Dynamic Expansion Diffusion Learning for Lifelong Generative Modelling

Fei Ye[1], Adrian G. Bors[2], Kun Zhang[3,4]

[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
[3]MBZUAI, Abu Dhabi, UAE,
[4]Carnegie Mellon University, Pittsburgh, PA, USA
feiye@uestc.edu.cn, adrian.bors@york.ac.uk, kunz1@cmu.edu

Abstract

The diffusion model has lately been shown to achieve remarkable performances through its ability of generating high quality images. However, current diffusion model studies consider only learning from a single data distribution, resulting in catastrophic forgetting when attempting to learn new data. In this paper, we explore a more realistic learning scenario where training data is continuously acquired. We propose the Dynamic Expansion Diffusion Model (DEDM) for addressing catastrophic forgetting and data distribution shifts under Online Task-Free Continual Learning (OTFCL) paradigm. New diffusion components are added to a mixture model following the evaluation of a criterion which compares the probabilistic representation of the new data with the existing knowledge of the DEDM model. In addition, to maintain an optimal architecture, we propose a component discovery approach that ensures the diversity of knowledge while minimizing the total number of parameters in the DEDM. Furthermore, we show how the proposed DEDM can be implemented as a teacher module in a unified framework for representation learning. In this approach, knowledge distillation is proposed for training a student module aiming to compress the teacher's knowledge into the latent space of the student.

Code — https://github.com/dtuzi123/DEDM

Introduction

The diffusion model (Ho, Jain, and Abbeel 2020), given its excellent properties and remarkable performances in image synthesis (Rombach et al. 2022), is increasingly becoming a popular image generation method. Compared to other deep-generative models, such as Generative Adversarial Nets (GANs) or Variational Autoencoders (VAEs), the Diffusion ModelS (DMS) enjoy several advantages, including stable training and good distribution coverage (Nichol and Dhariwal 2021). DMS has been successfully applied to many applications beyond image synthesis, including image super-resolution (Li et al. 2022), image inpainting (Song et al. 2021), graph generation (Niu et al. 2020), shape generation (Cai et al. 2020), and text-to-image generation (Gu et al. 2022a; Kim, Kwon, and Ye 2022). Despite all these advantages, current research on DMS only considers learning a single pre-defined and static data distribution, which is not realistic in real-world applications (Aljundi, Kelchtermans, and Tuytelaars 2019).

Training Dynamic Diffusion ModelS (DMS) in a realistic learning scenario where the data for training is provided successively (Aljundi, Kelchtermans, and Tuytelaars 2019) can enable the model to continuously learn novel data domains without forgetting its previously learnt knowledge. Recently, a few studies have proposed to enable diffusion models with continual learning abilities (Zajac et al. 2023; Gao and Liu 2023). However, these methods only consider a simple continual learning scenario where the task boundary is known and can not be applied in the context of the Online Task-Free Continual Learning (OTFCL) (Aljundi, Kelchtermans, and Tuytelaars 2019), where a model is trained on a dynamically changing data stream without accessing the task boundaries, representing a more realistic scenario. The primary obstacle for training a model under the OTFCL scenario is represented by catastrophic forgetting (Parisi et al. 2019), which results in significant drops in the model's performance for the information learnt earlier. Other approaches addressing forgetting in OTFCL employ a restricted memory buffer to store some critical samples (Jin et al. 2021) which then can be reused to train the model during subsequent learning. Such approaches require designing an appropriate sample selection strategy to store statistically diverse samples using a compact memory capacity (Aljundi et al. 2019a; De Lange and Tuytelaars 2021). However, most existing memory-based approaches focus only on supervised learning (De Lange and Tuytelaars 2021; Jin et al. 2021).

Some recent studies have considered the dynamic expansion of a mixture model for implementing unsupervised data generation in OTFCL (Lee et al. 2020), by using either GANs (Ye and Bors 2023) or VAEs (Lee et al. 2020; Rao et al. 2019). Ideally, each component in a mixture should aim to capture distinct information from others. The mixture's expansion is controlled by considering the performance of the network through the log-likelihood estimation (Lee et al. 2020; Rao et al. 2019) or by the Fréchet Inception Distance (FID) evaluation (Ye and Bors 2023), considering generated data. However, such dynamic expansion frameworks (Lee et al. 2020; Rao et al. 2019; Ye and Bors 2023) suffer from either mode collapse (Srivastava et al. 2017), or unstable GAN training (Wu et al. 2020), or from

poor VAEs output generation results (Liu et al. 2021). In this paper, we propose the Dynamic Expansion Diffusion Model (DEDM), which adds new diffusion model components to a mixture while implementing OTFCL-based generation tasks. Unlike existing approaches (Lee et al. 2020; Rao et al. 2019; Ye and Bors 2023), we propose implementing the dynamic expansion process from a different perspective. Specifically, we estimate probabilistic representations for the newly seen data batches and the knowledge preserved by each component. Then probabilistic distances are efficiently calculated between pairs of such tractable distributions and used as expansion signals, ensuring a compact architecture for DEDM. A component discarding approach is also proposed for further optimizing the mixture model architecture by automatically removing unnecessary components. Mixture components that are found to share similar knowledge, after analysing their knowledge similarities, are removed from the DEDM thus promoting further the knowledge diversity among the remaining components. We extend the abilities of the proposed framework for learning meaningful unsupervised representations in OTFCL through a Teacher-Student framework (Ye and Bors 2022), where DEDM is the Teacher, while the Student module is implemented by a VAE model. By means of knowledge distillation, the Student module captures meaningful latent representations across different data domains over time without forgetting, following the transfer of information from a very good image generator such as DEDM.

We summarise our contributions as follows: (1) To our best knowledge, this paper is the first study to explore a diffusion-based dynamic expansion model in OTFCL; (2) We propose a new approach to dynamically increase the capacity of the diffusion model for OTFCL without accessing any task and class labels; (3) A component discarding approach is proposed to reduce the model size without sacrificing its performance, which provides a new way to ensure the trade-off between the model's complexity and performance in OTFCL; (4) We extend the proposed DEDM to a teacher-student framework, which benefits many downstream tasks, including cross-domain reconstruction, interpolation and representation.

Related Work

Continual Learning : Employing a fixed-length memory buffer to store past samples is a popular approach for continual learning (Bang et al. 2022; Gu et al. 2022b; Liang and Li 2024; Jha et al. 2024). The model's performance can be further improved by considering knowledge distillation (Monaikul et al. 2021), or regularization optimization (Deng et al. 2021; Wang et al. 2021; Ye, Liu, and Wang 2023). In addition, the Generative Replay Mechanism (GRM) consists of training a Generative Adversarial Network (GAN) (Goodfellow et al. 2014) or a Variational Autoencoder (VAE) (Kingma and Welling 2013) for modelling and then reproducing past knowledge when learning new tasks (Rao et al. 2019; Shin et al. 2017). However, these models have rather limited memory resources, making it hard to deal with a growing number of tasks. Recently, mixture/ensemble models have been developed to address the learning of a long sequence of tasks by dynamically increasing the model's capacity (Cortes et al. 2017; Hung et al. 2019). Although in this way remarkable performances can be achieved, such models require to know the task information during the continual learning.

Online Task-Free Continual Learning (OTFCL) : OTFCL represents a special continual learning scenario which aims to train a model on a dynamically changing data stream without forgetting. In the initial OTFCL studies from (Aljundi, Kelchtermans, and Tuytelaars 2019; Liang and Li 2024), memory-based approaches had been shown to achieve promising performances. Specifically, the memory optimization strategies, implemented using the loss change (Aljundi et al. 2019a), gradients (Aljundi et al. 2019b) or the *learner-evaluator* framework (De Lange and Tuytelaars 2021) were shown to play critical roles in performance. However, these methods use a single memory buffer, limiting their ability to learn long-term data streams. Recent studies consider training a dynamic expansion model (Rao et al. 2019), which dynamically adds new inference models into a VAE-based framework when detecting data distribution changes. A similar idea was proposed in the Continual Neural Dirichlet Process Mixture (CN-DPM) which employs a Dirichlet process-based expansion mechanism for model expansion, (Lee et al. 2020). The dynamic expansion methodology was extended through the Continual Generative Knowledge Distillation (CGKD) (Ye and Bors 2023), to a teacher-student OTFCL-based framework.

Denoising Diffusion Probabilistic Models (DDPMs) : The diffusion model (Sohl-Dickstein et al. 2015), provides an excellent performance for image synthesis (Ho, Jain, and Abbeel 2020; Luhman and Luhman 2021; Song, Meng, and Ermon 2021). Unlike GANs or VAEs, which generate data in a unique step, diffusion-based methods generate images by means of an iterative process in which noise is generated and added to the image which is then gradually denoised, leading to learning the image through hundreds of diffusion steps (Ho, Jain, and Abbeel 2020). Most studies focus on shortening the reverse diffusion steps (Wang et al. 2023; Zheng et al. 2023) or by compressing the latent space (Rombach et al. 2022) to reduce the significant computational costs required by the original models. In this paper, we focus on enabling the DDPM, with continual learning capabilities, without considering any task information or boundaries, which has not been studied before. Other diffusion model variants can be smoothly embedded into our continual learning framework with only small modifications.

Methodology

Problem Definition

In online task-free continual learning (OTFCL), a model is trained on a dynamically evolving data stream under the assumption that data probabilistic representation shifts occur over time. Let $\mathcal{T} = \{\mathcal{T}_1, \cdots, \mathcal{T}_n\}$ be a series of n training times. Each $\mathcal{T}_i$ contains a batch of samples $\mathbf{X}_i = \{\mathbf{x}_j\}_{j=1}^{b}$, where b is the batch size. When a model is trained at a certain training time $\mathcal{T}_i$, we can only see the associated data batch $\mathbf{X}_i$ once, while all previously visited data batches

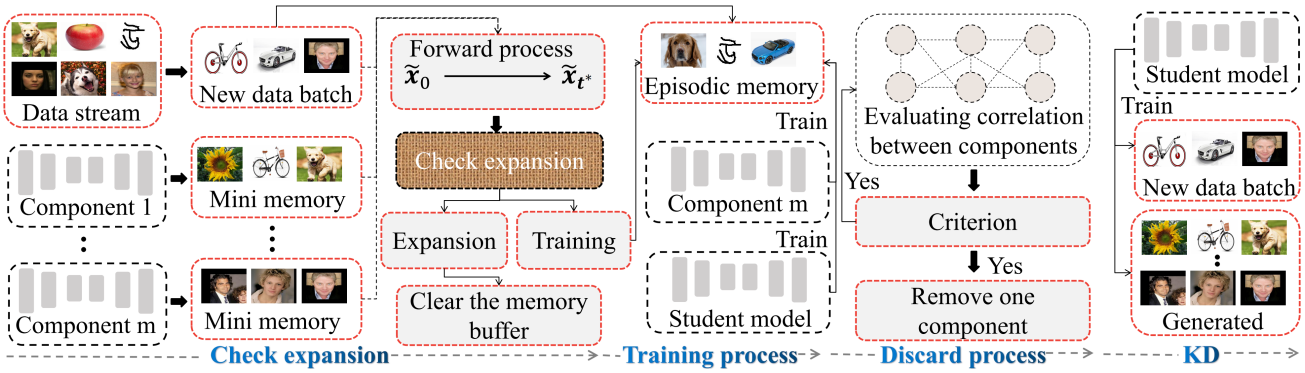


Figure 1: The training of the proposed Dynamic Expansion Diffusion Model (DEDM), consisting of four stages. In the first step, we check the model expansion by evaluating the novelty of a new data batch. During the second step, the current component and the student module are trained on memory buffers. Once all training times are finished, we perform the third step to remove redundant components, which are unnecessary. The final step consists in training the student using knowledge distillation.

Algorithm 1: Training algorithm

for $i < n$ **do**
  First step (Checking the expansion) :
  if $|\mathcal{M}_{i-1}| \geq |\mathcal{M}_{max}|$ **then**
    if Eq. (8) is satisfied **then**
      Build a new component
      Clear up the memory buffer $\mathcal{M}_{i-1}$
    end if
  end if
  Second step (Training process) :
  Get new samples $\mathbf{X}_i$
  if $|\mathcal{M}_{i-1}| = |\mathcal{M}_{max}|$ **then**
    $\mathcal{M}_i = \mathcal{M}_{i-1}[b : |\mathcal{M}_{max}|] \bigcup \mathbf{X}_i$
  else
    $\mathcal{M}_i = \mathcal{M}_{i-1} \bigcup \mathbf{X}_i$
  end if
  Train $\boldsymbol{\epsilon}_{\theta_m}$ on $\mathcal{M}_i$ using Eq. (1)
  Train the student using Eq. (12)
end for
Discard process and knowledge distillation :
Please see details in **Appendix-A** from SM

$\{\mathbf{X}_1, \cdots, \mathbf{X}_{i-1}\}$ are unavailable. Different from existing OTFCL approaches (Aljundi, Kelchtermans, and Tuytelaars 2019), which mainly focus on classification tasks, the proposed model aims to learn a generative model capable of generating and reconstructing all previously learnt data after finishing all training stages $\{\mathcal{T}_1, \cdots, \mathcal{T}_n\}$.

The Dynamic Expansion Diffusion Model (DEDM)

Existing research on the DMS only considers learning a predefined and static data distribution (Ho, Jain, and Abbeel 2020; Song, Meng, and Ermon 2021) using a single model. A model like this cannot be applied to a dynamically evolving probabilistic data environment. In this section, we train on a dynamically changing data stream by introducing a new framework called the Dynamic Expansion Diffusion Model (DEDM). When compared to the static model, the dynamic expansion framework has several advantages : (1) It is scalable to learn long or infinite data streams; (2) It can preserve the best performance on previously learnt samples. Let $\mathcal{H}$ be a dynamic expansion model which is assumed to continually train m components $\mathcal{H} = \{\boldsymbol{\epsilon}_{\theta_1}, \cdots, \boldsymbol{\epsilon}_{\theta_m}\}$ at $\mathcal{T}_i$, where each $\boldsymbol{\epsilon}_{\theta_j}$ can be seen as an independent component/expert implemented by a Denoising Diffusion Probabilistic Model (DDPM). In order to ensure that each component learns sufficient information, we follow the standard setting (Lee et al. 2020; Rao et al. 2019; Ye and Bors 2023) by introducing a fixed-length episodic memory buffer $\mathcal{M}_i$ to store more recent samples from a data stream, where the subscript i indicates that $\mathcal{M}_i$ was updated at $\mathcal{T}_i$. One reasonable approach to train the DEDM without forgetting is by constantly updating only a new component $\boldsymbol{\epsilon}_{\theta_m}$ while freezing all previously learnt components $\{\boldsymbol{\epsilon}_{\theta_1}, \cdots, \boldsymbol{\epsilon}_{\theta_{m-1}}\}$. We employ the Improved DDPM objective function ($\mathcal{L}_{\text{IDDPM}}$) (Nichol and Dhariwal 2021) for training $\boldsymbol{\epsilon}_{\theta_m}$ at $\mathcal{T}_i$:

$$\lambda \mathbb{E}_{q(\tilde{\mathbf{x}}_{0:T}^{\mathcal{M}_i} \mid \tilde{\mathbf{x}}_0^{\mathcal{M}_i})} \Big[-\log p_{\theta_m}(\tilde{\mathbf{x}}_0^{\mathcal{M}_i} \mid \tilde{\mathbf{x}}_1^{\mathcal{M}_i}) + \sum_{t=1}^{T-2} \{\mathcal{L}_t\} + D_{KL}[q(\tilde{\mathbf{x}}_T^{\mathcal{M}_i} \mid \tilde{\mathbf{x}}_0^{\mathcal{M}_i}) \,||\, p_{\theta_m}(\tilde{\mathbf{x}}_T^{\mathcal{M}_i})] \Big] \quad (1)$$
$$+ \mathbb{E}_{t, \tilde{\mathbf{x}}_0^{\mathcal{M}_i}, \boldsymbol{\epsilon}} \Big[\|\boldsymbol{\epsilon} - \boldsymbol{\epsilon}_{\theta_m}(\sqrt{\hat{\alpha}_t}\tilde{\mathbf{x}}_0^{\mathcal{M}_i} + \sqrt{1-\hat{\alpha}_t}\boldsymbol{\epsilon}, t)\|^2 \Big] ,$$

where $\mathcal{L}_t$ is defined as :

$$\mathcal{L}_t = D_{KL}[q(\tilde{\mathbf{x}}_t^{\mathcal{M}_i} \mid \tilde{\mathbf{x}}_{t+1}^{\mathcal{M}_i}, \tilde{\mathbf{x}}_0^{\mathcal{M}_i}) \,||\, p_{\theta_m}(\tilde{\mathbf{x}}_t^{\mathcal{M}_i} \mid \tilde{\mathbf{x}}_{t+1}^{\mathcal{M}_i})], \quad (2)$$

where $\tilde{\mathbf{x}}_0^{\mathcal{M}_i}$ is a real sample obtained from $\mathcal{M}_i$ and $\tilde{\mathbf{x}}_t^{\mathcal{M}_i}$ is a noise vector drawn from $\mathcal{N}(\mathbf{0}, \mathbf{I})$. $p_{\theta_m}(\tilde{\mathbf{x}}_0^{\mathcal{M}_i} \mid \tilde{\mathbf{x}}_1^{\mathcal{M}_i})$ and $q(\tilde{\mathbf{x}}_T^{\mathcal{M}_i} \mid \tilde{\mathbf{x}}_0^{\mathcal{M}_i})$ are the distributions defined within the backward and forward diffusion process, respectively. $D_{KL}(\cdot, \cdot)$ is the Kullback–Leibler (KL) divergence and $q(\tilde{\mathbf{x}}_t^{\mathcal{M}_i} \mid \tilde{\mathbf{x}}_{t+1}^{\mathcal{M}_i}, \tilde{\mathbf{x}}_0^{\mathcal{M}_i})$ is defined as :

$$q(\tilde{\mathbf{x}}_t^{\mathcal{M}_i} \mid \tilde{\mathbf{x}}_{t+1}^{\mathcal{M}_i}, \tilde{\mathbf{x}}_0^{\mathcal{M}_i}) = \mathcal{N}\Big(\tilde{\mathbf{x}}_t^{\mathcal{M}_i}; \frac{\sqrt{\hat{\alpha}_t}\beta_{t+1}}{1-\hat{\alpha}_t}\tilde{\mathbf{x}}_0^{\mathcal{M}_i} + \frac{\sqrt{\hat{\alpha}_{t+1}}(1-\hat{\alpha}_t)}{1-\hat{\alpha}_t}\tilde{\mathbf{x}}_{t+1}^{\mathcal{M}_i}, \hat{\beta}_{t+1}\mathbf{I}\Big) , \quad (3)$$

where $\hat{\beta}_{t+1} = \frac{(1-\hat{\alpha}_t)}{1-\hat{\alpha}_{t+1}}\beta_{t+1}$. In the experiments, we consider $\lambda = 0.001$ to ensure that the first term from the IDDPM loss, Eq. (1), does not overwhelm the second term.

Sampling from DEDM. One advantage of the proposed DEDM over a single DDPM is the ability to model multiple independent data distributions over time, expressed as :

$$p_{\theta_{1:m}}(\tilde{\mathbf{x}}_{0:T}^{\theta_1}, \cdots, \tilde{\mathbf{x}}_{0:T}^{\theta_m}) = \{p(\tilde{\mathbf{x}}_T^{\theta_1}) \prod_{t=1}^{T} p_{\theta_1}(\tilde{\mathbf{x}}_{t-1}^{\theta_1} \mid \tilde{\mathbf{x}}_t^{\theta_1}), \cdots, p(\tilde{\mathbf{x}}_T^{\theta_m}) \prod_{t=1}^{T} p_{\theta_m}(\tilde{\mathbf{x}}_{t-1}^{\theta_m} \mid \tilde{\mathbf{x}}_t^{\theta_m})\}, \quad (4)$$

where $p(\tilde{\mathbf{x}}_T^{\theta_1}) = p(\tilde{\mathbf{x}}_T^{\theta_m}) = \mathcal{N}(\mathbf{0}, \mathbf{I})$. $\tilde{\mathbf{x}}_0^{\theta_j}$ is a real sample drawn from an unknown data distribution $p(\tilde{\mathbf{x}}_0^{\theta_j})$ learnt by the j-th component $\boldsymbol{\epsilon}_{\theta_j}$ where $j \in [1, m]$. By using Eq. (4), we can recover a joint data distribution $p(\tilde{\mathbf{x}}_0^{\theta_1}, \cdots, \tilde{\mathbf{x}}_0^{\theta_m})$ through the backward diffusion process. In practice, the proposed DEDM can also produce a single image through the following generation process :

$$\tilde{\mathbf{x}}_{t-1}^{\theta_{s^\star}} = \frac{1}{\sqrt{\alpha_t}}\Big(\tilde{\mathbf{x}}_t^{\theta_{s^\star}} - \frac{1-\alpha_t}{\sqrt{1-\alpha_t}}\boldsymbol{\epsilon}_{\theta_{s^\star}}(\tilde{\mathbf{x}}_t^{\theta_{s^\star}}, t)\Big) + \sigma_t \tilde{\mathbf{z}} \quad , s^\star = \mathrm{Cat}(m, p_1, \cdots, p_m)\,, \quad (5)$$

where $\mathrm{Cat}(\cdot)$ is the categorical distribution and $\{p_1 = 1/m, \cdots, p_m = 1/m\}$ are event probabilities and $\sigma_t = \sqrt{\beta_t}$ for $t = T, \cdots, 1$, while $s^\star$ is the selected component index for the generation process. $\tilde{\mathbf{z}} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ is a random noise vector if $t > 1$, otherwise, is 0. We start from a random noisy image $\tilde{\mathbf{x}}_T^{\theta_{s^\star}}$ drawn from a Gaussian distribution $\mathcal{N}(\mathbf{0}, \mathbf{I})$ and then gradually refine $\tilde{\mathbf{x}}_T^{\theta_{s^\star}}$ using Eq. (5). Eq. (4) indicates that what was learnt by each component is key for the generation process. If each distribution $p(\tilde{\mathbf{x}}_0^{\theta_j})$ learnt by each component $\boldsymbol{\epsilon}_{\theta_j}$, $j = 1, \cdots, m$, represents overlapping information with what was learnt by other components then DEDM leads to generating similar data by multiple components, resulting in redundancies.

Model Expansion via Forward Diffusion Process

A continual learning system aims to learn as much information as possible during a succession of learning stages without forgetting. In our approach, the DDPM is expanded with new components when learning novel information creating a mixture of diffusion generative networks system. A novelty information criterion is proposed for deciding when to expand the DEDM model. By using existing DDPM models from the mixture to generate a large number of samples for comparing with the new data batches provided for training, is impractical. This is because DDPM would require considerable computational resources during the generation process. In consequence, we propose a new and efficient dynamic expansion mechanism, which compares a statistical distance between the data distribution associated with each component and that corresponding to a new data batch at $\mathcal{T}_i$ as :

$$\min_{j=1,\cdots,m} \left\{ f'(p(\mathbf{X}_i), \mathcal{P}_{\theta_j}) \right\} \geq \gamma\,, \quad (6)$$

where m represents the number of components, $f'(\cdot)$ is a probability distance and $\gamma \in [0, 40]$ is a hyperparameter that controls the expansion process. $p(\mathbf{X}_i)$ and $\mathcal{P}_{\theta_j}$ are the distributions characterizing a new data batch $\mathbf{X}_i$ obtained from the data stream at $\mathcal{T}_i$, and the probabilistic representation of the component $\boldsymbol{\epsilon}_{\theta_j}$, respectively. A small γ favours expanding the model frequently, while a large value has the opposite effect. For estimating the probability distance between $p(\mathbf{X}_i)$ and $\mathcal{P}_{\theta_j}$, one can use the Fréchet Inception Distance (FID) (Heusel et al. 2017) for $f'(\cdot)$ which can be estimated using empirical distributions. However, the FID evaluation relies on using an additional neural network for the evaluation. Another option is to implement $f'(\cdot)$ by using a discriminator trained using adversarial learning (Arjovsky, Chintala, and Bottou 2017; Goodfellow et al. 2014), to estimate the discrepancy between two distributions. This paper aims to develop an efficient approach to implement $f'(\cdot)$ without requiring additional network architectures or high computational costs. By considering the forward diffusion process, we transfer $p(\mathbf{X}_i)$ and $\mathcal{P}_{\theta_j}$ to Gaussian distributions through a few forward diffusion steps :

$$\begin{aligned} q(\tilde{\mathbf{x}}_{t^\star} \mid \tilde{\mathbf{x}}_0) &= \mathcal{N}(\tilde{\mathbf{x}}_{t^\star}; \sqrt{\hat{\alpha}_{t^\star}}\tilde{\mathbf{x}}_0, (1-\hat{\alpha}_{t^\star})\mathbf{I}), q(\tilde{\mathbf{x}}_{t^\star}^{\theta_j} \mid \tilde{\mathbf{x}}_0^{\theta_j}) \\ &= \mathcal{N}(\tilde{\mathbf{x}}_{t^\star}^{\theta_j}; \sqrt{\hat{\alpha}_{t^\star}}\tilde{\mathbf{x}}_0^{\theta_j}, (1-\hat{\alpha}_{t^\star})\mathbf{I})\,, \end{aligned} \quad (7)$$

where $t^\star$ is an appropriate diffusion step and we employ a small $t^\star$ to ensure that $q(\tilde{\mathbf{x}}_{t^\star} \mid \tilde{\mathbf{x}}_0)$ and $q(\tilde{\mathbf{x}}_{t^\star}^{\theta_j} \mid \tilde{\mathbf{x}}_0^{\theta_j})$ preserve enough statistical information for $p(\mathbf{X}_i)$ and $\mathcal{P}_{\theta_j}$. $\tilde{\mathbf{x}}_0$ and $\tilde{\mathbf{x}}_0^{\theta_j}$ are data samples drawn from $p(\mathbf{X}_i)$ and $\mathcal{P}_{\theta_j}$, and $\tilde{\mathbf{x}}_{t^\star}$ and $\tilde{\mathbf{x}}_{t^\star}^{\theta_j}$ are the corresponding corrupted samples at the diffusion step $t^\star$. By considering Eq. (7), we transform Eq. (6) into an efficient expansion criterion at $\mathcal{T}_i$, as :

$$\min_{j=1}^{m} \left\{ \mathbb{E}_{\tilde{\mathbf{x}}_0 \sim p(\mathbf{X}_i), \tilde{\mathbf{x}}_0^{\theta_j} \sim \mathcal{P}_{\theta_j}} [f'(q(\tilde{\mathbf{x}}_{t^\star} \mid \tilde{\mathbf{x}}_0), q(\tilde{\mathbf{x}}_{t^\star}^{\theta_j} \mid \tilde{\mathbf{x}}_0^{\theta_j}))] \right\} \geq \gamma\,. \quad (8)$$

Since $q(\tilde{\mathbf{x}}_{t^\star} \mid \tilde{\mathbf{x}}_0)$ and $q(\tilde{\mathbf{x}}_{t^\star}^{\theta_j} \mid \tilde{\mathbf{x}}_0^{\theta_j})$ have explicit density forms, we can easily evaluate their probability distance without extra training costs. In this paper, we consider implementing $f'(\cdot)$ from Eq. (8) using Jensen–Shannon (JS) divergence for two reasons : (1) The JS divergence is symmetric and can be calculated effectively on Gaussian distributions; (2) The JS divergence always has a finite value. On the other hand, evaluating Eq. (8) at each training time is still time-consuming since estimating each $\mathcal{P}_{\theta_j}$ involves multiple backward diffusion steps. To address this issue, we assign a tiny memory buffer $\mathcal{M}^j$ for each diffusion mixture component $\boldsymbol{\epsilon}_{\theta_j}$, for storing a few data samples representing the information learnt by $\boldsymbol{\epsilon}_{\theta_j}$. The tiny memory $\mathcal{M}^j$ initially stores the samples used to train $\boldsymbol{\epsilon}_{\theta_j}$ and then replaces them with randomly selected samples from $\mathcal{M}_i$ after finishing training $\boldsymbol{\epsilon}_{\theta_j}$ which is then frozen at $\mathcal{T}_i$. We employ the memory distribution $p(\mathcal{M}^j)$ of $\mathcal{M}^j$ to replace $\mathcal{P}_{\theta_j}$ in Eq. (8), which significantly reduces computational costs at each training step as we do not have to generate data by each diffusion component at each step of the continuous training. When the expansion criterion, defined by Eq. (8), is satisfied at $\mathcal{T}_i$, we build a new component for DEDM and train it in the subsequent training steps.

FID ↓ (Generation)							
Datasets	DEDM	Reservoir-VAE	Reservoir-DDPM	CGKD-GAN	CNDPM	CGKD-WVAE	CGKD-VAE
Split MNIST	34.26	56.10	65.06	56.61	65.34	48.57	46.47
Split Fashion	53.78	107.92	81.15	86.51	175.38	88.92	89.76
Split SVHN	54.79	66.24	86.23	103.91	153.36	101.25	104.29
Split CIFAR10	82.12	156.71	107.25	112.29	234.08	163.52	164.74
Average	**56.23**	96.74	84.92	89.83	157.04	100.56	101.31
PSNR ↑ (Reconstruction)							
Split MNIST	21.09	18.48	13.53	18.21	21.03	20.22	20.15
Split Fashion	20.01	15.67	11.14	16.08	18.13	19.90	19.84
Split SVHN	22.72	21.14	10.77	21.14	18.96	21.12	21.02
Split CIFAR10	18.66	17.38	9.91	18.52	17.65	17.79	17.69
Average	**20.62**	18.16	11.33	18.48	18.94	19.75	19.67
SSIM ↑ (Reconstruction)							
Split MNIST	0.93	0.86	0.59	0.86	0.94	0.92	0.92
Split Fashion	0.92	0.81	0.56	0.83	0.88	0.92	0.92
Split SVHN	0.91	0.86	0.47	0.87	0.76	0.86	0.86
Split CIFAR10	0.84	0.81	0.47	0.83	0.79	0.79	0.79
Average	**0.90**	0.83	0.53	0.84	0.84	0.87	0.87

Table 1: Image reconstruction performance for class-incremental learning.

The Component Discarding Mechanism

In order to further ensure a compact model and promote knowledge diversity among the DEDM's components, we introduce a component discarding mechanism, aiming at removing redundant components without sacrificing performance. The primary idea of the proposed approach is to evaluate the knowledge similarity for all pairs of mixture components and then identify and remove those that share similar information with others and therefore are redundant. To achieve this, we first evaluate the correlation on the knowledge of a pair of diffusion mixture components $\boldsymbol{\epsilon}_{\theta_a}$ and $\boldsymbol{\epsilon}_{\theta_g}$, using :

$$f_s(\boldsymbol{\epsilon}_{\theta_a}, \boldsymbol{\epsilon}_{\theta_g}) = \mathbb{E}_{\tilde{\mathbf{x}}_0^{\theta_a} \sim \mathcal{P}_{\theta_a}, \tilde{\mathbf{x}}_0^{\theta_g} \sim \mathcal{P}_{\theta_g}} \left[f'(q(\tilde{\mathbf{x}}_{t^\star}^{\theta_a} \mid \tilde{\mathbf{x}}_0^{\theta_a}), q(\tilde{\mathbf{x}}_{t^\star}^{\theta_g} \mid \tilde{\mathbf{x}}_0^{\theta_g}) \right], \tag{9}$$

where $\mathcal{P}_{\theta_a}$ and $\mathcal{P}_{\theta_g}$ are distributions of samples generated by the components $\boldsymbol{\epsilon}_{\theta_a}$ and $\boldsymbol{\epsilon}_{\theta_g}$, respectively. We assume that the DEDM has already trained m components and we produce a component knowledge relationship adjacency graph matrix $\mathbf{C} \in \mathbb{R}^{m \times m}$, whose weights are defined by Eq. (9), where each element $C_{a,g} = f_s(\boldsymbol{\epsilon}_{\theta_a}, \boldsymbol{\epsilon}_{\theta_g})$ describes the discrepancy score between each two DEDM components of indices $a, g = 1, \cdots, m$, $a \neq g$. Then, we identify a pair of components $\{a^\star, g^\star\}$ which share similar knowledge by using :

$$\{a^\star, g^\star\} = \arg\min_{a,g=1,\cdots,m, a \neq g} \{C_{a,g}\}. \tag{10}$$

We evaluate the diversity score of $\boldsymbol{\epsilon}_{\theta_{a^\star}}$ and $\boldsymbol{\epsilon}_{\theta_{g^\star}}$ by :

$$\begin{aligned} s_{a^\star} &= \sum\nolimits_{j=1, j \neq a^\star}^{m} \{C_{a^\star, j}\}, \\ s_{g^\star} &= \sum\nolimits_{j=1, j \neq g^\star}^{m} \{C_{g^\star, j}\}. \end{aligned} \tag{11}$$

A higher diversity score indicates that the selected component represents distinct knowledge from all other components and should be kept. If $s_{a^\star} > s_{g^\star}$, then we decide to remove $\boldsymbol{\epsilon}_{\theta_{g^\star}}$, while otherwise we would remove $\boldsymbol{\epsilon}_{\theta_{a^\star}}$. Meanwhile, the matrix $\mathbf{C}$ is also updated and reduced in size by removing the row and column corresponding to the deleted component. We continually discard redundant components using Eq. (10) and (11) until a certain number of components m are retained or a maximum overlapping knowledge between two components is achieved.

Representation Learning

Compressing information from multiple data distributions into a unified latent space can benefit many applications, such as cross-domain image reconstruction and interpolation. To achieve this goal, this study employs the DEDM as a teacher aiming to provide high-quality knowledge within a teacher-student framework, aiming to enable unsupervised representation learning into a student model. Specifically, we implement the student module as a VAE model and introduce a new loss function to learn meaningful representations across domains over time. The student module consists of an encoding distribution $q_{\varphi_{stu}}(\mathbf{z} \mid \mathbf{x})$ and a decoding distribution $p_{\theta_{stu}}(\mathbf{x} \mid \mathbf{z})$. A latent variable $\mathbf{z} = \boldsymbol{\mu}_{\varphi_{stu}} + \boldsymbol{\sigma}_{\varphi_{stu}} \odot \boldsymbol{\tau}$ is drawn from the encoding distribution $\mathcal{N}(\boldsymbol{\mu}_{\varphi_{stu}}, \boldsymbol{\sigma}_{\varphi_{stu}} \mathbf{I})$ using the reparameterization trick (Kingma and Welling 2013), where $\boldsymbol{\mu}_{\varphi_{stu}}$ and $\boldsymbol{\sigma}_{\varphi_{stu}}$ are given by the encoder. $\boldsymbol{\tau} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ while $\odot$ is the element-wise product. Because generating images using each component of DEDM requires substantial computational costs and time, we train the student module using the VAE objective function (Kingma and Welling 2013) on the memory $\mathcal{M}^j$ associated with each

Datasets	DEDM	Reservoir-VAE	Reservoir-DDPM	CGKD-GAN	CNDPM	CGKD-WVAE	CGKD-VAE
FID ↓ (Generation)							
CelebA-Chair	**69.52**	220.52	182.32	131.81	372.08	158.58	159.82
CelebA-ImageNet	**116.77**	252.64	196.62	127.54	349.94	123.27	168.96
S-MINIImageNet	**140.19**	206.86	180.03	177.89	314.51	244.43	250.97
PSNR ↑ (Reconstruction)							
CelebA-Chair	**18.89**	16.82	13.25	18.04	12.48	17.92	17.85
CelebA-ImageNet	**19.89**	16.52	12.62	19.21	17.99	19.61	19.50
S-MINIImageNet	**18.94**	18.87	9.55	18.35	18.19	18.23	18.21
SSIM ↑ (Reconstruction)							
CelebA-Chair	**0.93**	0.80	0.52	0.91	0.78	0.89	0.89
CelebA-ImageNet	**0.90**	0.78	0.56	0.87	0.82	0.88	0.88
S-MINIImageNet	**0.87**	0.85	0.47	0.86	0.82	0.83	0.83

Table 2: Image generation and reconstruction for the domain-incremental and few-shot datasets.

component, together with the episodic memory buffer $\mathcal{M}_i$, at $\mathcal{T}_i$:

$$\mathcal{L}_{\text{stu}} = \mathbb{E}_{\mathbf{x}\sim p(\mathcal{M}^1,\cdots,\mathcal{M}^m,\mathcal{M}_i)}\big[-\mathbb{E}_{q_\varphi(\mathbf{z}\,|\,\mathbf{x})}\left[\log p_\phi\left(\mathbf{x}\,|\,\mathbf{z}\right)\right] + D_{KL}\left[q_\varphi\left(\mathbf{z}\,|\,\mathbf{x}\right)\,||\,p\left(\mathbf{z}\right)\right]\big], \tag{12}$$

where $p(\mathcal{M}^1,\cdots,\mathcal{M}^m,\mathcal{M}_i)$ is the distribution of the memory buffers $\{\mathcal{M}^1,\cdots,\mathcal{M}^m,\mathcal{M}_i\}$. However, one weakness of using Eq. (12) is that the student module may not learn sufficient prior information because of the limitations of the tiny memory size $\mathcal{M}^j$, $j=1,\ldots,m$ in representing the entire knowledge learnt by each component. In order to address this issue, we propose an off-line training approach for the student module, involving $\mathcal{L}_{stu}$ and the knowledge distillation loss performed only after finishing the entire Teacher' training, as :

$$\mathcal{L}'_{\text{stu}} = \mathcal{L}_{\text{stu}} + \sum\nolimits_{j=1}^{m}\Big\{\mathbb{E}_{\mathbf{x}\sim\mathcal{P}_{\theta_j}}\big[D_{KL}\left[q_\varphi\left(\mathbf{z}\,|\,\mathbf{x}\right)\,||\,p\left(\mathbf{z}\right)\right] - \mathbb{E}_{q_\varphi(\mathbf{z}\,|\,\mathbf{x})}\left[\log p_\phi\left(\mathbf{x}\,|\,\mathbf{z}\right)\right]\big]\Big\}, \tag{13}$$

where the last term is used to transfer the knowledge learnt by the teacher to the student module.

Algorithm Implementation

The learning process of the DEDM, whose architecture is provided in Fig. 1, with the pseudo-code outlined in **Algorithm 1**, and can be summarized in four steps :

Step 1 (Checking the expansion). We check the model expansion by using Eq. (8), when the episodic memory buffer $\mathcal{M}_i$ at $\mathcal{T}_i$ is full $|\mathcal{M}_{i-1}| = |\mathcal{M}|_{max}$, where $|\mathcal{M}_{i-1}|$ and $|\mathcal{M}|_{max}$ represent the current and maximum buffer capacity. When fulfilling the expansion criterion from Eq. (8) we add a new component to DEDM, otherwise, we go to **Step 2**.

Step 2 (Training process). The episodic memory buffer is simply updated by adding a new data batch $\mathcal{M}_i = \mathcal{M}_{i-1}\bigcup\mathbf{X}_i$ until the memory buffer is full, $|\mathcal{M}_{i-1}| = |\mathcal{M}|_{max}$, otherwise, we remove the earliest stored samples and add the new data batch $\mathbf{X}_i$ to $\mathcal{M}_{i-1}$. We train the current component $\boldsymbol{\epsilon}_{\theta_m}$ and the student module using Eq. (1) and Eq. (12), respectively.

Step 3 (Component discarding process). When all training steps are completed, we perform Eq. (10) and Eq. (11) to remove the redundant components.

Step 4 (Knowledge distillation). We retrain the student module using Eq. (13) to learn the knowledge accumulated by the teacher module.

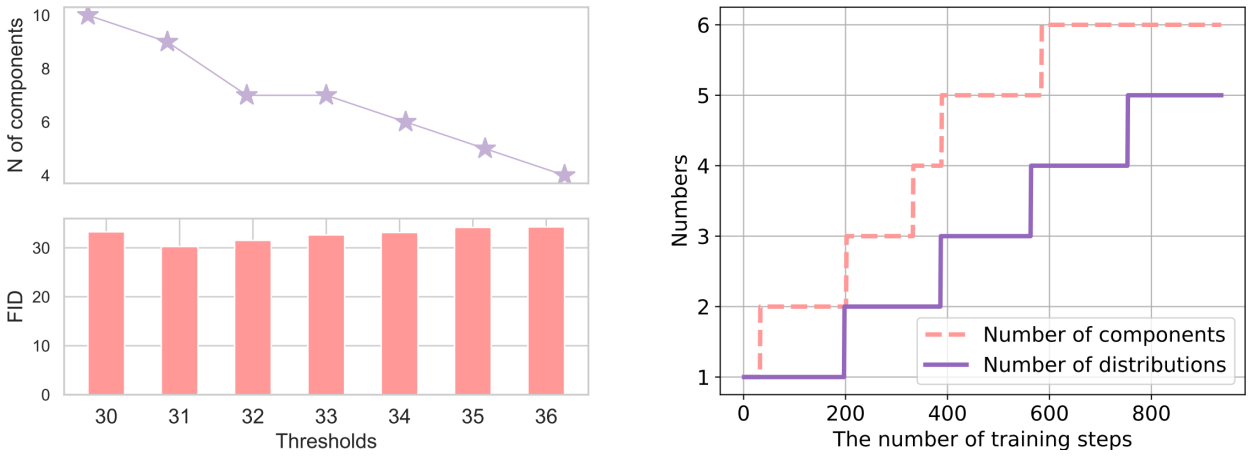
Experiments

Results for Class-Incremental Learning

We consider the class-incremental learning of the Split MNIST, Split Fashion, Split SVHN and Split CIFAR10, where each learning task consists of data from 2 consecutive classes of the original datasets MNIST, Fashion, SVHN and CIFAR10, as in (Aljundi, Kelchtermans, and Tuytelaars 2019), and the results are given in Tab. 1. For comparison, we also train a VAE as a Student module when considering CNDPM (Lee et al. 2020) as a Teacher and we evaluate image reconstruction and generation. From Tab. 1 DEDM outperforms other methods when evaluating image reconstructions using PSNR and SSIM. The VAE trained by a teacher module performs well on simple datasets such as MNIST and SVHN, but has worse results on datasets characterized by complex images. Furthermore, the GAN-based teacher module does not always perform well on each dataset due to the instability characterizing adversarial learning.

Results on the Domain-Incremental Learning

We investigate the performance of various models in a more challenging setting, that of domain-incremental learning, where a data stream is built by collecting samples from multiple data domains. We consider CelebA (Liu et al. 2015) and 3D-Chair (Aubry et al. 2014), namely CelebA-Chair and also by collecting images from CelebA to ImageNet (Krizhevsky, Sutskever, and Hinton 2012), namely CelebA-ImageNet. The image generation and reconstruction perfor-



(a) Changing γ in Eq. (6). (b) DEDM's expansion process.

Figure 2: DEDM expansion when considering the continual learning of Split MNIST. (a) The number of components and performance when changing the threshold γ. (b) The number of components and data distributions (tasks) at each training time.

Methods	Resolution	CelebA-HQ	CACD	FFHQ
DEDM	$128 \times 128 \times 3$	**52.63**	**70.97**	**85.50**
CGKD-GAN	$128 \times 128 \times 3$	132.65	142.66	157.03
CGKD-WVAE	$128 \times 128 \times 3$	139.96	158.32	179.59
DEDM	$256 \times 256 \times 3$	**61.52**	**101.32**	**105.25**
CGKD-GAN	$256 \times 256 \times 3$	168.52	236.98	254.32
CGKD-VAE	$256 \times 256 \times 3$	176.63	240.12	261.37

Table 3: FID scores for the image generation performance for datasets containing high-resolution images.

mance on CelebA-Chair and CelebA-ImageNet by various models is reported in Tab. 2. These results indicate that the proposed DEDM performs best on each dataset.

Results on the Task-Free Few-Shot Learning

Few-Shot Image Generation (FSIG) represents a challenging task considered in supervised learning (Tang et al. 2022; Zhao et al. 2022), where the number of training samples for each category is very limited. We create Split MINI-ImageNet (S-MINIImageNet) made up of 16 sets, with each set containing samples The results from Tab. 2 indicate that the proposed DEDM achieves the best performance in Task-Free Few-Shot Learning on S-MINIImageNet.

Results on the High-Resolution Datasets

We test the proposed approach on the CelebA-HQ (Liu et al. 2015), CACD (Chen, Chen, and Hsu 2014) and FFHQ (Karras, Laine, and Aila 2019) datasets, consisting of high resolution images. The maximum memory size of various models for all datasets is 2,000. The number of training epochs and the batch size in each training time are 10 and 64, respectively. We report the FID generation results on different resolution settings in Tab. 3, where we compare with CGKD which represents the state-of-the-art method for image generation under TFCL. These empirical results show the effectiveness of the proposed DEDM, which outperforms CGKD in high-resolution image generation.

Ablation Study

We perform some ablation tests analyzing the proposed DEDM. In Fig. 2-a we assess the number of components for DEDM and corresponding performance, when varying γ in Eq. (6). We can observe that the number of diffusion mixture components of DEDM is increasing when decreasing the threshold γ, while also DEDM provides a stable performance with respect to FID for the generated images.

We plot the expansion process of DEDM in Fig. 2-b, when the model is trained on Split MNIST and the proposed dynamic expansion mechanism adds a new component when the data distribution changes. This expansion process ensures that each component captures a unique data distribution while minimizing the overall model size.

Conclusion

This paper introduces the Dynamic Expansion Diffusion Model (DEDM), which enables an expanding diffusion model with the ability to learn continuously. The mixture model DEDM dynamically expands its capacity, when novel data is presented for learning, in order to capture the new knowledge without forgetting. The novelty of the information, provided for training, is assessed using the Jensen-Shanon divergence criterion. We also propose a component pruning approach when different components learn similar information. A Teacher-Student architecture is also proposed, where the knowledge learnt by the DEDM, considered as a teacher, is compressed in a student model through knowledge distillation.

Acknowledgments

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

References

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 the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 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.

Arjovsky, M.; Chintala, S.; and Bottou, L. 2017. Wasserstein Generative Adversarial Networks. In *Proc. Int. Conf. on Machine Learning (ICML)*, 214–223.

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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 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.

Cortes, C.; Gonzalvo, X.; Kuznetsov, V.; Mohri, M.; and Yang, S. 2017. Adanet: Adaptive structural learning of artificial neural networks. In *Proc. of Int. Conf. on Machine Learning (ICML), vol. PMLR 70*, 874–883.

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 (ICCV)*, 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.

Gao, R.; and Liu, W. 2023. DDGR: Continual learning with deep diffusion-based generative replay. In *Proc. International Conference on Machine Learning*, 10744–10763. PMLR 202.

Goodfellow, I.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A.; and Bengio, Y. 2014. Generative adversarial nets. In *Proc. Advances in Neural Inf. Proc. 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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 7442–7451.

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 (NeurIPS)*, 33: 6840–6851.

Hung, C.-Y.; Tu, C.-H.; Wu, C.-E.; Chen, C.-H.; Chan, Y.-M.; and Chen, C.-S. 2019. Compacting, Picking and Growing for Unforgetting Continual Learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, 13647–13657.

Jha, S.; Gong, D.; Zhao, H.; and Yao, L. 2024. NPCL: Neural Processes for Uncertainty-Aware Continual Learning. *Advances in Neural Information Processing Systems (NeurIPS)*, 36.

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 the 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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2426–2435.

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

Krizhevsky, A.; Sutskever, I.; and Hinton, G. E. 2012. Imagenet classification with deep convolutional neural networks. In *Advances in Neural Inf. Proc. Systems (NIPS)*, 1097–1105.

Lee, S.; Ha, J.; Zhang, D.; and Kim, G. 2020. A Neural Dirichlet Process Mixture Model for Task-Free Continual Learning. In *Int. Conf. on Learning Representations (ICLR), arXiv preprint arXiv:2001.00689*.

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)*, volume 36, 11151–11167.

Liu, X.; Xing, F.; Prince, J. L.; Carass, A.; Stone, M.; El Fakhri, G.; and Woo, J. 2021. Dual-cycle constrained bijective VAE-GAN for tagged-to-cine magnetic resonance image synthesis. In *Proc. IEEE International Symposium on Biomedical Imaging (ISBI)*, 1448–1452.

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.

Luhman, E.; and Luhman, T. 2021. Knowledge distillation in iterative generative models for improved sampling speed. *arXiv preprint arXiv:2101.02388*.

Monaikul, N.; Castellucci, G.; Filice, S.; and Rokhlenko, O. 2021. Continual learning for named entity recognition. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, 13570–13577.

Nichol, A. Q.; and Dhariwal, P. 2021. Improved Denoising Diffusion Probabilistic Models. In *Proc. of the International Conference on Machine Learning (ICML), vol. PMLR 139*, 8162–8171.

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.

Parisi, G. I.; Kemker, R.; Part, J. L.; Kanan, C.; and Wermter, S. 2019. Continual lifelong learning with neural networks: A review. *Neural Networks*, 113: 54–71.

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

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

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

Sohl-Dickstein, J.; Weiss, E.; Maheswaranathan, N.; and Ganguli, S. 2015. Deep unsupervised learning using nonequilibrium thermodynamics. In *Proc. International Conference on Machine Learning (ICML)*, 2256–2265. PMLR 37.

Song, J.; Meng, C.; and Ermon, S. 2021. Denoising diffusion implicit models. In *International Conference on Learning Representations (ICLR) arXiv preprint arXiv:2010.02502*.

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*.

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

Tang, L.; Cai, Y.; Liu, J.; Hong, Z.; Gong, M.; Fan, M.; Han, J.; Liu, J.; Ding, E.; and Wang, J. 2022. Few-Shot Font Generation by Learning Fine-Grained Local Styles. In *Proc. of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 7895–7904.

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. *Advances in Neural Information Processing Systems (NeurIPS)*, 34: 22379–22391.

Wang, Z.; Zheng, H.; He, P.; Chen, W.; and Zhou, M. 2023. Diffusion-GAN: Training GANs with diffusion. In *Internatioal Conference on Learning Representations (ICLR), arXiv preprint arXiv:2206.02262*.

Wu, Y.; Zhou, P.; Wilson, A. G.; Xing, E.; and Hu, Z. 2020. Improving GAN training with probability ratio clipping and sample reweighting. *Advances in Neural Information Processing Systems (NeurIPS)*, 33: 5729–5740.

Ye, F.; and Bors, A. G. 2022. Lifelong Teacher-Student Network Learning. *IEEE Trans. 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 AAAI Conference on Artificial Intelligence*, 10918–10926.

Ye, J.; Liu, S.; and Wang, X. 2023. Partial network cloning. In *Proc. of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 20137–20146.

Zajac, M.; Deja, K.; Kuzina, A.; Tomczak, J. M.; Trzciński, T.; Shkurti, F.; and Miłoś, P. 2023. Exploring continual learning of diffusion models. *arXiv preprint arXiv:2303.15342*.

Zhao, Y.; Ding, H.; Huang, H.; and Cheung, N.-M. 2022. A closer look at few-shot image generation. In *Proc. of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 9140–9150.

Zheng, H.; He, P.; Chen, W.; and Zhou, M. 2023. Truncated Diffusion Probabilistic Models and Diffusion-based Adversarial Auto-Encoders. In *International Conference on Learning Representations (ICLR), arXiv preprint arXiv:2202.09671*.