Wasserstein Expansible Variational Autoencoder for Discriminative and Generative Continual Learning

Fei Ye and Adrian G. Bors
Department of Computer Science, University of York, York YO10 5GH, UK
fy689@york.ac.uk, adrian.bors@york.ac.uk

Abstract

Task-Free Continual Learning (TFCL) represents a challenging learning paradigm where a model is trained on the non-stationary data distributions without any knowledge of the task information, thus representing a more practical approach. Despite promising achievements by the Variational Autoencoder (VAE) mixtures in continual learning, such methods ignore the redundancy among the probabilistic representations of their components when performing model expansion, leading to mixture components learning similar tasks. This paper proposes the Wasserstein Expansible Variational Autoencoder (WEVAE), which evaluates the statistical similarity between the probabilistic representation of new data and that represented by each mixture component and then uses it for deciding when to expand the model. Such a mechanism can avoid unnecessary model expansion while ensuring the knowledge diversity among the trained components. In addition, we propose an energy-based sample selection approach that assigns high energies to novel samples and low energies to the samples which are similar to the model's knowledge. Extensive empirical studies on both supervised and unsupervised benchmark tasks demonstrate that our model outperforms all competing methods. The code is available at https://github.com/dtuzi123/WEVAE/.

1. Introduction

The variational Autoencoder (VAE) [33] is one of the most popular deep generative models, underlined by a symmetric network structure in which an input $\mathbf{x}$ is transferred into a pseudo-similar data $\mathbf{x}'$ through an encoding-decoding process. Due to its powerful inference mechanisms, the VAE has been successfully used in many applications, including few-shot learning [54], semi-supervised learning [1], image synthesis [41], image-to-image translation [40], and density estimation [59]. However, using the VAE model in continual learning has not been sufficiently investigated so far. Similar to other deep neural networks in continual learning, VAEs suffer from a significant performance loss when continually learning new data domains, which is known as catastrophic forgetting [45].

Constructing a fixed-length memory buffer that stores some past samples and then replays them during the subsequent learning stages [14, 9] was shown to relieve catastrophic forgetting in continual learning. Meanwhile, other approaches focus on regulating the optimization procedure of the model by imposing a penalty term in the objective function for freezing certain parameters, [34, 39]. These approaches usually train a static model with a fixed capacity and cannot achieve good performance when learning a growing number of tasks [71]. The dynamic expansion model [52, 71] builds new hidden layers to handle incoming tasks, showing promising results in continual learning due to its scalability and generalization performance. However, despite their impressive performance in continual learning, most dynamic expansion methods still require the knowledge of the task boundaries, which limits their applicability in a more realistic scenario such as the Task-Free Continual Learning (TFCL) [6].

The dynamic expansible VAE framework was recently shown to provide good performance in TFCL, [48, 73]. The Continual Unsupervised Representation Learning (CURL) [48] dynamically builds new VAE inference models to capture the distribution shift over time. CURL relieves forgetting by retraining a generator (decoder) to reproduce past samples, inevitably leading to forgetting, [71]. A similar idea was proposed in [38], which uses the Dirichlet process for the component expansion in a mixture of VAEs in a method called the Continual Neural Dirichlet Process Mixture (CN-DPM). Unlike CURL, CN-DPM does not rely on the generative replay mechanism (GRM) [84] and therefore can preserve the best information for all previously learned samples. More recently, the dynamic VAE mixture model was upgraded by using the Online Cooperative Memorization (OCM) [73], which manages two cooperative memory buffers to preserve both the short- and long-term information. However, none of these models have theoretical guarantees for their expansion mechanisms. Moreover, they do not explicitly design a mechanism for ensuring the information diversity among the VAE mixture components.

In this paper, we propose a new approach called the Wasserstein Expansible Variational Autoencoder (WEVAE), aiming to learn a compact dynamic expansible VAE model for TFCL, which evaluates the knowledge correlation between the given information and the previously learned components, as an expansion criterion. The primary motivation for this expansion mechanism is to promote the knowledge diversity among mixture's components.

For ensuring a stable expansion process, we evaluate the Wasserstein distance for representing the dynamic change of the correlations between the currently trained and all previously learnt components over time. The proposed mechanism avoids the frequent expansion caused by outliers in the data, while ensuring the information diversity among components. In addition, autoencoders have naturally been employed in the Energy-Based Model (EBM) [85] aiming to learn an energy map in which the low energy is attributed to the data manifold [85]. Inspired by the EBM, we propose an energy-based sample selection approach, which aims to assign low energies to the samples that share similar information to that already learnt by the model.

Empirical validations show that the proposed Wasserstein Expansible Variational Autoencoder (WEVAE) can train statistically diverse components while outperforming the state-of-the-art using a compact structure. Our contributions consists of : (1) We propose a new model, WEVAE, for TFCL, which dynamically expands its capacity through the proposed Wasserstein expansion mechanism, ensuring the information diversity among components; (2) We formulate the dynamic expansion as the evaluation of time series data to avoid frequent expansion. This is the first work that employs the time series analysis for model expansion under TFCL; (3) We propose an energy-based sampling approach to manage the memory buffer, which can further promote knowledge diversity among model' s components. This is the first work employing the energy function for sample selection in TFCL; (4) We provide theoretical guarantees for the proposed expansion mechanism and analyze the forgetting behaviour of the proposed WEVAE.

2. Related Work

Continual learning. One popular and straightforward approach to relieve forgetting in continual learning (CL) is by managing a small-scale memory buffer that stores some past examples and replays them during subsequent learning [10, 12, 21, 22, 26, 30, 34, 35, 39, 44, 46, 49, 50, 51, 60, 69]. The memory buffer can be used in the regularization-based approaches for further improving the performance [32, 43, 7, 14, 13, 42, 18, 55, 66, 44, 3, 22, 83, 24, 28, 19, 17, 65]. Another approach in CL is by training a generator such as a VAE or a Generative Adversarial Network (GAN) [20] to produce past samples that are used for preventing forgetting [2, 47, 56, 84]. However, when learning a long sequence of tasks, a GAN can suffer from mode collapse [58] due to the frequent generative replay process [84]. To address this issue, recent works have developed dynamic expansion networks, which preserve the entire previously learnt information into frozen components while expanding the network to learn new tasks [15, 27, 39, 46, 48, 52, 67, 68, 86, 31, 62]. However, all these approaches still require knowing the task boundaries to evaluate the model expansion, which is not a realistic CL scenario, where the task information is missing.

Task-free continual learning. Several recent works have focused on the TFCL scenario, where the task information is unavailable. The memory-based approach was first explored in the context of TFCL [6] for training a classifier. Then, this approach was extended to the Maximal Interfered Retrieval (MIR), by using a new information retrieval mechanism that selectively stores the most representative samples to train both a classifier and VAEs [4]. More recently, the sample selection was implemented by comparing the gradient information between past and new samples, as in the Gradient Sample Selection (GSS) [7]. Furthermore, the sample selection approach can be implemented by a *learner-evaluator* evaluation framework, called the Continual Prototype Evolution (CoPE) [16], which ensures the balance replay and thus performs well in the context of the online unbalanced continual learning. Another approach was proposed to dynamically edit the stored samples, called the Gradient-based Memory EDiting (GMED) [29], where data samples are modified to increase the loss in the upcoming model updates. Although these approaches perform well on simple datasets, they are not scalable for learning an incoming long-term data stream due to their fixed model capacity. This inspired several attempts to apply the dynamic expansion model to TFCL [38, 48, 73, 76]. However, these approaches do not consider the repetitive learning of similar information when performing the expansion, leading to oversized model structures.

Continual generative modelling. Continual Generative modelling has been studied in several recent studies [47, 2, 73, 75, 72, 82, 74, 70, 81, 79, 77, 78, 71, 80]. The pioneering work in this direction consists in proposing a VAE-based continual learning framework [2], which learns both the task-specific and shared latent representations over time. Another approach is employing a teacher-student framework [47] where both the teacher and student are implemented by VAEs, teaching each other in order to accumulate knowledge over time. The teacher module can also be implemented by a more robust generative model, such as a GAN [20] [84]. These approaches still rely on the task information, which is not available in TFCL. Recently, some studies used dynamic expansion models under the TFCL scenarios [48, 73]. However, none of these works provides theoretical guarantees for its expansion mechanisms.

3. Methodology

Problem definition. Let $\mathcal{D}^T$ and $\mathcal{D}^S$ be the testing and training datasets, where $\mathbf{x} \in \mathcal{X} \sim \mathcal{D}^T$ is a data sample over the input space $\mathcal{X} \in \mathbb{R}^{d_x}$ and d_x is the dimension of the sample. In TFCL, a data stream $\mathcal{S}$ is usually defined by a series of data batches collected by $\mathcal{D}^S$ in a class-incremental manner [6], expressed as $\mathcal{S} = \bigcup_{i=1}^{n} \mathbf{X}_i$ where $\mathbf{X}_i \sim \mathcal{D}^T$ is the i-th data batch consisting of b samples $\{\mathbf{x}_1^i, \cdots, \mathbf{x}_b^i\}$ and n is the total number of data batches. The training goal in TFCL is to learn each data batch $\mathbf{X}_i$ at the i-th training time $\mathcal{T}_i$ while all previously seen data batches $\{\mathbf{X}_1, \cdots, \mathbf{X}_{i-1}\}$ are not available. The performance of the model is evaluated on the testing dataset $\mathcal{D}^S$.

Network architecture. We consider the expansible variational autoencoder for continual generative modelling [73]. Following from [73], we define an expansible mixture of VAEs, namely $\mathbf{G} = \{\mathcal{G}_1, \cdots, \mathcal{G}_k\}$ where each $\mathcal{G}_i$ represents the i-th VAE component in the mixture $\mathbf{G}$. Let $f_{\omega_i} : \mathcal{X} \to \mathcal{Z}$ and $f_{\theta_i} : \mathcal{Z} \to \mathcal{X}$ be an inference and decoder, respectively, where $\mathcal{Z}$ represents the latent space. We can define an encoding distribution $q_{\omega_i}(\mathbf{z} \,|\, \mathbf{x}) = \mathcal{N}(\mu_{\omega_i}(\mathbf{x}), \sigma^2_{\omega_i}(\mathbf{x})\mathbf{I})$, where $\mu_{\omega_i}(\mathbf{x})$ and $\sigma_{\omega_i}(\mathbf{x})$ are Gaussian hyperparameters given by $f_{\omega_i}(\mathbf{x})$ and $\mathbf{I}$ is the identity matrix. Then we define a decoding distribution $p_{\theta_i}(\mathbf{x} \,|\, \mathbf{z})$ implemented by the decoder f_{θ_i}. In TFCL, we propose only to train the current component $\mathcal{G}_i$, avoiding the negative knowledge transfer to previously learnt components, using the following VAE's objective function [33] :

$$\begin{aligned} \mathcal{L}_{ELBO}(\mathbf{x}; \mathcal{G}_i) = \mathbb{E}_{q_{\omega_i}(\mathbf{z} \,|\, \mathbf{x})} \left[\log p_{\theta_i} \left(\mathbf{x} \,|\, \mathbf{z} \right) \right] \\ - KL \left[q_{\omega_i} \left(\mathbf{z} \,|\, \mathbf{x} \right) \,||\, p \left(\mathbf{z} \right) \right] \,, \end{aligned} \tag{1}$$

where KL is the Kullback-Leibler (KL) divergence and the right-hand-side (RHS) expression represents a lower bound to the sample log-likelihood, called the Evidence Lower Bound (ELBO) [33].

3.1. Wasserstein Expansion Mechanism

Existing mixture VAE models do not consider the overlapping among the probabilistic data representations of their components when performing the model expansion [73]. Inspired by the theoretical analysis from Section 4.2, which shows that by encouraging the knowledge diversity between components can lead to a good performance in learning a diverse information while using a compact model structure, we propose a new expansion mechanism that evaluates the correlation between each previously trained and the currently learnt component, as expansion signals. Specifically, we expand the Wasserstein Expansible Variational Autoencoder (WEVAE) architecture after the current component learnt sufficiently novel knowledge with respect to what is know by the other components. Since each VAE component has its own generation process, we can evaluate the relevance of a pair of components using their generations $\mathcal{L}_d(\mathbb{P}_{\theta_i}, \mathbb{P}_{\theta_j})$, where $\mathcal{L}_d$ is an arbitrary distance measure and $\mathbb{P}_{\theta_i}$ is the distribution of samples produced by the generation process of $\mathcal{G}_i$. When training each new component $\mathcal{G}_k$ for WEVAE, we aim to minimize its knowledge overlap with all other components, $\mathbf{G} = \{\mathcal{G}_1, \cdots, \mathcal{G}_{k-1}\}$:

$$\begin{aligned} t^\star = \arg \max_{t=\mathcal{T}_{t_{k-1}}+1, \cdots, \mathcal{T}_n} \Big\{ \min \big\{ (\mathcal{L}_d(\mathbb{P}_{\theta_1^{t_1}}, \mathbb{P}_{\theta_k^t}), \cdots, \\ \mathcal{L}_d(\mathbb{P}_{\theta_{k-1}^{k-1}}, \mathbb{P}_{\theta_k^t}) \big\} \Big\} \,, \end{aligned} \tag{2}$$

where $t^\star$ is the index of the optimal training time $\mathcal{T}_{t^\star}$ required for maximizing the distance between $\mathcal{G}_k$ and other components. $\mathcal{T}_{t_{k-1}}$ is the required training time for the component $\mathcal{G}_{k-1}$ and $\mathbb{P}_{\theta_{k-1}^{t_{k-1}}}$ is the generator distribution formed by $\mathcal{G}_{k-1}$ at $\mathcal{T}_{t_{k-1}}$. Searching for the optimal solution in Eq. (2) is intractable since it requires accessing all training steps/times and data batches which are not available under the TFCL learning paradigm. To solve this issue, we formulate the optimization, defined by Eq. (2), as a dynamic expansion process of the WEVAE, where we continually add new components after the current one $\mathcal{G}_k$ has learnt sufficient novel information. To implement this goal, we first evaluate the minimum probabilistic distance between the current component and each previously learnt component as a time series data :

$$w_t = \min \big\{ \mathcal{L}_d(\mathbb{P}_{\theta_1^{t_1}}, \mathbb{P}_{\theta_k^t}), \cdots, \mathcal{L}_d(\mathbb{P}_{\theta_{k-1}^{t_{k-1}}}, \mathbb{P}_{\theta_k^t}) \big\} \,, \tag{3}$$

where $\mathcal{L}_d(\cdot, \cdot)$ is implemented using the Wasserstein distance (Earth-mover distance), which provides theoretical guarantees, according to our analysis from Section 4.2. Let $\mathbf{A}_t = \{w_1, \cdots, w_t\}$ be a set of historical time series samples generated using Eq. (3), forming a stochastic process updated at $\mathcal{T}_t$ and we consider $\mathbf{A}_t[j]$ to represent w_j. By employing the time series data, this paper proposes a simple but effective way to check the model's expansion at $\mathcal{T}_t$:

$$\frac{1}{t} \sum\nolimits_{j=1}^{t} \{\mathbf{A}_t[j]\} > \lambda \,, \tag{4}$$

where $\lambda \in [40, 100]$ is an expansion threshold balancing the model's performance and its size during the training. When Eq. (4) is satisfied, we add a new component to WEVAE while clearing up all historical time series data from $\mathbf{A}_t$.

3.2. Energy-based Sample Selection

The majority of sample selection approaches aim to store the information associated with older and newer tasks into memory buffers over time [9, 14]. However, the proposed WEVAE model has already accumulated prior knowledge in its parameters and using an approach as in other continual learning methods would lead to redundancy in the learnt information. In consequence, we propose a novel sample selection approach, aiming to enrich the information available

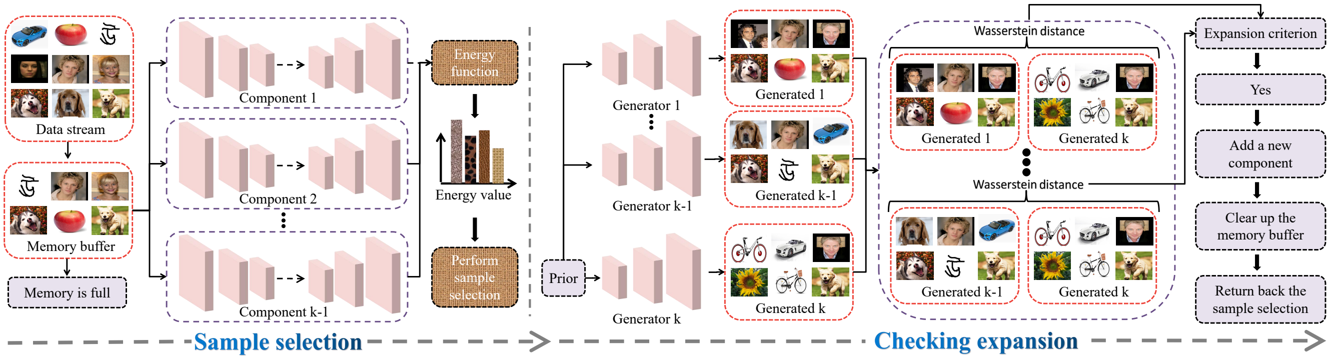


Figure 1. The learning procedure of the proposed WEVAE where we omit the training and testing procedures for simplification. In the sample selection process, we estimate the energy value for each memorized sample using all previously learnt components (‘Component 1’,$\cdots$, ‘Component k-1’), and results in updating the memory buffer $\mathcal{M}_t$, accordingly. During the training process, we only update the parameters of the current component (‘Component k’). If the memory buffer is full, we then check the model’s expansion using Eq. (4).

for training the next mixture component in order to capture different category information from what was already captured by the existing WEVAE’ components. To implement this goal, we propose to evaluate the selection score for each memorized sample by defining an evolved energy function for the i-th sample at $\mathcal{T}_t$ as :

$$E_{\mathcal{T}_t}(\mathbf{x}'_i) = \frac{1}{k-1}\sum\nolimits_{j=1}^{k-1}\left\{\mathrm{Rec}(\mathbf{x}'_i, f_{\theta_j}(f_{\omega_j}(\mathbf{x}'_i))\right\}, \quad (5)$$

where k is the number of components, and $\mathrm{Rec}(\cdot,\cdot)$ is the reconstruction error. $\mathbf{x}'_i$ is a sample drawn from the memory buffer $\mathcal{M}_t$ at time $\mathcal{T}_t$ and $f_{\theta_j}(f_{\omega_j}(\mathbf{x}'_i))$ is the reconstruction of a data sample $\mathbf{x}'_i$ using the j-th component $\mathcal{G}_j$. A higher energy value indicates that the data sample $\mathbf{x}'_i$ is novel to the already learnt knowledge and should remain in the memory buffer. Based on the energy function from Eq. (5), we update the memory buffer $\mathcal{M}_t$ at time $\mathcal{T}_t$:

$$\mathcal{M}_t = \bigcup\nolimits_{i=1}^{|\mathcal{M}_t|^{max}} \mathcal{M}'_t[i]\,, \quad (6)$$

where $|\mathcal{M}_t|^{max}$ is the maximum number of samples to be stored in $\mathcal{M}_t$ and $\mathcal{M}'_t$ is the sorted memory buffer satisfying $E_{\mathcal{T}_t}(\mathcal{M}'_t[a]) > E_{\mathcal{T}_t}(\mathcal{M}'_t[b]), a < b$. $\mathcal{M}'_t[a]$ denotes the a-th sample drawn from the sorted memory buffer.

3.3. Extension for the Prediction Task

Conditional VAE (CVAE) [57] is a popular probabilistic model which can be applied in classification tasks and guarantees a lower bound to the conditional log-likelihood. However, CVAE involves multiple neural networks, including the prior, recognition, and prediction networks, which require many parameters. In addition, CVAE can only decide the class $\mathbf{y}$, as in the prediction tasks, and does not reconstruct the data $\mathbf{x}$ as the proposed WEVAE model, which implements the generation process for each VAE component. Due to these limitations, in this study, we implement each VAE component by using a new probabilistic model $p(\mathbf{x}, \mathbf{y}, \mathbf{z})$ which treats the predictive label as the latent variable $\mathbf{y}$. We then derive an objective function for $p_\theta(\mathbf{x}, \mathbf{y}, \mathbf{z})$ (See details in **Appendix-A** from Supplemental Material (SM)), as :

$$\begin{aligned}\mathcal{L}_{class} &= \mathbb{E}_{q_{\omega,\varsigma}(\mathbf{z},\mathbf{y}|\mathbf{x})}\left[\log p_\theta(\mathbf{x}\mid\mathbf{z},\mathbf{y})\right] \\ &- KL(q_\omega(\mathbf{z}\,|\,\mathbf{x})\,||\,p(\mathbf{z})) - KL(q_\varsigma(\mathbf{y}\,|\,\mathbf{x})\,||\,p(\mathbf{y}))\,,\end{aligned} \quad (7)$$

where $p_\theta(\mathbf{x}\,|\,\mathbf{z},\mathbf{y})$ is the decoder and $q_\omega(\mathbf{z}\,|\,\mathbf{x})$ is the inference model that receives the input $\mathbf{x}$ and returns $\mathbf{z}$. $q_\varsigma(\mathbf{y}\,|\,\mathbf{x})$ is implemented by a classifier used for the prediction (classification) task. Each component in the WEVAE is implemented by using the model $p(\mathbf{x},\mathbf{z},\mathbf{y})$ which is trained to maximize Eq. (7). In addition, we also train $q_\varsigma(\mathbf{y}\,|\,\mathbf{x})$ on the memory buffer by using the cross-entropy loss to enhance its prediction performance. The learning procedure of WEVAE for both unsupervised and supervised settings is similar, as described in the following section.

3.4. Implementation

One popular approach to estimate the Wasserstein distance $\mathcal{L}_d(\cdot,\cdot)$ is by employing the Wasserstein GAN learning [8], which can guarantee a lower bound on the 1-Wasserstein distance [61]. However, such an approach requires additional discriminator and training processes, leading to additional computational costs. In this paper, we estimate the Wasserstein distance by using the results of Theorem 1 from [61], which is an efficient implementation not requiring extra computational costs :

$$\mathcal{L}_d(\mathbb{P}_{\theta_1^{t_1}}, \mathbb{P}_{\theta_k^{t_k}}) \le \inf_{\mathcal{P}\in\mathbb{P}_{\theta_1^{t_1}}\otimes\mathbb{P}_{\theta_k^{t_k}}} \mathbb{E}_{(\mathbf{x},\mathbf{x}')\sim\mathcal{P}}[\mathcal{L}_c(\mathbf{x},\mathbf{x}')]\,, \quad (8)$$

where $\mathbb{P}_{\theta_1^{t_1}}\otimes\mathbb{P}_{\theta_k^{t_k}}$ represents the set of all probabilistic couplings for $\mathbb{P}_{\theta_1^{t_1}}$ and $\mathbb{P}_{\theta_k^{t_k}}$, and $\mathcal{P}$ is one of them. $(\mathbf{x},\mathbf{x}')$ is a pair of samples drawn from $\mathcal{P}$. $\mathcal{L}_c(\cdot,\cdot)$ is a cost function implemented using the squared Euclidean distance $\mathcal{L}_c(\mathbf{x},\mathbf{x}') = \sum_i^{d_x}(\mathbf{x}[i]-\mathbf{x}'[i])^2$, where $\mathbf{x}[i]$ is the i-th dimension of $\mathbf{x}$.

We provide the pseudocode for WEVAE in **Algorithm 1**, while the learning process has three main steps and is also illustrated in Fig. 1 :

- **Step 1 (Sample selection).** At a certain training time $\mathcal{T}_t$, we add a new data batch to the memory buffer, expressed as $\mathcal{M}_t = \mathcal{M}_t \bigcup \mathbf{X}_t$, $\mathbf{X}_t \sim \mathcal{S}$. We then perform the sample selection using Eq. (5) if the memory buffer $\mathcal{M}_t$ is overloaded, $|\mathcal{M}_t| > |\mathcal{M}|^{max}$.
- **Step 2 (Training process).** At the time $\mathcal{T}_t$, we assume that $\mathbf{G}$ already has k components and we only train the current component $\mathcal{G}_k$ on $\mathcal{M}_t$ using Eq. (1).
- **Step 3 (Check the model's expansion).** If the memory buffer is full $|\mathcal{M}_t| = |\mathcal{M}|^{max}$, we check the model's expansion using Eq. (4) to reduce the computational costs. If Eq. (4) is satisfied, we add a new component $\mathcal{G}_{k+1}$ (will be the current component) to $\mathbf{G}$. We also clear up the memory buffer in order to allow the newly added component to learn new, statistically non-overlapping information with that accumulated by other components. We then return to **Step 1** for the next training step $\mathcal{T}_{t+1}$.

4. Theoretical Analysis

In this section, we extend the results from [73] to describe how the proposed Wasserstein Expansion Mechanism can ensure a compact WEVAE model, containing VAE components representing diverse information.

4.1. Forgetting Analysis

Definition 1 *(**The distribution of the memory buffer.**) For a given memory buffer $\mathcal{M}_i$, updated at the time $\mathcal{T}_i$, we define the probabilistic representation of $\mathcal{M}_i$ as $\mathbb{P}_{\mathcal{M}_i}$.*

Definition 2 *(**Mixture model and component.**) For a mixture model $\mathbf{G} = \{\mathcal{G}_1, \cdots, \mathcal{G}_k\}$ with k components, we define the generative replay process of the i-th VAE component as the sampling procedure $\mathbf{x} \sim p_{\theta_i}(\mathbf{x} \,|\, \mathbf{z})$, $\mathbf{z} \sim \mathbb{P}_{\mathbf{z}}$. Let $\mathbb{P}_{\widetilde{\mathbf{x}}_i}$ represent the distribution of a finite number of generative replay samples drawn from $\mathcal{G}_i$. Let $\mathcal{M}_{b_i}$ be a previous memory buffer for the previously trained $\mathcal{G}_i$ component where b_i is the index of the training time $\mathcal{T}_{b_i}$.*

The theoretical analysis in [73] assumes that the target distribution (training samples) is static, which is unrealistic in a practical TFCL training environment. In this section, we derive a new bound for the proposed WEVAE in which the source and target distributions are changing continuously over time.

Algorithm 1 The learning process for WEVAE

```
(Input:The data stream);
for $\mathcal{T}_t < \mathcal{T}_n$ do
    Sample selection in the memory buffer
    $\mathbf{X}_t \sim \mathcal{S}$
    $\mathcal{M}_t = \mathcal{M}_t \cup \mathbf{X}_t$
    if $|\mathcal{M}_t| > |\mathcal{M}|^{max}$ then
        for $t < |\mathcal{M}_t|$ do
            $E(\mathbf{x}'_t) = \frac{1}{k-1} \sum_{j=1}^{k-1} \left\{ \mathrm{Rec}(\mathbf{x}'_t, f_{\theta_j}(f_{\omega_j}(\mathbf{x}'_t)) \right\}$
        end for
        $\mathcal{M}_t = \bigcup_{i=1}^{|\mathcal{M}_t|^{max}} \mathcal{M}'_t[i]$
    end if
    Training process
    if $k = 1$ and $\mathcal{T}_t = |\mathcal{M}|^{max}$ then
        Add the second component $\mathcal{G}_2$
    end if
    Train the current VAE component $\mathcal{G}_k$ on $\mathcal{M}_t$ using $\mathcal{L}_{ELBO}$
    Check the expansion
    if $|\mathcal{M}_i| \geq |\mathcal{M}|^{max}$ then
        if $\frac{1}{t} \sum_{j=1}^{t} \{A_t[j]\} > \lambda$ then
            Add a new Component $\mathcal{G}_{k+1}$
        end if
    end if
end for
Testing phase
for $i < n'$ do
    $\mathbf{x} \sim \mathcal{D}^T$
    $s^\star = \arg\max_{s=1,\cdots,k} \{\mathcal{L}_{ELBO}(\mathbf{x}; \mathcal{G}_s)\}$
    Choose $\mathcal{G}_{s^\star}$ for the evaluation.
end for
```

Theorem 1 *Let $\mathbb{P}_{\widehat{\mathbf{x}}_i}$ denote a probabilistic measure of all previously seen data batches $\{\mathbf{X}_1, \cdots, \mathbf{X}_{i-1}\}$ at time $\mathcal{T}_i$. We assume that $\mathbb{P}_{\widehat{\mathbf{x}}_i}$ involves a_i underlying data distributions $\{\mathbb{P}_{\widehat{\mathbf{x}}_i^1}, \cdots, \mathbb{P}_{\widehat{\mathbf{x}}_i^{a_i}}\}$ at $\mathcal{T}_i$ and each $\mathbb{P}_{\widehat{\mathbf{x}}_i^j}$ is usually a distribution of data batches of a unique category. Let c_j represent the number of data batches of $\mathbb{P}_{\widehat{\mathbf{x}}_i^j}$. We note that a_i would be increased as the training time ($\mathcal{T}_i$) increases. Let $\mathbf{X}_i^j(t)$ be the t-th data batch of $\mathbb{P}_{\widehat{\mathbf{x}}_i^j}$ and $\mathbb{P}_{\mathbf{x}_i^j(t)}$ be the distribution of $\mathbf{X}_i^j(t)$. We derive a bound for $\mathbf{G}$ at $\mathcal{T}_i$:*

$$\begin{aligned} \sum\nolimits_{j=1}^{a_i} \Big\{ \sum\nolimits_{t=1}^{c_j} \Big\{ \widetilde{\mathrm{F}}_s(\mathbf{G}, \mathbb{P}_{\mathbf{x}_i^j(t)}) \Big\} \Big\} \leq \\ \sum\nolimits_{j=1}^{a_i} \Big\{ \sum\nolimits_{t=1}^{c_j} \Big\{ \mathrm{F}_s(\mathbf{G}, \mathbb{P}_{\mathbf{x}_i^j(t)}) \Big\} \Big\}, \end{aligned} \tag{9}$$

where $\mathrm{F}_s(\mathbf{G}, \mathbb{P}_{\mathbf{x}_i^j(t)})$ is a function that returns the maximum bound, using the mixture model $\mathbf{G}$, defined as :

$$\begin{aligned} \mathrm{F}_s(\mathbf{G}, \mathbb{P}_{\mathbf{x}_i^j(t)}) &= \max_{\mathcal{G}_c \in \mathbf{G}} \Big\{ \mathbb{E}_{\mathbb{P}_{\mathcal{M}_{b_s}}} [\mathcal{L}_{ELBO}(\mathbf{x}; \mathcal{G}_c^{b_c})))] \\ &+ 2\mathrm{W}^\star_{\mathcal{L}}(\mathbb{P}_{\mathcal{M}_{b_c}}, \mathbb{P}_{\widetilde{\mathbf{x}}_c^{b_c}}) - \mathrm{W}^\star_{\mathcal{L}}(\mathbb{P}_{\mathbf{x}_i^j(t)}, \mathbb{P}_{\mathcal{M}_{b_c}}) \\ &+ \widetilde{\mathrm{F}}(\mathbb{P}_{\widetilde{\mathbf{x}}_c^{b_c}}, \mathbb{P}_{\mathcal{M}_{b_c}}) \Big\}, \end{aligned} \tag{10}$$

and $\widetilde{\mathrm{F}}_s(\mathbf{G}, \mathbb{P}_{\mathbf{x}_i^j(t)})$ is defined as :

$$\widetilde{\mathrm{F}}_s(\mathbf{G}, \mathbb{P}_{\mathbf{x}_i^j(t)}) = \max_{\mathcal{G}_c^{b_c} \in \mathbf{G}} \Big\{ \mathbb{E}_{\mathbb{P}_{\mathbf{x}_i^j(t)}} [\mathcal{L}_{ELBO}(\mathbf{x}; \mathcal{G}_c^{b_c})] \Big\}, \tag{11}$$

	Split MNIST			Split Fashion			Split MNIST-Fashion			Cross-domain		
Methods	Log ↑	Memory	N	Log ↑	Memory	N	Log ↑	Memory	N	Log ↑	Memory	N
VAE-reservoir [64]	-144.17	3.0K	1	-276.60	3.0K	1	-240.02	3.0K	1	-239.42	3.0K	1
VAE-ELBO-MIR [5]	-143.27	3.0K	1	-274.72	3.0K	1	-238.68	3.0K	1	-237.93	3.0K	1
VAE-ELBO-Random	-150.79	3.0K	1	-280.54	3.0K	1	-247.46	3.0K	1	-239.71	3.0K	1
LIMix [71]	-146.23	2.0K	30	-262.52	2.0K	30	-238.63	2.0K	30	-226.63	2.0K	30
CNDPM [38]	-120.71	2.0K	30	-257.56	2.0K	30	-236.79	2.0K	30	-218.15	2.0K	30
VAE-ELBO-OCM [73]	-132.07	1.6K	1	-250.74	1.6K	1	-215.62	2.0K	1	-201.31	2.0K	1
VAE-IWVAE50-OCM [73]	-127.11	1.6K	1	-247.90	1.6K	1	-224.34	2.0K	1	-204.35	2.0K	1
Dynamic-ELBO-OCM [73]	-115.89	1.6K	5	-237.69	1.8K	10	-187.49	1.9K	10	-177.29	2.0K	11
WEVAE	**-89.66**	1.6K	3	**-225.98**	1.5K	10	**-172.47**	1.9K	8	**-161.26**	2.0K	9
WEVAE-NoS	-99.29	1.6K	5	-230.52	1.5K	10	-179.23	1.9K	10	-168.67	2.0K	11

Table 1. The log-likelihood estimation on testing data by using the Importance Weighted Variational Autoencoder (IWVAE) bound with 1000 importance samples. The results for the comparison baselines are taken from [73]. N represents the number of components.

where $\mathcal{G}_c^{b_c}$ is the c-th component updated at $\mathcal{T}_{b_c}$ and $\mathbb{P}_{\widetilde{\mathbf{x}}_c^{b_c}}$ is the distribution of generative replay samples drawn from $\mathcal{G}_c^{b_c}$. $\mathrm{W}^\star_{\mathcal{L}}(\cdot)$ is defined as :

$$\begin{aligned}&\mathrm{W}^\star_{\mathcal{L}}(\mathbb{P}_{\mathbf{x}_i^j(t)}, \mathbb{P}_{\mathcal{M}_{b_c}}) := \\ &\inf_{\mathcal{P}' \in \mathbb{P}_{\mathbf{x}_i^j(t)} \otimes \mathbb{P}_{\mathcal{M}_{b_c}}} \mathbb{E}_{(\mathbf{x},\mathbf{x}') \sim \mathcal{P}'}[\mathcal{L}_c(\mathbf{x}, \mathbf{x}')]\end{aligned} \tag{12}$$

where $\mathbb{P}_{\mathbf{x}_i^j(t)} \otimes \mathbb{P}_{\mathcal{M}_{b_c}}$ is the set of all probabilistic couplings for $\mathbb{P}_{\mathbf{x}_i^j(t)}$ and $\mathbb{P}_{\mathcal{M}_{b_c}}$ and $\widetilde{\mathrm{F}}(\mathbb{P}_{\widetilde{\mathbf{x}}_c^{b_c}}, \mathbb{P}_{\mathcal{M}_{b_c}})$ is defined as :

$$\begin{aligned}\widetilde{\mathrm{F}}(\mathbb{P}_{\widetilde{\mathbf{x}}_c^{b_c}}, \mathbb{P}_{\mathcal{M}_i}) &= \mathbb{E}_{\mathbb{P}_{\mathcal{M}_i}}[D_{KL}(q_{\omega_c^{b_c}}(\mathbf{z} \,|\, \mathbf{x}) \,||\, p(\mathbf{z}))] \\ &+ \Big| \mathbb{E}_{\mathbb{P}_{\mathcal{M}_i}} \mathbb{E}_{q_{\omega_j^i}(\mathbf{z} \,|\, \mathbf{x})}[-\mathcal{L}_c(\mathbf{x}, f_{\theta_c^{b_c}}(f_{\omega_c^{b_c}}(\mathbf{x}))] \\ &- \mathrm{W}^\star_{\mathcal{L}}(\mathbb{P}_{\mathcal{M}_i}, \mathbb{P}_{\widetilde{\mathbf{x}}_c^{b_c}}) \Big| .\end{aligned} \tag{13}$$

where $\{\theta_c^{b_c}, \omega_c^{b_c}\}$ are the parameters of $\mathcal{G}_c^{b_c}$. The proof is provided in **Appendix-B** from SM.

Remark. We have several observations from Theorem 1 : (1) As the training time $\mathcal{T}_i$ increases, $\mathbb{P}_{\widehat{\mathbf{x}}_i}$ involves more underlying data distributions (a_i is increased), which represents a challenge for the model; (2) The proposed WEVAE can achieve better performance by designing a specific procedure of adding new components, where each component models a unique underlying data distribution; (3) Compared to the static/single model, the proposed WEVAE achieves better generalization performance (See details in **Appendix-C** from SM);

4.2. Analysis of the Trade-off for A New Component

In this section, we theoretically study the trade-off between the model size and its generalization performance, and provide theoretical guarantees for the proposed model, which are not available in [73].

Theorem 2 *For a given mixture model $\mathbf{G}$ with k components, we can view $\mathbf{G}$ as a single model trained on all memories $\{\mathcal{M}_{b_1}, \cdots, \mathcal{M}_{b_k}\}$ by using the component selection (Eq. (10)). Let $\mathbb{P}_{\widetilde{\mathbf{x}}^i}$ be the distribution of samples uniformly drawn from each component $\{\mathcal{G}_j, j = 1, \cdots, k\}$ at $\mathcal{T}_i$. Let $\mathbb{P}_{\mathcal{M}_{b_{1:b_k}}}$ be the distribution of all memory buffers $\{\mathcal{M}_{b_1}, \cdots, \mathcal{M}_{b_k}\}$. We can derive a bound for $\mathbf{G}$ at $\mathcal{T}_i$ as :*

$$\begin{aligned}&\mathbb{E}_{\mathbb{P}_{\widehat{\mathbf{x}}_i}}[\mathcal{L}_{ELBO}(\mathbf{x}; \mathbf{G})] \leq \mathbb{E}_{\mathbb{P}_{\mathcal{M}_{b_{1:b_k}}}}[\mathcal{L}_{ELBO}(\mathbf{x}; \mathbf{G})] \\ &+ 2\mathrm{W}^\star_{\mathcal{L}}(\mathbb{P}_{\mathcal{M}_{b_{1:b_k}}}, \mathbb{P}_{\widetilde{\mathbf{x}}^i}) - \mathrm{W}^\star_{\mathcal{L}}(\mathbb{P}_{\widehat{\mathbf{x}}_i}, \mathbb{P}_{\mathcal{M}_{b_{1:b_k}}}) \\ &+ \widetilde{\mathrm{F}}(\mathbb{P}_{\widetilde{\mathbf{x}}^i} \mathbb{P}_{\mathcal{M}_{b_{1:b_k}}}) .\end{aligned} \tag{14}$$

Remark. Theorem 2 has several observations : (1) $\mathbb{P}_{\mathcal{M}_{b_{1:b_k}}}$ can not be estimated since we can no longer access any of the previous memory buffers. We approximate $\mathbb{P}_{\mathcal{M}_{b_{1:b_k}}}$ by using $\mathbb{P}_{\widetilde{\mathbf{x}}^i}$ which is a distribution of samples uniformly drawn from each component and we have $\mathrm{W}^\star_{\mathcal{L}}(\mathbb{P}_{\widehat{\mathbf{x}}_i}, \mathbb{P}_{\mathcal{M}_{b_{1:b_k}}}) \approx \mathrm{W}^\star_{\mathcal{L}}(\mathbb{P}_{\widehat{\mathbf{x}}_i}, \mathbb{P}_{\widetilde{\mathbf{x}}^i})$. Training more components would allow $\mathbb{P}_{\widetilde{\mathbf{x}}^i}$ to capture a richer knowledge and thus decrease the term $\mathrm{W}^\star_{\mathcal{L}}(\mathbb{P}_{\widehat{\mathbf{x}}_i}, \mathbb{P}_{\widetilde{\mathbf{x}}^i})$, leading to a better performance; (2) Ensuring the knowledge diversity among components has two advantages : a) it can capture more underlying data distributions and improve the performance; b) it can reduce the number of necessary components without sacrificing much performance; (3) The theoretical foundations of the expansion mechanism from Eq. (4) are grounded in Theorem 2 and Eq. (14), which ensures the knowledge diversity among the WEVAE components. Such theoretical guarantees have not been discussed in other DEM-based methods [38, 48, 73]; (4) The threshold λ in Eq. (4) defines the trade-off between the model size its performance. A large λ can increase the information diversity of $\mathbb{P}_{\widetilde{\mathbf{x}}^i}$ but would lose some underlying data distributions of $\mathbb{P}_{\widehat{\mathbf{x}}_i}$. Meanwhile, an appropriate λ ensures the knowledge diversity of $\mathbb{P}_{\widetilde{\mathbf{x}}^i}$ while capturing all distributions of $\mathbb{P}_{\widehat{\mathbf{x}}_i}$ (See **Appendix-D** from SM).

Methods	*IS* ↑	*FID* ↓	Memory	N
VAE-ELBO-Random	3.84	116.26	1.0K	1
CNDPM [38]	4.12	95.23	1.0K	30
LIMix [71]	3.02	156.46	1.0K	30
VAE-ELBO-OCM [73]	4.13	98.76	1.0K	1
Dynamic-ELBO-OCM [73]	4.16	92.99	1.0K	3
WEVAE	**4.26**	**89.12**	1.0K	3

Table 2. IS and FID scores under Split CIFAR10. Results of other baselines are taken from [73].

Methods	**Split MNIST**	**Split CIFAR10**	**Split CIFAR100**
finetune*	19.75 ± 0.05	18.55 ± 0.34	3.53 ± 0.04
GEM*	93.25 ± 0.36	24.13 ± 2.46	11.12 ± 2.48
iCARL*	83.95 ± 0.21	37.32 ± 2.66	10.80 ± 0.37
reservoir*	92.16 ± 0.75	42.48 ± 3.04	19.57 ± 1.79
MIR*	93.20 ± 0.36	42.80 ± 2.22	20.00 ± 0.57
GSS*	92.47 ± 0.92	38.45 ± 1.41	13.10 ± 0.94
CoPE-CE*	91.77 ± 0.87	39.73 ± 2.26	18.33 ± 1.52
CoPE*	93.94 ± 0.20	48.92 ± 1.32	21.62 ± 0.69
ER + GMED†	82.67 ± 1.90	34.84 ± 2.20	20.93 ± 1.60
ER$_a$ + GMED†	82.21 ± 2.90	47.47 ± 3.20	19.60 ± 1.50
WGF-SVGD	-	47.90 ± 2.50	19.90 ± 2.30
CURL*	92.59 ± 0.66	-	-
CNDPM*	93.23 ± 0.09	45.21 ± 0.18	20.10 ± 0.12
Dynamic-OCM	94.02 ± 0.23	49.16 ± 1.52	21.79 ± 0.68
WEVAE	**96.62** ± 0.27	**55.23** ± 1.26	**25.07** ± 0.59
WEVAE-NoS	95.12 ± 0.29	51.38 ± 1.16	23.18 ± 0.79

Table 3. Classification accuracy of five independent runs for various models on three datasets. * and † denote the results cited from [16] and [29], respectively.

Methods	M-S	N	M-C	N	Split MI	N
CNDPM	49.23	21	53.97	19	26.35	8
Dynamic-OCM	50.18	20	54.49	21	26.78	8
WEVAE	**53.01**	14	**59.07**	13	**29.28**	6
WEVAE-NoS	51.72	17	56.98	16	28.69	7

Table 4. Classification accuracy on challenging settings.

5. Experiments

In the following, we evaluate the proposed Wasserstein Expansible Variational Autoencoder (WEVAE) and we also test WEVAE-NoS, where we have the WEVAE model but without considering the sample selection mechanism, described in Section 3.2, on a series of unsupervised and supervised learning experiments. The hyperparameter setting is provided in **Appendix-E** from SM.

Baselines. We use the experimental setting from [73] and compare WEVAE with several continual learning models for the density estimation task, including : VAE-ELBO-OCM [73], VAE-IWVAE50-OCM [73], Dynamic-ELBO-OCM [73], CNDPM [38], VAE-ELBO-Random, LIMix [71], VAE-ELBO-MIR [5]. and VAE-reservoir [64]. For the classification task, we adopt the baselines from the recent TFCL benchmark [16] (See details in the **Appendices-E2, E3** from SM).

Performance criterion : Since this paper focuses on TFCL where the task boundaries are not available, we adopt the accuracy as performance criterion for supervised learning [16]. For the results in unsupervised learning, we estimate the sample log-likelihood (Log) by using the IWVAE bound [11], considering 1000 importance samples, [73].

Datasets : For the density estimation task, we consider the Split MNIST/Fashion [73] which divides MNIST/Fashion [37] into ten parts according to the category information. We also consider the Split MNIST-Fashion, which combines Split MNIST and Split Fashion into a single data stream. In addition, we consider a more challenging data stream, "Cross-Domain", which combines the Split MNIST-Fashion and OMNIGLOT [36]. For supervised learning, we consider Split MNIST, Split CIFAR10 and Split CIFAR100 (See details in **Appendix-E2** from SM).

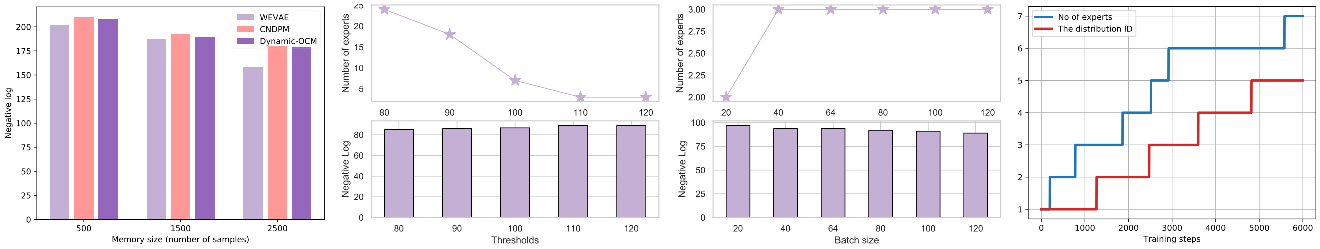
5.1. Density Estimation and Image Reconstruction

The results for density estimation task are provided in Table 1. We can observe that the dynamic expansion models outperform the static model in all datasets while requiring fewer memorized samples. In addition, the proposed WEVAE-NoS achieves a Log-likelihood of -99.29, -230.52, 179.23 and -168.67 on Split MNIST, Split Fashion, Split MNIST-Fashion and Cross-domain, respectively, which show that the proposed WEVAE-NoS outperforms other dynamic expansion models on all datasets. The model complexity is discussed in **Appendix-F10** from SM. The proposed WEVAE achieves better performance using fewer components than WEVAE-NoS. These results demonstrate that the proposed WEVAE can achieve the best performance using a compact model structure, which is consistent with the theoretical analysis from Theorem 2.

Based on the setting from [73], we evaluate the performance of WEVAE on the image reconstruction task on Split CIFAR10, and the results are provided in Table 2. The proposed WEVAE outperforms other baselines in terms of Inception Score (IS) [53] and Fréchet Inception Distance (FID) [25] criteria.

5.2. Classification Task

We follow the TFCL classification setting from [16], where a model only sees a batch of ten samples at a certain training time. The maximum memory size is set as $|\mathcal{M}|^{max} = \{1000, 2000, 5000\}$ for Split MNIST, Split CIFAR10, and Split CIFAR100, respectively. We implement



(a) Memory size. (b) Varying the threshold λ. (c) Changing batch size. (d) Model expansion.

Figure 2. Ablation study results. (a) The performance of various models on Cross-Domain data when changing the memory buffer size, $|\mathcal{M}|^{max}$. (b) The performance (negative log) and the number of components when changing the threshold λ in Eq. (4). (c) The performance and the number of components of WEVAE for learning Split MNIST when changing the batch size. (d) The number of components of WEVAE and the distribution change, measured as the change of data category (information not used for training), over time.

the classifier of each expert by using a simple, fully connected network and a ResNet18 [23] for Split MNIST and Split CIFAR10/100, respectively. According to the classification results from Table 3, the proposed WEVAE-NoS outperforms other static and dynamic expansion models on all datasets. In addition, WEVAE improves its performance by using the proposed sample selection approach described in Section 3.2. Further results can be found in **Appendix-E2** from SM.

We also investigate the effectiveness of WEVAE using more challenging continual learning settings. We build a data stream consisting of Split MNIST and Split SVHN, namely M-S. Similarly, we also create the data stream M-C, representing Split MNIST and Split CIFAR10. Meanwhile, Split MiniImageNet (Split IM) [63] divides MiniImageNet into 20 tasks, where each task collects the images of five classes [5]. The maximum memory is $|\mathcal{M}|^{max} = \{1,000, 1,000, 10,000\}$ for M-S, M-C and Split IM. The classification results from Table 4, show that the proposed WEVAE still outperforms other dynamic expansion models while using fewer components, indicated by N.

5.3. Ablation Study

In the ablation study we investigate the effectiveness of each module of the proposed WEVAE. Additional results are provided in **Appendix-F** from SM.

Size of the memory buffer : We evaluate the performance when changing the memory buffer size $|\mathcal{M}|^{Max}$ on Cross-Domain data, and the results are provided in Fig. 2-a. As the memory buffer increases its capacity, all models improve their performance. The proposed WEVAE outperforms other models on all memory configurations, even when the memory buffer stores only 500 samples.

Changing the threshold λ : We investigate the performance and the number of components of WEVAE on Split MNIST when changing the threshold λ from Eq. (4) and the results are shown in Fig. 2-b. Decreasing λ would increase the number of components but does not lead to a significant improvement in the negative log performance. Those results show that the proposed WEVAE can achieve good performance with only three components, proving the ability of each component (expert) to capture diverse knowledge.

Changing the batch size : We also investigate the performance and the number of components of WEVAE when changing the batch size, and the results are shown in Fig. 2-c. The proposed WEVAE does not suffer from a degenerated performance and requires a similar number of components when changing the batch size.

Model expansion process : In Fig. 2-d we provide the number of components for WEVAE and the change of the data distribution, measured as the change in the data class, on Split MNIST, considering the classification task. The proposed WEVAE frequently adds new components during the initial learning stages and less later on, during its further learning stages. The reason is that when WEVAE has accumulated the necessary knowledge, it does not need more components to learn the information from the later learning processes which are related to what it had already learnt previously. Each component learns a single or a few underlying data distributions. In the latter case the data distributions are similar in their statistical data representations.

6. Conclusion

In this paper, we propose the Wasserstein Expansible Variational Autoencoder (WEVAE), a new approach for Task Free Continual Learning (TFCL), which adaptively expands a VAE mixture model, by adding new components when the given new tasks are characterized by different underlying probabilistic representations than those learnt in the past. A memory buffer is considered for temporarily storing data samples from new databases. Data from the memory buffer are selected and used for training, according to a novelty detection mechanism in order to further promote the knowledge diversity among components. We theoretically and empirically demonstrate that the proposed WEVAE performs well while requiring a compact model structure.

References

[1] E. Abbasnejad, M. Dick, and A. van der Hengel. Infinite variational autoencoder for semi-supervised learning. In *Proc. of IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 5888–5897, 2017. 1

[2] A. Achille, T. Eccles, L. Matthey, C. Burgess, N. Watters, A. Lerchner, and I. Higgins. Life-long disentangled representation learning with cross-domain latent homologies. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 9873–9883, 2018. 2

[3] Hongjoon Ahn, Sungmin Cha, Donggyu Lee, and Taesup Moon. Uncertainty-based continual learning with adaptive regularization. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 4394–4404, 2019. 2

[4] Rahaf Aljundi, Eugene Belilovsky, Tinne Tuytelaars, Laurent Charlin, Massimo Caccia, Min Lin, and Lucas Page-Caccia. Online continual learning with maximal interfered retrieval. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 11872–11883, 2019. 2

[5] Rahaf Aljundi, Eugene Belilovsky, Tinne Tuytelaars, Laurent Charlin, Massimo Caccia, Min Lin, and Lucas Page-Caccia. Online continual learning with maximal interfered retrieval. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 11872–11883, 2019. 6, 7, 8

[6] Rahaf Aljundi, Klaas Kelchtermans, and Tinne Tuytelaars. Task-free continual learning. In *Proc. of IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 11254–11263, 2019. 1, 2, 3

[7] R. Aljundi, M. Lin, B. Goujaud, and Y. Bengio. Gradient based sample selection for online continual learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 11817–11826, 2019. 2

[8] M. Arjovsky, S. Chintala, and L. Bottou. Wasserstein generative adversarial networks. In *Proc. Int. Conf. on Machine Learning (ICML)*, pages 214–223, 2017. 4

[9] Jihwan Bang, Heesu Kim, YoungJoon Yoo, Jung-Woo Ha, and Jonghyun Choi. Rainbow memory: Continual learning with a memory of diverse samples. In *Proc. of IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8218–8227, 2021. 1, 3

[10] Jihwan Bang, Hyunseo Koh, Seulki Park, Hwanjun Song, Jung-Woo Ha, and Jonghyun Choi. Online continual learning on a contaminated data stream with blurry task boundaries. In *Proc. of IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9275–9284, 2022. 2

[11] Yuri Burda, Roger Grosse, and Ruslan Salakhutdinov. Importance weighted autoencoders. *arXiv preprint arXiv:1509.00519*, 2015. 7

[12] Hyuntak Cha, Jaeho Lee, and Jinwoo Shin. Co2l: Contrastive continual learning. In *Proc. of IEEE/CVF Int. Conf. on Computer Vision (ICCV)*, pages 9516–9525, 2021. 2

[13] Arslan Chaudhry, Marc'Aurelio Ranzato, Marcus Rohrbach, and Mohamed Elhoseiny. Efficient lifelong learning with A-GEM. In *International Conference on Learning Representations (ICLR), arXiv preprint arXiv:1812.00420*, 2019. 2

[14] A. Chaudhry, M. Rohrbach, M. Elhoseiny, T. Ajanthan, P. Dokania, P. H. S. Torr, and M.'A. Ranzato. On tiny episodic memories in continual learning. *arXiv preprint arXiv:1902.10486*, 2019. 1, 2, 3

[15] C. Cortes, X. Gonzalvo, V. Kuznetsov, M. Mohri, and S. Yang. AdaNet: Adaptive structural learning of artificial neural networks. In *Proc. of Int. Conf. on Machine Learning (ICML), vol. PMLR 70*, pages 874–883, 2017. 2

[16] Matthias De Lange and Tinne Tuytelaars. Continual prototype evolution: Learning online from non-stationary data streams. In *Proc. of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 8250–8259, 2021. 2, 7

[17] Danruo Deng, Guangyong Chen, Jianye Hao, Qiong Wang, and Pheng-Ann Heng. Flattening sharpness for dynamic gradient projection memory benefits continual learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 34, pages 18710–18721, 2021. 2

[18] Mohammad Derakhshani, Xiantong Zhen, Ling Shao, and Cees Snoek. Kernel continual learning. In *Proc. International Conference on Machine Learning (ICML)*, pages 2621–2631. PMLR 139, 2021. 2

[19] Evgenii Egorov, Anna Kuzina, and Evgeny Burnaev. Boovae: Boosting approach for continual learning of vae. *Advances in Neural Information Processing Systems (NeurIPS)*, 34:17889–17901, 2021. 2

[20] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial nets. In *Proc. Advances in Neural Inf. Proc. Systems (NIPS)*, pages 2672–2680, 2014. 2

[21] Yanan Gu, Xu Yang, Kun Wei, and Cheng Deng. Not just selection, but exploration: Online class-incremental continual learning via dual view consistency. In *Proc. of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7442–7451, June 2022. 2

[22] Yiduo Guo, Bing Liu, and Dongyan Zhao. Online continual learning through mutual information maximization. In *Proc. International Conference on Machine Learning (ICML)*, pages 8109–8126. PMLR 162, 2022. 2

[23] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proc. of IEEE Conf. on Computer Vision and Pattern Recog. (CVPR)*, pages 770–778, 2016. 8

[24] Christian Henning, Maria Cervera, Francesco D'Angelo, Johannes Von Oswald, Regina Traber, Benjamin Ehret, Seijin Kobayashi, Benjamin F Grewe, and João Sacramento. Posterior meta-replay for continual learning. *Advances in Neural Information Processing Systems*, 34:14135–14149, 2021. 2

[25] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local Nash equilibrium. In *Proc. Advances in Neural Information Processing Systems (NIPS)*, pages 6626–6637, 2017. 7

[26] G. Hinton, O. Vinyals, and J. Dean. Distilling the knowledge in a neural network. In *Proc. NIPS Deep Learning Workshop, arXiv preprint arXiv:1503.02531*, 2014. 2

[27] Ching-Yi Hung, Cheng-Hao Tu, Cheng-En Wu, Chien-Hung Chen, Yi-Ming Chan, and Chu-Song Chen. Compacting, picking and growing for unforgetting continual learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 13647–13657, 2019. 2

[28] Julio Hurtado, Alain Raymond, and Alvaro Soto. Optimizing reusable knowledge for continual learning via metalearning. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 34, pages 14150–14162, 2021. 2

[29] Xisen Jin, Arka Sadhu, Junyi Du, and Xiang Ren. Gradient-based editing of memory examples for online task-free continual learning. In *Advances in Neural Inf. Proc. Systems (NeurIPS), arXiv preprint arXiv:2006.15294*, 2021. 2, 7

[30] H. Jung, J. Ju, M. Jung, and J. Kim. Less-forgetting learning in deep neural networks. In *Proc. AAAI Conf. on Artificial Intelligence*, volume 32, pages 3358–3365, 2018. 2

[31] Haeyong Kang, Rusty Mina, Sultan Madjid, Jaehong Yoon, Mark Hasegawa-Johnson, Sung Ju Hwang, and Chang Yoo. Forget-free continual learning with winning subnetworks. In *Proc. Int. Conf. on Machine Learning (ICML)*, pages 10734–10750. PMLR 162, 2022. 2

[32] Ronald Kemker, Marc McClure, Angelina Abitino, Tyler Hayes, and Christopher Kanan. Measuring catastrophic forgetting in neural networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, 2018. 2

[33] D. P. Kingma and M. Welling. Auto-encoding variational Bayes. *arXiv preprint arXiv:1312.6114*, 2013. 1, 3

[34] J. Kirkpatrick, R. Pascanu, N. Rabinowitz, J. Veness, G. Desjardins, A. A. Rusu, K. Milan, J. Quan, T. Ramalho, A. Grabska-Barwinska, D. Hassabis, C. Clopath, D. Kumaran, and R. Hadsell. Overcoming catastrophic forgetting in neural networks. *Proc. of the National Academy of Sciences (PNAS)*, 114(13):3521–3526, 2017. 1, 2

[35] Richard Kurle, Botond Cseke, Alexej Klushyn, Patrick van der Smagt, and Stephan Günnemann. Continual learning with Bayesian neural networks for non-stationary data. In *Int. Conf. on Learning Representations (ICLR)*, 2020. 2

[36] Brenden Lake, Ruslan Salakhutdinov, and Joshua Tenenbaum. Human-level concept learning through probabilistic program induction. *Science*, 350(6266):1332–1338, 2015. 7

[37] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proc. of the IEEE*, 86(11):2278–2324, 1998. 7

[38] Soochan Lee, Junsoo Ha, Dongsu Zhang, and Gunhee Kim. A neural Dirichlet process mixture model for task-free continual learning. In *Int. Conf. on Learning Representations (ICLR), arXiv preprint arXiv:2001.00689*, 2020. 1, 2, 6, 7

[39] Z. Li and D. Hoiem. Learning without forgetting. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 40(12):2935–2947, 2017. 1, 2

[40] Ming-Yu Liu, Thomas Breuel, and Jan Kautz. Unsupervised image-to-image translation networks. In *Advances in Neural Information Proc. Systems (NIPS)*, pages 700–708, 2017. 1

[41] Xiaofeng Liu, Fangxu Xing, Jerry L Prince, Aaron Carass, Maureen Stone, Georges El Fakhri, and Jonghye Woo. Dual-cycle constrained bijective vae-gan for tagged-to-cine magnetic resonance image synthesis. In *Proc. IEEE Int. Symp. on Biomedical Imaging (ISBI)*, pages 1448–1452, 2021. 1

[42] David Lopez-Paz and Marc'Aurelio Ranzato. Gradient episodic memory for continual learning. In *Advances in Neural Information Processing Systems (NIPS)*, pages 6467–6476, 2017. 2

[43] James Martens and Roger B. Grosse. Optimizing neural networks with Kronecker-factored approximate curvature. In *Proc. of the International Conference on Machine Learning (ICML)*, volume PMLR 37, pages 2408–2417, 2015. 2

[44] Cuong Nguyen, Yingzhen Li, Thang D Bui, and Richard E Turner. Variational continual learning. In *Internatioanl Conference on Learning Representations (ICLR), arXiv preprint arXiv:1710.10628*, 2018. 2

[45] G. I. Parisi, R. Kemker, J. L. Part, C. Kanan, and S. Wermter. Continual lifelong learning with neural networks: A review. *Neural Networks*, 113:54–71, 2019. 1

[46] R. Polikar, L. Upda, S. S. Upda, and Vasant Honavar. Learn++: An incremental learning algorithm for supervised neural networks. *IEEE Trans. on Systems Man and Cybernetics, Part C*, 31(4):497–508, 2001. 2

[47] J. Ramapuram, M. Gregorova, and A. Kalousis. Lifelong generative modeling. In *International Conference on Learning Representations (ICLR), arXiv preprint arXiv:1705.09847*, 2017. 2

[48] Dushyant Rao, Francesco Visin, Andrei A. Rusu, Yee Whye Teh, Razvan Pascanu, and Raia Hadsell. Continual unsupervised representation learning. In *Proc. Neural Inf. Proc. Systems (NeurIPS)*, pages 7645–7655, 2019. 1, 2, 6

[49] Sylvestre-Alvise Rebuffi, Alexander Kolesnikov, Georg Sperl, and Christoph H Lampert. iCaRL: Incremental classifier and representation learning. In *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 2001–2010, 2017. 2

[50] B. Ren, H. Wang, J. Li, and H. Gao. Life-long learning based on dynamic combination model. *Applied Soft Computing*, 56:398–404, 2017. 2

[51] Hippolyt Ritter, Aleksandar Botev, and David Barber. Online structured Laplace approximations for overcoming catastrophic forgetting. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 3742–3752, 2018. 2

[52] Andrei A Rusu, Neil C Rabinowitz, Guillaume Desjardins, Hubert Soyer, James Kirkpatrick, Koray Kavukcuoglu, Razvan Pascanu, and Raia Hadsell. Progressive neural networks. *arXiv preprint arXiv:1606.04671*, 2016. 1, 2

[53] T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Radford, and X. Chen. Improved techniques for training GANs. In *Advances in Neural Inf. Proc. Systems (NIPS)*, pages 2234–2242, 2016. 7

[54] Edgar Schönfeld, Sayna Ebrahimi, Samarth Sinha, Trevor Darrell, and Zeynep Akata. Generalized zero- and few-shot learning via aligned variational autoencoders. In *Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8247–8255, 2019. 1

[55] Yujun Shi, Li Yuan, Yunpeng Chen, and Jiashi Feng. Continual learning via bit-level information preserving. In *Proc. of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 16674–16683, 2021. 2

[56] H. Shin, J. K. Lee, J. Kim, and J. Kim. Continual learning with deep generative replay. In *Advances in Neural Inf. Proc. Systems (NIPS)*, pages 2990–2999, 2017. 2

[57] Kihyuk Sohn, Honglak Lee, and Xinchen Yan. Learning structured output representation using deep conditional gen-

erative models. In *Advances in neural information processing systems (NIPS)*, pages 3483–3491, 2015. 4

[58] A. Srivastava, L. Valkov, C. Russell, M. U. Gutmann, and C. Sutton. VEEGAN: Reducing mode collapse in gans using implicit variational learning. In *Proc. Advances in Neural Inf. Proc. Systems (NIPS)*, pages 3308–3318, 2017. 2

[59] Hiroshi Takahashi, Tomoharu Iwata, Yuki Yamanaka, Masanori Yamada, and Satoshi Yagi. Variational autoencoder with implicit optimal priors. In *Proc. of AAAI Conference on Artificial Intelligence*, pages 5066–5073, 2019. 1

[60] Rishabh Tiwari, Krishnateja Killamsetty, Rishabh Iyer, and Pradeep Shenoy. GCR: Gradient coreset based replay buffer selection for continual learning. In *Proc. of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 99–108, 2022. 2

[61] Ilya Tolstikhin, Olivier Bousquet, Sylvain Gelly, and Bernhard Schoelkopf. Wasserstein auto-encoders. In *International Conference on Learning Representations (ICLR), arXiv preprint arXiv:1711.01558*, 2018. 4

[62] Vinay Verma, Kevin Liang, Nikhil Mehta, Piyush Rai, and Lawrence Carin. Efficient feature transformations for discriminative and generative continual learning. In *Proc. of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 13865–13875, 2021. 2

[63] Oriol Vinyals, Charles Blundell, Timothy Lillicrap, Koray Kavukcuoglu, and Daan Wierstra. Matching networks for one shot learning. *Advances in neural information processing systems (NIPS)*, 29:3637–3645, 2016. 8

[64] Jeffrey Vitter. Random sampling with a reservoir. *ACM Trans. on Mathematical Software (TOMS)*, 11(1):37–57, 1985. 6, 7

[65] Liyuan Wang, Mingtian Zhang, Zhongfan Jia, Qian Li, Chenglong Bao, Kaisheng Ma, Jun Zhu, and Yi Zhong. AFEC: Active forgetting of negative transfer in continual learning. *Advances in Neural Information Processing Systems (NeurIPS)*, 34:22379–22391, 2021. 2

[66] Shipeng Wang, Xiaorong Li, Jian Sun, and Zongben Xu. Training networks in null space of feature covariance for continual learning. In *Proc. of IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 184–193, 2021. 2

[67] Yeming Wen, Dustin Tran, and Jimmy Ba. BatchEnsemble: an alternative approach to efficient ensemble and lifelong learning. In *Proc. Int. Conf. on Learning Representations (ICLR), arXiv preprint arXiv:2002.06715*, 2020. 2

[68] T. Xiao, J. Zhang, K. Yang, Y. Peng, and Z. Zhang. Error-driven incremental learning in deep convolutional neural network for large-scale image classification. In *Proc. of ACM Int. Conf. on Multimedia*, pages 177–186, 2014. 2

[69] Qingsen Yan, Dong Gong, Yuhang Liu, Anton van den Hengel, and Javen Qinfeng Shi. Learning Bayesian sparse networks with full experience replay for continual learning. In *Proc. of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 109–118, 2022. 2

[70] Fei Ye and Adrian G. Bors. Lifelong learning of interpretable image representations. In *Proc. Int. Conf. on Image Processing Theory, Tools and Applications (IPTA)*, pages 1–6, 2020. 2

[71] Fei Ye and Adrian G. Bors. Lifelong infinite mixture model based on knowledge-driven Dirichlet process. In *Proc. of the IEEE International Conference on Computer Vision (ICCV)*, pages 10695–10704, 2021. 1, 2, 6, 7

[72] Fei Ye and Adrian G. Bors. Lifelong twin generative adversarial networks. In *Proc. IEEE Int. Conf. on Image Processing (ICIP)*, pages 1289–1293, 2021. 2

[73] Fei Ye and Adrian G. Bors. Continual variational autoencoder learning via online cooperative memorization. In *Proc. European Conference on Computer Vision (ECCV), vol. LNCS 13683*, pages 531–549, 2022. 1, 2, 3, 5, 6, 7

[74] Fei Ye and Adrian G. Bors. Lifelong generative modelling using dynamic expansion graph model. In *Proc. of AAAI Conference on Artificial Intelligence*, pages 8857–8865, 2022. 2

[75] Fei Ye and Adrian G. Bors. Lifelong teacher-student network learning. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 44(10):6280–6296, 2022. 2

[76] Fei Ye and Adrian G Bors. Task-free continual learning via online discrepancy distance learning. *Advances in Neural Information Processing Systems*, 35:23675–23688, 2022. 2

[77] Fei Ye and Adrian G Bors. Continual variational autoencoder via continual generative knowledge distillation. In *Proc. of AAAI Conference on Artificial Intelligence*, pages 10918–10926, 2023. 2

[78] Fei Ye and Adrian G Bors. Dynamic self-supervised teacher-student network learning. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 45(5):5731–5748, 2023. 2

[79] Fei Ye and Adrian G Bors. Learning dynamic latent spaces for lifelong generative modelling. In *Proc. of the AAAI Conference on Artificial Intelligence*, volume 37, pages 10891–10899, 2023. 2

[80] Fei Ye and Adrian G Bors. Lifelong dual generative adversarial nets learning in tandem. *IEEE Transactions on Cybernetics*, pages 1–13, 2023. 2

[81] Fei Ye and Adrian G Bors. Lifelong generative adversarial autoencoder. *IEEE Transactions on Neural Networks and Learning Systems*, pages 1–15, 2023. 2

[82] Fei Ye and Adrian G. Bors. Lifelong mixture of variational autoencoders. *IEEE Transactions on Neural Networks and Learning Systems*, 34(1):461–474, 2023. 2

[83] Haiyan Yin, Peng Yang, and Ping Li. Mitigating forgetting in online continual learning with neuron calibration. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 10260–10272, 2021. 2

[84] M. Zhai, L. Chen, F. Tung, J He, M. Nawhal, and G. Mori. Lifelong GAN: Continual learning for conditional image generation. In *Proc. of the IEEE/CVF Int. Conference on Computer Vision (ICCV)*, pages 2759–2768, 2019. 1, 2

[85] Junbo Jake Zhao, Michaël Mathieu, and Yann LeCun. Energy-based generative adversarial network. In *International Conference on Learning Representations (ICLR), arXiv preprint arXiv:1609.03126*, 2017. 2

[86] Guanyu Zhou, Kihyuk Sohn, and Honglak Lee. Online incremental feature learning with denoising autoencoders. In *Proc. Inter. Conf. on Artifical Intelligence (AISTATS), vol. PMLR 22*, pages 1453–1461, 2012. 2