, \mathbf{d}, \mathbf{u}), \end{aligned} \tag{11}$$

where $\text{Cat}(\cdot)$ is the Categorical distribution, and $p_\theta(\mathbf{x} \mid \mathbf{z}, \mathbf{u}, \mathbf{d})$ is the generator distribution implemented by a neural network with trainable parameters, θ. f_{OneHot} is the function that transfers the category variable to the one-hot vector. In order to train a powerful generative replay network, we use the Wasserstein GAN (WGAN) [57] objective function with the

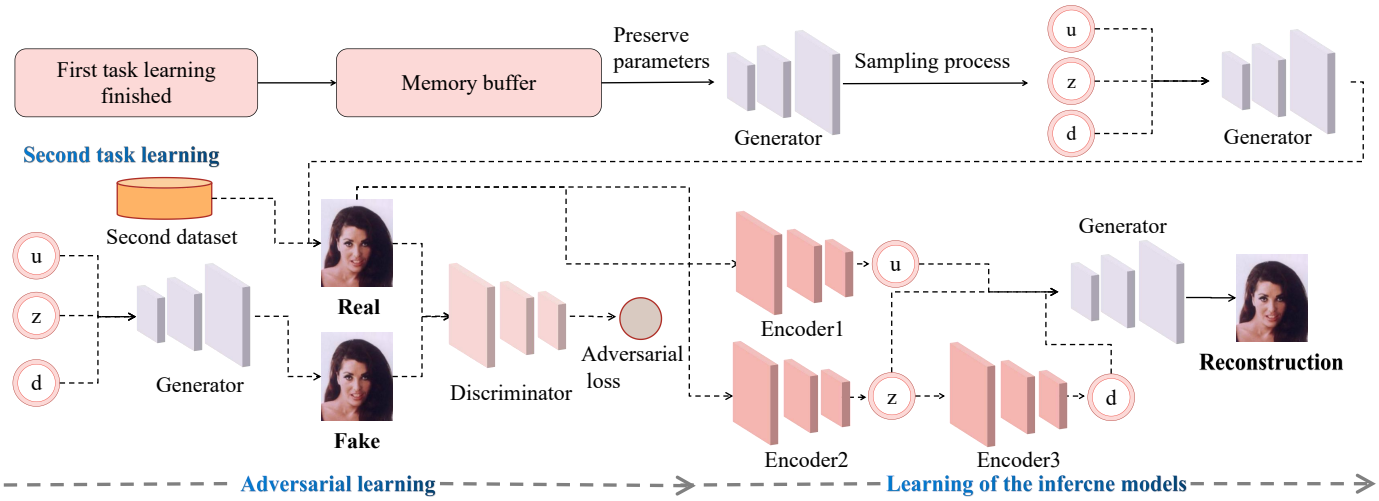


Fig. 2. Using the memory buffer in the LGAA framework. Once the first task is learnt, we use a buffer to preserve the parameters of the generator. Then during the second task learning, the preserved generator is used as a generative replay mechanism producing a batch of samples. The generated data samples are incorporated together with new samples drawn from the second task for training LGAA. Then the process of creating buffers for temporary storing generator parameters is repeated each time when learning a new task.

gradient penalty [59], defined by :

$$\min_{G}\max_{\mathcal{D}} \mathcal{L}^{G}_{GAN}(\theta,\omega) = \mathbb{E}_{\mathbf{z}\sim p(\mathbf{z}),\mathbf{u}\sim p(\mathbf{u}),\mathbf{d}\sim p(\mathbf{d})}[\mathcal{D}(G(\mathbf{u},\mathbf{z},\mathbf{d}))] - \mathbb{E}_{\mathbf{x}\sim\mathbb{P}_{\mathbf{x}}}[\mathcal{D}(\mathbf{x})] + \lambda\mathbb{E}_{\tilde{\mathbf{x}}\sim p(\tilde{\mathbf{x}})}[(\|\nabla_{\tilde{\mathbf{x}}}\mathcal{D}(\tilde{\mathbf{x}})\|_2 - 1)^2], \quad (12)$$

where $\mathcal{D}$ represents the discriminator, implemented by a neural network with trainable parameters ω, $\mathbb{P}_{\mathbf{x}}$ is the real data distribution, and the third term from the right-hand side is the gradient weighted by the penalty λ. The adversarial loss allows the generator and discriminator to be trained alternately such that the discriminator aims to distinguish real from generated data, while the generator aims to fool the discriminator by generating realistic data [28], [57].

C. The inference mechanism of LGAA

Most GAN-based lifelong methods [13], [14], [34] do not learn an accurate inference model and therefore can not derive a meaningful data representation. For the model proposed in this paper, we consider three differentiable non-linear functions $F_\varsigma(\cdot)$, $F_\varepsilon(\cdot)$, $F_\delta(\cdot)$, aiming to infer three different types of latent variables $\{\mathbf{z},\mathbf{d},\mathbf{u}\}$, as indicated in Section V-B. We implement $F_\varsigma(\cdot)$ considering the underlying Gaussian distribution $\mathcal{N}(\mu,\sigma)$, where $\{\mu,\sigma\}$ are the hyperparameters of the Gaussian distribution. We use the reparameterization trick [29], [60] for sampling $\mathbf{z} = \boldsymbol{\mu} + \boldsymbol{\pi}\odot\boldsymbol{\sigma}$, where $\boldsymbol{\pi}$ is a random noise vector sampled from $\mathcal{N}(\mathbf{0},\mathbf{I})$, in order to ensure end-to-end training.

We can not sample the discrete latent variables $\mathbf{d}$ and $\mathbf{u}$ from $F_\varepsilon(\cdot)$ and $F_\delta(\cdot)$, respectively, because the categorical representations are non-differentiable. In order to mitigate this, we use the Gumbel-Max trick [61], [62] for achieving the differentiable relaxation of discrete random variables. The Gumbel-softmax trick was also used in [51], [63], [64] and its capability of reducing the variation of gradients was studied in [65].

The sampling process of discrete latent variables is defined as:

$$d_j = \frac{\exp((\log d'_j + g_j)/T)}{\sum_{i=1}^{K}\exp((\log d'_i + \mathbf{g}_i)/T)} \quad (13)$$

where d'_i is the i-th entry of the probability defined by the softmax layer characterizing $F_\varepsilon(\cdot)$ and d_j is the continuous relaxation of the j-th dimension of the variable $\mathbf{d}$, while g_k is sampled from the distribution $\mathrm{Gumbel}(0,1)$ and T is the temperature parameter controlling the degree of smoothness. A small T indicates that $\mathbf{d}$ is close to the one-hot vector. In contrast, a large T indicates that $\mathbf{d}$ is close to the samples drawn from a uniform distribution [63]. In our experiment, we set $T = 0.5$ to encourage sampling the one-hot representation. We use the Gumbel softmax trick for sampling both domain $\mathbf{d}$ and discrete $\mathbf{u}$ variables.

D. The objective function for the inference models

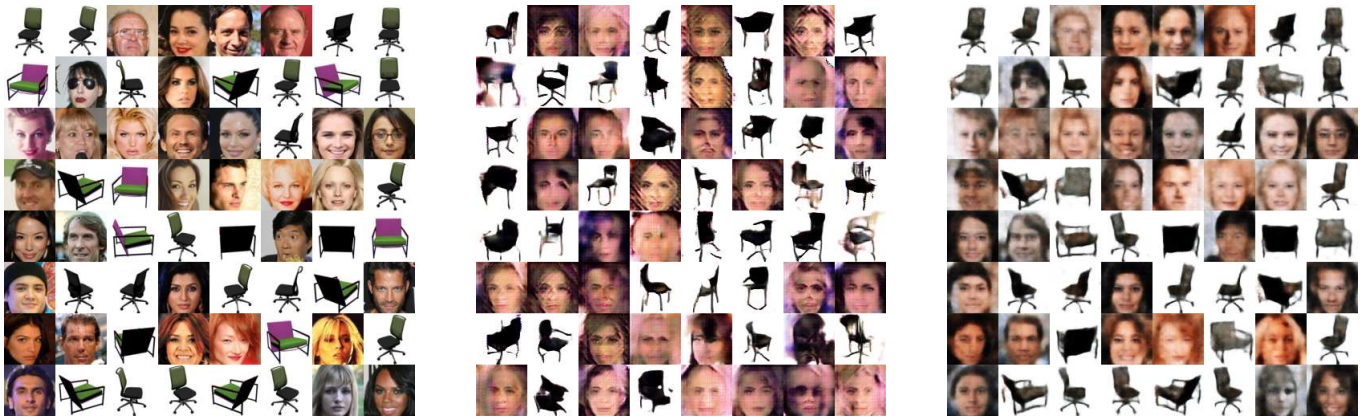
GANs lack an inference mechanism, preventing them from capturing data representations properly. In this paper we aim for updating inference models attached to a generative network using a VAE framework with three latent variables. Training a VAE framework usually maximises a lower bound to the sample log-likelihood. However, such optimization is intractable when involving multiple latent variables. Therefore, we derive a tractable VAE-based objective function as follows :

$$\mathcal{L}_{\mathrm{VAE}}(\theta,\varsigma,\varepsilon,\delta) = \mathbb{E}_{q_{\varsigma,\varepsilon,\delta}(\mathbf{z},\mathbf{u},\mathbf{d}\,|\,\mathbf{x})}\log[p_\theta(\mathbf{x}\,|\,\mathbf{z},\mathbf{u},\mathbf{d})] - D_{KL}[q_\varsigma(\mathbf{z}\,|\,\mathbf{x})\,||\,p(\mathbf{z})] - D_{KL}[q_\varepsilon(\mathbf{d}\,|\,\mathbf{z})\,||\,p(\mathbf{d})] - D_{KL}[q_\delta(\mathbf{u}\,|\,\mathbf{x})\,||\,p(\mathbf{u})]. \quad (14)$$

The detailed derivation is provided in Appendix-D. $q_\varsigma(\mathbf{z}\,|\,\mathbf{x})$, $q_\varepsilon(\mathbf{u}\,|\,\mathbf{z})$, $q_\delta(\mathbf{d}\,|\,\mathbf{x})$ are variational distributions modelled by $F_\varsigma(\cdot)$, $F_\varepsilon(\cdot)$, $F_\delta(\cdot)$, respectively. $p(\mathbf{d})$ is the prior distribution by the Concrete distribution, where each parameter is set to the same value $1/K$. We consider $q_\varepsilon(\mathbf{d}\,|\,\mathbf{z})$ as the task-inference

(a) Real samples. (b) Generations. (c) Reconstructions.

Fig. 3. The reconstruction and generation results under the CelebA to CACD lifelong learning.



(a) Real samples. (b) Generations. (c) Reconstructions.

Fig. 4. The reconstruction and generation results under the CelebA to 3D-Chair lifelong learning.

model which aims to infer the task ID for the given data samples.

For the supervised learning setting, auxiliary information such as class labels is used to guide the inference model. In this case we minimize the cross-entropy loss $\eta(\cdot,\cdot)$ for $q_\varepsilon(\mathbf{d}\,|\,\mathbf{z})$ and $q_\delta(\mathbf{u}\,|\,\mathbf{x})$, as:

$$\mathcal{L}_{\mathbf{d}}(\varepsilon) = \mathbb{E}_{(\mathbf{x},\mathbf{d}^*)\sim(\mathbf{X},\mathbf{A}),\mathbf{z}\sim q_\varsigma(\mathbf{z}\,|\,\mathbf{x})}\eta(q_\varepsilon(\mathbf{d}\,|\,\mathbf{z}),\mathbf{d}^*)\,, \tag{15}$$

$$\mathcal{L}_{\mathbf{u}}(\delta) = \mathbb{E}_{(\mathbf{x},\mathbf{y})\sim(\mathbf{X},\mathbf{Y})}\eta(q_\delta(\mathbf{u}\,|\,\mathbf{x}),\mathbf{y})\,, \tag{16}$$

where $\mathbf{X}$ and $\mathbf{Y}$ represent the empirical data and target distributions, respectively. $\mathbf{d}^*$ is the variable drawn from $\mathbf{A}$, which represents the domain variables distribution. The network architecture of the generator and inference networks of the proposed LGAA is shown in Fig. 1, where the variable $\mathbf{d}$ is conditioned on $\mathbf{z}$. The proposed model is flexible to be extended for recognizing new tasks by automatically appending the domain variable $\mathbf{d}$ and optimizing the task-inference model $q_\varepsilon(\mathbf{d}\,|\,\mathbf{z})$ when faced with learning a new task.

The inference models in the proposed LGAA aim to map an input into three compact low-dimensional feature vectors $\{\mathbf{z},\mathbf{d},\mathbf{u}\}$, with each describing different characteristics of the input. The decoder (generator) will recover an image from these three feature vectors. In the supervised learning framework we optimize these feature vectors by introducing additional cross-entropy loss functions, such as Eq. (15) and Eq. (16), which encourage each feature vector to capture different characteristics of an input. It allows for performing many downstream tasks, including image classification and interpolation, within a unified framework at the same time. Additionally, LGAA introduces a robust generative replay network for producing samples which are statistically consistent with those previously learnt from various databases during the lifelong learning. This allows the inference models to capture the context information and implicitly model the correlation between the new task and the previously learnt knowledge. The following section introduces the algorithm used for training LGAA.

VI. Lifelong training algorithm for LGAA

In the following, we introduce a new training algorithm that enables LGAA to learn knowledge from a sequence of tasks without forgetting. The key idea of the proposed training algorithm consists of enabling two distinct optimization

(a) CelebA-CACD (b) CelebA-3DChair

Fig. 5. Interpolation results under CelebA to CACD and CelebA to 3DChair, respectively. We first select two images from two different datasets and calculate their latent codes using the inference models. Then we perform the interpolation on these latent codes, and the resulting interpolated code is fed into the decoder to produce the interpolated reconstruction.

procedures where we firstly update the parameters of the generative replay network and afterwards those of the whole model. Our algorithm is different from existing hybrid models by three aspects : 1) Existing hybrid models are only trained for a single dataset. However, the proposed LGAA is able to learn several data domains successively without forgetting; 2) Existing hybrid models usually train the generator and inference modules with a single optimisation function [39] or learn an optimal coupling between the generator and inference modules using adversarial learning [39], [40], [41], [42], [43], [44], [45], [47]. The proposed LGAA introduces a training algorithm consisting of two optimization procedures in which we first update the parameters of the generative replay network and afterwards those of the whole network architecture, including the inference models; 3) Existing hybrid models usually learn a single latent variable during the training, which is not applicable for a wide range of applications. The proposed LGAA learns both discrete and continuous variables, which can be used in classification and disentangled representation learning. In the following, we introduce the loss functions for LGAA in order to adapt its parameters for supervised learning, semi-supervised learning and unsupervised learning.

A. Supervised learning

During the training, we first update the parameters of the generator by minimizing the Wasserstein distance between the generator distribution and the target distribution $\mathbb{P}_{(\tilde{\mathbf{x}}^{t-1},\mathbf{x}^t)}$ at the t-th task learning, as stated by *Theorem* 2 and the subsequent theoretical derivations from Section IV. The adversarial objective function for the generator ($\mathcal{G}$) and discriminator ($\mathcal{D}$) is defined as :

$$\min_G \max_{\mathcal{D}} \mathcal{L}^G_{GAN}(\theta_t, \omega_t) \triangleq \mathbb{E}_{p(\mathbf{z}),p(\mathbf{d}),p(\mathbf{u})}[\mathcal{D}(G(\mathbf{u},\mathbf{z},\mathbf{d}))] - \mathbb{E}_{\mathbf{x}\sim\mathbb{P}_{(\tilde{\mathbf{x}}^{t-1},\mathbf{x}^t)}}[\mathcal{D}(\mathbf{x})], \quad (17)$$

where we omit the weighted penalty term, as in Eq. (12), for the sake of simplification.

In the second training procedure, we update the parameters of the whole model by maximizing the sample log-likelihood on the joint distribution of the generated and empirical data. The loss function (ELBO) is defined as :

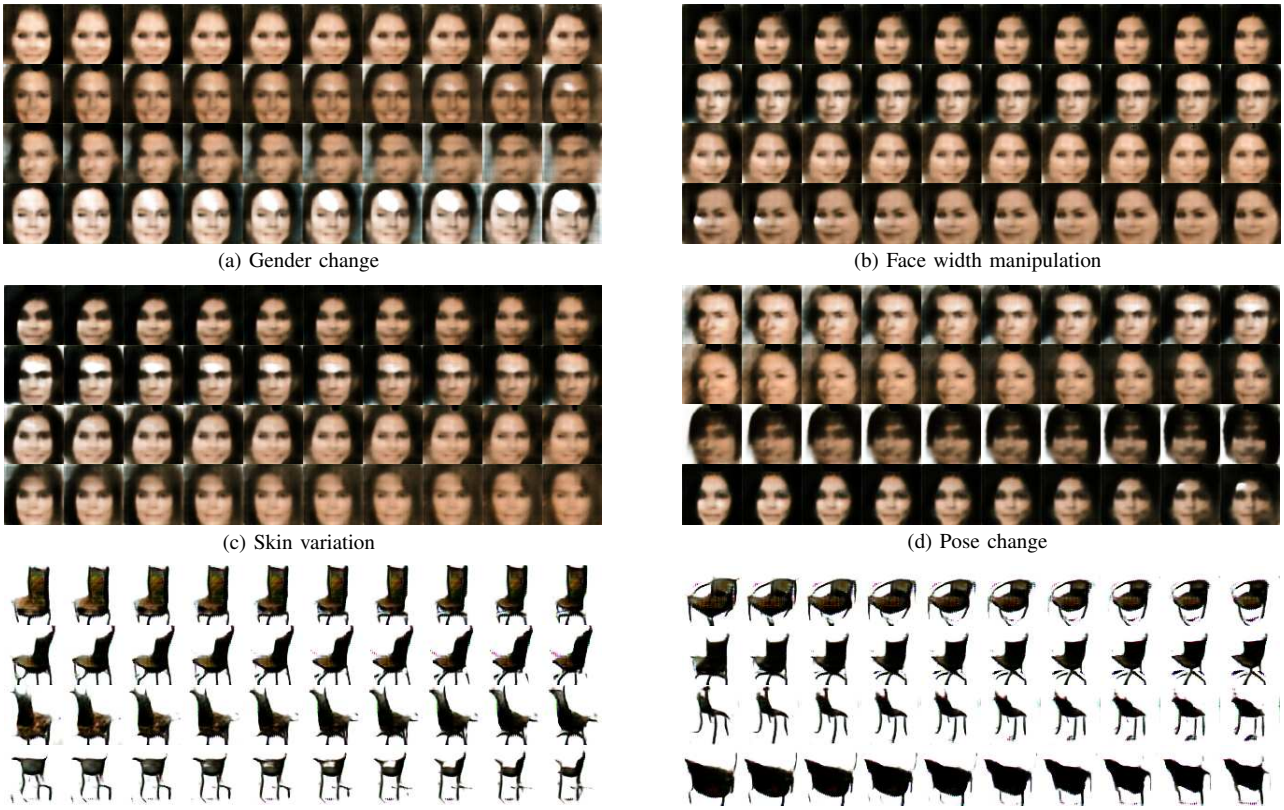
$$\mathcal{L}^{Sup}_{VAE}(\theta_t,\varsigma_t,\varepsilon_t,\delta_t) \triangleq \mathbb{E}_{q_{\varsigma,\varepsilon,\delta}(\mathbf{z},\mathbf{u},\mathbf{d}\,|\,\mathbf{x}^t)}\left[\log \frac{p_\theta(\mathbf{x}^t\,|\,\mathbf{z},\mathbf{u},\mathbf{d})}{q_{\varsigma,\varepsilon,\delta}(\mathbf{z},\mathbf{u},\mathbf{d}|\mathbf{x}^t)}\right] + \mathbb{E}_{q_{\varsigma,\varepsilon,\delta}(\mathbf{z},\mathbf{u},\mathbf{d}\,|\,\tilde{\mathbf{x}}^{t-1})}\left[\log \frac{p_\theta(\tilde{\mathbf{x}}^{t-1}\,|\,\mathbf{z},\mathbf{u},\mathbf{d})}{q_{\varsigma,\varepsilon,\delta}(\mathbf{z},\mathbf{u},\mathbf{d}\,|\,\tilde{\mathbf{x}}^{t-1})}\right]. \quad (18)$$

We also optimize the inference model $q_\delta(\mathbf{u}\,|\,\mathbf{x}^t)$ on both the real training samples from the t-th task, and the generative replay samples from the previously trained generator $p_{\theta_{t-1}}(\tilde{\mathbf{x}}^{t-1}\,|\,\mathbf{z},\mathbf{u},\mathbf{d})$, by minimizing the entropy loss defined in Eq. (16). During the testing phase, the inference model $q_\delta(\mathbf{u}\,|\,\mathbf{x})$ is used for classification.

We provide the pseudocode for the supervised learning of the LGAA in Algorithm 1, which can be summarized into three steps :

Step 1. Generative replay process : At the first task learning, the proposed models do not require the generative replay process to relive forgetting. We assume that the model was trained on $(i-1)$ tasks. In a new task learning (i), we perform the generative replay process to create a joint data distribution by combining $\mathbb{P}^i_{\mathbf{x}}$ and $\mathbb{P}_{\tilde{\mathbf{x}}^{(i-1)}}$, expressed as $\mathbb{P}_{(\mathbf{x}^i,\tilde{\mathbf{x}}^{(i-1)})} = \mathbb{P}_{\mathbf{x}^i} \bigcup \mathbb{P}_{\tilde{\mathbf{x}}^{(i-1)}}$ where $\bigcup$ represents the joint data distribution. Then we create the class label set $\mathbf{Y} = \mathbf{Y}^i \bigcup \tilde{\mathbf{Y}}^{(i-1)}$, corresponding to $\mathbb{P}_{(\mathbf{x}^i,\tilde{\mathbf{x}}^{(i-1)})}$, where $\mathbf{Y}^i$ denotes the class label set from the new task, and $\tilde{\mathbf{Y}}^{(i-1)}$ is formed by inferring generative replay samples using $F_{\delta_{(i-1)}}(\mathbf{u}\,|\,\mathbf{x})$. We also generate the domain label set $\tilde{\mathbf{A}}^{(i-1)}$ by using the sampling process described as follows : we firstly draw generative replay samples $\mathbf{x} \sim \mathbb{P}_{\tilde{\mathbf{x}}^{(i-1)}}$ and then latent variables $\mathbf{z} \sim q_{\varsigma_{(i-1)}}(\mathbf{z}\,|\,\mathbf{x})$. Then we can get the domain labels using $\mathbf{d} \sim q_{\varepsilon_{(i-1)}}(\mathbf{d}\,|\,\mathbf{z})$. $\mathbf{A}^i$ is the domain label set for the new task (i).

Step 2. Adversarial learning : We train the discriminator and generator on samples drawn from $\mathbb{P}_{(\mathbf{x}^i,\tilde{\mathbf{x}}^{(i-1)})}$ using $\min_G \max_D \mathcal{L}^G_{GAN}(\theta_i,\omega_i)$ at the i-th task learning.



(a) Gender change (b) Face width manipulation

(c) Skin variation (d) Pose change

(e) Chair style in 3D chairs (f) Chair size variation in 3D chairs

Fig. 6. Results obtained when manipulating the latent variables under the CelebA to 3D-Chair lifelong learning when considering the loss function from (23). We change a single latent variable in turns, in the latent space from -3.0 to 3.0 while fixing all others.

Step 3. The whole model optimization : In the i-th task learning, we firstly update all components on samples drawn from $\{\mathbb{P}_{(\mathbf{x}^i,\tilde{\mathbf{x}}^{(i-1)})}, \mathbf{A}, \mathbf{Y}\}$ using $L^{Sup}_{VAE}(\theta_i, \varsigma_i, \varepsilon_i, \delta_i)$. We then update $q_{\varepsilon_i}(\mathbf{d}\,|\,\mathbf{z})$ and $q_{\delta_i}(\mathbf{u}\,|\,\mathbf{x})$ on samples drawn from $\{\mathbb{P}_{(\mathbf{x}^i,\tilde{\mathbf{x}}^{(i-1)})}, \mathbf{A}, \mathbf{Y}\}$, using $\mathcal{L}_{\mathbf{d}}(\varepsilon_i)$ and $\mathcal{L}_{\mathbf{u}}(\delta_i)$, according to Eq. (15) and (16), respectively.

B. Semi-supervised learning

We apply our model to the lifelong semi-supervised learning setting where only a small subset of samples from each task is labeled, while the rest is unlabelled. We design different loss functions for the labelled and unlabelled samples. The generator training is the same as in Eq. (17). The whole model is optimized by the loss function for the labelled data without the inference model $q_\delta(\mathbf{u}\,|\,\mathbf{x})$:

$$\begin{aligned}
&\mathcal{L}^{S}_{VAE}(\theta_t, \varsigma_t, \varepsilon_t, \delta_t) \triangleq \mathbb{E}_{q_\varsigma(\mathbf{z}\,|\,\mathbf{x}^t), q_\varepsilon(\mathbf{d}|\mathbf{x}^t), p(\mathbf{y})}[\log p_\theta(\mathbf{x}^t\,|\,\mathbf{z}, \mathbf{d}, \mathbf{y})] \\
&- D_{KL}[q_\varsigma(\mathbf{z}\,|\,\mathbf{x}^t)\;||\;p(\mathbf{z})] - D_{KL}[q_\varepsilon(\mathbf{d}\,|\,\mathbf{z})\;||\;p(\mathbf{d})] \\
&- D_{KL}[q_\delta(\mathbf{u}\,|\,\mathbf{x}^t)\;||\;p(\mathbf{u})] \\
&+ \mathbb{E}_{q_\varsigma(\mathbf{z}\,|\,\tilde{\mathbf{x}}^{t-1}), q_\varepsilon(\mathbf{d}|\tilde{\mathbf{x}}^{t-1}), p(\mathbf{y})}[\log p_\theta(\tilde{\mathbf{x}}^{t-1}\,|\,\mathbf{z}, \mathbf{d}, \mathbf{y})] \\
&- D_{KL}[q_\varsigma(\mathbf{z}\,|\,\tilde{\mathbf{x}}^{t-1})\;||\;p(\mathbf{z})] \\
&- D_{KL}[q_\varepsilon(\mathbf{d}\,|\,\mathbf{z})\;||\;p(\mathbf{d})] \\
&- D_{KL}[q_\delta(\mathbf{u}\,|\,\mathbf{x})\;||\;p(\mathbf{u})]. \qquad (19)
\end{aligned}$$

In addition, we model the unlabeled data samples by using $\mathcal{L}_{VAE}(\theta_t, \varsigma_t, \varepsilon_t, \delta_t)$, where the discrete variable $\mathbf{u}$ is sampled from the Gumbel-softmax distribution whose probability vector is obtained by the encoder $q_\delta(\mathbf{u}\,|\,\mathbf{x})$. Then the semi-supervised loss used to train the hybrid model is defined as :

$$\mathcal{L}^{Semi}_{VAE} \triangleq \mathcal{L}^{S}_{VAE} + \beta \mathcal{L}_{VAE}, \qquad (20)$$

where β is used to control the importance of unsupervised learning when compared with the component associated with supervised learning, [33]. In addition, the entropy loss $\mathcal{L}_{\mathbf{u}}(\delta)$, as in Eq. (16), is considered for the labelled samples in order to enhance the prediction ability of $q_\delta(\mathbf{u}\,|\,\mathbf{x})$.

C. Unsupervised learning

In this section, we apply the proposed LGAA for the lifelong unsupervised learning setting, where the class labels for samples are not available. Similarly to the supervised learning framework and according to the theoretical derivations from Section IV, the first optimization stage of the training minimizes the Wasserstein distance between the generated data distribution and $\mathbb{P}_{(\tilde{\mathbf{x}}^{t-1}, \mathbf{x}^t)}$:

$$\begin{aligned}
\min_G \max_D \mathcal{L}^{U}_{GAN}(\theta_t, \omega_t) \triangleq\; &\mathbb{E}_{p(\mathbf{z}), p(\mathbf{d})}[D(G(\mathbf{z}, \mathbf{d}))] \\
&- \mathbb{E}_{\mathbf{x}\sim\mathbb{P}_{(\tilde{\mathbf{x}}^{t-1}, \mathbf{x}^t)}}[D(\mathbf{x})].
\end{aligned} \qquad (21)$$

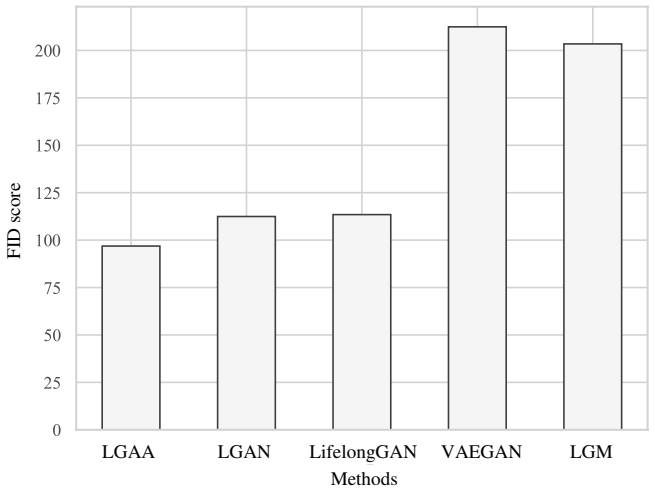


Fig. 7. Fréchet Inception Distance (FID) for generated images after CelebA and CACD lifelong learning.

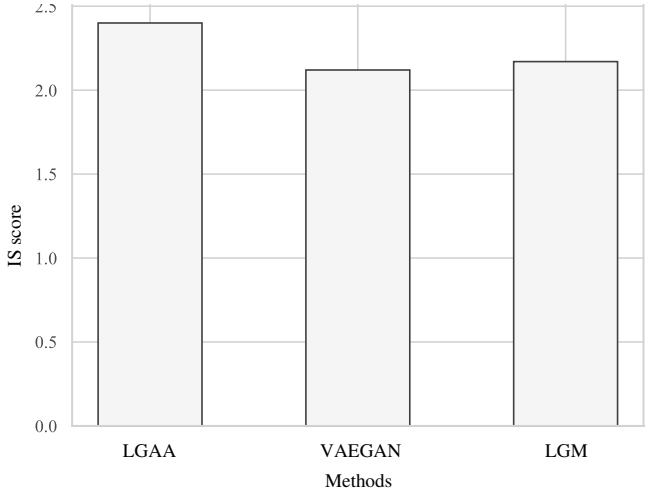


Fig. 8. Inception Score (IS) for image reconstructions after Cifar10 to MNIST lifelong learning.

At the second optimization stage of the training, the whole model is trained by:

$$\begin{aligned}\mathcal{L}^{U}_{VAE}(\theta_t,\varsigma_t,\varepsilon_t) \triangleq\ & \mathbb{E}_{q_\varsigma(\mathbf{z}\,|\,\tilde{\mathbf{x}}^{t-1}),q_\varepsilon(\mathbf{d}\,|\,\tilde{\mathbf{x}}^{t-1})}[\log p_\theta(\tilde{\mathbf{x}}^{t-1}\,|\,\mathbf{z},\mathbf{d})]\\ & - D_{KL}[q_\varsigma(\mathbf{z}\,|\,\tilde{\mathbf{x}}^{t-1})\,||\,p(\mathbf{z})]\\ & - D_{KL}[q_\varepsilon(\mathbf{d}\,|\,\mathbf{z})\,||\,p(\mathbf{d})]\\ & + \mathbb{E}_{q_\varsigma(\mathbf{z}\,|\,\mathbf{x}^{t}),q_\varepsilon(\mathbf{d}\,|\,\mathbf{x}^{t})}[\log p_\theta(\mathbf{x}^{t}\,|\,\mathbf{z},\mathbf{d})]\\ & - D_{KL}[q_\varsigma(\mathbf{z}\,|\,\mathbf{x}^{t})\,||\,p(\mathbf{z})]\\ & - D_{KL}[q_\varepsilon(\mathbf{d}\,|\,\mathbf{z})\,||\,p(\mathbf{d})]. \end{aligned} \tag{22}$$

For the generation process, we only have two variables, continuous $\mathbf{z}$, and $\mathbf{d}$ corresponding to the domain. In order to encourage learning disentangled representations across domains, we employ the Minimum Description Length (MDL) principle [50], [66] which replaces the second term, as well as the first before the last term from Eq. (22), resulting in :

$$\begin{aligned}\mathcal{L}^{U_{dis}}_{VAE}(\theta_t,\varsigma_t,\varepsilon_t) \triangleq\ & \mathbb{E}_{q_\varsigma(\mathbf{z}\,|\,\tilde{\mathbf{x}}^{t-1}),q_\varepsilon(\mathbf{d}|\tilde{\mathbf{x}}^{t-1})}[\log p_\theta(\tilde{\mathbf{x}}^{t-1}\,|\,\mathbf{z},\mathbf{d})]\\ & - \gamma|D_{KL}[q_\varsigma(\mathbf{z}\,|\,\tilde{\mathbf{x}}^{t-1})\,||\,p(\mathbf{z})] - C|\\ & - \mathbb{E}_{q_\varsigma(\mathbf{z}\,|\,\tilde{\mathbf{x}}^{t-1})}D_{KL}[q_\varepsilon(\mathbf{d}\,|\,\mathbf{z})\,||\,p(\mathbf{d}\,|\,\mathbf{z})]\\ & + \mathbb{E}_{q_\varsigma(\mathbf{z}\,|\,\mathbf{x}^{t}),q_\varepsilon(\mathbf{d}\,|\,\mathbf{x}^{t})}[\log p_\theta(\mathbf{x}^{t}\,|\,\mathbf{z},\mathbf{d})]\\ & - \gamma|D_{KL}[q_\varsigma(\mathbf{z}\,|\,\mathbf{x}^{t})\,||\,p(\mathbf{z})] - C|\\ & - D_{KL}[q_\varepsilon(\mathbf{d}\,|\,\mathbf{z})\,||\,p(\mathbf{d})], \end{aligned} \tag{23}$$

where γ and C are a multiplicative and a linear constant used for controlling the degree of disentanglement.

D. Minimizing the required memory

Instead of generating a collection of data samples by the generator, we can define a small memory buffer to preserve the current model's parameters before learning the next task. Then, the preserved model is used to generate a batch of data to be used for training together with data sampled from the database corresponding to the next task learning. The buffer is always fixed in size while increasing the number of tasks to be learnt during lifelong learning. After learning the current task, the old model parameters stored in the buffer will be replaced by the current model parameters. Then during the new task learning, the parameters from this buffer are used by the model for generating a batch of data corresponding to the stored model. The buffer used in our model can achieve a similar performance without the need to increase the required memory when adding new tasks to be learnt. This mechanism which reduces the memory required by the proposed model is illustrated in Fig. 2.

VII. Experiments

In this section, we investigate how the proposed Lifelong Generative Adversarial Autoencoder (LGAA) model learns meaningful and interpretable image representations under the lifelong learning of several tasks. We provide the source code at https://github.com/dtuzi123/Lifelong-Generative-Adversarial-Autoencoder .

A. Reconstruction and Interpolation results following unsupervised lifelong learning

We train the LGAA model using the loss functions $\mathcal{L}^{U}_{GAN}$ and $\mathcal{L}^{U}_{VAE}$ from equations (21) and (22), which contain adversarial and VAE learning terms, respectively, and we consider a learning rate of 0.001. The results for the unsupervised lifelong learning of CelebA [67] to CACD [68] are provided in Figures 3a-c where we show real images, generated images, and the image reconstructions for the images from Figures 3-a, respectively. Meanwhile, in Figures 4a-c we provide real images, generated images and the reconstructions of the real images from Fig. 4-a after the lifelong learning of CelebA to 3DChair [69]. From these results, it can be observed that the proposed approach can learn different data domains sequentially and provide good reconstruction results.

In the following we perform data interpolation experiments under the lifelong learning setting in order to evaluate the manifold continuity in the latent space. We call lifelong interpolation when the interpolation is performed between multiple data domains, by considering data from different databases, under the lifelong learning setting. We randomly select two

TABLE I
CLASSIFICATION RESULTS FOLLOWING THE LIFELONG LEARNING OF MNIST (M) AND FASHION (F) DATABASES.

Dataset	Lifelong	LGAA	LGAN [13]	LGM [35]	EWC [23]	Transfer	MeRGANs [34]
MNIST	M-F	**98.76**	98.41	97.29	37.7	40.63	98.34
MNIST	F-M	98.77	98.32	98.85	**99.12**	98.25	98.27
Fashion	M-F	**92.01**	91.42	91.71	91.38	91.01	91.12
Fashion	F-M	**89.24**	89.15	86.05	54.53	37.92	88.86

TABLE II
CLASSIFICATION RESULTS UNDER MNIST, SVHN, FASHION, INVERSEFASHION, INVERSEMNIST AND INVERSEMNIST LIFELONG LEARNING.

Dataset	LGAA	LGM [35]	MeRGANs [34]
MNIST	86.79	86.14	82.08
SVHN	52.18	23.87	34.20
Fashion	64.37	55.00	61.20
InverseFashion	78.60	49.83	74.17
InverseMNIST	97.33	86.49	93.44
CIFAR10	52.36	57.09	58.49
Average	**71.94**	59.74	67.27

TABLE III
SEMI-SUPERVISED CLASSIFICATION ERROR RESULTS ON MNIST DATABASE. LGAA AND LGAN ARE TRAINED UNDER THE MNIST TO FASHION LIFELONG LEARNING.

Methods	Lifelong	Error
LGAA	Yes	4.34
LGAN [13]	Yes	5.46
Neural networks (NN) [70]	No	10.7
Deep networks (CNN) [70]	No	6.45
TSVM [70]	No	5.38
CAE [70]	No	4.77
M1+TSVM [70]	No	4.24
M2 [70]	No	3.60
M1+M2 [70]	No	**2.40**
Semi-VAE [71]	No	2.88

images and then infer their discrete $\mathbf{u}$, and continuous $\mathbf{z}$ latent variables by using the inference model. Then we perform the interpolation on these latent variables and the resulting interpolated variables are used as inputs to the generator for reconstructing images and modelling smooth transitions between the chosen image pair. The interpolation results are shown in Fig. 5-a for CelebA to CACD, and in Fig. 5-b for CelebA to 3D-chair lifelong learning. We can observe from the images from the last two rows of Fig. 5-b that a chair is transformed into a human face, where the chair's seat and backside are smoothly changed into the eyes and hair of a person. This shows that the LGAA model can learn the joint latent space of two completely different data configurations.

B. Lifelong Disentangled Representations

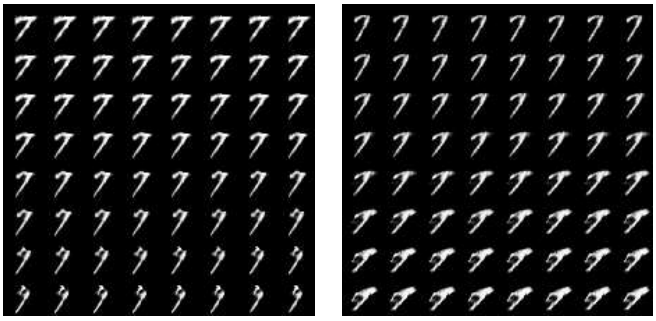
Within the unsupervised lifelong learning framework we train the LGAA model under the CelebA to 3D-Chairs lifelong learning by using the loss function from Eq. (23) in order to achieve unsupervised disentangled representations, as described in Section VI-C. We consider the multiplicative parameter $\gamma = 4$, while increasing the linear one C from 0.5 to 25.0 in Eq. (23), during the training. After the training, we change one of the dimensions of a continuous latent representation $\mathbf{z}$, inferred by using the inference model, for a given input, and then map it back to the visual data space by using the generator. The disentangled results are presented in Figures 6a-f, indicating changes in the appearance of gender, facial narrowing, skin tone variation, skin appearance, face pose, chairs' size, and chairs' style. These results show that the LGAA model can discover various disentangled representations in CelebA and 3D-Chairs databases following lifelong learning.

C. Quantitative assessment of the generated images quality

We use Inception score (IS) [72] and Fréchet Inception Distance (FID) [73] in order to evaluate the quality of generated images after lifelong learning. We train various methods under the CelebA to CACD lifelong learning setting. The FID scores, calculated between 5,000 target images and 5,000 generated images, where target images include samples from both CelebA and CACD databases, are provided in Fig. 7. The results by the proposed LGAA are compared with three other lifelong learning approaches: LGAN [74], LifelongGAN [75] and LGM [35]. We also consider Cifar10 [76] to MNIST database lifelong learning. The IS score on the reconstructions of 5,000 CIFAR10 testing samples, is provided in Fig. 8, where we compare with VAEGAN [45] and LGM [35]. The results from Fig. 8 show that GAN-based lifelong approaches achieve higher IS scores than VAE-based methods. This can be observed in the quality of the images generated, where VAE-based methods usually generate blurred images. The approach proposed in this paper produces higher-quality generative replay images and learns representations of data that other GAN-based lifelong learning approaches can not model.

D. Lifelong supervised learning

We compare LGAA with various methods under the lifelong supervised learning setting as described in Section VI-A. LGAN [13] typically trains a classifier (called Solver) on both images generated by the GAN module and form the training samples from the current task. We also consider an auxiliary classifier for LGM [35] by training it on the mixed data consisting of images generated by LGM and from the training samples of the current task.



(a) Changing the first dimension of $\mathbf{z}$ from -2 to 2 . (b) Changing the second dimension of $\mathbf{z}$ from -1 to 1.

Fig. 9. Reconstruction results on MNIST when changing a single continuous latent variable while fixing all others.

We train the LGAA model under the MNIST to Fashion [77] (M-F) lifelong learning and also when considering the reversed order of learning, as F-M. The classification results, after the lifelong training, are reported in Table I, where we compare with several models. We observe that GRM-based methods can prevent forgetting, and their performance relies on the quality of generative replay samples. GAN-based methods provide slightly better results than the VAE-based methods, since the generative replay network using a GAN can produce higher-quality data samples compared with models using VAEs. From Table I, we can also find that only the first task would suffer from the degenerated performance caused by the forgetting. When comparing with the results provided by the baselines, the proposed LGAA model achieves the best results on the first task, demonstrating that LGAA can generate high-quality samples which reduces the Wasserstein distance between the target and source distributions, resulting in a small target risk on the first task. Additionally, LGAA also outperforms other baselines when considering the average classification accuracy. In addition, from the results in Table I we can also observe that all models suffer from a significant degenerated performance when changing the lifelong learning order setting from M-F to F-M. The main reason for this phenomenon is that Fashion database contains more complex images than MNIST. When Fashion is used as the first task, we replay samples corresponding to Fashion at the second task learning in which the performance loss on Fashion is mainly caused by the GRM. In contrast, when MNIST is used as the first task, we do not see a significant drop in performance after lifelong learning. This is because the MNIST is a dataset with simpler images and the GRM can produce more realistic images corresponding to MNIST during the learning of the second task. These results also indicate that changing the order of tasks can influence the performance of the model.

We also investigate the performance of the proposed approach when learning a long sequence of tasks. We train various models under lifelong learning of MNIST, SVHN, Fashion, InverseFashion, InverseMNIST and CIFAR10, namely MSFIIC. For InverseFashion and InverseMNIST, we inverse the values of pixels x in each image as 255 - x. We report the classification results in Table II. We can observe that the proposed LGAA achieves the best results for almost every task when compared to the other methods. We also observe that GAN-based models can relieve forgetting better than the VAE-based models and outperform the latter in terms of the average classification accuracy.

TABLE IV
SEMI-SUPERVISED CLASSIFICATION ERROR RESULTS ON SVHN DATABASE, UNDER THE SVHN TO FASHION LIFELONG LEARNING.

Methods	Lifelong	Error
LGAA	Yes	60.36
LGAN [13]	Yes	63.25
kNN [70]	No	77.93
TSVN [70]	No	66.55
M1+KNN [70]	No	65.63
M1+TSVM [70]	No	54.33
M1+M2 [70]	No	**36.02**

E. Semi-supervised learning

For the semi-supervised lifelong training of LGAA, described in Section VI-B, we consider only a small number of labelled images from each database (1,000 for MNIST and 10,000 for Fashion) while the other images are not labelled (59,000 for MNIST and 50,000 for Fashion). The classification results following lifelong learning when using LGAA compared to other semi-supervised learning methods are provided in Table III. These results show that the proposed approach LGAA outperforms LGAN [13], under the semi-supervised learning setting.

In the following, we train LGAN and LGAA models under the lifelong learning of SVHN and Fashion. We only use 1,000 and 10,000 labelled training samples, while the remaining 72,257 and 50,000, for SVHN and Fashion, respectively, are unlabelled. The results from Table IV indicate that the proposed LGAA still outperforms LGAN for lifelong semi-supervised learning. In Tables III and IV we include the results for the methods which are only trained on one database, and we can observe that LGAA provides similar results to these methods, despite being at a great disadvantage when learning successively two databases. This demonstrates that the proposed approach can be potentially used in other semi-supervised learning applications such as in anomaly detection [78], [79].

F. Ablation study

In this section, we investigate the importance of different latent variables used in the proposed LGAA model.

The choice of the latent variables. First, we consider the proposed framework with only the continuous latent variable $\mathbf{z}$ as a baseline for comparison. Afterwards, we train the proposed framework with two inference models $\{\mathbf{z}, \mathbf{d}\}$. We investigate whether using the task inference model can degenerate the performance of the proposed approach. The average reconstruction error evaluated as the Mean Square Error (MSE) across all testing data is reported in Table V. We also train a simple classifier on the reconstructions produced by the model on all training samples. Then we evaluate the

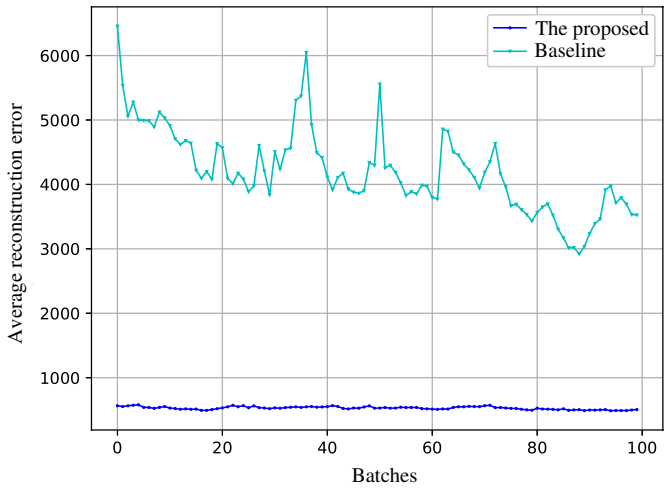


Fig. 10. Average reconstruction error on CACD during the lifelong CelebA to CACD.

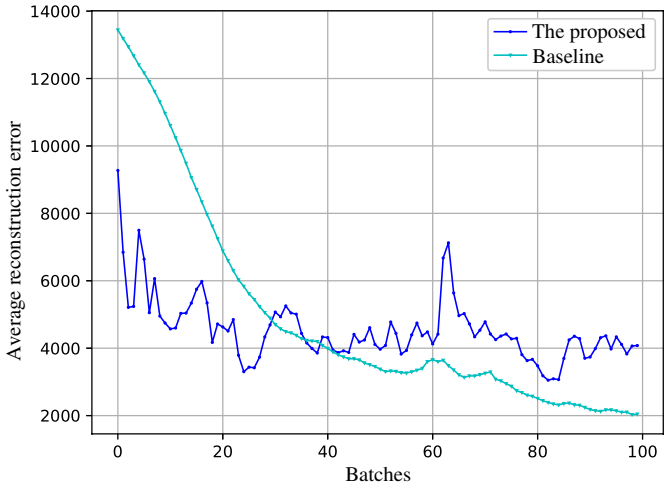


Fig. 11. Average reconstruction error on 3D-Chair during the lifelong CelebA to 3D-Chair.

classification accuracy on all testing samples using the classifier. The classification accuracy can reflect the quality of the reconstruction results and is reported in Table V. We observe that the proposed LGAA model performance, enabled with the task inference, does not deteriorate while the model learns the information from several databases. Then we perform the task inference experiments under the lifelong learning of MNIST followed by the Fashion, as well as the other way around under the F-M sequence. The results, when estimating the domain $\mathbf{d}$, are reported in Table VI. We find that the task-inference model can infer, in most cases, the task ID for the given data. This result also demonstrates that the latent variable $\mathbf{z}$ captures the task and implicitly domain information, which enables the task-inference model $q_\varepsilon(\mathbf{d}\,|\,\mathbf{z})$ to make accurate predictions.

Enforcing the disentanglement between the latent variables $\mathbf{z}$ and $\mathbf{u}$. We train the proposed model considering three latent vectors $\{\mathbf{u}, \mathbf{z}, \mathbf{d}\}$ under the lifelong supervised learning setting. After training, the inference model $q_\delta(\mathbf{u}\,|\,\mathbf{x})$ is used to make predictions. Then we change one dimension of the latent vector $\mathbf{z}$ inferred by $q_\varsigma(\mathbf{z}\,|\,\mathbf{x})$ while fixing the others. In Figures 9-a and 9-b we show the results for the images showing the digit '7' from MNIST, when changing the first and the second dimension of $\mathbf{z}$ within the ranges $[-2, 2]$ and $[-1, 1]$, respectively. From the results in Fig. 9 we observe that the latent variable $\mathbf{z}$ only represents the handwriting styles instead of the digit types in the images, which is modelled by the variable $\mathbf{u}$.

G. Transfer metric and transfer learning

By using the generative replay mechanism, the proposed approach can accelerate the training speed for learning the next tasks by transferring the previously learned knowledge. The transfer of knowledge between a past task and the currently given task is stronger when the new task is related to previously learned data distributions and then the model should be able to adapt quickly when learning the new task. In order to measure the task knowledge transferability in the network, we define a performance score for a model trained in the past, when shown a new batch sample while learning the (i)-th task :

$$F_\alpha(\mathbf{x}_{i,j}, f_{\theta_{i,j-1}}(\mathbf{x}_{i,j})), \quad j > 0, \tag{24}$$

where $f_{\theta_{i,j-1}}(\cdot)$ is the model updated using the $(j-1)$-th batch during the learning of the (i)-th task. Eq. (24) evaluates the performance for the (j)-th batch of training samples $\mathbf{x}_{i,j}$ on the (i)-th task, achieved by the model which was trained with the $(j-1)$-th data batch from the (i)-th task. $F_\alpha(\cdot,\cdot)$ is a performance metric which can be implemented by MSE or the classification accuracy, depending on the given task. This performance criterion has the ability to evaluate the capacity of transfer learning from one task to another.

In the following, we train the proposed model under the CelebA to CACD, and CelebA to 3D-Chair lifelong learning frameworks, respectively. We consider that the baseline is our model trained only on either CACD or 3D-Chair dataset. During the training, we evaluate the performance score $F_\alpha(\mathbf{x}_{i,j}, f_{\theta_{i,j-1}}(\mathbf{x}_{i,j}))$ from Eq. (24) when learning each data batch and use the average reconstruction error (MSE) as the performance metric $F_\alpha(\cdot,\cdot)$. The results are shown in Figures 10 and 11 for the CelebA to CACD database, and CelebA to 3D-Chair, respectively. From Fig. 10 we observe that our model provides reasonable reconstruction errors in the initial training phase of the second task while the learning for the baseline proceeds rather slowly. This is due to the fact the CACD and CelebA are both human face datasets, which means that they share similar facial feature information with each other. So the model can quickly adapt to a new task, when this is similar to one of the previously learnt tasks, as we can observe from the decrease of the average reconstruction errors during the learning. From Fig. 11 we observe that the proposed LGAA approach achieves lower reconstruction errors than the baseline at the beginning of the training. Then the baseline learns better than the proposed approach, while the proposed LGAA approach is actually a lifelong approach which knows the information from the datasets learnt in the past. The reason behind this is that a human face image dataset shares few features with the 3D-chair images, which have a completely

TABLE V
QUANTITATIVE EVALUATION ON THE REPRESENTATION LEARNING ABILITY

Methods	Lifelong	Dataset	Reconstruction error (MSE)	Accuracy (%)
LGAA	M-F	MNIST	4.75	**92.53**
Baseline	M-F	MNIST	**4.71**	91.29
LGAA	M-F	Fashion	17.44	67.66
Baseline	M-F	Fashion	**16.54**	**67.97**
LGAA	F-M	MNIST	**4.92**	**93.29**
Baseline	F-M	MNIST	5.14	92.34
LGAA	F-M	Fashion	**13.16**	**66.97**
Baseline	F-M	Fashion	14.78	66.45

TABLE VI
TASK INFERENCE ACCURACY ON MNIST AND FASHION EVALUATED USING THE TASK-INFERENCE MODEL.

Methods	Lifelong	Dataset	Accuracy (%)
LGAA	M-F	MNIST	**91.26**
LGAA	M-F	Fashion	91.12
LGAA	F-M	MNIST	94.25
LGAA	F-M	Fashion	**97.48**

distinct feature probabilistic representation. The knowledge learned by CelebA cannot have a positive transferable effect when learning an entirely different dataset.

VIII. CONCLUSION

In this paper, we propose a new approach for lifelong learning, called the Lifelong Generative Adversarial Autoencoder (LGAA) which benefits from the advantages of enabling GAN and VAE generative deep learning methods into a unified lifelong learning framework. The proposed LGAA can learn meaningful representations across domains without forgetting under lifelong learning. The proposed lifelong framework can be used in a wide range of applications, including data classification, semi-supervised learning, data reconstruction, generation and inter-domain interpolation. Another contribution of this paper consists in developing a theoretical framework for analyzing the information loss of GRM-based models under lifelong learning. The proposed theoretical analysis provides new insights into the forgetting behaviour of the GRM-based methods, resulting in guidelines for robust lifelong learning model design.

One limitation of this study is that the proposed LGAA is still limited when learning an infinite number of tasks since GAN would suffer from the mode collapse [47] when learning several entirely different datasets. This eventually results in catastrophic forgetting given that the model can not endlessly rely on reasonable good generative replay samples during its lifelong learning. This inspires us to explore in the future a dynamic expansion mechanism for the proposed LGAA, which would enable the model to deal with more tasks by dynamically increasing the model's capacity whenever necessary.

APPENDIX A
PROOF OF *Theorem* 1

By using mathematical induction over the lifelong learning of the probabilistic representations associated with various tasks, the marginal distribution is rewritten as:

$$\begin{aligned}
p(\tilde{\mathbf{x}}^t) &= \iint p(\tilde{\mathbf{x}}^t|\tilde{\mathbf{x}}^{t-1},\mathbf{x}^t)p(\tilde{\mathbf{x}}^{t-1},\mathbf{x}^t)d\tilde{\mathbf{x}}^{t-1}d\mathbf{x}^t \\
&= \iint p(\tilde{\mathbf{x}}^t|\tilde{\mathbf{x}}^{t-1},\mathbf{x}^t)p(\tilde{\mathbf{x}}^{t-1})p(\mathbf{x}^t)d\tilde{\mathbf{x}}^{t-1}d\mathbf{x}^t \\
&= \iiiint p(\tilde{\mathbf{x}}^t|\tilde{\mathbf{x}}^{t-1},\mathbf{x}^t)p(\tilde{\mathbf{x}}^{t-1}|\tilde{\mathbf{x}}^{t-2},\mathbf{x}^{t-1})\cdot \\
&\cdot p(\tilde{\mathbf{x}}^{t-2})p(\mathbf{x}^t)p(\mathbf{x}^{t-1})d\tilde{\mathbf{x}}^{t-1}d\mathbf{x}^t d\tilde{\mathbf{x}}^{t-2}d\mathbf{x}^{t-1} \\
&= \int\dots\int p(\tilde{\mathbf{x}}^1)\prod_{i=0}^{t-2}p(\tilde{\mathbf{x}}^{t-i}|\tilde{\mathbf{x}}^{t-i-1},\mathbf{x}^{t-i})\prod_{i=0}^{t-2}p(\mathbf{x}^{t-i}) \\
&\quad d\tilde{\mathbf{x}}^1\dots d\tilde{\mathbf{x}}^{t-1}d\mathbf{x}^2\dots d\mathbf{x}^t
\end{aligned} \tag{25}$$

□

This corresponds to Eq. (5) and proves *Theorem* 1.

APPENDIX B
PROOF OF LEMMA 1

In order to have $p(\tilde{\mathbf{x}}^t)=\prod_{i=1}^{t}p(\mathbf{x}^i)$, we must firstly satisfy the following condition:

$$p(\tilde{\mathbf{x}}^t\,|\,\tilde{\mathbf{x}}^{t-1},\mathbf{x}^t)=1\Rightarrow p(\tilde{\mathbf{x}}^t)=p(\tilde{\mathbf{x}}^{t-1},\mathbf{x}^t) \tag{26}$$

where the right hand side can be decomposed as $p(\tilde{\mathbf{x}}^{t-1})p(\mathbf{x}^t)$ since $p(\tilde{\mathbf{x}}^{t-1})$ is independent from $p(\mathbf{x}^t)$. We further decompose $p(\tilde{\mathbf{x}}^{t-1})=p(\tilde{\mathbf{x}}^{t-2})p(\mathbf{x}^{t-1})$ if $p(\tilde{\mathbf{x}}^{t-1}\,|\,\tilde{\mathbf{x}}^{t-2},\mathbf{x}^{t-1})=1$. By considering all decompositions through induction :

$$\begin{aligned}
&\textstyle\prod_{i=0}^{t-2}p(\tilde{\mathbf{x}}^{t-i}\,|\,\tilde{\mathbf{x}}^{t-i-1},\mathbf{x}^{t-i})=1, \\
&p(\tilde{\mathbf{x}}^1)=p(\mathbf{x}^1)\Rightarrow p(\tilde{\mathbf{x}}^t)=p(\mathbf{x}^1,\dots,\mathbf{x}^t)
\end{aligned} \tag{27}$$

□

This proves Lemma 1.

APPENDIX C
PROOF OF LEMMA 2

Firstly, we consider that the model has finished learning the (j)-th task and then we evaluate the risk bound for $\mathbb{P}_{\mathbf{x}^j}$ and $\mathbb{P}_{\tilde{\mathbf{x}}}^{(j,j)}$, according to *Theorem* 2, as :

$$\begin{aligned}
\mathcal{R}\left(h,h_{(\nu_j)}\right) \le\ & \mathcal{R}\left(h,h_{(\nu_{(j,j)})}\right)+W\left(\nu_j,\nu_{(j,j)}\right) \\
&+\sqrt{2\log\left(\frac{1}{\delta}\right)/a'}\left(\sqrt{\frac{1}{n_{\nu_j}}}+\sqrt{\frac{1}{n_{\nu_{(j,j)}}}}\right) \\
&+D(\mathcal{R}(h,h_{(\nu_j)})+\mathcal{R}(h,h_{(\nu_{(j,j)})}))
\end{aligned} \tag{28}$$

where $\{\nu_{(j,j)}\in\mathbb{R}^s|\nu_{(j,j)}\sim\mathbb{P}_{\tilde{\mathbf{x}}}^{(j,j)}\}$ and $n_{\nu_{(j,j)}}$ is the corresponding sample size. $h_{(\nu_{(j,j)})}$ is the identity function for samples drawn from $\nu_{(j,j)}$. $D(\cdot)$ is the optimal combined error. Eq. (28) describes the risk bound after the (j)-th task learning. In the following, we consider $\mathbb{P}_{\tilde{\mathbf{x}}}^{(j,j)}$ to be the target distribution

and $\mathbb{P}_{\tilde{\mathbf{x}}}^{(j+1,j)}$ to be the source distribution and we have the risk bound :

$$\begin{aligned}\mathcal{R}\left(h, h_{(\nu_{(j,j)})}\right) \leq \mathcal{R}\left(h, h_{(\nu_{(j+1,j)})}\right) + W\left(\nu_{(j,j)}, \nu_{(j+1,j)}\right) \\ + \sqrt{2\log\left(\frac{1}{\delta}\right)/a'}\left(\sqrt{\frac{1}{n_{\nu_{(j,j)}}}} + \sqrt{\frac{1}{n_{\nu_{(j+1,j)}}}}\right) \\ + D(\mathcal{R}(h, h_{(\nu_{(j,j)})}) + \mathcal{R}(h, h_{(\nu_{(j+1,j)})}))\end{aligned} \tag{29}$$

Similarly, we can repeat the risk calculation, until the t-th task learning :

$$\begin{aligned}&\mathcal{R}\left(h, h_{(\nu_{(j+1,j)})}\right) \leq \mathcal{R}\left(h, h_{(\nu_{(j+1,j+2)})}\right) \\ &+ W\left(\nu_{(j+1,j)}, \nu_{(j+2,j)}\right) \\ &+ \sqrt{2\log\left(\frac{1}{\delta}\right)/a'}\left(\sqrt{\frac{1}{n_{\nu_{(j+1,j)}}}} + \sqrt{\frac{1}{n_{\nu_{(j+2,j)}}}}\right) \\ &+ D(\mathcal{R}(h, h_{(\nu_{(j+1,j)})}) + \mathcal{R}(h, h_{(\nu_{(j+2,j)})})) \\ &\cdots \\ &\mathcal{R}\left(h, h_{(\nu_{(t-1,j)})}\right) \leq \mathcal{R}\left(h, h_{(\nu_{(t,j)})}\right) + W\left(\nu_{(t-1,j)}, \nu_{(t,j)}\right) \\ &+ \sqrt{2\log\left(\frac{1}{\delta}\right)/a'}\left(\sqrt{\frac{1}{n_{\nu_{(t-1,j)}}}} + \sqrt{\frac{1}{n_{\nu_{(t,j)}}}}\right) \\ &+ D(\mathcal{R}(h, h_{(\nu_{(t-1,j)})}) + \mathcal{R}(h, h_{(\nu_{(t,j)})})\end{aligned} \tag{30}$$

Then we replace each risk into another successively for all the above inequalities, resulting in :

$$\begin{aligned}&\mathcal{R}\left(h, h_{(\nu_j)}\right) \leq \mathcal{R}\left(h, h_{(\nu_{(t,j)})}\right) + \sum_{k=j-1}^{t-1}\Big\{W\left(\nu_{(k,j)}, \nu_{(k+1,j)}\right) \\ &+ \sqrt{2\log\left(\frac{1}{\delta}\right)/a'}\left(\sqrt{\frac{1}{n_{\nu_{(k,j)}}}} + \sqrt{\frac{1}{n_{\nu_{(k+1,j)}}}}\right) \\ &+ D(\mathcal{R}(h, h_{(\nu_{(k,j)})}) + \mathcal{R}(h, h_{(\nu_{(k+1,j)})})\Big\} \qquad \square\end{aligned} \tag{31}$$

This proves Lemma 2.

Appendix D
The derivation of $\mathcal{L}_{\text{VAE}}$

In the following we consider modeling a single task:

$$\log p(\mathbf{x}) = \log \mathbb{E}_{q_{\varsigma,\varepsilon,\delta}(\mathbf{z},\mathbf{u},\mathbf{d}|\mathbf{x})}\left[\frac{p_\theta(\mathbf{x},\mathbf{z},\mathbf{u},\mathbf{d})}{q_{\varsigma,\varepsilon,\delta}(\mathbf{z},\mathbf{u},\mathbf{d}|\mathbf{x})}\right] \tag{32}$$

Then, according to Jensens' inequality, we have:

$$\log p(\mathbf{x}) \geq \mathbb{E}_{q_{\varsigma,\varepsilon,\delta}(\mathbf{z},\mathbf{u},\mathbf{d}|\mathbf{x})}\left[\log \frac{p_\theta(\mathbf{x},\mathbf{z},\mathbf{u},\mathbf{d})}{q_{\varsigma,\varepsilon,\delta}(\mathbf{z},\mathbf{u},\mathbf{d}|\mathbf{x})}\right] \tag{33}$$

$$\begin{aligned}\mathcal{L}_{\text{VAE}}(\theta,\varsigma,\varepsilon,\delta) &= \mathbb{E}_{q_{\varsigma,\varepsilon,\delta}(\mathbf{z},\mathbf{u},\mathbf{d}|\mathbf{x})}\log\left[\frac{p_\theta(\mathbf{x},\mathbf{z},\mathbf{u},\mathbf{d})}{q_{\varsigma,\varepsilon,\delta}(\mathbf{z},\mathbf{u},\mathbf{d}|\mathbf{x})}\right] \\ &= \mathbb{E}_{q_\delta(\mathbf{u}|\mathbf{x})q_\varepsilon(\mathbf{d}|\mathbf{z})q_\varsigma(\mathbf{z}|\mathbf{x})}\log\left[\frac{p_\theta(\mathbf{x}|\mathbf{z},\mathbf{u},\mathbf{d})p(\mathbf{d}|\mathbf{z})p(\mathbf{z})p(\mathbf{u})}{q_\delta(\mathbf{u}|\mathbf{x})q_\varepsilon(\mathbf{d}|\mathbf{z})q_\varsigma(\mathbf{z}|\mathbf{x})}\right]\end{aligned} \tag{34}$$

Since $q_\varepsilon(\mathbf{d}|\mathbf{x})$ takes an image as the input that is high-dimensional data, we usually process this input data using a CNN network with several convolutional layers, which leads to more parameters. To further reduce the number of parameters, we consider replacing $q_\varepsilon(\mathbf{d}|\mathbf{x})$ by using a lightweight inference model $q_\varepsilon(\mathbf{d}|\mathbf{z})$ implemented by a simple fully connected network with fewer parameters since $\mathbf{z}$ is a low-dimensional latent representation. Then, Eq. (34) is rewritten as :

$$\begin{aligned}\mathcal{L}_{\text{VAE}}(\theta,\varsigma,\varepsilon,\delta) = \mathbb{E}_{q_{\varsigma,\varepsilon,\delta}(\mathbf{z},\mathbf{d},\mathbf{u}\,|\,\mathbf{x})}\log[p_\theta(\mathbf{x}\,|\,\mathbf{z},\mathbf{d},\mathbf{u})] \\ - D_{KL}[q_\varsigma(\mathbf{z}\,|\,\mathbf{x})\,||\,p(\mathbf{z})] \\ - D_{KL}[q_\varepsilon(\mathbf{d}\,|\,\mathbf{z})\,||\,p(\mathbf{d})] \\ - D_{KL}[q_\delta(\mathbf{u}\,|\,\mathbf{x})\,||\,p(\mathbf{u})]\end{aligned} \tag{35}$$

where we have separated the Kullback-Leibler (KL) divergence components for the continuous $\mathbf{z}$ space, as well as for the discrete and domain spaces $\mathbf{u}$ and $\mathbf{d}$, respectively. Meanwhile, $\theta, \varsigma, \varepsilon, \delta$ represent the parameters of the corresponding networks.

References

[1] 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.

[2] D. Erhan, C. Szegedy, A. Toshev, and D. Anguelov, "Scalable object detection using deep neural networks," in *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2014, pp. 2147–2154.

[3] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," in *Advances Neural Inf. Proc. Systems (NeurIPS)*, 2015, pp. 91–99.

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

[5] Z. Chen, N. Ma, and B. Liu, "Lifelong learning for sentiment classification," in *Proc. of the Annual Meeting of the Assoc. for Comp. Linguistics and Int. Joint Conf. on Natural Language Processing*, 2015, pp. 750–756.

[6] J. Fagot and R. G. Cook, "Evidence for large long-term memory capacities in baboons and pigeons and its implications for learning and the evolution of cognition," *Proc. of the National Academy of Sciences (PNAS)*, vol. 103, no. 46, pp. 17 564–17 567, 2006.

[7] A. Rannen, R. Aljundi, M. Blaschko, and T. Tuytelaars, "Encoder based lifelong learning," in *Proc. of the IEEE Int. Conf. on Computer Vision (ICCV)*, 2017, pp. 1320–1328.

[8] C. Tessler, S. Givony, T. Zahavy, D. J. Mankowitz, and S. Mannor, "A deep hierarchical approach to lifelong learning in Minecraft," in *Proc. AAAI Conf. on Artificial Intelligence*, 2017, pp. 1553–1561.

[9] J. Yoon, E. Yang, J. Lee, and S. J. Hwang, "Lifelong learning with dynamically expandable networks," in *Int. Conf. on Learning Representations (ICLR), arXiv preprint arXiv:1708.01547*, 2017.

[10] R. Aljundi, E. Belilovsky, T. Tuytelaars, L. Charlin, M. Caccia, M. Lin, and L. Page-Caccia, "Online continual learning with maximal interfered retrieval," in *Advances Neural Inf. Proc. Systems (NeurIPS)*, 2019, pp. 11 849–11 860.

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

[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," *arXiv preprint arXiv:1902.10486*, 2019.

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

[14] 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.

[15] J.-Y. Zhu, R. Zhang, D. Pathak, T. Darrell, A. A. Efros, O. Wang, and E. Shechtman, "Multimodal image-to-image translation by enforcing bicycle consistency," in *Advances Neural Inf. Proc. Systems (NeurIPS)*, 2017, pp. 465–476.

[16] A. Oring, Z. Yakhini, and Y. Hel-Or, "Autoencoder image interpolation by shaping the latent space," in *Proc. of Int. Conf. on Machine Learning (ICML), vol. PMLR 139*, 2021, pp. 8281–8290.

[17] H. Kim and A. Mnih, "Disentangling by factorising," in *Proc. of Int. Conf. on Machine Learning (ICML), vol. PMLR 80*, 2018, pp. 2649–2658.

[18] F. Ye and A. G. Bors, "Deep mixture generative autoencoders," *IEEE Trans. on Neural Networks and Learning Systems*, vol. 33, no. 10, pp. 5789–5803, 2022.

[19] 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 Neural Inf. Proc. Systems (NeurIPS)*, 2018, pp. 9873–9883.

[20] W. Dai, Q. Yang, G. R. Xue, and Y. Yu, "Boosting for transfer learning," in *Proc. of Int. Conf. on Machine Learning (ICML), vol. PMLR 227*, 2007, pp. 193–200.

[21] G. Hinton, O. Vinyals, and J. Dean, "Distilling the knowledge in a neural network," in *NeurIPS Workshop in Deep Learning*, 2014.

[22] H. Jung, J. Ju, M. Jung, and J. Kim, "Less-forgetting learning in deep neural networks," *arXiv preprint arXiv:1607.00122*, 2016.

[23] 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.

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

[25] R. Polikar, L. Upda, S. 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.

[26] 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.

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

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

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

[30] F. Ye and A. G. Bors, "Lifelong learning of interpretable image representations," in *Proc. Int. Conf. on Image Processing Theory, Tools and Applications (IPTA)*, 2020, pp. 1–6.

[31] ——, "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.

[32] ——, "Lifelong twin generative adversarial networks," in *Proc. IEEE Int. Conf. on Image Processing (ICIP)*, 2021, pp. 1289–1293.

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

[34] 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 Neural Inf. Proc. Systems (NeurIPS)*, 2018, pp. 5962–5972.

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

[36] B. Heo, M. Lee, S. Yun, and J. Y. Choi, "Knowledge distillation with adversarial samples supporting decision boundary," in *Proc. AAAI Conf. on Artificial Intelligence*, 2019, pp. 3771–3778.

[37] 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.

[38] ——, "Learning joint latent representations based on information maximization," *Information Sciences*, vol. 567, pp. 216–236, 2021.

[39] A. B. L. Larsen, S. K. Sønderby, H. Larochelle, and O. Winther, "Autoencoding beyond pixels using a learned similarity metric," in *Proc. of Int. Conf. on Machine Learning (ICML), vol. PMLR 48*, 2015, pp. 1558–1566.

[40] A. Makhzani, J. Shlens, N. Jaitly, I. Goodfellow, and B. Frey, "Adversarial autoencoders," in *Int. Conf. on Learniong Representations (ICLR), arXiv preprint arXiv:1511.05644.*, 2016.

[41] L. Chen, S. Dai, Y. Pu, C. Li, Q. Su, and L. Carin, "Symmetric variational autoencoder and connections to adversarial learning," in *Proc. Int. Conf. on Artificial Intel. and Statistics (AISTATS) , vol. PMLR 84*, 2018, pp. 661–669.

[42] J. Donahue, P. Krähenbühl, and T. Darrell, "Adversarial feature learning," in *Int. Conf. on Learning Representations (ICLR), arXiv preprint arXiv:1605.09782*, 2017.

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

[44] C. Li, H. Liu, C. Chen, Y. Pu, L. Chen, R. Henao, and L. Carin, "Alice: Towards understanding adversarial learning for joint distribution matching," in *Advances Neural Inf. Proc. Systems (NeurIPS)*, 2017, pp. 5495–5503.

[45] L. Mescheder, S. Nowozin, and A. Geiger, "Adversarial variational Bayes: Unifying variational autoencoders and generative adversarial networks," in *Proc. Int. Conf. on Machine Learning (ICML), vol. PMLR 70*, 2017, pp. 2391–2400.

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

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

[48] H. Huang, R. He, Z. Sun, and T. Tan, "Introvae: Introspective variational autoencoders for photographic image synthesis," in *Advances Neural Inf. Proc. Systems (NeurIPS)*, 2018, pp. 52–63.

[49] A. Kumar, P. Sattigeri, and A. Balakrishnan, "Variational inference of disentangled latent concepts from unlabeled observations," in *Int. Conf. on Learning Representations (ICLR), arXiv preprint arXiv:1711.00848*, 2018.

[50] C. P. Burgess, I. Higgins, A. Pal, L. Matthey, N. Watters, G. Desjardins, and A. Lerchner, "Understanding disentangling in β-VAE," in *NeurIPS Workshop on Learning Disentangled Representation.*, 2017.

[51] E. Dupont, "Learning disentangled joint continuous and discrete representations," in *Advances Neural Inf. Proc. Systems (NeurIPS)*, 2018, pp. 710–720.

[52] 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 *Int. Conf. on Learning Representations (ICLR)*, 2017.

[53] T. Q. Chen, X. Li, R. B. Grosse, and D. K. Duvenaud, "Isolating sources of disentanglement in variational autoencoders," in *Advances Neural Inf. Proc. Systems (NeurIPS)*, 2018, pp. 2615–2625.

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

[55] Y. Jeong and H. O. Song, "Learning discrete and continuous factors of data via alternating disentanglement," in *Proc. Int. Conf. on Machine Learning (ICML), vol. PMLR 97*, 2019, pp. 3091–3099.

[56] S. Nowozin, B. Cseke, and R. Tomioka, "f-gan: Training generative neural samplers using variational divergence minimization," in *Advances Neural Inf. Proc. Systems (NeurIPS)*, 2016, pp. 271–279.

[57] M. Arjovsky, S. Chintala, and L. Bottou, "Wasserstein generative adversarial networks," in *Proc. Int. Conf. on Machine Learning (ICML), vol. PMLR 70*, 2017, pp. 214–223.

[58] I. Redko, A. Habrard, and M. Sebban, "Theoretical analysis of domain adaptation with optimal transport," in *Joint European Conf. on Machine Learning and Knowledge Discovery in Databases, vol. LNCS 10535*, 2017, pp. 737–753.

[59] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, and A. C. Courville, "Improved training of Wasserstein GANs," in *Advances Neural Inf. Proc. Systems (NeurIPS)*, 2017, pp. 5767–5777.

[60] D. J. Rezende, S. Mohamed, and D. Wierstra, "Stochastic backpropagation and approximate inference in deep generative models," in *Proc. Int. Conf. on Machine Learning (ICML), vol. PMLR 32*, 2014, pp. 1278–1286.

[61] E. J. Gumbel and J. Lieblein, "Some applications of extreme-value methods," *The American Statistician*, vol. 8, no. 5, pp. 14–17, 1954.

[62] C. Maddison, D. Tarlow, and T. Minka, "A* sampling," in *Advances Neural Inf. Proc. Systems (NeurIPS)*, 2014, pp. 3086–3094.

[63] E. Jang, S. Gu, and B. Poole, "Categorical reparameterization with Gumbel-Softmax," in *Int. Conf. on Learning Representations (ICLR), arXiv preprint arXiv:1611.01144*, 2017.

[64] C. J. Maddison, A. Mnih, and Y. W. Teh, "The concrete distribution: A continuous relaxation of discrete random variables," in *Int. Conf.*

on Learning Representations (ICLR), arXiv preprint arXiv:1611.00712, 2016.
[65] X. Wang, R. Zhang, Y. Sun, and J. Qi, "KDGAN: knowledge distillation with generative adversarial networks," in *Advances Neural Inf. Proc. Systems (NeurIPS)*, 2018, pp. 775–786.
[66] J. Rissanen, "Modeling by shortest data description," *Automatica*, vol. 14, no. 5, pp. 465–471, 1978.
[67] 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.
[68] 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.
[69] 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.
[70] D. P. Kingma, S. Mohamed, D. J. Rezende, and M. Welling, "Semi-supervised learning with deep generative models," in *Advances Neural Inf. Proc. Systems (NeurIPS)*, 2014, pp. 3581–3589.
[71] S. Narayanaswamy, T. 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 Neural Inf. Proc. Systems (NeurIPS)*, 2017, pp. 5925–5935.
[72] T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Radford, and X. Chen, "Improved techniques for training GANs," in *Advances Neural Inf. Proc. Systems (NeurIPS)*, 2016, pp. 2234–2242.
[73] M. Heusel, H. Ramsauer, T. Unterthiner, B. Nessler, and S. Hochreiter, "Gans trained by a two time-scale update rule converge to a local Nash equilibrium," in *Advances Neural Inf. Proc. Systems (NeurIPS)*, 2017, pp. 6626–6637.
[74] 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.
[75] A. Seff, A. Beatson, D. Suo, and H. Liu, "Continual learning in generative adversarial nets," *arXiv preprint arXiv:1705.08395*, 2017.
[76] A. Krizhevsky and G. Hinton, "Learning multiple layers of features from tiny images," Tech. Rep., 2009.
[77] H. Xiao, K. Rasul, and R. Vollgraf, "Fashion-MNIST: a novel image dataset for benchmarking machine learning algorithms," *arXiv preprint arXiv:1708.07747*, 2017.
[78] G. Pang, C. Shen, and A. van den Hengel, "Deep anomaly detection with deviation networks," in *Proc. of the ACM SIGKDD Int. Conf. on Knowledge Discovery & Data Mining*, 2019, pp. 353–362.
[79] Y. Zhou, X. Song, Y. Zhang, F. Liu, C. Zhu, and L. Liu, "Feature encoding with autoencoders for weakly supervised anomaly detection," *IEEE Trans. on Neural Networks and Learning Systems*, vol. 33, no. 6, pp. 2454–2465, 2022.

Fei Ye is currently a PHD candidate in computer science from the University of York. 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, U.K., where he is currently an Associate Professor. Dr. Bors was a Research Scientist at Tampere Univ. of Technology, Finland, a Visiting Scholar at the Univ. of California at San Diego (UCSD), and an Invited Professor at the Univ. of Montpellier, France. Dr. Bors has authored and co-authored more than 160 research papers, including 40 in journals. His research interests include computational intelligence, computer vision, pattern recognition and image processing.

Dr. Bors was a member of the organizing committees for IEEE WIFS 2021, IPTA 2020, IEEE ICIP 2018, BMVC 2016, IPTA 2014, CAIP 2013, and IEEE ICIP 2001. 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 a Co-Guest Editor for a special issue on machine vision for the International Journal for Computer Vision in 2018 and the Journal of Pattern Recognition in 2015.