Dynamic Self-Supervised Teacher-Student Network Learning

Fei Ye and Adrian G. Bors, Senior Member, IEEE
Department of Computer Science, University of York, York YO10 5GH, UK
E-mail: fy689@york.ac.uk, adrian.bors@york.ac.uk

Abstract—Lifelong learning (LLL) represents the ability of an artificial intelligence system to learn successively a sequence of different databases. In this paper we introduce the Dynamic Self-Supervised Teacher-Student Network (D-TS), representing a more general LLL framework, where the Teacher is implemented as a dynamically expanding mixture model which automatically increases its capacity to deal with a growing number of tasks. We propose the Knowledge Discrepancy Score (KDS) criterion for measuring the relevance of the incoming information characterizing a new task when compared to the existing knowledge accumulated by the Teacher module from its previous training. The KDS ensures a light Teacher architecture while also enabling to reuse the learned knowledge whenever appropriate, accelerating the learning of given tasks. The Student module is implemented as a lightweight probabilistic generative model. We introduce a novel self-supervised learning for the Student that allows to capture cross-domain latent representations from the entire knowledge accumulated by the Teacher as well as from novel data. We perform several experiments which show that D-TS can achieve the state of the art results in LLL while requiring fewer parameters than other methods.

Index Terms—Lifelong Learning, Teacher-Student framework, Self-Supervised learning, Representation Learning.

1 Introduction

Lifelong learning (LLL) is an essential characteristic of all living beings, which enables them to adapt to their environments through learning from experiences. The lifelong learning is also an important desired function of an artificial intelligence system, capable of continually acquiring and assimilating novel concepts from a dynamically changing data stream without forgetting previously learnt knowledge. In recent years, machine learning models have surpassed human-level performance in individual tasks [1], such as in Atari gaming [2] or in image recognition [3]. However, LLL remains challenging to any machine learning model, usually resulting in a significantly degraded performance on the previously learnt tasks when successively retraining on multiple tasks. This phenomenon is called catastrophic forgetting [4].

Many existing research studies have proposed to relieve forgetting by using episodic memory systems [5], [6], [7], [8], [9] or Generative Replay Mechanisms (GRMs) [10], [11], [12], [13], [14], [15]. Episodic memory systems usually build a small buffer which stores a subset of data samples from past tasks while the model is regularized by a penalty term on the change of network' weights important to the past learnt tasks [5]. However, episodic memory-based approaches are not scalable to an infinite number of tasks. The GRM-based approaches developed in the last few years exhibit several advantages over episodic memory systems, such as not having to store the past data while being able to provide very large amounts of generated samples. In this paper, we mainly focus on the GRM based methods since GRMs do not rely on real data from prior tasks which actually may no longer be available. The first GRM-based work, proposed in [13], employs the Generative Adversarial Network (GAN) [16] to preserve and replay past data while a classifier is used to make the prediction. The subsequent GRM-based studies [13], [15] are followed by similar learning processes [13] and would focus on the classification task. Moreover, GRM performs well on a sequence of tasks within a single domain but would lead to degenerated performance when each task is characterized by a different data domain due to the mode collapse [17] (See empirical results in Fig. 12a).

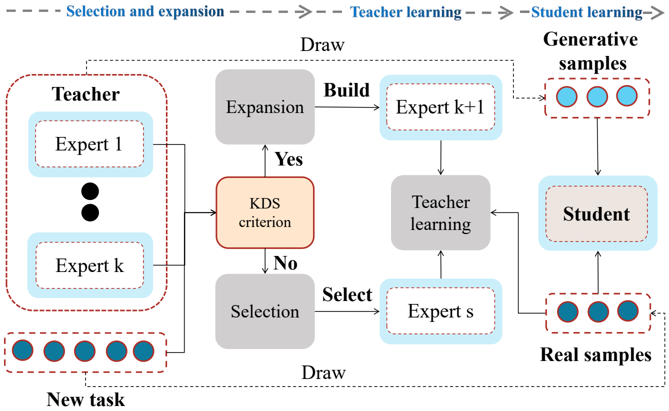


Fig. 1. The overview of the learning procedure for the proposed lifelong framework, which consists of three steps. First, when seeing a new task, we perform the KDS evaluation, by employing either NLL or KFD criterion, which guides us to perform either selection or expansion process (Eq. (6)). Second, we train the Teacher module by using Eq. (7), where we omit the GRM process when a selected expert is reused for learning a new task. Third, we train the student module on real samples from the current task combined with generative replay samples drawn from the teacher module. The detailed pseudocode is provided in Section 3.7.

A natural approach to LLL is to use a mixture model, where different mixture components would be specialized in learning different data domains. However, existing architecture expansible approaches [12], [18], besides focusing only on classification tasks on a single domain, they also have two major drawbacks : 1) The expansion and selection processes in [12], [18], [19], using Variational Autoencoders (VAEs), relies on the sample log-likelihood of each component, which is limited when considering a more powerful implicit generative model [20]; 2) The approaches from [12], [18] learn only a single latent representation space and therefore can not capture the correlations of underlying latent structures between different domains/tasks.

In this paper, we study a more challenging LLL setting in which each task is defined on a different data domain. Our learning goal is not only to capture the domain-specific underlying factors but also to model the correlations on the factors between multiple domains into a single embedded space. To implement this goal, we propose a novel lifelong learning framework, called the Dynamic Self-Supervised Teacher-Student (D-TS), where the Teacher module is implemented by a dynamically expandable Generating Adversarial Network (GAN) mixture model which expands its network architecture according to the given tasks complexity. In order to control the expansion of the Teacher and to use the previously learnt knowledge for accelerating the learning of new tasks, we introduce a new criterion, called the Knowledge Discrepancy Score (KDS) that evaluates the relevance between each learnt Teacher expert and the incoming task. Specifically, we employ KDS to determine the novelty of an incoming task after each task switch, guiding us to either reuse an existing expert for learning a related task or build a new expert for learning an entirely different task. To model the correlations on the underlying factors between multiple domains, we develop a lightweight latent variable generative model as the Student module and propose a self-supervised learning approach that trains the Student module on joint training set made up by mixing real training samples with generative replay samples drawn from the Teacher module, as shown in Fig. 1. In addition, we introduce a new regularized term in the Student's objective function, which minimizes the distance between the posterior distribution learnt by the Student and the conditional distribution parameterized by the identity information of each expert. This regularized term encourages embedding multiple knowledge sources from the Teacher into several clusters in the latent space of the Student, which further improves the cross-domain reconstruction and interpolation performance.

The main contributions of this paper are :

- We study a more challenging LLL setting in which we desire to learn domain-specific representations while also inferring the characteristics of these representations into a single latent space.
- We propose a new LLL framework, namely the Dynamic Self-Supervised Teacher-Student Network (D-TS), which enables the Teacher to expand its network architecture in order to learn an infinite number of tasks. Meanwhile, the Student module in D-TS is self-supervise trained to learn both predictive as well as generative representations across domains.
- We propose a new criterion, called the Knowledge Discrepancy Score (KDS) which controls the selection and expansion of the Teacher module without the need to infer from each component.
- We introduce a new conditional prior that computes correlations between the latent representations of the previously learnt databases and the domain of a newly given task, encouraging to embed different knowledge sources (modelled by different Teacher experts) into different clusters from the latent space of the Student during the knowledge discovery process.
- We analyze the forgetting behaviour of the Student when the Teacher dynamically changes its architecture by developing a new theoretical framework.

The rest of the paper is organized as in the following. Section 2 provides an overview of the lifelong learning area. The Dynamic Self-Supervised Teacher-Student (D-TS) framework is described in Section 3 and some of its applications in Section 4. The theoretical knowledge bounds are provided in Section 5. The experimental results are provided in Section 6 and the conclusions of this study are drawn in Section 7.

2 RELATED RESEARCH

In this section, we provide a brief background of related works.

2.1 Knowledge distillation

Knowledge distillation (KD) aims to transfer information from a large network, called the Teacher, to a smaller network, called the Student. KD has been widely used for classification tasks [21], [22], [23], [24], [25]. Some of the works also explore training a single Student model from an ensemble of networks to achieve higher performance with fewer computations [26], [27], [28], [29]. KD has also been used in lifelong learning (LLL), in the Teacher-Student (LTS) framework [30], where the Teacher module is trained across multiple tasks while the Student module learns both the information from a new task and that generated by the Teacher, representing the accumulated knowledge of previous tasks. However, LTS lacks scalability to learning an infinite sequence of tasks given that the Teacher module does not expand its network architecture when learning new tasks while its information capacity is limited. The quality of the Student's representations is also affected, especially when the Teacher learns a sequence of entirely different tasks. Additionally, most KD approaches focus on predictive tasks while KD remains unexplored for generative modelling under LLL.

2.2 Dynamic architectures

Dynamic Architecture Methods (DAMs) represent a popular approach for Continual Learning (CL) consisting of dynamically adding new network layers and hidden nodes in order to learn novel tasks [31], [32], [33], [34], [35], [36], [37]. These approaches would usually divide the network layers into shared and task-specific components, where the number of the latter can be expanded for new tasks [33]. The primary drawback of these models is that most existing DAMs would only focus on classification tasks and can not learn meaningful representations across domains under the unsupervised learning setting. Furthermore, there were some attempts for using mixture models to learn complex datasets [38], [39], [40] or learn an infinite number of tasks [12], [18], [41], [42]. However, the expansion of these models relies on the estimation of the sample log-likelihood, which requires each expert to have an explicit probabilistic function form. This is not the case in the proposed D-TS model since the expansion and selection of components in D-TS is based on the proposed KDS criterion requiring each expert to have a sampling procedure.

2.3 Memory buffer based methods

Storing few past data samples in a small memory buffer, used to alleviate catastrophic forgetting, was shown to achieve promising results in CL [5], [6], [7], [8], [43], [44], [45]. However, when increasing the number of tasks, the memory size and computational complexity burden of memory based methods would increase indefinitely [8]. Therefore, some works propose to employ a Generative Replay Network (GRM) [10], [11], [12], [13], [14], [15], [40], [46]. GRMs are able to generate data which are probabilistic consistent with the training data without employing any memory buffers. Thus the aim of GRMs is to preserve the knowledge of the previously learnt tasks. For instance, a typical GRM based approach usually trains a generator (VAE or GAN) for producing data samples after each task learning switch. These generative replay samples are incorporated together with samples drawn from a new task, to form a joint dataset. The generator and a classifier are trained on this joint dataset during the current task learning in order to assimilate knowledge from both the past and novel data sets. However, most existing GRM based methods are focused on the supervised learning setting [13], [15] while learning representations under CL is addressed by very few works [10], [11], [12]. The method from [10] is the first work to explore learning disentangled representations across domains under CL by using the VAE framework, where the environment-specific latent variables are used to model generative factors from a specific domain/task. Then a dynamic expansion mechanism is introduced in [12] to increase the inference capacity for novel data samples. Recently, the Teacher-Student structure is also used in the VAE framework [11] in order to maintain the performance on previous tasks by transferring the knowledge between its two modules. However, these GRM based models are not suitable for learning a long sequence of tasks due to their limited capacity and for frequently employing generative replay processes [42], [45].

In this paper, we introduce a new theoretical framework that not only provides insights into the forgetting behaviour of GRM based methods but also assesses the performance change in the Student module when modifying the Teacher architecture.

3 Dynamic Self-Supervised Teacher-Student Network (D-TS) framework

3.1 Problem definition

Let $\mathcal{P}_i$ and P_i be the distributions for the training set and testing set of the i-th task, respectively. For a sequence of domains (tasks) $\{\mathcal{T}_1, \mathcal{T}_2, \ldots, \mathcal{T}_K\}$, we assume that each distribution $\mathcal{P}_i$, defined on the data space $\mathcal{X} \in \mathbb{R}^d$, is drawn from a domain $\mathcal{T}_i$. In the context of lifelong learning, a model $\mathcal{M}$ only accesses samples drawn from $\mathcal{P}_i$ at the i-th task learning. Our learning goal is to train $\mathcal{M}$ to capture the generative factors from a sequence of tasks $\{\mathcal{T}_i \,|\, i = 1, \ldots, K\}$ without forgetting previously learnt latent representations. Our model consists of a Teacher module made up of a mixture of Generative Adversarial Networks (GAN) and a Student module, implemented by a generative latent variable model.

3.2 Preliminaries

In the following we describe the Generative Adversarial Network (GAN), which is used as an expert in the proposed mixture for the Teacher module [30]. A GAN [16] consists of two components : a generator network $G_\varepsilon \colon \mathcal{Z} \to \mathcal{X}$ and a discriminator network $D_\beta \colon \mathcal{X} \to \mathbb{R}$. The generation process is started by drawing a random noise vector $\mathbf{z} \in \mathcal{Z}$ from a fixed multivariate Gaussian distribution as the input of the generator $G_\varepsilon(\mathbf{z})$ which outputs a fake image $\mathbf{x}'$. The discriminator network $D_\rho(\mathbf{x})$, of parameters ρ, is trained to distinguish $\mathbf{x}'$ from a real image $\mathbf{x}$, while the generator is trained to generate fake images $\mathbf{x}'$ that can fool the discriminator. In this paper, we employ the Wasserstein GAN (WGAN) [47], [48] in the Teacher module whose training is defined by the loss function :

$$\begin{aligned}\min_{\mathbb{P}_\varepsilon} \max_{D_\rho \in \Theta} & \mathbb{E}_{\mathbf{x}_i \sim \mathcal{P}_i}\left[D_\rho\left(\mathbf{x}_i\right)\right] - \mathbb{E}_{\mathbf{x}' \sim \mathbb{P}_\varepsilon}\left[D_\rho\left(\mathbf{x}'\right)\right] \\ & + \gamma \mathbb{E}_{\widehat{\mathbf{x}} \sim \mathbb{P}_{\widehat{\mathbf{x}}}}\left[\left(\left\|\nabla_{\widehat{\mathbf{x}}} D_\rho\left(\widehat{\mathbf{x}}\right)\right\|_2 - 1\right)^2\right],\end{aligned} \tag{1}$$

where $\mathbb{P}_\varepsilon$ is the distribution approximated by the generator $G_\varepsilon(\mathbf{z})$. $\widehat{\mathbf{x}}$ is the interpolated image produced by $\widehat{\mathbf{x}} = a\mathbf{x}_i + (1-a)\mathbf{x}'$ where a is drawn from a uniform distribution $U(0,1)$ and $\mathbb{P}_{\widehat{\mathbf{x}}}$ is the distribution of the interpolated images. The last term is used to ensure the discriminator's Lipschitz constraint [48], and its weighting hyperparameter is considered as $\gamma = 10$ in the experiments.

3.3 The knowledge discrepancy score

Before we introduce the D-TS framework, we firstly describe the Knowledge Discrepancy Score (KDS) which is used to control the expansion of the Teacher module. Existing mixture models [12], [18] use the log-likelihood, evaluated by each component with a new training set, for the expansion or selection process. However, these mixture models require each component to have an inference mechanism (sample log-likelihood estimation), which does not allow for the use of explicit generative models as experts. Additionally, the selection of experts in existing mixture models do not consider a measure of comparison between the characteristic distribution of each expert and the probabilistic representation of the novel task. In the following, we introduce a novel approach to evaluate the expansion and selection of components for the mixture model using KDS, which addresses these two drawbacks.

Definition 1. The Knowledge Discrepancy Score (KDS). Given two distributions C and Q sampled from $\mathcal{X}$, let us define a distance measure function $\eta(\cdot)$. The KDS between two distributions is defined as :

$$\mathrm{KDS}_\eta\left(C, Q\right) = \eta\left(\Phi\left(\mathbf{X}_C\right), \Phi\left(\mathbf{X}_Q\right)\right), \tag{2}$$

where $\mathbf{X}_C \in \mathbb{R}^{n \times d}$ and $\mathbf{X}_Q \in \mathbb{R}^{n \times d}$ are two data matrices formed by n samples drawn from C and Q, respectively, where d is the dimension of each sample vector. $\Phi(\cdot)$ is a mapping which can be of arbitrary complexity. In the following, we introduce two measures for implementing KDS.

Knowledge Fréchet Distance (KFD). A direct approach is to evaluate the distance between two distributions in the high-dimensional data space by using a probabilistic measure, requiring additional computations [49], or auxiliary training [50], [51]. Recently, perceptual features extracted from deep Convolutional Neural Networks (CNN), pre-trained on ImageNet [3], have shown benefits in style matching [52] and transfer learning [53]. This motivates us to measure the KDS in the feature space to reduce the required computation complexity. In the following we propose the Knowledge Fréchet Distance (KFD), which uses the Fréchet distance [54] to implement $\eta(\cdot)$, evaluated on the low-dimensional feature space, as :

$$\begin{aligned}\mathrm{KDS}_\eta^{\mathrm{KFD}}(C, Q) = & \left\|e\left(\Phi\left(\mathbf{X}_C\right)\right) - e\left(\Phi\left(\mathbf{X}_Q\right)\right)\right\| \\ & + \operatorname{Tr}\left[\kappa(\Phi\left(\mathbf{X}_C\right)) + \kappa(\Phi\left(\mathbf{X}_Q\right)) - \right. \\ & \left. 2\left(\kappa\left(\Phi\left(\mathbf{X}_C\right)\right)\kappa\left(\Phi\left(\mathbf{X}_Q\right)\right)\right)^{1/2}\right],\end{aligned} \tag{3}$$

where $\Phi(\cdot)$ is a mapping function transforming $\mathbf{X}_C$ into the feature matrix $\mathbf{X}'_C \in \mathbb{R}^{n \times d'}, d' < d$ by using the feature extractor, which in our experiments is implemented by the last layer of an Inception network, trained on ImageNet, while $\operatorname{Tr}(\cdot)$ is the trace. $e(\cdot)$ and $\kappa(\cdot)$ are used to calculate the mean vector and covariance matrix for $\mathbf{X}'_S$. The Knowledge Fréchet Distance (KFD) represents the generalization of the Fréchet Distance Score [55].

Negative log-likelihood (NLL). For evaluating the novelty and similarity between the knowledge associated with each expert and that of the incoming task we can use a measure of the knowledge learned by the Student, which is implemented as a VAE, and represents the depository of the knowledge from all previous tasks. The Student is able to estimate the sample log-likelihood across domains. A similar log-likelihood indicates that the given task is known to the Student module. So the similarity measure function $\eta(\cdot)$ in this case for KDS is given by :

$$\operatorname{KDS}^{NLL}_{\eta}(C, Q) = \left|\mathcal{L}'\left(\Phi\left(\mathbf{X}_C\right)\right) - \mathcal{L}'\left(\Phi\left(\mathbf{X}_Q\right)\right)\right|, \quad (4)$$

where $\Phi(\cdot)$ is implemented as the identity function and $\mathcal{L}'(\mathbf{X}_C) = (1/n)\sum_{i=1}^{n} NLOG(\mathbf{X}_C[i])$ where $\mathbf{X}_C[i]$ is the i-th row of $\mathbf{X}_C$ and $NLOG(\cdot)$ is the estimator for the negative sample log-likelihood, implemented by the Student module (See Eq. (8)). NLL can be computed more efficiently than KFD because it is directly estimated by the Student module and does not require an externally pre-trained network.

3.4 The Teacher module

Existing Teacher-Student frameworks, such as the Lifelong Teacher-Student (LTS) [30], use a single GAN as the Teacher, but such approaches have limitations when learning several different datasets due to the mode collapse problem [17] (see also the empirical results in Section 6.6). In this paper, we develop a novel infinite mixture of GANs as a dynamically expandable experts-based memory system for the Teacher module in order to learn a growing number of different tasks. To ensure a compact network architecture, we require that a certain expert learns several data domains from tasks that have similarities with each other. We assume that after the t-th task learning is finished, we have trained K experts $\{G_{\varepsilon_1}, \ldots, G_{\varepsilon_K}\}$. Let $\mathbb{P}_i$ represent the probabilistic representation of G_{ε_i}. The dynamic expansion and selection mechanism of the Teacher module is shown in Fig. 2, where $\operatorname{KDS}_{\eta}(\cdot, \cdot)$ is evaluated between the knowledge accumulated by each expert and that corresponding to the incoming task. .

Selection and expansion using KDS. From Fig. 2, after learning the t-th task, the component selection and network expansion procedure is performed by a non-parametric inference process in which we firstly evaluate the Knowledge Discrepancy Score (KDS) between the probabilistic representations of the new dataset $\mathcal{P}_{t+1}$, and those corresponding to the Teacher's experts $\mathbb{P}_i$, $i = 1, \ldots, K$:

$$q = \begin{cases} 0, & \min\left\{\operatorname{KDS}_{\eta}\left(\mathbb{P}_i, \mathcal{P}_{t+1}\right)\right\}_{i=1,\ldots,K} > \mathit{hold}\,; \\ 1, & \min\left\{\operatorname{KDS}_{\eta}\left(\mathbb{P}_i, \mathcal{P}_{t+1}\right)\right\}_{i=1,\ldots,K} \leq \mathit{hold}\,, \end{cases} \quad (5)$$

where q is the probability of performing either network expansion or component selection for updating, and *hold* is a threshold. Then the selection probability p_i of each expert is updated as :

$$p_i = \begin{cases} \dfrac{q \times \left(1/\operatorname{KDS}_{\eta}\left(\mathbb{P}_i, \mathcal{P}_{t+1}\right)\right)}{\sum_{j=1}^{K}\left(1/\operatorname{KDS}_{\eta}\left(\mathbb{P}_j, \mathcal{P}_{t+1}\right)\right)}, & i < K+1\,; \\ 1-q, & i = K+1\,. \end{cases} \quad (6)$$

If $q = 0$, then the Teacher module expands its capacity, $p_{(K+1)} = 1$, otherwise the Teacher module selects the most appropriate expert for learning the new task, according to $\{p_1, p_2, \ldots, p_K\}$. Fewer epochs would be used for updating an existing expert when compared to training a new component added to the Teacher.

Training the infinite mixture model. After determining the selection probability, we define the Teacher's loss function for the following $(t+1)$-th task, as :

$$\begin{aligned} \min_{\varepsilon_1,\ldots,\varepsilon_{S^\star}} \max_{D_\rho \in \Theta} \sum_{i=1}^{S^*} \Big\{ & w_i \left(\mathbb{E}_{\mathbf{x}_{t+1} \sim \mathcal{P}_{t+1}}\left[D_\rho\left(\mathbf{x}_{t+1}\right)\right]\right. \\ & -\mathbb{E}_{\mathbf{x}'_i \sim \mathbb{P}_i}\left[D_\rho\left(\mathbf{x}'_i\right)\right] \\ & \left.+\lambda \mathbb{E}_{\widehat{\mathbf{x}} \sim \mathbb{P}_{\widehat{\mathbf{x}}}}\left[\left(\left\|\nabla_{\widehat{\mathbf{x}}} D_\rho\left(\widehat{\mathbf{x}}\right)\right\|_2 - 1\right)^2\right]\right) \Big\}, \end{aligned} \quad (7)$$

where $S^\star$ represents the number of experts, determined by $S^* = K$ if the Teacher does not expand ($q = 1$ in (5)) at the $(t+1)$-th task learning, otherwise $S^\star = K + 1$ when a new expert is added. The regularization weight is considered in the experiments as $\lambda = 10$. The Teacher is then trained using Eq. (7) with the expert's weights $\mathbf{w}$ sampled from a Categorical distribution $Cat(p_1, \ldots, p_{S^\star})$. We name D-TS-KFD when considering the Knowledge Fréchet Distance (KFD), and D-TS-NLL for using the Negative log-likelihood, for KDS in order to decide whether to select a new component for the Teacher module.

3.5 The Student module

For the design of the Student we consider two crucial requirements : 1) A light architecture with fewer parameters than the Teacher module; 2) A powerful inference mechanism for representation learning. Let us consider the latent variable generative model $p_\theta(\mathbf{x}, \mathbf{z}, \mathbf{u}) = p_\theta(\mathbf{x} \mid \mathbf{z}, \mathbf{u}) p(\mathbf{z}, \mathbf{u})$, where the categorical variable $\mathbf{u}$, also called expert-variable, represents the identity information while the continuous variable $\mathbf{z}$ represents the fundamental generative factors. The learning goal of the Student module for a single task is that of maximizing the intractable marginal log-likelihood $\log p(\mathbf{x}) = \iint \log p_\theta(\mathbf{x} \mid \mathbf{z}, \mathbf{u}) p(\mathbf{z}, \mathbf{u})\, \mathrm{d}\mathbf{z}\, \mathrm{d}\mathbf{u}$ by optimizing a lower bound (derived in Appendix A from the Supplementary Material) :

$$\begin{aligned} \log p(\mathbf{x}) \geq\ & \mathbb{E}_{q(\mathbf{z}, \mathbf{u} \mid \mathbf{x})}\left[\log p(\mathbf{x} \mid \mathbf{z}, \mathbf{u})\right] \\ & - D_{KL}(q(\mathbf{z} \mid \mathbf{x}) \,||\, p(\mathbf{z})) \\ & - D_{KL}(q(\mathbf{u} \mid \mathbf{x}) \,||\, p(\mathbf{u}))\,, \end{aligned} \quad (8)$$

where the posterior $p(\mathbf{z}, \mathbf{u} \mid \mathbf{x})$ is intractable and is approximated by the variational distribution $q(\mathbf{z}, \mathbf{u} \mid \mathbf{x})$. We introduce two variational distributions, $q_\omega(\mathbf{z} \mid \mathbf{x})$ and $q_\psi(\mathbf{u} \mid \mathbf{z})$ to model $q(\mathbf{z}, \mathbf{u} \mid \mathbf{x})$, where the former is implemented by a neural network of input $\mathbf{x}$ and yielding the latent representation $\mathbf{z}$. $q_\psi(\mathbf{u} \mid \mathbf{z})$ is implemented by an expert-inference network whose last layer outputs the softmax probabilities $\{\mathbf{u}'_1, \mathbf{u}'_2, \ldots, \mathbf{u}'_k\}$, indicating the likelihood that $\mathbf{z}$ is associated with one of the experts. In order to reduce the variation of gradients [56], we adopt the Gumbel-Max trick [57], [58], which was also used in [56], [59], [60], [61], to calculate a differentiable relaxation for the discrete variables $\mathbf{u}$:

$$\mathbf{b}_k = \frac{\exp\left((\log \mathbf{u}'_k + \mathbf{g}_k)/T\right)}{\sum_{i=1}^{k} \exp\left((\log \mathbf{u}'_i + \mathbf{g}_i)/T\right)}\,, \quad (9)$$

where $\mathbf{u}'_k$ is the probability defined by the softmax function $q_\psi(\mathbf{u} \mid \mathbf{z})$ and $\mathbf{b} = \{\mathbf{b}_1, \cdots, \mathbf{b}_k\}$ is the continuous relaxation of $\mathbf{u}$. $\mathbf{g}_k$ is sampled from $\operatorname{Gumbel}(0, 1)$ distribution, while T is the

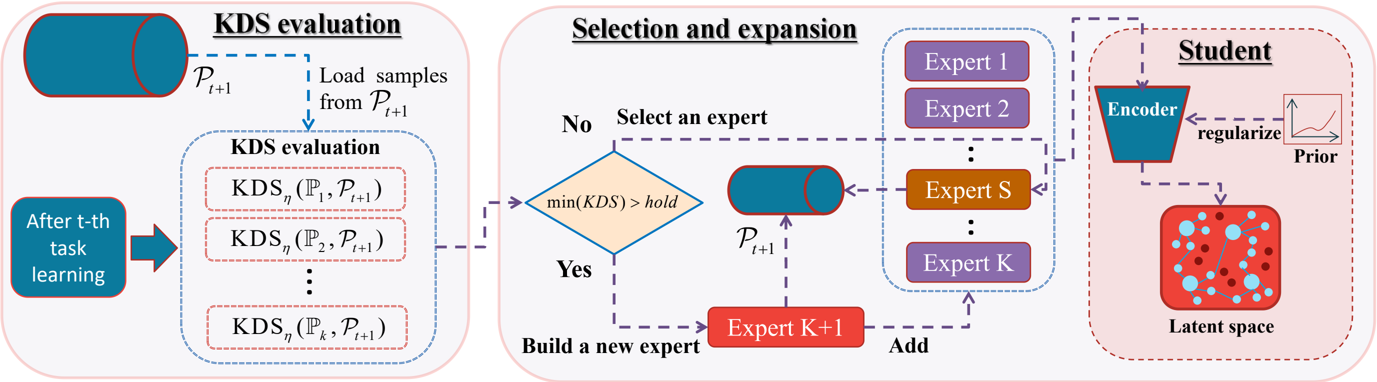


Fig. 2. The learning procedure for D-TS. When learning the $(t+1)$-th task learning, we perform the KDS evaluation between its probabilistic representation $\mathcal{P}_{t+1}$, and those corresponding to the Teacher's experts, $\mathbb{P}_i$, $i = 1, \ldots, K$. If the minimum KDS is larger than a threshold *hold*, then we add a new expert to the mixture system, otherwise, we select the expert with the minimum KDS for learning the $(t+1)$-th task. The activated experts are shown in red. The Student is trained along with the Teacher aiming to compress the knowledge from different sources (experts) into a compact latent space.

temperature parameter controlling the smoothness. This sampling process is implemented during both inference and generation.

Given that we use a simple fixed prior $p(\mathbf{z})$ in Eq. (8), this can not be helpful for embedding multiple knowledge sources from the Teacher into several clusters in the Student's latent space. In addition, the simple fixed prior would lead to posterior collapse, [62]. Furthermore, Eq. (8) still requires an extra network to approximate $q(\mathbf{u}\,|\,\mathbf{x})$. To address these issues, we propose to modify the two Kullback-Leibler (KL) terms in Eq. (8). Firstly, we approximate $q(\mathbf{u}\,|\,\mathbf{x})$ by using the expert inference network in which $\mathbf{z}$ is firstly sampled from $q_\omega(\mathbf{z}\,|\,\mathbf{x})$ and then is used as input to $q_\psi(\mathbf{u}\,|\,\mathbf{z})$. Therefore, the second KL term in Eq. (8) is replaced by $\mathbb{E}_{q_\omega(\mathbf{z}\,|\,\mathbf{x})} D_{KL}(q_\psi(\mathbf{u}\,|\,\mathbf{z})\,||\,p(\mathbf{u}))$, where $q_\psi(\mathbf{u}\,|\,\mathbf{z})$ is optimized to match the expert-variable. Secondly, we replace the prior $p(\mathbf{z})$ with a new conditional distribution $p(\mathbf{z}\,|\,\mathbf{u})$ that dependents on $\mathbf{u}$. Then the first KL term in the right-hand side of Eq. (8) is expressed by $D_{KL}(q_\omega(\mathbf{z}\,|\,\mathbf{x})\,||\,p(\mathbf{z}\,|\,\mathbf{u}))$. We implement $p(\mathbf{z}\,|\,\mathbf{u})$ considering the conditional Gaussian distribution $\mathcal{N}(B(\mathbf{u}_j^*), \mathbf{I})$, where $\mathbf{u}_j^* \in \mathbb{R}$ represents the index of the j-th expert G_{ε_j} and $B(\cdot)$ transforms the ground truth expert-variable $\mathbf{u}_j^*$ into a vector where each item is $\mathbf{u}_j^*$, and $\mathbf{I}$ is the identity matrix. When a single expert G_{ε_j} is used for the Student training, the objective function for the student is defined as :

$$\begin{aligned} &\mathbb{E}_{q(\mathbf{z},\mathbf{u}\,|\,\mathbf{x})}\left[\log p_\theta(\mathbf{x}\,|\,\mathbf{z},\mathbf{u})\right] - D_{KL}(q_\omega(\mathbf{z}|\mathbf{x})\,||\,p(\mathbf{z}\,|\,\mathbf{u})) \\ &\quad - \mathbb{E}_{q_\omega(\mathbf{z}\,|\,\mathbf{x})} D_{KL}(q_\psi(\mathbf{u}\,|\,\mathbf{z})\,||\,p(\mathbf{u}))\,. \end{aligned} \tag{10}$$

We desire to minimize the KL divergence between $q_\omega(\mathbf{z}\,|\,\mathbf{x})$ and the prior $p(\mathbf{z}\,|\,\mathbf{u})$, assumed to be Gaussian functions, in order to allow the Student to embed knowledge inferred from different sources (experts that are assigned by the unique expert-variable from the Teacher module) into different regions of its latent space.

-

3.6 Self-supervised learning for the Student

Existing KD approaches assume that data samples are provided by the user during the training. However, in the context of the lifelong learning setting, we do not have access to past samples and these KD approaches can not be applied in our framework. In this paper, we introduce a novel Self-Supervised Learning (SSL) approach in which past data samples are generated by the Teacher. Then, these pseudo samples can be used for training the Student. Additionally, unlike existing KD approaches that transfer knowledge only at the logit-level [63], [64], [65], the proposed SSL can transfer statistic data representations through sampling without accessing any real samples and labels. Moreover, the proposed SSL transfers the knowledge from the Teacher represented by multiple source distributions, implemented by mixtures of expert GAN models as described in Section 3.4, to a compact Student latent space. An ideal solution for the knowledge transfer is to minimize the distance between the Teacher's and Student's probabilistic representations. While KL divergence was used for KD [24] before, this paper is the first work to explore KL for generative modelling under the LLL. KL can have a tractable optimization form, as shown in the following.

Proposition 1. Let $\mathbb{S}$ be the Teacher's distribution and $S(\mathbf{x})$ be the density function of $\mathbb{S}$. Let $\mathbb{P}_\theta$ represent the Student's distribution. Minimizing $D_{KL}(\mathbb{S}\,||\,\mathbb{P}_\theta)$ can be formulated as maximizing the log-likelihood of $p_\theta(\mathbf{x})$ with the expectation of $\mathbb{S}$.

$$D_{KL}\left(\mathbb{S}\,||\,\mathbb{P}_\theta\right) = \mathbb{E}_{\mathbf{x}\sim\mathbb{S}}\left[\log S\left(\mathbf{x}\right) - \log p_\theta\left(\mathbf{x}\right)\right]\,. \tag{11}$$

The detailed proof is provided in Appendix B from the Supplementary Material. We omit the first term from the right hand side of Eq. (11) because we only update the Student's parameters during the optimization of Eq. (11). Therefore, the minimization of $D_{KL}\left(\mathbb{S}\,||\,\mathbb{P}_\theta\right)$ becomes the maximization of $\mathbb{E}_{\mathbf{x}\sim\mathbb{S}}\left[\log p_\theta(\mathbf{x})\right]$. Since our Teacher has several experts, we implement $S(\mathbf{x})$ as the mixture density function $S(\mathbf{x}) = (1/Z)\sum_{i=1}^{K} \pi_i m_{\varepsilon_i}(\mathbf{x})$, where K is the number of experts and Z is the normalizing term. $m_{\varepsilon_i}(\mathbf{x})$ is the density function for the i-th expert, parameterized by ε_i. By considering $\pi_i = 1/K$ we ensure that each expert has equal contribution. The mixture density function is optimized by maximizing :

$$\begin{aligned} \mathbb{E}_{\mathbf{x}\sim\mathbb{S}}[\log p_\theta(\mathbf{x})] &= \int \frac{1}{Z}\sum\nolimits_{i=1}^{K}\left\{\pi_i m_{\varepsilon_i}\left(\mathbf{x}\right)\log p_\theta\left(\mathbf{x}\right)\,\mathrm{d}\mathbf{x}\right\} \\ &= \frac{1}{Z}\sum\nolimits_{i=1}^{K}\left\{\pi_i \mathbb{E}_{\mathbf{x}\sim\mathbb{P}_i}\left[\log p_\theta\left(\mathbf{x}\right)\right]\right\}\,. \end{aligned} \tag{12}$$

The normalization term $(1/Z)$ is omitted during the optimization for the sake of simplification. The final loss function for the Student when learning a certain task, including the loss function from Eq. (12), and the Evidence Lower Bound (ELBO) on the current task, Eq. (10) (details in Appendix C from the Supplementary Material), is given by :

$$\mathcal{L}_{Stu} = \underbrace{\mathbb{E}_{q_{\omega,\psi}(\mathbf{z},\mathbf{u}\,|\,\mathbf{x}_t)}\left[\log \frac{p_\theta\left(\mathbf{x}_t,\mathbf{z},\mathbf{u}\right)}{q_{\omega,\psi}\left(\mathbf{z},\mathbf{u}\,|\,\mathbf{x}_t\right)}\right]}_{\text{ELBO on the t-th task}} + \underbrace{\sum\nolimits_{i=1}^{K} \pi_i \mathbb{E}_{\widehat{\mathbf{x}}_i\sim\mathbb{P}_i}\mathbb{E}_{q_{\omega,\psi}(\mathbf{z},\mathbf{u}\,|\,\widehat{\mathbf{x}}_i)}\left[\log \frac{p_\theta\left(\widehat{\mathbf{x}}_i,\mathbf{z},\mathbf{u}\right)}{q_{\omega,\psi}\left(\mathbf{z},\mathbf{u}\,|\,\widehat{\mathbf{x}}_i\right)}\right]}_{\text{ELBO on the generative replay samples}}, \tag{13}$$

where the random variables $\{\widehat{\mathbf{x}}_1,\dots,\widehat{\mathbf{x}}_K,\mathbf{x}_t\}$ are mutually independent and are sampled from their corresponding distributions $\{\mathbb{P}_1,\dots,\mathbb{P}_K,\mathcal{P}_t\}$ characterizing the generated data by K experts from the Teacher module and the given new t-th task, respectively.

3.7 The training algorithm

In the following, we provide **Algorithm 1**, describing the processing pipeline of the proposed lifelong learning strategy, summarized in three steps :

- **Step 1. Selection and expansion mechanism :** When starting learning the first task $\mathcal{T}_1$, the Teacher module has no experts. In this case, we build a new expert according to the training procedure from **Step 2**, otherwise we verify the Teacher's expansion and selection as follows : we evaluate the KDS between each Teacher's expert and the training set of the current task $\mathcal{T}_i$ by using the criterion KFD or NLL, as described in Section 3.4. Then we employ the threshold $hold$ from Eq. (5) and Eq. (6), to decide either the selection of an expert to be updated, or initiating the expansion process for the Teacher module.
- **Step 2. Training an expert for the Teacher :** If the Teacher module performs the expansion at the current task learning $\mathcal{T}_t$, then we directly train the newly added expert on samples from $\mathcal{T}_t$, otherwise, we form a joint dataset $\widehat{\mathbf{X}}^t = \mathbf{X}^t \bigcup \mathbf{X}'$, including the new training set $\mathbf{X}^t \sim \mathcal{T}_t$ and $\mathbf{X}' \sim G_{\varepsilon_s}$, where s is the index of the selected expert. We then update the selected expert on the dataset $\widehat{\mathbf{X}}^t$ at $\mathcal{T}_t$ by using Eq. (7).
- **Step 3. Student learning :** During each batch learning, we draw the same number of generative samples from each Teacher expert. These generative samples are incorporated with real training samples of the current task, considered in an equal probability with that generated by each of the experts. These will form a batch of samples for training the Student module using Eq. (13). The Student model is initialized each time with the parameters learnt previously, while its parameters are randomly generated only when trained for the first time.

4 Applications

In the following we outline some applications of the proposed lifelong learning framework.

4.1 Prediction tasks

In this section, we extend the D-TS framework for classification tasks. We implement each expert from the Teacher module by using a combination of a generator and a solver. The solver is a neural network, which outputs the class probability, after being trained by minimizing the cross-entropy loss :

$$\mathcal{L}^{c}_{Teach} = \mathbb{E}_{\mathbf{x}\sim\mathcal{P}_t,\mathbf{y}\sim\mathcal{P}_{\mathbf{Y},t}}\mathcal{L}(T_{\widehat{\delta}_s}(\mathbf{y}\mid\mathbf{x}),\mathbf{y})\,, \tag{14}$$

where $\{\mathcal{P}_t,\mathcal{P}_{\mathbf{Y},t}\}$ represent the distributions of data and their labels, from the t-th task, $\mathcal{L}(\cdot)$ is the cross-entropy loss and $T_{\widehat{\delta}_s}(\mathbf{y}\,|\,\mathbf{x})$ is the solver, defined by parameters $\widehat{\delta}_s$ in the selected expert. In order to allow the Student to perform data classification tasks, we introduce an inference model $S_\lambda(\mathbf{y}\,|\,\mathbf{x})$ for the Student module, trained on images and labels sampled from the data generated by the Teacher and by using samples from the current database :

$$\begin{aligned}\mathcal{L}^{c}_{Stu} &= \mathbb{E}_{\mathbf{x}\sim\mathcal{P}_t,\mathbf{y}\sim\mathcal{P}_{\mathbf{Y},t}}\mathcal{L}\left(S_\lambda\left(\mathbf{y}\,|\,\mathbf{x}\right),\mathbf{y}\right)\\ &+\sum\nolimits_{i=1}^{K}\sum\nolimits_{j=1}^{N_i}\left\{\mathcal{L}\left(S_\lambda\left(\mathbf{y}\,|\,\widehat{\mathbf{x}}_i^j\right),T_{\delta_i}\left(\mathbf{y}\,|\,\widehat{\mathbf{x}}_i^j\right)\right)\right\},\end{aligned} \tag{15}$$

where $\widehat{\mathbf{x}}_i^j$ is the j-th data sample, $j=1,\dots,N_i$ drawn by the i-th expert from the Teacher module, while K is the number of experts. The first term compares the outputs predicted by the Student against the ground-truth labels from the current task. The second term represents the distillation loss calculated by the cross-entropy between Softmax outputs provided by the Student module while the target labels are produced by the Teacher. Eq. (15) is only used to optimize $S_\lambda\left(\mathbf{y}\,|\,\mathbf{x}\right)$ and we also introduce a loss to optimize both $S_\lambda\left(\mathbf{y}\,|\,\mathbf{x}\right)$ and other components of the Student module by incorporating the variable $\mathbf{y}$ into Eq. (13), resulting in :

$$\mathcal{L}^{Dec}_{Stu} = \mathbb{E}_{q(\mathbf{z},\mathbf{u},\mathbf{y}|\mathbf{x_t})}\left[\log\frac{p\left(\mathbf{x_t}\mid\mathbf{z},\mathbf{u},\mathbf{y}\right)}{q\left(\mathbf{z},\mathbf{u},\mathbf{y}\mid\mathbf{x_t}\right)}\right]+ \sum\nolimits_{i=1}^{K}\left\{\pi_i\mathbb{E}_{\widehat{\mathbf{x}}_i\sim\mathbb{P}_i}\mathbb{E}_{q(\mathbf{z},\mathbf{u},\mathbf{y}_i|\widehat{\mathbf{x}}_i)}\left[\log\frac{p\left(\widehat{\mathbf{x}}_i\mid\mathbf{z},\mathbf{u},\mathbf{y}\right)}{q\left(\mathbf{z},\mathbf{u},\mathbf{y}\mid\widehat{\mathbf{x}}_i\right)}\right]\right\}, \tag{16}$$

where we no longer specify the network parameters. The inference model $q(\mathbf{y}\,|\,\mathbf{x})$ is modelled by the Gumble-max distribution [57] whose parameters are sampled from the probabilistic outputs of $S_\lambda(\mathbf{y}\,|\,\mathbf{x})$ in order to enable the end-to-end optimization.

4.2 Learning disentangled representations

Most artificial learning approaches aiming to describe meaningful feature variations through generative learning are based on the latent variable model and in VAEs would impose a penalty term in the loss function in order to induce disentangled representations [59], [66], [67], [68], [69], [70]. For enticing the learning of disentangled representations under the lifelong learning framework, we consider the following loss function for the Student module :

$$\begin{aligned}\mathcal{L}^{Dis}_{Stu} = \sum\nolimits_{i=1}^{K}&\left\{\pi_i\mathbb{E}_{\mathbf{x}'_i\sim\mathbb{P}_i}\mathbb{E}_{q_{\omega,\psi}\left(\mathbf{z},\mathbf{u}\,|\,\mathbf{x}'_i\right)}\left[\log p_\theta\left(\mathbf{x}'_i\,|\,\mathbf{z},\mathbf{u}\right)\right]\right.\\ &\left.-\tau\big\|A_i-C\big\|\right\}+\mathbb{E}_{q_{\omega,\psi}(\mathbf{z},\mathbf{u}\,|\,\mathbf{x}_t)}\left[\log p_\theta\left(\mathbf{x}_t\,|\,\mathbf{z},\mathbf{u}\right)\right]\\ &-\tau\left|A_t-C\right|,\end{aligned} \tag{17}$$

where $A_t = D_{KL}(q_\omega(\mathbf{z}\,|\,\mathbf{x}_t)\,||\,p(\mathbf{z}\,|\,\mathbf{u}))$ is the KL divergence between two distributions estimated on the current task t. $A_i = D_{KL}(q_\omega(\mathbf{z}\,|\,\mathbf{x}'_i)\,||\,p(\mathbf{z}\,|\,\mathbf{u}))$ is the KL divergence estimated by using past data. τ and C are parameters controlling the disentanglement. We set $\tau=4$ in our experiments to avoid sacrificing much reconstruction ability, while C is linearly increased from a very small value 0.5 to 25.0 during the training, [69]. We omit the KL term on $q\left(\mathbf{u}\,|\,\mathbf{z}\right)$ since this term would not benefit from disentangled representation learning.

4.3 Inter-domain interpolation

After the lifelong learning, the model $\mathcal{M}$ provides several latent representations $\{\mathbf{z}_1,\mathbf{z}_2\dots,\mathbf{z}_K\}$ where each $\mathbf{z}_i$ represents the generative factors for images sampled from $\mathcal{P}_i$. The LLL model

TABLE 1
The performance of various models under the MSFIR lifelong learning setting.

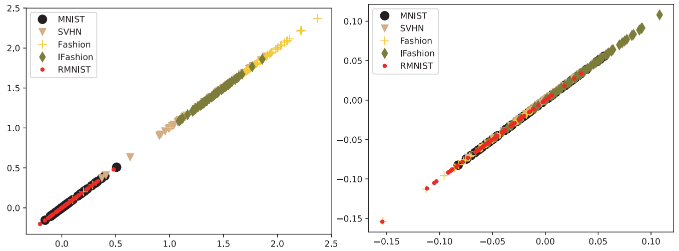
	MSE					SSMI					PSNR				
Datasets	LGM	D-TS-KFD	D-TS-NLL	BE-Stu	LTS	LGM	D-TS-KFD	D-TS-NLL	BE-Stu	LTS	LGM	D-TS-KFD	D-TS-NLL	BE-Stu	LTS
MNIST	19.60	26.84	28.61	33.66	73.97	0.90	0.88	0.87	0.86	0.73	22.51	21.14	20.64	20.13	17.10
SVHN	292.15	29.67	31.04	71.58	42.98	0.36	0.65	0.64	0.47	0.54	11.33	12.65	12.58	12.13	11.91
Fashion	80.95	39.35	48.96	149.26	45.64	0.17	0.72	0.66	0.47	0.72	12.65	18.38	17.01	13.57	17.77
IFashion	94.32	35.92	37.94	83.44	37.60	0.58	0.74	0.75	0.61	0.76	16.35	18.70	17.86	15.89	18.34
RMNIST	19.58	24.09	23.81	24.29	21.97	0.90	0.89	0.89	0.90	0.90	22.51	21.46	21.53	21.57	21.64
Average	101.32	**31.17**	34.07	72.45	44.83	0.58	**0.78**	0.76	0.66	0.73	17.07	**18.47**	17.92	16.66	17.35

Algorithm 1: D-TS-KFD training algorithm

```
Input: All training databases
Output: The model's parameters
The total number of tasks ($n$) ;
for $i < n$ do
    Step 1 : Selection and expansion mechanisms ;
    if $i == 1$ then
        Build a new expert $G_{\varepsilon_1}$ for the teacher module ;
    end
    else
        Calculate KDS between each teacher expert and $\mathcal{P}_i$ ;
        Check the selection and expansion using Eq. (5), (6) ;
        if expansion = True then
            Build a new expert for the teacher module ;
        end
        else
            Select an expert for the current task learning ;
        end
    end
    Step 2 : Teacher learning;
    $\mathbf{X}^t \sim \mathcal{P}_i$ from the training set ;
    The index of the selected expert ($s$) ;
    if expansion = False then
        $\widehat{\mathbf{X}}^t = \mathbf{X}^t \bigcup \mathbf{X}', \mathbf{X}' \sim G_{\varepsilon_s}$ Form a joint dataset ;
    end
    The total number of training steps ($B_i$) at $\mathcal{T}_i$ ;
    $\mathbf{w} \sim Cat(p_1, \ldots, p_{S^\star})$ Draw expert's weight ;
    for $j < B_i$ do
        Train the Teacher on $\mathbf{X}^t$ using Eq. (7) ;
        Step 3 : Student learning ;
        $\{\widehat{\mathbf{x}}_1, \ldots, \widehat{\mathbf{x}}_k, \mathbf{x}_t\} \sim \{\mathbb{P}_1, \ldots, \mathbb{P}_k, \mathcal{P}_t\}$ ;
        $\mathbf{x}_b = \{\widehat{\mathbf{x}}_1, \ldots, \widehat{\mathbf{x}}_k, \mathbf{x}_t\}$ Form a batch of samples ;
        Train the student on $\mathbf{x}_b$ using $\mathcal{L}_{Stu}$ ;
    end
end
```

besides representing the generative factors for each domain $\mathcal{T}_i$ it also captures the shared generative factors across domains. To evaluate this property, we are interested in manipulating the latent variable space and interpolating between images drawn from two different domains. Let $\mathbf{x}^i$ and $\mathbf{x}^j$ be two images drawn from the i-th task and j-th task, respectively. Let $\tilde{\mathbf{x}}_{i\to j}(c) = \mathrm{Dec}(\mathbf{z}^i c + \mathbf{z}^j(1-c))$ be the interpolated image for the interpolation parameter $c \in [0,1]$, where $\mathrm{Dec}(\cdot)$ is the decoder and $\{\mathbf{z}^i, \mathbf{z}^j\}$ are the latent representations of $\mathbf{x}^i$ and $\mathbf{x}^j$, respectively. We extend the image interpolation from [71] into exploring the joint latent space of $\{\mathcal{T}_1, \mathcal{T}_2, \ldots, \mathcal{T}_K\}$, under the LLL setting:

- **Boundary conditions.** $\tilde{\mathbf{x}}_{i\to j}(0) = \mathbf{x}^i$ and $\tilde{\mathbf{x}}_{i\to j}(1) = \mathbf{x}^j$ when $\mathrm{Dec}(\cdot)$ is the optimal decoder.



(a) With conditional prior.

(b) Without conditional prior.

Fig. 3. Latent space projections for D-TS model.

- **Monotonicity.** We assume that $\mathrm{Dec}(.)$ is the optimal decoder. For a given distance measure $\phi(\cdot,\cdot)$ evaluating the similarity between two images, we can define the distance between the interpolated image and the original input :

$$\phi(\tilde{\mathbf{x}}_{i\to j}(c), \mathbf{x}^i) \le \phi(\tilde{\mathbf{x}}_{i\to j}(c'), \mathbf{x}^i)\,, \tag{18}$$

where $c' > c$ and

$$\phi(\tilde{\mathbf{x}}_{i\to j}(c'), \mathbf{x}^j) \le \phi(\tilde{\mathbf{x}}_{i\to j}(c), \mathbf{x}^j)\,, \tag{19}$$

- **Smoothness.** The interpolation function $\tilde{\mathbf{x}}_{i\to j}(c)$ is Lipschitz continuous with a constant V.

$$\|\tilde{\mathbf{x}}_{i\to j}(c), \tilde{\mathbf{x}}_{i\to j}(c+a)\| \le V|a|\,. \tag{20}$$

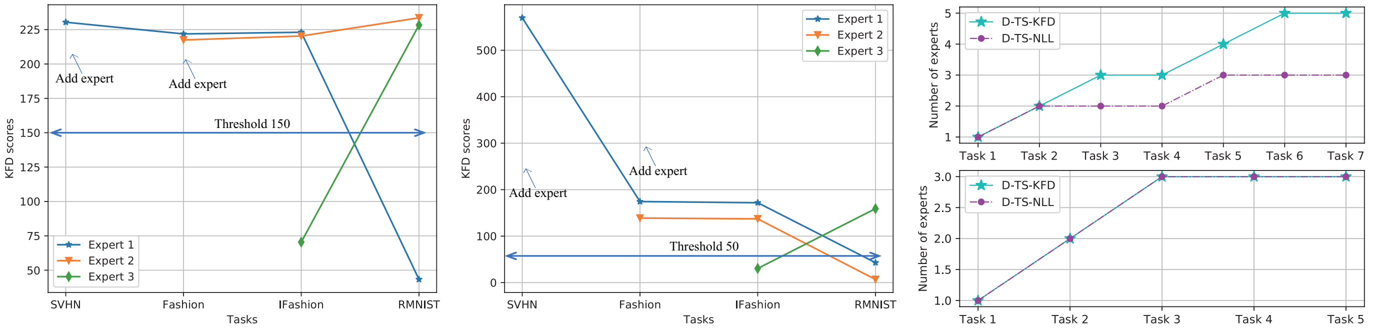
Different from [71], we aim to learn latent representations under the LLL setting, which is more challenging because neural network models would forget previously learnt latent representations when trained on a new task. Given the image interpolation properties from above, we define a new criterion evaluating the effectiveness of the model when performing image interpolation:

$$\Psi(\tilde{\mathbf{x}}_{i\to j}(c), \mathbf{x}^j), \quad c > 0.5\,, \tag{21}$$

where $\Psi(\cdot)$ is a pre-defined criterion which can be implemented as the image reconstruction error. If $\Psi(\tilde{\mathbf{x}}_{i\to j}(c), \mathbf{x}^j)$ is small, this means that the interpolated result $\tilde{\mathbf{x}}_{i\to j}(c)$ is very similar to $\mathbf{x}^j$, as c increases.

5 Theoretical analysis of the forgetting behaviour for the Student

In this section, we extend the theoretical analysis from [42] to the proposed Teacher-Student framework, with the emphasis on studying the forgetting behaviour of the Student.



(a) KFD when learning 5 tasks.

(b) NLL when learning 5 tasks.

(c) Network expansion in D-TS.

Fig. 4. Knowledge discrepancy evaluation and the expansion of the network during the training.

5.1 Preliminary

Definition 2. Let $h\colon \mathcal{X} \to \mathcal{Y}$ be a classifier defined as $h \in \mathcal{H}$, where $\mathcal{H}$ is the classifiers' domain and $\mathcal{Y}$ is an output space, represented as $\{-1, 1\}$ for binary classification, and $\{1, 2, \ldots, n'\}, n' > 2$ for multi-class classification.

Definition 3. Let $\mathbb{P}_i^{t-i}$ be the pseudo distribution of $\mathcal{P}_i$ formed by samples drawn from $\{\mathbf{x}, \mathbf{y}\} \sim \mathbb{P}_{\varepsilon_1}^t$ if $\mathrm{F}_{task}(\mathbf{x}) = i$, where $\mathrm{F}_{task}(\cdot)$ is the task labelling function that returns the true task label for $\mathbf{x}$. We assume that the Teacher has a single component. Therefore, $\mathbb{P}_{\varepsilon_1}^t$ is the distribution of the joint samples $\{\mathbf{x}, y\}$ drawn from the probabilistic representation of the Teacher trained at the t-th task, where the label y for each generated data $\mathbf{x}$ is given by the solver of the Teacher. The superscript of $\mathbb{P}_i^{t-i}$ represents that the pseudo distribution $\mathbb{P}_i^1$ is refined to $\mathbb{P}_i^{t-i}$, when the Teacher is trained for $t - i$ times from the $(i + 1)$-th task to the t-th task through GRM.

Definition 4. **(Discrepancy distance.)** Let P_i and $\mathcal{P}_i$ be two distributions over the space $\mathcal{X} \times \mathcal{Y}$. Let $\mathcal{L}$ be a symmetric and bounded loss function $\forall (y, y') \in \mathcal{Y}^2, \mathcal{L}(y, y') \leq A'$, and $\mathcal{L}$ obeys the triangle inequality, where A' is a positive number. We define the discrepancy distance $\mathrm{Disc}(\cdot,\cdot)$ between two marginals, P_i and $\mathcal{P}_i$ as :

$$\begin{aligned}\mathrm{Disc}\left(\mathcal{P}_i^{\mathcal{X}}, P_i^{\mathcal{X}}\right) = \sup_{(h,h')\in\mathcal{H}^2} \Big| &\mathop{\mathbb{E}}_{\mathcal{P}_i^{\mathcal{X}}}\left[\mathcal{L}\left(h'\left(\mathbf{x}\right), h\left(\mathbf{x}\right)\right)\right] \\ &- \mathop{\mathbb{E}}_{P_i^{\mathcal{X}}}\left[\mathcal{L}\left(h'\left(\mathbf{x}\right), h(\mathbf{x})\right)\right]\Big|,\end{aligned} \tag{22}$$

where $\mathcal{P}_i^{\mathcal{X}}$ and $P_i^{\mathcal{X}}$ are marginal distributions of $\mathcal{P}_i$ and P_i on $\mathcal{X}$, respectively.

Definition 5. **(Error function.)** Let $\mathcal{L}$ be the bounded loss function satisfying the symmetric and triangle inequality. We define the error function for the distribution P_i as :

$$\mathcal{R}\left(h, P_i\right) = \frac{1}{n}\sum_{j=1}^{n} \mathcal{L}\left(h\left(\mathbf{x}_j^i\right), y_j^i\right), \tag{23}$$

where n is the number of samples and $\{\mathbf{x}_j^i, y_j^i\}$ is the j-th labelled sample drawn from P_i.

5.2 Forgetting analysis when the Teacher does not change its network architecture

In this section, we firstly analyze the forgetting behaviour of the Student when the Teacher does not change its network architecture.

Theorem 1. Let $h \in \mathcal{H}$ be the classifier of the Student. We define the risk bound of h for a certain task $(\mathcal{T}_i)$ at the t-th task learning as :

$$\begin{aligned}\mathcal{R}\left(h, P_i\right) \leq &\mathcal{R}'\left(h, h_i^{t-i}, \mathbb{P}_i^{t-i}\right) + \mathrm{Disc}\left(P_i^{\mathcal{X}}, \mathbb{P}_{i,\mathcal{X}}^{t-i}\right) \\ &+ \sigma\left(P_i, \mathbb{P}_i^{t-i}\right),\end{aligned} \tag{24}$$

where $\sigma(P_i, \mathbb{P}_i^{t-i})$ is the optimal combined error, given by :

$$\sigma\left(P_i, \mathbb{P}_i^{t-i}\right) = \mathcal{R}'\left(h_i^\star, h_i, P_i\right) + \mathcal{R}'\left(h_i, h_i^{t-i}, \mathbb{P}_i^{t-i}\right), \tag{25}$$

and we define $\mathcal{R}'(h_i^\star, h_i, P_i)$ as :

$$\mathcal{R}'\left(h_i^\star, h_i, P_i\right) = \frac{1}{n}\sum_{j=1}^{n} \mathcal{L}\left(h_i^\star\left(\mathbf{x}_j^i\right), h_i(x_j^i)\right), \tag{26}$$

and h_i^{t-i} is the optimal classifier for $\mathbb{P}_i^{t-i}$, expressed by :

$$h_i^{t-i} = \arg\min_{h\in\mathcal{H}} \mathcal{R}(h, \mathbb{P}_i^{t-i}). \tag{27}$$

$h_i^\star$ and h_i are the true labelling function and the optimal classifier for P_i.

The detailed proof can be seen in [42]. From Theorem 1 we can not explicitly know how the previously learnt knowledge is lost. In the following, we derive an analytical risk bound which can explain what factors would explain the forgetting of the Student while the Teacher continually learns multiple tasks.

Lemma 1. Let $h \in \mathcal{H}$ be the classifier of the Student. We define the analytical risk bound of h for the i-th dataset at the t-th task learning as :

$$\begin{aligned}\mathcal{R}\left(h, P_i\right) \leq &\mathcal{R}'(h, h_i^{t-i}, \mathbb{P}_i^{t-i}) \\ &+ \sum_{j=-1}^{t-i-1} \mathrm{Disc}(\mathbb{P}_{i,\mathcal{X}}^j, \mathbb{P}_{i,\mathcal{X}}^{j+1}) + \sigma(\mathbb{P}_i^j, \mathbb{P}_i^{j+1}).\end{aligned} \tag{28}$$

Proof. Firstly, we derive the risk bound between P_i and $\mathbb{P}_i^0$ according to Eq. (24) :

$$\begin{aligned}\mathcal{R}\left(h, P_i\right) \leq &\mathcal{R}'(h, h_i^0, \mathbb{P}_i^0) \\ &+ \mathrm{Disc}(P_i^{\mathcal{X}}, \mathbb{P}_{i,\mathcal{X}}^0) + \sigma(P_i, \mathbb{P}_i^0)\end{aligned} \tag{29}$$

Then we treat $\mathbb{P}_i^0$ as the target distribution and $\mathbb{P}_i^1$ as the source distribution, and derive the following risk bound :

$$\begin{aligned}\mathcal{R}'(h, h_i^0, \mathbb{P}_i^0) \leq &\mathcal{R}'(h, h_i^1, \mathbb{P}_i^1) + \mathrm{Disc}(\mathbb{P}_{i,\mathcal{X}}^0, \mathbb{P}_{i,\mathcal{X}}^1) \\ &+ \sigma(\mathbb{P}_i^0, \mathbb{P}_i^1)\end{aligned} \tag{30}$$

TABLE 2
The performance when learning a sequence of six tasks.

	MSE					SSMI					PSNR				
Datasets	BE-Stu	D-TS-KFD	D-TS-NLL	LGM	LTS	BE-Stu	D-TS-KFD	D-TS-NLL	LGM	LTS	BE-Stu	D-TS-KFD	D-TS-NLL	LGM	LTS
MNIST	24.32	23.71	23.21	18.97	26.96	0.89	0.90	0.90	0.90	0.89	21.52	21.55	21.66	22.62	21.03
SVHN	85.21	31.93	30.07	229.13	61.45	0.49	0.64	0.65	0.35	0.45	11.80	12.77	13.20	11.45	13.42
Fashion	167.38	43.72	42.14	90.62	81.56	0.47	0.71	0.71	0.15	0.56	13.36	17.75	17.95	11.67	16.52
IFashion	113.90	41.62	41.18	173.60	60.84	0.62	0.74	0.74	0.38	0.66	15.10	17.57	17.19	12.58	16.62
CIFAR10	359.09	203.70	208.72	676.13	220.72	0.21	0.35	0.33	0.04	0.33	14.91	15.08	15.19	12.22	15.33
Ommiglot	275.66	179.83	182.27	273.54	147.43	0.65	0.82	0.80	0.68	0.84	16.91	18.80	18.36	17.46	19.26
Average	170.93	**87.42**	87.93	243.66	99.83	0.56	0.69	0.69	0.42	0.62	15.60	17.25	**17.26**	14.67	17.03

In the same way, we can evaluate the risk bounds until learning the distribution $\mathbb{P}_i^{t-i}$ during lifelong learning :

$$\begin{aligned}\mathcal{R}'(h, h_i^1, \mathbb{P}_i^1) \leq & \mathcal{R}'(h, h_i^2, \mathbb{P}_i^2) + \mathrm{Disc}(\mathbb{P}_{i,\mathcal{X}}^1, \mathbb{P}_{i,\mathcal{X}}^2) \\ & + \sigma(\mathbb{P}_i^1, \mathbb{P}_i^2) \\ & \cdots \\ \mathcal{R}'(h, h_i^{t-i-1}, \mathbb{P}_i^{t-i-1}) \leq & \mathcal{R}'(h, h_i^{t-i}, \mathbb{P}_i^{t-i}) \\ & + \mathrm{Disc}(\mathbb{P}_{i,\mathcal{X}}^{t-i-1}, \mathbb{P}_{i,\mathcal{X}}^{t-i}) \\ & + \sigma(\mathbb{P}_i^{t-i-1}, \mathbb{P}_i^{t-i})\end{aligned} \tag{31}$$

Then we consider all these inequalities, resulting in :

$$\begin{aligned}\mathcal{R}\left(h, P_i\right) \leq & \mathcal{R}'(h, h_i^{t-i}, \mathbb{P}_i^{t-i}) \\ & + \sum_{j=-1}^{t-i-1} \mathrm{Disc}(\mathbb{P}_{i,\mathcal{X}}^j, \mathbb{P}_{i,\mathcal{X}}^{j+1}) + \sigma(\mathbb{P}_i^j, \mathbb{P}_i^{j+1}).\end{aligned} \tag{32}$$

where $\mathbb{P}_i^{-1}$ and $\mathbb{P}_i^0$ represent P_i and $\mathcal{P}_i$, respectively. h_i^{-1} and h_i^0 are the optimal classifiers for $\mathcal{P}_i^{-1}$ and $\mathcal{P}_i^0$, respectively.

Remark 1. We have the following observations from Lemma 1 :

- To ensure a tight bound for $\mathcal{T}_i$, the discrepancy distance between the generator's distribution and the target distribution must be minimized optimally during each task learning .
- While learning more tasks (t is increased), the accumulated errors (the last two terms in the right-hand side of Eq. (28)) can lead to a large gap between the target risk and the source risk, deteriorating the generalization performance of the Student.

Lemma 2. Based on the results from Eq. (28), the risk bound for all t tasks is defined as :

$$\begin{aligned}\sum_{i=1}^{t} \mathcal{R}\left(h, P_i\right) \leq & \sum_{i=1}^{t} \Big\{ \mathcal{R}'(h, h_i^{t-i}, \mathbb{P}_i^{t-i}) \\ + & \sum_{j=-1}^{t-i-1} \mathrm{Disc}(\mathbb{P}_{i,X}^j, \mathbb{P}_{i,X}^{j+1}) + \sigma(\mathbb{P}_i^j, \mathbb{P}_i^{j+1}) \Big\}.\end{aligned} \tag{33}$$

The proof consists of the summation of all risk bounds between each task and the model using Eq. (28).

Remark 2. Observations from Lemma 2 :

- The information learnt from the early tasks is forgotten more quickly than that learnt from recent tasks since learning early tasks results in more accumulated errors, representing the last two terms in the right-hand side of Eq. (33).
- A single generator used as Teacher can not guarantee the optimal performance of the Student when learning several entirely different tasks. As shown in Eq. (33), the optimal performance is achieved when the generator's distribution approximates the target distribution exactly, following the training with each task.

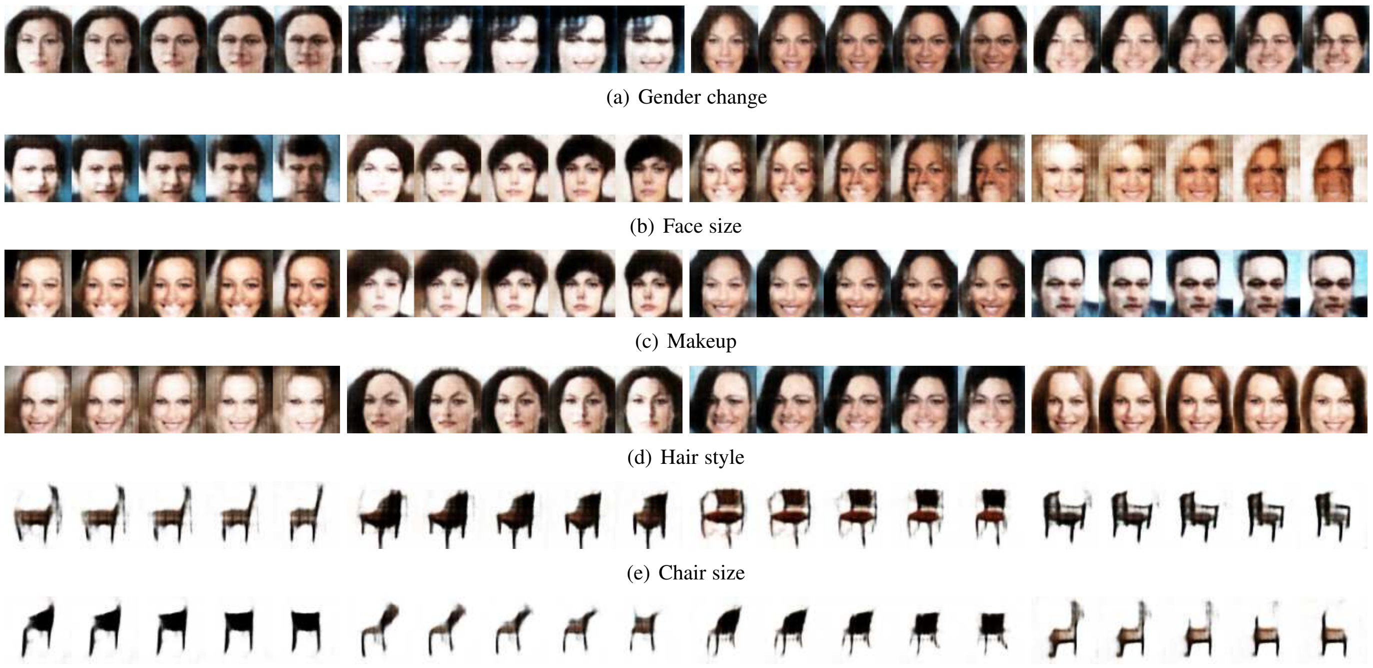
5.3 Forgetting analysis when the Teacher dynamically expands its network architecture

A single generator used as Teacher would not lead to the optimal performance for the Student when learning several different tasks, according to the discussion from Section 5.2. This is usually caused by the mode collapse [17] following repeat training processes [45]. The use of an expansion mechanism leading to a mixture of experts enhances the capacity of the Teacher to learn multiple tasks without suffering degeneration.

Let $\mathcal{A} = \{a_1, \cdots, a_n\}$ be the set which contains the task index for the tasks that are only used once for training. Let $\mathcal{B} = \{b_1, \cdots, b_{t-n}\}$ be the task index for these tasks that are trained more than once (the Teacher generates the pseudo samples and is retrained on these samples). Let $\mathcal{C} = \{c_1, \cdots, c_{t-n}\}$ be the set where each c_i represents how many times b_i is retrained. Let $\mathcal{U} = \{u_1, \cdots, u_t\}$ be the set where each u_i represents the component index of the Teacher for learning the i-th task. Let $|\cdot|$ represent the number of elements in a set.

Theorem 2. The risk bound for the Student when the Teacher changes its network architecture at the t-th task learning is given by :

$$\begin{aligned}& \sum_{i=1}^{|\mathcal{A}|} \{\mathcal{R}\left(h, P_{a_i}\right)\} + \sum_{i=1}^{|\mathcal{B}|} \{\mathcal{R}\left(h, P_{b_i}\right)\} \leq \\ & \sum_{i=1}^{|\mathcal{A}|-1} \Big\{ \mathcal{R}'(h, h_{a_i}^1, \mathbb{P}_{a_i}^1) + \mathrm{Disc}(\mathbb{P}_{a_i,X}^1, \mathbb{P}_{a_i,X}^{-1}) + \sigma(\mathbb{P}_{a_i}^1, \mathbb{P}_{a_i}^{-1}) \Big\} \\ & + \mathcal{R}'(h, h_t^0, \mathbb{P}_t^0) + \mathrm{Disc}(\mathbb{P}_{t,X}^0, \mathbb{P}_{t,X}^{-1}) + \sigma(\mathbb{P}_{a_i}^0, \mathbb{P}_t^{-1}) \\ & + \sum_{i=1}^{|\mathcal{B}|} \Big\{ \mathcal{R}'(h, h_{b_i}^{c_i}, \mathbb{P}_{b_i}^{c_i}) \\ & + \sum_{j=-1}^{c_i-1} \big\{ \mathrm{Disc}(\mathbb{P}_{i,X}^j, \mathbb{P}_{i,X}^{j+1}) + \sigma(\mathbb{P}_i^j, \mathbb{P}_i^{j+1}) \big\} \Big\}.\end{aligned} \tag{34}$$



(a) Gender change

(b) Face size

(c) Makeup

(d) Hair style

(e) Chair size

(f) Chair turning

Fig. 5. Results when varying a single latent variable from the latent space in between -3.0 and 3.0, while fixing the others, under the CelebA to 3D-Chair lifelong learning.

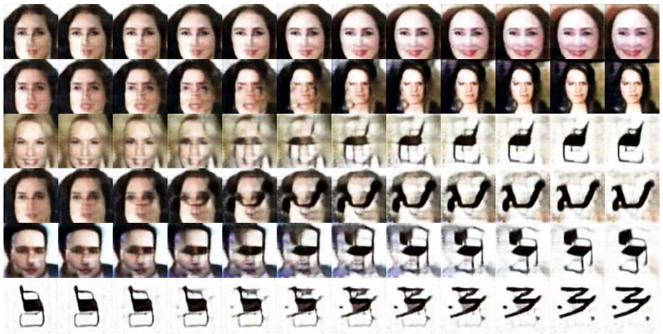


Fig. 6. Interpolation results under the CelebA, CACD, 3D-Chair and Omniglot lifelong learning.

Proof. Firstly, we consider the risk bound for the tasks that are trained only once, which can be derived by using Eq. (24) :

$$\sum_{i=1}^{|\mathcal{A}|}\mathcal{R}\left(h,P_{a_i}\right) \le \sum_{i=1}^{|\mathcal{A}|-1}\Big\{\mathcal{R}'(h,h^1_{a_i},\mathbb{P}^1_{a_i}) + \mathrm{Disc}(\mathbb{P}^1_{a_i,X},\mathbb{P}^{-1}_{a_i,X}) + \sigma(\mathbb{P}^1_{a_i},\mathbb{P}^{-1}_{a_i})\Big\} + R'(h,h^0_t,\mathbb{P}^0_t) + \mathrm{Disc}(\mathbb{P}^0_{t,X},\mathbb{P}^{-1}_{t,X}) + \sigma(\mathbb{P}^0_{a_i},\mathbb{P}^{-1}_t). \tag{35}$$

Secondly, we derive the risk bound for the tasks that are trained more than once by using Eq. (28) :

$$\sum_{i=1}^{|\mathcal{B}|}\mathcal{R}\left(h,P_{b_i}\right) \le \sum_{i=1}^{|\mathcal{B}|}\Big\{\mathcal{R}'(h,h^{c_i}_{b_i},\mathbb{P}^{c_i}_{b_i}) + \sum_{j=-1}^{c_i-1}\left\{\mathrm{Disc}(\mathbb{P}^j_{i,X},\mathbb{P}^{j+1}_{i,X}) + \sigma(\mathbb{P}^j_i,\mathbb{P}^{j+1}_i)\right\}\Big\}. \tag{36}$$

We sum up Eq. (35) and Eq. (36) and prove Theorem 2.

Remark 3. We have the following observations from Theorem 2 :

- If the Teacher dynamically builds t components, then $|\mathcal{B}| = 0$ and there are no accumulated errors. Then the discrepancy between the target distribution and the approximation distribution achieved by each component plays an important role for the generalization performance of the Student.
- If the Teacher does not expand, Eq. (34) is the same with Eq. (33), which has a large gap between the target risk and source risk. As the Teacher increases the number of components, $|\mathcal{A}|$ is increased and $|\mathcal{B}|$ is reduced, leading to a smaller gap in Eq. (34) since additional components would allow for each one to model a certain task only.
- The proposed expansion and selection process can achieve a tight bound in two ways. Firstly, KDS can help choose a component in which the discrepancy between the component's distribution and the distribution of the new task is small, allowing to represent the knowledge associated with the new task easily and avoid mode collapse. Secondly, when increasing the number of components, the accumulated errors for each component is reduced, given that each component is used for fewer GRMs.

6 Experiments

In the following we evaluate the Dynamic Self-Supervised Teacher-Student Network (D-TS) in lifelong learning tasks. We use the Adam optimization algorithm [72], with a learning rate of 0.0002 and the hyperparameter $\beta = 0.5$. The number of training epochs for each task is set to 20. In all experiments, we consider 60,000 randomly selected images from each database for training and 10,000 for testing, unless specified otherwise. The code is available at: https://github.com/dtuzi123/DynamicTeacherStudent.

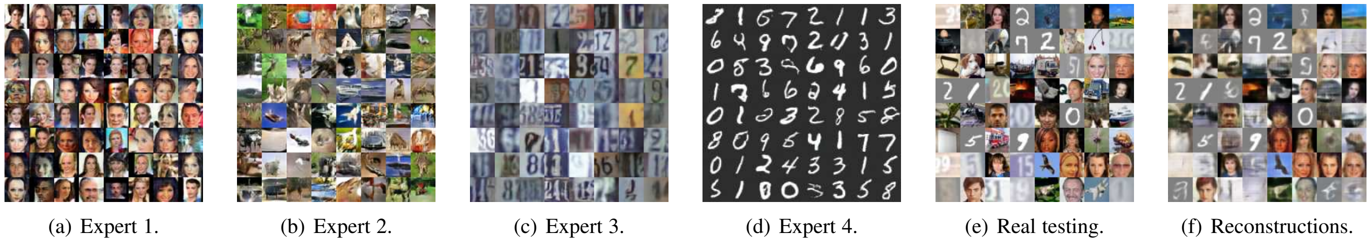

(a) Expert 1. (b) Expert 2. (c) Expert 3. (d) Expert 4. (e) Real testing. (f) Reconstructions.

Fig. 7. Generation and reconstruction of images when considering D-TS-KFD under CelebA, CACD, CIFAR10, Sub-ImageNet, SVHN and MNIST (CCCSSM) lifelong learning.

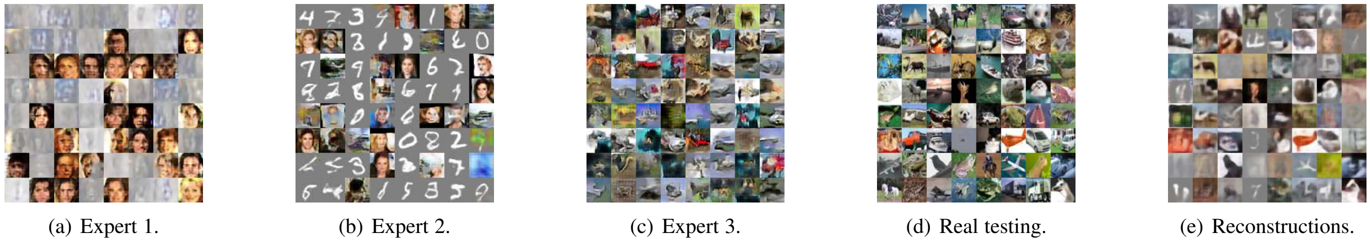

(a) Expert 1. (b) Expert 2. (c) Expert 3. (d) Real testing. (e) Reconstructions.

Fig. 8. Image generation and reconstruction from D-TS-NLL after the CCCSSM database sequence lifelong learning.

6.1 The evaluation of representation learning during unsupervised lifelong learning

We evaluate the performance of various methods for unsupervised lifelong learning. We consider five tasks, in a sequence called MFSIR, defined by the databases: MNIST [73], Fashion [74], SVHN [75], InverseFashion (IFashion) and Rotated MNIST (RMNIST). The results are reported in Table 1, where we use the threshold $hold = 150$ for D-TS-KFD and $hold = 50$ for D-TS-NLL for expanding the model in Eq. (5). We consider the Mean Square Error (MSE), the Structural Similarity Index Measure (SSIM) [76], and the Peak-Signal-to-Noise Ratio (PSNR) [76] for evaluating the image reconstruction quality.

We compare the proposed methodology with the Lifelong Teacher-Student (LTS) [30] which uses a large network architecture, defined as a single processing module, for the Teacher, and we consider the Lifelong Generative Modeling (LGM) [11]; we also adapt the BatchEnsemble [41] in order to train a Student model, under the unsupervised lifelong learning setting. We consider building an ensemble of VAEs as the Teacher module, where the number of components is equal to the number of tasks. Then we train a VAE as the Student module which accumulates knowledge from both the data generated by the Teacher and the tasks learnt during the lifelong learning. We can observe that the models D-TS-NLL or D-TS-KFD, which employ the knowledge distillation for D-TS, as explained in Section 3.3, by using either NLL or KFD, as the expert selection criterion, achieve the best result for every task.

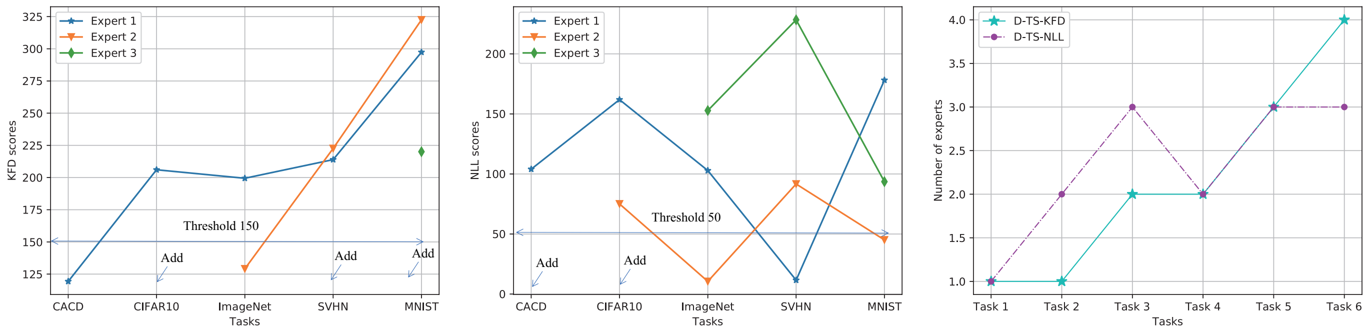
We evaluate the performance when learning a sequence of seven challenging tasks, defined by databases containing complex and diverse images: MNIST, SVHN, Fashion, IFashion (Inverse Fashion), CIFAR10 [77], Omniglot [78] and MNIST. We consider a threshold $hold = 150$ for both D-TS-KFD and D-TS-NLL. The results are provided in Table 2. The proposed method outperforms other models in this challenging learning setting.

We also investigate the procedure for adding a new expert by the proposed framework during LLL, by evaluating either KFD or NLL after each task switch, and the results are shown in Figures 4a and 4b. After learning the first task, KDS between the first expert and the next task (SVHN database), is 230 and therefore the Teacher module adds a new generator to learn SVHN. Then, after learning the third task, KDS between each expert and the next task (IFashion database) is smaller than 150, and therefore the Teacher module reuses the third expert in order to learn IFashion. KFD and NLL measures exhibit different characteristics. For instance, KFD is small when two tasks share similar visual concepts while for example the NLL score is small when two databases share similar global structures and colour palettes. The architecture expansion of the Teacher module is followed in Fig. 4c, where D-TS-KFD and D-TS-NLL lead to a reasonable number of experts, each capturing specific knowledge from the databases.

6.2 Study of the latent space of the Student

Projection of the latent variables. In order to investigate how the information from similar knowledge sources is embedded into the same cluster from the latent space, we project the latent variables extracted by the Student module considering images drawn from different domains: MNIST, SVHN, Fashion, IFashion and RMNIST (MSFIR sequence). For this analysis, we train D-TS under MSFIR lifelong learning and the threshold for adding a new component is set to $hold = 220$ in Eq. (5). After the training, we select a batch of 64 images for each domain. Then we use the inference model $q_\omega(\mathbf{z} \,|\, \mathbf{x})$ to produce the mean vector (hyperparameter of the Gaussian distribution) for each image and we average the results as $\mathbf{z}^*$ which is used as coordinates $(\mathbf{z}^*, \mathbf{z}^*)$ for each image in Fig. 3a and 3b, when considering and without the conditional prior, respectively. We can observe that the Student module embeds similar domains, as they are modelled by a certain expert from the Teacher, into the same cluster in the latent space. We can observe the overlap between the probabilistic representations of MNIST and RMNIST, as well as between Fashion and IFashion, where the latent spaces are better separated when using the conditional prior, according to Fig. 3a.

Lifelong learnt disentangled representations. In the following experiments we evaluate the ability of the D-TS model to create disentangled representations under the LLL, as discussed in Section 4.2. We train D-TS-KFD under CelebA to 3D-Chair lifelong



(a) KFD measure. (b) NLL measure. (c) Selected number of experts.

Fig. 9. Results for the measures used for the knowledge discrepancy score for the expansion of the Teacher module under the CCCSSM lifelong learning.

learning using the loss function from Eq. (17). After changing a single latent variable within the range [-3.0,3.0], while fixing the others, we obtain the disentangled results shown in Figures 5a-f. We can observe that the Student module can capture meaningful generative factors of images, such as changing the gender of the person, face size, face makeup, hair style, chair size or by rotating the object (chair) shown in the image.

Inter-domain interpolations enabled by lifelong learning. Interpolations in the latent spaces was previously used for exploring model representations [79]. Following the description from Section 4.3 we show that the proposed model not only that it can learn meaningful representations across domains over time, but it can also be used to explore the inter-domain latent spaces. We train the proposed D-TS model under the CelebA [80], CACD [81], 3D-Chair [82] and Omniglot (CCCO) lifelong learning for exploring their joint latent spaces. We show the interpolation results in Fig. 6, where we can observe how a human face can be smoothly transformed into images of multiple domains, while a chair is transformed into a human face when its frame gradually becomes the eyes and mouth. These results indicate that the Student module has additional modelling abilities and can capture surprising relationships between different latent space regions from multiple domains.

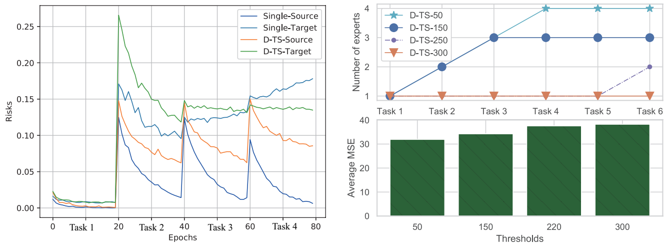
We train the proposed D-TS model considering CelebA, CACD, CIFAR10, Sub-ImageNet, SVHN and MNIST (CCCSSM) database lifelong learning. After the training, we extract the latent variables $\mathbf{z}_1$ and $\mathbf{z}_2$ from two images belonging to different domains $\mathbf{x}_1 \sim \mathcal{T}_i$, $\mathbf{x}_2 \sim \mathcal{T}_j$, respectively, using the Student module. We consider 1000 such image pairs from different domains. Then we generate the interpolated reconstructions :

$$\mathbf{x}' \sim p(\mathbf{x} \mid \mathbf{z}_1 * 0.2 + \mathbf{z}_2 * 0.8, \mathbf{u}). \tag{37}$$

The results when reconstructing the interpolated images in the latent space, by evaluating MSE, are provided in Table 3, where 'D-TS-KFD-Without' represents D-TS-KFD without using the regularized variable $\mathbf{u}$ in the KL divergence term $D_{KL}(q_\omega(\mathbf{z} \mid \mathbf{x}) \,||\, p(\mathbf{z} \mid \mathbf{u}))$ from Eq. (10) ($p(\mathbf{z} \mid \mathbf{u}) = \mathcal{N}(0, \mathbf{I})$ is fixed). These results indicate that D-TS-KFD can provide smaller reconstruction errors than other baselines such as LTS, which demonstrates that D-TS-KFD can learn a smooth latent space for multiple domains under LLL.

TABLE 3
MSE of the reconstructed interpolated images using Eq. (37).

Interpolation	D-TS-KFD	D-TS-KFD-Without	LTS
CelebA → CACD	208.53	249.51	440.11
CACD → CelebA	179.14	198.05	409.43
CIFAR10 → Sub-ImageNet	234.32	230.74	346.77
Sub-ImageNet →CIFAR10	221.35	218.82	316.61
Average	**210.84**	224.28	378.23



(a) Risks for D-TS-KFD and baseline. (b) Varying the threshold 'hold'.

Fig. 10. Risk bound evaluation and the performance of the Student in D-TS-KFD, when changing the threshold *hold*, from Eq. (5), in D-TS-hold.

6.3 Lifelong learning of databases with complex images

For this experiment, we train various models on CelebA, CACD, CIFAR10, Sub-ImageNet, SVHN and MNIST (CCCSSM) database sequence lifelong learning. These databases contain a variety of rather complex images showing human faces as well as natural images among others. We evaluate the Mean Square Error (MSE), Structural Similarity Index Measure (SSMI) and PSNR results for the reconstructed images from the six datasets, and the results are provided in Table 4. From this table we can observe that the proposed framework performs better on these complex image datasets, when compared to other methods by a large margin. The proposed D-TS-NLL, employing the NLL criterion for deciding whether to add or not a new expert, performs better than D-TS-KFD, which uses the KFD criterion, for the lifelong learning of sequences of complex and simple image databases. The visual results after learning the CCCSSM sequence of databases, when using either D-TS-KFD or D-TS-NLL, are provided in Fig. 7 and Fig. 8, respectively, where we can observe that each expert is able to capture information which is different from that associated

TABLE 4
Image reconstruction errors when learning CCCSSM sequence of datasets, containing complex images.

	MSE					**SSMI**					**PSNR**				
Datasets	LGM	D-TS-KFD	D-TS-NLL	BE-Stu	LTS	LGM	D-TS-KFD	D-TS-NLL	BE-Stu	LTS	LGM	D-TS-KFD	D-TS-NLL	BE-Stu	LTS
CelebA	703.62	137.67	141.47	153.25	215.43	0.05	0.55	0.56	0.54	0.40	12.18	18.42	18.81	19.00	16.37
CACD	979.18	160.66	123.49	265.80	246.99	0.03	0.58	0.65	0.45	0.44	10.86	18.15	19.39	16.80	16.12
CIFAR10	515.66	161.05	150.78	306.72	215.42	0.08	0.42	0.44	0.23	0.33	13.35	16.23	16.82	15.81	15.32
Sub-ImageNet	551.39	172.56	154.41	303.50	230.55	0.08	0.41	0.45	0.24	0.33	13.14	16.00	16.82	15.88	15.08
SVHN	62.15	28.76	34.08	52.71	34.90	0.20	0.65	0.62	0.50	0.60	13.50	12.95	13.70	13.43	13.94
MNIST	22.44	31.41	28.34	25.17	25.66	0.88	0.86	0.88	0.89	0.89	21.74	20.18	20.69	21.27	21.16
Average	472.51	115.35	**105.43**	184.53	161.49	0.22	0.58	**0.60**	0.48	0.50	14.13	16.99	**17.71**	17.03	16.33

TABLE 5
Classification accuracy under the supervised LLL of MNIST, Fashion, SVHN and InverseFashion (IFashion) databases.

Dataset	D-TS-KFD	LGM [11]	LGAN [13]	TS-EWC [83]	EWC [83]	D-TS-NLL	MeRGANs [84]	CURL [12]	BE-Stu [41]
MNIST	96.40	94.05	51.34	66.67	64.87	96.81	59.30	80.74	84.46
SVHN	65.21	47.24	48.16	55.63	54.12	68.68	55.31	68.46	62.78
Fashion	80.09	85.86	89.04	90.49	89.68	65.55	89.49	86.28	78.26
IFashion	86.68	89.08	92.15	92.30	92.76	88.48	92.17	91.48	81.94
Average	**82.09**	79.06	70.17	76.27	75.35	79.88	74.06	81.74	76.86

with any of the other experts. Furthermore, the Teacher module is able to embed, when appropriate, the information associated with two similar databases into a single expert, which accelerates the training speed and reduces the required memory. The Teacher module expansion, when trained with the CCCSSM sequence, is analysed in the plots from Figures 9a-c, when considering the threshold $hold = 150$ for D-TS-KFD and $hold = 50$ for D-TS-NLL, in Eq. (5). From Figures 9a,c we can observe that D-TS-KFD requires four experts for learning the CCCSSM sequence and is able to distinguish between different visual concepts from several tasks while assigning the relevant experts to the incoming tasks. For instance, the first expert only learns the probabilistic representations of CelebA and CACD, while the second learns those of CIFAR10 and Sub-ImangeNet databases. In contrast D-TS-NLL, as it can be seen from Fig. 9c, requires only three experts after the LLL.

6.4 Supervised learning

We evaluate the performance of the proposed approach in supervised classification tasks. The results when training for 20 epochs for the LLL of MNIST, SVHN, Fashion and IFashion, are provided in Table 5. We observe that GRM based methods used for comparison provide good results on the most recently learned tasks and tend to achieve a lower performance on the earlier tasks. In contrast, the proposed approach is able to balance its performance across all learned tasks during the supervised LLL. The Continual Unsupervised Representation Learning (CURL) [12] uses a mixture model and is better in three tasks than D-TS-KFD. For comparison, similar to the unsupervised learning we also consider the BatchEnsemble (BE) [41] as a Teacher-Student model in the supervised LLL setting. We first build an ensemble model as the Teacher, based on BE, where each expert contains a VAE and a classifier. We then train a classifier as the Student module which accumulates the predictive knowledge from both the Teacher and the tasks learned during the lifelong learning. During the lifelong supervised learning, each BE expert generates data samples and the associated classifier infers the class labels for the generated images. Then, the paired images and their corresponding classes are used to train the Student module in order to overcome catastrophic forgetting. We name this supervised model as BE-Stu. However, from the results in Table 5, BE-Stu performs worse than D-TS in every task.

6.5 Model complexity

In the following we evaluate the complexity of the models, by counting the number of parameters used for various unsupervised lifelong learning methods when considering three sets of databases: MFSIR - MNIST, Fashion, SVHN, IFashion and RMNIST; MSFICOM - MNIST, SVHN, Fashion, IFashion, CIFAR10, Omniglot and MNIST; CCCSSM - CelebA, CACD, CIFAR10, Sub-ImageNet, SVHN and MNIST. The results for the number of parameters required are provided in Table 6. D-TS-Stu represents the number of parameters for the Student module. From Table 6 it can be observed that the Student module has significantly fewer parameters while achieving the state of the art results when compared to other LLL methods.

6.6 Ablation study

Firstly, we consider a baseline model which uses a single GAN for the Teacher, as in LTS [30], and does not use the selection and dynamical expansion mechanism as proposed for D-TS model in this paper. We evaluate the source and target risks, where the former is evaluated for the training data and the latter for the testing data. All risks are calculated as the average classification errors by using the Student, across the LLL of MNIST, SVHN, Fashion, IFashion (MSFI sequence). The results are provided in Fig. 10a where ‘Single-Source’ represents the source risk evaluated by the baseline and ‘D-TS-Target’ represents the target

TABLE 6
Model complexity, evaluated as the number of parameters, for unsupervised lifelong learning models.

LLL sequence	LGM [11]	D-TS-KFD	D-TS-NLL	D-TS-Stu	BE-Stu	LTS [30]
MFSIR	3.3×10^8	2.3×10^8	2.3×10^8	8.0×10^7	4.7×10^8	3.3×10^8
MSTICOM	3.3×10^8	3.1×10^8	2.3×10^8	8.0×10^7	5.2×10^8	3.3×10^8
CCCSSM	3.3×10^8	3.1×10^8	2.3×10^8	8.0×10^7	5.2×10^8	3.3×10^8

risk calculated on all testing samples by using D-TS-KFD. We can observe that both D-TS-KFD and the baseline achieve low source risks but the baseline has a high target risk, which is conformed with Theorem 2.

The results for D-TS-KFD, when varying the threshold $hold \in \{50, 150, 250, 300\}$ in Eq. (5), are shown in the upper plot from Fig. 10b. We can observe that a lower threshold $hold$ leads to more components, while a single component is considered when $hold = 300$. From the bar-plot at the bottom of Fig. 10b we can observe that the reconstruction MSE error would decrease for $hold = 50$.

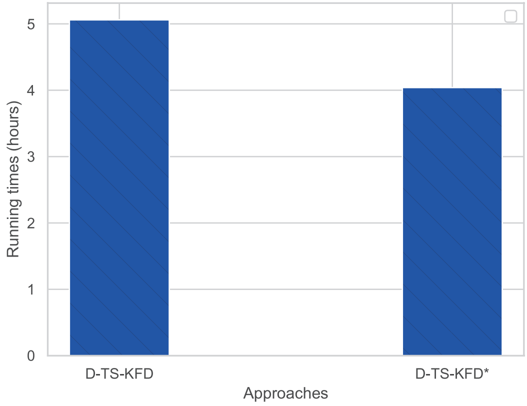
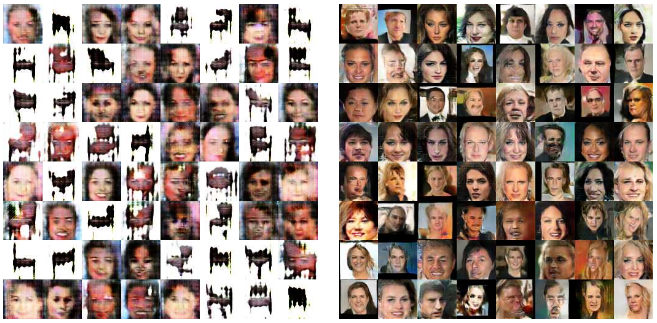


Fig. 11. The running time for D-TS-KFD when considering 20 epochs for both training and updating a component, while for D-TS-KFD* we consider only 5 epochs when updating a component under CelebA, CACD, CIFAR10, Sub-ImageNet, SVHN and MNIST lifelong learning.



(a) CelebA to 3D-Chair. (b) CelebA to CACD.

Fig. 12. Images generated by WGAN when considering GRMs.

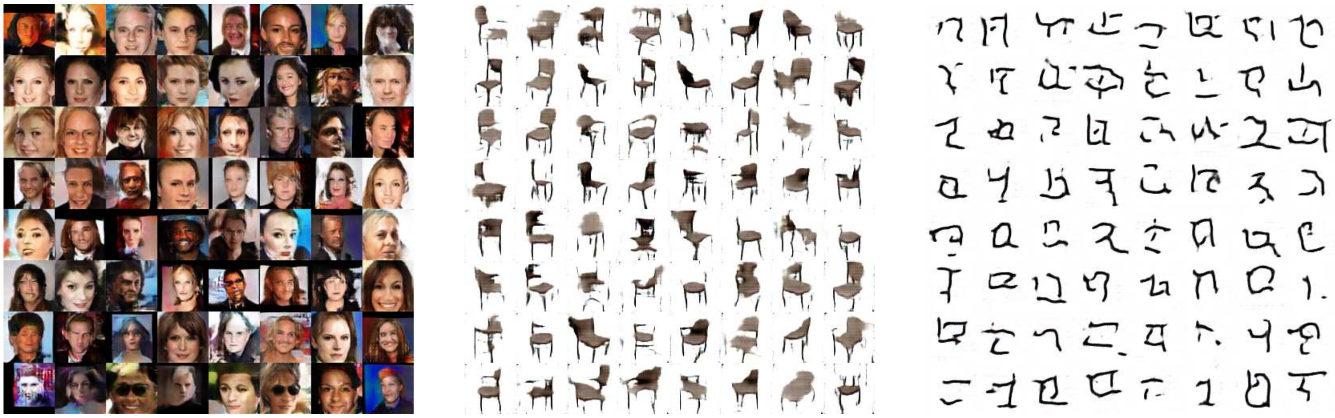
Robustness to the missing data during the training: In the following we evaluate whether the proposed framework can handle missing data well during the lifelong learning. We consider the learning setting, where the model is trained under the CelebA, CACD*, CIFAR10, Sub-ImageNet*, SVHN and MNIST lifelong learning. We create CACD* and Sub-ImageNet* by considering only 10,000 samples from each database, mixing them for training with another 60,000 samples from the other database, respectively. The average results are provided in Table 7. From these results we can observe that the proposed D-TS-KFD framework still achieves very good results despite having just a few training data from the original CACD* and Sub-ImageNet*. Meanwhile, LTS [30] tends to forget more information from the tasks learned earlier on during the LLL.

TABLE 7
The performance on testing data, when assuming that training data are missing for certain databases (marked with '*').

	MSE		SSMI	
Datasets	D-TS-KFD	LTS	D-TS-KFD	LTS
CelebA	185.54	312.24	0.43	0.26
CACD*	124.92	400.32	0.63	0.28
CIFAR10	164.68	330.85	0.41	0.21
Sub-ImageNet*	176.34	337.68	0.41	0.22
SVHN	31.37	40.47	0.63	0.54
MNIST	30.11	25.82	0.87	0.89
Average	118.83	241.23	0.56	0.40

Accelerating the future task learning: The proposed methodology is efficient in reusing the learned knowledge when updating an existing expert, based on the similarity of the accumulated knowledge during LLL with the information from a new database. These results indicate the ability of D-TS to accelerate the learning of those tasks which contain similar information to what was already learned in the past. In the following, we consider fewer training epochs when the model reuses a selected expert of the Teacher for learning the next task, according to Eq. (5). The results are provided in Table 8, where D-TS-KFD* denotes using only 5 training epochs for updating an existing component, while 20 epochs are used for training a new component. We observe that D-TS-KFD* still achieves good results while it also accelerates the training, as shown in the bar-plots from Figure 11, where D-TS-KFD* would reduce the time required for the full lifelong training by D-TS-KFD, where the latter uses 20 training epochs for both training a new component as well as when updating an existing one. Both D-TS-KFD* and D-TS-KFD use four experts for their Teacher models.

The mode collapse in GRM. We investigate how the mode collapse occurs during lifelong learning. We consider training a Wasserstein GAN (WGAN) [85] model on two databases, CelebA and 3D-Chair, which do not have any common characteristics, with the first representing human faces while the other contains images of chairs. In order to overcome the forgetfulness of WGAN, we use GRM [84] during the training. After the lifelong learning, we generate images using WGAN, which are shown in



(a) Expert 1.

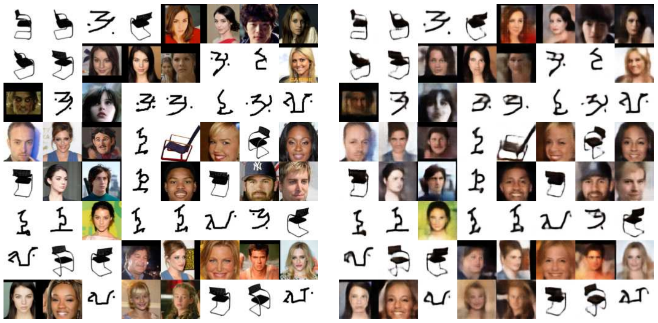
(b) Expert 2.

(c) Expert 3.

Fig. 13. Images generated by the Teacher module, containing 3 experts, from the proposed D-TS model.

TABLE 8
The results when considering just five training epochs for updating an existing component and the condition to expand the model is not fulfilled in Eq. (5).

	MSE		SSMI	
Datasets	D-TS-KFD*	LTS	D-TS-KFD*	LTS
CelebA	117.61	215.43	0.60	0.40
CACD	148.95	246.43	0.59	0.44
CIFAR10	177.95	215.42	0.40	0.33
Sub-ImageNet	190.47	230.55	0.39	0.33
SVHN	33.03	34.90	0.63	0.60
MNIST	32.00	25.66	0.86	0.89
Average	116.6	161.49	0.58	0.50



(a) Real images.

(b) Images reconstructed by the student module.

Fig. 14. Image reconstructions by the Student module from the proposed D-TS model, after the lifelong learning of CelebA, CACD, 3D-Chair and Omniglot.

Fig. 12a. From these images we can observe that WGAN with GRM cannot generate clear images for the two given domains, CelebA and 3D-Chair. The reason for this is that CelebA contains images which have completely different characteristics from those of the 3D-Chair database. In the following, we train a single WGAN with GRM under the CelebA and CACD lifelong learning and the images generated by the WGAN are shown in Fig. 12b. These generated images are of rather good quality. This shows that WGAN is able to learn multiple similar databases. However, existing GRM based methods cannot be applied to long sequences of tasks, where the datasets are entirely different from each other.

The drawback outlined by this example motivated us to develop a novel dynamical memory system for the Teacher module. The proposed Knowledge Discrepancy Score (KDS) can detect and identify the novelty of the incoming tasks and guides the Teacher module to expand its capacity in order to learn databases containing images with entirely different characteristics. In the following we train the proposed Dynamic Self-Supervised Lifelong Teacher-Student Learning model D-TS-KFD under the CelebA, CACD, 3D-chair and Omniglot lifelong learning. After the LLL, our Teacher module adds two new experts to the initial one, and the images generated by the 3 experts of the Teacher module are shown in Fig. 13a-c. In Fig. 13a, we can observe that Expert 1 captures well the information from databases with images from the same category (human faces) such as CelebA and CACD. We also show the reconstructions made by the Student module in Fig. 14. We can observe that the Student module is also able to provide high-quality reconstructions across domains. These results indicate that the proposed Lifelong D-TS provides better results than Generative Reply Mechanism (GRM) methods.

7 Conclusion and Future Work

A novel Dynamic Self-Supervised Teacher-Student Network (D-TS) learning framework, capable of continually learning data representations without forgetting, is proposed in this paper. The model is made up of a Teacher module, which is allowed to expand its architecture with new components, and a Student module. The Knowledge Discrepancy Score (KDS) criterion is proposed for comparing the probabilistic representations of incoming data with the information already acquired by the Teacher. For implementing KDS we consider two measures : the Knowledge Frécher Distance (KFD) and the Negative Log-likelihood (NLL). A new component (expert) is added to the Teacher module when KDS is above a certain threshold, when learning a new database. Otherwise, the most efficient and flexible component is selected by the KDS from the mixture of experts forming the Teacher, in order to be updated with the information from the new database. The selection mechanism contributes to reusing the learned knowledge for accelerating the future task learning. In the experimental results we show that D-TS can train a compressed Student module which outperforms other methods in various multi-task applications while also requiring fewer parameters to train. Our future work will extend the proposed D-TS framework to the Task-Free Continual Learning (TFCL) where the task information is not provided during the training. TFCL represents a more realistic continual learning setting which has not been sufficiently explored.

APPENDIX A
THE DERIVATION OF THE LOG-LIKELIHOOD

In the following we provide the detailed derivation of the log-likelihood from Eq. (8) of the paper, where we consider the independence between the continuous $\mathbf{z}$ and the expert $\mathbf{u}$ variables.

$$\begin{aligned}\log p(\mathbf{x}) \geq & \mathbb{E}_{q(\mathbf{z},\mathbf{u}|\mathbf{x})}\left[\log\frac{p(\mathbf{x},\mathbf{z},\mathbf{u})}{q(\mathbf{z},\mathbf{u}|\mathbf{x})}\right] \\ =& \mathbb{E}_{q(\mathbf{z}|\mathbf{x})q(\mathbf{u}|\mathbf{x})}\left[\log\frac{p(\mathbf{x}|\mathbf{z},\mathbf{u})p(\mathbf{z})p(\mathbf{u})}{q(\mathbf{z}|\mathbf{x})q(\mathbf{u}|\mathbf{x})}\right] \\ =& \mathbb{E}_{q(\mathbf{z}|\mathbf{x})q(\mathbf{u}|\mathbf{x})}\left[\log p(\mathbf{x}|\mathbf{z},\mathbf{u})\right]+ \\ & \mathbb{E}_{q(\mathbf{z}|\mathbf{x})q(\mathbf{u}|\mathbf{x})}\left[\log\frac{p(\mathbf{z})}{q(\mathbf{z}|\mathbf{x})}\right]+\mathbb{E}_{q(\mathbf{z}|\mathbf{x})q(\mathbf{u}|\mathbf{x})}\left[\log\frac{p(\mathbf{u})}{q(\mathbf{u}|\mathbf{x})}\right] \\ =& \mathbb{E}_{q(\mathbf{z}|\mathbf{x})q(\mathbf{u}|\mathbf{z})}\left[\log p(\mathbf{x}|\mathbf{z},\mathbf{u})\right]-D_{KL}(q(\mathbf{z}|\mathbf{x})||p(\mathbf{z})) \\ & -D_{KL}(q(\mathbf{u}|\mathbf{x})||p(\mathbf{u}))\end{aligned} \tag{38}$$

The expert-variable $\mathbf{u}$ is defined by the specific experts from the mixture, or by the new task, and is used as the ground-truth label for training the expert-inference network defined by Eq. (11).

APPENDIX B
PROOF OF PROPOSITION 1

In order to provide a general proof for the lower bound on $D_{KL}(\mathbb{S}||\mathbb{P}_\theta)$, we first consider a simple Student module $p_\theta(\mathbf{x}) = p_\theta(\mathbf{x}|\mathbf{z})p_\theta(\mathbf{z})$ which has a latent variable vector $\mathbf{z}$. Then, we consider the KL divergence between $p_\theta(\mathbf{z}|\mathbf{x})$ and $q(\mathbf{z}|\mathbf{x})$, [86]:

$$D_{KL}(q(\mathbf{z}|\mathbf{x})||p_\theta(\mathbf{z}|\mathbf{x})) = \mathbb{E}_{q(\mathbf{z}|\mathbf{x})}[\log q(\mathbf{z}|\mathbf{x}) - \log p_\theta(\mathbf{z}|\mathbf{x})], \tag{39}$$

we apply the Bayes rule to $p_\theta(\mathbf{z}|\mathbf{x})$ and the above equation can be rewritten as:

$$\begin{aligned}D_{KL}(q(\mathbf{z}|\mathbf{x})||p_\theta(\mathbf{z}|\mathbf{x})) = \mathbb{E}_{q(\mathbf{z}|\mathbf{x})}[&\log q(\mathbf{z}|\mathbf{x}) - \log p_\theta(\mathbf{x}|\mathbf{z}) \\ &- \log p(\mathbf{z})] + \log p_\theta(\mathbf{x}).\end{aligned} \tag{40}$$

Then, by rewriting the expression from above, $\log p_\theta(\mathbf{x})$ is expressed as:

$$\begin{aligned}\log p_\theta(\mathbf{x}) = & \mathbb{E}_{q(\mathbf{z}|\mathbf{x})}[\log p_\theta(\mathbf{x}|\mathbf{z})] - D_{KL}(q(\mathbf{z}|\mathbf{x})||p(\mathbf{z})) \\ & + D_{KL}(q(\mathbf{z}|\mathbf{x})||p_\theta(\mathbf{z}|\mathbf{x}))\end{aligned} \tag{41}$$

Through Proposition 1, we aim to maximize $\mathbb{E}_{\mathbf{x}\sim\mathbb{S}}[\log p_\theta(\mathbf{x})]$ which becomes:

$$\begin{aligned}\mathbb{E}_{\mathbf{x}\sim\mathbb{S}}[\log p_\theta(\mathbf{x})] = & \int S(\mathbf{x})\left\{\mathbb{E}_{q(\mathbf{z}|\mathbf{x})}[\log p_\theta(\mathbf{x}|\mathbf{z})]\right. \\ & \left.-D_{KL}(q(\mathbf{z}|\mathbf{x})||p(\mathbf{z})) + D_{KL}(q(\mathbf{z}|\mathbf{x})||p_\theta(\mathbf{z}|\mathbf{x}))\right\} d\mathbf{x}\end{aligned} \tag{42}$$

where $S(\mathbf{x})$ is the density function of $\mathbb{S}$. We move the last term of the right-hand side to the left-hand side:

$$\begin{aligned}&\mathbb{E}_{\mathbf{x}\sim\mathbb{S}}[\log p_\theta(\mathbf{x})] - \int S(\mathbf{x})\left\{D_{KL}(q(\mathbf{z}|\mathbf{x})||p_\theta(\mathbf{z}|\mathbf{x}))\right\} d\mathbf{x} = \\ &\int S(\mathbf{x})\left\{\mathbb{E}_{q(\mathbf{z}|\mathbf{x})}[\log p_\theta(\mathbf{x}|\mathbf{z})]] - D_{KL}(q(\mathbf{z}|\mathbf{x})||p(\mathbf{z}))\right\} d\mathbf{x}\end{aligned} \tag{43}$$

From the fact that $D_{KL}(\cdot,\cdot) \geq 0$ we conclude that the right hand side is a lower bound on $\mathbb{E}_{\mathbf{x}\sim\mathbb{Q}}[\log p_\theta(\mathbf{x})]$ and the gap to $\mathbb{E}_{\mathbf{x}\sim\mathbb{S}}[\log p_\theta(\mathbf{x})]$ is equal to $\mathbb{E}_{\mathbb{S}}[D_{KL}(q(\mathbf{z}|\mathbf{x})||p_\theta(\mathbf{z}|\mathbf{x}))]$.

Extension to two types of latent variables, continuous and discrete. In the following, we consider that the Student module has two types of latent variables, $\mathbf{z}$ and $\mathbf{u}$, representing continuous and discrete variables, and we have :

$$p_\theta(\mathbf{x}) = p_\theta(\mathbf{x}|\mathbf{z},\mathbf{u})p(\mathbf{z},\mathbf{u}) \tag{44}$$

and we consider the following KL divergence:

$$\begin{aligned}D_{KL}(q(\mathbf{z},\mathbf{u}|\mathbf{x})||p_\theta(\mathbf{z},\mathbf{u}|\mathbf{x})) = \mathbb{E}_{q(\mathbf{z},\mathbf{u}|\mathbf{x})}[&\log q(\mathbf{z},\mathbf{u}|\mathbf{x}) \\ &- \log p_\theta(\mathbf{z},\mathbf{u}|\mathbf{x})]\end{aligned} \tag{45}$$

By considering the Bayes rule we have :

$$p_\theta(\mathbf{z},\mathbf{u}|\mathbf{x}) = \frac{p_\theta(\mathbf{x}|\mathbf{z},\mathbf{u})p(\mathbf{z},\mathbf{u})}{p_\theta(\mathbf{x})} \tag{46}$$

and after replacing in (45) we obtain :

$$\begin{aligned}&D_{KL}(q(\mathbf{z},\mathbf{u}|\mathbf{x})||p_\theta(\mathbf{z},\mathbf{u}|\mathbf{x})) = \mathbb{E}_{q(\mathbf{z},\mathbf{u}|\mathbf{x})}[\log q(\mathbf{z},\mathbf{u}|\mathbf{x}) \\ &- \log p_\theta(\mathbf{x},\mathbf{z},\mathbf{u})] + \log p_\theta(\mathbf{x})\end{aligned} \tag{47}$$

We rewrite the above equation by moving terms from the right-hand side to the left-hand side :

$$\begin{aligned}&\log p_\theta(\mathbf{x}) = \mathbb{E}_{q(\mathbf{z},\mathbf{u}|\mathbf{x})}[\log p_\theta(\mathbf{x},\mathbf{z},\mathbf{u}) - \log q(\mathbf{z},\mathbf{u}|\mathbf{x})] \\ &+ D_{KL}(q(\mathbf{z},\mathbf{u}|\mathbf{x})||p_\theta(\mathbf{z},\mathbf{u}|\mathbf{x}))\end{aligned} \tag{48}$$

Then we consider $\mathbb{E}_{\mathbf{x}\sim\mathbb{S}}[\log p_\theta(\mathbf{x})]$, similar to (42) :

$$\begin{aligned}&\mathbb{E}_{\mathbf{x}\sim\mathbb{S}}[\log p_\theta(\mathbf{x})] = \int S(\mathbf{x})\left\{\mathbb{E}_{q(\mathbf{z},\mathbf{u}|\mathbf{x})}[\log p(\mathbf{x},\mathbf{z},\mathbf{u})\right. \\ &\left.- \log q(\mathbf{z},\mathbf{u}|\mathbf{x})] + D_{KL}(q(\mathbf{z},\mathbf{u}|\mathbf{x})||p(\mathbf{z},\mathbf{u}|\mathbf{x}))\right\} d\mathbf{x}\end{aligned} \tag{49}$$

We move the last term from the right hand side to the left hand side:

$$\begin{aligned}&\mathbb{E}_{\mathbf{x}\sim\mathbb{S}}[\log p_\theta(\mathbf{x})] - \int S(\mathbf{x})\left\{D_{KL}(q(\mathbf{z},\mathbf{u}|\mathbf{x})||p(\mathbf{z},\mathbf{u}|\mathbf{x}))\right\} d\mathbf{x} = \\ &\int S(\mathbf{x})\left\{\mathbb{E}_{q(\mathbf{z},\mathbf{u}|\mathbf{x})}[\log p_\theta(\mathbf{x},\mathbf{z},\mathbf{u}) - \log q(\mathbf{z},\mathbf{u}|\mathbf{x})]\right\} d\mathbf{x}\end{aligned} \tag{50}$$

where the right hand side is a lower bound on $\mathbb{E}_{\mathbf{x}\sim\mathbb{S}}[\log p_\theta(\mathbf{x})]$ and the gap to $\mathbb{E}_{\mathbf{x}\sim\mathbb{S}}[\log p_\theta(\mathbf{x})]$ is equal to $\mathbb{E}_{\mathbb{S}}[D_{KL}(q(\mathbf{z},\mathbf{u}|\mathbf{x})||p_\theta(\mathbf{z},\mathbf{u}|\mathbf{x}))]$. Eventually, we can maximize this lower bound to approximate $\mathbb{E}_{\mathbf{x}\sim\mathbb{S}}[\log p_\theta(\mathbf{x})]$:

$$\mathbb{E}_{\mathbf{x}\sim\mathbb{S}}[\log p_\theta(\mathbf{x})] \geq \mathbb{E}_{\mathbb{S}}\left\{\mathbb{E}_{q(\mathbf{z},\mathbf{u}|\mathbf{x})}\left[\log\frac{p_\theta(\mathbf{x},\mathbf{z},\mathbf{u})}{q(\mathbf{z},\mathbf{u}|\mathbf{x})}\right]\right\} \tag{51}$$

□

We should note that $\mathbb{S}$ is treated as a mixture distribution since the Teacher module has several experts.

APPENDIX C
THE OBJECTIVE FUNCTION FOR THE STUDENT MODULE

The Kullback–Leibler (KL) divergence measures the distance between two distributions. In the proposed framework, the Teacher module can have several experts. We assume that at the t-th task learning, the Teacher module has K experts and the information recorded by these experts is represented by the distributions $\mathbb{P}_1, \dots, \mathbb{P}_K$. As demonstrated in Proposition 1, the minimization of the KL divergence between the probabilistic representations of the Teacher and Student, defined by $\mathbb{S}$ and $\mathbb{P}_\theta$, respectively, is expressed by maximizing $\mathbb{E}_{\mathbb{S}}\{\log p_\theta(\mathbf{x})\}$:

$$\begin{aligned}\mathbb{E}_{\mathbb{S}}\{\log p_\theta(\mathbf{x})\} &= \frac{1}{Z}\sum_{i=1}^{K}\int \pi_i m_{\varepsilon_i}\left(\mathbf{x}\right)\log p_\theta\left(\mathbf{x}\right) d\mathbf{x} \\ &= \frac{1}{Z}\sum_{i=1}^{K}\pi_i \mathbb{E}_{\mathbf{x}\sim\mathbb{P}_i}\left[\log p_\theta\left(\mathbf{x}\right)\right],\end{aligned} \tag{52}$$

where $m_{\varepsilon_i}(\mathbf{x})$ is the density function for $\mathbb{P}_i$, where ε_i are the parameters of the i-th expert, and the optimization problem consists of maximizing each $\mathbb{E}_{\mathbb{P}_i}\left[\log p_\theta\left(\mathbf{x}\right)\right]$:

$$\mathbb{E}_{\mathbb{P}_i}\left[\log p_\theta\left(\mathbf{x}\right)\right] = \int m_{\varepsilon_i}\left(\mathbf{x}\right)\log p_\theta\left(\mathbf{x}\right) d\mathbf{x} \tag{53}$$

From the proof of Proposition 1, provided in Appendix B, we show how to approximate $\mathbb{E}_{\mathbb{P}_i(\mathbf{x})}\left[\log p_\theta\left(\mathbf{x}\right)\right]$ by maximizing a lower bound, after considering the relationship (51), we have :

$$\begin{aligned}&\int m_{\varepsilon_i}\left(\mathbf{x}\right)\log p_\theta(\mathbf{x}) d\mathbf{x} \geq \\ &\int m_{\varepsilon_i}\left(\mathbf{x}\right)\left(\mathbb{E}_{q(\mathbf{z},\mathbf{u}|\mathbf{x})}\left[\log\frac{p(\mathbf{x},\mathbf{z},\mathbf{u})}{q(\mathbf{z},\mathbf{u}|\mathbf{x})}\right]\right) d\mathbf{x}\end{aligned} \tag{54}$$

The intractable optimization problem $D_{KL}(\mathbb{S}||\mathbb{P}_\theta)$ can be addressed by maximizing the summation of tractable lower bounds of each $\mathbb{E}_{\mathbb{P}_i}\left[\log p_\theta\left(\widehat{\mathbf{x}}_i\right)\right]$ (the knowledge distillation term):

$$\begin{aligned}&\sum\nolimits_{i=1}^{k}\pi_i\mathbb{E}_{\mathbb{P}_i}\log p_\theta\left(\widehat{\mathbf{x}}_i\right) \geq \\ &\left(\sum\nolimits_{i=1}^{k}\pi_i\mathbb{E}_{\widehat{\mathbf{x}}_i\sim\mathbb{P}_i}\mathbb{E}_{q(\mathbf{z},\mathbf{u}|\widehat{\mathbf{x}}_i)}\left[\log\frac{p\left(\widehat{\mathbf{x}}_i,\mathbf{z},\mathbf{u}\right)}{q\left(\mathbf{z},\mathbf{u}|\widehat{\mathbf{x}}_i\right)}\right]\right),\end{aligned} \tag{55}$$

where $\widehat{\mathbf{x}}_i$ are samples drawn from $\mathbb{P}_i$. Eventually, the Student training objective function involves the knowledge distillation term as well as the ELBO on the data samples from the given data associated with the t-th task learning :

$$\begin{aligned}\mathcal{L}_{Stu} \triangleq &\underbrace{\left(\sum\nolimits_{i=1}^{k}\pi_i\mathbb{E}_{\widehat{\mathbf{x}}_i\sim\mathbb{P}_i}\mathbb{E}_{q(\mathbf{z},\mathbf{u}|\widehat{\mathbf{x}}_i)}\left[\log\frac{p\left(\widehat{\mathbf{x}}_i,\mathbf{z},\mathbf{u}\right)}{q\left(\mathbf{z},\mathbf{u}|\widehat{\mathbf{x}}_i\right)}\right]\right)}_{\text{Knowledge distillation optimization}} \\ &+ \underbrace{\mathbb{E}_{q(\mathbf{z},\mathbf{u}|\mathbf{x}_t)}\left[\log\frac{p\left(\mathbf{x}_t,\mathbf{z},\mathbf{u}\right)}{q\left(\mathbf{z},\mathbf{u}|\mathbf{x}_t\right)}\right]}_{\text{ELBO on the } t\text{-th task}}.\end{aligned} \tag{56}$$

References

[1] M. Delange, R. Aljundi, M. Masana, S. Parisot, X. Jia, A. Leonardis, G. Slabaugh, and T. Tuytelaars, "A continual learning survey: Defying forgetting in classification tasks," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 44, no. 7, pp. 3366–3385, 2022.

[2] D. Silver, T. Hubert, J. Schrittwieser, I. Antonoglou, M. Lai, A. Guez, M. Lanctot, L. Sifre, D. Kumaran, T. Graepel *et al.*, "A general reinforcement learning algorithm that masters chess, shogi, and go through self-play," *Science*, vol. 362, no. 6419, pp. 1140–1144, 2018.

[3] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein *et al.*, "Imagenet large scale visual recognition challenge," *International journal of computer vision*, vol. 115, no. 3, pp. 211–252, 2015.

[4] 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.

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

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

[7] A. Chaudhry, M. Rohrbach, M. Elhoseiny, T. Ajanthan, D. Dokania, P. H. S. Torr, and M. Ranzato, "On tiny episodic memories in continual learning," *arXiv preprint arXiv:1902.10486*, 2019.

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

[9] G. Sun, Y. Cong, J. Dong, Y. Liu, Z. Ding, and H. Yu, "What and how: Generalized lifelong spectral clustering via dual memory," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 44, no. 7, pp. 3895–3908, 2022.

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

[11] J. Ramapuram, M. Gregorova, and A. Kalousis, "Lifelong generative modeling," in *Proc. Int. Conf. on Learning Representations (ICLR), arXiv preprint arXiv:1705.09847*, 2017.

[12] D. Rao, F. Visin, A. A. Rusu, Y. W. Teh, R. Pascanu, and R. Hadsell, "Continual unsupervised representation learning," in *Advances Neural Information Processing Systems (NeurIPS)*, 2020, pp. 7645–7655.

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

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

[15] 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.

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

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

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

[19] F. Ye and A. G. Bors, "Lifelong generative modelling using dynamic expansion graph model," in *Proc. AAAI on Artificial Intelligence*, vol. 36, 2022, pp. 8857–8865.

[20] S. Mohamed and B. Lakshminarayanan, "Learning in implicit generative models," in *Proc. Int. Conf. on Learning Representations (ICLR), arXiv preprint arXiv:1610.03483*, 2017.

[21] G. Chen, W. Choi, X. Yu, T. Han, and M. Chandraker, "Learning efficient object detection models with knowledge distillation," in *Advances in Neural Information Processing Systems (NIPS)*, 2017, pp. 742–751.

[22] H. Chen, Y. Wang, C. Xu, Z. Yang, C. Liu, B. Shi, C. Xu, C. Xu, and Q. Tian, "Data-free learning of student networks," in *Proc. IEEE Int. Conf. on Computer Vision (ICCV)*, 2019, pp. 3514–3522.

[23] B. Heo, M. Lee, S. Yun, and J. Y. Choi, "Knowledge distillation with adversarial samples supporting decision boundary," in *Proc. of the AAAI Conf. on Artificial Intelligence*, vol. 33, 2019, pp. 3771–3778.

[24] G. Hinton, O. Vinyals, and J. Dean, "Distilling the knowledge in a neural network," in *Proc. NIPS Deep Learning Workshop, arXiv preprint arXiv:1503.02531*, 2015.

[25] J. Yim, D. Joo, J. Bae, and J. Kim, "A gift from knowledge distillation: Fast optimization, network minimization and transfer learning," in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 4133–4141.

[26] M. G. Andrey Malinin, Bruno Mlodozeniec, "Ensemble distribution distillation," in *Int. Conf. on Learning Representations (ICLR), arXiv preprint arXiv:1905.00076*, 2020.

[27] E. Englesson and H. Azizpour, "Efficient evaluation-time uncertainty estimation by improved distillation," in *Proc. ICML-Workshop on Uncertainty & Robustness in Deep Learning, arXiv preprint arXiv:1906.05419*, 2019.

[28] T. Furlanello, Z. Lipton, M. Tschannen, L. Itti, and A. Anandkumar, "Born again neural networks," in *Proc. Int. Conf. on Machine Learning (ICML), vol. PMLR 80*, 2018, pp. 1607–1616.

[29] Z. Li and D. Hoiem, "Improving confidence estimates for unfamiliar examples," in *Proc. of the IEEE/CVF Int. Conf. on Computer Vision (CVPR)*, 2020, pp. 2686–2695.

[30] F. Ye and A. Bors, “Lifelong teacher-student network learning,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 44, no. 10, pp. 6280–6296, 2022.

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

[32] C.-Y. Hung, C.-H. Tu, C.-E. Wu, C.-H. Chen, Y.-M. Chan, and C.-S. Chen, “Compacting, picking and growing for unforgetting continual learning,” in *Advances in Neural Information Processing Systems (NeurIPS)*, 2019, pp. 13 669–13 679.

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

[34] 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.

[35] 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.

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

[37] G. Zhou, K. Sohn, and H. Lee, “Online incremental feature learning with denoising autoencoders,” in *Proc. Artificial intelligence and statistics (AISTATS), vol. PMLR 22*, 2012, pp. 1453–1461.

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

[39] ——, “Mixtures of variational autoencoders,” in *Proc. Int. Conf. on Image Processing Theory, Tools and Applications (IPTA)*, 2020, pp. 1–6.

[40] ——, “Lifelong mixture of variational autoencoders,” *IEEE Trans. on Neural Networks and Learning Systems*, pp. 1–14, 2021.

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

[42] F. Ye and A. G. Bors, “Lifelong infinite mixture model based on knowledge-driven Dirichlet process,” in *Proc. IEEE Int. Conf. on Computer Vision (ICCV)*, 2021, pp. 1–10.

[43] ——, “Lifelong learning of interpretable image representations,” in *Proc. Int. Conf. on Image Processing Theory, Tools and Applications (IPTA)*, 2020, pp. 1–6.

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

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

[46] ——, “Lifelong twin generative adversarial networks,” in *Proc. IEEE Int. Conf. on Image Processing (ICIP)*, 2021.

[47] M. Arjovsky, S. Chintala, and L. Bottou, “Wasserstein GAN,” *arXiv preprint arXiv:1701.07875*, 2017.

[48] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, and A. C. Courville, “Improved training of Wasserstein GANs,” in *Advances in Neural Inf. Proc. Systems (NIPS)*, 2017, pp. 5769–5779.

[49] J. Goldberger, S. Gordon, H. Greenspan *et al.*, “An efficient image similarity measure based on approximations of KL-divergence between two Gaussian mixtures,” in *Proc. IEEE Int. Conf. on Computer Vision (ICCV)*, vol. 3, 2003, pp. 487–493.

[50] M. I. Belghazi, A. Baratin, S. Rajeshwar, S. Ozair, Y. Bengio, A. Courville, and D. Hjelm, “Mutual information neural estimation,” in *Proc. Inter. Conference on Machine Learning (ICML), vol. PMLR 80*, 2018, pp. 531–540.

[51] H. Liu, X. Gu, and D. Samaras, “Wasserstein GAN with quadratic transport cost,” in *Proc. of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019, pp. 4832–4841.

[52] L. A. Gatys, A. S. Ecker, and M. Bethge, “Image style transfer using convolutional neural networks,” in *Proc. of IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 2414–2423.

[53] J. Yosinski, J. Clune, Y. Bengio, and H. Lipson, “How transferable are features in deep neural networks?” in *Advances in Neural Information Processing Systems (NIPS)*, 2014, pp. 3320–3328.

[54] D. Dowson and B. Landau, “The Fréchet distance between multivariate Normal distributions,” *Journal of Multivariate Analysis*, vol. 12, no. 3, pp. 450–455, 1982.

[55] 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 in Neural Information Processing Systems (NIPS)*, 2017, pp. 6626–6637.

[56] X. Wang, R. Zhang, Y. Sun, and J. Qi, “KDGAN: knowledge distillation with generative adversarial networks,” in *Advances in Neural Inf. Proc. Systems (NeurIPS)*, 2018, pp. 775–786.

[57] E. J. Gumbel, *Statistical theory of extreme values and some practical applications:a series of lectures*, 1954.

[58] C. J. Maddison, D. Tarlow, and T. Minka, “A* sampling,” in *Advances in Neural Information Processing Systems (NIPS)*, 2014, p. 1–10.

[59] E. Dupont, “Learning disentangled joint continuous and discrete representations,” in *Advances in Neural Inf. Proc. Systems (NeurIPS)*, 2018, pp. 708–718.

[60] 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.

[61] C. J. Maddison, A. Mnih, and Y. W. Teh, “The concrete distribution: A continuous relaxation of discrete random variables,” in *Proc. Int. Conf. on Learning Representations (ICLR), arXiv preprint arXiv:1611.00712*, 2016.

[62] J. He, D. Spokoyny, G. Neubig, and T. Berg-Kirkpatrick, “Lagging inference networks and posterior collapse in variational autoencoders,” in *Int. Conf. on Learning Representations (ICLR), arXiv preprint arXiv:1901.05534*, 2019.

[63] Y. Kim and A. M. Rush, “Sequence-level knowledge distillation,” in *Empirical Methods in Natural Language Processing (EMNLP), arXiv preprint arXiv:1606.07947*, 2016.

[64] W. Park, D. Kim, Y. Lu, and M. Cho, “Relational knowledge distillation,” in *Proc. of IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 3967–3976.

[65] M. Phuong and C. Lampert, “Towards understanding knowledge distillation,” in *Proc. Int. Conf. on Machine Learning (ICML), vol. PMLR 97*, 2019, pp. 5142–5151.

[66] C. P. Burgess, I. Higgins, A. Pal, L. Matthey, N. Watters, G. Desjardins, and A. Lerchner, “Understanding disentangling in β-vae,” in *Proc. NIPS Workshop on Learning Disentangled Representation, arXiv preprint arXiv:1804.03599*, 2017.

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

[68] 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) 2018, vol. PMLR 89*, 2019, pp. 1157–1166.

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

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

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

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

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

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

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

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

[77] A. Krizhevsky and G. Hinton, “Learning multiple layers of features from tiny images,” Univ. of Toronto, Tech. Rep., 2009.

[78] B. M. Lake, R. Salakhutdinov, and J. B. Tenenbaum, “Human-level concept learning through probabilistic program induction,” *Science*, vol. 350, no. 6266, pp. 1332–1338, 2015.

[79] F. Ye and A. G. Bors, “Learning joint latent representations based on information maximization,” *Information Sciences*, vol. 567, pp. 216–236, 2021.

[80] 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.

[81] 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.

[82] M. Aubry, D. Maturana, A. A. Efros, B. 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.

[83] 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.

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

[85] 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.

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

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 150 research papers, including 36 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.