Appendix for Lifelong Variational Autoencoder Using Online Adversarial Expansion Strategy

December 1, 2022

Contents

A Additional information for the proposed OAES

In this section, we provide the detailed learning procedure of the proposed model in Fig. 1.

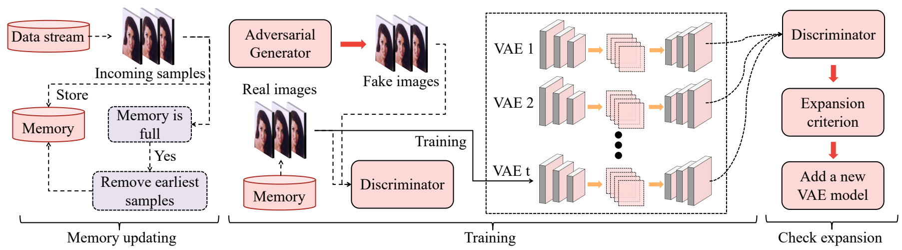


Figure 1: The learning procedure of the proposed Online Adversarial Expansion Strategy (OAES) model. During the memory updating stage, we continually add incoming samples to the memory buffer. When the memory buffer is full, we automatically remove the earliest added samples. In the second stage, if the proposed OAES does not have any VAE components, we build the first VAE component which is then trained until for $\mathcal{T}_{100}$ steps. Then we freeze the first component and treat it as the adversary in the subsequent learning. If we have already learnt t components $\mathbf{V} = \{\mathcal{V}_1^{c_1}, \cdots, \mathcal{V}_t^{c_t}\}$ at $\mathcal{T}_i$, we only train $\mathcal{V}_t^{c_t}$ on $\mathcal{M}_i$ at $\mathcal{T}_i$ by using Eq.(1) of the paper to avoid forgetting previously learnt knowledge. In addition, we train the generator G_{ε^i} and the discriminator D_{ψ^i} on $\mathcal{M}_i$ using adversarial loss (Eq.(14) of the paper). In the third stage, we check the expansion criterion using the evaluation stage of OAES (Eq.(15) of the paper). If Eq.(15) of the paper is satisfied, we add a new component $\mathcal{V}_{t+1}$ into $\mathbf{V}$ and clear up $\mathcal{M}_i$ in order to learn data samples which are statistically non-overlapping in the following training step.

In the following, we provide the pseudocode of the proposed OAES in Algorithm 1, which can be summarized into four stages :

Stage 1 . Memory updating : Let $|\mathcal{M}|^{max}$ be the maximum number of samples in the memory buffer. Since this paper does not focus on the sample selection for the memory buffer, we simply remove the earliest stored samples and add new incoming samples into $\mathcal{M}_i$ at the i-th training step ($\mathcal{T}_i$), if the maximum memory buffer size $|\mathcal{M}|^{max}$ is reached.

Stage 2 . Training a component : If $\mathbf{V}$ has a single component, then we train $\mathcal{V}_1$ on $\mathcal{M}_i$ until finishes the training step $\mathcal{T}_i = |\mathcal{M}|^{max}$ in order to preserve the initial information about the data stream. Then we build the second component $\mathcal{V}_2$ in the subsequent learning. We describe the following training as follows. Let us suppose that we have already learnt t components $\mathbf{V} = \{\mathcal{V}_1^{c_1}, \cdots, \mathcal{V}_t^{c_t}\}$ at $\mathcal{T}_i$, we only train $\mathcal{V}_t^{c_t}$ on $\mathcal{M}_i$ at $\mathcal{T}_i$ by using Eq.(1) of the paper to avoid forgetting previously learnt knowledge. In addition, we train the generator G_{ε^i} and the discriminator D_{ψ^i} on $\mathcal{M}_i$ using adversarial loss (Eq.(14) of the paper).

Stage 3 . Check the expansion : If $|\mathcal{M}_i|$ reaches $|\mathcal{M}|^{max}$, then we check the expansion criterion using the evaluation stage of OAES (Eq.(15) of the paper). If Eq.(15) of the paper is satisfied, we add a new component $\mathcal{V}_{t+1}$ into $\mathbf{V}$ and clear up the memory $\mathcal{M}_i$ in order to learn statistically non-overlapping samples in the following training step.

Stage 4 . Component selection at the testing phase : For a given sample $\mathbf{x}$, we choose a component with the highest sample log-likelihood by :

$$s^{\star} = \arg \max_{s=1,\cdots,|\mathbf{V}|} \{\mathcal{L}_{ELBO}(\mathbf{x}; \mathcal{V}_s)\}\,, \tag{1}$$

Algorithm 1 Algorithm for OAES

(Input:The data stream);
for $i < n$ **do**
  Memory updating
  $\mathcal{B}_i \sim W$
  $\mathcal{M}_i = \mathcal{M}_i \cup \mathcal{B}_i$
  if $|\mathcal{M}_i| > |\mathcal{M}|^{max}$ **then**
    $\mathcal{M}_i = \bigcup_{j=10}^{|\mathcal{M}|^{max}+10} \mathcal{M}_i$
  end if
  Training the component
  if $|\mathbf{V}| = 1$ and $\mathcal{T}_i = |\mathcal{M}|^{max}$ **then**
    Add the second component $\mathcal{V}_2$
  end if
  Train the current VAE component $\mathcal{V}_t$ on $\mathcal{M}_i$ using $\mathcal{L}_{ELBO}$
  Train the generator G_{ε^i} and the discriminator D_{ψ^i} on $\mathcal{M}_i$ using adversarial loss
  Check the expansion
  if $|\mathcal{M}_i| > |\mathcal{M}|^{max}$ **then**
    if $\min\left\{C_{\psi^i}(\mathbb{P}_{\theta_1^{c_1}}, \mathbb{P}_{\theta_t^i}), \cdots, C_{\psi^i}(\mathbb{P}_{\theta_{t-1}^{c_{t-1}}}, \mathbb{P}_{\theta_t^i})\right\} \geq \beta$ **then**
      Add a new Component $\mathcal{V}_{t+1}$
    end if
  end if
end for
for $i < n'$ **do**
  $\mathbf{x} \sim \mathbf{X}_{test}$
  $s^{\star} = \arg\max_{s=1,\cdots,|\mathbf{V}|}\{\mathcal{L}_{ELBO}(\mathbf{x}; \mathcal{V}_s)\}$
  Choose $\mathcal{V}_{s^{\star}}$ for the evaluation.
end for

B Theorem 1

Theorem 1 *Let $p_{\theta_i}(\mathbf{x})$ be a probability density function for a single model $\mathcal{V}^i$ updated at $\mathcal{T}_i$. Let $\mathbb{P}_i^W$ denote a distribution of all visited data batches $\{\mathcal{B}_1, \cdots, \mathcal{B}_i\}$ drawn* *from W at $\mathcal{T}_i$. Let $p_{\mathcal{M}_i}(\mathbf{x})$ and $p_{W^i}(\mathbf{x})$ denote the density function for $\mathbb{P}_{\mathcal{M}_i}$ and $\mathbb{P}_i^W$, respectively. We then derive an upper bound for a single VAE model trained on $\mathcal{M}_i$ at $\mathcal{T}_i$ as :*

$$\begin{aligned}\mathbb{E}_{\mathbb{P}_i^W}\big[\log p_{\theta^i}(\mathbf{x})\big] \geq\ & \mathbb{E}_{\mathbb{P}_{\mathcal{M}_i}}\big[\log p_{\theta^i}(\mathbf{x})\big] \\ & - D_{KL}\big(\mathbb{P}_i^W \mid\mid \mathbb{P}_{\mathcal{M}_i}\big) - \mathcal{F}_{\mathrm{DL}}\big(\mathbb{P}_{\mathcal{M}_i}, \mathbb{P}_i^W, \mathbb{P}_{\theta^i}\big) \\ & + \mathcal{F}_{\mathrm{dis}}\big(\mathbb{P}_i^W, \mathbb{P}_{\mathcal{M}_i}\big)\,,\end{aligned} \tag{2}$$

where $\mathcal{F}_{\mathrm{DL}}(\mathbb{P}_{\mathcal{M}_i}, \mathbb{P}_i^W, \mathbb{P}_{\theta^i})$ is definied as :

$$\begin{aligned}\mathcal{F}_{\mathrm{DL}}\big(\mathbb{P}_{\mathcal{M}_i}, \mathbb{P}_i^W, \mathbb{P}_{\theta^i}\big) \triangleq\ & \big|D_{KL}\big(\mathbb{P}_{\mathcal{M}_i} \mid\mid \mathbb{P}_{\theta^i}\big) \\ & - D_{KL}\big(\mathbb{P}_i^W \mid\mid \mathbb{P}_{\theta^i}\big)\big|\end{aligned} \tag{3}$$

and $\mathcal{F}_{\mathrm{dis}}(\mathbb{P}_i^W, \mathbb{P}_{\mathcal{M}_i})$ is :

$$\begin{aligned}\mathcal{F}_{\mathrm{dis}}\big(\mathbb{P}_i^W, \mathbb{P}_{\mathcal{M}_i}\big) \triangleq\ & \mathbb{E}_{\mathbb{P}_I^W}\big[p_{W^i}(\mathbf{x}) \log p_{W^i}(\mathbf{x})\big] \\ & - \mathbb{E}_{\mathbb{P}_{\mathcal{M}_i}}\big[p_{\mathcal{M}_i}(\mathbf{x}) \log p_{\mathcal{M}_i}(\mathbf{x})\big]\end{aligned} \tag{4}$$

We can observe that $\mathcal{F}_{\mathrm{dis}}(\mathbb{P}_i^W, \mathbb{P}_{\mathcal{M}_i})$ is constant if and only if $\mathbb{P}_i^W$ and $\mathbb{P}_{\mathcal{M}_i}$ are fixed. $\mathcal{F}_{\mathrm{dis}}(\mathbb{P}_{W^i}, \mathbb{P}_{\mathcal{M}_i})$ can also be bounded by $|D_{KL}(\mathbb{P}_i^W \mid\mid \mathbb{P}_{\mathcal{M}_i}) - D_{KL}(\mathbb{P}_{\mathcal{M}_i} \mid\mid (\mathbb{P}_i^W)|$. Based on Eq. (2), we can estimate the sample log-likelihood of $\mathbb{P}_i^W$ by ELBO :

$$\begin{aligned}\mathbb{E}_{\mathbb{P}_i^W}\big[\log p_{\theta^i}(\mathbf{x})\big] \geq\ & \mathbb{E}_{\mathbb{P}_{\mathcal{M}_i}}\big[\mathcal{L}_{ELBO}(\mathbf{x}; \theta^i, \omega^i)\big] \\ & - D_{KL}\big(\mathbb{P}_i^W \mid\mid \mathbb{P}_{\mathcal{M}_i}\big) - \mathcal{F}_{\mathrm{DL}}\big(\mathbb{P}_{\mathcal{M}_i}, \mathbb{P}_i^W, \mathbb{P}_{\theta^i}\big) \\ & + \mathcal{F}_{\mathrm{dis}}\big(\mathbb{P}_i^W, \mathbb{P}_{\mathcal{M}_i}\big)\,,\end{aligned} \tag{5}$$

We find that Eq. (5) can be recovered to a standard ELBO (Eq. (1) of the paper) if and only if $\mathbb{P}_i^W$ is equal to $\mathbb{P}_{\mathcal{M}_i}$.

Proof. Firstly, we consider the JS divergence $D_{JS}(\mathbb{P}_i^W \mid\mid \mathbb{P}_{\mathcal{M}_i})$ and two KL divergences $D_{KL}(\mathbb{P}_{\mathcal{M}_i} \mid\mid \mathbb{P}_{\theta^i})$ and $D_{KL}(\mathbb{P}_i^W \mid\mid \mathbb{P}_{\theta^i})$. Then we have :

$$\begin{aligned}D_{KL}(\mathbb{P}_i^W \mid\mid \mathbb{P}_{\theta^i}) \leq\ & D_{KL}(\mathbb{P}_{\mathcal{M}_i} \mid\mid \mathbb{P}_{\theta^i}) + |D_{KL}(\mathbb{P}_{\mathcal{M}_i} \mid\mid \mathbb{P}_{\theta^i}) - D_{KL}(\mathbb{P}_i^W \mid\mid \mathbb{P}_{\theta^i})| \\ & + D_{JS}(\mathbb{P}_i^W \mid\mid \mathbb{P}_{\mathcal{M}_i})\end{aligned} \tag{6}$$

Eq. (6) holds because the sum of last two terms in RHS is larger than LHS while

$D_{KL}(\mathbb{P}_i^W \;||\; \mathbb{P}_{\mathcal{M}_i}) \geq 0$. In the following, we can rewrite $D_{KL}(\mathbb{P}_i^W \;||\; \mathbb{P}_{\theta^i})$ and $D_{KL}(\mathbb{P}_{\mathcal{M}_i} \;||\; \mathbb{P}_{\theta^i})$ as :

$$D_{KL}(\mathbb{P}_i^W \;||\; \mathbb{P}_{\theta^i}) = \mathbb{E}_{\mathbb{P}_I^W}[p_{W^i}(\mathbf{x}) \log p_{W^i}(\mathbf{x})] - \mathbb{E}_{\mathbb{P}_i^W}[\log p_{\theta^i}(\mathbf{x})] \tag{7}$$

$$D_{KL}(\mathbb{P}_{\mathcal{M}_i} \;||\; \mathbb{P}_{\theta^i}) = \mathbb{E}_{\mathbb{P}_{\mathcal{M}_i}}[p_{\mathcal{M}_i}(\mathbf{x}) \log p_{\mathcal{M}_i}(\mathbf{x})] - \mathbb{E}_{\mathbb{P}_{\mathcal{M}_i}}[\log p_{\theta^i}(\mathbf{x})] \tag{8}$$

Then we take Eq. (7) and Eq. (8) into Eq. (6), resulting in :

$$\begin{aligned}
\mathbb{E}_{\mathbb{P}_I^W}[p_{W^i}(\mathbf{x}) \log p_{W^i}(\mathbf{x})] - \mathbb{E}_{\mathbb{P}_i^W}[\log p_{\theta^i}(\mathbf{x})] \leq\; & \mathbb{E}_{\mathbb{P}_{\mathcal{M}_i}}[p_{\mathcal{M}_i}(\mathbf{x}) \log p_{\mathcal{M}_i}(\mathbf{x})] \\
& - \mathbb{E}_{\mathbb{P}_{\mathcal{M}_i}}[\log p_{\theta^i}(\mathbf{x})] \\
& + |D_{KL}(\mathbb{P}_{\mathcal{M}_i} \;||\; \mathbb{P}_{\theta^i}) - D_{KL}(\mathbb{P}_i^W \;||\; \mathbb{P}_{\theta^i})| \\
& + D_{JS}(\mathbb{P}_i^W \;||\; \mathbb{P}_{\mathcal{M}_i})
\end{aligned} \tag{9}$$

We rearrange Eq. (9) as :

$$\begin{aligned}
& - \mathbb{E}_{\mathbb{P}_i^W}[\log p_{\theta^i}(\mathbf{x})] \leq \mathbb{E}_{\mathbb{P}_{\mathcal{M}_i}}[p_{\mathcal{M}_i}(\mathbf{x}) \log p_{\mathcal{M}_i}(\mathbf{x})] - \mathbb{E}_{\mathbb{P}_I^W}[p_{W^i}(\mathbf{x}) \log p_{W^i}(\mathbf{x})] \\
& - \mathbb{E}_{\mathbb{P}_{\mathcal{M}_i}}[\log p_{\theta^i}(\mathbf{x})] + |D_{KL}(\mathbb{P}_{\mathcal{M}_i} \;||\; \mathbb{P}_{\theta^i}) - D_{KL}(\mathbb{P}_i^W \;||\; \mathbb{P}_{\theta^i})| \\
& + D_{JS}(\mathbb{P}_i^W \;||\; \mathbb{P}_{\mathcal{M}_i})
\end{aligned} \tag{10}$$

We then multiply with -1 both LHS and RHS of Eq. (10), resulting in :

$$\begin{aligned}
& \mathbb{E}_{\mathbb{P}_i^W}[\log p_{\theta^i}(\mathbf{x})] \geq -\mathbb{E}_{\mathbb{P}_{\mathcal{M}_i}}[p_{\mathcal{M}_i}(\mathbf{x}) \log p_{\mathcal{M}_i}(\mathbf{x})] + \mathbb{E}_{\mathbb{P}_I^W}[p_{W^i}(\mathbf{x}) \log p_{W^i}(\mathbf{x})] \\
& + \mathbb{E}_{\mathbb{P}_{\mathcal{M}_i}}[\log p_{\theta^i}(\mathbf{x})] - |D_{KL}(\mathbb{P}_{\mathcal{M}_i} \;||\; \mathbb{P}_{\theta^i}) - D_{KL}(\mathbb{P}_i^W \;||\; \mathbb{P}_{\theta^i})| \\
& - D_{JS}(\mathbb{P}_i^W \;||\; \mathbb{P}_{\mathcal{M}_i})
\end{aligned} \tag{11}$$

$\square$

This proves Theorem 1.

C Lemma 2

Lemma 2 Let $\{D^T(1,k), \cdots, D^T(C(T,k),k)\}$ be several target sets where each target set $D^T(c,k)$ can be divided into several data batches $\{\mathcal{B}^T(c,1), \cdots, \mathcal{B}^T(c, n(T,c,k))\}$ where $n(T,c,k)$ is the total number of data batches for $D^T(c,k)$. Let $\mathbb{P}_T^{\mathcal{B}}(c,j)$ represent the probabilistic representation of the data batch $\mathcal{B}^T(c,j)$. We suppose that $\mathbf{V}$ has already learnt t components trained on $\mathcal{M}_i$ at $\mathcal{T}_i$. The generalization performance on all target sets, achieved by $\mathbf{V}$ at $\mathcal{T}_i$, is defined as :

$$\begin{aligned} &\sum\nolimits_{c=1}^{C_k^T} \Big\{ \sum\nolimits_{j=1}^{n_{c,k}^T} \Big\{ \mathbb{E}_{\mathbb{P}^{\mathcal{B}}_{T_{(c,j)}}} \big[\log p_{\Theta^i}(\mathbf{x}) \big] \Big\} \Big\} \geq \\ &\sum\nolimits_{c=1}^{C_k^T} \Big\{ \sum\nolimits_{j=1}^{n_{c,k}^T} \Big\{ \mathcal{F}_s(\mathbb{P}^{\mathcal{B}}_T(c,j), \mathbf{V}) \Big\} \Big\} , \end{aligned} \tag{12}$$

which is Eq. (10) in the paper.

Similar to the conclusion of Theorem 2, increasing the number of components in $\mathbf{V}$ would lead to better generalization performance on all target sets. In practice, we can not use Eq.(10) of the paper for the component selection since it involves several intractable terms. Instead, we perform the component selection by comparing the sample log-likelihood, expressed as :

$$\mathcal{F}^r_s(\mathbb{P}^{\mathcal{B}}_j, \mathbf{V}) \stackrel{\Delta}{=} \arg \max_{i=1,\cdots,t} \Big\{ \mathbb{E}_{\mathbb{P}^{\mathcal{B}}_j} \big[\mathcal{L}_{ELBO}(\mathbf{x}; \theta_i^{c_i}, \omega_i^{c_i}) \big] \Big\} , \tag{13}$$

We can rewrite Eq. (12) by using the component selection function (Eq. (13)) :

$$\begin{aligned} &\sum\nolimits_{c=1}^{C_k^T} \Big\{ \sum\nolimits_{j=1}^{n_{c,k}^T} \Big\{ \mathbb{E}_{\mathbb{P}^{\mathcal{B}}_{T_{(c,j)}}} \big[\log p_{\Theta^i}(\mathbf{x}) \big] \Big\} \Big\} \geq \\ &\sum\nolimits_{c=1}^{C_k^T} \Big\{ \sum\nolimits_{j=1}^{n_{c,k}^T} \Big\{ \mathcal{F}^r_s(\mathbb{P}^{\mathcal{B}}_T(c,j), \mathbf{V}) \Big\} \Big\} , \end{aligned} \tag{14}$$

Since Eq. (13) can not guarantee the best performance for Eq. (14) because Eq. (13) does not always select the component that has the highest function value (RHS of Eq. (14)). We can measure this error caused by Eq. (13) as :

$$\sum_{c=1}^{C_k^T} \Big\{ \sum_{j=1}^{n_{c,k}^T} \Big\{ \mathcal{F}_s(\mathbb{P}^{\mathcal{B}}_T(c,j), \mathbf{V}) - \mathcal{F}^r_s(\mathbb{P}^{\mathcal{B}}_T(c,j), \mathbf{V}) \Big\} \Big\} , \tag{15}$$

From Eq. (14), we can observe that the generalization performance of $\mathbf{V}$ on all target sets relies on not only the KL divergence term but also the error caused by the component selection. In the following section, we study how component diversity in a mixture system affects performance.

D Lemma 3

In this section, we investigate whether a large number of components in a DEM can always ensure an optimal performance.

Lemma 3 Let $\{D^T(1,k),\cdots,D^T(C(T,k),k)\}$ be several target sets and we use $\mathbb{P}^T_{(j,k)}$ to denote the probabilistic representation of $D^T(j,k)$. Learning a large number of components in $\mathcal{V}$ can not always ensure an optimal performance.

Proof We assume that $\mathcal{V}$ has learnt $u > C(T,k)$ components and several components would capture the information corresponding to a unique target data distribution. We assume that a certain target distribution is ignored by all components. According to Theorem 3, Eq. (11) of the paper, we have :

$$\sum_{j=1}^{C'} \Big\{ \mathbb{E}_{\mathbb{P}^T_{(j,k)}} \big[\log p_{\Theta^i}(\mathbf{x}) \big] \Big\} \geq \sum_{j=1}^{C'} \Big\{ \mathcal{F}_{\mathrm{dis}}\big(\mathbb{P}^T_{(j,k)}, \mathbb{P}_{\mathcal{M}_{c_{1:t}}}\big) + \mathbb{E}_{\mathbb{P}_{\mathcal{M}_{c_{1:t}}}} \big[\mathcal{L}_{ELBO}(\mathbf{x};\Theta^i,\Omega^i) \big] - D_{JS}\big(\mathbb{P}^T_{(j,k)} \mid\mid \mathbb{P}_{\mathbb{P}_{\mathcal{M}_{c_{1:t}}}}\big) - \mathcal{F}_{\mathrm{DL}}\big(\mathbb{P}_{\mathcal{M}_{c_{1:t}}}, \mathbb{P}^T_{(j,k)}, \mathbb{P}_{\Theta^i}\big) \Big\}, \tag{16}$$

□

From Eq. (16), we observe that the model $\mathcal{V}$ can not achieve the optimal performance due to the error caused by the JS divergence term.

E Theoretical analysis for the expansion threshold

In this section, we provide the theoretical analysis for the expansion threshold β from Eq. (13) of the paper. From **Theorem 3**, we know that encouraging the knowledge diversity would improve the generalization performance of the DEM with a minimal number of components. A large expansion threshold β in Eq.(15) of the paper can ensure knowledge diversity among components, leading however to losing some information learnt previously. According to **Theorem 3**, we have :

$$\sum_{j=1}^{C'} \Big\{ \mathbb{E}_{\mathbb{P}^T_{(j,k)}} \big[\log p_{\Theta^i}(\mathbf{x}) \big] \Big\} \geq \sum_{j=1}^{C'} \Big\{ \mathcal{F}_{\text{dis}} \big(\mathbb{P}^T_{(j,k)}, \mathbb{P}_{\mathcal{M}_{c_{1:t}}} \big) + \mathbb{E}_{\mathbb{P}_{\mathcal{M}_{c_{1:t}}}} \big[\mathcal{L}_{ELBO}(\mathbf{x}; \Theta^i, \Omega^i) \big] - D_{JS} \big(\mathbb{P}^T_{(j,k)} \mid\mid \mathbb{P}_{\mathbb{P}_{\mathcal{M}_{c_{1:t}}}} \big) - \mathcal{F}_{\text{DL}} \big(\mathbb{P}_{\mathcal{M}_{c_{1:t}}}, \mathbb{P}^T_{(j,k)}, \mathbb{P}_{\Theta^i} \big) \Big\}, \tag{17}$$

if we choose a very large expansion threshold β, the model $\mathbf{V}$ would use fewer components while some underlying data distributions $\{\mathbb{P}^T_{j,1}, \mathbb{P}^T_{j,2}\}$ would not be captured by the components. Therefore, RHS of Eq. (17) would be increased, resulting in a degenerated performance. In contrast, a minimal expansion threshold β can allow $\mathbf{V}$ to create more components, which would capture all underlying data distributions. However, this would also lead to an increase in the number of parameters. Therefore, a suitable trade-off for the expansion threshold β can ensure good performance with a fair number of components.

F Theoretical analysis for other VAE models

In this section, we extend the proposed theoretical framework to analyze the forgetting behaviour of the existing VAE models.

F.1 Importance weighted autoencoders

The Importance Weighted Autoencoder (IWELBO) Burda et al. (2015) is another VAE model employing a recognition network to generate multiple samples during the optimization to lead to better modelling of the posterior probabilities. The corresponding ELBO for sampling K' samples is defined as :

$$\mathcal{L}_{ELBO_{K'}}(\mathbf{x}; \theta, \omega) = \mathbb{E}_{\mathbf{z}_1,..,\mathbf{z}_{K'} \sim q(\mathbf{z}|\mathbf{x})} \left[\log \frac{1}{K'} \sum_{i=1}^{K'} \frac{p(\mathbf{x}, \mathbf{z}_i)}{q(\mathbf{z}_i|\mathbf{x})} \right] \tag{18}$$

where K' is the number of weighted samples and $K' = 1$ is equivalent to the standard ELBO.

Lemma 4 Let $p_{\theta_i}(\mathbf{x})$ be a probability density function for a single model $\mathcal{V}^i$ updated at $\mathcal{T}_i$. Let $\mathbb{P}^W_i$ denote a distribution of all visited data batches $\{\mathcal{B}_1, \cdots, \mathcal{B}_i\}$ drawn from W at $\mathcal{T}_i$. Let $p_{\mathcal{M}_i}(\mathbf{x})$ and $p_{W^i}(\mathbf{x})$ denote the density function for $\mathbb{P}_{\mathcal{M}_i}$ and $\mathbb{P}^W_i$, respectively. We then derive an upper bound for a single VAE model trained on $\mathcal{M}_i$ at $\mathcal{T}_i$ as :

$$\begin{aligned}\mathbb{E}_{\mathbb{P}_i^W}\left[\log p_{\theta^i}(\mathbf{x})\right] \geq\ & \mathbb{E}_{\mathbf{x}\sim\mathbb{P}_{\mathcal{M}_i}}\mathbb{E}_{\mathbf{z_1},..,\mathbf{z}_{K'}\sim q(\mathbf{z}|\mathbf{x})}\left[\log\frac{1}{K'}\sum_{i=1}^{K'}\frac{p\left(\mathbf{x},\mathbf{z}_i\right)}{q\left(\mathbf{z}_i|\mathbf{x}\right)}\right] \\ & - D_{JS}\big(\mathbb{P}_i^W \mid\mid \mathbb{P}_{\mathcal{M}_i}\big) - \mathcal{F}_{\mathrm{DL}}\big(\mathbb{P}_{\mathcal{M}_i}, \mathbb{P}_i^W, \mathbb{P}_{\theta^i}\big) \\ & + \mathcal{F}_{\mathrm{dis}}\big(\mathbb{P}_i^W, \mathbb{P}_{\mathcal{M}_i}\big)\,,\end{aligned} \tag{19}$$

Proof Since we have $\mathcal{L}_{ELBO_{K'}}(\mathbf{x};\theta^i,\omega^i) < \mathcal{L}_{ELBO_{K'+1}}(\mathbf{x};\theta^i,\omega^i)$, we can replace $\mathcal{L}_{ELBO}(\mathbf{x};\theta^i,\omega^i)$ by $\mathcal{L}_{ELBO_{K'+1}}(\mathbf{x};\theta^i,\omega^i)$ in Eq. (5), resulting in Eq. (19).

From Eq. (19), we can observe that by increasing the number of weighted samples K' can not ensure an optimal performance since RHS of Eq. (19) is also relying on the JS divergence term $D_{JS}\big(\mathbb{P}_i^W \mid\mid \mathbb{P}_{\mathcal{M}_i}\big)$.

F.2 Hierarchical Variational Inference

In this section, we extend our theoretical analysis to the Auxiliary Deep Generative Models (ADGM) Maaløe et al. (2016). ADGM is a classical hierarchical latent variable model. ADGM introduces an auxiliary variable $\mathbf{a}$ into the variational distribution $q(\mathbf{a},\mathbf{z}\mid\mathbf{x}) = q(\mathbf{z}\mid\mathbf{a},\mathbf{x})q(\mathbf{a}\mid\mathbf{x})$ and its ELBO is expressed as :

$$\begin{aligned}\log p\left(\mathbf{x}\right) = \log\iint p\left(\mathbf{x},\mathbf{a},\mathbf{z}\right)d\mathbf{a}d\mathbf{z} \geq \mathbb{E}_{q(\mathbf{a},\mathbf{z}|\mathbf{x})}\left[\log\frac{p\left(\mathbf{a}\mid\mathbf{z},\mathbf{x}\right)p\left(\mathbf{x}\mid\mathbf{z}\right)p\left(\mathbf{z}\right)}{q\left(\mathbf{a}\mid\mathbf{x}\right)q\left(\mathbf{z}\mid\mathbf{a},\mathbf{x}\right)}\right] \\ = \mathcal{L}_{ADGM}\left(\mathbf{x};\theta,\omega\right)\ .\end{aligned} \tag{20}$$

Then according to the results from **Lemma 4**, we have :

$$\begin{aligned}\mathbb{E}_{\mathbb{P}_i^W}\left[\log p_{\theta^i}(\mathbf{x})\right] \geq\ & \mathbb{E}_{\mathbb{P}_{\mathcal{M}_i}}\mathcal{L}_{ADGM}\left(\mathbf{x};\theta^i,\omega^i\right) \\ & - D_{KL}\big(\mathbb{P}_i^W \mid\mid \mathbb{P}_{\mathcal{M}_i}\big) - \mathcal{F}_{\mathrm{DL}}\big(\mathbb{P}_{\mathcal{M}_i}, \mathbb{P}_i^W, \mathbb{P}_{\theta^i}\big) \\ & + \mathcal{F}_{\mathrm{dis}}\big(\mathbb{P}_i^W, \mathbb{P}_{\mathcal{M}_i}\big)\,,\end{aligned} \tag{21}$$

Similar to **Lemma 4**, maximizing $\mathcal{L}_{ADGM}\left(\mathbf{x};\theta^i,\omega^i\right)$ only can not ensure the optimal performance on the target distribution $\mathbb{P}_i^W$. These results show that deriving a tight ELBO on the source distribution can not ensure the optimal performance on the target distribution under continual learning.

G Theoretical analysis for task-known continual learning

In this section, we extend our theoretical analysis to a general continual learning where the task boundary is known during the training.

G.1 Theoretical analysis for the static network architecture

In this section, we firstly provide theoretical analysis for the static model and then extend it to the dynamic expansion model in the next section. We provide the necessary notations below.

G.1.1 Memory-based model

Definition 5. Let $\mathcal{C} = \{\mathcal{C}_1, \cdots, \mathcal{C}_n\}$ be a sequence of tasks, where each task $\mathcal{C}_i$ is associated with the training dataset D_i^S as well as the testing dataset D_i^S. Let $\mathbb{P}_i^T$ and $\mathbb{P}_i^S$ represent the probabilistic representation of D_i^T and D_i^S, respectively. First, we provide the theoretical analysis for a single VAE model learning a sequence of n tasks.

Definition 6. In the task-known continual learning, let $\mathcal{M}_i$ represent the memory buffer updated at the i-th task learning (does not store the samples from the i-th task).

Lemma 5 Let $\{\mathbb{P}_1^T, \cdots, \mathbb{P}_n^T\}$ be the distribution of the testing sets from a sequence of n tasks $\{\mathcal{C}_1, \cdots, \mathcal{C}_n\}$. Let $p_{\theta^i}(\mathbf{x})$ be a probability density function for a single model $\mathcal{V}^i$ updated at $\mathcal{T}_i$. Let $p_{\mathcal{M}_i}(\mathbf{x})$ and $p_{T^i}(\mathbf{x})$ denote the density function for $\mathbb{P}_{\mathcal{M}_i}$ and $\mathbb{P}_i^T$, respectively. Let $\mathbb{P}_j^S \otimes \mathbb{P}_{\mathcal{M}_j}$ be the combined distribution of D_j^S and $\mathcal{M}_j$. We then derive a lower bound for a single VAE model trained on $\mathcal{M}_j$ at $\mathcal{C}_j$ as :

$$\begin{aligned}\sum_{i=1}^{n} \big\{ \mathbb{E}_{\mathbb{P}_i^T} \big[\log p_{\theta^i}(\mathbf{x}) \big] \big\} \geq \sum_{i=1}^{n} \big\{ & \mathbb{E}_{\mathbb{P}_{\mathcal{M}_j} \otimes \mathbb{P}_j^S} \big[\mathcal{L}_{ELBO}(\mathbf{x}; \theta^i, \omega^i) \big] \\ & - D_{JS}\big(\mathbb{P}_i^T \mid\mid \mathbb{P}_{\mathcal{M}_j} \otimes \mathbb{P}_j^S \big) \\ & - \mathcal{F}_{\mathrm{DL}}\big(\mathbb{P}_{\mathcal{M}_j} \otimes \mathbb{P}_j^S, \mathbb{P}_i^T, \mathbb{P}_{\theta^i} \big) \\ & + \mathcal{F}_{\mathrm{dis}}\big(\mathbb{P}_i^T, \mathbb{P}_{\mathcal{M}_j} \otimes \mathbb{P}_j^S \big) \big\} \,, \end{aligned} \tag{22}$$

Eq. (22) can explain the forgetting behaviour of the existing memory-based methods Ye and Bors (2022a); Egorov et al. (2021); Deja et al. (2021) when learning a sequence of n tasks. As learning more tasks (j is increased), the model would accumulate more knowledge over time and can thus improve its performance. However, when the memory buffer $\mathcal{M}_j$ has a fixed capacity, the model would suffer from a performance loss on past tasks since the memory buffer can not preserve all previously learnt knowledge.

In the following, we study the forgetting behaviour for the generative replay approaches that trains a generator to replay past samples. Unlike the memory-based approach, the generative replay approaches can provide an infinite number of generative replay samples over time.

G.1.2 GRM-based model

Definition 7. (Generative Replay Mechanism (GRM)). Let $\mathbb{P}_{\widehat{\mathbf{x}}^i}$ represent the distribution of the generative replay samples drawn from a single VAE model that has learnt i tasks $\{\mathcal{C}_1, \cdots, \mathcal{C}_i\}$. Let $F_{\mathcal{C}} \colon \mathcal{X} \to \mathcal{C}$ be an optimal task-inference model that always return the true task label for a given input $\mathbf{x}$. By using $F_{\mathcal{C}}$, we can form an approximate distribution $\mathbb{P}_{\widehat{\mathbf{x}}(j,i-j)}$ through the sampling process $\mathbf{x} \sim \mathbb{P}_{\widehat{\mathbf{x}}^i}$ if $F_{\mathcal{C}}(\mathbf{x}) = j$ where the superscript $i-j$ denotes that $\mathbb{P}^S_j = \mathbb{P}_{\widehat{\mathbf{x}}(j,0)}$ is transformed to $\mathbb{P}_{\widehat{\mathbf{x}}(j,i-j)}, j < i$ through $i-j$ GRM processes. Therefore, $\mathbb{P}_{\widehat{\mathbf{x}}(j,i-j)}$ is an approximate distribution of the training set of a certain task ($\mathcal{C}_j$), generated by a single model that has learnt i tasks.

Based on the definition of the GRM process, we derive new theoretical analysis for the GRM-based methods in the following.

Lemma 6 Let $\{\mathbb{P}^T_1, \cdots, \mathbb{P}^T_n\}$ be the distribution of the testing sets from a sequence of n tasks $\{\mathcal{C}_1, \cdots, \mathcal{C}_n\}$. Let $p_{\theta^i}(\mathbf{x})$ be a probability density function for a single model $\mathcal{V}^i$ updated at $\mathcal{T}_i$. We assume that a VAE model is enabled with the GRM process to relieve forgetting. We then derive a lower bound for the j-th task, achieved by a single VAE model trained at $\mathcal{C}_i$ as :

$$\begin{aligned}
\mathbb{E}_{\mathbb{P}^T_j}\big[\log p_{\theta^i}(\mathbf{x})\big] \geq\ & \mathbb{E}_{\mathbb{P}_{\widehat{\mathbf{x}}(j,i-j)} \otimes \mathbb{P}^S_i}\big[\mathcal{L}_{ELBO}(\mathbf{x};\theta^i,\omega^i)\big] \\
& - D_{JS}\big(\mathbb{P}^T_j \mid\mid \mathbb{P}_{\widehat{\mathbf{x}}(j,i-j)} \otimes \mathbb{P}^S_i\big) \\
& - \mathcal{F}_{\mathrm{DL}}\big(\mathbb{P}_{\widehat{\mathbf{x}}(j,i-j)} \otimes \mathbb{P}^S_i, \mathbb{P}^T_j, \mathbb{P}_{\theta^i}\big) \\
& + \mathcal{F}_{\mathrm{dis}}\big(\mathbb{P}^T_j, \mathbb{P}_{\widehat{\mathbf{x}}(j,i-j)} \otimes \mathbb{P}^S_i\big)\,,
\end{aligned} \tag{23}$$

From Eq. (23), it observes that the quality of the approximate distribution $\mathbb{P}_{\widehat{\mathbf{x}}(j,i-j)}$ is crucial for the performance on the j-th task, achieved by a single VAE model trained at $\mathcal{C}_i$. Specifically, if $\mathbb{P}_{\widehat{\mathbf{x}}(j,i-j)}$ can approximate the corresponding data distribution $\mathbb{P}^T_j$ exactly, RHS of Eq. (23) would be large, resulting in better performance. As the number of tasks to be increased (i is increased), the model would suffer from more forgetting because the approximate distribution $\mathbb{P}_{\widehat{\mathbf{x}}(j,i-j)}$ would far away from $\mathbb{P}^T_j$, caused by the frequent GRM process ($i-j$ is large). In the following, we derive a new lower bound that describes how a single VAE model forgets its previously learnt knowledge in each task learning.

Lemma 7 Let $\{\mathbb{P}^T_1, \cdots, \mathbb{P}^T_n\}$ be the distribution of the testing sets from a sequence of n tasks $\{\mathcal{C}_1, \cdots, \mathcal{C}_n\}$. Let $p_{\theta^i}(\mathbf{x})$ be a probability density function for a single model $\mathcal{V}^i$ updated at $\mathcal{T}_i$. We assume that a VAE model is enabled with the GRM process to relieve forgetting. We then derive a lower bound for the j-th task, achieved by a single VAE model trained at $\mathcal{C}_i$ as :

$$
\begin{aligned}
&\mathbb{E}_{\mathbb{P}^T_j}\big[\log p_{\theta^i}(\mathbf{x})\big] \geq \mathbb{E}_{\mathbb{P}_{\widehat{\mathbf{x}}(j,i-j)} \otimes \mathbb{P}^S_i}\big[\mathcal{L}_{ELBO}(\mathbf{x};\theta^i,\omega^i)\big] - D_{JS}\big(\mathbb{P}^T_j \,||\, \mathbb{P}_{\widehat{\mathbf{x}}(j,0)} \otimes \mathbb{P}^S_i\big) \\
&- \mathcal{F}_{\mathrm{DL}}\big(\mathbb{P}_{\widehat{\mathbf{x}}(j,0)} \otimes \mathbb{P}^S_i, \mathbb{P}^T_j, \mathbb{P}_{\theta^i}\big) + \mathcal{F}_{\mathrm{dis}}\big(\mathbb{P}^T_j, \mathbb{P}_{\widehat{\mathbf{x}}(j,0)} \otimes \mathbb{P}^S_i\big) \\
&- \sum_{c=2}^{i-j}\Big\{D_{JS}\big(\mathbb{P}_{\widehat{\mathbf{x}}(j,c-1)} \,||\, \mathbb{P}_{\widehat{\mathbf{x}}(j,c)} \otimes \mathbb{P}^S_i\big) \\
&+ \mathcal{F}_{\mathrm{DL}}\big(\mathbb{P}_{\widehat{\mathbf{x}}(j,c)} \otimes \mathbb{P}^S_i, \mathbb{P}_{\widehat{\mathbf{x}}(j,c-1)}, \mathbb{P}_{\theta^i}\big) - \mathcal{F}_{\mathrm{dis}}\big(\mathbb{P}_{\widehat{\mathbf{x}}(j,c-1)}, \mathbb{P}_{\widehat{\mathbf{x}}(j,c)} \otimes \mathbb{P}^S_i\big)\Big\}, \qquad (24)
\end{aligned}
$$

Remark. We have several observations from Lemma 7 :

- As the number of tasks increases (i is increased), the performance on the j-th task, achieved by a single model, would be degenerated caused by the accumulated JS divergence terms (the fifth term in RHS of Eq. (24)).
- Learning early tasks would suffer from more forgetting than the learning of the recent tasks because early tasks cause more accumulated JS divergence terms ($i-j$ is large in RHS of Eq. (24)).

Proof. First, we consider $\mathbb{P}^T_j$ and $\mathbb{P}_{\widehat{\mathbf{x}}(j,1)}$ as the target and source distribution, respectively. Based on Lemma 6, we have :

$$
\begin{aligned}
\mathbb{E}_{\mathbb{P}^T_j}\big[\log p_{\theta^i}(\mathbf{x})\big] \geq\; &\mathbb{E}_{\mathbb{P}_{\widehat{\mathbf{x}}(j,0)} \otimes \mathbb{P}^S_i}\big[\mathcal{L}_{ELBO}(\mathbf{x};\theta^i,\omega^i)\big] \\
&- D_{JS}\big(\mathbb{P}^T_j \,||\, \mathbb{P}_{\widehat{\mathbf{x}}(j,0)} \otimes \mathbb{P}^S_i\big) \\
&- \mathcal{F}_{\mathrm{DL}}\big(\mathbb{P}_{\widehat{\mathbf{x}}(j,0)} \otimes \mathbb{P}^S_i, \mathbb{P}^T_j, \mathbb{P}_{\theta^i}\big) \\
&+ \mathcal{F}_{\mathrm{dis}}\big(\mathbb{P}^T_j, \mathbb{P}_{\widehat{\mathbf{x}}(j,0)} \otimes \mathbb{P}^S_i\big), \qquad (25)
\end{aligned}
$$

Then, we consider $\mathbb{P}_{\widehat{\mathbf{x}}(j,0)}$ and $\mathbb{P}_{\widehat{\mathbf{x}}(j,1)}$ as the target and source data distribution, respectively. This setting is reasonable since $\mathbb{P}_{\widehat{\mathbf{x}}(j,0)}$ is more closed to the distribution $\mathbb{P}^T_j$ of the testing set compared with $\mathbb{P}_{\widehat{\mathbf{x}}(j,1)}$. We can derive a lower bound between

$\mathbb{P}_{\widehat{\mathbf{x}}(j,0)}$ and $\mathbb{P}_{\widehat{\mathbf{x}}(j,1)}$.

$$\begin{aligned}
\mathbb{E}_{\mathbb{P}_{\widehat{\mathbf{x}}(j,0)}}\big[\log p_{\theta^i}(\mathbf{x})\big] \geq\ & \mathbb{E}_{\mathbb{P}_{\widehat{\mathbf{x}}(j,1)} \otimes \mathbb{P}_i^S}\big[\mathcal{L}_{ELBO}(\mathbf{x};\theta^i,\omega^i)\big] \\
& - D_{JS}\big(\mathbb{P}_{\widehat{\mathbf{x}}(j,0)} \,||\, \mathbb{P}_{\widehat{\mathbf{x}}(j,1)} \otimes \mathbb{P}_i^S\big) \\
& - \mathcal{F}_{\mathrm{DL}}\big(\mathbb{P}_{\widehat{\mathbf{x}}(j,1)} \otimes \mathbb{P}_i^S, \mathbb{P}_{\widehat{\mathbf{x}}(j,0)}, \mathbb{P}_{\theta^i}\big) \\
& + \mathcal{F}_{\mathrm{dis}}\big(\mathbb{P}_{\widehat{\mathbf{x}}(j,0)}, \mathbb{P}_{\widehat{\mathbf{x}}(j,1)} \otimes \mathbb{P}_i^S\big)\,,
\end{aligned} \tag{26}$$

Then we treat $\mathbb{P}_{\widehat{\mathbf{x}}(j,1)}$ and $\mathbb{P}_{\widehat{\mathbf{x}}(j,2)}$ as the target and source distribution, respectively. We have :

$$\begin{aligned}
\mathbb{E}_{\mathbb{P}_{\widehat{\mathbf{x}}(j,1)}}\big[\log p_{\theta^i}(\mathbf{x})\big] \geq\ & \mathbb{E}_{\mathbb{P}_{\widehat{\mathbf{x}}(j,2)} \otimes \mathbb{P}_i^S}\big[\mathcal{L}_{ELBO}(\mathbf{x};\theta^i,\omega^i)\big] \\
& - D_{JS}\big(\mathbb{P}_{\widehat{\mathbf{x}}(j,1)} \,||\, \mathbb{P}_{\widehat{\mathbf{x}}(j,2)} \otimes \mathbb{P}_i^S\big) \\
& - \mathcal{F}_{\mathrm{DL}}\big(\mathbb{P}_{\widehat{\mathbf{x}}(j,2)} \otimes \mathbb{P}_i^S, \mathbb{P}_{\widehat{\mathbf{x}}(j,1)}, \mathbb{P}_{\theta^i}\big) \\
& + \mathcal{F}_{\mathrm{dis}}\big(\mathbb{P}_{\widehat{\mathbf{x}}(j,1)}, \mathbb{P}_{\widehat{\mathbf{x}}(j,2)} \otimes \mathbb{P}_i^S\big)\,,
\end{aligned} \tag{27}$$

According to the summary, we can have the following bounds :

$$\begin{aligned}
& \mathbb{E}_{\mathbb{P}_{\widehat{\mathbf{x}}(j,2)}}\big[\log p_{\theta^i}(\mathbf{x})\big] \geq \mathbb{E}_{\mathbb{P}_{\widehat{\mathbf{x}}(j,3)} \otimes \mathbb{P}_i^S}\big[\mathcal{L}_{ELBO}(\mathbf{x};\theta^i,\omega^i)\big] \\
& - D_{JS}\big(\mathbb{P}_{\widehat{\mathbf{x}}(j,2)} \,||\, \mathbb{P}_{\widehat{\mathbf{x}}(j,3)} \otimes \mathbb{P}_i^S\big) \\
& - \mathcal{F}_{\mathrm{DL}}\big(\mathbb{P}_{\widehat{\mathbf{x}}(j,3)} \otimes \mathbb{P}_i^S, \mathbb{P}_{\widehat{\mathbf{x}}(j,2)}, \mathbb{P}_{\theta^i}\big) \\
& + \mathcal{F}_{\mathrm{dis}}\big(\mathbb{P}_{\widehat{\mathbf{x}}(j,2)}, \mathbb{P}_{\widehat{\mathbf{x}}(j,3)} \otimes \mathbb{P}_i^S\big)\,, \\
& \cdots \\
& \mathbb{E}_{\mathbb{P}_{\widehat{\mathbf{x}}(j,i-j-1)}}\big[\log p_{\theta^i}(\mathbf{x})\big] \geq \mathbb{E}_{\mathbb{P}_{\widehat{\mathbf{x}}(j,i-j)} \otimes \mathbb{P}_i^S}\big[\mathcal{L}_{ELBO}(\mathbf{x};\theta^i,\omega^i)\big] \\
& - D_{JS}\big(\mathbb{P}_{\widehat{\mathbf{x}}(j,i-j-1)} \,||\, \mathbb{P}_{\widehat{\mathbf{x}}(j,i-j)} \otimes \mathbb{P}_i^S\big) \\
& - \mathcal{F}_{\mathrm{DL}}\big(\mathbb{P}_{\widehat{\mathbf{x}}(j,i-j)} \otimes \mathbb{P}_i^S, \mathbb{P}_{\widehat{\mathbf{x}}(j,i-j-1)}, \mathbb{P}_{\theta^i}\big) \\
& + \mathcal{F}_{\mathrm{dis}}\big(\mathbb{P}_{\widehat{\mathbf{x}}(j,i-j-1)}, \mathbb{P}_{\widehat{\mathbf{x}}(j,i-j)} \otimes \mathbb{P}_i^S\big)\,,
\end{aligned}$$

In the final, we sum up the above bounds, resulting in :

$$
\begin{aligned}
\mathbb{E}_{\mathbb{P}_j^T}\big[\log p_{\theta^i}(\mathbf{x})\big] \geq\ & \mathbb{E}_{\mathbb{P}_{\widehat{\mathbf{x}}^{(j,i-j)}}\otimes\mathbb{P}_i^S}\big[\mathcal{L}_{ELBO}(\mathbf{x};\theta^i,\omega^i)\big] \\
& - D_{JS}\big(\mathbb{P}_j^T \,||\, \mathbb{P}_{\widehat{\mathbf{x}}^{(j,0)}} \otimes \mathbb{P}_i^S\big) \\
& - \mathcal{F}_{\mathrm{DL}}\big(\mathbb{P}_{\widehat{\mathbf{x}}^{(j,0)}} \otimes \mathbb{P}_i^S, \mathbb{P}_j^T, \mathbb{P}_{\theta^i}\big) + \mathcal{F}_{\mathrm{dis}}\big(\mathbb{P}_j^T, \mathbb{P}_{\widehat{\mathbf{x}}^{(j,0)}} \otimes \mathbb{P}_i^S\big) \\
& - \sum_{c=2}^{i-j}\{D_{JS}\big(\mathbb{P}_{\widehat{\mathbf{x}}^{(j,c-1)}} \,||\, \mathbb{P}_{\widehat{\mathbf{x}}^{(j,c)}} \otimes \mathbb{P}_i^S\big) \\
& - \mathcal{F}_{\mathrm{DL}}\big(\mathbb{P}_{\widehat{\mathbf{x}}^{(j,c)}} \otimes \mathbb{P}_i^S, \mathbb{P}_{\widehat{\mathbf{x}}^{(j,c-1)}}, \mathbb{P}_{\theta^i}\big) \\
& + \mathcal{F}_{\mathrm{dis}}\big(\mathbb{P}_{\widehat{\mathbf{x}}^{(j,c-1)}}, \mathbb{P}_{\widehat{\mathbf{x}}^{(j,c)}} \otimes \mathbb{P}_i^S\big)\}\,,
\end{aligned} \tag{28}
$$

□

Based on Lemma 7, we can derive a lower bound for all tasks in the following.

Lemma 8 Let $\{\mathbb{P}_1^T, \cdots, \mathbb{P}_n^T\}$ be the distribution of the testing sets from a sequence of n tasks $\{\mathcal{C}_1, \cdots, \mathcal{C}_n\}$. Let $p_{\theta^i}(\mathbf{x})$ be a probability density function for a single model $\mathcal{V}^i$ updated at $\mathcal{T}_i$. We assume that a VAE model is enabled with the GRM process to relieve forgetting. We then derive a lower bound for all tasks, achieved by a single VAE model trained at $\mathcal{C}_i$ as :

$$
\begin{aligned}
& \sum_{j=1}^{i}\Big\{\mathbb{E}_{\mathbb{P}_j^T}\big[\log p_{\theta_j}(\mathbf{x})\big]\Big\} \geq \sum_{j=1}^{i}\Big\{\mathbb{E}_{\mathbb{P}_{\widehat{\mathbf{x}}^{(j,i-j)}}\otimes\mathbb{P}_i^S}\big[\mathcal{L}_{ELBO}(\mathbf{x};\theta^i,\omega^i)\big] \\
& - D_{JS}\big(\mathbb{P}_j^T \,||\, \mathbb{P}_{\widehat{\mathbf{x}}^{(j,0)}} \otimes \mathbb{P}_i^S\big) - \mathcal{F}_{\mathrm{DL}}\big(\mathbb{P}_{\widehat{\mathbf{x}}^{(j,0)}} \otimes \mathbb{P}_i^S, \mathbb{P}_j^T, \mathbb{P}_{\theta^i}\big) \\
& + \mathcal{F}_{\mathrm{dis}}\big(\mathbb{P}_j^T, \mathbb{P}_{\widehat{\mathbf{x}}^{(j,0)}} \otimes \mathbb{P}_i^S\big) - \sum_{c=2}^{i-j}\Big\{D_{JS}\big(\mathbb{P}_{\widehat{\mathbf{x}}^{(j,c-1)}} \,||\, \mathbb{P}_{\widehat{\mathbf{x}}^{(j,c)}} \otimes \mathbb{P}_i^S\big) \\
& + \mathcal{F}_{\mathrm{DL}}\big(\mathbb{P}_{\widehat{\mathbf{x}}^{(j,c)}} \otimes \mathbb{P}_i^S, \mathbb{P}_{\widehat{\mathbf{x}}^{(j,c-1)}}, \mathbb{P}_{\theta^i}\big) - \mathcal{F}_{\mathrm{dis}}\big(\mathbb{P}_{\widehat{\mathbf{x}}^{(j,c-1)}}, \mathbb{P}_{\widehat{\mathbf{x}}^{(j,c)}} \otimes \mathbb{P}_i^S\big)\Big\}\Big\}\,,
\end{aligned} \tag{29}
$$

From Eq. (29), a robust generative replay network plays a vital role in the GRM-based model's performance. If the generative replay network can approximate each target data distribution precisely, the RHS of Eq. (29) would be increased, and thus, the model can achieve optimal performance on all target distributions. However, In practice, the generator such as VAE or GAN, can not produce reasonable generative replay samples when learning a sequence of different data domains. In the following section, we extend the proposed theoretical analysis to the dynamic expansion model that can overcome the limitations of the static models.

G.2 Theoretical analysis for the dynamic expansion model

In this section, we study the forgetting behaviour of the dynamic expansion model in a general continual learning setting where the task boundary is known during the training.

Lemma 9 Let $\mathbf{V} = \{\mathcal{V}_1, \cdots, \mathcal{V}_t\}$ be a mixture model with t components. We consider an idea solution that the number of components is equal to the number of tasks ($t = b$). We then derive a lower bound for $\mathbf{V}$ as :

$$\sum_{j=1}^{n} \Big\{ \mathbb{E}_{\mathbb{P}_j^T} \big[\log p_{\theta_j}(\mathbf{x}) \big] \Big\} \geq \sum_{j=1}^{n} \Big\{ \mathbb{E}_{\mathbb{P}_j^S} \big[\mathcal{L}_{ELBO}(\mathbf{x}; \theta^i, \omega^i) \big] - D_{JS}\big(\mathbb{P}_j^T \,||\, \mathbb{P}_j^S\big) - \mathcal{F}_{\mathrm{DL}}\big(\mathbb{P}_j^S, \mathbb{P}_j^T, \mathbb{P}_{\theta_j}\big) + \mathcal{F}_{\mathrm{dis}}\big(\mathbb{P}_j^T, \mathbb{P}_j^S\big) \Big\}, \tag{30}$$

From Eq. (30), it finds that the mixture model $\mathbf{V}$ can achieve optimal performance on all tasks since it does not cause forgetting during the training. In practice, the mixture model $\mathbf{V}$ would dynamically change its network architecture according to the complexity of the tasks. In the following, we derive new lower bound for $\mathbf{V}$ that can dynamically change the number of components during the training.

Lemma 10 Let $B = \{b_1, \ldots, b_j\}$ denote a set of labels where each one b_i represents the task that is only trained once. The corresponding distribution for b_i is defined as $\mathbb{P}_{\mathbf{x}^{(i,0)}}$. Let $C = \{c_1, \cdots, c_j\}$ be a set where each c_i denotes the index of the component that learns the b_i-th task. Let $B' = \{b'_1, \ldots, b'_n\}$ represent a set of labels where each one b'_j indicates the b'_j-th task was used for re-training more than once. Let $C' = \{c'_1, \cdots, c'_n\}$ represent a set where each c'_i denotes that the index of the component that learns the b'_i-th task. We also define a set $\hat{B} = \{\hat{b}_1, \ldots, \hat{b}_n\}$ where each one $\hat{b}_i > 1$ denotes that the b'_i-th task has been learnt for $\hat{b}_i$ times $\mathbb{P}_{\widetilde{\mathbf{x}}^{(b'_i,0)}} \to \mathbb{P}_{\widetilde{\mathbf{x}}^{(b'_i,\hat{b}_i-1)}}$ where $\mathbb{P}_{\widetilde{\mathbf{x}}^{(b'_i,\hat{b}_i-1)}}$ represents the corresponding probabilistic representation. We derive a lower bound for $\mathbf{V}$ at the t-th task learning :

$$
\begin{aligned}
&\sum_{j=1}^{|B|} \Big\{ \mathbb{E}_{\mathbb{P}^T_{b_j}} \big[\log p_{\theta_{c_j}}(\mathbf{x}) \big] \Big\} + \sum_{j=1}^{|B'|} \Big\{ \mathbb{E}_{\mathbb{P}^T_{b'_j}} \big[\log p_{\theta_{c'_j}}(\mathbf{x}) \big] \Big\} \geq \\
&\sum_{j=1}^{|B|} \Big\{ \mathbb{E}_{\mathbb{P}^S_{b_j}} \big[\mathcal{L}_{ELBO}(\mathbf{x}; \theta_{c_j}, \omega_{c_j}) \big] - D_{JS}\big(\mathbb{P}^T_{b_j} \,||\, \mathbb{P}^S_{c_j} \big) \\
&- \mathcal{F}_{\mathrm{DL}}\big(\mathbb{P}^S_{b_j}, \mathbb{P}^T_{b_j}, \mathbb{P}_{\theta_{c_j}} \big) + \mathcal{F}_{\mathrm{dis}}\big(\mathbb{P}^T_{b_j}, \mathbb{P}^S_{b_j} \big) \Big\} \\
&+ \sum_{j=1}^{|B'|} \Big\{ \mathbb{E}_{\mathbb{P}_{\widehat{\mathbf{x}}^{(j,\hat{b}_j)}} \otimes \mathbb{P}^S_{c'_j}} \big[\mathcal{L}_{ELBO}(\mathbf{x}; \theta_{c'_j}, \omega_{c'_j}) \big] - D_{JS}\big(\mathbb{P}^T_{b'_j} \,||\, \mathbb{P}_{\widehat{\mathbf{x}}^{(j,0)}} \otimes \mathbb{P}^S_{c'_j} \big) \\
&- \mathcal{F}_{\mathrm{DL}}\big(\mathbb{P}_{\widehat{\mathbf{x}}^{(j,0)}} \otimes \mathbb{P}^S_{c'_j}, \mathbb{P}^T_{b'_j}, \mathbb{P}_{\theta^{c'_j}} \big) + \mathcal{F}_{\mathrm{dis}}\big(\mathbb{P}^T_{b'_j}, \mathbb{P}_{\widehat{\mathbf{x}}^{(j,0)}} \otimes \mathbb{P}^S_{b'_j} \big) \\
&- \sum_{c=2}^{\hat{b}_j} \Big\{ D_{JS}\big(\mathbb{P}_{\widehat{\mathbf{x}}^{(j,c-1)}} \,||\, \mathbb{P}_{\widehat{\mathbf{x}}^{(j,c)}} \otimes \mathbb{P}^S_{b'_j} \big) \\
&+ \mathcal{F}_{\mathrm{DL}}\big(\mathbb{P}_{\widehat{\mathbf{x}}^{(j,c)}} \otimes \mathbb{P}^S_{b'_j}, \mathbb{P}_{\widehat{\mathbf{x}}^{(j,c-1)}}, \mathbb{P}_{\theta_{c'_j}} \big) - \mathcal{F}_{\mathrm{dis}}\big(\mathbb{P}_{\widehat{\mathbf{x}}^{(j,c-1)}}, \mathbb{P}_{\widehat{\mathbf{x}}^{(j,c)}} \otimes \mathbb{P}^S_{b'_j} \big) \Big\} \Big\}, \qquad (31)
\end{aligned}
$$

Remark From Eq. (31), we have several observations :

- Increasing the number of components in $\mathbf{V}$ can improve the performance. Specifically, when the number of components matches the number of tasks, Eq. (31) is transformed to Eq. (30) which does not have forgetting.
- Reducing the number of components would lead to degenerated performance because the accumulated errors is increased in RHS of Eq. (31). $|B'| = 1$ indicates that $\mathbf{V}$ only learns a single component and would suffer from more forgetting due to the more accumulated errors.

Proof. Firstly, let we consider the tasks that are only trained once. We have :

$$
\begin{aligned}
\sum_{j=1}^{|B|} \Big\{ \mathbb{E}_{\mathbb{P}^T_{b_j}} \big[\log p_{\theta_{c_j}}(\mathbf{x}) \big] \Big\} \geq \sum_{j=1}^{|B|} \Big\{ & \mathbb{E}_{\mathbb{P}^S_{b_j}} \big[\mathcal{L}_{ELBO}(\mathbf{x}; \theta_{c_j}, \omega_{c_j}) \big] - D_{JS}\big(\mathbb{P}^T_{b_j} \,||\, \mathbb{P}^S_{c_j} \big) \\
& - \mathcal{F}_{\mathrm{DL}}\big(\mathbb{P}^S_{b_j}, \mathbb{P}^T_{b_j}, \mathbb{P}_{\theta_{c_j}} \big) + \mathcal{F}_{\mathrm{dis}}\big(\mathbb{P}^T_{b_j}, \mathbb{P}^S_{b_j} \big) \Big\}, \qquad (32)
\end{aligned}
$$

Secondly, we consider the tasks that are trained more than once. We have :

$$
\begin{aligned}
&\sum_{j=1}^{|B'|}\Big\{\mathbb{E}_{\mathbb{P}^T_{b'_j}}\big[\log p_{\theta_{c'_j}}(\mathbf{x})\big]\Big\} \geq \sum_{j=1}^{|B'|}\Big\{\mathbb{E}_{\mathbb{P}_{\widehat{\mathbf{x}}^{(j,\hat{b}_j)}} \otimes \mathbb{P}^S_{c'_j}}\big[\mathcal{L}_{ELBO}(\mathbf{x};\theta_{c'_j},\omega_{c'_j})\big] \\
&- D_{JS}\big(\mathbb{P}^T_{b'_j} \,||\, \mathbb{P}_{\widehat{\mathbf{x}}^{(j,0)}} \otimes \mathbb{P}^S_{c'_j}\big) - \mathcal{F}_{\mathrm{DL}}\big(\mathbb{P}_{\widehat{\mathbf{x}}^{(j,0)}} \otimes \mathbb{P}^S_{c'_j}, \mathbb{P}^T_{b'_j}, \mathbb{P}_{\theta^{c'_j}}\big) \\
&+ \mathcal{F}_{\mathrm{dis}}\big(\mathbb{P}^T_{b'_j}, \mathbb{P}_{\widehat{\mathbf{x}}^{(j,0)}} \otimes \mathbb{P}^S_{b'_j}\big) - \sum_{c=2}^{\hat{b}_j}\Big\{D_{JS}\big(\mathbb{P}_{\widehat{\mathbf{x}}^{(j,c-1)}} \,||\, \mathbb{P}_{\widehat{\mathbf{x}}^{(j,c)}} \otimes \mathbb{P}^S_{b'_j}\big) \\
&+ \mathcal{F}_{\mathrm{DL}}\big(\mathbb{P}_{\widehat{\mathbf{x}}^{(j,c)}} \otimes \mathbb{P}^S_{b'_j}, \mathbb{P}_{\widehat{\mathbf{x}}^{(j,c-1)}}, \mathbb{P}_{\theta_{c'_j}}\big) - \mathcal{F}_{\mathrm{dis}}\big(\mathbb{P}_{\widehat{\mathbf{x}}^{(j,c-1)}}, \mathbb{P}_{\widehat{\mathbf{x}}^{(j,c)}} \otimes \mathbb{P}^S_{b'_j}\big)\Big\}\Big\}, \qquad (33)
\end{aligned}
$$

Then we sum up Eq. (32) and Eq. (33), resulting in :

$$
\begin{aligned}
&\sum_{j=1}^{|B|}\Big\{\mathbb{E}_{\mathbb{P}^T_{b_j}}\big[\log p_{\theta_{c_j}}(\mathbf{x})\big]\Big\} + \sum_{j=1}^{|B'|}\Big\{\mathbb{E}_{\mathbb{P}^T_{b'_j}}\big[\log p_{\theta_{c'_j}}(\mathbf{x})\big]\Big\} \geq \\
&\sum_{j=1}^{|B|}\Big\{\mathbb{E}_{\mathbb{P}^S_{b_j}}\big[\mathcal{L}_{ELBO}(\mathbf{x};\theta_{c_j},\omega_{c_j})\big] - D_{JS}\big(\mathbb{P}^T_{b_j} \,||\, \mathbb{P}^S_{c_j}\big) \\
&- \mathcal{F}_{\mathrm{DL}}\big(\mathbb{P}^S_{b_j}, \mathbb{P}^T_{b_j}, \mathbb{P}_{\theta_{c_j}}\big) + \mathcal{F}_{\mathrm{dis}}\big(\mathbb{P}^T_{b_j}, \mathbb{P}^S_{b_j}\big)\Big\} \\
&+ \sum_{j=1}^{|B'|}\Big\{\mathbb{E}_{\mathbb{P}_{\widehat{\mathbf{x}}^{(j,\hat{b}_j)}} \otimes \mathbb{P}^S_{c'_j}}\big[\mathcal{L}_{ELBO}(\mathbf{x};\theta_{c'_j},\omega_{c'_j})\big] - D_{JS}\big(\mathbb{P}^T_{b'_j} \,||\, \mathbb{P}_{\widehat{\mathbf{x}}^{(j,0)}} \otimes \mathbb{P}^S_{c'_j}\big) \\
&- \mathcal{F}_{\mathrm{DL}}\big(\mathbb{P}_{\widehat{\mathbf{x}}^{(j,0)}} \otimes \mathbb{P}^S_{c'_j}, \mathbb{P}^T_{b'_j}, \mathbb{P}_{\theta^{c'_j}}\big) + \mathcal{F}_{\mathrm{dis}}\big(\mathbb{P}^T_{b'_j}, \mathbb{P}_{\widehat{\mathbf{x}}^{(j,0)}} \otimes \mathbb{P}^S_{b'_j}\big) \\
&- \sum_{c=2}^{\hat{b}_j}\Big\{D_{JS}\big(\mathbb{P}_{\widehat{\mathbf{x}}^{(j,c-1)}} \,||\, \mathbb{P}_{\widehat{\mathbf{x}}^{(j,c)}} \otimes \mathbb{P}^S_{b'_j}\big) \\
&+ \mathcal{F}_{\mathrm{DL}}\big(\mathbb{P}_{\widehat{\mathbf{x}}^{(j,c)}} \otimes \mathbb{P}^S_{b'_j}, \mathbb{P}_{\widehat{\mathbf{x}}^{(j,c-1)}}, \mathbb{P}_{\theta_{c'_j}}\big) - \mathcal{F}_{\mathrm{dis}}\big(\mathbb{P}_{\widehat{\mathbf{x}}^{(j,c-1)}}, \mathbb{P}_{\widehat{\mathbf{x}}^{(j,c)}} \otimes \mathbb{P}^S_{b'_j}\big)\Big\}\Big\}, \qquad (34)
\end{aligned}
$$

$\square$

G.3 Theoretical analysis for the existing GRM-based models

In this section, we employ the proposed theoretical framework to analyze the forgetting behaviour of the existing GRM-based models in a general continual learning setting where task label is given during the training.

G.3.1 Lifelong VAEGAN

Lifelong VAENGAN Ye and Bors (2020a) is a popular GRM-based model. Unlike other GRM-based approaches that lack an inference mechanism, Lifelong VAEGAN introduces to combine the powerful inference mechanism of VAE and the robust generation capacity of GAN into a unified framework. Lifelong VAEGAN produces not only high-quality generative replay samples but also captures meaningful latent representations across domains over time. In this section, we extend our theoretical analysis to Lifelong VAEGAN. According to **Lemma 7**, we can derive a lower bound for Lifelong VAEGAN as :

$$
\begin{aligned}
&\mathbb{E}_{\mathbb{P}^T_j}\big[\log p_{\theta^i}(\mathbf{x})\big] \geq \mathbb{E}_{\mathbb{P}_{\widehat{\mathbf{x}}(j,i-j)}\otimes\mathbb{P}^S_i}\big[\mathcal{L}_{ELBO}(\mathbf{x};\theta^i,\omega^i)\big] - D_{JS}\big(\mathbb{P}^T_j \,||\, \mathbb{P}_{\widehat{\mathbf{x}}(j,0)} \otimes \mathbb{P}^S_i\big) \\
&- \mathcal{F}_{\mathrm{DL}}\big(\mathbb{P}_{\widehat{\mathbf{x}}(j,0)} \otimes \mathbb{P}^S_i, \mathbb{P}^T_j, \mathbb{P}_{\theta^i}\big) + \mathcal{F}_{\mathrm{dis}}\big(\mathbb{P}^T_j, \mathbb{P}_{\widehat{\mathbf{x}}(j,0)} \otimes \mathbb{P}^S_i\big) \\
&- \sum_{c=2}^{i-j}\Big\{D_{JS}\big(\mathbb{P}_{\widehat{\mathbf{x}}(j,c-1)} \,||\, \mathbb{P}_{\widehat{\mathbf{x}}(j,c)} \otimes \mathbb{P}^S_i\big) \\
&+ \mathcal{F}_{\mathrm{DL}}\big(\mathbb{P}_{\widehat{\mathbf{x}}(j,c)} \otimes \mathbb{P}^S_i, \mathbb{P}_{\widehat{\mathbf{x}}(j,c-1)}, \mathbb{P}_{\theta^i}\big) - \mathcal{F}_{\mathrm{dis}}\big(\mathbb{P}_{\widehat{\mathbf{x}}(j,c-1)}, \mathbb{P}_{\widehat{\mathbf{x}}(j,c)} \otimes \mathbb{P}^S_i\big)\Big\}\,,
\end{aligned}
\tag{35}
$$

Eq. (35) describes the forgetting behaviour of Lifelong VAEGAN on the j-th task when it is trained on the i-th task where $i > j$. Since Lifelong VAEGAN employs GAN as the generative replay network, it can provide high-quality generative replay samples that approximate the target distribution exactly. Furthermore, compared with the VAE-based models, Lifelong VAEGAN can have the small JS divergence terms in RHS of Eq. (35) and thus can achieve better performance.

G.3.2 Lifelong infinite mixture model

Lifelong Infinite Mixture (LIMix) Ye and Bors (2021a) is a popular dynamic expansion model which can dynamically change its network architectures to deal with new tasks. LIMix also introduces to evaluate the knowledge similarity between each trained component and a new task, which aims to reuse an appropriate component to learn several similar tasks. According to **Lemma 10**, the forgetting behaviour of LMix can be described by :

$$\begin{aligned}
&\sum_{j=1}^{|B|}\Big\{\mathbb{E}_{\mathbb{P}^T_{b_j}}\big[\log p_{\theta_{c_j}}(\mathbf{x})\big]\Big\}+\sum_{j=1}^{|B'|}\Big\{\mathbb{E}_{\mathbb{P}^T_{b'_j}}\big[\log p_{\theta_{c'_j}}(\mathbf{x})\big]\Big\}\geq\\
&\sum_{j=1}^{|B|}\Big\{\mathbb{E}_{\mathbb{P}^S_{b_j}}\big[\mathcal{L}_{ELBO}(\mathbf{x};\theta_{c_j},\omega_{c_j})\big]-D_{JS}\big(\mathbb{P}^T_{b_j}\,||\,\mathbb{P}^S_{c_j}\big)\\
&-\mathcal{F}_{\mathrm{DL}}\big(\mathbb{P}^S_{b_j},\mathbb{P}^T_{b_j},\mathbb{P}_{\theta_{c_j}}\big)+\mathcal{F}_{\mathrm{dis}}\big(\mathbb{P}^T_{b_j},\mathbb{P}^S_{b_j}\big)\Big\}\\
&+\sum_{j=1}^{|B'|}\Big\{\mathbb{E}_{\mathbb{P}_{\widehat{\mathbf{x}}^{(j,\hat{b}_j)}}\otimes\mathbb{P}^S_{c'_j}}\big[\mathcal{L}_{ELBO}(\mathbf{x};\theta_{c'_j},\omega_{c'_j})\big]-D_{JS}\big(\mathbb{P}^T_{b'_j}\,||\,\mathbb{P}_{\widehat{\mathbf{x}}^{(j,0)}}\otimes\mathbb{P}^S_{c'_j}\big)\\
&-\mathcal{F}_{\mathrm{DL}}\big(\mathbb{P}_{\widehat{\mathbf{x}}^{(j,0)}}\otimes\mathbb{P}^S_{c'_j},\mathbb{P}^T_{b'_j},\mathbb{P}_{\theta^{c'_j}}\big)+\mathcal{F}_{\mathrm{dis}}\big(\mathbb{P}^T_{b'_j},\mathbb{P}_{\widehat{\mathbf{x}}^{(j,0)}}\otimes\mathbb{P}^S_{b'_j}\big)\\
&-\sum_{c=2}^{\hat{b}_j}\Big\{D_{JS}\big(\mathbb{P}_{\widehat{\mathbf{x}}^{(j,c-1)}}\,||\,\mathbb{P}_{\widehat{\mathbf{x}}^{(j,c)}}\otimes\mathbb{P}^S_{b'_j}\big)\\
&+\mathcal{F}_{\mathrm{DL}}\big(\mathbb{P}_{\widehat{\mathbf{x}}^{(j,c)}}\otimes\mathbb{P}^S_{b'_j},\mathbb{P}_{\widehat{\mathbf{x}}^{(j,c-1)}},\mathbb{P}_{\theta_{c'_j}}\big)-\mathcal{F}_{\mathrm{dis}}\big(\mathbb{P}_{\widehat{\mathbf{x}}^{(j,c-1)}},\mathbb{P}_{\widehat{\mathbf{x}}^{(j,c)}}\otimes\mathbb{P}^S_{b'_j}\big)\Big\}\Big\}, \qquad (36)
\end{aligned}$$

From Eq. (36), it observes that increasing the number of components in LIMix can significantly improve the generalization performance since the JS divergence terms in the RHS of Eq. (36) are reduced. In addition, if a particular component is reused to learn several similar tasks, the JS divergence terms caused by the GRM process would be small, resulting in better relieving forgetting and reducing the total number of components. However, LIMix can not be applied in TFCL since it requires access to the task labels in order to perform the component selection and expansion strategy.

H Additional information for experiments

H.1 Additional information for experiment settings

The release of the code. We have provided the detailed implementation of the proposed Online Adversarial Expansion Strategy (OAES) model. We also provide the source code in the supplemental material. In addition, We will organize the source code of the OAES model for the sake of easy understanding and for facilitating the re-implementation and we will release it publicly on https://github.com/ if the paper is accepted.

H.2 Experiment setting

The hyperparameter configuration and GPU hardware. To perform the density estimation task, we use Adam Kingma and Ba (2015) with a learning rate of 0.0001 and its default hyperparameters. We set the batch size and the number of epochs for each training step as 64 and 1, respectively. Following from Ye and Bors (2022a,b,c, 2021b,c); Aljundi et al. (2019a); Ye and Bors (2022d, 2021d, 2020a,b, 2022e, 2021e,a, 2020c), the experiments are constructed in the computer with the operating system (Ubuntu 18.04.5). We also use the GPU (GeForce GTX 1080) for all our experiments.

The configuration of the network architecture for log-likelihood estimation task. We adapt the network architecture from Burda et al. (2015) where two fully connected layers implement the inference and generator models. Each layer has 200 hidden units. The shared modules use the expansion mechanism as a single fully-connected neural network with a layer (200 hidden units). A single layer also implements each individual component with 200 hidden units for both the generator and inference models.

Hyperparameter setting. The batch size is of 64 images, and we consider 1 epochs for each training stage. The maximum memory size for Split MNIST, Split Fashion, Split MNIST-Fashion, Cross-domain is 1.5K, 1.5K, 1.9K and 2.0K, respectively.

Additional information for the evaluation. All results reported in the paper are evaluated on the testing datasets after task-free continual learning.

H.3 The configuration for the classification task.

First, we introduce the details about the datasets used in our classification task as follows. The threshold β for Split MNIST, Split CIFAR10, Split CIFAR100 and Split MiniImageNe is 4.2, 4 and 4.5 and 4.8 respectively. The number of components of OAES for Split MNIST, Split CIFAR10, Split CIFAR100 and Split MiniImageNe is 6, 6, 7 and 6, respectively. In the following, we describe the detailed information for each dataset that is used in our classification tasks.

Split MNIST. We divide MNIST which contains 60k training samples into five tasks, each consisting of images from two classes, in consecutive order of their displayed digits, while increasing the numbers represented in the images De Lange and Tuytelaars (2021).

Split CIFAR10. We split CIFAR10 into five tasks where each task consists of samples from two different classes De Lange and Tuytelaars (2021).

Split CIFAR100. We split CIFAR100 into 20 tasks where each task has 2500 examples from five different classes Lopez-Paz and Ranzato (2017).

Split MiniImageNet. We divide the MiniImageNet into 20 tasks Vinyals et al. (2016), where each task collects the images of five classes Aljundi et al. (2019b).

In the following, we describe the detailed information of the network architecture used in our classification task.

We adapt ResNet 18 He et al. (2016) for Split CIFAR10 and Split CIFAR100. We use an MLP network with 2 hidden layers of 400 units each De Lange and Tuytelaars (2021) for Split MNIST. The maximimum memory size for Split MNIST, Split CIFAR10, Split CIFAR100 are 2000, 1000 and 5000, respectively. To enable the proposed OAES for the classification, we train an individual classifier for each component, similar to Ye and Bors (2022a). At the testing phase, we make the component selection by comparing the sample log-likelihood and the classifier of the selected component is used for prediction.

We introduce the baselines used for the classification task but which are not mentioned in the paper.

Finetune trains a single model directly on a new batch of images during the online continual learning.

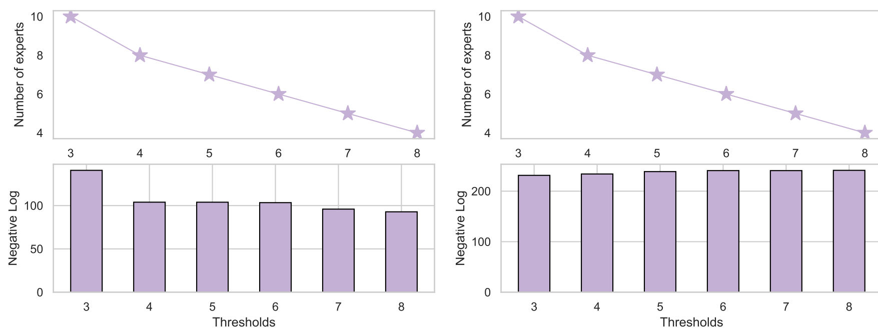
Gradient Episodic Memory (GEM) Lopez-Paz and Ranzato (2017) is a memory-based approach that would use the memory to store past samples. GEM is also required to access both the task label and class label during the training.

Dynamic-OCM Ye and Bors (2022a) is a dynamic expansion model which proposes an online cooperative memorization (OCM) approach. OCM manages two memory buffers, aiming to store short- and long-term knowledge during training. In addition, Dynamic-OCM detects the change of the loss value as expansion signals, which does not have theoretical guarantees.

Incremental Classifier and Representation Learning (iCARL) Rebuffi et al. (2017) is a standard memory-based method used in a class incremental setup.

reservoir* Vitter (1985) is a memory-based approach that stores the observed sample into a memory buffer $\mathcal{M}$ with probability $|\mathcal{M}|/n$ where n is the number of stored samples, and $|\cdot|$ represents the cardinality of a set.

MIR Aljundi et al. (2019b) introduces a retrieval strategy for the sample selection in the memory during the Online Continual Learning (OCL). However, the retrieval strategy in MIR requires evaluating the loss in each training session. This means that MIR requires modifying the retrieval strategy for different tasks such as classification or



(a) Split MNIST (b) Split Fashion

Figure 2: The performance and the number of components of OAES when changing the threshold β.

generation tasks. The proposed OCM does not change the sample selection strategy for different tasks since we evaluate the sample similarity in the given feature space using the kernel function from Eq. (16) from the paper.

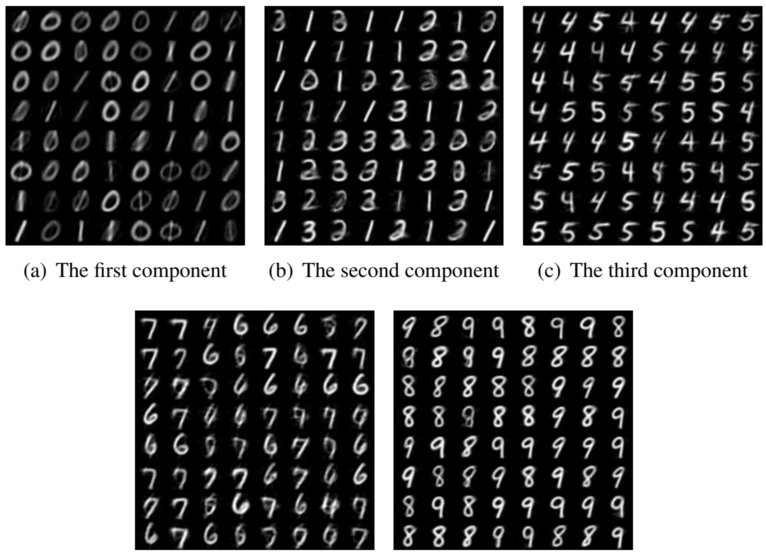
GSS Aljundi et al. (2019c) formulates the sample selection process as a constraint reduction problem. GSS stores samples in a buffer based on the gradient information which requires to access the class labels and can not be applied in the unsupervised learning setting.

I Additional information for ablation study

In this section, we provide more ablation studies in order to investigate the effectiveness of each modules in the proposed OAES.

I.1 The impact of the threshold β

We investigate the effect of β by training OAES on Split MNIST with different threshold values and the results are reported in Fig. 2a. It observes that a small β leads to learning more components while improving the performance. In contrast, a large β allows OAES to employ fewer components. We train OAEs on Split Fashion with different threshold values and the results are reported in Fig. 2a. We observe that increasing the number of components can improve the performance of Split Fashion.



(a) The first component (b) The second component (c) The third component

(d) The fourth component (e) The fifth component

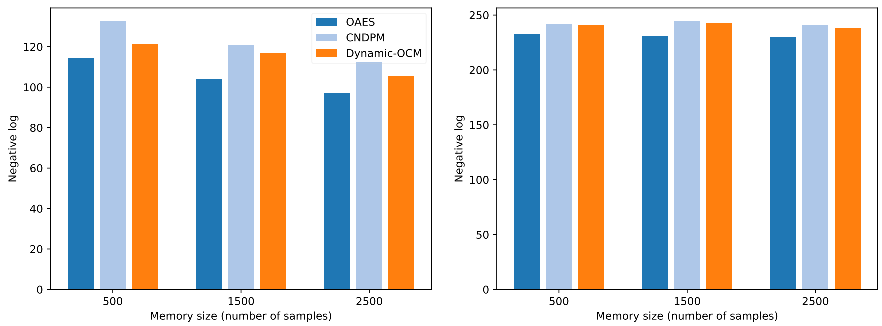
Figure 3: The generation results from each learned VAE component under Split MNIST.

I.2 The impcat of the memory buffer size

We train various models under Split MNIST by using different memory buffer sizes and the results are reported in Fig. 4a. These results show that a large-scale memory buffer can improve the performance for all DEM models. We also train various models on Split Fashion with different memory configurations and the results are reported in Fig. 4b. The proposed OAES outperforms other baselines, especially when the memory buffer size is very small (500).

I.3 Knowledge diversity among experts

We train OAES with $\beta = 1.7$ on Split MNIST for the classification task where we evaluate the discrepancy score (the left-hand-side (LHS) of Eq.(15) of the paper) at each training step. We plot the results in Fig. 5, which shows that there are five peak-to-valley discrepancy scores, corresponding to the five different underlying distributions (tasks). We also show the generation results from each learned component of the proposed model in Fig. 3. It observes that the first component learns the same underlying data distribution while other components modelling a unique underlying data distri-



(a) Split MNIST (b) Split Fashion

Figure 4: The performance of various models when changing the memory buffer size.

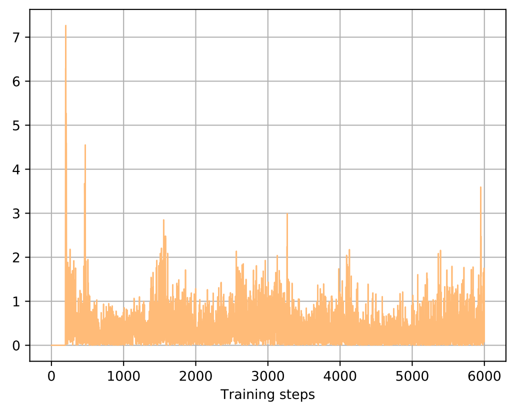


Figure 5: The expansion evaluation (the left-hand-side (LHS) of Eq.(15) of the paper) evaluated by OAES at each training step.

bution. These results show that the proposed OAES can accurately detect the data distribution shift and provide appropriate signals for the model expansion.

I.4 The impact of batch size change

In this section, we investigate the impact of the batch size change of the proposed OAES. We train the OAES under Split MNIST using the different batch sizes, and the empirical results are reported in Fig. 6. It observes that changing the batch size has little effect on the performance and the model size. This shows that the proposed OAES is robust to the change in the batch size.

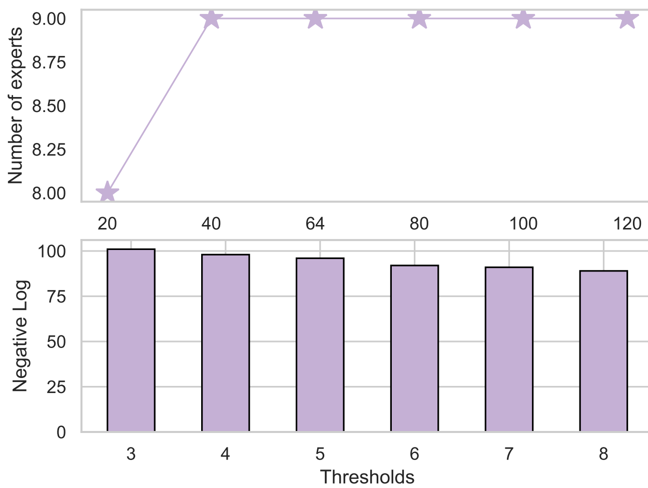


Figure 6: The performance and the number of experts of the proposed OAES under Split MNIST when changing the batch size.

I.5 Theoretical results

We train the proposed OAES and the static model (ER) under Split MNIST in which we evaluate the source risk (the LHS of Eq. (2)) and the target risk (the first term in RHS of Eq. (2)). We plot the results evaluated by each training step in Fig. 7, where 'OAES-source' and 'OAES-target' denote the source and target risk evaluated by the OAES. It observes that ER firstly gets good performance on the initial training phase and gradually losses performance as the number of training steps increases. This is because the memory-based approaches can not store all previously learnt information during the training. In contrast, the proposed OAES gradually improves the performance of the target samples when performing more training steps. These results demonstrate that the proposed OAES can accumulate knowledge without forgetting it and can achieve better generalization performance than the static model.

I.6 Classification on fuzzy task boundaries

In this section, we apply our model in a real-world and more challenging TFCL setting, called fuzzy task boundaries Lee et al. (2020). In this setting, we randomly swap samples between two tasks for each data stream. The results for the fuzzy task boundaries are reported in Tab. 1. These results show that the proposed OAES still outperforms other baselines in this challenging setting.

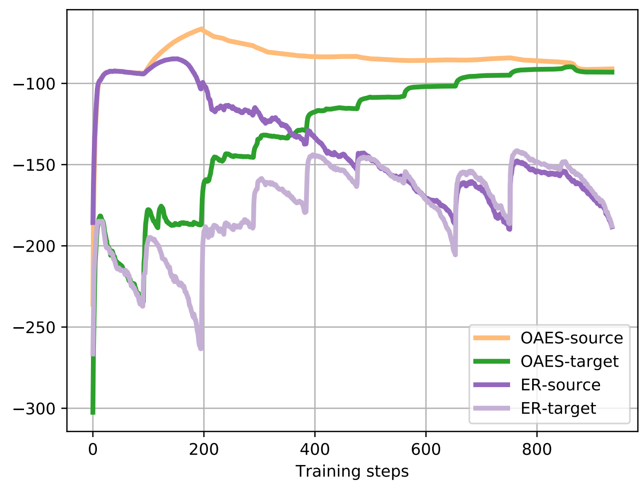


Figure 7: The source and target risks (sample log-likelihood) of the OAES and the static model under Split MNIST.

Methods	Split MNIST	Split CIFAR10	Split MImageNet
Vanilla	21.53 $\pm$ 0.1	20.69 $\pm$ 2.4	3.05 $\pm$ 0.6
ER	79.74 $\pm$ 4.0	37.15 $\pm$ 1.6	26.47 $\pm$ 2.3
MIR	84.80 $\pm$ 1.9	38.70 $\pm$ 1.7	25.83 $\pm$ 1.5
ER + GMED	82.73 $\pm$ 2.6	40.57 $\pm$ 1.7	28.20 $\pm$ 0.6
MIR+GMED	86.17 $\pm$ 1.7	41.22 $\pm$ 1.1	26.86 $\pm$ 0.7
OAES	90.23 $\pm$1.2	44.26 $\pm$ 1.1	29.63 $\pm$ 0.8

Table 1: The classification accuracy of five indepdnent runs for various models over data streams with fuzzy task boundaries.

I.7 Comparison of computational costs

In this section, we compare our model (OAES) with Dynamic-OCM in terms of computational costs. The training times (minutes) of various models under the density estimation task are reported in Tab. 2. It observes that the proposed OAES requires less computational costs than Dynamic-OCM while achieving better performance than Dynamic-OCM.

I.8 The other performance criterion on the generative modelling

In this section, we employ the other performance criteria including the Inception Score (IS) and Fréchet Inception Distance (FID) for the generative modelling task. We train our model (OAES) under Split CIFAR10 and report the results in Tab. 3 where the

Methods	**Split MNIST**	**Split Fashion**	**Split MNIST-Fashion**	**Cross domain**
OAES	30.2	35.6	71.2	105.4
Dynamic-OCM	46.6	52.6	82.7	120.6

Table 2: The training times (minutes) for various models under the continual generative modelling task.

Methods	*IS*	*FID*	Memory	N
VAE-ELBO-Random	3.84	116.26	1.0K	1
CNDPM Lee et al. (2020)	4.12	95.23	1.0K	30
LIMix Ye and Bors (2021a)	3.02	156.46	1.0K	30
VAE-ELBO-OCM	4.13	98.76	1.0K	1
Dynamic-ELBO-OCM	4.16	92.99	0.9K	3
OAES	**4.25**	**88.62**	0.9K	3

Table 3: IS and FID scores under Split CIFAR10.

results of other baselines are taken from Ye and Bors (2022a). These results show that the proposed OAES outperforms other baselines in terms of IS and FID.

References

Yuri Burda, Roger Grosse, and Ruslan Salakhutdinov. Importance weighted autoencoders. In *Proc. Int. Cont. of Learning Representations (ICLR), arXiv preprint arXiv:1509.00519*, 2015. 9, 21

Lars Maaløe, Casper Kaae Sønderby, Søren Kaae Sønderby, and Ole Winther. Auxiliary deep generative models. In *Proc. Int. Conf. on Machine Learning (ICML), vol. PMLR 48*, pages 1445–1453, 2016. 10

Fei Ye and Adrian G. Bors. Continual variational autoencoder learning via online cooperative memorization, 2022a. 11, 21, 22, 28

Evgenii Egorov, Anna Kuzina, and Evgeny Burnaev. Boovae: Boosting approach for continual learning of vae. *Advances in Neural Information Processing Systems*, 34: 17889–17901, 2021. 11

Kamil Deja, Paweł Wawrzyński, Daniel Marczak, Wojciech Masarczyk, and Tomasz

Trzciński. Multiband vae: Latent space partitioning for knowledge consolidation in continual learning. *arXiv preprint arXiv:2106.12196*, 2021. 11

Fei Ye and Adrian G.Bors Bors. Learning latent representations across multiple data domains using lifelong vaegan. In *Proc. of European Conference on Computer Vision (ECCV), vol. LNCS 12365*, pages 777–795, 2020a. 19, 21

Fei Ye and Adrian G. Bors. Lifelong infinite mixture model based on knowledge-driven dirichlet process. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 10695–10704, October 2021a. 19, 21, 28

D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. In *Proc. Int. Conf. on Learning Representations (ICLR), arXiv preprint arXiv:1412.6980*, 2015. 21

Fei Ye and Adrian G Bors. Task-free continual learning via online discrepancy distance learning. *arXiv preprint arXiv:2210.06579*, 2022b. 21

Fei Ye and Adrian G Bors. Dynamic self-supervised teacher-student network learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2022c. 21

Fei Ye and Adrian G. Bors. Lifelong twin generative adversarial networks. In *Proc. IEEE Int. Conf. on Image Processing (ICIP)*, pages 1289–1293, 2021b. doi: 10.1109/ICIP42928.2021.9506116. 21

Fei Ye and Adrian G. Bors. Lifelong mixture of variational autoencoders. *IEEE Transactions on Neural Networks and Learning Systems*, pages 1–14, 2021c. doi: 10.1109/TNNLS.2021.3096457. 21

Rahaf Aljundi, Klaas Kelchtermans, and Tinne Tuytelaars. Task-free continual learning. In *Proc. of IEEE/CVF Conf. on Computer Vision and Pattern Recognition*, pages 11254–11263, 2019a. 21

Fei Ye and Adrian G Bors. Learning an evolved mixture model for task-free continual learning. In *2022 IEEE International Conference on Image Processing (ICIP)*, pages 1936–1940. IEEE, 2022d. 21

Fei Ye and Adrian Bors. Lifelong teacher-student network learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021d. 21

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, 2020b. doi: 10.1109/IPTA50016.2020.9286663. 21

Fei Ye and Adrian G. Bors. Lifelong generative modelling using dynamic expansion graph model. In *Thirty-Sixth AAAI Conference on Artificial Intelligence, AAAI*, pages 8857–8865. AAAI Press, 2022e. URL https://ojs.aaai.org/index.php/AAAI/article/view/20867. 21

Fei Ye and Adrian G. Bors. Deep mixture generative autoencoders. *IEEE Transactions on Neural Networks and Learning Systems*, pages 1–15, 2021e. doi: 10.1109/TNNLS.2021.3071401. 21

Fei Ye and Adrian G Bors. Mixtures of variational autoencoders. In *Proc. Int. Conf. on Image Processing Theory, Tools and Applications (IPTA)*, pages 1–6, 2020c. 21

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*, pages 8250–8259, 2021. 21, 22

David Lopez-Paz and Marc'Aurelio Ranzato. Gradient episodic memory for continual learning. In *Advances in Neural Information Processing Systems*, pages 6467–6476, 2017. 22

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. 22

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, 2019b. 22

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 Recognition (CVPR)*, pages 770–778, 2016. 22

Sylvestre-Alvise Rebuffi, Alexander Kolesnikov, Georg Sperl, and Christoph H Lampert. icarl: Incremental classifier and representation learning. In *Proceedings of the*

IEEE conference on Computer Vision and Pattern Recognition, pages 2001–2010, 2017. 22

Jeffrey S Vitter. Random sampling with a reservoir. *ACM Transactions on Mathematical Software (TOMS)*, 11(1):37–57, 1985. 22

Rahaf Aljundi, Min Lin, Baptiste Goujaud, and Yoshua Bengio. Gradient based sample selection for online continual learning. In *Proc. Neural Inf. Proc. Systems (NeurIPS)*, pages 11817–11826, 2019c. 23

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. 26, 28