Continual Unsupervised Generative Modelling via Online Optimal Transport

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

Lately, deep generative models have achieved excellent results after learning pre-defined and static data distribution. Meanwhile, their performance on continual learning suffers from degeneration, caused by catastrophic forgetting. In this paper, we study the unsupervised generative modelling in a more realistic continual learning scenario, where class and task information are absent during both training and inference learning phases. To implement this goal, the proposed memory approach consists of a temporary memory system, which stores data examples while a dynamic expansion memory system would gradually preserve those samples that are crucial for long-term memorization. A novel memory expansion mechanism is then proposed, by employing optimal transport distances between the statistics of memorized samples and each newly seen datum. This paper proposes the Sinkhorn-based Dual Dynamic Memory (SDDM) method, by considering Sinkhorn distance as an optimal transport measure, for evaluating the significance of the data to be stored in the memory buffer. The Sinkhorn transport algorithm leads to preserving a diversity of samples within a compact memory capacity. The memory buffering approach does not interact with the model's training process and can be optimized independently in both supervised and unsupervised learning without any modifications. Moreover, we also propose a novel dynamic model expansion mechanism to automatically increase the model's capacity whenever necessary, which can deal with infinite data streams and further improve the model's performance. Experimental results show that the proposed approach achieves state-of-the-art performance in both supervised and unsupervised learning.

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

Introduction

Learning data from a dynamically changing environment is very important, yet challenging, for an artificial intelligence system. Such an ability would enable the machine to continually capture new knowledge over time. In Continual Learning (CL) a model can only access a subset during each task learning while past data are not available (Ye and Bors 2024). Although modern deep learning models such as ResNet (He et al. 2016) or the Vision Transformer (Dosovitskiy et al. 2021) have achieved excellent processing performances on individual tasks (Lin et al. 2015; Wu et al. 2015), training them in CL remains challenging, resulting in significant performance degeneration on past data, caused by catastrophic forgetting (Parisi et al. 2019).

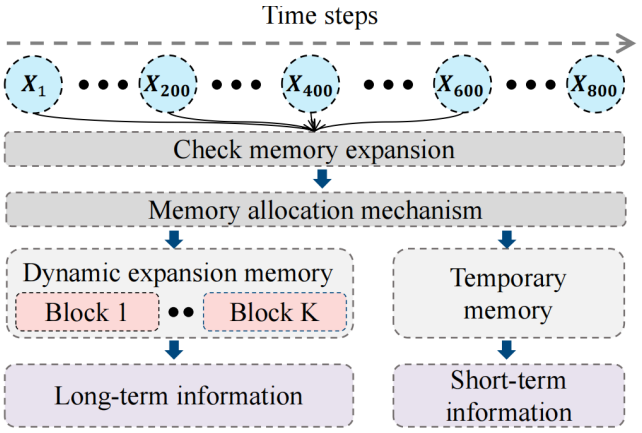


Figure 1: The proposed memory management approach consists of two memory buffer systems with different storage mechanisms for short-term and long-term information.

Several approaches to address network forgetting in CL have been proposed so far. Among these, using a fixed-size memory buffer for storing important data samples represents a natural choice (Bang et al. 2021). These memory-based methods usually require knowing the task boundaries and therefore would fail to deal with the challenges of most real-world applications, where data are unsupervised, (Aljundi, Kelchtermans, and Tuytelaars 2019). Some recent studies have explored memory-based approaches in a more realistic continual learning scenario, called the Online Task-Free Continual Learning (OTFCL) (Aljundi, Kelchtermans, and Tuytelaars 2019; Zhou, Sohn, and Lee 2012), where task identity information is not available during the training. These memory systems usually design a sample selection algorithm using gradient information (Jin et al. 2021) or mutual information (Guo, Liu, and Zhao 2022), while they still rely on supervised signals such as class labels or model prediction, making them unsuitable for unsupervised learn-

ing.

In this paper, we study image generation modeling under continual unsupervised learning, which has scarcely been explored before. Generative image modeling is essential for AI-Generated Content (AIGC) (Jiang, Zhang, and Gong 2023) and was recently considered when learning a generator network to approximate the empirical data distributions for lifelong generative modeling, (Achille et al. 2018; Ramapuram, Gregorova, and Kalousis 2017; Zhai et al. 2019; Ye and Bors 2024). Most existing memory-based methods fail to implement lifelong generation modeling under unsupervised learning because they rely on supervised signals. This issue is addressed in the paper by proposing a novel memory management approach inspired by a biological perspective. Specifically, the brain can quickly remember short-term information and gradually preserve important information through different memory mechanisms when necessary, (Izquierdo et al. 1998). Inspired by this, the proposed memory approach for CL, illustrated in Fig. 1, consists of a temporary memory system used for the preservation of short-term information and a dynamic expansion memory to dynamically store critical samples for persistent information. We propose a novel memory expansion mechanism, which gradually accumulates data with similar semantic information in specific fixed-size memory blocks while aiming to capture new underlying data distributions. To achieve this goal, we encode the probabilistic representation of each memory block when accessing new data and formulate the memory expansion as a dynamic process through the Online Optimal Transport (OOT) framework. This framework continually evaluates the knowledge similarity between each memory block and the incoming data in an online processing manner, using the optimal transportation distance as a memory expansion signal. Thus it maintains a diversified statistics over all data categories without accessing any supervised signals. We also propose a novel memory allocation mechanism that initially assigns more memory buffers for preserving short-term information and then gradually allocates memory capacities to preserve more timely data. Such a mechanism can provide sufficient samples for training the model at the initial stage while maintaining enough capacities to preserve permanent information for later stages.

In order to address the learning of long-term and complex data streams consisting of multiple data domains, we dynamically add new experts, each implemented by a generative model, or a classifier, resulting in a mixture system. This is achieved by proposing a novel model expansion mechanism which evaluates the knowledge diversity among experts as the signal for expanding the network architecture, leading to a compact model structure. Furthermore, we use the proposed memory management approach to enable the Denoising Diffusion Probabilistic Model (DDPM) (Ho, Jain, and Abbeel 2020) for implementing lifelong generative modeling under OTFCL, achieving a good performance.

The contributions of this paper are summarized as follows : (1) It proposes a biologically inspired memory approach to address unsupervised generative modelling under OTFCL; (2) An online optimal transport framework, based on the Sinkhorn distance, is proposed to regulate the memory expansion process, ensuring the storage of a diversity of data samples; (3) We propose a novel dual dynamic memory allocation mechanism, which provides more data samples for the initial training phase while maintaining enough memory capacities for the later learning stages; (4) A novel model expansion mechanism is proposed to deal with infinite and complex data streams, further improving the model's performance; (5) We theoretically demonstrate that the proposed model can preserve diverse samples and achieve good results.

Related Work

Continual Learning. CL methods are roughly divided into three categories : memory-based, regulation-based and architecture expansion-driven. Using a memory buffer for storing past examples is a natural choice and has been shown to achieve good performances (Bang et al. 2022; Guo, Liu, and Zhao 2022; Liang and Li 2024; Tiwari et al. 2022; Villa et al. 2023). Memory buffer-based approaches can adopt regularization-based implementations that penalize changes of certain network weights, leading to performance improvements (Deng et al. 2021; Hurtado, Raymond, and Soto 2021; Wang et al. 2021; Lyu et al. 2024). Besides preserving real training examples, a memory-based approach can be implemented by training a generator that learns data representations and then replays characteristic past samples (Achille et al. 2018; Ramapuram, Gregorova, and Kalousis 2017; Shin et al. 2017; Zhai et al. 2019). Moreover, dynamic architecture expansion methods are suitable for addressing an increasing number of tasks (Polikar et al. 2001; Rusu et al. 2016; Xiao et al. 2014; Zhou, Sohn, and Lee 2012).

Online Task-Free Continual Learning (OTFCL). Unlike traditional CL, the OTFCL represents a more challenging scenario, dealing with the learning of changing data distributions over time. Using a memory buffer, relying on storing data when either the loss (Aljundi, Kelchtermans, and Tuytelaars 2019; Aljundi et al. 2019a) or the gradient (Aljundi et al. 2019b) changes significantly, was shown to be an efficient approach for OTFCL. In addition, mutual information (Guo, Liu, and Zhao 2022) and an *learner-evaluator* framework (De Lange and Tuytelaars 2021) have been used for OTFCL. Another approach to the OTFCL is to dynamically create new experts to capture data distribution changes over time (Lee et al. 2020).

Generative Modeling in CL. Training a generator network to preserve past information and then replay data characteristic of past tasks, when learning new tasks was considered in (Achille et al. 2018), where a teacher-student framework was used in (Ramapuram, Gregorova, and Kalousis 2017). However, these methods are applied only in a learning environment with explicit task boundaries and do not address OTFCL. To deal with this issue, Ye and Bors (2023) proposed using the Fréchet Inception Distance (FID) as a criterion to expand network architecture when learning new tasks. However, such an approach requires additional network resources.

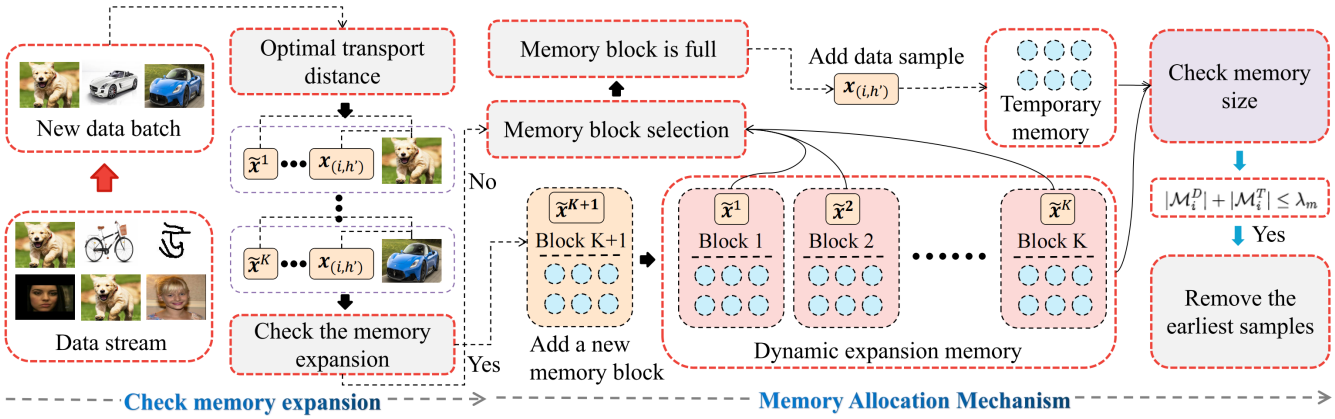


Figure 2: The proposed Sinkhorn-based Dual Dynamic Memory (SDDM) continual learning method. The knowledge discrepancy between each existing representative sample $\tilde{\mathbf{x}}^j$ and an incoming sample $\mathbf{x}_{(i,h)}$ at S_i is evaluated using the optimal transport. We add a new memory block $\mathcal{M}_i^D(K+1)$ with the associated representative sample $\tilde{\mathbf{x}}^{K+1}$ in $\mathcal{M}_i^D$ if Eq. (9) is satisfied. At the memory allocation process, we choose an appropriate memory block through the selection mechanism (Eq. (10)). If the selected memory block $\mathcal{M}_i^D(c^\star)$ is not full, we store $\mathbf{x}_{(i,h)}$ in $\mathcal{M}_i^D(c^\star)$, otherwise, we preserve $\mathbf{x}_{(i,h)}$ in the temporary memory $\mathcal{M}_i^T$ at S_i. If the overall memory is overloaded, $|\mathcal{M}_i^T| + |\mathcal{M}_i^D| > |\mathcal{M}|_{max}$, the temporal memory is emptied, as necessary.

Methodology

Problem Definition

This paper focuses on online continual unsupervised generative modeling in which a model aims to learn underlying data distributions without having access to the task information. Let $\mathcal{D}_i^s = \{\mathbf{x}_j\}_{j=1}^{n_i^s}$ and $\mathcal{D}_i^t = \{\mathbf{x}_j\}_{j=1}^{n_i^t}$ denote the i-th unlabelled training and testing datasets, where n_i^s and n_i^t are their numbers of samples, respectively. Each $\mathbf{x}_j \in \mathbb{R}^d$ is an image over the space $\mathcal{X}$, where d represents the data dimension. In a class-incremental learning setting (Aljundi, Kelchtermans, and Tuytelaars 2019), a training dataset $\mathcal{D}_i^s$ is divided into c subsets $\{\mathcal{D}_{i,1}^s, \cdots, \mathcal{D}_{i,c}^s\}$, where each part contains data samples from several consecutive categories. A data stream U is built by sequentially collecting these subsets, expressed as $U = \{\mathcal{D}_{i,1}^s, \mathcal{D}_{i,2}^s, \cdots, \mathcal{D}_{i,c}^s\}$. At a certain training session/time (the j-th training session S_j), a model can only obtain one data batch $\mathbf{X}_j$ consisting of b examples while all previously learnt data batches $\{\mathbf{X}_1, \cdots, \mathbf{X}_{j-1}\}$ are inaccessible. After the final training session (S_n) is completed, we evaluate the model's performance on the whole testing dataset $\mathcal{D}_j^t$ by assessing the image generation performance for all learnt tasks. The proposed approach is also extended to be used for the supervised learning, where the class label is available for each instance.

Dual Dynamic Memory

Current memory-based approaches would usually consider employing a single memory buffer to store previously seen data samples (Chrysakis and Moens 2020), which would not easily capture both short-term and long-term information during the whole training process. In this paper, we propose a novel memory management system inspired from a biological perspective (Izquierdo et al. 1998), with the aim of storing both temporary and persistent information. Specifically, the proposed memory approach consists of a temporary memory buffer, denoted as $\mathcal{M}_i^T$ and a dynamic expansion memory, considered as persistent, denoted as $\mathcal{M}_i^D$, where i indicates that the memories are updated at S_i.

The temporary memory buffer $\mathcal{M}_i^T$ aims to store the more recently given samples, which can provide enough information for training the model at the initial learning phase. We consider that the memory buffer $\mathcal{M}_i^T$ adds a new sample at the front of a list, while removing the samples from its tail like in the first-in-fist-out (FIFO). In contrast, optimizing the dynamic expansion memory $\mathcal{M}_i^D$ is challenging since it may easily store many samples during the early stages, overwhelming its capacity, and therefore it risks being left without any space to further preserve critical information during subsequent learning. We address this issue by proposing a novel dynamic memory expansion mechanism that gradually expands $\mathcal{M}_i^D$ appropriately, while capturing diverse information during the whole training phase. Specifically, we initially add a single representative sample $\tilde{\mathbf{x}}^j$ into $\mathcal{M}_i^D$, characterizing a different category/distribution, and then gradually add samples that share similar semantic information with respect to this sample $\tilde{\mathbf{x}}^j$, avoiding storing too many permanent samples at an early stage, where the subscript j denotes the sample index. We formulate the updating of the dynamic expansion memory $\mathcal{M}_i^D$ as an optimization problem with constraints at S_i as :

$$\begin{aligned} \mathbf{x}_{(s^\star,k^\star)} &= \arg\max_{\mathbf{x}_{(s',h)}} \sum_{j=1}^{K} f_d(\tilde{\mathbf{x}}^j, \mathbf{x}_{(s',h)}), \\ &\text{s.t. } f_d(\tilde{\mathbf{x}}^j, \tilde{\mathbf{x}}^{j'}) > \lambda, j \neq j', \end{aligned} \tag{1}$$

where $f_d(\cdot,\cdot)$ is a distance measure function that evaluates the similarity of a pair of data samples. $\mathbf{x}_{(s',h)}$ is the h-th data sample from the data batch

$\mathbf{X}_{s'} = \{\mathbf{x}_{(s',1)}, \cdots, \mathbf{x}_{(s',b)}\}$, where $s' = i, \cdots, n$ and $h = 1, \cdots, b$. $\lambda > 0$ is a hyperparameter that ensures an appropriate knowledge discrepancy among the representative samples in $\mathcal{M}_i^D$ and K is the number of existing representative samples in $\mathcal{M}_i^D$ at S_i. The optimization procedure from Eq. (1) aims to find the next best representative sample $\mathbf{x}_{(s^\star,h^\star)}$ that is different enough from the other representative samples $\{\tilde{\mathbf{x}}^1, \tilde{\mathbf{x}}^2, \cdots, \tilde{\mathbf{x}}^K\}$. However, searching for the optimal solution in Eq. (1) can be challenging because it requires accessing all data stream samples at once, which is intractable in OTFCL. We address this issue by formulating Eq. (1) as a dynamic memory expansion mechanism that adds a new representative sample to $\mathcal{M}_i^D$ when detecting the data distribution shift at S_i :

$$\begin{aligned} \min \{ & f_d(\tilde{\mathbf{x}}^1, \mathbf{x}_{(i,h)}), f_d(\tilde{\mathbf{x}}^2, \mathbf{x}_{(i,h)}), \cdots, \\ & f_d(\tilde{\mathbf{x}}^K, \mathbf{x}_{(i,h)})\} > \lambda \,, h = 1, \cdots, b \,. \end{aligned} \tag{2}$$

We get a new data batch $\mathbf{X}_i = \{\mathbf{x}_{(i,1)}, \cdots, \mathbf{x}_{(i,b)}\}$ at $\mathcal{S}_i$ and evaluate the memory expansion criterion for each new sample $\mathbf{x}_{(i,h)}$. If Eq. (2) is satisfied, we then dynamically build a new representative sample $\tilde{\mathbf{x}}^{K+1} = \mathbf{x}_{(i,h)}$ and add it in $\mathcal{M}_i^D$. Eq. (2) can ensure that each newly added $\tilde{\mathbf{x}}^{K+1}$ represents information which is different with respect to that characterizing the other representative samples $\{\tilde{\mathbf{x}}^1, \cdots, \tilde{\mathbf{x}}^K\}$. Such an expansion mechanism enables $\mathcal{M}_i^D$ to capture a diversity of information over time while considering the constraints of a compact memory.

Optimizing the Memory via Online Optimal Transport

Various distance measures can be used for implementing $f_d(\cdot,\cdot)$ in Eq. (2), including the Kullback–Leibler (KL) divergence, total variation, and the L2 distance. In this paper, we consider employing the optimal transportation distance for several reasons. Firstly, the optimal transportation distance has been a popular distance measure in machine learning, (Courty et al. 2016; Villani 2009). The optimal transportation distance provides a good convergence for matching two corresponding probabilistic data representations, (Arjovsky, Chintala, and Bottou 2017; Gulrajani et al. 2017). Secondly, we theoretically demonstrate that the optimal transportation distance used in the proposed approach can achieve a small target risk. In the following, we consider a more specific learning setting for the theoretical analysis in which the class labels are given.

Definition 1 ***(Data distributions.)*** *Let Ω and $\mathcal{C}$ denote an input measurable space of dimension d and the set of class labels. For a given data stream U, we have seen i data batches $\{(\mathbf{X}_1, \mathbf{Y}_1), \cdots, (\mathbf{X}_i, \mathbf{Y}_i)\}$ at the time step ($\mathcal{S}_i$). Let $\mathbb{P}_i$ be a data distribution of all previous data batches $\{(\mathbf{X}_1, \mathbf{Y}_1), \cdots, (\mathbf{X}_i, \mathbf{Y}_i)\}$. In addition, we also define the empirical version of $\mathbb{P}_i$ as :*

$$\tilde{\mathbb{P}}_i = \frac{1}{N_{\tilde{\mathbb{P}}_i}} \sum\nolimits_{j=1}^{N_{\tilde{\mathbb{P}}_i}} \{\delta_{\mathbf{x}_j, y_j}\}\,, (\mathbf{x}_j, y_j) \;\sim \mathbb{P}_i \,, \tag{3}$$

where $N_{\tilde{\mathbb{P}}_i}$ is the total number of data samples. $\mathbf{X}_i$ and $\mathbf{Y}_i$ denote the i-th labelled data batch.

Definition 2 ***(The proposed memory system.)*** *Let $\mathbb{P}_{\mathcal{M}_i^T}$ and $\mathbb{P}_{\mathcal{M}_i^D(j)}$ denote the distribution of the temporary memory buffer $\mathcal{M}_i^T$ and the j-th memory block $\mathcal{M}_i^D(j)$ at $\mathcal{S}_i$. Let $\mathbb{P}_{\mathcal{M}_i^T} \otimes \mathbb{P}_{\mathcal{M}_i^D(1)} \cdots \mathbb{P}_{\mathcal{M}_i^D(K)}$ represent the distribution of the combined memory buffers $\{\mathcal{M}_i^T, \mathcal{M}_i^D(1), \cdots, \mathcal{M}_i^D(K)\}$.*

Definition 3 ***(Error function.)*** *Let $\mathcal{H}$ be a class of classifiers and f is one of them. The model's risk for a given data distribution $\mathbb{P}_i$ is defined as :*

$$\mathcal{E}(f, \mathbb{P}_i) = \mathbb{E}_{(\mathbf{x},y)\sim\mathbb{P}_i}[\mathcal{L}(y, f(\mathbf{x}))]\,, \tag{4}$$

where $\mathcal{L}(\cdot,\cdot)$ is a bounded, symmetric and k-lipschitz loss function, which satisfies the triangle inequality and $f(\mathbf{x})$ represents the prediction of $\mathbf{x}$ made by the classifier f.

Inspired by the domain adaptation theory (Courty et al. 2017), we derive a new theory framework to analyze the performance and provide a theoretical explanation for the proposed memory approach.

Theorem 1 ***(The generalization bound.)*** *Let us define $\mathrm{W}_1(\tilde{\mathbb{P}}_i, \tilde{\mathbb{P}}_{\mathcal{M}_i})$ as the 1-Wasserstein distance between two distributions, which is one of the optimal transportation distances. Let $\mathcal{L}$ be a symmetric loss function, which satisfies the k-Lipschitz and the triangle inequality. We assume that the given data are bounded $|f^*(\mathbf{x}_1) - f^*(\mathbf{x}_2)| \leq M'$ for all $\mathbf{x}_1, \mathbf{x}_2$. Then, there exists c' for $\lambda' > 0$ and a risk bound for a classifier trained using the proposed memory approach at $\mathcal{S}_i$ is derived with the probability of at least $1 - \delta$:*

$$\begin{aligned} \mathcal{E}(f, \tilde{\mathbb{P}}_i) \geq\; & \mathrm{W}_1(\tilde{\mathbb{P}}_i, \tilde{\mathbb{P}}_{\mathcal{M}_i^T} \otimes \tilde{\mathbb{P}}_{\mathcal{M}_i^D(1)} \cdots \tilde{\mathbb{P}}_{\mathcal{M}_i^D(K)}) \\ & + \mathcal{E}(f^*, \mathbb{P}_{\mathcal{M}_i^T} \otimes \mathbb{P}_{\mathcal{M}_i^D(1)} \cdots \mathbb{P}_{\mathcal{M}_i^D(K)}) \\ & + \sqrt{\frac{2}{c'} \log(\frac{2}{\delta})}(\frac{1}{\sqrt{N_{\tilde{\mathbb{P}}_i}}} + \frac{1}{\sqrt{N_{\tilde{\mathbb{P}}_{\mathcal{M}_i^T} \otimes \tilde{\mathbb{P}}_{\mathcal{M}_i^D(1)} \cdots \tilde{\mathbb{P}}_{\mathcal{M}_i^D(K)}}}}) \\ & + kM'\zeta(\lambda') + \mathcal{E}(f^*, \mathbb{P}_i)\,. \end{aligned} \tag{5}$$

The last term $\zeta(\lambda')$ from Eq. (5) *evaluates the probability under which the probabilistic Lipschitzness does not hold (Courty et al. 2017). $f^* \in \mathcal{H}$ denotes a Lipschitz labeling function, which satisfies the ζ-probabilistic transfer Lipschitzness assumption with respect to $\kappa*$ and minimizes the joint error $\mathcal{E}(f^*, \mathbb{P}_i) + \mathcal{E}(f^*, \mathbb{P}_{\mathcal{M}_i^T} \otimes \mathbb{P}_{\mathcal{M}_i^D(1)} \cdots \mathbb{P}_{\mathcal{M}_i^D(K)})$, where $\kappa*$ is defined as :*

$$\begin{aligned} \kappa* = \arg\min\nolimits_{\kappa \in \kappa(\mu_i, \mu_{\mathcal{M}_i})} & \int_{(\Omega\times\mathcal{C})^2} \alpha \mathrm{d}(\mathbf{x}_i, \mathbf{x}_{\mathcal{M}_i}) \\ & + \mathcal{L}(y_i, y_{\mathcal{M}_i}) \mathrm{d}\kappa(\mathbf{x}_i, y_i; \mathbf{x}_{\mathcal{M}_i} y_{\mathcal{M}_i})\,, \end{aligned} \tag{6}$$

where $\kappa(\mu_i, \mu_{\mathcal{M}_i}) = \{\gamma \,|\, p^+ \# \gamma = \mu_i, p^- \# \gamma = \mu_{\mathcal{M}_i}\}$. $\#$ denotes that γ transports to μ_i and $\mu_{\mathcal{M}_i}$ through two marginal projections p^+ and p^-. μ_i and $\mu_{\mathcal{M}_i}$ are the marginal distribution of $\mathbb{P}_i$ and $\mathbb{P}_{\mathcal{M}_i}$ over Ω, respectively. The detailed proof is provided in Appendix-D from SM[1].

Theorem 1 indicates that minimizing the distance term $\mathrm{W}_1(\tilde{\mathbb{P}}_i, \tilde{\mathbb{P}}_{\mathcal{M}_i^T} \otimes \tilde{\mathbb{P}}_{\mathcal{M}_i^D(1)} \cdots \tilde{\mathbb{P}}_{\mathcal{M}_i^D(K)})$ in the right-hand-side of Eq. (5) can lead to a small target error (the left-hand-side of Eq. (5)). However, directly minimizing this distance term in the algorithm implementation is impossible because we cannot access the entire distribution $\mathbb{P}_i$ during

continual learning. Instead, the proposed approach achieves this goal by promoting the knowledge discrepancy among memory blocks using Eq. (2), with the aim of allowing $\tilde{\mathbb{P}}_{\mathcal{M}_i^T} \otimes \tilde{\mathbb{P}}_{\mathcal{M}_i^D(1)} \cdots \tilde{\mathbb{P}}_{\mathcal{M}_i^D(K)}$ to preserve the past information as much as possible, helping to approximate $\tilde{\mathbb{P}}_i$ closely over time. This theoretical result is empirically demonstrated in Fig. 3b, showing that each newly constructed memory block distribution $\tilde{\mathbb{P}}_{\mathcal{M}_i^D(j)}$ is built when a unique data distribution is presented. We provide the theoretical analysis in Appendix-D from SM[1] showing that the proposed model expansion mechanism can further improve the performance.

Based on the results of **Theorem 1**, we implement $f_d(\cdot,\cdot)$ using Wasserstein Distance (WD) in Eq. (2). However, estimating WD is time-consuming in the high-dimensional space (Cuturi 2013). This paper proposes the Sinkhorn-based Dual Dynamic Memory (SDDM) method, by considering Sinkhorn distance (Cuturi 2013; Frogner, Mirzazadeh, and Solomon 2019) as $f_d(\cdot,\cdot)$. The Sinkhorn distance is an optimal transportation distance, which shares similar theoretical properties with the WD, expressed as :

$$f_{M,\alpha}(\mathbf{r},\mathbf{c}) := \min_{P \in U_\alpha(\mathbf{r},\mathbf{c})} \langle P, M \rangle \ , \tag{7}$$

where $U_\alpha(\mathbf{r},\mathbf{c})$ is the convex set, defined as :

$$\begin{aligned} U_\alpha(\mathbf{r},\mathbf{c}) &:= \{P \in U(\mathbf{r},\mathbf{c}) \,|\, \mathbf{KL}(P \,||\, \mathbf{r}\mathbf{c}^{\mathrm{T}}) \leq \alpha\} \\ &= \{P \in U(\mathbf{r},\mathbf{c}) \,|\, h'(P) \geq h'(\mathbf{c}) + h'(\mathbf{c}) - \alpha\} \\ &\subset U(\mathbf{r},\mathbf{c}) \,, \end{aligned} \tag{8}$$

where $\mathbf{KL}(P \,||\, \mathbf{r}\mathbf{c}^T) = h'(\mathbf{c}) + h'(\mathbf{r}) - h'(P)$ is the Kullback-Leibler (KL) divergence. $U(\mathbf{r},\mathbf{c})$ is the transport polytope of $\mathbf{c}$ and $\mathbf{r}$, expressed as $U(\mathbf{r},\mathbf{c}) := \{P \in \mathbb{R}_+^{d\times d}, P\mathbf{I}_d = \mathbf{r}, P^{\mathrm{T}}\mathbf{I}_d = \mathbf{c}\}$. $\mathbf{r}$ and $\mathbf{c}$ are two probability vectors and $h'(\cdot)$ is the entropy function. P is a joint probability and M is a cost matrix mapping of $\mathbf{r}$ to $\mathbf{c}$. We also use T as the superscript to denote the transposition of a matrix. Based on the Sinkhorn distance, defined in Eq. (7), we replace $f_d(\cdot)$ in Eq. (2), using $f_{M,\alpha}(\cdot,\cdot)$, resulting in :

$$\begin{aligned} \min \{&f_{M,\alpha}(\tilde{\mathbf{x}}^1, \mathbf{x}_{(i,h)}), f_{M,\alpha}(\tilde{\mathbf{x}}^2, \mathbf{x}_{(i,h)}), \cdots , \\ &f_{M,\alpha}(\tilde{\mathbf{x}}^t, \mathbf{x}_{(i,h)})\} > \lambda \,. \end{aligned} \tag{9}$$

We compute $f_{M,\alpha}(\cdot,\cdot)$ using the matrix scaling algorithm (Cuturi 2013), evaluating the Sinkhorn distance for a pair of data samples. Compared to the Wasserstein distance, the Sinkhorn distance is faster to compute and is straight forward to be evaluated on high-dimensional spaces. A large λ value in Eq. (9) favours expanding $\mathcal{M}^D$ slowly while a small threshold λ encourages frequently expanding $\mathcal{M}^D$ during the optimization process. The choice λ in Eq. (9) is determined experimentally, considering $\lambda \in [0, 60]$.

The Dual Dynamic Memory Allocation Mechanism

When deploying the proposed SDDM memory approach on a resource-constrained device, it is imperative to restrict the maximum memory size, defined as $|\mathcal{M}|_{max}$. In the initial stages of continuous learning, we assign a larger temporary memory buffer capacity, $\mathcal{M}_i^T$, to store as many recent examples as possible, which then can provide enough information for training. In subsequent learning, we gradually assign more capacities for the dynamic expansion memory $\mathcal{M}_i^D$ while reducing the temporary memory size to avoid memory overload, as shown in Fig. 2. This mechanism helps preserve diverse and critical data samples over time by employing a fixed-size memory capacity.

In the following, we describe the updating process of the dynamic expansion memory $\mathcal{M}_i^D$. At the i-th training session (time), we select $c^\star$, as the memory block index for storing the new data $\mathbf{x}_{new}$:

$$c^\star = \arg\min_{c'=1,\cdots,K} \{f_d(\tilde{\mathbf{x}}^{c'}, \mathbf{x}_{new})\} \,. \tag{10}$$

We add $\mathbf{x}_{new}$ to $\mathcal{M}_i^D(c^\star)$ at $\mathcal{S}_i$, if this memory buffer did not reach its maximum capacity $\hat{a}$. Otherwise, we add a new sample $\mathbf{x}_{new}$ to the temporary memory buffer $\mathcal{M}_i^T$. We also remove the earliest memorized samples from the temporary memory buffer when the overall memory is overloaded, $|\mathcal{M}_i^T| + |\mathcal{M}_i^D| > |\mathcal{M}|_{max}$, where $|\mathcal{M}_i^T|$ and $|\mathcal{M}_i^D|$ denote the number of samples for $\mathcal{M}_i^T$ and $\mathcal{M}_i^D$, respectively.

The Dynamic Expandable Generative Model

Current OTFCL models are designed for supervised learning (Aljundi et al. 2019a; Aljundi, Kelchtermans, and Tuytelaars 2019), while unsupervised generative modeling is less explored. In this section, we address the online continual unsupervised generative modeling by considering enabling the Denoising Diffusion Probabilistic Model (DDPM) (Croitoru et al. 2023; Ho, Jain, and Abbeel 2020) with SDDM under the OTFCL paradigm. We consider an improved DDPM model (Nichol and Dhariwal 2021) and train it using the proposed SDDM :

$$\begin{aligned} \mathcal{L}_{\mathrm{SDDM}} = \lambda^\star \mathbb{E}_{q(\hat{\mathbf{x}}_{0:T} \,|\, \hat{\mathbf{x}}_0)} \Big[&- \log p_\theta(\hat{\mathbf{x}}_0 \,|\, \hat{\mathbf{x}}_1) \\ &+ \sum\nolimits_{t=1}^{T-2} \{\mathcal{L}_t\} + D_{KL}[q(\hat{\mathbf{x}}_T \,|\, \hat{\mathbf{x}}_0) \,||\, p_\theta(\hat{\mathbf{x}}_T)] \Big] \\ &+ \mathbb{E}_{t,\hat{\mathbf{x}}_0,\boldsymbol{\epsilon}} \Big[\|\boldsymbol{\epsilon} - \boldsymbol{\epsilon}_\theta(\sqrt{\hat{\alpha}_t}\hat{\mathbf{x}}_0 + \sqrt{1-\hat{\alpha}_t}\boldsymbol{\epsilon}, t)\|^2 \Big] \ , \end{aligned} \tag{11}$$

where T is the total number of diffusion steps. $\mathcal{L}_t$ is defined as $\mathcal{L}_t = D_{KL}\big[q(\hat{\mathbf{x}}_t \,|\, \hat{\mathbf{x}}_{t+1}, \hat{\mathbf{x}}_0) \,||\, p_\theta(\hat{\mathbf{x}}_t \,|\, \hat{\mathbf{x}}_{t+1})\big]$ and $q(\hat{\mathbf{x}}_{1:T} \,|\, \hat{\mathbf{x}}_0) = \prod_{t=1}^T q(\hat{\mathbf{x}}_t \,|\, \hat{\mathbf{x}}_{t-1})$. $\hat{\mathbf{x}}_0$ represents a real sample obtained from the proposed memory $\{\mathcal{M}_i^D, \mathcal{M}_i^T\}$ at S_i, and $\hat{\mathbf{x}}_T$ denotes a noise vector drawn from a normal distribution $\mathcal{N}(\mathbf{0}, \mathbf{I})$. $q(\tilde{\mathbf{x}}_T \,|\, \tilde{\mathbf{x}}_0)$ and $p_\theta(\hat{\mathbf{x}}_0 \,|\, \hat{\mathbf{x}}_1)$ represent the distributions defined during the forward and backward diffusion process, respectively. $\boldsymbol{\epsilon}_\theta(\cdot)$ is a noise estimator parameterized by θ and λ^* is a hyperparameter (we use the default value from (Nichol and Dhariwal 2021)). $p_\theta(\hat{\mathbf{x}}_T)$ is a normal distribution $\mathcal{N}(\mathbf{0}, \mathbf{I})$.

The Dynamic Expandable SDDM (SDDM-Dyn). One major weakness of training the DDPM model is its fixed capacity, which does not allow the model to deal with learning an infinite data stream. We address this issue by proposing a novel dynamic expansion model, aiming to gradually increase the model's capacity to accumulate new information over time. Specifically, we treat each noise estimator $\boldsymbol{\epsilon}_{\theta_j}$ as

Datasets	SDDM	SDDM-Dyn	LTS	LGM	R-VAE	R-DDPM	CGKD-GAN	CGKD-DDPM	CGKD-WAE	CGKD-VAE	CVA
Split MNIST	23.36	18.48	71.67	66.31	55.67	63.26	54.34	30.71	47.98	48.72	21.46
Split Fashion	47.43	42.76	128.84	109.20	103.25	82.23	85.23	51.21	87.92	88.16	67.28
Split SVHN	55.16	51.21	87.25	72.60	65.18	87.22	101.26	63.52	100.15	102.87	57.14
Split CIFAR10	79.83	73.52	124.22	177.15	155.72	106.18	115.38	80.15	162.12	163.75	74.97
Average	51.44	**46.49**	102.99	106.31	94.95	84.72	89.05	56.39	99.54	100.87	55.21

Table 1: Evaluation of the mage generation performance using FID for class-incremental learning.

an expert in a mixture system $\mathcal{A}$, where θ_j represents the parameter set of j-th expert, aiming to learn different information from the other experts. To implement this goal, we initially consider the first expert $\boldsymbol{\epsilon}_{\theta_1} \in \mathcal{A}$, and then, when enough data are gathered, gradually add new experts which are then trained with the given data. We assume that the mixture system $\mathcal{A}$ has already learnt K' experts at S_{i-1} and we introduce a novel dynamic expansion mechanism that evaluates the knowledge discrepancy between each previously learnt expert and the currently updated one $\boldsymbol{\epsilon}_{\theta_{K'}}$, at the new training session (S_i) :

$$\begin{aligned} \min \{ & f'(\mathbb{P}_{\mathbf{x}^1}, \mathbb{P}_{\mathbf{x}^{K'}}), f'(\mathbb{P}_{\mathbf{x}^2}, \mathbb{P}_{\mathbf{x}^{K'}}), \cdots, \\ & f'(\mathbb{P}_{\mathbf{x}^{K'-1}}, \mathbb{P}_{\mathbf{x}^{K'}}) \} > \lambda_d \,, \end{aligned} \tag{12}$$

where $\mathbb{P}_{\mathbf{x}^j}, j = 1, \cdots, K'$ represent data distributions, each formed by the data samples generated by the j-th expert $\boldsymbol{\epsilon}_{\theta_j}$, which represents the knowledge learnt previously by $\boldsymbol{\epsilon}_{\theta_j}$. $f'(\cdot)$ is a probabilistic distance measure and $\lambda_d \in [0, 100]$ is an architecture expansion threshold. However, evaluating Eq. (12) requires performing multiple sampling processes each time, by each expert, leading to considerable computational costs. We address this issue by assigning a tiny memory buffer $\tilde{\mathcal{M}}_j$ to store a few data samples from the SDDM-Dyn's memory $\mathcal{M}^D \bigcup \mathcal{M}^T$, instead of using diffusion models for generating data. We also implement $f'(\cdot)$ in Eq. (12) using $f_{M,\alpha}(\cdot, \cdot)$ and Eq. (12) is reformulated as :

$$\begin{aligned} \min \{ & f_{M,\alpha}(\tilde{\mathcal{M}}_1, \tilde{\mathcal{M}}_{K'}), \cdots, \\ & f_{M,\alpha}(\tilde{\mathcal{M}}_{K'-1}, \tilde{\mathcal{M}}_{K'}) \} > \lambda_d \,. \end{aligned} \tag{13}$$

If Eq. (13) is satisfied at S_i, then we add a new expert $\boldsymbol{\epsilon}_{\theta_{K+1}}$ and freeze the previous one, $\boldsymbol{\epsilon}_{\theta_K}$. We also clear up the SDDM-Dyn memory system $\mathcal{M}^D \bigcup \mathcal{M}^T$ to ensure that the newly created expert $\boldsymbol{\epsilon}_{\theta_{K+1}}$ learns distinct information during the next training steps.

Algorithm Framework

The pseudocode used for implementing the SDDM memory system with the dynamic model mechanism is provided in **Algorithm 2** in the **Appendix A** from SM[1]. Each training session (time) consists of three steps :

Step 1 (Check memory expansion). If the memory buffer $\mathcal{M}^D$ at the initial training phase does not store any data samples, we create the first memory block $\mathcal{M}_1^D(1)$ and add the incoming data batch $\mathbf{X}_1$ to $\mathcal{M}_1^D(1)$. Then, we select a representative sample $\tilde{\mathbf{x}}^1$ through :

$$h^\star = \underset{h=1,\cdots,b}{\operatorname{argmin}} \Big\{ \sum\nolimits_{j=1, j\neq h}^{b} \{ f_{M,\alpha}(\mathbf{x}_{(1,j)}, \mathbf{x}_{(1,h)}) \} \Big\} , \tag{14}$$

Methods	Split CIFAR10	Split TI	P-MNIST
GSS	49.73 ± 4.78	-	76.00 ± 0.87
DER	70.51 ± 1.67	17.75 ± 1.14	87.29 ± 0.46
DER++	72.70 ± 1.36	19.38 ± 1.41	88.21 ± 0.39
DER+++refresh	74.42 ± 0.82	20.81 ± 1.28	-
SDDM	**74.63** ± 1.32	**21.36** ± 1.17	**90.39** ± 0.41

Table 2: Average classification accuracy on continual learning benchmarks, considering 10 runs for various models.

where $\mathbf{x}_{(1,j)}$ is the j-th sample from the first data batch $\mathbf{X}_1 = \{\mathbf{x}_{(1,1)}, \cdots, \mathbf{x}_{(1,b)}\}$ at S_1, and $h^\star$ is the optimal sample index. The first representative sample $\tilde{\mathbf{x}}^1$ for the memory block $\mathcal{M}_1^D(1)$ is determined by $\mathbf{x}_{(1,h^\star)}$ which has the smallest distance to the other data samples. During the subsequent training phase, if the memory expansion criterion from Eq. (9) is satisfied, then we add new representative samples to the memory block in $\mathcal{M}^D$.

Step 2 (Memory allocation mechanism). For each new data sample $\mathbf{x}_{new}$ at S_i, we perform the memory block selection using Eq. (10). If the selected memory block is not full, we add $\mathbf{x}_{new}$, otherwise, we add $\mathbf{x}_{new}$ to the temporary memory $\mathcal{M}_i^T$. If the overall memory $\mathcal{M}_i^T$ is overloaded, we reduce the temporary memory size until $|\mathcal{M}_i^D| + |\mathcal{M}_i^T| \leq |\mathcal{M}|_{max}$.

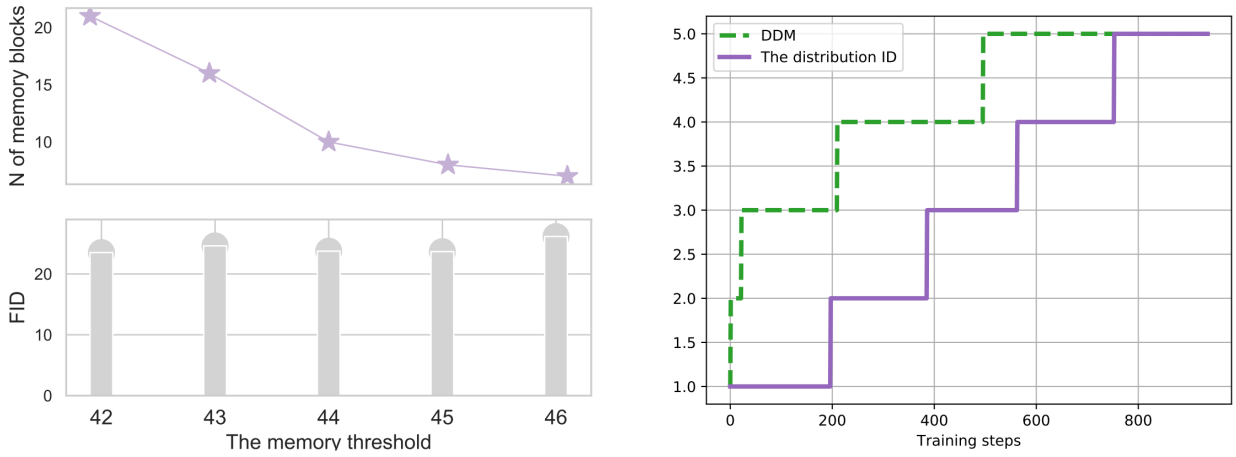
Step 3 (Check the model expansion). Since the proposed dynamic expansion mechanism requires at least two experts for the expansion evaluation, we automatically create the second expert $\boldsymbol{\epsilon}_{\theta_2}$ when the memory buffer is full, $|\mathcal{M}_i^D| + |\mathcal{M}_i^T| = |\mathcal{M}|_{max}$. During subsequent learning, we dynamically add a new expert into $\mathcal{A}$ when the model expansion criterion from Eq. (13) is fulfilled.

Experiments

We provide the detailed experiment setting in Appendix-E from SM[1].

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, CIFAR10, as in (Aljundi, Kelchtermans, and Tuytelaars 2019), using a memory buffer of 2,000 data samples. The number of training epochs for each training session (time) is of 6 for all models and the FID score is calculated on 5,000 test-



(a) Changing λ in Eq. (9).

(b) Memory clusters for $\mathcal{M}^D$.

Figure 3: Ablation results for SDDM. (a) FID results (bottom plot) and the resulting number of memory blocks (top) when varying λ. (b) The number of memory clusters for $\mathcal{M}^D$ and the data distribution (task) when $\lambda = 46$.

Methods	Resolution	CelebA-HQ	CACD	FFHQ
SDDM	$128 \times 128 \times 3$	**98.25**	**94.62**	**99.57**
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
SDDM	$256 \times 256 \times 3$	**101.53**	**104.28**	**99.75**
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: Assessing image generation performance using FID for datasets containing high-resolution images.

ing samples after the whole training process is completed. The final hyperparameter λ for Split MNIST, Split Fashion, Split SVHN and Split CIFAR10 is 44, 44, 43 and 44, respectively. We report the performance of various models in Tab. 1. The results indicate that the dynamic expansion models outperform the generative replay-based methods. The proposed SDDM approach achieves better performance than dynamic expansion models, such as CGKD-GAN. Furthermore, the performance is improved by using the proposed dynamic expansion model, SDDM-Dyn. The hyperparameter λ_d from Eq. (13) is 32, 33, 32 and 33, for Split MNIST, Split Fashion, Split SVHN and Split CIFAR10, respectively.

Continual Classification Tasks Learning

In addition to unsupervised learning, the proposed approach can also be used for supervised learning without any modifications. We follow the experiment setting from the standard benchmark (Buzzega et al. 2020) in which the maximum memory size is 500 for the Split CIFAR10, Split TinyImageNet (Split TI) and P-MNIST. We employ the proposed memory approach to learn a classifier with the objective function of the DER++ (Buzzega et al. 2020) and the results of various models are shown in Tab. **??**. The empirical results show that the proposed approach outperforms other baselines as well as the current state-of-the-art method DER+++refresh (Wang et al. 2024), demonstrating the effectiveness of the proposed SDDM in supervised learning.

Ablation Study

We train the proposed SDDM on Split MNIST, considering various thresholds λ in Eq. (9) for expanding the dynamic memory buffers and the results are presented in Fig. 3a. A small threshold λ encourages adding additional memory blocks while a large threshold leads to the opposite. In addition, the proposed SDDM can still achieve good performance even if it does not create more memory blocks. We also train the proposed SDDM on Split MNIST considering $\lambda = 46$ and we plot the number of memory blocks and distributions in Fig. 3b. We observe that the proposed SDDM can appropriately create a new memory block when facing data distribution changes. The results show that the temporary memory can provide sufficient samples for managing the memory assignment mechanism during the dynamic expansion memory of the SDDM. We provide more ablation results in the Appendix-F from SM[1].

Conclusion

This paper addresses the online unsupervised generative modeling by proposing the Sinkhorn-based Dual Dynamic Memory (SDDM) method, which manages two memory systems to preserve both short-term and long-term information. The proposed memory expansion approach uses the Sinkhorn optimal transport algorithm for data selection, which ensures sample diversity while using a compact memory capacity without requiring any supervised signals. We also introduce a novel dynamic model, which expands the model's architecture to further improve performance.

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.

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

Bang, J.; Kim, H.; Yoo, Y.; Ha, J.-W.; and Choi, J. 2021. Rainbow Memory: Continual Learning with a Memory of Diverse Samples. In *Proc. of IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 8218–8227.

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 the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 9275–9284.

Buzzega, P.; Boschini, M.; Porrello, A.; Abati, D.; and Calderara, S. 2020. Dark experience for general continual learning: a strong, simple baseline. *Advances in Neural Information Processing Systems (NeurIPS)*, 33: 15920–15930.

Chrysakis, A.; and Moens, M.-F. 2020. Online continual learning from imbalanced data. In *Proc. International Conference on Machine Learning (ICML)*, 1952–1961. PMLR 119.

Courty, N.; Flamary, R.; Habrard, A.; and Rakotomamonjy, A. 2017. Joint distribution optimal transportation for domain adaptation. *Advances in Neural Information Processing Systems (NeurIPS)*, 30: 3733–3742.

Courty, N.; Flamary, R.; Tuia, D.; and Rakotomamonjy, A. 2016. Optimal transport for domain adaptation. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 39(9): 1853–1865.

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.

Cuturi, M. 2013. Sinkhorn distances: Lightspeed computation of optimal transport. *Advances in Neural Information Processing Systems (NIPS)*, 26: 2292–2300.

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.

Dosovitskiy, A.; Beyer, L.; Kolesnikov, A.; Weissenborn, D.; Zhai, X.; Unterthiner, T.; Dehghani, M.; Minderer, M.; Heigold, G.; Gelly, S.; Uszkoreit, J.; and Houlsby, N. 2021. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations (ICLR), arXiv preprint arXiv:2010.11929*.

Frogner, C.; Mirzazadeh, F.; and Solomon, J. 2019. Learning embeddings into entropic Wasserstein spaces. In *International Conference on Learning Representations (ICLR), arXiv preprint arXiv:1905.03329*.

Gulrajani, I.; Ahmed, F.; Arjovsky, M.; Dumoulin, V.; and Courville, A. C. 2017. Improved training of Wasserstein GANs. In *Advances in Neural Inf. Proc. Systems (NIPS)*, 5767–5777.

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

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep residual learning for image recognition. In *Proc. of IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 770–778.

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

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.

Izquierdo, I.; Medina, J. H.; Izquierdo, L. A.; Barros, D. M.; de Souza, M. M.; and e Souza, T. M. 1998. Short-and long-term memory are differentially regulated by monoaminergic systems in the rat brain. *Neurobiology of learning and memory*, 69(3): 219–224.

Jiang, Z.; Zhang, J.; and Gong, N. Z. 2023. Evading watermark based detection of AI-generated content. In *Proc. of ACM SIGSAC Conference on Computer and Communications Security*, 1168–1181.

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.

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*.

Liang, Y.-S.; and Li, W.-J. 2024. Loss decoupling for task-agnostic continual learning. *Advances in Neural Information Processing Systems (NeurIPS)*, 36: 11151–11167.

Lin, K.; Yang, H. F.; Hsiao, J. H.; and Chen, C. S. 2015. Deep learning of binary hash codes for fast image retrieval. In *Proc. of IEEE CVPR-Workshops*, 27–35.

Lyu, Y.; Wang, L.; Zhang, X.; Sun, Z.; Su, H.; Zhu, J.; and Jing, L. 2024. Overcoming recency bias of normalization statistics in continual learning: Balance and adaptation. *Advances in Neural Information Processing Systems (NeurIPS)*, 36: 25475–25494.

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.

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.

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.

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

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*.

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.

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

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 the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 24214–24223.

Villani, C. 2009. *Optimal transport: old and new*, volume GL 338. Springer.

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.; Li, Y.; Shen, L.; and Huang, H. 2024. A Unified and General Framework for Continual Learning. In *International Conference on Learning Representations (ICLR), arXiv preprint arXiv:2403.13249*.

Wu, J.; Yu, Y.; Huang, C.; and Yu, K. 2015. Deep multiple instance learning for image classification and auto-annotation. In *Proc. of IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 3460–3469.

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. 2023. Continual Variational Autoencoder via Continual Generative Knowledge Distillation. In *Proc. of the AAAI Conference on Artificial Intelligence*, volume 37, 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.

Zhai, M.; Chen, L.; Tung, F.; He, J.; Nawhal, M.; and Mori, G. 2019. Lifelong GAN: Continual Learning for Conditional Image Generation. In *Proc. of the IEEE/CVF Int. Conf. on Computer Vision (ICCV)*, 2759–2768.

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.