Task-Free Continual Generative Modelling Via Dynamic Teacher-Student Framework

Fei Ye[a], Adrian. G. Bors[b]

[a]*School of Information and Software Engineering, University of Electronic Science and Technology of China, Chengdu, China, E-mail: feiye@uestc.edu.cn*
[b]*Department of Computer Science, University of York, York YO10 5GH, UK, E-mail: adrian.bors@york.ac.uk*

Abstract

Continually learning and acquiring new concepts from a dynamically changing environment is an important requirement for an artificial intelligence system. However, most existing deep learning methods fail to achieve this goal and suffer from significant performance degeneration under continual learning. We propose a new unsupervised continual learning framework combining Long- and Short-Term Memory management used for training deep learning generative models. The former memory system uses a dynamic expansion model (Teacher), while the latter uses a fixed-capacity memory buffer to store the update-to-date information. A novel Teacher model expansion approach, called the Knowledge Incremental Assimilation Mechanism (KIAM) is proposed. KIAM evaluates the probabilistic distance between the already accumulated information and that contained in the Short Term Memory (STM). The proposed KIAM adaptively expands the Teacher's capacity and promotes knowledge diversity among the Teacher's experts. As Teacher experts, we consider generative deep learning models such as : the Variational Autocencoder (VAE), the Generative Adversarial Network (GAN) or the Denoising Diffusion Probabilistic Model (DDPM). We also extend the KIAM-based model to a Teacher-Student framework in which we use a data-free Knowledge Distillation (KD) process to train a VAE-based Student without using any task information. The results on Task Free Continual Learning (TFCL) benchmarks show that the proposed approach outperforms other models.

Keywords: Continual Learning, Dynamic Expansion Model, Teacher-Student Network

1. Introduction

The continuous learning of new concepts and their latent representations from a dynamically evolving environment is a fundamental requirement for an artificial intelligence machine. Modern deep learning models are already outperforming humans in learning certain individual tasks [1], but would suffer dramatic performance degeneration when attempting to learn a sequence of different data domains. This is due to the fact that weights are rewritten when retraining the network, a phenomenon referred in AI as "catastrophic forgetting" [2].

A widely adopted strategy for mitigating catastrophic forgetting involves the Generative Replay Mechanism (GRM) [3], typically realized through architectures such as Variational Autoencoders (VAEs) [4] or Generative Adversarial Networks (GANs) [5]. The central premise of GRM-based methodologies is to synthesize pseudo-samples that encapsulate previously acquired knowledge, subsequently leveraging these generative replay instances to counteract forgetting during the acquisition of new tasks [6, 7, 8, 9, 10]. Although these approaches have demonstrated notable efficacy in continual classification and generative tasks, their performance tends to deteriorate as the number of sequentially learned tasks increases, primarily due to the constraints imposed by fixed model capacity, as theoretically substantiated in [11]. Furthermore, GRM-based techniques are susceptible to generating suboptimal replay samples, a consequence of phenomena such as mode collapse and unstable training dynamics [12], particularly when confronted with heterogeneous data domains. To address these limitations, one alternative is to substitute the GRM network with a memory buffer that retains a limited set of past learning exemplars [13]. During the assimilation of new tasks, samples from this buffer are replayed to facilitate model updates and alleviate forgetting. Nevertheless, memory-based strategies may introduce concerns regarding data privacy and security [14]. When faced with an unbounded sequence of tasks, both GRM- and memory-based approaches may prove inadequate in maintaining sufficient data for effective model training. Recent advancements have explored the integration of GRM with dynamic expansion mechanisms within a unified continual learning framework [11] to further mitigate forgetting. However, these hybrid methods still rely on the GRM, which ultimately leads to performance degradation as the number of tasks increases. Alternative research directions have investigated ensemble-based architectures [15, 16, 17], wherein each constituent network is constructed atop a shared backbone, enabling the preservation of optimal performance for prior tasks by maintaining frozen components while dynamically incorporating new modules for learning novel information [15]. Despite

their promise, most dynamic expansion models necessitate explicit access to task identities and related metadata to facilitate the expansion process, a requirement that is often impractical in real-world continual learning settings. In the Task-Free Continual Learning (TFCL) paradigm [18], such task-specific information and boundaries remain inaccessible during training.

Motivation. Existing continual learning research predominantly concentrates on classification tasks [19], whereas the application of generative modeling for images within continual learning frameworks remains underexplored. Generative modeling aims to learn the underlying data distributions and patterns to produce new samples that are statistically consistent with the target data distribution. Investigating image generative modeling in continual learning can enhance various downstream tasks, including real-time image editing systems, unsupervised feature representation learning, and online image compression. However, integrating generative models into task-free continual learning (TFCL) presents significant challenges such as catastrophic forgetting and distributional shifts. The absence of explicit task and class labels hampers the deployment of existing continual learning methodologies for this complex paradigm. Consequently, there is a pressing need to develop innovative frameworks capable of mitigating catastrophic forgetting and adapting to data distribution shifts within TFCL settings.

In this study, we investigate lifelong generative modeling within the Task-Free Continual Learning (TFCL) paradigm, with the objective of developing a generative architecture that consistently produces high-fidelity image reconstructions and learns robust, semantically meaningful latent representations over time without succumbing to catastrophic forgetting. The Dynamic Expansion Model (DEM) [20, 21] has demonstrated strong performance in TFCL scenarios and exhibits potential for deployment in unbounded data stream environments. DEM's core mechanism involves adaptively instantiating new expert modules upon detecting shifts in the underlying data distribution, while preserving the parameters of previously trained modules to retain historical knowledge. Unlike static architectures, DEM offers scalability and adaptability to non-stationary learning environments, thereby enhancing generalization capabilities. The criterion for model expansion in DEM is critical for maintaining an optimal trade-off between generalization and model complexity. Current dynamic expansion approaches typically leverage Neural Dirichlet Processes [20] and sample log-likelihood metrics [21] to determine when expansion is warranted. However, prior work has not considered the use of probabilistic distance measures between the statistical representations of historical and incoming data as an expansion signal, which limits the effectiveness of model adaptation. Additionally, these models often employ a multi-head

architecture, necessitating a component selection step during inference that incurs significant computational overhead. Moreover, the absence of a shared latent space in such frameworks restricts their applicability to downstream tasks such as cross-domain image reconstruction, interpolation, and disentanglement [22, 23].

Furthermore, we address the shortcomings of current Dynamic Expansion Mixture (DEM) methodologies. Existing DEM architectures tend to expand uncontrollably when learning large data streams, typically retaining the entire ensemble during inference, resulting in significant computational overheads. To overcome these challenges, we propose a scalable knowledge distillation framework tailored for continual learning in data streams characterized by evolving distributions, specifically within a Task-Free Continual Learning (TFCL) context. This framework is designed to ensure a lightweight model architecture during the inference phase. We leverage the teacher-student paradigm [10, 24, 25], which facilitates incremental knowledge accumulation and efficient compression of learned representations into a compact student network. An additional benefit of the teacher-student approach over traditional DEMs is the architectural flexibility of the student model, enabling its application across various paradigms, including unsupervised, supervised, disentangled, and causal representation learning [10, 25]. However, existing lifelong teacher-student frameworks [10, 24, 25] typically require explicit task information, which is often unavailable in real-world continual learning scenarios. To address this limitation, we introduce a dynamic expansion mixture model as the Teacher, complemented by a short-term memory buffer that retains recent data stream samples. The integration of the short-term memory buffer serves two primary objectives: (1) It provides enough samples for training one of the Teacher's experts, enabling it to learn appropriate knowledge from a data stream without accessing task information; (2) It aims to store more recently seen data samples, representing the short-term information, which can be combined with the knowledge preserved by the Teacher module, thus representing longer-term information, to provide a better knowledge distillation process for student learning.

A further challenge in implementing the teacher-student paradigm for Task-Free Continual Learning (TFCL) lies in the dynamic expansion criterion. Existing lifelong teacher-student frameworks [10] utilize the Knowledge Discrepancy Score (KDS) to regulate the Teacher's expansion, a process contingent on explicit task identities and boundaries. To address this limitation, we introduce the Knowledge Incremental Assimilation Mechanism (KIAM), empowering the Teacher module with autonomous network architecture expansion without relying on task-specific information. The core function of KIAM is to quantify the

divergence between the current short-term memory buffer (STM) and the cumulative knowledge of the Teacher, thereby signaling when to augment the Teacher's memorization capacity. The proposed mechanism facilitates the progressive accumulation of knowledge within the Teacher module, mitigating catastrophic forgetting and promoting knowledge diversity across all frozen experts in the Teacher ensemble. Additionally, we seek to optimize the Teacher model's footprint by introducing an expert pruning strategy that eliminates redundant or superfluous experts from the Teacher ensemble, resulting in a more compact network architecture with reduced computational overhead during knowledge distillation. All these knowledge transfer processes take place in an online, task-agnostic manner.

In summary, our contributions in this paper are as follows:

- We introduce an innovative teacher-student architecture designed for lifelong generative modeling within the Task Free Continual Learning (TFCL) scenario. To facilitate the assimilation of new information over time, we propose a novel Teacher expansion strategy, termed the Knowledge Incremental Assimilation Mechanism (KIAM), which dynamically augments the Teacher's knowledge base in the absence of supervised signals.

- We present an innovative data-free knowledge distillation (KD) framework that facilitates the online transfer of knowledge through data generation from a Teacher to a Student model, thereby reducing computational overhead during inference. Additionally, we propose a novel expert pruning strategy designed to compress the Teacher mixture model while preserving and enhancing knowledge diversity among the retained Teacher experts.

- We introduce an innovative theoretical framework to examine the forgetting dynamics of the proposed method within the context of TFCL. Our theoretical analysis indicates that the proposed framework delivers remarkable performance.

The rest of the paper is structured as follows. Section 2 describes the literature review for this paper. The proposed methodology, when using either GANs, VAEs or the DDPM as Teacher experts, is explained in Section 3. The theoretical analysis framework is introduced in Section 4. The experimental results are presented and discussed in Section 5, while the conclusion and future work are provided in Section 6.

2. Related Works

2.1. Continual Learning

Most existing approaches to continual learning consider the task label or some other information alleviating forgetting. These methods can be roughly divided into three approaches : memory-based approaches [6, 7, 26], regularisation-based methods [27, 28] and dynamic architectures [24, 29, 30]. The memory-based approaches either train a generative model [31] or employ a memory buffer [32] to preserve and replay past training samples. These memorized or generated samples are combined with the new samples for training the model. However, due to the fixed model and memory capacity, these methods do not perform well on a long sequence of tasks [11]. The regularization-based methods usually employ a regularization term in the objective function, aiming to minimize the change in the critical network weights when learning a new task [33]. Some regularization approaches use memory buffers to further improve the model's performance [34]. However, such methods require a significant computational effort when the number of tasks to be learned is large. Meanwhile, dynamic expansion models add new layers of processing nodes [35] or sub-networks into a mixture of networks [11, 17, 20, 21] when learning a new task. Past information can be preserved in the frozen network's parameters while only updating a new component when learning a new task. Dynamic expansion models achieve significant better performances than static network architectures [11] while they are scalable to learning a growing number of tasks [25]. Some studies have also explored CLIP-based approaches to improve the model's performance in continual learning [36, 37]. However, all these methods require accessing the task information and boundaries and thus cannot be applied in the TFCL framework.

2.2. Knowledge Distillation

Knowledge Distillation (KD) is a popular approach in deep learning research, first proposed in [28], which aims to transfer the knowledge learnt by a neural network, usually called Teacher, to another network, called Student. Most knowledge distillation methods employ a fixed and pre-trained Teacher model while a Student model (usually a classifier) is trained by accessing information provided by the Teacher and/or new data. Recent works propose implementing the Teacher module using an ensemble framework consisting of multiple networks, while the Student can learn multi-modality defined knowledge from the Teacher to improve its generalization performance [38, 39]. Knowledge distillation has recently been

used to relieve forgetting in continual learning. Li *et al.* [40] proposed a new approach, namely Learning without Forgetting (LwF), which employs Knowledge Distillation [28] to ensure that the prediction on each data sample is similar to the outputs of the previously learnt network. Zhai *et al.* [41] introduced a GAN-based learning framework for conditional image generation, called Lifelong GAN. Unlike existing GAN-based GRM, which mainly focuses on the classification task, Lifelong GAN can be used for classification and conditional image generation tasks. The main idea for the Lifelong GAN is to employ knowledge distillation that enforces the matching between the latent representations of the current task and the previously learnt task. However, this approach still requires to use the auxiliary data during the knowledge distillation process, which is intractable when learning an infinite number of tasks. Pietro *et al.* [42] introduced a simple and effective approach for continual learning using knowledge distillation called the Dark Experience Replay (DER). This approach employs a memory buffer to store samples for past tasks and then minimizes the distance on the network's output between the current task and memorized samples. DER was initially designed for general continual learning and can be extended for task-free continual learning by adapting reservoir sampling [43] that randomly selects samples from the data stream and then adds them into the memory buffer. Although the KD-based approaches achieve promising performances, they can not deal with infinite data streams due to their fixed model capacities.

2.3. Lifelong Generative Modeling

Unlike continual supervised learning, which mainly performs classification tasks, lifelong generative modelling aims to learn a generative model that can continually produce data generations and reconstructions without forgetting [7, 44]. Lifelong generative modelling, based on the Variational Autoencoder (VAE) [4], was firstly explored in [6], and was shown to be able to induce disentangled representations across different data domains without forgetting. Furthermore, VAE-based lifelong learning approaches were extended to a teacher-student framework [7]. More recently, lifelong generative modeling was implemented using Generative Adversarial Networks (GAN)-based frameworks [10, 45] that trains a GAN-based module as the Teacher and transfers its learnt knowledge to the Student implemented by a VAE-based model. In addition, combining the VAE and GAN into a unified framework has shown promising performance in lifelong generative modeling [23]. Such VAE-GAN hybrid frameworks were shown to implement many downstream tasks, including supervised, semi-supervised, unsupervised learning and disentangled representation learning. However, despite

achieving good performance in lifelong generative modeling, these GRM-based approaches are not scalable to learning long sequences of tasks due to the repeating generative replay processes [41] that eventually would affect the performance. This issue was addressed by employing a dynamic expansion GAN mixture model as the Teacher [25], which adds new Teacher experts into the system when learning new tasks. However, these frameworks still need task labels and information about when and where should perform the generative replay process or expand. On the other hand, the Dynamic Expansion Model (DEM) can be used in image generative modeling under task-free continual learning by developing new model expansion mechanisms. The first TFCL method was the Continual Unsupervised Representation Learning (CURL) [21]. CURL requires a threshold to control the expansion process of a mixture architecture. A similar idea is employed in the Continual Neural Dirichlet Process Mixture (CN-DPM) [20], where a Neural Dirichlet process-based expansion mechanism is used to dynamically increase the model's capacity. CN-DPM, unlike CURL, updates only a single VAE component at each training time while freezing the weights of all previously learnt components, thus preserving the best information from the past data. However, these approaches would not lead to optimal network architectures since they ignore the discrepancy and knowledge diversity among components/experts when performing model expansion.

3. Methodology

3.1. Problem Definition

For a given i-th data domain, we have the training dataset $D_S^i = \{\mathbf{x}_S^j\}_{j=1}^{N_i^S}$ and the testing dataset $D_T^i = \{\mathbf{x}_T^j\}_{j=1}^{N_i^T}$, respectively, where N_i^S and N_i^T denote the number of samples for the training and testing sets, respectively. In this paper, we study a more challenging learning setting, aiming to train a model on a data stream consisting of multiple data distributions provided in an online manner. We construct a data stream $\mathcal{S}$ by including all incoming training sets in a sequence manner, expressed as :

$$\mathcal{S} = \{D_S^1 \cup D_S^2 \cdots \cup D_S^k\}\,, \tag{1}$$

where k is the total number of datasets to be learnt. In addition to the domain-incremental setting, defined above in Eq. (1), we also consider a more challenging setting involving class and domain shifts. We divide each dataset D_S^1 into five parts $\{D_S^{1,1}, \cdots, D_S^{1,5}\}$, according to the category information [18], with each part

containing samples from two classes [18]. This setting is more challenging than that from Eq. (1) since the model requires dealing with the change in both domain and class over time, resulting in the following sequence :

$$\mathcal{S} = \{D_S^{1,1} \cup D_S^{1,2} \cup \cdots \cup D_S^{1,5} \cdots D_S^{k,5}\}\,. \tag{2}$$

Let $\{\mathcal{T}_1', \cdots, \mathcal{T}_{N'}'\}$ be the total number of training time/steps for the data stream $\mathcal{S}$. At a training step/time $\mathcal{T}_i'$, we can only see a small batch of samples $\{\mathbf{x}_{i,j}\}_{j=1}^{b}$ provided from $\mathcal{S}$ while all previous data batches $\{\{\mathbf{x}_{1,j}\}_j^b, \cdots, \{\mathbf{x}_{i-1,j}\}_j^b\}$ are unavailable. We assume that learning the whole data stream $\mathcal{S}$ requires a total of N' training steps. Once the model finishes the training step/time $\mathcal{T}_{N'}'$, we evaluate its performance on all test sets $\{D_T^1, \cdots, D_T^k\}$. Compared with the problem definition described in [10, 25], which can access the task information during the training, TFCL represents a more challenging learning scenario where task identities and information are unavailable and the model can only see a small data batch once at each training time/step.

3.2. The Knowledge Incremental Assimilation Mechanism (KIAM)

Humans can incrementally learn and memorise novel concepts throughout their entire lifespan [46]. Specifically, biological research [47] shows that short- and long-term information/skills can be learned using distinct brain networks at different stages. Inspired by this, we propose using two memory systems storing short- and long-term information to address catastrophic forgetting in Task Free Continual Learning (TFCL). Firstly, we introduce a short-term memory buffer to store the more recently seen data samples from the data stream $\mathcal{S}$ during the training, aiming to shortly preserve the up-to-date information. Then we introduce a long-term memory system to preserve the permanent information during training. Using a memory buffer for storing long-term crucial information may lead to significant storage requirements if the data stream $\mathcal{S}$ involves a large number of underlying data distributions. Inspired by the dynamic Teacher-Student framework from [25], which enables a Teacher module to capture multiple data distributions without forgetting, we propose to implement a long-term memory system employing a dynamic expansion model as a Teacher. Such an approach has several advantages when compared to using a memory buffer : (1) The dynamic expansion model can generate an unlimited number of samples while the memory buffer can only store and replay a limited data set; (2) The dynamic expansion model is scalable to the data stream $\mathcal{S}$ that consists of multiple data domains; (3) Storing long-term information using a memory buffer requires designing an appropriate sample selection approach to selectively filter out many unnecessary samples.

However, such an approach would lose some previous important data samples, resulting in performance degeneration. In contrast, the dynamic expansion model can permanently preserve past information by freezing all previously trained components and learning novel knowledge through an appropriate dynamic expansion mechanism.

Another challenge for the Teacher module is its dynamic expansion process, which is hard to control under the TFCL since we can not access the task information during training. In this paper, we address this issue by introducing the Knowledge Incremental Assimilation Mechanism (KIAM), which aims to reduce the Teacher's parameters without decreasing its performance too much. Let $\mathcal{M}_i$ denote a short-term memory of fixed size, where the subscript i denotes that $\mathcal{M}_i$ was updated at $\mathcal{T}_i$. Let $\mathbf{A}_i = \{\mathcal{A}_1, \cdots, \mathcal{A}_c\}$ denote a Teacher module aiming to learn c experts at $\mathcal{T}_i$. Each expert $\mathcal{A}_j$ can be implemented by using either a GAN [5], or a VAE [4], or a diffusion model [48] to learn a generator distribution P_{θ_i} with trainable parameters θ_i. Without the task information and labels, detecting whether the data distribution changes at a certain training time/step is a major challenge. In order to address this challenge, we propose a new expansion criterion that evaluates the probabilistic distance between the knowledge learnt by each previous Teacher expert and the information from the short-term memory and uses this measure as an expansion signal:

$$\min\left\{ f_p(D'_j, \mathcal{M}_i) \,|\, j = 1, \cdots, c-1 \right\} \geq \nu \;, \tag{3}$$

where D'_j is a dataset consisting of samples drawn from P_{θ_j}. $f_p(\cdot,\cdot)$ is a distance measure evaluating the similarity between two datasets, and in practice, we consider sampling from $\mathcal{M}_i$ and calculate $f_p(D'_j, \mathcal{M}_i)$ in Eq. (3).. We omit the current expert $\mathcal{A}_c$ in Eq. (3) since $\mathcal{A}_c$ is knowledgeable about $\mathcal{M}_i$. One potential approach for implementing $f_p(\cdot,\cdot)$ is training a discriminator using adversarial learning [5], to estimate the discrepancy between the current memory buffer and the already-learnt knowledge. However, such an approach may require considerable computational costs, especially when the total number of training steps/times is large. In addition, adversarial learning suffers from unstable training behaviour and mode collapse [12], leading to unstable signals for the model expansion in Eq. (3). Consequently, we implement $f_p(\cdot,\cdot)$ by using a statistical measure which is easy to compute such as the Frechét Inception Distance (FID) score. FID is a non-parametric measure which is evaluated between non-parametric data distributions, extensively used to evaluate the performance of GAN models [49]. In this study we employ the FID to evaluate the knowledge similarity between each previously trained expert and the current memory buffer (STM). Compared to

other dynamic expansion models, which employ the log-likelihood [21] or Neural Dirichlet processes [20] to implement the expansion criterion, Eq. (3) relies on a non-parametric distance measure which can better distinguish new data from the already-learnt knowledge.

If the dynamic expansion criterion from Eq. (3) is satisfied at $\mathcal{T}_i'$, we freeze the current expert $\mathcal{A}_c$ to preserve the information stored by the STM $\mathcal{M}_i$, representing the knowledge learnt by $\mathcal{A}_c$. We also construct a new expert $\mathcal{A}_{c+1}$ to be added to the Teacher module at the next training step $\mathcal{T}_{i+1}'$, in order to implement the subsequent learning processes. After being used for training we remove all data samples from the current STM memory $\mathcal{M}_i$ at $\mathcal{T}_{i+1}'$, avoiding the storage of similar data samples for the future learning of $\mathcal{A}_{c+1}$. Compared to the expansion process described in [25], Eq. (3) does not require accessing any task information and thus enables the model for TFCL. In addition, dynamic expansion methods, such as the Continual Neural Dirichlet Process Mixture (CN-DPM)) [20] also employs a short-term memory buffer to store recent data samples, but the proposed KIAM has several differences from CN-DPM: (1) The CN-DPM does not learn a Student module while the proposed KIAM implements the dynamic expansion model as a Teacher module within a unified framework to eventually support Student's learning; (2) The CN-DPM employs the Neural Dirichlet process for the model expansion, while the proposed KIAM uses the FID-based criterion; (3) The CN-DPM implements each expert using a VAE model, resulting in rather blurred image generation results. In contrast, the proposed KIAM can employ a more powerful generative model, such as a diffusion model or a GAN to implement each expert, providing better generative knowledge for Student's learning.

3.3. Continual Generative Knowledge Distillation

Most existing Knowledge Distillation (KD) methods transfer class and category information from a large and complex Teacher module to a small-scale Student, which is mainly used in classification tasks [50]. However, these approaches fail to transfer unsupervised generative factors from the Teacher to the Student due to lacking class label information. In addition, these methods usually require accessing the entire training dataset and thus can not be used in Task-Free Continual Learning (TFCL) [51]. Next, we propose a new KD approach for continual generative modelling without accessing any of the previous data. First, let us consider implementing the Student using a latent variable generative model $p_{\theta_{stu}}(\mathbf{x}, \mathbf{z}) = p_{\theta_{stu}}(\mathbf{x} \,|\, \mathbf{z})p(\mathbf{z})$, where $\mathbf{z}$ and $\mathbf{x}$ denote the latent and observed variables, respectively. $p_{\theta_{stu}}(\mathbf{x} \,|\, \mathbf{z})$ is the decoder, and $p(\mathbf{z}) = \mathcal{N}(\mathbf{0}, \mathbf{I})$ is the prior distribution. To transfer the knowledge stored by the Teacher to the Student, one

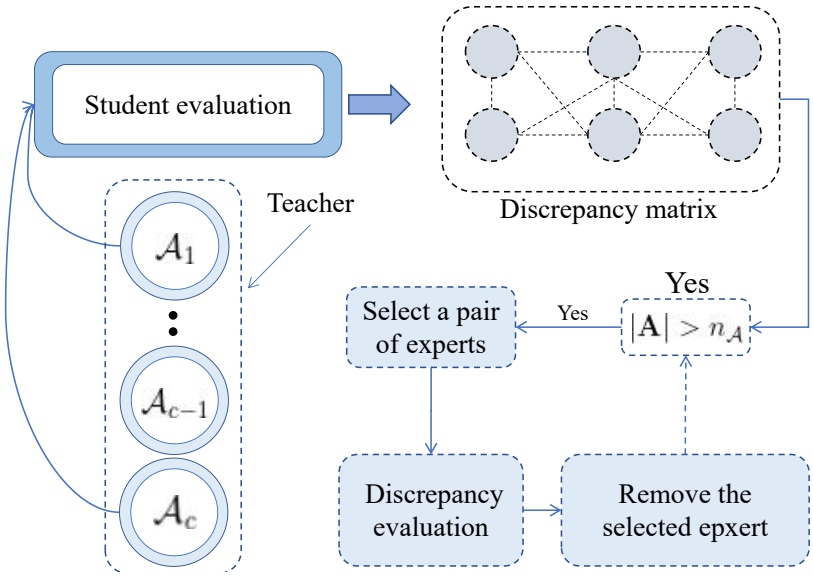


Figure 1: The procedure for removing redundant experts.

can minimise the KL divergence between the generator distribution P_{θ_j} of each Teacher expert $\mathcal{A}_j$ and the generator distribution $\mathrm{P}_{\theta_{stu}}$ of the Student. However, this KL divergence optimization is computationally intractable due to the lack of explicit density function for the distribution $\mathbb{P}_{\theta_j}$. To solve this problem, we minimize the cross entropy between $\mathbb{P}_{\theta_j}$ and $p_{\theta_{stu}}(\mathbf{x})$ to replace the KL divergence :

$$\mathcal{L}_{KD} = \sum\nolimits_{j=1}^{c-1} \left\{ -\mathbb{E}_{\mathbf{x}\sim\mathbb{P}_{\theta_j}}[\log p_{\theta_{stu}}(\mathbf{x})] \right\}. \tag{4}$$

However, directly evaluating Eq. (4) is intractable since the sample log-likelihood function $\log p_{\theta_{stu}}(\mathbf{x}) = \log \int p_{\theta_{stu}}(\mathbf{x} \mid \mathbf{z}) p(\mathbf{z}) \mathrm{d}\mathbf{z}$ is hard to estimate. In this study, we propose employing a variational distribution $q_\phi(\mathbf{z} \mid \mathbf{x})$, parameterised by ϕ, to approximate the true posterior $p_{\theta_{stu}}(\mathbf{z} \mid \mathbf{x})$. As a result, the sample log-likelihood $\log p_{\theta_{stu}}(\mathbf{x})$ can be estimated using the Evidence Lower Bound (ELBO) [4]. Then, Eq (4) can be calculated as :

$$\begin{aligned} \mathcal{L}_{KD} = \sum\nolimits_{j=1}^{c-1} \big\{ & \mathbb{E}_{\mathbf{x}\sim\mathbb{P}_{\theta_j}}[-\mathbb{E}_{q_{\varsigma_{stu}}(\mathbf{z}\mid\mathbf{x})}[\log p_{\theta_{stu}}(\mathbf{x} \mid \mathbf{z})] \\ & + D_{KL}[q_{\varsigma_{stu}}(\mathbf{z} \mid \mathbf{x}) \mid\mid p(\mathbf{z})]] \big\}. \end{aligned} \tag{5}$$

By combining the KL loss from Eq. (5) and the loss on the current memory buffer

$\mathcal{M}_i$, we propose a unified objective function for learning the Student at $\mathcal{T}_i$:

$$\begin{aligned}\mathcal{L}_{Stu} = \mathbb{E}_{\mathbf{x}\sim\mathbb{P}_{\mathcal{M}_i}}[-\mathbb{E}_{q_{\varsigma_{stu}}(\mathbf{z}\,|\,\mathbf{x})}\left[\log p_{\theta_{stu}}\left(\mathbf{x}\,|\,\mathbf{z}\right)\right] \\ + D_{KL}\left[q_{\varsigma_{stu}}\left(\mathbf{z}\,|\,\mathbf{x}\right)\,||\,p\left(\mathbf{z}\right)\right]] + \mathcal{L}_{KD}\,,\end{aligned} \tag{6}$$

where $\mathbb{P}_{\mathcal{M}_i}$ is the distribution of the memory buffer $\mathcal{M}_i$ at $\mathcal{T}_i$. The first term encourages the Student module to learn samples obtained from the memory buffer $\mathcal{M}_i$ that preserves the short-term information, and the second term $\mathcal{L}_{KD}$ transfers the Teacher's knowledge, representing the long-term information, to the Student. We train the Student module using Eq. (6) in the mini-batch learning manner [1], which has been widely used in deep learning [52]. Compared to the D-TS framework described in [25], the proposed KD approach, defined by Eq. (5), has several differences: (1) Eq. (5) is used for transferring the knowledge at each training step under TFCL, while D-TS can only access data corresponding to the current task learning; (2) Eq. (5) employs a memory buffer $\mathcal{M}_i$ to store more recent samples, which enables the Student module to learn both short- and long-term information during the training. In contrast, the D-TS framework does not have a memory buffer and its learning relies heavily on the task identity; (3) The Teacher module in the D-TS framework always increases its model size/capacity when learning a growing number of tasks. In contrast, the proposed framework introduces an expert pruning approach, which is introduced in the next section, that can maintain a fixed Teacher's model size without sacrificing performance.

3.4. The Expert Pruning Approach

When deploying the model on a resource-constrained device, such as training robots or drones among other platforms, we have to consider the memory and computational resources limitations. Consequently, we can not expand the Teacher module forever and should keep a compact network architecture for the Teacher. To address this issue, we introduce a novel expert pruning approach to automatically remove Teacher expert components over time. The primary idea of the proposed expert pruning approach is to find and remove those Teacher experts that share significant amounts of knowledge with each other. However, directly searching multiple similar Teacher experts is computationally intractable. Instead, we formulate the selection process as a simple problem that first finds a pair of Teacher experts containing overlapping knowledge and then one of them is removed. Then, the expert pruning process is repeated until a specific criterion is met.

Let us consider a Teacher module with c experts $\mathbf{A} = \{\mathcal{A}_1, \cdots, \mathcal{A}_c\}$, updated at the training step $\mathcal{T}_i$. We define a knowledge matrix $\mathbf{Q} \in \mathbf{R}^{c\times c}$, where each

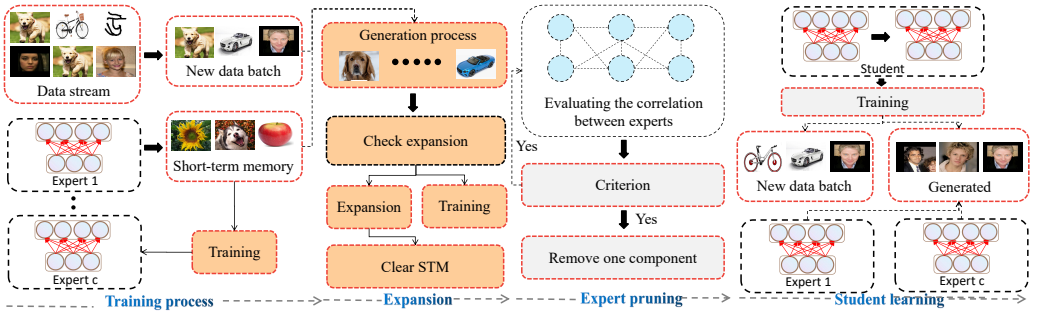


Figure 2: The learning procedure of the proposed framework where we omit the updating of the memory for the sake of simplification. In the Teacher learning process, we always update the current Teacher expert on the memory buffer $\mathcal{M}_i$ at $\mathcal{T}_i$. In the dynamic expansion process, if the criterion from Eq. (3) is satisfied, we then build a new Teacher expert while freezing the current one. In the expert pruning process, we remove those Teacher experts having overlapping knowledge with other experts until the number of Teacher experts is not larger than $n_{\mathcal{A}}$. In the knowledge distillation process, we update the Student module using Eq. (6). The pseudo-code corresponding to the implementation is provided in Section 3.7.

$\mathbf{Q}(a, g)$ records the knowledge discrepancy score between experts $\mathcal{A}_g$ and $\mathcal{A}_a$. Given that the Student module is always updated to capture the entire information from the Teacher and the memory buffer, we can employ the Student module to identify whether two Teacher experts contain statistically overlapping knowledge. The knowledge discrepancy score $\mathbf{Q}(a, g)$ is estimated using the square loss $\|\cdot\|^2$ on the latent variables, inferred by the inference model $q_{\varsigma_{stu}}(\mathbf{z} \,|\, \mathbf{x})$ of the Student :

$$\mathcal{L}_{ks}(\mathcal{A}_a, \mathcal{A}_g) = \frac{1}{m'} \sum_{j=1}^{m'} \|\mathbf{z}_{a,j} - \mathbf{z}_{g,j}\|^2 , \tag{7}$$

where $\mathbf{z}_{a,j}$ and $\mathbf{z}_{g,j}$ are latent variables, drawn from $q_{\varsigma_{stu}}(\mathbf{z} \,|\, \mathbf{x})$ that receives the data samples $\mathbf{x}_{a,j}$ and $\mathbf{x}_{g,j}$ generated by $\mathcal{A}_a$ and $\mathcal{A}_g$, respectively. In the experimental results we consider $m' = 1000$ as the total number of generated samples. Eq. (7) is computationally efficient since it is evaluated on the low-dimensional latent space. Although the FID can be used to evaluate the distance on a pair of Teacher experts, it still has much more additional computational requirements than that used for Eq. (7). If the two experts, $\mathcal{A}_a$ and $\mathcal{A}_b$ have learnt most similar knowledge, the latent variables $\mathbf{z}_a$ and $\mathbf{z}_b$, representing the latent information of $\mathcal{A}_a$ and $\mathcal{A}_b$ tend to be similar, resulting in a small $\mathcal{L}_{ks}(\mathcal{A}_a, \mathcal{A}_g)$ in Eq. (7). We use the square loss to evaluate the distance between latent variables due to its computational efficiency, while other choices are discussed in Section 6. Once the discrepancy matrix $\mathbf{Q}$ is evaluated, we can use it to determine a pair of experts

with the minimal discrepancy score :

$$\{a^\star, g^\star\} = \arg\min_{a,g=1,\cdots,c,a\neq g} \big\{\mathbf{Q}(a,g)\big\}\,, \tag{8}$$

where $A^\star$ and $G^\star$ are the indices of the chosen experts. To decide which selected expert should be removed, we calculate the discrepancy score between each other expert from the Teacher's ensemble and either $\mathcal{A}_{a^\star}$ or $\mathcal{A}_{g^\star}$:

$$\begin{aligned} s_{a^\star} &= \sum\nolimits_{j=1,j\neq a^\star}^{c} \{\mathbf{Q}(a^\star, j)\}\,, \\ s_{g^\star} &= \sum\nolimits_{j=1,j\neq g^\star}^{c} \{\mathbf{Q}(g^\star, j)\}\,. \end{aligned} \tag{9}$$

A higher diversity score, calculated using Eq. (9), indicates that the selected component represents distinct knowledge from the other components and should be kept. The primary goal for Eq. (9) is to remove a component that has a small discrepancy score with respect to the remaining experts. This mechanism can maintain the discrepancy and knowledge diversity between Teacher experts, benefiting from capturing more information by using a compact network architecture. If $s_{a^\star} > s_{g^\star}$, then we decide to remove $\mathcal{A}_{g^\star}$, while otherwise we remove $\mathcal{A}_{a^\star}$. We then continually evaluate (8) and (9) to identify and remove all other unnecessary experts from the Teacher' knowledge matrix representation $\mathbf{Q}$. This redundant expert removal process is finished when the total number of experts becomes equal to a predefined number $n_{\mathcal{A}}$. This process is illustrated in Fig. 1.

We also can use a threshold for evaluating the importance of the experts to be preserved for the teacher ensemble. Once a matrix $\mathbf{Q}$ is evaluated in Eq. (7), we identify a pair of information overlapping experts by searching the minimal discrepancy score using Eq. (8). We then consider a threshold λ_2 to determine whether we remove one of the paired experts as follows:

$$\min\Big\{\sum\nolimits_{j=1,j\neq a^\star}^{c} \mathbf{Q}(a^\star, j), \sum\nolimits_{j=1,j\neq g^\star}^{c} \mathbf{Q}(g^\star, j)\Big\} < \lambda_2\,. \tag{10}$$

This approach does not restrict the total number of Teacher experts and thus is suitable for learning infinite data streams. We refer to this approach as KIAM-GAN-λ_2, and KIAM-VAE-λ_2 when the Teacher uses GANs or VAEs, respectively, as experts.

3.5. Disentangled Representation Learning

Disentangled representation learning is an important topic in computer vision [53], which aims to find the latent variables which indicate specific directions of

variation for certain image characteristics. Learning disentangled representations is usually implemented using the β-VAE [54], which employs an additional hyperparameter $\beta > 1$ to penalise the KL divergence term in the main objective function. However, the β-VAE suffers from poor image reconstruction results when using a large value for the hyperparameter β. Then, a new approach was proposed in [55] to progressively increase the information capacity of the latent variable in the β-VAE, during training. Currently, disentangled representation methods only consider learning a static and predefined data distribution, while the lifelong learning of disentangled representations is a rather new research topic [56]. For the student's disentangled representation knowledge distillation, we consider the improved β-VAE method [55], by extending Eq. (5) as :

$$\begin{aligned}\mathcal{L}_{KD}^{Dis} = \sum\nolimits_{j=1}^{c-1} \big\{ \mathbb{E}_{\mathbf{x}\sim\mathbb{P}_{\theta_j}}[-\mathbb{E}_{q_{\zeta_{stu}}(\mathbf{z}\,|\,\mathbf{x})}\left[\log p_{\theta_{stu}}\left(\mathbf{x}\,|\,\mathbf{z}\right)\right] \\ + \gamma|D_{KL}\left[q_{\zeta_{stu}}\left(\mathbf{z}\,|\,\mathbf{x}\right)\,||\,p\left(\mathbf{z}\right)\right] - C|\big\}\,,\end{aligned} \tag{11}$$

where C and γ are hyperparameters used to control the degree of disentanglement. In the experiments, we consider the recommended hyperparameters from [55], where the multiplicative parameter is $\gamma = 4$, while the hyperparameter C is chosen within the interval [0.5,25.0] during training. The final objective function for encouraging the Student to learn disentangled representations is defined as :

$$\begin{aligned}\mathcal{L}_{Stu}^{Dis} = \mathbb{E}_{\mathbf{x}\sim\mathbb{P}_{\mathcal{M}_i}}\{-\mathbb{E}_{q_{\zeta_{stu}}(\mathbf{z}\,|\,\mathbf{x})}\left[\log p_{\theta_{stu}}\left(\mathbf{x}\,|\,\mathbf{z}\right)\right] \\ + \gamma|D_{KL}\left[q_{\zeta_{stu}}\left(\mathbf{z}\,|\,\mathbf{x}\right)\,||\,p\left(\mathbf{z}\right)\right] - C|\} + \mathcal{L}_{KD}^{Dis}\,,\end{aligned} \tag{12}$$

where $\mathcal{L}_{KD}^{Dis}$ is the dissentangled representation knowledge distillation loss provided in Eq. (11).

3.6. Generative models as Teachers

In the proposed approach, the Teacher module dynamically builds several components to capture the knowledge from a data stream during the training. Only the newly created component is updated while all others are frozen in order to preserve the information learnt in the past. This allows the Teacher module to gradually store new knowledge without forgetting previously learnt information. Unlike in the Lifelong Teacher-Student (LTS) framework, described in [25], where the task information is known, thus allowing the Teacher module to reuse one of its existing components to learn a new task, the proposed Knowledge Incremental Assimilation Mechanism (KIAM) does not use any labels.

Traditionally, Variational Autoencoders (VAEs) [4] or Generative Adversarial Networks (GANs) [5] have been used as generative models. Both VAEs [25] and GANs [57] have also been successfully employed as Generative Replay Mechanisms in Teacher-Student configurations. More recently, diffusion models [48] have been successfully used as high-quality image generators. In the following we discuss all these models as implementations for the Teacher.

GAN-based Teacher. For the GAN-based Teacher model, each expert $\mathcal{A}_c = \{\mathcal{D}_\eta, \mathcal{G}_{\theta_c}\}$ has a discriminator network $\mathcal{D}_\eta$ parameterized by η and a generator $\mathcal{G}_{\theta_c}$ parameterized by θ_c. We consider the WGAN-GP loss [58] for training the current c-th expert $\mathcal{A}_c$ at $\mathcal{T}_i$:

$$\mathcal{L}_{\mathcal{G}_{\theta_c}} = \frac{1}{m}\sum_{t=1}^{m}\Big\{ -\mathcal{D}_\eta(\mathcal{G}_{\theta_c}(\mathbf{z}_t))\Big\}, \tag{13}$$

$$\mathcal{L}_{\mathcal{D}_\eta} = \frac{1}{m}\sum_{t=1}^{m}\Big\{[\mathcal{D}_\eta(\mathcal{G}_{\theta_c}(\mathbf{z}_t)] - \mathcal{D}_\eta(\mathbf{x}_t) + \gamma(\|\nabla_{\tilde{\mathbf{x}}_t}\mathcal{D}_\eta(\tilde{\mathbf{x}}_t)\|_2 - 1)^2\Big\}, \tag{14}$$

where in the experimental study we consider a batch size of $m = 64$, and $\mathbf{z}_t \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ is a random vector. $\mathbf{x}_t$ is the t-th real sample from the data batch $\mathbf{X}_{batch}$ obtained from the memory buffer $\mathcal{M}_i$. γ is a hyperparameter, aiming to regulate the last term from the right hand side of Eq. (14) that enforces the discriminator's Lipschitz constraint, [58]. $\tilde{\mathbf{x}}_t$ is the interpolated image produced by $\tilde{\mathbf{x}}_t = s\mathbf{x}_t + (1-a)\mathbf{x}'_t$, where s is drawn from a uniform distribution $U(0,1)$ and $\mathbf{x}'_i$ is a generated image. Actually, we only expand the number of GAN generators, which are frozen after being learnt one-by-one, while keeping a single discriminator, thus reducing the overall number of parameters. The basic GAN-based teacher model integrated with our proposed Knowledge Incremental Assimilation Mechanism (KIAM) is named as KIAM-GAN, while when considering the expert pruning it becomes KIAM*-GAN.

VAE-based Teacher. For the VAE-based Teacher model, each expert $\mathcal{A}_c$ is implemented as a VAE model consisting of a decoder $p_{\theta_c}(\mathbf{x}\,|\,\mathbf{z})$ and an encoder, defined by an encoding distribution $q_\varsigma(\mathbf{z}\,|\,\mathbf{x}) = \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\sigma}^2\mathbf{I})$, whose hyperparameters $\{\boldsymbol{\mu}, \boldsymbol{\sigma}\}$ are given by the encoder. A latent variable $\mathbf{z} = \boldsymbol{\mu} + \boldsymbol{\sigma} \odot \boldsymbol{\tau}$ is then drawn from $\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\sigma}^2\mathbf{I})$ using the reparameterization trick [4], where $\boldsymbol{\tau} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$, and $\odot$ is the element-wise product. We employ the VAE loss function for training the current expert $\mathcal{A}_c$, expressed as :

$$\mathcal{L}_{VAE}(\mathbf{x}; \mathcal{A}_c) = -\mathbb{E}_{q_\varsigma(\mathbf{z}\,|\,\mathbf{x})}\left[\log p_{\theta_c}\left(\mathbf{x}\,|\,\mathbf{z}\right)\right] + D_{KL}\left[q_\varsigma\left(\mathbf{z}\,|\,\mathbf{x}\right) \,||\, p\left(\mathbf{z}\right)\right]. \tag{15}$$

Similar to the GAN-based experts, only a single encoding distribution $q_\varsigma(\mathbf{z} \mid \mathbf{x})$ is needed in the whole training phase because all previously frozen experts of the decoding modules for $\{\mathcal{A}_1, \cdots, \mathcal{A}_{c-1}\}$ do not have to be updated during subsequent learning. The basic VAE-based teacher model is named as KIAM-VAE, while KIAM*-VAE involves also expert pruning.

Diffusion-based Teacher. The Denoising Diffusion Probabilistic Model (DDPM) [48] has been successfully applied in many applications, including image super-resolution [59, 60], image synthesis [61, 62, 63, 64, 65], inpainting [66], graph generation [67], text-to-image generation [68, 69] and shape generation [70]. When compared to the GAN-based models, which suffer from unstable training [12, 71], the DDPM model has a stable training procedure resulting in high-quality generation results when compared to the VAE-based models.

DDPM consists of forward and backward diffusion processes. The goal of the former is to corrupt the given data by gradually adding Gaussian noise that is usually drawn from an empirical data distribution $p(\mathbf{x}^0)$, to a real sample $\mathbf{x}^0$, leading to the generation of a sequence of noisy samples $\{\mathbf{x}^1, \cdots, \mathbf{x}^T\}$, where T is the total number of steps being considered. We formulate the forward process as follows, [48] :

$$q(\mathbf{x}^{1:T} \mid \mathbf{x}^0) := \prod_{t=1}^{T} q(\mathbf{x}^t \mid \mathbf{x}^{t-1}), \tag{16}$$

where $\mathbf{x}^{1:T}$ denotes $\{\mathbf{x}^1, \cdots, \mathbf{x}^T\}$. The DDPM framework introduces a variance schedule defined by the parameters $\{\beta_t \in (0,1) \mid t = 1, \cdots, T\}$ in order to ensure a smooth transfer between $\mathbf{x}^{t-1}$ and $\mathbf{x}^t$, expressed as :

$$q(\mathbf{x}^t \mid \mathbf{x}^{t-1}) := \mathcal{N}(\mathbf{x}_t; \sqrt{1-\beta_t}\mathbf{x}^{t-1}, \beta_t \mathbf{I}). \tag{17}$$

We set each variance controlling parameter β_t as being small enough for ensuring that the reverse distribution $q(\mathbf{x}^{t-1} \mid \mathbf{x}^t)$ is a Gaussian distribution, [72, 73]. In practice, estimating $q(\mathbf{x}^{t-1} \mid \mathbf{x}^t)$ is intractable because we need to access the whole training dataset. The DDPM framework solves this issue by learning a model $p_\theta(\mathbf{x}^{t-1} \mid \mathbf{x}^t)$ with trainable parameters θ, and with the backward diffusion process redefined as :

$$p_\theta(\mathbf{x}^{0:T}) := p(\mathbf{x}_T) \prod_{t=1}^{T} p_\theta(\mathbf{x}^{t-1} \mid \mathbf{x}^t), \tag{18}$$

where $p_\theta(\mathbf{x}^{t-1} \mid \mathbf{x}^t)$ is defined as:

$$p_\theta(\mathbf{x}^{t-1} \mid \mathbf{x}^t) := \mathcal{N}(\mathbf{x}^{t-1}; \boldsymbol{\mu}_\theta(\mathbf{x}^t, t), \boldsymbol{\Sigma}_\theta(\mathbf{x}^t, t)), \tag{19}$$

where $\boldsymbol{\Sigma}_\theta(\cdot,\cdot)$ and $\boldsymbol{\mu}_\theta(\cdot,\cdot)$ are trainable functions implemented by a neural network. The DDPM framework employs a simple but effective training objective

function, [48] :

$$\mathcal{L}_{\text{DDPM}}(\theta) := \mathbb{E}_{t,\mathbf{x}^0,\boldsymbol{\epsilon}}\left[\|\boldsymbol{\epsilon} - \boldsymbol{\epsilon}_\theta(\sqrt{\hat{\alpha}_t}\mathbf{x}^0 + \sqrt{1-\hat{\alpha}_t}\boldsymbol{\epsilon}, t)\|^2\right], \tag{20}$$

where $\hat{\alpha}_t := \prod_{s=1}^{t}\{\alpha_s\}$ and $\alpha_t := 1 - \beta_t$. $\boldsymbol{\epsilon}$ is a random vector drawn from $\mathcal{N}(\mathbf{0}, \mathbf{I})$. $\boldsymbol{\epsilon}_\theta(\cdot,\cdot)$ is implemented by a noise estimator model that predicts $\boldsymbol{\epsilon}$ from $\mathbf{x}^t$ and t.

A natural approach to enhance the quality of knowledge transfer from the teacher to student module is to use the DDPM $\boldsymbol{\epsilon}_{\theta_i}(\cdot,\cdot)$ for implementing each teacher expert, where the subscript i represents the expert index. However, the DDPM requires performing thousands of iterations during the generation process of a single image, leading to considerable computational costs. Consequently, using the dynamic expansion criterion, defined in Eq. (3), to regulate the expansion process of the teacher module is intractable due to frequently employing generative processes. We address this issue by proposing a new dynamic expansion criterion that does not involve the generation process by assigning a tiny memory buffer M^i for each diffusion-based teacher expert $\boldsymbol{\epsilon}_{\theta_i}(\cdot,\cdot)$, aiming to preserve a few data samples, characteristic of the probabilistic representation of the diffusion component. The tiny memory buffer M^i randomly stores data samples obtained from the main incoming data buffer $\mathcal{M}_i$ and is fixed when the i-th teacher expert $\boldsymbol{\epsilon}_{\theta_i}(\cdot,\cdot)$ is frozen. Based on the tiny memory buffer, we introduce a new dynamic expansion criterion for the diffusion-based teacher module :

$$\min\left\{f_p(M^j, \mathcal{M}_i) \,|\, j = 1, \cdots, c-1\right\} \geq \nu_1 \,. \tag{21}$$

Eq. (21), unlike the criterion defined in Eq. (3), only compares the knowledge representation distance between two memory buffers, one storing the incoming data and the other storing the data associated with each diffusion-based expert. This approach is computationally efficient because it does not involve the image generation process. We call the proposed framework using the diffusion-based teacher module as KIAM-DDPM.

We summarise several differences in the Teacher module between the approach for the proposed KIAM and the Dynamic Teacher Student (D-TS) [25] : (1) The teacher module in D-TS can select an existing component to learn a new task, which requires accessing the task information. The proposed KIAM always updates the current expert to learn new knowledge while freezing all previously learnt experts, which can be used in TFCL; (2) The Teacher model size in D-TS grows as the number of tasks increases. In contrast, we can fix the Teacher's

Algorithm 1: Training algorithm of KIAM*-GAN

```
for i < t do
    Updating of the memory;
    if (|M_i| ≥ |M_i|_max) then
        Remove earliest samples from M_i;
    end
    Teacher learning; ;
    if (i == 100) then
        Build a new expert A_2 while fixing A_1 ;
        else
            for s < iterations do
                {x_{s,j}}_{j=1}^b obtained from M_i ;
                Update the generator of the current expert using Eq. (13);
                Update the discriminator on M_i using Eq. (14) ;
            end
        end
    end
    Checking the expansion ;
    if (|M_i| ≥ |M_i|_max) then
        if Eq. (3) is stratified then
            Build a new expert and cleaning up M_i ;
        end
    end
    Perform the expert pruning (See Algorithm 2) ;
    Perform the Student learning (See Algorithm 3);
end
```

model size in the KIAM by employing the expert pruning approach, as described in Section 3.4; (3) The D-TS only uses a GAN-based Teacher module. In contrast, the KIAM can employ either a GAN, or a VAE, or a Diffusion model as an expert in the mixture model.

3.7. Algorithm Implementation

Algorithm. The detailed pseudo-code of KIAM*-GAN is provided in **Algorithm 1**, which summarizes the training algorithm in five steps.

- **(Updating the memory buffer $\mathcal{M}_i$).** When seeing a new data batch $\{\mathbf{x}_{i,j}\}_{j=1}^{b}$ at $\mathcal{T}_i'$, we update the memory buffer $\mathcal{M}_i$ by adding a new data batch $\{\mathbf{x}_{i,j}\}_{j=1}^{b}$,

Algorithm 2: Expert pruning.

while *True* **do**
  if *(*$|\mathcal{G}| > n_{\mathcal{A}}$*)* **then**
    Calculate $\mathbf{Q}$ using Eq. (7) ;
    Find a pair of experts by $a^\star, g^\star = \arg\min_{a,g=1,\cdots,c} \{\mathbf{Q}(a, g)\}$;
    $s_{a^\star} = \sum_{j=1,j\neq a^\star}^{c} \{\mathbf{Q}(a^\star, j)\}$, $s_{g^\star} = \sum_{j=1,j\neq g^\star}^{c} \{\mathbf{Q}(g^\star, j)\}$;
    if $s_{a^\star} > s_{g^\star}$ **then**
      Remove the expert $\mathcal{A}_{g^\star}$ from the Teacher module ;
    end
    else
      Remove the expert $\mathcal{A}_{a^\star}$ from the Teacher module
    end
  end
  else
    break;
  end
end

if the memory is not already full $|\mathcal{M}_i| < |\mathcal{M}_i|_{max}$. Otherwise, we delete the earliest memorized samples from $\mathcal{M}_i$ and then add the new batch $\{\mathbf{x}_{i,j}\}_{j=1}^{b}$.

- **(Teacher learning).** In the initial training phase from $\mathcal{T}_0'$ to $\mathcal{T}_{99}'$, we only learn a single expert in the Teacher module. We build the second expert $\mathcal{A}_2$ and freeze $\mathcal{A}_1$ at the training step $\mathcal{T}_{100}'$, in order to preserve the initial knowledge into the Teacher module such that it can support the dynamic expansion process during subsequent learning. We update the current expert on the memory buffer $\mathcal{M}_i$ using Eq. (13) and Eq. (14) for the GAN expert in a mini-batch learning manner, in which 'iterations' in **Algorithm 2** is determined by $iterations = (|\mathcal{M}_i|/b)epoch$ where $b = 64$ is the batch size, and $epoch$ is the training epoch.

- **(Checking the expansion).** In order to avoid frequently evaluating the dynamic expansion criterion, we only check the model expansion when the memory buffer is full, *i.e.* $|\mathcal{M}_i| = |\mathcal{M}|_{max}$. If the criterion from Eq. (3) is satisfied, we freeze the current expert $\mathcal{A}_c$ and add a new one $\mathcal{A}_{c+1}$, to the Teacher module. We also clean up $\mathcal{M}_i$ to avoid learning overlapping samples in subsequent learning.

Algorithm 3: Student learning.

for $s < iterations$ **do**
  $\{\mathbf{x}_{s,j}\}_{j=1}^{b}$ obtained from $\mathcal{M}_i$;
  for $s' < c-1$ **do**
    Generate the data batch $\{\mathbf{x}'_{s',j}\}_{j=1}^{b}$ using $\mathcal{A}_{s'}$;
  end
  Update the Student module on $\{\{\mathbf{x}_{s,j}\}_{j=1}^{b}, \{\mathbf{x}'_{1,j}\}_{j=1}^{b}, \cdots, \{\mathbf{x}'_{c-1,j}\}_{j=1}^{b}\}$ using Eq. (6) ;
end

- **(Expert pruning).** The detailed pruning process is provided in **Algorithm 2** through which we continually remove experts from the Teacher module, considered non-essential, using Eq. (8) and Eq. (9) until the number of experts in $\mathbf{A}$ is less than $n_{\mathcal{A}}$.
- **(Student learning).** We train the Student module by using Eq. (6), which involves generated and real data samples from the Teacher module which are mixed with those from the memory buffer $\mathcal{M}_i$, respectively. The Student is updated in a mini-batch learning manner, and the pseudocode for its updating process is provided in **Algorithm 3**. Then we return to **Step 1** for the next training step $\mathcal{T}_{i+1}$.

4. Theoretical analysis framework

In this section, we theoretically analyze the forgetting process of a generative model under task-free continual learning and how the proposed approach can relieve forgetting. This theoretical analysis extends the previous theoretical studies from [24, 51]. In the following section, we provide some important and necessary notations.

4.1. Preliminary

Notations. For a given VAE model, denoted as $\mathcal{B}$, let $q_{\varsigma}(\mathbf{z} \mid \mathbf{x})$ and $p_{\theta}(\mathbf{x} \mid \mathbf{z})$ denote the encoder and decoding distribution for $\mathcal{B}$, respectively. Let h represent a hypothesis function in the hypotheses space $\{h \in \mathcal{H} \mid \mathcal{H} : \mathcal{X} \to \mathcal{X}\}$, where $\mathcal{X} \in \mathbf{R}^d$ is the data space with d dimensions. We consider implementing $h \in \mathcal{H}$ using the reconstruction process, achieved using the VAE model $\mathcal{B}$. Let $\mathcal{L}\colon \mathcal{X} \times \mathcal{X} \to \mathbf{R}$ denote an error function, which is implemented using the square-loss function

Algorithm 4: Expert pruning.

```
while True do
    if (|G| > n_A) then
        Calculate Q using Eq.(7) of the paper ;
        Find a pair of experts by a*, g* = arg min_{a,g=1,...,c} {Q(a, g)};
        s_{a*} = Σ_{j=1,j≠a*}^{c} {Q(a*, j)} , s_{g*} = Σ_{j=1,j≠g*}^{c} {Q(g*, j)};
        if s_{a*} > s_{g*} then
            Remove the expert A_{g*} from the Teacher module ;
        end
        else
            Remove the expert A_{a*} from the Teacher module
        end
    end
    else
        break;
    end
end
```

$\mathcal{L}(\mathbf{x}, \mathbf{x}') = \|\mathbf{x} - \mathbf{x}'\|^2, \mathbf{x}, \mathbf{x}' \in \mathcal{X}$. We assume that the error function $\mathcal{L}\colon \mathcal{X} \times \mathcal{X} \to \mathbf{R}$ is bounded, $\forall(\mathbf{x}, \mathbf{x}') \in \mathcal{X}^2, \mathcal{L}(\mathbf{x}, \mathbf{x}') \leq C$, for some $C > 0$.

Definition 1. ***(Model risk.)*** *For a given memory distribution $\mathrm{P}_{\mathcal{M}_i}$, a risk for $\mathcal{B}$ on $\mathrm{P}_{\mathcal{M}_i}$ is defined as $\mathcal{E}_{\mathrm{P}_{\mathcal{M}_i}}(h, f_{\mathrm{P}_{\mathcal{M}_i}}) = \mathbb{E}_{\mathbf{x}\sim\mathrm{P}_{\mathcal{M}_i}}\mathcal{L}(h(\mathbf{x}), f_{\mathrm{P}_{\mathcal{M}_i}}(\mathbf{x}))$, where $f_{\mathrm{P}_{\mathcal{M}_i}} \in \mathcal{H}$ is an identity function.*

In the following, we define the discrepancy distance for measuring the similarity between two distributions.

Definition 2. ***(Discrepancy distance.)*** *Let $\mathrm{P}_{D_j^T}$ be a probability distribution for D_j^T over $\mathcal{X}$. The discrepancy distance on two distributions $\mathrm{P}_{\mathrm{D_i^T}}$ and $\mathrm{P}_{\mathrm{D_j^T}}$, is defined as:*

$$\mathcal{L}_{\mathrm{disc}}\left(\mathrm{P}_{\mathrm{D_i^T}}, \mathrm{P}_{\mathrm{D_j^T}}\right) = \sup_{(h,h')\in\mathcal{H}} \Big| \mathbb{E}_{\mathbf{x}\sim\mathrm{P}_{D_i^T}}\left[\mathcal{L}\left(h'\left(\mathbf{x}\right), h\left(\mathbf{x}\right)\right)\right] - \mathbb{E}_{\mathbf{x}\sim\mathrm{P}_{D_j^T}}\left[\mathcal{L}\left(h'\left(\mathbf{x}\right), h\left(\mathbf{x}\right)\right)\right] \Big|. \tag{22}$$

Definition 3. ***(Rademacher complexity).*** *Let $\mathcal{H}$ represent a hypothesis class, For a given unlabeled sample $U = \{\mathbf{x}_i\}_{i=1}^m$, the Rademacher complexity of $\mathcal{H}$ with*

Algorithm 5: Student learning.

```
for s < iterations do
    {x_{s,j}}_{j=1}^b obtained from M_i ;
    for s' < c − 1 do
        Generate the data batch {x'_{s',j}}_{j=1}^b using A_{s'};
    end
    Update the Student module on {{x_{s,j}}_{j=1}^b, {x'_{1,j}}_{j=1}^b, ··· , {x'_{c−1,j}}_{j=1}^b}
      using Eq.(6) of the paper ;
end
```

respect to the sample U is defined as follows :

$$\mathrm{Re}_U\left(\mathcal{H}\right) = \mathop{\mathbb{E}}_{\mathcal{K}}\left[\sup_{h\in\mathcal{H}}\frac{2}{m}\sum_{i=1}^{m}\mathcal{K}_i h\left(\mathbf{x}_i\right)\right], \tag{23}$$

where $\mathcal{K}_i$ is an independent uniform random variable within $\{-1,+1\}$. The Rademacher complexity for the whole hypothesis class is defined as :

$$\mathrm{Re}_n\left(\mathcal{H}\right) = \mathbb{E}_{U\sim(D)^n}\mathrm{Re}_U\left(\mathcal{H}\right). \tag{24}$$

Definition 4. ***(Empirical discrepancy distance.)*** *In practice, we only have access to finite training sets $\widehat{D}_i^T$ and $\widehat{D}_j^T$ of sample sizes m_i and m_j, respectively. Let $\widehat{\mathrm{P}}_{D_i^T}$ and $\widehat{\mathrm{P}}_{D_j^T}$ denote the empirical distribution for $\widehat{D}_i^T$ and $\widehat{D}_j^T$, respectively. Then we estimate the discrepancy distance with probability $1-\delta, \delta\in(0,1)$:*

$$\begin{aligned}\mathcal{L}_{\mathrm{disc}}\big(\mathrm{P}_{D_i^T},\mathrm{P}_{D_j^T}\big) \le \mathcal{L}_{\mathrm{disc}}\big(\widehat{\mathrm{P}}_{D_i^T},\widehat{\mathrm{P}}_{D_j^T}\big) + 8\big(\mathrm{Re}_{\widehat{D}_i^T}\left(\mathcal{H}\right)+\mathrm{Re}_{\widehat{D}_i^T}\left(\mathcal{H}\right)\big) \\ + 3M\left(\sqrt{\frac{\log\left(\frac{4}{\delta}\right)}{2m_i}}+\sqrt{\frac{\log\left(\frac{4}{\delta}\right)}{2m_j}}\right),\end{aligned} \tag{25}$$

where $\mathrm{Re}_{\widehat{D}_i^T}$ denotes the Rademacher complexity and $M > 0$. We consider $\widehat{\mathcal{L}}_{\mathrm{disc}}(\cdot)$ to represent the right-hand side (RHS) of Eq. (25)*.*

4.2. Forgetting Analysis for the Static Model

The ELBO is the primary objective function for training a VAE model, which has also been employed as a criterion for performance evaluation [74, 75, 4]. In the following, we introduce an upper bound to the negative ELBO to investigate the forgetting behaviour of a single VAE model in task-free continual learning.

Theorem 1. *Let $\mathcal{S}$ be a data stream and $\mathrm{P}_{\mathbf{x}'(1:i)}$ be a probability distribution of all previously visited i data batches $\{\{\mathbf{x}_{1,j}\}_{j=1}^{b}, \cdots, \{\mathbf{x}_{i,j}\}_{j=1}^{b}\} \in \mathcal{S}$ at the training time $\mathcal{T}_i$. We assume that the probabilistic representation of the decoder for a single VAE model is a Gaussian distribution with a diagonal covariance matrix (we consider that the diagonal element is $1/\sqrt{2}$). We derive an upper bound to the negative ELBO at $\mathcal{T}_i$, as :*

$$\begin{aligned}\mathbb{E}_{\mathrm{P}_{\mathbf{x}'(1:i)}}\Big[-\mathcal{L}_{ELBO}\big(\mathbf{x}'(1:i);h\big)\Big] \leq\ & \mathcal{E}_A\big(\mathrm{P}_{\mathbf{x}'(1:i)},\mathrm{P}_{\mathcal{M}_i}\big)\\ &+\mathbb{E}_{\mathrm{P}_{\mathcal{M}_i}}\Big[-\mathcal{L}_{ELBO}\big(\mathbf{x}_{\mathcal{M}_i};h\big)\Big]+\big|KL_1-KL_2\big|,\end{aligned} \tag{26}$$

where $\mathbf{x}_{\mathcal{M}_i}$ and $\mathbf{x}'(1:i)$ are the variables drawn from $\mathrm{P}_{\mathcal{M}_i}$ and $\mathrm{P}_{\mathbf{x}'(1:i)}$, respectively. $\mathcal{E}_A(\mathrm{P}_{\mathbf{x}'(1:i)},\mathrm{P}_{\mathcal{M}_i})$ is defined as :

$$\begin{aligned}\mathcal{E}_A\big(\mathrm{P}_{\mathbf{x}'(1:i)},\mathrm{P}_{\mathcal{M}_i}\big) = \widehat{\mathcal{L}}_{\text{disc}}\big(\mathrm{P}_{\mathbf{x}'(1:i)},\mathrm{P}_{\mathcal{M}_i}\big)+\mathcal{E}_{\mathrm{P}_{\mathbf{x}'(1:i)}}\big(h^*_{\mathbf{x}'(1:i)},f_{\mathbf{x}'(1:i)}\big)\\ +\mathcal{E}_{\mathrm{P}_{\mathbf{x}'(1:i)}}\big(h^*_{\mathbf{x}'(1:i)},h^*_{\mathcal{M}_i}\big),\end{aligned} \tag{27}$$

where $h^*_{\mathcal{M}_i}=\arg\min_{h\in\mathcal{H}}\mathcal{E}_{\mathrm{P}_{\mathbf{x}_{\mathcal{M}_i}}}(h,f_{\mathcal{M}_i})$ and $h^*_{\mathbf{x}'(1:i)}=\arg\min_{h\in\mathcal{H}}\mathcal{E}_{\mathrm{P}_{\mathbf{x}'(1:i)}}(h,f_{\mathbf{x}'(1:i)})$ are the optimal hypotheses for $\mathrm{P}_{\mathcal{M}_i}$ and $\mathrm{P}_{\mathbf{x}'(1:i)}$, respectively. KL_1 and KL_2 are defined as :

$$\begin{aligned}KL_1&=\mathbb{E}_{\mathrm{P}_{\mathbf{x}'(1:i)}}KL\big(q_{\phi^i}(\mathbf{z}\,|\,\mathbf{x}'(1:i))\;||\;p(\mathbf{z})\big),\\ KL_2&=\mathbb{E}_{\mathrm{P}_{\mathcal{M}_i}}KL\big(q_{\phi^i}(\mathbf{z}\,|\,\mathbf{x}_{\mathcal{M}_i})\;||\;p(\mathbf{z})\big).\end{aligned} \tag{28}$$

$q_{\phi^i}(\mathbf{z}\,|\,\cdot)$ is the encoding distribution implemented by a single VAE model trained on the memory buffer $\mathcal{M}_i$ at $\mathcal{T}_i$.

Proof. From **Definition** 4 , we know that $\mathcal{L}_{\text{disc}}(\mathbb{P}_{\mathbf{x}'(1:i)},\mathbb{P}_{\mathcal{M}_i})\leq\widehat{\mathcal{L}}_{\text{disc}}(\mathbb{P}_{\mathbf{x}'(1:i)},\mathbb{P}_{\mathcal{M}_i})$. By considering Theorem 1 from [51], see also Theorem 1 from [24], we replace $\mathcal{L}_{\text{disc}}(\mathbb{P}_{\mathbf{x}'(1:i)},\mathbb{P}_{\mathcal{M}_i})$ by using $\widehat{\mathcal{L}}_{\text{disc}}(\mathbb{P}_{\mathbf{x}'(1:i)},\mathbb{P}_{\mathcal{M}_i})$, resulting in :

$$\begin{aligned}\mathcal{E}_{\mathbb{P}_{\mathbf{x}'(1:i)}}\big(h,f_{\mathbb{P}_{\mathbf{x}'(1:i)}}\big)\leq\ & \mathcal{E}_{\mathbb{P}_{\mathcal{M}_i}}\big(h,h^*_{\mathbb{P}_{\mathcal{M}_i}}\big)\\ &+\widehat{\mathcal{L}}_{\text{disc}}\big(\mathbb{P}_{\mathbf{x}'(1:i)},\mathbb{P}_{\mathcal{M}_i}\big)\\ &+\mathcal{E}_{\mathcal{C}_i}\big(h^*_{\mathbb{P}_{\mathcal{M}_i}},h^*_{\mathbb{P}_{\mathbf{x}'(1:i)}}\big)\\ &+\mathcal{E}_{\mathbb{P}_{\mathbf{x}'(1:i)}}\big(h^*_{\mathbb{P}_{\mathbf{x}'(1:i)}},f_{\mathbb{P}_{\mathbf{x}'(1:i)}}\big),\end{aligned} \tag{29}$$

which is proved in Theorem 1 from [51].

From Eq. (29), according to the bound on the KL divergence :

$$\begin{aligned}
&\mathbb{E}_{\mathbb{P}_{\mathbf{x}'(1:i)}} KL(q_{\phi^i}(\mathbf{z} \mid \mathbf{x}'(1:i)) \mid\mid p(\mathbf{z})) \leq \\
&\mathbb{E}_{\mathbb{P}_{\mathcal{M}_i}} KL(q_{\phi^i}(\mathbf{z} \mid \mathbf{x}_{\mathcal{M}_i}) \mid\mid p(\mathbf{z})) \\
&+ |\mathbb{E}_{\mathbb{P}_{\mathcal{M}_i}} KL(q_{\phi^i}(\mathbf{z} \mid \mathbf{x}_{\mathcal{M}_i}) \mid\mid p(\mathbf{z})) \\
&- \mathbb{E}_{\mathbb{P}_{\mathbf{x}'(1:i)}} KL(q_{\phi^i}(\mathbf{z} \mid \mathbf{x}'(1:i)) \mid\mid p(\mathbf{z}))| \,.
\end{aligned} \tag{30}$$

We also know that $\mathcal{L}_{ELBO}\left(\mathbf{x}; \{\theta, \phi\}\right)$ is expressed as :

$$\begin{aligned}
\mathcal{L}_{ELBO}\left(\mathbf{x}; \{\theta, \phi\}\right) := &\mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})}\left[\log p_\theta\left(\mathbf{x} \mid \mathbf{z}\right)\right] \\
&- KL\left[q_\phi\left(\mathbf{z} \mid \mathbf{x}\right) \mid\mid p\left(\mathbf{z}\right)\right] \,,
\end{aligned} \tag{31}$$

When the decoder models a Gaussian distribution, $\log p_\theta\left(\mathbf{x} \mid \mathbf{z}\right)$ is represented as :

$$\log p_\theta\left(\mathbf{x} \mid \mathbf{z}\right) = -\frac{1}{2\sigma_\theta^2\left(\mathbf{z}\right)}\|\mathbf{x} - \mu_\theta\left(\mathbf{z}\right)\|^2 - \frac{1}{2}\log 2\pi\sigma_\theta^2\left(\mathbf{z}\right) \tag{32}$$

where $\sigma_\theta\left(\mathbf{z}\right)$ and $\mu_\theta\left(\mathbf{z}\right)$ are the variance and mean of the Gaussian distribution, obtained by the decoder. $\|\cdot\|^2$ represents the reconstruction error (square loss). We implement the decoder by a Gaussian distribution $\mathcal{N}(\mu_\theta\left(\mathbf{z}\right), \sigma\mathbf{I})$, where $\mu_\theta\left(\mathbf{z}\right)$ is a deep convolutional neural network and $\mathbf{I}$ is the identity matrix and σ is a fixed variance.

Since h is the hypothesis of the model, implemented as an encoding-decoding process, we have

$$\begin{aligned}
&\mathcal{L}_{ELBO}(\mathbf{x}'(1:i); h) \\
&= -\frac{1}{2\sigma^2}\mathcal{L}(h(\mathbf{x}'(1:i)), f_{\mathbb{P}_{\mathbf{x}'(1:i)}}(\mathbf{x}'(1:i))) \\
&- \frac{1}{2}\log 2\pi\sigma^2 - KL(q_{\phi^i}(\mathbf{z} \mid \mathbf{x}'(1:i)) \mid\mid p(\mathbf{z}))
\end{aligned} \tag{33}$$

Then by considering the negative ELBO, we have :

$$\begin{aligned}
&- \mathcal{L}_{ELBO}(\mathbf{x}'(1:i); h) = \\
&\frac{1}{2\sigma^2}\mathcal{L}(h(\mathbf{x}'(1:i)), f_{\mathbb{P}_{\mathbf{x}'(1:i)}}(\mathbf{x}'(1:i))) \\
&+ \frac{1}{2}\log 2\pi\sigma^2 + KL(q_{\phi^i}(\mathbf{z} \mid \mathbf{x}'(1:i)) \mid\mid p(\mathbf{z}))
\end{aligned} \tag{34}$$

And we know that:

$$\mathcal{R}_{\mathbb{P}_{\mathbf{x}'(1:i)}}(h, f_{\mathbb{P}_{\mathbf{x}'(1:i)}}) = \mathbb{E}_{\mathbf{x}\sim\mathbb{P}_{\mathbf{x}'(1:i)}}\mathcal{L}(h(\mathbf{x}'(1:i)), f_{\mathbb{P}_{\mathbf{x}'(1:i)}}(\mathbf{x}'(1:i))) \tag{35}$$

and we have :

$$\begin{aligned}
&\mathbb{E}_{\mathbf{x}'(1:i)\sim\mathbb{P}_{\mathbf{x}'(1:i)}}\left[-\mathcal{L}_{ELBO}(\mathbf{x}'(1:i); h)\right] = \\
&\mathbb{E}_{\mathbf{x}'(1:i)\sim\mathbb{P}_{\mathbf{x}'(1:i)}}\left\{\frac{1}{2\sigma^2}\mathcal{L}(h(\mathbf{x}'(1:i)), f_{\mathbb{P}_{\mathbf{x}'(1:i)}}(\mathbf{x}'(1:i)))\right. \\
&\left. + KL(q_{\phi^i}(\mathbf{z} \mid \mathbf{x}'(1:i)) \mid\mid p(\mathbf{z}))\right\} + \frac{1}{2}\log 2\pi\sigma^2\,.
\end{aligned} \tag{36}$$

We observe that $\frac{1}{2}\log 2\pi\sigma^2$ and $\frac{1}{2\sigma^2}$ are constants and we set $\sigma = \frac{1}{\sqrt{2}}$ in order to simplify the notations. Therefore, Eq. (36) is rewritten as :

$$\begin{aligned}
&\mathbb{E}_{\mathbf{x}'(1:i)\sim\mathbb{P}_{\mathbf{x}'(1:i)}}\left[-\mathcal{L}_{ELBO}(\mathbf{x}'(1:i); h)\right] = \\
&\mathbb{E}_{\mathbf{x}'(1:i)\sim\mathbb{P}_{\mathbf{x}'(1:i)}}\{\mathcal{L}(h(\mathbf{x}'(1:i)), f_{\mathbb{P}_{\mathbf{x}'(1:i)}}(\mathbf{x}'(1:i))) \\
&+ KL(q_{\phi^i}(\mathbf{z} \mid \mathbf{x}'(1:i)) \mid\mid p(\mathbf{z}))\} + \frac{1}{2}\log\pi
\end{aligned} \tag{37}$$

We then consider Equations (30) and (29) and we have :

$$\begin{aligned}
&\mathbb{E}_{\mathbf{x}'(1:i)\sim\mathbb{P}_{\mathbf{x}'(1:i)}}\{\mathcal{L}(h(\mathbf{x}'(1:i)), f_{\mathbb{P}_{\mathbf{x}'(1:i)}}(\mathbf{x}'(1:i))) \\
&+ KL(q_{\phi^i}(\mathbf{z}|\mathbf{x}'(1:i)) \mid\mid p(\mathbf{z}))\} \leq \\
&\mathbb{E}_{\mathbf{x}_{\mathcal{M}_i}\sim\mathbb{P}_{\mathcal{M}_i}}\left\{\mathcal{L}\left(h(\mathbf{x}_{\mathcal{M}_i}), h^*_{\mathbb{P}_{\mathcal{M}_i}}(\mathbf{x}_{\mathcal{M}_i})\right)\right. \\
&\left. + KL(q_{\phi^i}(\mathbf{z} \mid \mathbf{x}_{\mathcal{M}_i}) \mid\mid p(\mathbf{z}))\right\} \\
&+ |KL_1 - KL_2| \; + \mathcal{E}_A\left(\mathbb{P}_{\mathbf{x}'(1:i)}, \mathbb{P}_{\mathcal{M}_i}\right)\;,
\end{aligned} \tag{38}$$

where KL_1 and KL_2 are defined in Eq. (28). We assume that $h^*_{\mathbb{P}_{\mathcal{M}_i}}(\mathbf{x}_{\mathcal{M}_i})$ is a perfect model for $\mathbb{P}_{\mathcal{M}_i}$. Then we have $h^*_{\mathbb{P}_{\mathcal{M}_i}}(\mathbf{x}_{\mathcal{M}_i}) = f^*_{\mathbb{P}_{\mathcal{M}_i}}(\mathbf{x}_{\mathcal{M}_i})$. It notes that we can add the constant $\frac{1}{2}\log\pi$ in both sides of Eq. (38). According to Eq. (37), we can rewrite Eq. (38) as :

$$\begin{aligned}
&\mathbb{E}_{\mathbf{x}'(1:i)\sim\mathbb{P}_{\mathbf{x}'(1:i)}}[-\mathcal{L}_{ELBO}\left(\mathbf{x}'(1:i); h\right)] \leq \\
&\mathbb{E}_{\mathbf{x}\sim\mathbb{P}_{\mathcal{M}_i}}\left[-\mathcal{L}_{ELBO}\left(\mathbf{x}_{\mathcal{M}_i}; h\right)\right] \\
&+ |KL_1 - KL_2| + \mathcal{E}_A\left(\mathbb{P}_{\mathbf{x}'(1:i)}, \mathbb{P}_{\mathcal{M}_i}\right)
\end{aligned} \tag{39}$$

□

This corresponds to Eq. (26) and proves Theorem 1.

We have several observations regarding Theorem 1 :

- The knowledge loss and gain achieved by a single VAE model at each training time $\{\mathcal{T}_i \,|\, i = 1, \cdots, t\}$ can be evaluated using Eq. (26).

- The first term from the right-hand side of Eq. (26), representing the discrepancy distance, plays a vital role in the model's performance. A large discrepancy distance term indicates that the memory buffer $\mathcal{M}_i$ does not store sufficient information about all previously learnt samples and thus the right-hand side of Eq. (26) is large, resulting in a small ELBO on the target distribution achieved by a single VAE model, corresponding to the forgetting process.

- A single fixed-capacity memory buffer can not store sufficient information for all previously learnt samples, especially when learning a long-term data stream. From Eq. (26), we also find that an appropriate sample selection approach is critical for the memory-based methods.

4.3. Theoretical Guarantees

In this section, we analyze the forgetting process of a dynamic expansion model by extending the theoretical analysis of using a single VAE model. In addition, we also theoretically show that the proposed dynamic expansion mechanism can ensure a lightweight network architecture while providing good performance.

Theorem 2. *For a given Teacher module with c already learnt experts $\mathbf{A} = \{\mathcal{A}_1, \cdots, \mathcal{A}_c\}$ at $\mathcal{T}_i$, let h be a Student model implemented using a VAE model. We derive an upper bound to the negative ELBO at $\mathcal{T}_i$, as :*

$$\begin{aligned}\mathbb{E}_{\mathrm{P}_{\mathbf{x}'(1:i)}}\Big[-\mathcal{L}_{ELBO}\big(\mathbf{x}'(1:i);h\big)\Big] &\leq \mathcal{E}_A\big(\mathrm{P}_{\mathbf{x}'(1:i)}, \mathrm{P}_{\mathcal{M}_i\otimes\theta_{(1:c)}}\big) \\ &+ \mathbb{E}_{\mathrm{P}_{\mathcal{M}_i\otimes\theta_{(1:c)}}}\Big[-\mathcal{L}_{ELBO}\big(\mathbf{x}'';h\big)\Big] + \big|KL_1 - KL_{\mathcal{M}_i\otimes\theta_{(1:c)}}\big|,\end{aligned} \tag{40}$$

where $KL_{\mathcal{M}_i\otimes\theta_{(1:c)}} = KL(q_{\phi^i}(\mathbf{z}\,|\,\mathbf{x}'') \,||\, p(\mathbf{z}))$ and $\mathbf{x}''$ is the latent variable drawn from $\mathrm{P}_{\mathcal{M}_i\otimes\theta_{(1:c)}}$ which is a probability distribution formed by the samples uniformly drawn from $\{\mathrm{P}_{\theta_1}, \cdots, \mathrm{P}_{\theta_c}, \mathcal{P}_{\mathcal{M}_i}\}$. From Theorem 2, we have several observations:

- Compared to the static/single model (Theorem 1), the Student model h in Eq. (40) can significantly reduce forgetting when the Teacher experts capture and preserve more knowledge.

- When each Teacher expert is appropriately built to learn a unique data distribution during the training, the discrepancy distance term $\mathcal{E}_A(\mathrm{P}_{\mathbf{x}'(1:i)}, \mathrm{P}_{\mathcal{M}_i \otimes \theta_{(1:c)}})$ in Eq. (40) would be stable, thus relieving forgetting.

- The study from [24] demonstrates a similar conclusion to Eq. (40). However, the theoretical analysis in [24] relies on the task information, which can only be applied in the task-known continual learning case. In contrast, our theoretical analysis can be used in a more realistic continual learning paradigm, such as TFCL.

We also find that encouraging the diversity of the generator's characteristic distributions $\{\mathrm{P}_{\theta_1}, \cdots, \mathrm{P}_{\theta_c}\}$ would maintain a small discrepancy distance term in Eq. (40) while reducing the necessary number of experts. A reasonable approach for expanding the Teacher module is to maximize the distance between the trained experts $\{\mathcal{A}_1, \cdots, \mathcal{A}_{c-1}\}$ and the current expert $\mathcal{A}_c$, expressed as :

$$\mathrm{P}^{\star}_{\theta_c^m} = \arg\max_{\{\mathrm{P}_{\theta_c^m}\}_{m=i+1}^{t}} \frac{1}{c-1} \sum\nolimits_{j=1}^{c-1} \left\{ f_p(\mathrm{P}_{\theta_j}, \mathrm{P}_{\theta_c^m}) \right\}, \tag{41}$$

where $\mathrm{P}^{\star}_{\theta_c^m}$ is an optimal solution that maximizes the distance and m is the index of the training step. However, searching $\mathrm{P}^{\star}_{\theta_c^m}$ in Eq. (41) needs accessing all future training steps $\{\mathcal{T}_{i+1}, \cdots, \mathcal{T}_t\}$ simultaneously, which is not feasible in continual learning. To deal with this problem, we have proposed a novel expansion mechanism in Eq. (3), which implements the goal expressed in Eq. (41), given that it can ensure maintaining the discrepancy among Teacher experts when performing model expansion. Unlike Eq. (41), which requires accessing all training steps, the proposed dynamic expansion mechanism from Eq. (3) introduces employing a threshold ν to automatically enhance the Teacher's capacity, which can capture novel information and avoid forgetting. However, existing dynamic expansion approaches [20, 21] fail to achieve the goal of Eq. (41) because their expansion mechanisms do not consider the knowledge diversity among experts.

5. Experiments

5.1. Settings And Baselines

Parameter setting : In the experiment we employ a set of six data domains, including MNIST [77], SVHN [78], Fashion [79], InverseFashion (IFashion), Rotated MNIST (RMNIST) and CIFAR10 [80]. The IFashion is created by inverting each pixel for all images. For Rotated MNIST (RMNIST), we rotate all images from

Table 1: Evaluation of the FID for the images generated by various models under the MSFIRC setting. Last column represents the final number of Teacher experts after finishing the whole training.

Methods	**MNIST**	**SVHN**	**Fashion**	**IFashion**	**RMNIST**	**CIFAR10**	**Average**	**No**
finetune	174.1	148.3	237.0	229.1	159.2	216.4	194.0	1
Reservoir [43]	127.2	159.3	213.4	201.6	110.2	113.3	154.2	1
LTS [10]	44.8	62,9	92.9	83.1	41.8	80.3	67.7	1
LGM [7]	104.8	134.3	194.3	168.1	94.8	91.5	131.3	1
CN-DPM [20]	118.7	73.4	120.7	120.3	97.9	97.6	104.8	18
DEDM [76]	10.6	75.2	66.8	11.0	70.4	69.5	50.6	9
KIAM-GAN	11.6	70.6	101.9	29.9	11.41	68.6	49.0	16
KIAM-VAE	122.9	73.6	109.2	104.3	119.1	86.4	102.6	11
KIAM*-GAN	12.0	74.6	69.8	22.3	11.4	68.5	43.1	7
KIAM*-VAE	82.6	82.5	127.0	132.9	88.8	86.3	100.0	7
KIAM-DDPM	10.6	71.3	65.9	23.6	11.2	69.0	**42.0**	7

the MNIST database by 180 degrees. We create a data stream $\mathcal{S}$, from all these training sets, named MSFIRC. By considering IFashion and RMNIST datasets in the data stream $\mathcal{S}$, we can involve both dissimilar and similar datasets in our experiment. We also consider a popular setting, class-incremental learning, in which each dataset/domain is split into five sets, each set consisting of images from two different classes [81]. Unlike [81], which only considers a single dataset, we build a more complex data stream $\mathcal{S}$ that consists of all parts of the six datasets, and we call this setting as Class Incremental CI-MSFIRC.

The batch size for processing the images is considered as $b = 64$. We set the number of epochs for each training step to 1. The threshold used in Eq. (3) for controlling the number of experts for the Teacher module is chosen within the range $\nu \in [0, 200]$. In addition, we consider setting the maximum memory size for STM as 5000 samples for MSFIRC and CI-MSFIRC. Since we mainly evaluate the performance of various models on lifelong generative modelling under task-free continual learning, we follow the set-up from [23] which adopts the Fréchet Inception Distance (FID) [49] to evaluate the performance of various models. We use a Tesla V100-SXM2 (32GB) GPU and the RHEL 8 operating system for our experiments.

Network Architecture. We introduce the details of the network architecture. For the VAE generator of each expert in the teacher module and the decoder of the Student, we adopt the same CNN network that consists of five convolution layers $\{256, 256, 256, 256, 3\}$. For the inference model $q(\mathbf{z} \mid \mathbf{x})$ of the Student, we consider using a network architecture consisting of four convolution layers and two

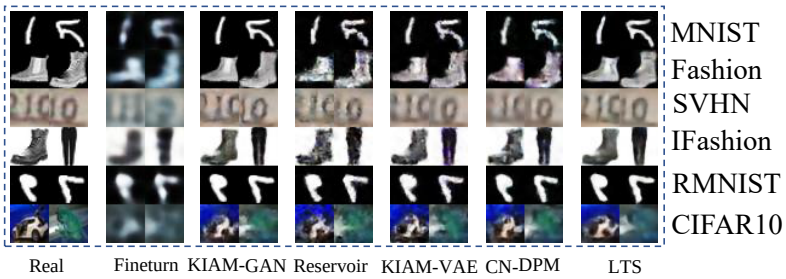


Figure 3: The cross-domain reconstruction results of various models under MSFIRC setting.

fully connected layers, aiming to return the hyperparameters of a Gaussian distribution. The number of kernels in each convolution layer is of $\{64, 128, 256, 512\}$ and the kernel size is 3×3. The number of hidden nodes in the fully connected layer is 256. For the input size of $64 \times 64 \times 3$, the generator for each expert consists of 6 convolution layers $\{256, 256, 256, 256, 128, 3\}$. The decoder of the Student also uses the same network architecture as the generator. The inference model $q(\mathbf{z} \mid \mathbf{x})$ of the Student and each expert consists of four convolution layers $\{64, 128, 256, 512\}$, one hidden layers $\{1024\}$ and two separate layers $\{256\}$ which are used to infer the hyperparameters of a Gaussian distribution. For each expert of the GAN-based teacher, we use for the same network structure as in [82]. For the DDPM-based teacher module, we employ the same network architecture from the improved DDPM model [83], which is a U-Net [84] and the detailed network information can be found in [85]. Compared to the GAN and VAE-based teacher modules, the DDPM-based teacher module uses a large-scale neural network with more parameters, leading to considerable computational costs and memory resources. In addition, the DDPM also requires many more optimization iterations than the GANs and VAEs when generating one image, which leads to additional generation times for the images.

Baselines : The majority of continual learning methods only focus on the classification tasks and thus can not be used for image generative modelling. Therefore, we evaluate the proposed approach with the baselines that have been used for image generative modelling, such as : Reservoir [43], Lifelong Teacher Student (LTS) [10], Lifelong Generative Modelling (LGM) [7], and CN-DPM [20], respectively. For a fair comparison with CN-DPM, we also train a Student model to learn data generated by CN-DPM and from the memory buffer. In addition, we also assign a short-term memory buffer with a maximum size of 5,000 for LGM and LTS to support TFCL. Furthermore, we consider a learning rate of 0.0002 for the Adam algorithm [86], when training all models.

Table 2: The FID of various models under the CI-MSFIRC setting.

Methods	MNIST	SVHN	Fashion	IFashion	RMNIST	CIFAR10	Average	No
finetune	158.1	167.6	246.2	233.3	138.6	229.4	195.6	1
Reservoir [43]	141.7	163.6	220.0	200.1	127.1	115.5	161.3	1
LTS [10]	101.9	99.4	140.6	139.5	99.9	95.5	112.8	1
LGM [7]	108.5	122.1	189.5	175.9	96.6	92.4	130.9	1
CN-DPM [20]	90.9	62.0	109.0	95.0	77.9	95.5	88.4	18
KIAM-GAN	16.7	65.1	44.5	43.9	27.9	85.2	**47.2**	11
KIAM-VAE	102.6	69.9	117.1	99.5	113.0	82.7	97.5	11
KIAM*-GAN	13.5	72.7	89.9	52.1	12.4	71.9	52.1	7
KIAM*-VAE	131.0	70.3	106.7	92.2	126.5	87.7	102.4	7

5.2. Generative Modeling Tasks Under TFCL

In this section, we investigate the performance of the Student module on a data stream with multiple data distributions. We provide the FID performance for generated images, following the MSFIRC setting in Tab. 1. KIAM-VAE, KIAM-GAN and KIAM-DDPM are the dynamic expansion models whose Teacher experts are implemented using VAE, GAN, or the Denoising Diffusion Probabilistic Model (DDPM), respectively. Adding '*' to the name of the generator in Table 1 denotes that the dynamic expansion model employs the proposed expert pruning approach from Section 3.4 to remove the redundant Teacher experts from the model. KIAM-GAN-based methods usually have higher FID scores than KIAM-VAE-based methods, while the best results are by the KIAM-DDPM model. KIAM-GAN has better performance than CN-DPM while employing fewer Teacher experts. Image reconstruction results by various models, after training on MSFIRC are provided in Fig. 3. Overall, the DDPM and GAN-based teachers provide better results by a large margin when comparing with using VAEs as teacher experts. In addition, KIAM*-GAN, when compared to KIAM-GAN, reduces the number of experts from 16 to 7 while still maintaining a good performance. In addition, KIAM*-VAE reduces the number of components by pruning from 11 to 7. These results show that the proposed expert pruning approach can further reduce the Teacher model size and eventually employ fewer parameters without sacrificing performance. By using the powerful DDPM-based teacher module can further improve the student's performance as demonstrated by KIAM-DDPM.

In the following, we investigate the performance of various models under the class-incremental setting. In this setting, each dataset is divided into five parts, and

Table 3: The FID for the generated results of various models under the CelebA-Chair learning setting.

Methods	CelebA	3D-Chair	Average	No
finetune	35.79	9.90	22.85	1
Reservoir [43]	20.82	11.04	15.93	1
LTS [10]	20.68	11.47	16.07	1
LGM [7]	21.58	11.84	16.71	1
CN-DPM [20]	20.19	11.45	15.82	11
KIAM-GAN	18.05	11.32	14.68	3
KIAM-VAE	20.63	11.79	16.21	5
KIAM-DDPM	17.14	11.35	**14.24**	3

each part consists of samples from two classes. This setting is more challenging since the model requires dealing with both class and domain shifts. We report the performance of the Student module in Tab. 2. From this table we can observe that the static models, including finetune, Reservoir, LTS and LGM, perform worse on the class-incremental setting when compared with the domain-incremental setting. This is because the static model has a fixed capacity and can not capture more classes during the training. From the results of Tab. 2, we observe that KIAM-GAN outperforms KIAM*-GAN, which shows that learning more components can benefit the performance for the class-incremental setting. The reason behind this case is that more components can potentially capture more category information and thus are suitable for the class-incremental setting in which the class changes over time, at the cost of requiring more parameters. As in the previous experiments, GAN-based models perform better than the VAE-based methods in the class-incremental setting, showing that the high-quality generative replay samples play a critical role in lifelong generative modelling. Furthermore, the proposed KIAM-GAN outperforms other baselines in the class-incremental setting.

5.3. Learning Complex Data Streams Under TFCL

We study the performance achieved by the Student module when learning datasets consisting of complex images. Following the setting from [10], we consider 5,000 sample images for testing from each of the datasets CelebA [87] and 3D-chair [88] while the remaining data samples are used for training. We create a data stream named CelebA-Chair consisting of these training samples that are resized as $64 \times 64 \times 3$. The same setting from MSFIRC is used for CelebA-Chair, and we provide the results achieved by the Student module in Table 3. We observe that the static model can achieve good performance on the last dataset (3D-Chair)

(a) The GAN-based teacher module.

(b) The DDPM-based teacher module.

Figure 4: The generation results from the Teacher module after the CelebA to 3D-Chair continual learning.

while performing worse on the first dataset (CelebA) in the data stream, which shows that the static model suffers from forgetting. In contrast, the dynamic expansion models can perform well on both the first and second datasets. In addition, the proposed KIAM-GAN outperforms other baselines while employing fewer components when compared with CN-DPM. Furthermore, compared with KIAM-VAE, CN-DPM achieves a slightly better performance but employs much more experts than KIAM-VAE. Since both KIAM-GAN and KIAM-VAE employ fewer components after CelebA-Chair lifelong learning, we do not further reduce the model size for them using the proposed expert pruning. In addition, we also compare the generation results from KIAM-GAN and KIAM-DDPM in Fig. 4-a and 4-b, which indicate that the DDPM-based teacher module provides better generation results than the GAN-based teacher module.

We also investigate whether the proposed Student model can learn meaningful latent representations across multiple data domains without forgetting. When the learning of CelebA to 3D-Chair is finished, we perform an interpolation experiment in which we interpolate latent variables from two different images and the interpolated code is then fed into the decoder for the image reconstruction. We provide the visual results in Fig. 5, which shows on the first row a 3D chair which is gradually becoming a human face image, with the outline of the chair gradually becoming the eyes of the human. These results demonstrate that the Student module can implicitly capture the correlations between different regions of two data domains by exploring their joint latent space.

Figure 5: Image interpolation results by the Student module, considering KIAM-GAN as the Teacher under CelebA-Chair setting.

5.4. Learning Classification Tasks

In this section, we extend our model for the continual learning of classification tasks. We adopt the TFCL benchmark from [89], which is used for supervised learning. Similar to [89], we only consider a teacher module where each expert is implemented by a VAE model, while we also train a classifier for each expert. Therefore, each expert $\mathcal{A}_i$ in the teacher module consists of a VAE model $\{p_{\theta_i}(\mathbf{x} \mid \mathbf{z}), q_{\eta_i}(\mathbf{z} \mid \mathbf{x})\}$ and a classifier C_{δ_i}, where δ_i denotes the classifier's parameters of the i-th expert, implemented by a ResNet-18 network.

We train the current classifier and the VAE using the samples drawn from the memory buffer. We also use the proposed Knowledge Incremental Assimilation Mechanism (KIAM) to dynamically add new experts during the training. In the testing phase, the VAE is used for the expert selection by comparing the sample log-likelihood for all Teacher experts. We employ ResNet 18 [52] as the classifier for Split CIFAR10 and Split CIFAR100, and a fully connected network with two hidden layers of 400 units for Split MNIST [81]. The maximum memory sizes for Split MNIST, Split CIFAR10, and Split CIFAR100 are 2000, 1000 and 5000, respectively. We adopt the same setting and datasets from [89, 81] and report the results in Table 4, where the performance of all baselines are taken from [89, 81]. These results show that our model is applicable for the classification task with better performance than other approaches.

5.5. Lifelong Disentangled Representation

In this section, we extend the proposed framework to lifelong disentangled representation learning where we employ the objective function from Eq. (12) for the Student module. We train the proposed framework on a data stream consisting of data samples from CelebA and 3D-chair datasets. After finishing all training

Table 4: Classification accuracy of five independent runs for various models on three datasets.

Methods	**Split MNIST**	**Split CIFAR10**	**Split CIFAR100**
finetune	19.75 ± 0.05	18.55 ± 0.34	3.53 ± 0.04
GEM	93.25 ± 0.36	24.13 ± 2.46	11.12 ± 2.48
iCARL	83.95 ± 0.21	37.32 ± 2.66	10.80 ± 0.37
reservoir	92.16 ± 0.75	42.48 ± 3.04	19.57 ± 1.79
MIR	93.20 ± 0.36	42.80 ± 2.22	20.00 ± 0.57
GSS	92.47 ± 0.92	38.45 ± 1.41	13.10 ± 0.94
CoPE-CE	91.77 ± 0.87	39.73 ± 2.26	18.33 ± 1.52
CoPE	93.94 ± 0.20	48.92 ± 1.32	21.62 ± 0.69
ER + GMED†	82.67 ± 1.90	34.84 ± 2.20	20.93 ± 1.60
ER_a + GMED†	82.21 ± 2.90	47.47 ± 3.20	19.60 ± 1.50
CURL	92.59 ± 0.66	-	-
CNDPM	93.23 ± 0.09	45.21 ± 0.18	20.10 ± 0.12
Dynamic-OCM	94.02 ± 0.23	49.16 ± 1.52	21.79 ± 0.68
KIAM-VAE	**95.78** ± 0.27	**53.98** ± 1.27	**26.92** ± 1.17

steps, we employ the Student module to map an image $\mathbf{x}$ to a latent vector $\mathbf{z}$ and only change one dimension of $\mathbf{z}$ from -3 to 3 while fixing the other dimensions. The resulting latent code $\mathbf{z}$ is fed into the decoder of the Student for the image reconstruction. We provide the visual results in Fig. 6, which shows that the Student module can discover four disentangled representations for two different data domains under TFCL.

5.6. Extension to Task-Aware Continual Learning

Recent research has investigated leveraging pre-trained Vision Transformer (ViT) models to improve generalization performance in continual learning scenarios [90, 91, 92]. However, these approaches are limited to task-aware continual learning settings. In this section, we extend the proposed framework to the task-aware continual learning paradigm by utilizing a pre-trained ViT as the core backbone for expert construction. Specifically, we adopt the ViT model from [93] within our framework. Each expert is implemented using a Variational Autoencoder (VAE) to capture the underlying data distribution, coupled with a classifier to learn discriminative features for prediction. The VAE functions as the expert selector during inference. To maximize the representational capacity of the pre-trained ViT backbone, we propose a straightforward yet effective incremental update mechanism that calibrates the predictions of each historical expert during backbone updates. This involves updating only the final three representation layers of the ViT to minimize computational overhead. Let F_{β_i} denote the ViT

(a) Morphing a chair image into a face image.

(b) Changing chair colour.

(c) Changing chair style.

Figure 6: Lifelong disentangled representation results achieved by the Student module.

backbone updated at the i-th task. We define E_{θ_j} as the classifier associated with the j-th expert. Upon completion of the j-th task, we duplicate the three trainable representation layers of the ViT backbone to form an auxiliary network $F_{\beta_i^\star}$, which is frozen to preserve previously learned representations for subsequent tasks. The current active ViT and the auxiliary network serve as the primary backbones for each historical expert. Additionally, we introduce a novel regularization loss that penalizes significant parameter shifts in the ViT backbone, formulated as :

$$\mathcal{L}_r = \frac{1}{j+1} \sum_{c=1}^{j+1} \left\{ F_d(F_{\beta_{j+1}}(E_{\theta_c}(\mathbf{x})), F_{\beta_j^\star}(E_{\theta_c}(\mathbf{x}))) \right\}, \tag{42}$$

where $F_d(\cdot, \cdot)$ represents the Mean Squared Error (MSE) loss metric. The ultimate objective function for training the current active expert's classifier is formulated as :

$$\mathcal{L}_{total} = \mathcal{L}_{CE} + \lambda_r \mathcal{L}_r \,, \tag{43}$$

where λ_r is a hyperparameter that modulates the weight of the regularization penalty, and $\mathcal{L}_{CE}$ denotes the cross-entropy loss function.

We enhance the proposed KIAM architecture for classification tasks utilizing the identical training protocols outlined in [93]. Tab. 5 presents the classification accuracy metrics across different models. Empirical evaluations demonstrate that the KIAM framework consistently outperforms alternative baseline models. Additional classification performance results have been incorporated into the revised version.

Table 5: The average accuracy calculated by various models on complex continual learning benchmarks. The results of baselines are taken from [93].

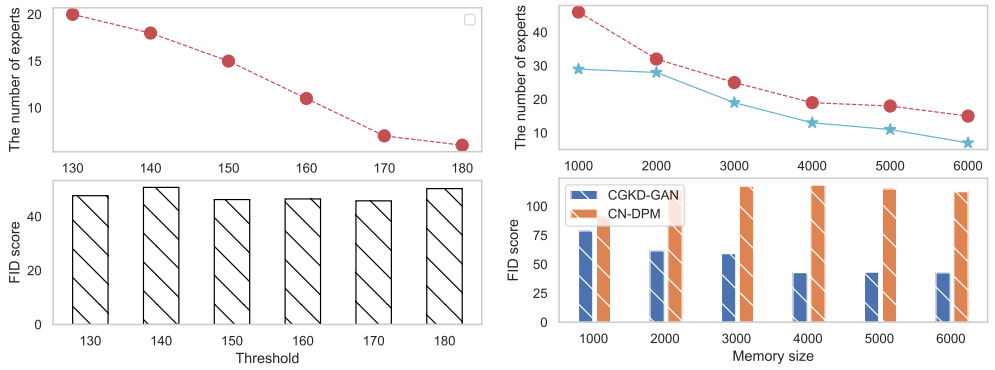
Methods	ImageNet-R	CUB200	Cars
DualPrompt [94]	71.00	79.50	40.10
RanPAC [93]	77.90	90.30	77.50
L2P [95]	72.40	65.20	38.20
CODA-Prompt [96]	75.50	79.50	43.20
ADaM [97]	72.30	87.10	41.40
KIAM	**77.92**	**91.02**	**80.12**

5.7. Ablation Study

In this section, we perform a full ablation study to investigate the performance of the proposed framework under various configurations.

The analysis when changing ν. We investigate the performance of the proposed KIAM-GAN when varying the threshold ν in Eq. (3). We train the KIAM-GAN under the MSFIRC lifelong learning setting and the average FID score on all six testing datasets is provided in Fig. 7a. The value for ν represents a trade-off between model complexity and performance. Nevertheless, we observe from Fig. 7a that the proposed KIAM-GAN does not lead to significant changes in the performance when changing the threshold ν. A small value for ν tends to result in adding more experts to the Teacher module. This result shows that more experts do not lead to significantly greater performance gains because some of these experts would learn and generate similar data samples.

Changing the memory size. We examine the performance and number of experts for the proposed KIAM-GAN and CN-DPM with different memory configurations under the MSFIRC learning settings. We consider 1000, 2000, 3000, 4000, 5000 and 6000, respectively, as the maximum number of samples stored in memory. The results are provided in Fig. 7b. We can observe that both KIAM-GAN and CN-DPM gradually increase their number of experts as the maximum memory size decreases. A large memory buffer would also increase the performance and reduce the complexity of the model for KIAM-GAN as it can store more training samples. In contrast, by reducing the maximum memory size leads to learning more experts for KIAM-GAN. This result indicates that the proposed expansion mechanism can automatically increase the model's capacity to combat small-scale memory. The proposed KIAM-GAN achieves better results than CN-DPM, despite using fewer experts with different memory configurations.



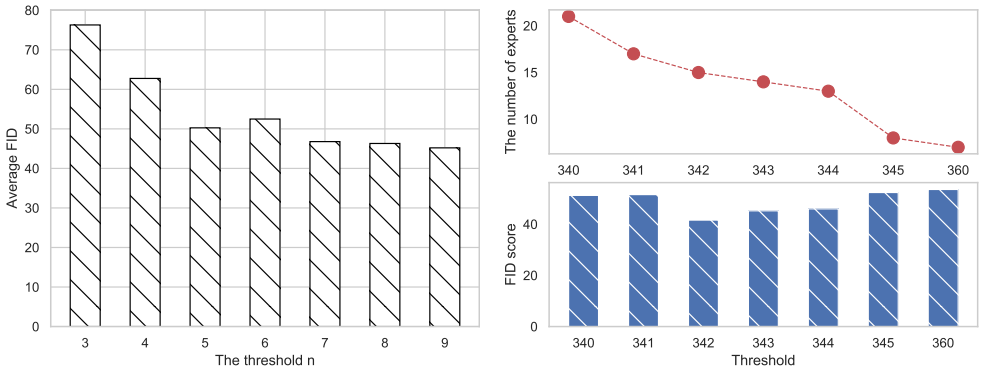
(a) Changing ν. (b) Different memory configurations.

Figure 7: Ablation results. (a) The performance and the number of Teacher experts when varying ν when learning MSFIRC. (b) The performance and the number of experts for KIAM-GAN and CN-DPM with different memory configurations under the MSFIRC learning setting.

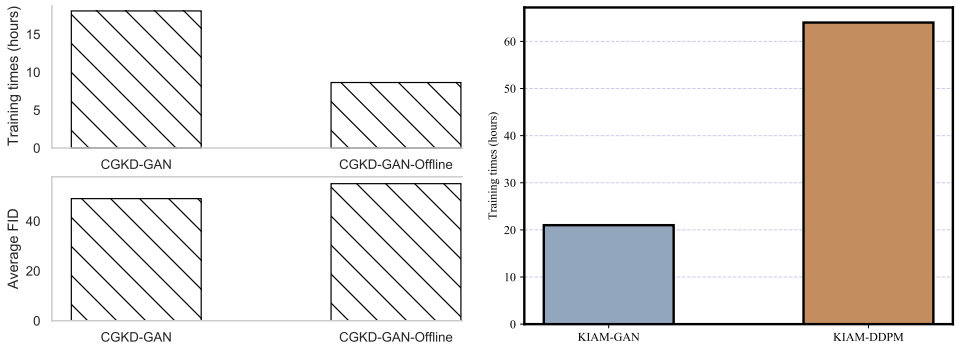
Assessing the number of experts $n_{\mathcal{A}}$. We investigate the impact of the number of experts $n_{\mathcal{A}}$ following the pruning of those having redundant knowledge, as discussed in Section 3.4, for the proposed KIAM*-GAN. We train KIAM*-GAN under the MSFIRC setting when considering different limits in the number of experts $n_{\mathcal{A}}$ and the results are shown in Fig. 8a. It can be seen that the proposed KIAM*-GAN would lose in its performance when $n_{\mathcal{A}}$ is very small. On the other hand, when $n_{\mathcal{A}}$ is equal to or larger than 5, the proposed KIAM*-GAN achieves a stable performance.

The consequences when varying λ_2 in KIAM-GAN-λ_2. We investigate the performance of the proposed KIAM-GAN-λ_2 when changing the threshold λ_2 in Eq. (10), for choosing the number of Teacher's experts, as detailed at the end of Section 3.4. We train KIAM-GAN-λ_2 under the MSFIRC learning setting with the different threshold λ_2 and the empirical results are shown in Fig. 8b. We observe from these results that a small threshold λ_2 leads to more experts for KIAM-GAN-λ_2 while a large threshold λ_2 leads to fewer experts for KIAM-GAN-λ_2. However, varying the threshold λ_2 in KIAM-GAN-λ_2 does not significantly change the final performance, as it can be observed in Fig. 8b. The Teacher module with less than five experts hardly can capture all underlying data distributions from the MSFIRC continual learning.

The assessment of the offline learning for the student module. We explore how we can further reduce the training time for KIAM-GAN without sacrificing performance. We create a baseline, called KIAM-GAN-Offline, which trains the Student only when the number of experts in the teacher module exceeds a certain number (6 in our experiments). In this way, KIAM-GAN-Offline updates the



(a) Changing $n_{\mathcal{A}}$. (b) Changing λ_2.



(c) The offline setting. (d) The training time.

Figure 8: Ablation results. (a) The performance for KIAM*-GAN when varying the number of experts $n_{\mathcal{A}}$ under the MSFIRC setting. (b) The performance for KIAM-GAN-λ_2 when changing the threshold λ_2 under the MSFIRC setting. (c) The results of KIAM-GAN and KIAM-GAN-Offline under the MSFIRC setting. (d) The training times (hours) required for the GAN- and DDPM-based teacher module.

teacher module only at certain training times thus reducing the overall computational cost. We train KIAM-GAN and KIAM-GAN-Offline with the same setting under MSFIRC and the empirical results are shown in Fig. 8c. From the results, KIAM-GAN-Offline can significantly reduce the total training time while the performance remains competitive with KIAM-GAN.

Changing the learning order of the data domains. We investigate the performance of various models when changing the learning order of data domains in a data stream $\mathcal{S}$. First, we consider creating a data stream consisting of Fashion, SVHN, IFashion, RMNIST, CIFAR10 and MNIST, namely FSIRCM. We train various models under FSIRCM and report the results in Table 6. These results show that the proposed KIAM outperforms other baselines when learning the data sequence FSIRCM with a suitable number of Teacher's components. We also consider a

Table 6: The FID of various models after the FSIRCM lifelong learning.

Methods	MNIST	SVHN	Fashion	IFashion	RMNIST	CIFAR10	Average	No
finetune	30.5	127.0	185.4	179.9	28.9	206.1	126.3	1
Reservoir [43]	5.7	213.8	137.1	216.2	5.0	197.5	129.2	1
LTS [10]	6.7	155.1	109.3	167.8	6.2	230.5	112.6	1
LGM [7]	4.4	213.2	164.7	227.8	4.0	192.4	134.4	1
CN-DPM [20]	6.5	76.3	99.8	112.6	5.6	128.7	71.6	26
KIAM-GAN	6.1	79.2	16.1	118.2	13.8	110.4	57.3	10
KIAM-VAE	4.1	75.5	94.5	139.7	3.7	135.8	75.6	11
KIAM*-GAN	5.0	60.6	15.1	139.7	8.6	98.1	**54.5**	7
KIAM*-VAE	4.3	77.0	75.6	101.7	3.9	109.1	61.9	7

Table 7: The FID of various models after the IFIMSC lifelong learning.

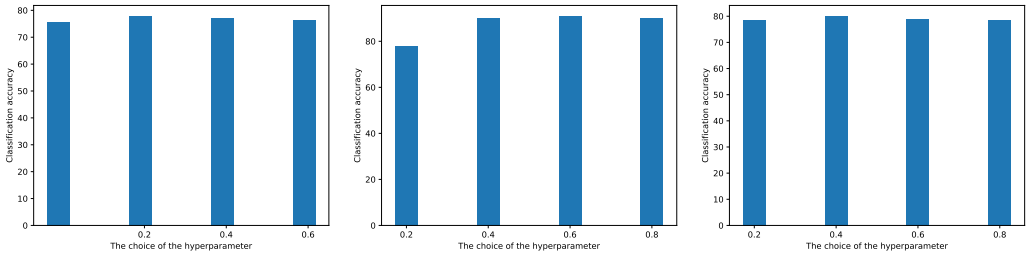
Methods	MNIST	SVHN	Fashion	IFashion	RMNIST	CIFAR10	Average	No
finetune	206.1	142.2	265.8	236.3	227.1	204.6	213.7	1
Reservoir [43]	185.5	120.8	223.9	193.5	191.1	110.4	170.9	1
LTS [10]	7.1	186.0	122.9	193.7	6.5	256.8	128.8	1
LGM [7]	215.7	109.2	250.7	202.2	223.0	119.6	186.7	1
CN-DPM [20]	89.7	87.2	135.9	125.4	103.7	98.6	106.7	26
KIAM-GAN	16.4	57.0	75.9	28.8	15.4	88.3	47.0	12
KIAM-VAE	58.8	64.3	96.3	93.9	38.8	97.3	74.9	10
KIAM*-GAN	10.5	49.0	33.2	20.2	9.7	76.2	**33.1**	7
KIAM*-VAE	112.5	77.9	140.2	123.3	104.0	90.4	108.0	7

data stream consisting of IMNIST, Fashion, IFashion, MNIST, SVHN and CIFAR10, namely IFIMSC. We report the results in Table 7. The proposed KIAM still outperforms other baselines under the IFIMSC setting. Together with the results from Tables 6 and 7, we show that the proposed KIAM is robust to changing the learning order of data domains in a data stream.

Exploring other expansion criteria. Instead of using the FID as the expansion signal $f_p(\cdot, \cdot)$ from Eq. (3), we consider a different dynamic expansion mechanism that employs the student module as a pre-trained model for checking the model expansion. Specifically, we consider that the student module can accumulate knowledge over time and can thus be used to evaluate the novelty of incoming samples. Once the Student finishes the training at a given time, we treat the student module as a pre-trained evaluator which aims to detect the data distribution shifts. Since the student module, implemented as a VAE, can estimate the sample log-likelihood, we can replace the FID metric for $f_p(\cdot, \cdot)$ in Eq. (3) by considering the

Table 8: FID for various models under the MSFIRC setting.

Methods	MNIST	SVHN	Fashion	IFashion	RMNIST	CIFAR10	Average	No
KIAM-GAN-StuLL	21.2	62.6	102.1	30.5	23.2	84.0	53.9	14
KIAM-VAE-StuLL	125.9	75.5	108.4	101.5	120.6	86.7	103.1	12



(a) ImageNet-R. (b) CUB200. (c) Cars.

Figure 9: The performance on ImageNet-R, CUB200 and Cars, achieved by the proposed framework with different λ_r configurations.

difference on the sample log-likelihood, as :

$$D_s(\mathrm{P}_{\theta_\mathrm{j}}, \mathrm{P}_{\mathcal{M}_\mathrm{i}}) = \mathbb{E}_{\mathbf{x}\sim\mathrm{P}_{\theta_\mathrm{j}}, \mathbf{x}'\sim\mathrm{P}_{\mathcal{M}_\mathrm{i}}} \left| \mathcal{L}_{ELBO}(\mathbf{x}; Stu) - \mathcal{L}_{ELBO}(\mathbf{x}'; Stu) \right| , \tag{44}$$

where $\mathcal{L}_{ELBO}(\mathbf{x}'; Stu)$ is the sample log-likelihood estimated by using the Student module and $|\cdot|$ denotes the absolute value. Therefore, the dynamic expansion criterion from Eq. (3) can be replaced by:

$$D_s(\mathrm{P}_{\theta_\mathrm{j}}, \mathrm{P}_{\mathcal{M}_\mathrm{i}}) \geq \nu \, . \tag{45}$$

Similar images usually tend to have close sample log-likelihood values and,zz therefore, the above measure can be used to evaluate the knowledge similarity between two distributions. Furthermore, the new dynamic expansion criterion does not use FID, which requires an extra pre-trained network and thus it is easier to implement. The results of the proposed framework using the new dynamic expansion criterion are provided in Tab. 8, where "KIAM-GAN-StuLL" denotes that the proposed KIAM-GAN uses the new dynamic expansion mechanism based on the Student log-likelihood evaluation. These results show that the proposed framework still performs similarly with OGKD-GAN and OGKD-VAE.

Training time. We evaluate the training times required for the proposed KIAM-GAN and KIAM-DDPM. Specifically, we train KIAM-GAN and KIAM-DDPM under the CelebA to 3D-Chair setting and the training times are reported in Fig. 8d.

From these results, we observe that the proposed KIAM-DDPM requires much more training time than KIAM-GAN. The primary reason is due to the considerable training costs from the DDPM model [48, 98]. Moreover, if we consider the generation process of the DDPM-based teacher module for the knowledge distillation procedure, instead of using a memory buffer M^i, as proposed in Section 3.6, would require significantly more training time.

The changes of the hyperparameter λ_r. To analyze the impact of hyperparameter λ_r on the performance of the proposed continual learning framework, we conduct experiments with various λ_r settings. The empirical results, depicted in Fig. 9, indicate that the framework attains optimal accuracy on ImageNet-R and Cars datasets when λ_r is set to 0.4. Conversely, the highest performance on the CUB200 dataset is observed when λ_r is configured at 0.6.

6. Conclusions and limitation

In this paper, we propose a new framework for task-agnostic lifelong generative modeling after learning from several different data domains without forgetting. Unlike existing memory-based methods, which employ a single memory buffer [18], the key idea of the proposed framework is to manage a two-memory system to store short- and long-term information. In order to implement this goal, a memory buffer with a fixed capacity is used to store the more recent data samples while the long-term memory system is implemented using a dynamic expansion memory buffer system. In order to learn data representations, we treat the long-term memory as a Teacher module as part of a unified framework that transfers its knowledge to a Student module. However, expanding the Teacher's capacity remains challenging under task-free continual learning due to the lack of task information. To solve this issue, this study introduces the Knowledge Incremental Assimilation Mechanism (KIAM) to progressively increase the Teacher's knowledge through a dynamic expansion mechanism. The resulting model eventually achieves a minimal number of parameters following the implementation of an expert pruning approach that automatically removes unimportant experts from the Teacher module. This mechanism also promotes knowledge diversity among experts and reduces computational costs for knowledge distillation.

Below, we outline several limitations inherent to the proposed teacher-student framework. Firstly, the computational overhead associated with KIAM remains substantial, particularly due to the dynamic expansion mechanism, which necessitates frequent monitoring of data distribution shifts throughout training. To mitigate this, a promising direction would involve substituting the current generation

process with a compact memory buffer that retains a select set of pivotal historical samples. This strategy will be investigated in subsequent research. Secondly, the expert pruning procedure within KIAM incurs significant computational expenses, primarily due to the repeated computation of inter-expert knowledge similarity. Addressing this challenge may be feasible by devising a more efficient knowledge similarity assessment method, leveraging low-dimensional feature representations rather than high-dimensional data samples. Additionally, the reliance on DDPM as the generative model introduces further computational demands during sample generation. To overcome this, future work will examine the adoption of more efficient generative architectures, such as the latent diffusion model [62], which has demonstrated effectiveness in accelerated sample synthesis.

The proposed lifelong learning framework can only address a sequence of datasets that belong to the same application. For instance, an artificial intelligence system should be able to continually deal with different applications, including classification, object detection, generation and language processing tasks, like in a continual federated learning system. Such a system should be able to infer knowledge from any sort of different tasks after lifelong learning. However, the main challenge for such systems is that different applications require different network architecture designs and requirements, while existing methods fail to address such challenges.

References

[1] Y. Bengio, I. Goodfellow, A. Courville, Deep learning, Vol. 1, MIT press Cambridge, MA, USA, 2017.

[2] G. I. Parisi, R. Kemker, J. L. Part, C. Kanan, S. Wermter, Continual lifelong learning with neural networks: A review, Neural Networks 113 (2019) 54–71.

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

[4] D. P. Kingma, M. Welling, Auto-encoding variational Bayes, arXiv preprint arXiv:1312.6114 (2013).

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

[6] A. Achille, T. Eccles, L. Matthey, C. Burgess, N. Watters, A. Lerchner, I. Higgins, Life-long disentangled representation learning with cross-domain latent homologies,

in: Advances in Neural Information Processing Systems (NeurIPS), 2018, pp. 9873–9883.

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

[8] M. Rostami, S. Kolouri, P. K. Pilly, J. McClelland, Generative continual concept learning, in: Proc. AAAI Conf. on Artificial Intelligence, 2020, pp. 5545–5552.

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

[10] F. Ye, A. G. Bors, Lifelong teacher-student network learning, IEEE Transactions on Pattern Analysis and Machine Intelligence 44 (10) (2022) 6280–6296.

[11] F. Ye, A. G. Bors, Lifelong infinite mixture model based on knowledge-driven Dirichlet process, in: Proc. of the IEEE/CVF Int. Conf. on Computer Vision (ICCV), 2021, pp. 10695–10704.

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

[13] J. Bang, H. Kim, Y. Yoo, J.-W. Ha, J. Choi, Rainbow memory: Continual learning with a memory of diverse samples, in: Proc. of IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2021, pp. 8218–8227.

[14] Z. Mai, R. Li, J. Jeong, D. Quispe, H. Kim, S. Sanner, Online continual learning in image classification: An empirical survey, Neurocomputing 469 (2022) 28–51.

[15] S. Ebrahimi, F. Meier, R. Calandra, T. Darrell, M. Rohrbach, Adversarial continual learning, in: Proc. European Conference on Computer Vision (ECCV), vol. LNCS 12356, Vol. 12356, 2020, pp. 386–402.

[16] G. Jerfel, E. Grant, T. Griffiths, K. A. Heller, Reconciling meta-learning and continual learning with online mixtures of tasks, in: Advances in Neural Information Processing Systems, 2019, pp. 9122–9133.

[17] Y. Wen, D. Tran, 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.

[18] R. Aljundi, K. Kelchtermans, T. Tuytelaars, Task-free continual learning, in: Proc. of IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR), 2019, pp. 11254–11263.

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

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

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

[22] A. Oring, Z. Yakhini, Y. Hel-Or, Autoencoder image interpolation by shaping the latent space, in: M. Meila, T. Zhang (Eds.), Proc. of the International Conference on Machine Learning (ICML), vol. PMLR 139, 2021, pp. 8281–8290.

[23] F. Ye, 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.

[24] F. Ye, A. G. Bors, Lifelong generative modelling using dynamic expansion graph model, in: Proc. AAAI on Artificial Intelligence, 2022, pp. 8857–8865.

[25] F. Ye, A. G. Bors, Dynamic self-supervised teacher-student network learning, IEEE Transactions on Pattern Analysis and Machine Intelligence 45 (5) (2023) 5731–5748.

[26] J. Yoon, D. Madaan, E. Yang, S. J. Hwang, Online coreset selection for rehearsal-based continual learning, in: International Conference on Learning Representations (ICLR), arXiv preprint arXiv:2106.01085, 2022.

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

[28] G. Hinton, O. Vinyals, J. Dean, Distilling the knowledge in a neural network, in: Proc. NIPS Deep Learning Workshop, arXiv preprint arXiv:1503.02531, 2014.

[29] C. Fernando, D. Banarse, C. Blundell, Y. Zwols, D. Ha, A. A. Rusu, A. Pritzel, D. Wierstra, PathNet: Evolution channels gradient descent in super neural networks, arXiv preprint arXiv:1701.08734 (2017).

[30] S. Golkar, M. Kagan, K. Cho, Continual learning via neural pruning, in: NeurIPS Workshop on Neuro AI, arXiv preprint arXiv:1903.04476, 2019.

[31] F. Ye, A. G. Bors, Lifelong mixture of variational autoencoders, IEEE Transactions on Neural Networks and Learning Systems 34 (1) (2023) 461–474.

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

[33] 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, R. Hadsell, Overcoming catastrophic forgetting in neural networks, Proc. of the National Academy of Sciences (PNAS) 114 (13) (2017) 3521–3526.

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

[35] C. Cortes, X. Gonzalvo, V. Kuznetsov, M. Mohri, 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.

[36] W. Liu, F. Zhu, L. Wei, Q. Tian, C-clip: Multimodal continual learning for vision-language model, in: The Thirteenth International Conference on Learning Representations, 2025.

[37] L. Jiao, L. Cao, T. Wang, Prompt-based continual learning for extending pretrained clip models' knowledge, in: Proceedings of the 6th ACM International Conference on Multimedia in Asia, 2024, pp. 1–8.

[38] M. Phuong, C. Lampert, Towards understanding knowledge distillation, in: International Conference on Machine Learning (ICML), vol. 97, 2019, pp. 5142–5151.

[39] G. Nam, J. Yoon, Y. Lee, J. Lee, Diversity matters when learning from ensembles, Advances in Neural Information Processing Systems 34 (2021).

[40] Z. Li, D. Hoiem, Learning without forgetting, IEEE Trans. on Pattern Analysis and Machine Intelligence 40 (12) (2017) 2935–2947.

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

[42] P. Buzzega, M. Boschini, A. Porrello, D. Abati, S. Calderara, Dark experience for general continual learning: a strong, simple baseline, in: Advances in Neural Information Processing Systems (NeurIPS), 2020, pp. 15920–15930.

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

[44] E. Egorov, A. Kuzina, E. Burnaev, BooVAE: Boosting approach for continual learning of VAE, Advances in Neural Information Processing Systems (NeurIPS) 34 (2021).

[45] F. Ye, A. G. Bors, Lifelong twin generative adversarial networks, in: Proc. IEEE Int. Conf. on Image Processing (ICIP), 2021, pp. 1289–1293.

[46] M. Banayeeanzade, R. Mirzaiezadeh, H. Hasani, M. Soleymani, Generative vs. discriminative: Rethinking the meta-continual learning, Advances in Neural Information Processing Systems 34 (2021).

[47] A. Floyer-Lea, P. M. Matthews, Distinguishable brain activation networks for short- and long-term motor skill learning, Journal of neurophysiology 94 (1) (2005) 512–518.

[48] J. Ho, A. Jain, P. Abbeel, Denoising diffusion probabilistic models, Advances in Neural Information Processing Systems 33 (2020) 6840–6851.

[49] M. Heusel, H. Ramsauer, T. Unterthiner, B. Nessler, 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.

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

[51] F. Ye, A. G. Bors, Continual variational autoencoder via continual generative knowledge distillation, in: Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 37, 2023, pp. 10918–10926.

[52] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: Proc. of IEEE Conf. on Computer Vision and Pattern Recognition (CVPR), 2016, pp. 770–778.

[53] G. Desjardins, A. Courville, Y. Bengio, Disentangling factors of variation via generative entangling, arXiv preprint arXiv:1210.5474 (2012).

[54] I. Higgins, L. Matthey, A. Pal, C. Burgess, X. Glorot, M. Botvinick, S. Mohamed, A. Lerchner, β-VAE: Learning basic visual concepts with a constrained variational framework, in: International Conference on Learning Representations (ICLR), 2017.

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

[56] J. Rissanen, Modeling by shortest data description, Automatica 14 (5) (1978) 465–471.

[57] F. Ye, A. G. Bors, Lifelong dual generative adversarial nets learning in tandem, IEEE Transactions on Cybernetics 54 (3) (2024) 1353–1365.

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

[59] H. Li, Y. Yang, M. Chang, S. Chen, H. Feng, Z. Xu, Q. Li, Y. Chen, Srdiff: Single image super-resolution with diffusion probabilistic models, Neurocomputing 479 (2022) 47–59.

[60] J. Ho, C. Saharia, W. Chan, D. J. Fleet, M. Norouzi, T. Salimans, Cascaded diffusion models for high fidelity image generation., J. Mach. Learn. Res. 23 (47) (2022) 1–33.

[61] P. Dhariwal, A. Nichol, Diffusion models beat gans on image synthesis, Advances in Neural Information Processing Systems 34 (2021) 8780–8794.

[62] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, B. Ommer, High-resolution image synthesis with latent diffusion models, in: Proc. of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2022, pp. 10684–10695.

[63] Z. Wang, H. Zheng, P. He, W. Chen, M. Zhou, Diffusion-GAN: Training GANs with diffusion, in: International Conference on Learning Representations (ICLR), arXiv preprint arXiv:2206.02262, 2023.

[64] A. Karnewar, A. Vedaldi, D. Novotny, N. J. Mitra, Holodiffusion: Training a 3D diffusion model using 2D images, in: Proc. of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2023, pp. 18423–18433.

[65] S. Chen, P. Sun, Y. Song, P. Luo, DiffusionDet: diffusion model for object detection, in: Proc. of the IEEE/CVF Int., Conference on Computer Vision (ICCV), 2023, pp. 19830–19843.

[66] Y. Song, J. Sohl-Dickstein, D. P. Kingma, A. Kumar, S. Ermon, B. Poole, Score-based generative modeling through stochastic differential equations, in: International Conference on Learning Representations (ICLR), arXiv preprint arXiv:2011.13456, 2021.

[67] C. Niu, Y. Song, J. Song, S. Zhao, A. Grover, S. Ermon, Permutation invariant graph generation via score-based generative modeling, in: Proc. International Conference on Artificial Intelligence and Statistics (AISTATS), PMLR 108, 2020, pp. 4474–4484.

[68] S. Gu, D. Chen, J. Bao, F. Wen, B. Zhang, D. Chen, L. Yuan, B. Guo, Vector quantized diffusion model for text-to-image synthesis, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2022, pp. 10696–10706.

[69] G. Kim, T. Kwon, J. C. Ye, Diffusionclip: Text-guided diffusion models for robust image manipulation, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2022, pp. 2426–2435.

[70] R. Cai, G. Yang, H. Averbuch-Elor, Z. Hao, S. Belongie, N. Snavely, B. Hariharan, Learning gradient fields for shape generation, in: Proc. European Conference on Computer Vision (ECCV), vol. LNCS 1234, 2020, pp. 364–381.

[71] I. Goodfellow, Nips 2016 tutorial: Generative adversarial networks, arXiv preprint arXiv:1701.00160 (2016).

[72] W. Feller, On the theory of stochastic processes, with particular reference to applications, in: Proc. of the First Berkeley Symposium on Mathematical Statistics and Probability, 1949, pp. 403–432.

[73] J. Sohl-Dickstein, E. Weiss, N. Maheswaranathan, S. Ganguli, Deep unsupervised learning using nonequilibrium thermodynamics, in: International Conference on Machine Learning (ICML), PMLR 37, 2015, pp. 2256–2265.

[74] Y. Burda, R. Grosse, R. Salakhutdinov, Importance weighted autoencoders, arXiv preprint arXiv:1509.00519 (2015).

[75] J. Domke, D. R. Sheldon, Importance weighting and variational inference, in: Advances in Neural Information Processing Systems (NeurIPS), 2018, pp. 4470–4479.

[76] F. Ye, A. G. Bors, K. Zhang, Dynamic expansion diffusion learning for lifelong generative modelling, in: Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 39, 2025, pp. 22101–22109.

[77] Y. LeCun, L. Bottou, Y. Bengio, P. Haffner, Gradient-based learning applied to document recognition, Proc. of the IEEE 86 (11) (1998) 2278–2324.

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

[79] H. Xiao, K. Rasul, R. Vollgraf, Fashion-MNIST: a novel image dataset for benchmarking machine learning algorithms, arXiv preprint arXiv:1708.07747 (2017).

[80] Y. Abouelnaga, O. S. Ali, H. Rady, M. Moustafa, Cifar-10: Knn-based ensemble of classifiers, in: Proc. International Conference on Computational Science and Computational Intelligence (CSCI), 2016, pp. 1192–1195.

[81] M. De Lange, T. Tuytelaars, Continual prototype evolution: Learning online from non-stationary data streams, in: Proc. of the IEEE/CVF International Conference on Computer Vision (ICCV), 2021, pp. 8250–8259.

[82] F. Ye, A. G. Bors, Lifelong generative adversarial autoencoder, IEEE Transactions on Neural Networks and Learning Systems (2023) 1–15.
URL doi:10.1109/TNNLS.2023.3281091

[83] A. Q. Nichol, P. Dhariwal, Improved denoising diffusion probabilistic models, in: Proc. of the International Conference on Machine Learning (ICML), vol. PMLR 139, 2021, pp. 8162–8171.

[84] O. Ronneberger, P. Fischer, T. Brox, U-net: Convolutional networks for biomedical image segmentation, in: Medical image computing and computer-assisted intervention–MICCAI 2015: 18th international conference, Munich, Germany, October 5-9, 2015, proceedings, part III 18, Springer, 2015, pp. 234–241.

[85] S. Zagoruyko, N. Komodakis, Wide residual networks, arXiv preprint arXiv:1605.07146 (2016).

[86] D. P. Kingma, J. Ba, Adam: A method for stochastic optimization, in: International Conference on Learning Representations (ICLR), arXiv preprint arXiv:1412.6980, 2015.

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

[88] M. Aubry, D. Maturana, A. A. Efros, B. C. Russell, 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.

[89] F. Ye, A. G. Bors, Continual variational autoencoder learning via online cooperative memorization, in: Proc. European Conference on Computer Vision (ECCV), vol. LNCS 13683, 2022, pp. 531–549.

[90] D.-W. Zhou, H.-L. Sun, J. Ning, H.-J. Ye, D.-C. Zhan, Continual learning with pre-trained models: A survey, arXiv preprint arXiv:2401.16386 (2024).

[91] G. Zhang, L. Wang, G. Kang, L. Chen, Y. Wei, Slca: Slow learner with classifier alignment for continual learning on a pre-trained model, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2023, pp. 19148–19158.

[92] L. Wang, J. Xie, X. Zhang, H. Su, J. Zhu, Hide-pet: continual learning via hierarchical decomposition of parameter-efficient tuning, IEEE Transactions on Pattern Analysis and Machine Intelligence (2025).

[93] M. D. McDonnell, D. Gong, A. Parvaneh, E. Abbasnejad, A. v. d. Hengel, RanPAC: Random projections and pre-trained models for continual learning, in: Advances in Neural Information Processing Systems 36, article no. 526, 2023, pp. 12022–12053.

[94] Z. Wang, Z. Zhang, S. Ebrahimi, R. Sun, H. Zhang, C. Lee, X. Ren, G. Su, V. Perot, J. G. Dy, T. Pfister, Dualprompt: Complementary prompting for rehearsal-free continual learning, in: Proc. Eucopean Conference on Computer Vision (ECCV), vol. LNCS 13686, 2022, pp. 631–648.

[95] Z. Wang, Z. Zhang, C.-Y. Lee, H. Zhang, R. Sun, X. Ren, G. Su, V. Perot, J. Dy, T. Pfister, Learning to prompt for continual learning, in: Proc. of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2022, pp. 139–149.

[96] J. S. Smith, L. Karlinsky, V. Gutta, P. Cascante-Bonilla, D. Kim, A. Arbelle, R. Panda, R. Feris, Z. Kira, CODA-Prompt: continual decomposed attention-based prompting for rehearsal-free continual learning, in: Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR, 2023, pp. 11909–11919.

[97] D.-W. Zhou, Z.-W. Cai, H.-J. Ye, D.-C. Zhan, Z. Liu, Revisiting class-incremental learning with pre-trained models: Generalizability and adaptivity are all you need, International Journal of Computer Vision 133 (3) (2025) 1012–1032.

[98] J. Song, C. Meng, S. Ermon, Denoising diffusion implicit models, in: International Conference on Learning Representations (ICLR) arXiv preprint arXiv:2010.02502, 2021.