Lifelong Scalable Generative System via Online Maximum Mean Discrepancy

Fei Ye[1], **Adrian G. Bors**[2]

[1]School of Information and Software Engineering, University of Electronic Science and Technology of China, Chengdu
[2]Department of Computer Science, University of York, York YO10 5GH, UK
feiye@uestc.edu.cn, adrian.bors@york.ac.uk

Abstract

Diffusion-based models have been recently shown to be high-quality data generators. However, their performance severely degrades when training on non-stationary changing data distributions in an online manner, due to the catastrophic forgetting. In this paper, we propose enabling the diffusion model with a novel Dynamic Expansion Memory Unit (DEMU) methodology that adaptively creates new memory buffers, to be added to a memory system, in order to preserve information deemed critical for training the model. Having a selective memory unit is essential for training diffusion networks, which are expensive to train, especially when deployed in resource-constrained environments. A Maximum Mean Discrepancy (MMD) based expansion mechanism, that evaluates probabilistic distances between each of the previously defined memory buffers and the newly given data, and uses them as expansion signals, is employed for ensuring the diversity of information learning. We propose a new model expansion mechanism to automatically add new diffusion models as experts in a mixture system, which enhances the multi-domain image generation performance. Also a novel memory compaction approach is proposed to automatically remove statistically overlapping memory units, through a graph relationship evaluation, preventing the limitless expansion of DEMU. Comprehensive results show that the proposed approach performs better than the state-of-the-art.

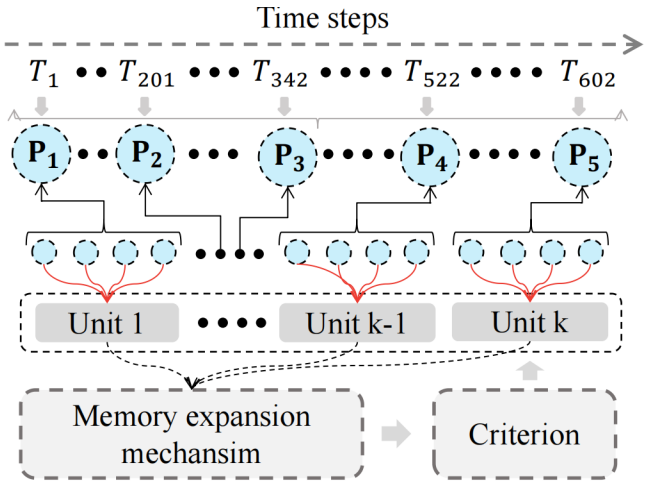


Figure 1: The core idea of the proposed DEMU. $\{\mathbf{P}_1, \cdots, \mathbf{P}_5\}$ represent five different underlying data distributions at different times $\{T_1, \cdots, T_{602}\}$. The DEMU dynamically creates new memory units, denoted as 'Unit 1,..., Unit k', to capture new distributions over time. This is achieved by using a memory expansion mechanism enabled by a criterion to decide when to add new memory units.

Introduction

In recent years, the Denoising Diffusion Generative Model (DDPM) (Ho, Jain, and Abbeel 2020) emerged as an increasingly popular approach used, due to its data representation properties, for image synthesis (Croitoru et al. 2023; Rombach et al. 2022). DDPM has been successfully employed in various applications, including image inpainting (Song et al. 2021), image super-resolution generation (Ho et al. 2022; Li et al. 2022), shape generation (Cai et al. 2020), graph generation (Niu et al. 2020), text-to-image generation (Gu et al. 2022a; Kim, Kwon, and Ye 2022) and object detection (Chen et al. 2023), producing remarkable results. However, the DDPM model performance highly relies on the availability of massive labelled data, which can not be easily obtained in certain real-time applications where data is not labelled and which is only available once at a time. Such a learning scenario is referred to as the Task-Free Unsupervised Continual Learning (TFUCL) (Rao et al. 2019). The goal of a TFUCL-based model is that of learning representations without having any task identity knowledge.

Recently, continual image generation has attracted a lot of attentions in several studies (Achille et al. 2018; Ramapuram, Gregorova, and Kalousis 2017; Ye and Bors 2023). However, these methods employ either Generative Adversarial Nets (GANs) (Goodfellow et al. 2014) or Variational Autoencoders (VAEs) (Kingma and Welling 2013) as generative models, while the DDPM has not been explored in continual unsupervised image generation. When compared to the GAN or VAE models, DDPM has superior capabilities of generating high-fidelity images following a stable training process (Rombach et al. 2022).

Learning a DDPM model under the TFUCL faces two primary challenges : catastrophic forgetting, and the data distribution shift when class and task information are not

known. To address the first challenge, an efficient approach is that of storing past data samples in memory buffers, (Jin et al. 2021). The sample selection for memory-based methods is usually implemented by using either gradient information (Aljundi et al. 2019b) or a *learner-evaluator* evaluation framework (De Lange and Tuytelaars 2021). However, most existing memory-based approaches require accessing the class information or the prediction by a classifier to implement sample selection mechanisms, which can not address the second challenge. In this paper, we simultaneously address those two TFUCL challenges by proposing a novel Dynamic Expansion Memory Unit (DEMU) approach, aiming to store diverse samples without having any information about either the task or class. The primary idea of DEMU is illustrated in Fig. 1, where $\{\mathbf{P}_1, \cdots, \mathbf{P}_5\}$ are different underlying data distributions, each one emerging at a different time $\{T_1, \cdots, T_{602}\}$. In order to model the distribution change over time, the proposed DEMU dynamically creates new and fixed-size memory units that can store groups of samples of similar semantic information to capture the new data distribution configurations. We implement this memory optimization process by comparing the discrepancy distance between the distributions of the new data and that of each existing memory unit. A large distance represents a good signal for creating a new memory unit to preserve the novel information from the data. Such a memory expansion mechanism ensures the preservation of diverse knowledge, benefiting the capturing of all given underlying data distributions while using a compact memory capacity. To implement this goal, we propose a novel memory expansion mechanism that uses the Maximum Mean Discrepancy (MMD) (Tolstikhin, Sriperumbudur, and Schölkopf 2016) criterion to evaluate the discrepancy distance on a pair of memory units. The MMD is efficiently computed in the embedding space and was used as a distance criterion between probabilistic representations in other models (Li et al. 2017). Furthermore, when deploying the model on a small device, it is necessary to restrict the overall memory capacity. To address this issue, we propose a novel memory reduction approach that formulates each memory unit as a node in a graph structure and evaluates a graph relation matrix to guide the removal of the statistically overlapping memory units. This approach can prevent the DEMU from growing forever while maintaining the knowledge diversity among the memory units.

The proposed DEMU approach enables diffusion-based image generation under the Task-Free Unsupervised Continual Learning (TFUCL) paradigm. A new dynamic model expansion mechanism, which adaptively builds new DDPM models as experts in a mixture system, is proposed for dynamically increasing the diffusion model's capacity when learning complex data statistics. The proposed mechanism evaluates expansion signals measured by the MMD distance between the information recorded by each expert and that of the incoming data batch, expanding the model whenever needed. We perform a series of experiments on unsupervised image generation, showing that DEMU significantly relieves the DDPM model forgetting.

The contributions of this research study are as follows: (1) We propose a new memory buffer management approach that dynamically preserves critical data samples through an MMD-based memory expansion mechanism. This memory approach is plug-and-play and can be used in different variants of the DDPM model; (2) A new memory reduction approach is proposed to prevent DEMU from expanding forever, enabling it to be deployed on resource-constrained devices; (3) A novel dynamic expansion mechanism for increasing the capacity of the DDPM model is also proposed; (4) We perform a series of experiments showing that the proposed methodology achieves state-of-the-art performance. The code is available [1]

Related Work

Continual Learning (CL). Managing a fixed-capacity memory buffer to store data samples has shown good performances in CL (Bang et al. 2022; Gu et al. 2022b; Guo, Liu, and Zhao 2022; Jha et al. 2024; Liang and Li 2024; Smith et al. 2023b; Tiwari et al. 2022; Villa et al. 2023). Regularization penalizing changes in certain important network parameters when learning a new task was shown to further improve the results (Deng et al. 2021; Egorov, Kuzina, and Burnaev 2021; Hurtado, Raymond, and Soto 2021; Wang et al. 2021). Training a generator network to learn and regenerate past data samples was also shown to be an efficient replay-based CL method. However, these methods perform well on simple datasets while facing significant performance degeneration when learning long sequences of tasks. This challenge is addressed by the Dynamic Expansion Model (DEM) that dynamically expands the model's capacity to deal with new tasks (Polikar et al. 2001; Rusu et al. 2016; Xiao et al. 2014; Zhou, Sohn, and Lee 2012). However, most DEM models rely on the availability of the task information and cannot be used in the more realistic TFUCL scenario, where the task information is missing.

Task-Free Continual Learning (TFCL). Employing a memory buffer was shown to be efficient in the TFCL, (Aljundi, Kelchtermans, and Tuytelaars 2019). The memory approach usually designs an appropriate sample selection criterion to store the most representative samples (Aljundi et al. 2019a). In (Aljundi et al. 2019b) the sample selection was performed by comparing the gradient information between new and past data sets, thus formulating CL as a constrained optimization problem. Furthermore, the sample selection process was implemented using a *learner-evaluator* evaluation framework (De Lange and Tuytelaars 2021), which can ensure storing balanced data sets from all categories. Moreover, sample editing, which modifies, the data used for training, in such a way that it increases the loss in the upcoming model updates was enacted on certain memorized samples. Furthermore, the memory approach can be combined with a model expansion mechanism in (Lee et al. 2020) to further improve performance. Although these methods provide promising results in TFCL, they can not be used in unsupervised learning.

Continual generative modeling. Training generative models under CL has been studied recently, (Achille et al. 2018; Ramapuram, Gregorova, and Kalousis 2017). The study

[1] https://github.com/dtuzi123/DEMU

from Achille et al. (2018) introduces a VAE-based framework that learns shared and task-specific latent representations over time. Continual generative modeling was implemented by using a teacher-student framework (Ramapuram, Gregorova, and Kalousis 2017) or a GAN-based model (Wu et al. 2018). However, these approaches require knowing the task boundaries, which can not be considered in TFUCL. This issue was addressed by the dynamic expansion model from (Ye and Bors 2023), which introduces a Fréchet Inception Distance (FID)-based model expansion mechanism without requiring any supervision signals. However, this expansion criterion relies on the extra knowledge stored in a pre-trained network, while our approach does not need either such knowledge or pre-trained models.

Methodology

Preliminary

In this paper, we focus on unsupervised image generation tasks under Task-Free Unsupervised Continual Learning (TFUCL). Let $\mathcal{D}_i^S = \{\mathbf{x}_j\}_{j=1}^{N_i^S}$ and $\mathcal{D}_i^T = \{\mathbf{x}_j\}_{j=1}^{N_i^T}$ be the i-th unlabeled training and testing datasets containing N_i^S and N_i^T samples, respectively. In the class-incremental learning, we consider a data stream $\mathcal{A}$ incorporating sequentially the training subsets, $\mathcal{A} = \{\mathcal{D}_{i,1}^S, \mathcal{D}_{i,2}^S, \cdots, \mathcal{D}_{i,c}^S\}$, originating from one or several classes, where c is the total number of subsets. Under the online learning paradigm, training a model on $\mathcal{A}$ is divided into n time intervals $\{\mathcal{T}_1, \cdots, \mathcal{T}_n\}$. During each time interval $\mathcal{T}_j$, the model can only see once a small batch of b samples ($\mathbf{x}_j$). Besides the class-incremental learning we also study the dataset-incremental setting in which $\mathcal{A}$ involves both domain and class shifts over time, expressed by $\mathcal{A} = \{\mathcal{D}_{1,1}^S, \mathcal{D}_{1,2}^S, \cdots, \mathcal{D}_{t,c}^S\}$, where t represents the total number of datasets. The model's performance is evaluated using image generation performance after finishing its learning, at each time $\{T_1, \cdots, T_n\}$, on all testing datasets $\{\mathcal{D}_1^T, \cdots, \mathcal{D}_t^T\}$, .

Denoising Diffusion Generative Model (DDPM): DDPM (Ho, Jain, and Abbeel 2020), consists of two independent processes of iterative series of optimizations. Firstly, the forward diffusion process gradually adds noise to the data to produce a sequence of noisy samples $\{\tilde{\mathbf{x}}_1, \cdots, \tilde{\mathbf{x}}_T\}$, as :

$$q(\tilde{\mathbf{x}}_{1:T} \,|\, \tilde{\mathbf{x}}_0) := \prod_{j=1}^{T} q(\tilde{\mathbf{x}}_j \,|\, \tilde{\mathbf{x}}_{j-1}) \,, \tag{1}$$

where T is the total number of diffusion steps, and $q(\tilde{\mathbf{x}}_j \,|\, \tilde{\mathbf{x}}_{j-1}) := \mathcal{N}(\tilde{\mathbf{x}}_j; \sqrt{1-\beta_j}\tilde{\mathbf{x}}_{j-1}, \beta_j \mathbf{I})$. $\tilde{\mathbf{x}}_0$ is a real training sample drawn from a certain data distribution $p(\tilde{\mathbf{x}}_0)$. $\{\beta_j \in (0,1) \,|\, j = 1, \cdots, T\}$ is a hyperparameter that controls the diffusion step size. A sufficiently small β_j can ensure that $q(\tilde{\mathbf{x}}_{j-1} \,|\, \tilde{\mathbf{x}}_j)$ follows a Gaussian distribution. The forward diffusion process transforms the data distribution $p(\tilde{\mathbf{x}}_0)$ to become as close as possible to a Gaussian distribution $\mathcal{N}(\mathbf{0}, \mathbf{I})$, when a large diffusion step T is considered.

During the second processing stage of the DDPM, after a real image is converted into a noise vector, the backward diffusion process recovers the data from the noisy samples. However, estimating the distribution $q(\tilde{\mathbf{x}}_{j-1} \,|\, \tilde{\mathbf{x}}_j)$ in the backward diffusion is challenging requiring the entire dataset. The DDPM solves this issue by learning a model $p_\theta(\tilde{\mathbf{x}}_{j-1} \,|\, \tilde{\mathbf{x}}_j)$ parameterized by θ, and then the noise image is repeatedly refined through :

$$p_\theta(\tilde{\mathbf{x}}_{0:T}) := p(\tilde{\mathbf{x}}_T) \prod_{j=1}^{T} p_\theta(\tilde{\mathbf{x}}_{j-1} \,|\, \tilde{\mathbf{x}}_j) \,, \tag{2}$$

where $p_\theta(\tilde{\mathbf{x}}_{j-1} \,|\, \tilde{\mathbf{x}}_j) := \mathcal{N}(\tilde{\mathbf{x}}_{j-1}; \boldsymbol{\mu}_\theta(\tilde{\mathbf{x}}_j, j), \boldsymbol{\Sigma}_\theta(\tilde{\mathbf{x}}_j, j))$. $\boldsymbol{\Sigma}_\theta(\cdot,\cdot)$ and $p(\tilde{\mathbf{x}}_T) = \mathcal{N}(\mathbf{0}, \mathbf{I})$ represents a Normal distribution. $\boldsymbol{\mu}_\theta(\cdot,\cdot)$, $\boldsymbol{\Sigma}_\theta(\cdot,\cdot)$ are trainable functions implemented by deep neural networks. Training the DDPM uses a simple objective function leading to a stable learning procedure, as :

$$\begin{aligned} \mathcal{L}_{\text{DDPM}}(\theta) := \mathbb{E}_{j, \tilde{\mathbf{x}}_0, \boldsymbol{\epsilon}} \big[\| \boldsymbol{\epsilon} - \boldsymbol{\epsilon}_\theta(\sqrt{\hat{\alpha}_j} \tilde{\mathbf{x}}_0 \\ + \sqrt{1-\hat{\alpha}_j} \boldsymbol{\epsilon}, j) \|^2 \big] \,, \end{aligned} \tag{3}$$

where $\alpha_j := 1 - \beta_j$ and $\hat{\alpha}_j := \prod_{s=1}^{j} \{\alpha_s\}$. $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ is a random vector and $\boldsymbol{\epsilon}_\theta(\cdot,\cdot)$ is implemented using a deep neural network that predicts $\boldsymbol{\epsilon}$ for the given $\tilde{\mathbf{x}}_j$ and j.

The Dynamic Expansion Memory Unit (DEMU)

Most memory-buffer based methods rely heavily on knowing the task identity or class information (Bang et al. 2022; Gu et al. 2022b; Tiwari et al. 2022; Smith et al. 2023a); however, these are actually not available in the unsupervised image generation under TFUCL. In the following we introduce the Dynamic Expansion Memory Unit (DEMU), which enables the learning of a DDPM model on a non-stationary data distribution without accessing any supervised signals. The key idea for the DEMU is to continually build several memory units over time, each storing sufficiently different information with respect to the others. Suppose that the DEMU has already created k memory units $\mathcal{M} = \{\mathcal{M}_1, \cdots, \mathcal{M}_k\}$ at T_i, where each $\mathcal{M}_j$, $j = 1, \cdots, k$ has a fixed capacity of maximum γ samples. To ensure an appropriate discrepancy between the memory units during the memory expansion process, we compare the information distance between each stored memory unit $\mathcal{M}_j$, and the existing data buffer $\mathcal{M}_k$, $k > j$, currently considered for learning, and use this measure to guide the memory expansion :

$$s = \min_{j=1}^{k-1} \{ f_d(\mathcal{M}_j, \mathcal{M}_k) \} \,, \tag{4}$$

where $f_d(\cdot,\cdot)$ is an information distance measure evaluating the knowledge similarity between two memory units, where $\mathcal{M}_k$ is the current memory buffer considered for learning. A large s in Eq. (4) indicates that the current memory buffer unit $\mathcal{M}_k$ has a large discrepancy in comparison to existing memory units. Consequently, we consider an efficient and effective memory expansion criterion, expressed as :

$$s > \lambda \,, \tag{5}$$

where $\lambda \in [0, 3]$ is a threshold controlling the memory expansion. If Eq. (5) is fulfilled, we freeze the current memory unit $\mathcal{M}_k$ to promote the overall knowledge diversity among the memory units while building a new memory buffer unit $\mathcal{M}_{k+1}$ for future learning. A small λ leads adding more memory units, while a large λ has the opposite effect.

The distance measure function $f_d(\cdot,\cdot)$ from Eq. (4) can be implemented by using the Fréchet Inception Distance

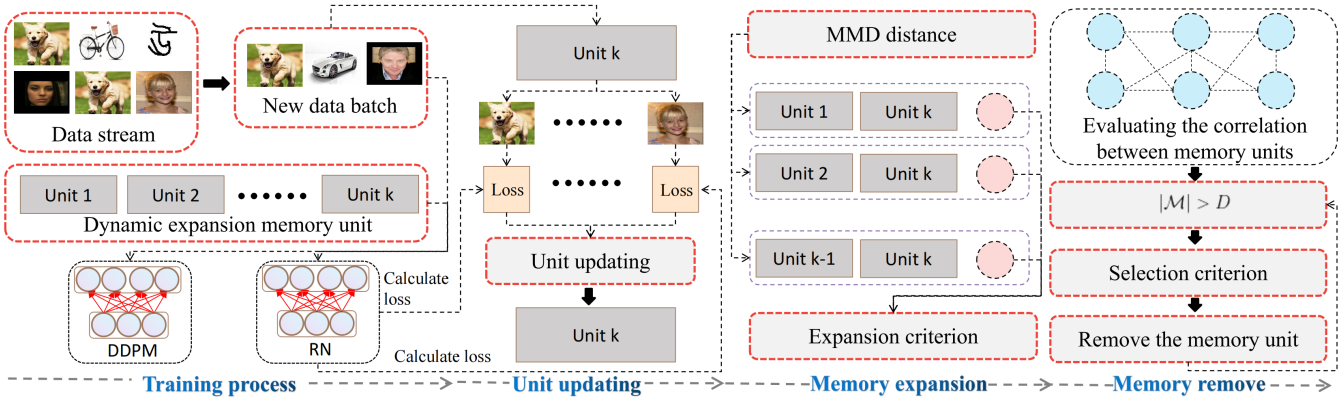


Figure 2: Static framework learning diagram consisting of a DDPM model $\boldsymbol{\epsilon}_\theta$, a representation network ('RN') $\{\text{enc}_\phi, \text{dec}_\eta\}$ and a memory $\mathcal{M}$, where 'Unit k' denotes $\mathcal{M}_k$. The first step is to update the DDPM model and the representation network on $\mathcal{M} \cup \mathbf{x}_i$ at T_i. In the second step, we perform the memory unit optimization to update the current memory unit $\mathcal{M}_k$ via Eq. (9). If the expansion criterion (Eq. (5)), in the third step is satisfied, then we freeze $\mathcal{M}_k$ and create a new memory unit $\mathcal{M}_{k+1}$ in the subsequent learning. The fourth step is to remove overlapped memory units when the size of $\mathcal{M}$ exceeds its maximum D.

score (FID), representing a symmetrical distance measure used for assessing the quality of generated images, (Heusel et al. 2017). The FID score can be efficiently calculated in the embedding space and it relies on the extra knowledge stored by a pre-trained network. In this paper, we propose an alternative approach to implement $f_d(\cdot)$ on the embedding space without requiring extra knowledge. Specifically, we consider an autoencoder, as a data representation network, consisting of an encoder enc_ϕ and a decoder dec_η, parameterized by parameters $\{\phi, \eta\}$, which is then trained on $\mathcal{M}$ using the reconstruction error loss, aiming to learn an embedding space in order to support the distance evaluation. As a result, we implement f_d in the embedding space using the Maximum Mean Discrepancy (MMD), expressed as :

$$f_d(\mathcal{M}_j,\mathcal{M}_k) = \frac{1}{|\mathcal{M}_j|(|\mathcal{M}_j|-1)} \sum_{a\neq g}^{|\mathcal{M}_j|} \Big\{ f(\mathbf{x}_a^j, \mathbf{x}_g^j) + f(\mathbf{x}_a^k, \mathbf{x}_g^k) - f(\mathbf{x}_a^j, \mathbf{x}_g^k) - f(\mathbf{x}_g^j, \mathbf{x}_a^k) \Big\}, \quad (6)$$

where $|\mathcal{M}_j|$ is the number of samples for $\mathcal{M}_j$. $\mathbf{x}_a^j$ and $\mathbf{x}_g^k$ are the a-th and g-th sample selected from $\mathcal{M}_j$ and $\mathcal{M}_k$, respectively. γ is the total number of samples for each memory buffer. $f(\cdot,\cdot)$ is a distance function for a data pair :

$$f(\mathbf{x}_a^j, \mathbf{x}_g^k) = f'(\text{enc}_\phi(\mathbf{x}_a^j), \text{enc}_\phi(\mathbf{x}_g^k)), \quad (7)$$

where f' is considered as linear in the experiments, while kernel functions can be considered. The MMD measure is used for implementing f_d for two reasons: (1) The MMD is a statistics grounded-measure, which has been widely used to evaluate the similarity between two distributions, (Tolstikhin, Sriperumbudur, and Schölkopf 2016); (2) The MMD is computationally efficient in the embedding space and does not require additional information;

Memory Unit Optimization

Most memory-based approaches focus on supervised learning, relying on class information, when predictions made by a classifier can guide the sample selection process (Kurle et al. 2020; Raghavan and Balaprakash 2021). However, such information is not available for unsupervised image generation. We introduce a task-free and class-free memory update approach that aims to selectively store the samples considered important in the current memory unit $\mathcal{M}_k$ containing the latest data samples for training. Consider that the memory $\mathcal{M} = \{\mathcal{M}_1, \cdots, \mathcal{M}_{k-1}\}$ has $(k-1)$ frozen memory units, while we also have an active memory unit $\mathcal{M}_k$, which is used to store novel information in $\mathcal{T}_i$. Our key idea is to encourage $\mathcal{M}_k$ to store those data samples that trigger large loss values and tend to be forgotten by the model. Given the data representation by the autoencoder $\{\text{enc}_\phi, \text{dec}_\eta\}$, we perform sample selection by using the loss value, given by the difference between the reconstruction error of each new sample $\mathbf{x}_{new} \in \mathcal{M}_k$ and the original :

$$f_s(\mathbf{x}_{new}) = ||\mathbf{x}_{new} - \text{dec}_\eta(\text{enc}_\phi(\mathbf{x}_{new}))||^2, \quad (8)$$

where $||\cdot||^2$ is the reconstruction error (square loss) function and the memory unit $\mathcal{M}_k$ at T_i is updated by :

$$\mathcal{M}_k = \mathcal{M}'[1] \cup \mathcal{M}'[2] \cup \cdots \cup \mathcal{M}'[\gamma], \quad (9)$$

where $\mathcal{M}'$ is the sorted joint set of $\mathcal{M}_k$ and $\mathbf{x}_i$, where each sample $\mathcal{M}'[a]$ denotes the a-th sample from $\mathcal{M}'$ satisfying $f_s(\mathcal{M}'[a]) > f_s(\mathcal{M}'[g])$ for $a < g$.

Memory Optimization Via Graph Relationship

Memory $\mathcal{M}$ cannot grow forever, especially when the model is deployed on a resource-constrained device such as a mobile phone or a drone. We address this issue by proposing a new memory reduction mechanism to selectively remove redundant memory units when the size of $\mathcal{M}$ exceeds a predefined maximum number D of memorized samples. Our primary aim is to identify the memory units that record similar information with each other in order and remove them to reduce redundancy. To do this we construct a graph, where each node represents a memory unit $\mathcal{M}_j$ and consider an adjacency matrix $\mathbf{A} \in \mathbb{R}^{k\times k}$ to represent the graph

Datasets	DEMU	DEMU-D	LTS	LGM	R-VAE	R-DDPM	CGKD-GAN	CVA	CGKD-WAE	CGKD-VAE
Split MNIST	20.18	19.23	71.67	66.31	55.67	63.26	54.34	21.46	47.98	48.72
Split Fashion	43.34	37.64	128.84	109.20	103.25	82.23	85.23	67.28	87.92	88.16
Split SVHN	62.12	53.49	87.25	72.60	65.18	87.22	101.26	57.14	100.15	102.87
Split CIFAR10	83.67	78.70	124.22	177.15	155.72	106.18	115.38	74.97	162.12	163.75
Average	54.07	**47.26**	102.99	106.31	94.95	84.72	89.05	55.21	99.54	100.87

Table 1: Image generation performance using the Fréchet Inception Distance score (FID) for class-incremental learning.

Datasets	DEMU	DEMU-D	LTS	LGM	R-VAE	R-DDPM	CGKD-GAN	CVA	CGKD-WAE	CGKD-VAE
CelebA-3DChair	81.53	**79.79**	186.25	241.14	210.18	183.72	132.12	142.62	154.45	156.62
CelebA-CACD	73.11	**69.85**	124.87	117.76	121.52	103.52	78.00	92.83	142.52	145.23
Split MINIImageNet	141.66	**123.65**	179.78	216.06	205.12	181.15	176.18	177.17	241.11	243.37

Table 2: FID scores for the image generation performance for datasets with complex images.

relationships, where $\mathbf{A}(a,g) = 1/f_d(\mathcal{M}_a, \mathcal{M}_g)$ denotes the knowledge similarity score between memory units $\mathcal{M}_a$ and $\mathcal{M}_g$, evaluated using the MMD-based distance measure from Eq. (6). Using the matrix $\mathbf{A}$, we find a pair of memory units with the maximum knowledge similarity :

$$a', g' = \arg \max_{a \neq g, a,g=1,\cdots,k} \{\mathbf{A}(a,g)\}\,, \tag{10}$$

where a' and g' are the indices of similar memory units. We then evaluate the discrepancy score for each candidate to decide which one should be removed as :

$$f_{\text{div}}(\mathcal{M}_{a'}) = \sum\nolimits_{m=1, m\neq a'}^{k} \{\mathbf{A}(a', m)\}\,. \tag{11}$$

If $f_{\text{div}}(\mathcal{M}_{a'}) > f_{\text{div}}(\mathcal{M}_{g'})$ indicates that $\mathcal{M}_{a'}$ has a large discrepancy with respect to all other memory units compared to $\mathcal{M}_{g'}$ and consequently we remove $\mathcal{M}_{g'}$ while retaining $\mathcal{M}_{a'}$. Otherwise, we remove $\mathcal{M}_{a'}$ and retain $\mathcal{M}_{g'}$. This mechanism removes redundant memory units while preserving knowledge diversity among the preserved units. We continually apply Eq. (10) and Eq. (11) to remove as many memory units as necessary until the number of memorized samples is $|\mathcal{M}| \leq D$.

Model Expansion Mechanism

Dynamically adding new components in a mixture system provided promising results for TFCL (Ye and Bors 2023). However, dynamically expanding the capacity of the DDPM model under TFUCL has not been studied so far. This paper bridges this gap by proposing a new dynamic expansion mechanism that automatically expands the capacity of the DDPM model over time, which can be used for learning infinite data streams. Let $\mathbf{G}$ be a dynamic expansion model and we assume that $\mathbf{G}$ has already built v experts $\mathbf{G} = \{\mathcal{G}_j\}_{j=1}^{v}$ at T_{i-1}, where each $\mathcal{G}_j$ consists of a DDPM model $\boldsymbol{\epsilon}_{\theta_j}$ and a data representation network $\{\text{enc}_{\phi_j}, \text{dec}_{\eta_j}\}$, where j is the expert index. To check whether to expand the model, we compare the relationship between the knowledge preserved by each expert and that corresponding to a new data batch $\mathbf{X}_i$ at T_i, and use this as an expansion signal criterion. This is achieved by means of the proposed MMD-based dynamic expansion mechanism :

$$\min\{f_d(\mathbf{x}_i, \mathbf{x}'_j) \,|\, j = 1, \cdots, v\} > \lambda_2\,, \tag{12}$$

where $\mathbf{x}'_j$ is a generated data batch using the DDPM model $\boldsymbol{\epsilon}_{\theta_j}$, representing the knowledge preserved by the j-th expert $\mathcal{G}_j$ and $\lambda_2 \in [0, 3]$ is a threshold controlling the dynamic expansion process. Since $\mathbf{G}$ consists of several experts, we design a function $f^\star$ to replace f in Eq. (6) to use all previously trained representation networks for the feature extraction, expressed as :

$$f^\star(\mathbf{x}_a^j, \mathbf{x}_g^k) = \frac{1}{v} \sum\nolimits_{m=1}^{v} \left\{ f'(\text{enc}_{\phi_m}(\mathbf{x}_a^j), \text{enc}_{\phi_m}(\mathbf{x}_g^k)) \right\}. \tag{13}$$

Then $\mathbf{G}$ builds a new expert $\mathcal{G}_{v+1}$ when the criterion from Eq. (12) is satisfied. Since the dynamic expansion model would have learned several experts, the sample selection criterion from Eq. (8), originally used for the static model, is redesigned for the dynamic expansion model $\mathbf{G}$, as :

$$f_s^\star(\mathbf{x}_j) = \frac{1}{v} \sum\nolimits_{m=1}^{v} ||\mathbf{x}_j - \text{dec}_{\eta_m}(\text{enc}_{\phi_m}(\mathbf{x}_j))||^2\,. \tag{14}$$

Eq. (14) evaluates the average reconstruction loss using the reconstructions from all previous network components of $\mathbf{G}$. We employ $f_s^\star(\cdot)$ to update the current memory unit $\mathcal{M}_k$ through (9) at T_i. We call our model using the dynamic expansion mechanism as DEMU-D.

Algorithm Implementation

We illustrate the learning procedure of a single model for DEMU in Fig. 2. Training DEMU consists of 4 steps :

Step 1 (Training). Initially, if the memory is empty $\mathcal{M} = \emptyset$, we build the first memory unit $\mathcal{M}_1 = \{\mathbf{x}_1\}$ and add it to $\mathcal{M}$. We train the DDPM model $\boldsymbol{\epsilon}_\theta$ and the representation network $\{\text{enc}_\phi, \text{dec}_\eta\}$ on $\mathcal{M} \cup \mathbf{x}_i$ using the DDPM objective function and reconstruction loss, respectively.

Methods	Resolution	CelebA-HQ	CACD	FFHQ
DEMU	$128 \times 128 \times 3$	**92.35**	**81.62**	**90.52**
CGKD-WVAE	$128 \times 128 \times 3$	139.96	158.32	179.59
CGKD-GAN	$128 \times 128 \times 3$	132.65	142.66	157.03

Table 3: FID scores for the image generation performance for datasets with images of high resolution.

Step 2 (Sample selection). We form a joint set $\mathcal{M}_k \cup \mathbf{x}_i$ and sort it to $\mathcal{M}'$ using Eq. (8). Then the current memory unit $\mathcal{M}_k$ is updated using (9).

Step 3 (Memory expansion). We build the second memory unit $\mathcal{M}_2$ at the training step (T_{100}) to use the memory expansion criterion from Eq. (5). Other samples are added in the same way to the memory buffers.

Step 4 (Memory reduction). If the memory buffer is full, we repeatedly perform Eq. (10) and Eq. (11) to remove those memory units containing redundant information units, until the memory reaches its maximum capacity, $|\mathcal{M}| = D$.

Experiments

Experimental Settings

Baselines and hyperparameters. Since most other TFCL methods can only perform classification tasks, we only consider several baselines that can be used for lifelong generative modeling, including Continual Variational Autoencoder (CVA) (Ye and Bors 2024), CGKD-GAN (Ye and Bors 2023) and CGKD-WAE, where 'WAE' indicates that each component of CGKD is implemented by a Wasserstein auto-encoder, (Tolstikhin et al. 2018). We also consider generative replay methods, including the Lifelong Teacher-Student (LTS) (Ye and Bors 2022) and Lifelong Generative Modelling (LGM) (Ramapuram, Gregorova, and Kalousis 2017). To enable LTS and LGM for TFCL, we assign a memory buffer to store the incoming data samples. Furthermore, we also consider employing a random sample selection approach, as a baseline. Specifically, we use this random sampling to train the DDPM and a VAE, respectively, resulting in R-DDPM and R-VAE. The batch size b for each processing time is considered as $b = 64$. The maximum memory size for all models is $D = 2,000$.

Performance criteria. Following from (Ye and Bors 2023), we employ the Fréchet Inception Distance (FID) score (Heusel et al. 2017) to evaluate the generation performance.

Class-Incremental Learning (CIL)

We divide each original dataset into five independent parts, where each part contains samples from two consecutive classes, as in (Aljundi, Kelchtermans, and Tuytelaars 2019), resulting in Split MNIST, Split Fashion, Split SVHN and Split CIFAR10. The FID scores for the image generation by various models after CIL are reported in Tab. 1. We observe that the dynamic expansion models, including CGKD-GAN and CGKD-WAE usually perform better than the baselines with static architectures such as R-VAE and R-DDPM. In addition, using a GAN as an expert in the dynamic expansion model, as in CGKD, can further improve its performance on all datasets when compared to using VAEs. Despite generative replay methods, including LTS and LGM, providing promising results in task-aware learning scenarios, they perform worse than other baselines in more challenging scenarios such as TFCL. Furthermore, the proposed DEMU with the static network architecture still outperforms all baselines, including the dynamic expansion model, by a large margin. Moreover, by dynamically increasing the model's capacity over time using the proposed dynamic expansion mechanism (DEMU-D), our model can further improve its performance on all datasets, as shown in Tab. 1. After the training, the number of components for DEMU-D in Split MNIST, Split Fashion, Split SVHN and Split CIFAR10 is of 7, 5, 7 and 6, respectively.

Settings					**Dataset**
λ	λ_2	MRP	SS	MDE	Split MNIST
0.01	0.02	✓	✓	✓	**19.23**
0.01	0.02	✗	✓	✓	33.79
0.01	0.02	✓	✗	✓	25.41
0.01	0.02	✓	✓	✗	23.12
0.04	0.02	✓	✓	✗	20.18

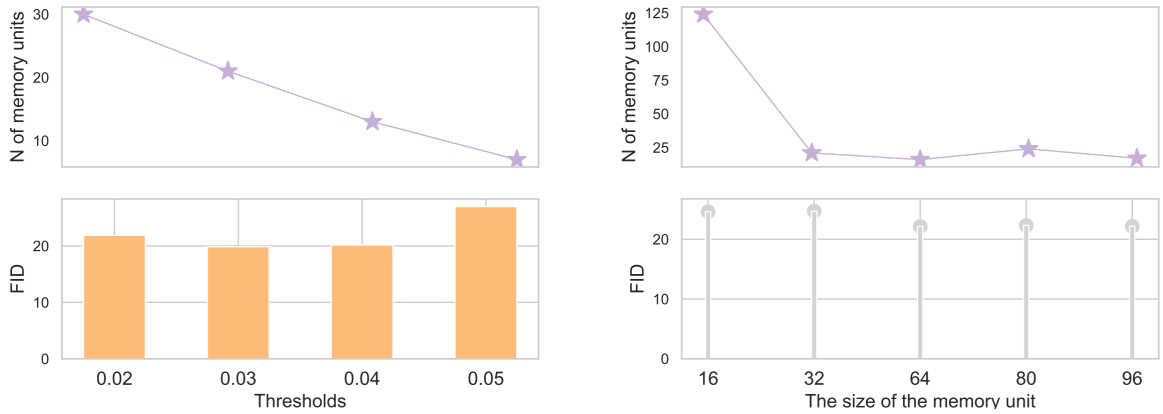
Table 4: Image generation (FID) when considering different settings for the memory expansion with λ from Eq. (5), and for the model expansion with λ_2 from Eq. (12).

Performance on Datasets with Complex Images

We train various models on a data stream $\mathcal{A}$ consisting of images of higher complexity, by grouping face images from CelebA (Liu et al. 2015) and CACD (Chen, Chen, and Hsu 2014) in a learning sequence CelebA-CACD. We also consider a more challenging data stream $\mathcal{A}$ that is comprised of images from two entirely different datasets, CelebA and 3DChair (Aubry et al. 2014) and we call this setting as CelebA-3DChair. The image generation performance of various models for CelebA-CACD and CelebA-3DChair is reported in Tab. 2. The proposed DEMU outperforms other baselines by a large margin on complex datasets. In addition, the proposed model expansion mechanism can further improve the model's performance, as demonstrated by the results achieved by DEMU-D.

Results on High-Resolution Images

We evaluate the proposed DEMU on datasets containing high-resolution images, including CACD (Chen, Chen, and Hsu 2014), CelebAHQ (Liu et al. 2015) and FFHQ (Karras, Laine, and Aila 2019). For each dataset, we create a data stream $\mathcal{A}$ by using all training data, considering $D = 2,000$ and $b = 64$. The FID for the image generation performance of various models on the three datasets is provided in Tab. 3. From these results, we observe that the proposed DEMU outperforms the state-of-the-art methods on these datasets with high-resolution images.



(a) Changing λ in Eq. (5). (b) The size of a memory unit.

Figure 3: Analysis results of the proposed approach DEMU. (a) The number of memory units (top) and the performance, according to the FID, (bottom) when changing λ. (b) Memory expansion and distribution (task) change over time.

Performance in Few-Shot Datasets

We investigate the effectiveness of various models on MINI-ImageNet (Vinyals et al. 2016) which was used in few-shot learning experiments (Schönfeld et al. 2019). The MINI-ImageNet (Vinyals et al. 2016) consists of 100 image categories, divided into 64, 16, and 20 classes corresponding to meta-training, meta-validation, and meta-test, respectively, in few-shot learning. We combine meta-training and meta-validation as a single training dataset, while the Split MINI-ImageNet data stream is divided into 16 tasks, each containing samples from five successive classes. From Tab. 2, we observe that DEMU-D provides the best image generation results for the continual learning of Split MINIImageNet.

Ablation Study

We perform a full ablation study to examine the performance of the proposed approach under different configurations.

Changing λ in Eq. (5). We change the threshold $\lambda \in [0.02, 0.05]$ in Eq. (5) when training on Split MNIST and examine the DEMU's performance. The performance results as well as the number of memory units are provided in Fig. 3-a. We observe that by gradually decreasing the threshold λ allows the DEMU to frequently build new memory units. In contrast, when increasing λ number of memory units for DEMU decreases while the model's performance does not change much. Using less than ten memory units can still provide a good performance for DEMU. Such results indicate that storing just a few, but diverse, samples in the memory buffer can achieve good performance.

The memory expansion process. We investigate the memory expansion for the proposed DEMU, during training. We train DEMU in Split MNIST using the threshold $\lambda = 0.05$ and plot the number of memory units used and distributions (tasks) learnt during time intervals in Fig. 3-b. We find that the proposed DEMU adds a new memory unit when the data distribution changes. A small threshold $\lambda = 0.02$ leads to frequently building memory units while the total memory size $|\mathcal{M}|$ reaches the maximum D at the middle of the entire training phase. DEMU still provides good performance for $\lambda = 0.02$ because the proposed memory reduction process removes only redundant memory units.

Changing the size of a memory unit. We train DEMU under Split MNIST by considering various memory unit sizes and the results are shown in Fig. 3-c. Using small-size memory units encourages DEMU to add more units during training. We can observe that changing the memory unit size does not influence much the performance. DEMU achieves the best performance when the memory unit has a capacity of 64 samples while requiring a low number of memory units.

The effect of each mechanism. In Tab. 4 we evaluate the image generation performance using FID, for the proposed dynamic expansion model without the Memory Reduction Process (MRP) and without the Sample Selection (SS) mechanism. The best performance is achieved by using all proposed mechanisms, including MRP, SS and the Model Dynamic Expansion (MDE). We also observe that without considering the memory buffer reduction process, the memory is full before the final training stage and then it is not able to store new data samples during the subsequent learning processes, resulting in degenerated performance.

Conclusion

In this paper, we propose the Dynamic Expansion Memory Unit (DEMU) in order to enable the Denoising Diffusion Generative Model (DDPM) model to learn continuously non-stationary data distributions without knowing any task or class information. To maintain a compact memory capacity, we introduce a novel memory buffer expansion approach that dynamically creates new memory units to capture novel information through a Maximum Mean Discrepancy (MMD) based criterion. We also provide an expansion mechanism for defining a mixture of generators.

Acknowledgments

This paper is supported by Sichuan Provincial Natural Fund Project (25NSFSC1269).

References

Achille, A.; Eccles, T.; Matthey, L.; Burgess, C.; Watters, N.; Lerchner, A.; and Higgins, I. 2018. Life-long disentangled representation learning with cross-domain latent homologies. In *Advances in Neural Information Processing Systems (NeurIPS)*, 9873–9883.

Aljundi, R.; Belilovsky, E.; Tuytelaars, T.; Charlin, L.; Caccia, M.; Lin, M.; and Page-Caccia, L. 2019a. Online Continual Learning with Maximal Interfered Retrieval. In *Advances in Neural Information Processing Systems (NeurIPS)*, 11872–11883.

Aljundi, R.; Kelchtermans, K.; and Tuytelaars, T. 2019. Task-free continual learning. In *Proc. of IEEE/CVF Conf. on Computer Vision and Pattern Recognition*, 11254–11263.

Aljundi, R.; Lin, M.; Goujaud, B.; and Bengio, Y. 2019b. Gradient based sample selection for online continual learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, 11817–11826.

Aubry, M.; Maturana, D.; Efros, A. A.; Russell, B. C.; and Sivic, J. 2014. 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)*, 3762–3769.

Bang, J.; Koh, H.; Park, S.; Song, H.; Ha, J.-W.; and Choi, J. 2022. Online Continual Learning on a Contaminated Data Stream With Blurry Task Boundaries. In *Proc. of IEEE/CVF Conf. on Computer Vision and Pattern Recog.*, 9275–9284.

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

Chen, B.-C.; Chen, C.-S.; and Hsu, W. H. 2014. Cross-Age Reference Coding for Age-Invariant Face Recognition and Retrieval. In *Proc. European Conf on Computer Vision (ECCV), vol. LNCS 8694*, 768–783.

Chen, S.; Sun, P.; Song, Y.; and Luo, P. 2023. DiffusionDet: Diffusion model for object detection. In *Proc. of IEEE/CVF International Conf. on Computer Vision*, 19830–19843.

Croitoru, F.-A.; Hondru, V.; Ionescu, R. T.; and Shah, M. 2023. Diffusion models in vision: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(9): 10850–10869.

De Lange, M.; and Tuytelaars, T. 2021. Continual prototype evolution: Learning online from non-stationary data streams. In *Proc. of the IEEE/CVF International Conference on Computer Vision*, 8250–8259.

Deng, D.; Chen, G.; Hao, J.; Wang, Q.; and Heng, P.-A. 2021. Flattening Sharpness for Dynamic Gradient Projection Memory Benefits Continual Learning. *Advances in Neural Information Processing Systems*, 34: 18710–18721.

Egorov, E.; Kuzina, A.; and Burnaev, E. 2021. BooVAE: Boosting Approach for Continual Learning of VAE. *Advances in Neural Information Processing Systems (NeurIPS)*, 34: 17889–17901.

Goodfellow, I.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A.; and Bengio, Y. 2014. Generative adversarial nets. In *Advances in Neural Information Processing Systems (NIPS)*, 2672–2680.

Gu, S.; Chen, D.; Bao, J.; Wen, F.; Zhang, B.; Chen, D.; Yuan, L.; and Guo, B. 2022a. Vector quantized diffusion model for text-to-image synthesis. In *Proc. of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 10696–10706.

Gu, Y.; Yang, X.; Wei, K.; and Deng, C. 2022b. Not Just Selection, but Exploration: Online Class-Incremental Continual Learning via Dual View Consistency. In *Proc. of IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 7442–7451.

Guo, Y.; Liu, B.; and Zhao, D. 2022. Online continual learning through mutual information maximization. In *International Conf. on Machine Learning (ICML)*, 8109–8126. PMLR 162.

Heusel, M.; Ramsauer, H.; Unterthiner, T.; Nessler, B.; and Hochreiter, S. 2017. GANs trained by a two time-scale update rule converge to a local Nash equilibrium. In *Proc. Advances in Neural Information Processing Systems (NIPS)*, 6626–6637.

Ho, J.; Jain, A.; and Abbeel, P. 2020. Denoising diffusion probabilistic models. *Advances in Neural Information Processing Systems*, 33: 6840–6851.

Ho, J.; Saharia, C.; Chan, W.; Fleet, D. J.; Norouzi, M.; and Salimans, T. 2022. Cascaded Diffusion Models for High Fidelity Image Generation. *J. Machine Learning Research*, 23(47): 1–33.

Hurtado, J.; Raymond, A.; and Soto, A. 2021. Optimizing reusable knowledge for continual learning via metalearning. *Advances in Neural Information Processing Systems*, 34: 14150–14162.

Jha, S.; Gong, D.; Zhao, H.; and Yao, L. 2024. NPCL: Neural Processes for Uncertainty-Aware Continual Learning. *Advances in Neural Information Proc. Systems*, 36: 4329–34353.

Jin, X.; Sadhu, A.; Du, J.; and Ren, X. 2021. Gradient-based Editing of Memory Examples for Online Task-free Continual Learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, 29193–29205.

Karras, T.; Laine, S.; and Aila, T. 2019. A style-based generator architecture for generative adversarial networks. In *Proc. of IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 4401–4410.

Kim, G.; Kwon, T.; and Ye, J. C. 2022. DiffusionCLIP: Text-guided diffusion models for robust image manipulation. In *Proc. of IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2426–2435.

Kingma, D. P.; and Welling, M. 2013. Auto-encoding variational Bayes. *arXiv preprint arXiv:1312.6114*.

Kurle, R.; Cseke, B.; Klushyn, A.; van der Smagt, P.; and Günnemann, S. 2020. Continual Learning with Bayesian Neural Networks for Non-Stationary Data. In *International Conference on Learning Representations (ICLR)*.

Lee, S.; Ha, J.; Zhang, D.; and Kim, G. 2020. A Neural Dirichlet Process Mixture Model for Task-Free Continual

Learning. In *International Conference on Learning Representations (ICLR), arXiv preprint arXiv:2001.00689*.

Li, C.-L.; Chang, W.-C.; Cheng, Y.; Yang, Y.; and Póczos, B. 2017. MMD GAN: Towards deeper understanding of moment matching network. *Advances in Neural Information Processing Systems (NIPS)*, 30: 2200–2210.

Li, H.; Yang, Y.; Chang, M.; Chen, S.; Feng, H.; Xu, Z.; Li, Q.; and Chen, Y. 2022. SRDIFF: Single image super-resolution with diffusion probabilistic models. *Neurocomputing*, 479: 47–59.

Liang, Y.-S.; and Li, W.-J. 2024. Loss decoupling for task-agnostic continual learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, 11151–11167.

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

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

Polikar, R.; Upda, L.; Upda, S. S.; and Honavar, V. 2001. Learn++: An incremental learning algorithm for supervised neural networks. *IEEE Trans. on Systems Man and Cybernetics, Part C*, 31(4): 497–508.

Raghavan, K.; and Balaprakash, P. 2021. Formalizing the Generalization-Forgetting Trade-off in Continual Learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, 17284– 17297.

Ramapuram, J.; Gregorova, M.; and Kalousis, A. 2017. Lifelong generative modeling. In *Proc. Int. Conf. on Learning Representations (ICLR), arXiv preprint arXiv:1705.09847*.

Rao, D.; Visin, F.; Rusu, A. A.; Teh, Y. W.; Pascanu, R.; and Hadsell, R. 2019. Continual Unsupervised Representation Learning. In *Advances in Neural Inf. Proc. Systems*, 7645–7655.

Rombach, R.; Blattmann, A.; Lorenz, D.; Esser, P.; and Ommer, B. 2022. High-resolution image synthesis with latent diffusion models. In *Proc. of IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 10684–10695.

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

Schönfeld, E.; Ebrahimi, S.; Sinha, S.; Darrell, T.; and Akata, Z. 2019. Generalized Zero- and Few-Shot Learning via Aligned Variational Autoencoders. In *Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition, (CVPR)*, 8247–8255.

Smith, J. S.; Cascante-Bonilla, P.; Arbelle, A.; Kim, D.; Panda, R.; Cox, D.; Yang, D.; Kira, Z.; Feris, R.; and Karlinsky, L. 2023a. Construct-VL: Data-free continual structured VL concepts learning. In *Proc. of IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 14994–15004.

Smith, J. S.; Tian, J.; Halbe, S.; Hsu, Y.-C.; and Kira, Z. 2023b. A closer look at rehearsal-free continual learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2409–2419.

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

Tiwari, R.; Killamsetty, K.; Iyer, R.; and Shenoy, P. 2022. GCR: Gradient Coreset Based Replay Buffer Selection for Continual Learning. In *Proc. of IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 99–108.

Tolstikhin, I.; Bousquet, O.; Gelly, S.; and Schoelkopf, B. 2018. Wasserstein Auto-Encoders. In *International Conference on Learning Representations (ICLR), arXiv preprint arXiv:1711.01558*.

Tolstikhin, I. O.; Sriperumbudur, B. K.; and Schölkopf, B. 2016. Minimax estimation of maximum mean discrepancy with radial kernels. In *Advances in Neural Information Processing Systems (NIPS)*, 1930–1938.

Villa, A.; Alcázar, J. L.; Alfarra, M.; Alhamoud, K.; Hurtado, J.; Heilbron, F. C.; Soto, A.; and Ghanem, B. 2023. Pivot: Prompting for video continual learning. In *Proc. of IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 24214–24223.

Vinyals, O.; Blundell, C.; Lillicrap, T.; Kavukcuoglu, K.; and Wierstra, D. 2016. Matching networks for one shot learning. *Advances in Neural Information Processing Systems (NIPS)*, 29: 3637–3645.

Wang, L.; Zhang, M.; Jia, Z.; Li, Q.; Bao, C.; Ma, K.; Zhu, J.; and Zhong, Y. 2021. AFEC: Active forgetting of negative transfer in continual learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, 22379–22391.

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

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

Ye, F.; and Bors, A. G. 2022. Lifelong Teacher-Student Network Learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(10): 6280–6296.

Ye, F.; and Bors, A. G. 2023. Continual Variational Autoencoder via Continual Generative Knowledge Distillation. In *Proc. of the AAAI Conference on Artificial Intelligence*, 10918–10926.

Ye, F.; and Bors, A. G. 2024. Task-Free Continual Generation and Representation Learning via Dynamic Expansionable Memory Cluster. In *Proc. of the AAAI Conference on Artificial Intelligence*, volume 38, 16451–16459.

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