Appendix for Continual Variational Autoencoder Learning via Online Cooperative Memorization

July 20, 2022

Contents

A The proof of Theorem 1

Theorem 1 *Let $p_\theta(\mathbf{x}\,|\,\mathbf{z}) = \mathcal{N}(\mathrm{G}_i(\mathbf{z}), \sigma^2\mathbf{I}_d)$ be the Gaussian decoder where $\sigma = 1/\sqrt{2}$. We can derive ELBO based on the optimal transport :*

$$\inf_{q_\omega(\mathbf{z})=p(\mathbf{z})} \mathbb{E}_{\mathbb{P}_\mathbf{x}}[\mathcal{L}_{ELBO}(\mathbf{x};\theta,\omega)] \leq -\mathrm{W}^\star_\mathcal{L}(\mathbb{P}_\mathbf{x}, \mathbb{P}_{\mathbf{G}_i}) - \frac{1}{2}\log\pi\,, \tag{1}$$

Proof. When $p_\theta(\mathbf{x}\,|\,\mathbf{z})$ is the Gaussian decoder, the computation of $\log p_\theta(\mathbf{x}\,|\,\mathbf{z})$ involves the noise value σ :

$$\log p_\theta\left(\mathbf{x}\,|\,\mathbf{z}\right) = -\frac{1}{2\sigma^2}\|\mathbf{x} - \mu_\theta\left(\mathbf{z}\right)\|_2^2 - \frac{1}{2}\log 2\pi\sigma^2\,, \tag{2}$$

where $\mu_\theta(\mathbf{z})$ is the mean of distribution $p_\theta(\mathbf{x}\,|\,\mathbf{z})$. In order to simplify Eq. (2), the noise σ is set to $1/\sqrt{2}$, resulting in :

$$\log p_\theta\left(\mathbf{x}\,|\,\mathbf{z}\right) = -\|\mathbf{x} - \mu_\theta\left(\mathbf{z}\right)\|_2^2 - \frac{1}{2}\log\pi\,. \tag{3}$$

We substract the KL divergence term in Eq. (3), resulting in :

$$\log p_\theta\left(\mathbf{x}\,|\,\mathbf{z}\right) - D_{KL}(q_\omega(\mathbf{x}\,|\,\mathbf{z})\,|\,p(\mathbf{z})) = -\|\mathbf{x} - \mu_\theta\left(\mathbf{z}\right)\|_2^2 - D_{KL}(q_\omega(\mathbf{x}\,|\,\mathbf{z})\,|\,p(\mathbf{z})) - \frac{1}{2}\log\pi\,. \tag{4}$$

Then we consider the expectation in both sides of Eq. (4), resulting in :

$$\begin{aligned}
&\inf_{q_\omega(\mathbf{z})=p(\mathbf{z})} \mathbb{E}_{\mathbb{P}_\mathbf{x}}\mathbb{E}_{q_\omega(\mathbf{z}\,|\,\mathbf{x})}\big[\log p_\theta\left(\mathbf{x}\,|\,\mathbf{z}\right) - D_{KL}(q_\omega(\mathbf{x}\,|\,\mathbf{z})\,|\,p(\mathbf{z}))\big] = \\
&\inf_{q_\omega(\mathbf{z})=p(\mathbf{z})} \mathbb{E}_{\mathbb{P}_\mathbf{x}}\mathbb{E}_{q_\omega(\mathbf{z}\,|\,\mathbf{x})}\big[- \|\mathbf{x} - \mu_\theta\left(\mathbf{z}\right)\|_2^2 \\
&- D_{KL}(q_\omega(\mathbf{x}\,|\,\mathbf{z})\,|\,p(\mathbf{z})) - \frac{1}{2}\log\pi\big]\,.
\end{aligned} \tag{5}$$

where the first term in the right-hand side of Eq. (5) can be rewritten as $\mathcal{L}(\mathbf{x}, \mathrm{G}_i(\mathbf{z}))$, then Eq. (5) can be rewritten as :

$$\begin{aligned}
&\inf_{q_\omega(\mathbf{z})=p(\mathbf{z})} \mathbb{E}_{\mathbb{P}_\mathbf{x}}\mathbb{E}_{q_\omega(\mathbf{z}\,|\,\mathbf{x})}\big[\log p_\theta\left(\mathbf{x}\,|\,\mathbf{z}\right) - D_{KL}(q_\omega(\mathbf{x}\,|\,\mathbf{z})\,|\,p(\mathbf{z}))\big] = \\
&\inf_{q_\omega(\mathbf{z})=p(\mathbf{z})} \mathbb{E}_{\mathbb{P}_\mathbf{x}}\mathbb{E}_{q_\omega(\mathbf{z}\,|\,\mathbf{x})}\big[- \mathcal{L}(\mathbf{x}, \mathrm{G}_i(\mathbf{z})) \\
&- D_{KL}(q_\omega(\mathbf{x}\,|\,\mathbf{z})\,|\,p(\mathbf{z})) - \frac{1}{2}\log\pi\big]\,.
\end{aligned} \tag{6}$$

where the first term in the left-hand side (LHS) of Eq. (6) is the ELBO, defined in Eq. (1) of the paper. Since the KL divergence $D_{KL}(\cdot)$ is equal or larger than 0, we have the following inequality :

$$\begin{aligned}
\inf_{q_\omega(\mathbf{z})=p(\mathbf{z})} \mathbb{E}_{\mathbb{P}_\mathbf{x}}[\mathcal{L}_{ELBO}(\mathbf{x};\theta,\omega)] &= \inf_{q_\omega(\mathbf{z})=p(\mathbf{z})} \mathbb{E}_{\mathbb{P}_\mathbf{x}}\mathbb{E}_{q_\omega(\mathbf{z}\,|\,\mathbf{x})}[-\mathcal{L}(\mathbf{x}, \mathrm{G}_i(\mathbf{z})) \\
&\quad - D_{KL}(q_\omega(\mathbf{z}\,|\,\mathbf{x})\,||\,p(\mathbf{z})) - \frac{1}{2}\log\pi] \\
&\leq \inf_{q_\omega(\mathbf{z})=p(\mathbf{z})} \mathbb{E}_{\mathbb{P}_\mathbf{x}}\mathbb{E}_{q_\omega(\mathbf{z}\,|\,\mathbf{x})}[-\mathcal{L}(\mathbf{x}, \mathrm{G}_i(\mathbf{z}))] - \frac{1}{2}\log\pi\,,
\end{aligned} \tag{7}$$

Eq. (7) holds because we have the inequality (Eq.(6) in the paper) :

$$-\mathrm{W}^{\star}_{\mathcal{L}}(\mathbb{P}_{\mathbf{x}}, \mathbb{P}_{\mathbf{G}_i}) \geq \inf_{q_\omega(\mathbf{z})=p(\mathbf{z})} \mathbb{E}_{\mathbb{P}_{\mathbf{x}}} \mathbb{E}_{q_\omega(\mathbf{z}\,|\,\mathbf{x})}[-\mathcal{L}(\mathbf{x}, \mathrm{G}_i(\mathbf{z}))]\,, \tag{8}$$

We can rewrite Eq. (7) by considering Eq. (8), resulting in :

$$\inf_{q_\omega(\mathbf{z})=p(\mathbf{z})} \mathbb{E}_{\mathbb{P}_{\mathbf{x}}}[\mathcal{L}_{ELBO}(\mathbf{x};\theta,\omega)] \leq -\mathrm{W}^{\star}_{\mathcal{L}}(\mathbb{P}_{\mathbf{x}}, \mathbb{P}_{\mathrm{G}_i}) - \frac{1}{2}\log\pi\,, \tag{9}$$

Eq. (9) proves Theorem 1 $\square$

B The proof of Theorem 2

Theorem 2 *Let $\mathbb{P}_{m_i}$ and $\mathbb{P}_{\mathbf{x}}$ be the source and target domain, respectively. Based on the results from Theorem 1, we derive the bound on ELBO between $\mathbb{P}_{m_i}$ and $\mathbb{P}_{\mathbf{x}}$ at the training step (t_i) :*

$$\begin{aligned}\inf_{q_\omega(\mathbf{z})=p(\mathbf{z})} \mathbb{E}_{\mathbb{P}_{\mathbf{x}}} \mathbb{E}_{q_\omega(\mathbf{z}\,|\,\mathbf{x})}[\mathcal{L}(\mathbf{x}, \mathrm{G}_i(\mathbf{z}))] \leq & \inf_{q_\omega(\mathbf{z})=p(\mathbf{z})} \mathbb{E}_{\mathbb{P}_{m_i}}[\mathcal{L}_{ELBO}(\mathbf{x};\theta,\omega)] \\ & + 2\mathrm{W}^{\star}_{\mathcal{L}}(\mathbb{P}_{m_i}, \mathbb{P}_{\mathrm{G}_i}) \\ & - \mathrm{W}^{\star}_{\mathcal{L}}(\mathbb{P}_{\mathbf{x}}, \mathbb{P}_{m_i}) + \tilde{\mathrm{F}}(\mathbb{P}_{\mathrm{G}_i}, \mathbb{P}_{m_i})\,,\end{aligned} \tag{10}$$

Proof. We first consider Eq. (9), expressed as :

$$\inf_{q_\omega(\mathbf{z})=p(\mathbf{z})} \mathbb{E}_{\mathbb{P}_{\mathbf{x}}}[\mathcal{L}_{ELBO}(\mathbf{x};\theta,\omega)] \leq -\mathrm{W}^{\star}_{\mathcal{L}}(\mathbb{P}_{\mathbf{x}}, \mathbb{P}_{\mathrm{G}_i}) - \frac{1}{2}\log\pi\,, \tag{11}$$

We then add $-\mathrm{W}^{\star}_{\mathcal{L}}(\mathbb{P}_{m_i}, \mathbb{P}_{\mathrm{G}_i})$ in both sides of Eq. (11), resulting in :

$$\begin{aligned}&\inf_{q_\omega(\mathbf{z})=p(\mathbf{z})} \mathbb{E}_{\mathbb{P}_{\mathbf{x}}}[\mathcal{L}_{ELBO}(\mathbf{x};\theta,\omega)] - \mathrm{W}^{\star}_{\mathcal{L}}(\mathbb{P}_{m_i}, \mathbb{P}_{\mathrm{G}_i}) \leq -\mathrm{W}^{\star}_{\mathcal{L}}(\mathbb{P}_{m_i}, \mathbb{P}_{\mathrm{G}_i}) - \mathrm{W}^{\star}_{\mathcal{L}}(\mathbb{P}_{\mathbf{x}}, \mathbb{P}_{\mathrm{G_i}}) \\ &- \frac{1}{2}\log\pi\,,\end{aligned} \tag{12}$$

The first term in the right-hand side (RHS) of Eq. (12) is bounded by, (see Eq.(6) in the paper) :

$$\inf_{q_\omega(\mathbf{z})=p(\mathbf{z})} \mathbb{E}_{\mathbb{P}_{m_i}} \mathbb{E}_{q_\omega(\mathbf{z}\,|\,\mathbf{x})}[-\mathcal{L}(\mathbf{x}, \mathrm{G}_i(\mathbf{z}))] \leq -\mathrm{W}^{\star}_{\mathcal{L}}(\mathbb{P}_{m_i}, \mathbb{P}_{\mathrm{G_i}}), \tag{13}$$

From Eq. (13), we have :

$$\begin{aligned}&\inf_{q_\omega(\mathbf{z})=p(\mathbf{z})} \mathbb{E}_{\mathbb{P}_{m_i}} \mathbb{E}_{q_\omega(\mathbf{z}\,|\,\mathbf{x})}[-\mathcal{L}(\mathbf{x}, \mathrm{G}_i(\mathbf{z}))] \\ &+ \Big| \inf_{q_\omega(\mathbf{z})=p(\mathbf{z})} \mathbb{E}_{\mathbb{P}_{m_i}} \mathbb{E}_{q_\omega(\mathbf{z}\,|\,\mathbf{x})}[-\mathcal{L}(\mathbf{x}, \mathrm{G}_i(\mathbf{z}))] - \mathrm{W}^{\star}_{\mathcal{L}}(\mathbb{P}_{m_i}, \mathbb{P}_{\mathrm{G}_i}) \Big| \geq -\mathrm{W}^{\star}_{\mathcal{L}}(\mathbb{P}_{m_i}, \mathbb{P}_{\mathrm{G}_i}),\end{aligned} \tag{14}$$

We then replace the first term in the RHS of Eq. (12) by the above equation, resulting in :

$$\begin{aligned}
&\inf_{q_\omega(\mathbf{z})=p(\mathbf{z})} \mathbb{E}_{\mathbb{P}_\mathbf{x}}[\mathcal{L}_{ELBO}(\mathbf{x};\theta,\omega)] - \mathrm{W}^\star_\mathcal{L}(\mathbb{P}_{m_i},\mathbb{P}_{\mathrm{G}_i}) \\
&\leq \inf_{q_\omega(\mathbf{z})=p(\mathbf{z})} \mathbb{E}_{\mathbb{P}_{m_i}} \mathbb{E}_{q_\omega(\mathbf{z}\,|\,\mathbf{x})}[-\mathcal{L}(\mathbf{x},\mathrm{G}_i(\mathbf{z}))] - \mathrm{W}^\star_\mathcal{L}(\mathbb{P}_\mathbf{x},\mathbb{P}_{\mathbf{G}_i}) \\
&+ \Big| \inf_{q_\omega(\mathbf{z})=p(\mathbf{z})} \mathbb{E}_{\mathbb{P}_{m_i}} \mathbb{E}_{q_\omega(\mathbf{z}\,|\,\mathbf{x})}[-\mathcal{L}(\mathbf{x},\mathrm{G}_i(\mathbf{z}))] - \mathrm{W}^\star_\mathcal{L}(\mathbb{P}_{m_i},\mathbb{P}_{\mathbf{G}_i}) \Big| \\
&- \frac{1}{2}\log\pi\,,
\end{aligned} \tag{15}$$

We then add the negative KL divergence term in both sides of Eq. (15) :

$$\begin{aligned}
&\inf_{q_\omega(\mathbf{z})=p(\mathbf{z})} \mathbb{E}_{\mathbb{P}_\mathbf{x}}[\mathcal{L}_{ELBO}(\mathbf{x};\theta,\omega)] - \mathrm{W}^\star_\mathcal{L}(\mathbb{P}_{m_i},\mathbb{P}_{\mathrm{G}_i}) - \inf_{q_\omega(\mathbf{z})=p(\mathbf{z})} \mathbb{E}_{\mathbb{P}_{m_i}}[D_{KL}(q_\omega(\mathbf{z}\,|\,\mathbf{x})\,||\,p(\mathbf{z}))] \leq \\
&\underbrace{\inf_{q_\omega(\mathbf{z})=p(\mathbf{z})} \mathbb{E}_{\mathbb{P}_{m_i}} \mathbb{E}_{q_\omega(\mathbf{z}\,|\,\mathbf{x})}[-\mathcal{L}(\mathbf{x},\mathrm{G}_i(\mathbf{z}))] - \inf_{q_\omega(\mathbf{z})=p(\mathbf{z})} \mathbb{E}_{\mathbb{P}_{m_i}}[D_{KL}(q_\omega(\mathbf{z}\,|\,\mathbf{x})\,||\,p(\mathbf{z}))] - \frac{1}{2}\log\pi}_{\text{ELBO}} \\
&- \mathrm{W}^\star_\mathcal{L}(\mathbb{P}_\mathbf{x},\mathbb{P}_{\mathbf{G}_i}) + \Big| \inf_{q_\omega(\mathbf{z})=p(\mathbf{z})} \mathbb{E}_{\mathbb{P}_{m_i}} \mathbb{E}_{q_\omega(\mathbf{z}\,|\,\mathbf{x})}[-\mathcal{L}(\mathbf{x},\mathrm{G}_i(\mathbf{z}))] - \mathrm{W}^\star_\mathcal{L}(\mathbb{P}_{m_i},\mathbb{P}_{\mathbf{G}_i}) \Big|\,,
\end{aligned} \tag{16}$$

According to the definition of ELBO, Eq. (16) can be rewritten as :

$$\begin{aligned}
&\inf_{q_\omega(\mathbf{z})=p(\mathbf{z})} \mathbb{E}_{\mathbb{P}_\mathbf{x}}[\mathcal{L}_{ELBO}(\mathbf{x};\theta,\omega)] - \mathrm{W}^\star_\mathcal{L}(\mathbb{P}_{m_i},\mathbb{P}_{\mathrm{G}_i}) - \inf_{q_\omega(\mathbf{z})=p(\mathbf{z})} \mathbb{E}_{\mathbb{P}_{m_i}}[D_{KL}(q_\omega(\mathbf{z}\,|\,\mathbf{x})\,||\,p(\mathbf{z}))] \leq \\
&\inf_{q_\omega(\mathbf{z})=p(\mathbf{z})} \mathbb{E}_{\mathbb{P}_{m_i}}[\mathcal{L}_{ELBO}(\mathbf{x};\theta,\omega)] - \mathrm{W}^\star_\mathcal{L}(\mathbb{P}_\mathbf{x},\mathbb{P}_{\mathbf{G}_i}) \\
&+ \Big| \inf_{q_\omega(\mathbf{z})=p(\mathbf{z})} \mathbb{E}_{\mathbb{P}_{m_i}} \mathbb{E}_{q_\omega(\mathbf{z}\,|\,\mathbf{x})}[-\mathcal{L}(\mathbf{x},\mathrm{G}_i(\mathbf{z}))] - \mathrm{W}^\star_\mathcal{L}(\mathbb{P}_{m_i},\mathbb{P}_{\mathrm{G}_i}) \Big|\,,
\end{aligned} \tag{17}$$

Then we rewrite Eq. (17), resulting in :

$$\begin{aligned}
\inf_{q_\omega(\mathbf{z})=p(\mathbf{z})} \mathbb{E}_{\mathbb{P}_\mathbf{x}}[\mathcal{L}_{ELBO}(\mathbf{x};\theta,\omega)] \leq &\inf_{q_\omega(\mathbf{z})=p(\mathbf{z})} \mathbb{E}_{\mathbb{P}_{m_i}}[\mathcal{L}_{ELBO}(\mathbf{x};\theta,\omega)] + \mathrm{W}^\star_\mathcal{L}(\mathbb{P}_{m_i},\mathbb{P}_{\mathrm{G}_i}) \\
&- \mathrm{W}^\star_\mathcal{L}(\mathbb{P}_\mathbf{x},\mathbb{P}_{\mathrm{G}_\mathrm{i}}) \\
&+ \inf_{q_\omega(\mathbf{z})=p(\mathbf{z})} \mathbb{E}_{\mathbb{P}_{m_i}}[D_{KL}(q_\omega(\mathbf{z}\,|\,\mathbf{x})\,||\,p(\mathbf{z}))] \\
&+ \Big| \inf_{q_\omega(\mathbf{z})=p(\mathbf{z})} \mathbb{E}_{\mathbb{P}_{m_i}} \mathbb{E}_{q_\omega(\mathbf{z}\,|\,\mathbf{x})}[-\mathcal{L}(\mathbf{x},\mathrm{G}_i(\mathbf{z}))] - \mathrm{W}^\star_\mathcal{L}(\mathbb{P}_{m_i},\mathbb{P}_{\mathrm{G}_i}) \Big|\,,
\end{aligned} \tag{18}$$

We consider that $\mathcal{L}(\cdot)$ satisfies triangle inequality, we have :

$$\mathrm{W}^\star_\mathcal{L}(\mathbb{P}_{m_i},\mathbb{P}_{\mathrm{G}_\mathrm{i}}) + \mathrm{W}^\star_\mathcal{L}(\mathbb{P}_\mathbf{x},\mathbb{P}_{\mathrm{G}_\mathrm{i}}) \geq \mathrm{W}^\star_\mathcal{L}(\mathbb{P}_\mathbf{x},\mathbb{P}_{m_i}) \tag{19}$$

We move the second term in the left-hand side of Eq. (19) in the right-hand side :

$$\mathrm{W}^\star_\mathcal{L}(\mathbb{P}_\mathbf{x},\mathbb{P}_{\mathrm{G}_\mathrm{i}}) \geq \mathrm{W}^\star_\mathcal{L}(\mathbb{P}_\mathbf{x},\mathbb{P}_{m_i}) - \mathrm{W}^\star_\mathcal{L}(\mathbb{P}_{m_i},\mathbb{P}_{\mathrm{G}_i}) \tag{20}$$

Then we replace $\mathrm{W}^{\star}_{\mathcal{L}}(\mathbb{P}_{\mathbf{x}}, \mathbb{P}_{\mathrm{G_i}})$ from Eq. (18) by the expression of Eq. (20), resulting in :

$$\begin{aligned}\inf_{q_\omega(\mathbf{z})=p(\mathbf{z})} \mathbb{E}_{\mathbb{P}_{\mathbf{x}}}[\mathcal{L}_{ELBO}(\mathbf{x};\theta,\omega)] \leq & \inf_{q_\omega(\mathbf{z})=p(\mathbf{z})} \mathbb{E}_{\mathbb{P}_{m_i}}[\mathcal{L}_{ELBO}(\mathbf{x};\theta,\omega)] \\ & + 2\mathrm{W}^{\star}_{\mathcal{L}}(\mathbb{P}_{m_i}, \mathbb{P}_{\mathrm{G}_i}) \\ & - \mathrm{W}^{\star}_{\mathcal{L}}(\mathbb{P}_{\mathbf{x}}, \mathbb{P}_{m_i}) + \tilde{\mathrm{F}}(\mathbb{P}_{\mathrm{G}_i}, \mathbb{P}_{m_i}) \,,\end{aligned} \tag{21}$$

where $\tilde{\mathrm{F}}(\mathbb{P}_{\mathrm{G}_i}, \mathbb{P}_{m_i})$ is expressed as :

$$\begin{aligned}\tilde{\mathrm{F}}(\mathbb{P}_{\mathrm{G}_i}, \mathbb{P}_{m_i}) = & \inf_{q_\omega(\mathbf{z})=p(\mathbf{z})} \mathbb{E}_{\mathbb{P}_{m_i}}[D_{KL}(q_\omega(\mathbf{z}\,|\,\mathbf{x}) \,||\, p(\mathbf{z}))] \\ & + \Big| \inf_{q_\omega(\mathbf{z})=p(\mathbf{z})} \mathbb{E}_{\mathbb{P}_{m_i}} \mathbb{E}_{q_\omega(\mathbf{z}\,|\,\mathbf{x})}[-\mathcal{L}(\mathbf{x}, \mathrm{G}_i(\mathbf{z}))] - \mathrm{W}^{\star}_{\mathcal{L}}(\mathbb{P}_{m_i}, \mathbb{P}_{\mathbf{G}_i}) \Big|\end{aligned} \tag{22}$$

□

C The proof of Lemma 2

Lemma 2. Let $\{\mathbb{P}_{\mathbf{x}^1}, \cdots, \mathbb{P}_{\mathbf{x}^n}\}$ be a set of n target domains. Based on Definition 4 of the paper, the bound on ELBO for the mixture model is derived as :

$$\sum_{j=1}^{n} \mathbb{E}_{\mathbb{P}_{\mathbf{x}^j}}[\mathcal{L}_{ELBO}(\mathbf{x};\theta,\omega)] \leq \sum_{i=1}^{n} \{\mathrm{F}^{\star}(\mathbb{P}_{\mathbf{x}^i})\} \,. \tag{23}$$

where $\mathrm{F}^{\star}(\mathbb{P}_{\mathbf{x}^j})$ is the selection function, defined as :

$$\begin{aligned}\mathrm{F}^{\star}(\mathbb{P}_{\mathbf{x}^i}) = & \max_{j=1,\cdots,k} \Big\{ \mathbb{E}_{\mathbb{P}_{m_{q_j}}}[\mathcal{L}_{ELBO}(\mathbf{x};\theta,\omega)] \\ & + 2\mathrm{W}^{\star}_{\mathcal{L}}(\mathbb{P}_{m_{q_j}}, \mathbb{P}_{\mathrm{G}_i}) - \mathrm{W}^{\star}_{\mathcal{L}}(\mathbb{P}_{\mathbf{x}^i}, \mathbb{P}_{m_{q_j}}) \\ & + \tilde{\mathrm{F}}(\mathbb{P}_{\mathrm{G}_{q_j}}, \mathbb{P}_{m_{q_j}}) \Big\} \,.\end{aligned} \tag{24}$$

The advantage of the dynamic expansion model than a single model. Lemma 1 of the paper have demonstrated that the diversity in the memory can relieve the negative transfer in the past target sets. However, when the memory size is restricted, the stored samples for a certain past target set would be few and thus can not represent the exact underlying distribution of the target set. Eq. (23) shows that the dynamic expansion model can relieve this issue by learning a diversity of mixing components where each component would capture different underlying data distributions. For instance, we assume that we have trained n components where each component (i-th component) only captures a certain target domain $\mathbb{P}_{\mathbf{x}^i}$. We can have the maximum upper bound to Eq. (23) when we make the component selection (Eq. (24)). This inspires us to combine the proposed Online Cooperative Memorization (OCM) with the expansion mechanism to further improve the performance.

Proof. Since each component h_i had converged at the training step t_{q_i}, with the memory $\mathcal{M}_{q_i}$, we then can perform the component selection for the evaluation of ELBO. In

Lemma 2, we assume that we have trained k components at the training step t_{q_k}. For a given target domain $\mathbb{P}_{\mathbf{x}^i}$, from Theorem 2 of the paper, the bound for the mixture model with the component selection can be defined as :

$$\begin{aligned}\mathbb{E}_{\mathbb{P}_{\mathbf{x}^j}}[\mathcal{L}_{ELBO}(\mathbf{x};\theta,\omega)] \leq \max_{j=1,\cdots,k} \Big\{ & \mathbb{E}_{\mathbb{P}_{m_{q_j}}}[\mathcal{L}_{ELBO}(\mathbf{x};\theta,\omega)] + 2\mathrm{W}^{\star}_{\mathcal{L}}(\mathbb{P}_{m_{q_j}}, \mathbb{P}_{\mathrm{G}_i}) \\ & - \mathrm{W}^{\star}_{\mathcal{L}}(\mathbb{P}_{\mathbf{x}^i}, \mathbb{P}_{m_{q_j}}) \\ & + \tilde{\mathrm{F}}(\mathbb{P}_{\mathrm{G}_{q_j}}, \mathbb{P}_{m_{q_j}}) \Big\}.\end{aligned} \tag{25}$$

Then we extend Eq. (25) into multiple target domains, expressed as :

$$\begin{aligned}\sum_{j=1}^{n} \mathbb{E}_{\mathbb{P}_{\mathbf{x}^j}}[\mathcal{L}_{ELBO}(\mathbf{x};\theta,\omega)] \leq \sum_{i=1}^{n} \Big\{ \max_{j=1,\cdots,k} \Big\{ & \mathbb{E}_{\mathbb{P}_{m_{q_j}}}[\mathcal{L}_{ELBO}(\mathbf{x};\theta,\omega)] \\ & + 2\mathrm{W}^{\star}_{\mathcal{L}}(\mathbb{P}_{m_{q_j}}, \mathbb{P}_{\mathrm{G}_i}) - \mathrm{W}^{\star}_{\mathcal{L}}(\mathbb{P}_{\mathbf{x}^i}, \mathbb{P}_{m_{q_j}}) \\ & + \tilde{\mathrm{F}}(\mathbb{P}_{\mathrm{G}_{q_j}}, \mathbb{P}_{m_{q_j}}) \Big\} \Big\}.\end{aligned} \tag{26}$$

Eq. (26) proves **Lemma 2**

□

D The proof of Theorem 3

Theorem 3 *Let $\mathcal{A} = \{a_1, \cdots, a_n\}$ be a set where each a_i represents the index of the component that has trained only once. Let $\tilde{\mathcal{A}} = \{\tilde{a}_1, \cdots, \tilde{a}_n\}$ be a set of task labels where each $\tilde{a}_i$ represents the index of the task learned by the a_i-th component. Let $\mathcal{B} = \{b_1, \cdots, b_{k-n}\}$ be a set where each b_i represents the index of the component that is trained more than once. Let $\tilde{b}_i = \{\tilde{b}_i^1, \cdots, \tilde{b}_i^m\}$ be a set of task labels for the b_i-th component. Let c_i^j represent the time of the generative replay processes for the $\tilde{b}_{(i,j)}$-th task, achieved by the b_i-th component.*

$$\sum_{i=1}^{|\mathcal{A}|} \Big\{ \mathbb{E}_{\mathbb{P}_{\hat{\mathbf{x}}^{\tilde{a}_i}}}[\mathcal{L}_{ELBO}(\mathbf{x};\theta,\omega)] \Big\} + \sum_{i=1}^{|\mathcal{B}|} \Big\{ \sum_{q=1}^{|\tilde{b}_i|} \big\{ \mathbb{E}_{\mathbb{P}_{\hat{\mathbf{x}}^{\tilde{b}_i^q}}}[\mathcal{L}_{ELBO}(\mathbf{x};\theta,\omega)] \big\} \Big\} \leq \mathcal{R}_S + \mathcal{R}_M \tag{27}$$

where $\mathcal{R}_S$ and $\mathcal{R}_M$ are defined as follows.

$$\begin{aligned}\mathcal{R}_S = \sum_{i=1}^{|\mathcal{A}|} \Big\{ & \mathbb{E}_{\mathbb{P}_{\mathbf{x}^{\tilde{a}_i}}}[\mathcal{L}_{ELBO}(\mathbf{x};\theta,\omega)] + 2\mathrm{W}^{\star}_{\mathcal{L}}(\mathbb{P}_{\mathbf{x}^{\tilde{a}_i}}, \mathbb{P}_{\mathrm{G}^{a_i}}) - \mathrm{W}^{\star}_{\mathcal{L}}(\mathbb{P}_{\hat{\mathbf{x}}^{\tilde{a}_i}}, \mathbb{P}_{\mathbf{x}^{\tilde{a}_i}}) \\ & + \tilde{\mathrm{F}}(\mathbb{P}_{\mathrm{G}^{a_i}}, \mathbb{P}_{\mathbf{x}^{\tilde{a}_i}}) \Big\}\end{aligned} \tag{28}$$

$$\begin{aligned}\mathcal{R}_M = \sum_{i=1}^{|\mathcal{B}|} \Big\{ \sum_{q=1}^{|\tilde{b}_i|} \Big\{ & \mathbb{E}_{\mathbb{P}_{\tilde{\mathbf{x}}^{(\tilde{b}_i^q, c_i^q)}}}[\mathcal{L}_{ELBO}(\mathbf{x};\theta,\omega)] + \sum_{s=0}^{c_i^q} \big\{ 2\mathrm{W}^{\star}_{\mathcal{L}}(\mathbb{P}_{\tilde{\mathbf{x}}^{(\tilde{b}_i^q, s)}}, \mathbb{P}_{\mathrm{G}^{b_i}}) \\ & - \mathrm{W}^{\star}_{\mathcal{L}}(\mathbb{P}_{\hat{\mathbf{x}}^{(\tilde{b}_i^q, s-1)}}, \mathbb{P}_{\tilde{\mathbf{x}}^{(\tilde{b}_i^q, s)}}) + \tilde{\mathrm{F}}(\mathbb{P}_{\mathrm{G}^{b_i}}, \mathbb{P}_{\tilde{\mathbf{x}}^{(\tilde{b}_i^q, s)}}) \big\} \Big\} \Big\}\end{aligned} \tag{29}$$

When the mixture has only a single component. Let us assume that we train a mixture model having a single VAE component ($k = 1$) for learning a sequence of N tasks. By using Theorem 2 the bound on ELBO for this mixture model is defined as :

$$\begin{aligned}\sum_{q=1}^{N}\big\{\mathbb{E}_{\mathbb{P}_{\hat{\mathbf{x}}^{\tilde{b}_1^q}}}[\mathcal{L}_{ELBO}(\mathbf{x};\theta,\omega)]\big\} \leq \sum_{q=1}^{N}\Big\{&\mathbb{E}_{\mathbb{P}_{\tilde{\mathbf{x}}^{(\tilde{b}_1^q,c_1^q)}}}[\mathcal{L}_{ELBO}(\mathbf{x};\theta,\omega)] \\ &+\sum_{s=0}^{c_1^q}\big\{2\mathrm{W}^{\star}_{\mathcal{L}}(\mathbb{P}_{\tilde{\mathbf{x}}^{(\tilde{b}_1^q,s)}},\mathbb{P}_{\mathrm{G}^{b_1}}) \\ &-\mathrm{W}^{\star}_{\mathcal{L}}(\mathbb{P}_{\hat{\mathbf{x}}^{(\tilde{b}_1^q,s-1)}},\mathbb{P}_{\tilde{\mathbf{x}}^{(\tilde{b}_1^q,s)}})+\tilde{\mathrm{F}}(\mathbb{P}_{\mathrm{G}^{b_1}},\mathbb{P}_{\tilde{\mathbf{x}}^{(\tilde{b}_1^q,s)}})\big\}\Big\}\end{aligned} \tag{30}$$

where each c_1^q is equal to $N - q$ and each $\tilde{b}_1^q$ corresponds the q-th task label. We then rewrite Eq. (30) as :

$$\begin{aligned}\sum_{q=1}^{N}\big\{\mathbb{E}_{\mathbb{P}_{\tilde{\mathbf{x}}^q}}[\mathcal{L}_{ELBO}(\mathbf{x};\theta,\omega)]\big\} \leq \sum_{q=1}^{N}\Big\{&\mathbb{E}_{\mathbb{P}_{\tilde{\mathbf{x}}^{(q,N-q)}}}[\mathcal{L}_{ELBO}(\mathbf{x};\theta,\omega)] \\ &+\sum_{s=0}^{N-q}\big\{2\mathrm{W}^{\star}_{\mathcal{L}}(\mathbb{P}_{\tilde{\mathbf{x}}^{(q,s)}},\mathbb{P}_{\mathrm{G}^1}) \\ &-\mathrm{W}^{\star}_{\mathcal{L}}(\mathbb{P}_{\hat{\mathbf{x}}^{(q,s-1)}},\mathbb{P}_{\tilde{\mathbf{x}}^{(q,s)}})+\tilde{\mathrm{F}}(\mathbb{P}_{\mathrm{G}^1},\mathbb{P}_{\tilde{\mathbf{x}}^{(q,s)}})\big\}\Big\}\end{aligned} \tag{31}$$

From Eq. (31), we observe that task trained earlier in the continuously learning process (q is small) tends to be forgotten easier than the recently trained tasks (q is large), because the knowledge (remembering) of the earlier learned task accumulates more error terms $\sum_{s=0}^{N-q}\big\{2\mathrm{W}^{\star}_{\mathcal{L}}(\mathbb{P}_{\tilde{\mathbf{x}}^{(q,s)}},\mathbb{P}_{\mathrm{G}^1})-\mathrm{W}^{\star}_{\mathcal{L}}(\mathbb{P}_{\hat{\mathbf{x}}^{(q,s-1)}},\mathbb{P}_{\tilde{\mathbf{x}}^{(q,s)}})+\tilde{\mathrm{F}}(\mathbb{P}_{\mathrm{G}^1},\mathbb{P}_{\tilde{\mathbf{x}}^{(q,s)}})\big\}$. This result supports the statements made in the **Remark** to **Theorem 3** of the paper.

Proof. Let us firstly consider a certain component (a_i-th component) that has been trained only once. From Theorem 2 we derive the bound as follows :

$$\begin{aligned}\mathbb{E}_{\mathbb{P}_{\hat{\mathbf{x}}^{\tilde{a}_i}}}[\mathcal{L}_{ELBO}(\mathbf{x};\theta,\omega)] \leq \mathbb{E}_{\mathbb{P}_{\mathbf{x}^{\tilde{a}_i}}}[\mathcal{L}_{ELBO}(\mathbf{x};\theta,\omega)] &+ 2\mathrm{W}^{\star}_{\mathcal{L}}(\mathbb{P}_{\mathbf{x}^{\tilde{a}_i}},\mathbb{P}_{\mathrm{G}^{a_i}}) \\ &- \mathrm{W}^{\star}_{\mathcal{L}}(\mathbb{P}_{\hat{\mathbf{x}}^{\tilde{a}_i}},\mathbb{P}_{\mathbf{x}^{\tilde{a}_i}}) \\ &+ \tilde{\mathrm{F}}(\mathbb{P}_{\mathrm{G}^{a_i}},\mathbb{P}_{\mathbf{x}^{\tilde{a}_i}}),\end{aligned} \tag{32}$$

Eq. (32) holds because we treat $\mathbb{P}_{\hat{\mathbf{x}}^{\tilde{a}_i}}$ and $\mathbb{P}_{\mathbf{x}^{\tilde{a}_i}}$ as the target and source domain respectively. In the following, we consider a component (b_i-th component) that has been trained more than once. Since the b_i-th component would learn more than one task, we particularly focus on a certain task ($\tilde{b}_i^q$-th task). We firstly consider to treat $\mathbb{P}_{\hat{\mathbf{x}}^{\tilde{b}_i^q}}$ and $\mathbb{P}_{\mathbf{x}^{\tilde{b}_i^q}}$ as the target and source domain respectively. Then we derive the bound as :

$$\begin{aligned}\mathbb{E}_{\mathbb{P}_{\hat{\mathbf{x}}^{\tilde{b}_i^q}}}[\mathcal{L}_{ELBO}(\mathbf{x};\theta,\omega)] \leq \mathbb{E}_{\mathbb{P}_{\mathbf{x}^{\tilde{b}_i^q}}}[\mathcal{L}_{ELBO}(\mathbf{x};\theta,\omega)] &+ 2\mathrm{W}^{\star}_{\mathcal{L}}(\mathbb{P}_{\mathbf{x}^{\tilde{b}_i^q}},\mathbb{P}_{\mathrm{G}^{b_i}}) \\ &- \mathrm{W}^{\star}_{\mathcal{L}}(\mathbb{P}_{\hat{\mathbf{x}}^{\tilde{b}_i^q}},\mathbb{P}_{\mathbf{x}^{\tilde{b}_i^q}}) \\ &+ \tilde{\mathrm{F}}(\mathbb{P}_{\mathrm{G}^{b_i}},\mathbb{P}_{\mathbf{x}^{\tilde{b}_i^q}}),\end{aligned} \tag{33}$$

We do not specify the state (the number of retraining processes) of each generator distribution $\mathbb{P}_{\mathrm{G}i}$ in order to simplify the notation. If $c_i^j > 0$, then we can have the empirical distribution $\mathbb{P}_{\hat{\mathbf{x}}^{(\tilde{b}_i^q,1)}}$ for one time of the generative replay processes (See Definition 6 in the paper). We treat $\mathbb{P}_{\hat{\mathbf{x}}^{(\tilde{b}_i^q,0)}} = \mathbb{P}_{\hat{\mathbf{x}}^{\tilde{b}_i^q}}$ and $\mathbb{P}_{\hat{\mathbf{x}}^{(\tilde{b}_i^q,1)}}$ as the target and source domain, respectively. We then derive the bound between $\mathbb{P}_{\hat{\mathbf{x}}^{(\tilde{b}_{(i,q)},0)}}$ and $\mathbb{P}_{\hat{\mathbf{x}}^{(\tilde{b}_{(i,q)},1)}}$ as follows :

$$\begin{aligned}\mathbb{E}_{\mathbb{P}_{\tilde{\mathbf{x}}^{(\tilde{b}_i^q,0)}}}[\mathcal{L}_{ELBO}(\mathbf{x};\theta,\omega)] \leq \mathbb{E}_{\mathbb{P}_{\tilde{\mathbf{x}}^{(\tilde{b}_i^q,1)}}}[\mathcal{L}_{ELBO}(\mathbf{x};\theta,\omega)] + 2\mathrm{W}_{\mathcal{L}}^{\star}(\mathbb{P}_{\tilde{\mathbf{x}}^{(\tilde{b}_{(i,q)},1)}}, \mathbb{P}_{\mathrm{G}^{b_i}}) \\ - \mathrm{W}_{\mathcal{L}}^{\star}(\mathbb{P}_{\tilde{\mathbf{x}}^{(\tilde{b}_i^q,1)}}, \mathbb{P}_{\tilde{\mathbf{x}}^{(\tilde{b}_i^q,0)}}) \\ + \tilde{\mathrm{F}}(\mathbb{P}_{\mathrm{G}^{b_i}}, \mathbb{P}_{\tilde{\mathbf{x}}^{(\tilde{b}_i^q,1)}}),\end{aligned} \tag{34}$$

Through mathematical induction, we have the bounds :

$$\begin{aligned}\mathbb{E}_{\mathbb{P}_{\tilde{\mathbf{x}}^{(\tilde{b}_i^q,1)}}}[\mathcal{L}_{ELBO}(\mathbf{x};\theta,\omega)] &\leq \mathbb{E}_{\mathbb{P}_{\tilde{\mathbf{x}}^{(\tilde{b}_i^q,2)}}}[\mathcal{L}_{ELBO}(\mathbf{x};\theta,\omega)] + 2\mathrm{W}_{\mathcal{L}}^{\star}(\mathbb{P}_{\tilde{\mathbf{x}}^{(\tilde{b}_i^q,2)}}, \mathbb{P}_{\mathrm{G}^{b_i}}) \\ &- \mathrm{W}_{\mathcal{L}}^{\star}(\mathbb{P}_{\tilde{\mathbf{x}}^{(\tilde{b}_i^q,2)}}, \mathbb{P}_{\tilde{\mathbf{x}}^{(\tilde{b}_i^q,1)}}) \\ &+ \tilde{\mathrm{F}}(\mathbb{P}_{\mathrm{G}^{b_i}}, \mathbb{P}_{\tilde{\mathbf{x}}^{(\tilde{b}_i^q,2)}}) \\ &\dots \\ \mathbb{E}_{\mathbb{P}_{\tilde{\mathbf{x}}^{(\tilde{b}_i^q,c_i^q-1)}}}[\mathcal{L}_{ELBO}(\mathbf{x};\theta,\omega)] &\leq \mathbb{E}_{\mathbb{P}_{\tilde{\mathbf{x}}^{(\tilde{b}_i^q,c_i^q)}}}[\mathcal{L}_{ELBO}(\mathbf{x};\theta,\omega)] + 2\mathrm{W}_{\mathcal{L}}^{\star}(\mathbb{P}_{\tilde{\mathbf{x}}^{(\tilde{b}_i^q,c_i^q)}}, \mathbb{P}_{\mathrm{G}^{b_i}}) \\ &- \mathrm{W}_{\mathcal{L}}^{\star}(\mathbb{P}_{\tilde{\mathbf{x}}^{(\tilde{b}_i^q,c_i^q)}}, \mathbb{P}_{\tilde{\mathbf{x}}^{(\tilde{b}_i^q,c_i^q-1)}}) + \tilde{\mathrm{F}}(\mathbb{P}_{\mathrm{G}^{b_i}}, \mathbb{P}_{\tilde{\mathbf{x}}^{(\tilde{b}_i^q,c_i^q)}})\end{aligned} \tag{35}$$

We then sum up all above inequalities, resulting in :

$$\begin{aligned}\mathbb{E}_{\mathbb{P}_{\tilde{\mathbf{x}}^{\tilde{b}_i^q}}}[\mathcal{L}_{ELBO}(\mathbf{x};\theta,\omega)] \leq \mathbb{E}_{\mathbb{P}_{\tilde{\mathbf{x}}^{(\tilde{b}_i^q,c_i^q)}}}[\mathcal{L}_{ELBO}(\mathbf{x};\theta,\omega)] + \sum_{s=0}^{\tilde{c}_{(i,q)}} \Big\{2\mathrm{W}_{\mathcal{L}}^{\star}(\mathbb{P}_{\tilde{\mathbf{x}}^{(\tilde{b}_i^q,s)}}, \mathbb{P}_{\mathrm{G}^{b_i}}) \\ - \mathrm{W}_{\mathcal{L}}^{\star}(\mathbb{P}_{\hat{\mathbf{x}}^{(\tilde{b}i^q,s-1)}}, \mathbb{P}_{\tilde{\mathbf{x}}^{(\tilde{b}_i^q,s)}}) \\ + \tilde{\mathrm{F}}(\mathbb{P}_{\mathrm{G}^{b_i}}, \mathbb{P}_{\tilde{\mathbf{x}}^{(\tilde{b}_i^q,s)}})\Big\},\end{aligned} \tag{36}$$

Eq. (36) describes the bound for a single task. We then extend this bound to components learning more than one task :

$$\begin{aligned}\sum_{i=1}^{|\mathcal{B}|}\Big\{\sum_{q=1}^{|\tilde{b}_i|}\big\{\mathbb{E}_{\mathbb{P}_{\tilde{\mathbf{x}}^{\tilde{b}_i^q}}}[\mathcal{L}_{ELBO}(\mathbf{x};\theta,\omega)]\big\}\Big\} \leq \sum_{i=1}^{|\mathcal{B}|}\Big\{\sum_{q=1}^{|\tilde{b}_i|}\Big\{\mathbb{E}_{\mathbb{P}_{\tilde{\mathbf{x}}^{(\tilde{b}_i^q,c_i^q)}}}[\mathcal{L}_{ELBO}(\mathbf{x};\theta,\omega)] \\ + \sum_{s=0}^{c_i^q}\big\{2\mathrm{W}_{\mathcal{L}}^{\star}(\mathbb{P}_{\tilde{\mathbf{x}}^{(\tilde{b}_i^q,s)}}, \mathbb{P}_{\mathrm{G}^{b_i}}) \\ - \mathrm{W}_{\mathcal{L}}^{\star}(\mathbb{P}_{\hat{\mathbf{x}}^{(\tilde{b}_i^q,s-1)}}, \mathbb{P}_{\tilde{\mathbf{x}}^{(\tilde{b}_i^q,s)}}) + \tilde{\mathrm{F}}(\mathbb{P}_{\mathrm{G}^{b_i}}, \mathbb{P}_{\tilde{\mathbf{x}}^{(\tilde{b}_i^q,s)}})\big\}\Big\}\Big\},\end{aligned} \tag{37}$$

We also extend the bound from Eq. (32) to components that would only learn one

task each :

$$\sum_{i=1}^{|\mathcal{A}|}\left\{\mathbb{E}_{\mathbb{P}_{\hat{\mathbf{x}}^{\tilde{a}_i}}}\left[\mathcal{L}_{ELBO}(\mathbf{x};\theta,\omega)\right]\right\} \leq \sum_{i=1}^{|\mathcal{A}|}\Big\{\mathbb{E}_{\mathbb{P}_{\mathbf{x}^{\tilde{a}_i}}}\left[\mathcal{L}_{ELBO}(\mathbf{x};\theta,\omega)\right] + 2\mathrm{W}^{\star}_{\mathcal{L}}(\mathbb{P}_{\mathbf{x}^{\tilde{a}_i}},\mathbb{P}_{\mathrm{G}^{a_i}}) - \mathrm{W}^{\star}_{\mathcal{L}}(\mathbb{P}_{\hat{\mathbf{x}}^{\tilde{a}_i}},\mathbb{P}_{\mathbf{x}^{\tilde{a}_i}}) + \tilde{\mathrm{F}}(\mathbb{P}_{\mathrm{G}^{a_i}},\mathbb{P}_{\mathbf{x}^{\tilde{a}_i}})\Big\}, \tag{38}$$

Eventually, the bound for all components is defined by the combination between Eq. (37) and Eq. (38), resulting in :

$$\begin{aligned}
&\sum_{i=1}^{|\mathcal{A}|}\left\{\mathbb{E}_{\mathbb{P}_{\hat{\mathbf{x}}^{\tilde{a}_i}}}\left[\mathcal{L}_{ELBO}(\mathbf{x};\theta,\omega)\right]\right\} + \sum_{i=1}^{|\mathcal{B}|}\left\{\sum_{q=1}^{|\tilde{b}_i|}\left\{\mathbb{E}_{\mathbb{P}_{\hat{\mathbf{x}}^{\tilde{b}_i^q}}}\left[\mathcal{L}_{ELBO}(\mathbf{x};\theta,\omega)\right]\right\}\right\} \leq \\
&\sum_{i=1}^{|\mathcal{A}|}\left\{\mathbb{E}_{\mathbb{P}_{\mathbf{x}^{\tilde{a}_i}}}\left[\mathcal{L}_{ELBO}(\mathbf{x};\theta,\omega)\right] + 2\mathrm{W}^{\star}_{\mathcal{L}}(\mathbb{P}_{\mathbf{x}^{\tilde{a}_i}},\mathbb{P}_{\mathrm{G}^{a_i}}) - \mathrm{W}^{\star}_{\mathcal{L}}(\mathbb{P}_{\hat{\mathbf{x}}^{\tilde{a}_i}},\mathbb{P}_{\mathbf{x}^{\tilde{a}_i}}) + \tilde{\mathrm{F}}(\mathbb{P}_{\mathrm{G}^{a_i}},\mathbb{P}_{\mathbf{x}^{\tilde{a}_i}})\right\} \\
&+ \sum_{i=1}^{|\mathcal{B}|}\left\{\sum_{q=1}^{|\tilde{b}_i|}\left\{\mathbb{E}_{\mathbb{P}_{\tilde{\mathbf{x}}^{(\tilde{b}_i^q,c_i^q)}}}\left[\mathcal{L}_{ELBO}(\mathbf{x};\theta,\omega)\right] + \sum_{s=0}^{c_i^q}\left\{2\mathrm{W}^{\star}_{\mathcal{L}}(\mathbb{P}_{\tilde{\mathbf{x}}^{(\tilde{b}_i^q,s)}},\mathbb{P}_{\mathrm{G}^{b_i}})\right.\right.\right. \\
&\left.\left.\left. - \mathrm{W}^{\star}_{\mathcal{L}}(\mathbb{P}_{\hat{\mathbf{x}}^{(\tilde{b}_i^q,s-1)}},\mathbb{P}_{\tilde{\mathbf{x}}^{(\tilde{b}_i^q,s)}}) + \tilde{\mathrm{F}}(\mathbb{P}_{\mathrm{G}^{b_i}},\mathbb{P}_{\tilde{\mathbf{x}}^{(\tilde{b}_i^q,s)}})\right\}\right\}\right\}
\end{aligned} \tag{39}$$

□

E Unsupervised forward/backward transfer

The concept of forward/backward transfer was firstly used in [15] for continual learning. Under this concept, three metrics are defined for the forward/backward transfer of the classification task. We firstly define the average accuracy (ACC) on all testing sets :

$$\mathrm{ACC} = \frac{1}{N}\sum_{i=1}^{N}\mathrm{A}_{(N,i)}\,. \tag{40}$$

where $\mathrm{A}_{(N,i)}$ represents the accuracy on the i-th testing set, predicted by the model $A(\cdot)$, which was trained on the N-th task. Then we define the backward and forward transfer criterion, denoted as BWT and FWT :

$$\mathrm{BWT} = \frac{1}{N-1}\sum_{i=1}^{N-1}\left(\mathrm{A}_{(N,i)} - \mathrm{A}_{(i,i)}\right). \tag{41}$$

$$\mathrm{FWT} = \frac{1}{N-1}\sum_{i=2}^{N}\left(\mathrm{A}_{(i-1,i)} - \mathrm{A}'_{i}\right). \tag{42}$$

where A'_i is the test accuracy for the i-th task, predicted by a model randomly initialized. However, the criteria defined by Eq. (40), (41) and (42) require to know the task

labels as well as the class labels, which is not applicable under the Online Continuous Learning (OCL) framework, which is considered in this paper.

In this section, we introduce new metrics for evaluating the forward and backward transfer abilities when the the class labels are not available. Firstly, we define the average performance as :

$$\mathrm{ELBO}_{\mathrm{avg}} = \frac{1}{N'} \sum_{i=1}^{N'} \mathrm{E}_{(N',i)} \tag{43}$$

where $\mathrm{ELBO}_{\mathrm{avg}}$ is the average ELBO on all testing sets, and $\mathrm{E}(N', i)$ is the ELBO on the i-th testing set, evaluated by the model trained at the training step N', and N' is the total number of training steps. We then define the metric for the backward transfer as :

$$\mathrm{BWT} = \frac{1}{\mathrm{N}' - 1} \sum_{\mathrm{i}=1}^{\mathrm{N}'-1} \left| \mathrm{E}_{(\mathrm{N}',\mathrm{i})} - \mathrm{E}'_{\mathrm{t_i}} \right| , \tag{44}$$

where we calculate $\mathrm{E}_{(N',i)}$ as the ELBO on the data batch $\mathbf{X}_b^i$, achieved by a model trained with the memory learnt at N', and E'_{t_i} is the ELBO calculated on the data batch $\mathbf{X}_b^i$, achieved by an auxiliary VAE model trained on data batchs $\{\mathbf{X}_b^1, \cdots, \mathbf{X}_b^i\}$. It notes that OCL does not require to access the task label and therefore the task is not identified during the training.

We define the forward transfer as :

$$\mathrm{FWT} = \frac{1}{N' - 1} \sum_{i=2}^{N'} |\mathrm{E}'_{t_i} - \mathrm{E}_{(t_{i-1},i)}| \tag{45}$$

where $\mathrm{E}_{(t_{(i-1)},i)}$ is the ELBO on $\mathbf{X}_b^i$, evaluated by a single VAE model trained with the memory learnt at $t_{(i-1)}$. The main motivation of the proposed criteria (Eq. (44) and Eq. 45) is that we use an auxiliary VAE model to give the exact approximation of the true data likelihood in each training step because there are no labels or the exact data likelihood under the unsupervised learning setting. Therefore, a small value in Eq. (44) and Eq. 45 means the small gap between the ELBO estimated by the proposed model and the approximation of the data likelihood, estimated by the auxiliary VAE model.

F Theoretical analysis for existing approaches

In this section, we apply the proposed theoretical framework for analyzing the forgetting behaviour of existing approaches. Different from prior theoretical analysis works [23, 21], the proposed theoretical analysis can be used in both the OCL and in a general continual learning setting where the task information is provided.

F.1 Online continual learning approaches

Online continual learning approaches can be divided into two branches. The former usually uses a small memory buffer to store a few past samples to avoid forgetting.

The latter combines the memory buffer and the dynamic expansion mechanism to further improve the performance. Theorem 2 and Lemma 1 of the paper can explain the forgetting behaviour of most of the existing memory-based approaches under OCL. In this section, we apply the proposed theoretical analysis for CURL [18] and CN-DPM [14].

CURL [18]. is a hybrid approach that combines the generative replay and dynamic expansion mechanism into a unified learning framework. Let $\mathbf{H}' = \{h'_1, \cdots, h'_k\}$ be a CURL model trained at t_i, which has built k components during the learning, where each h'_i is a single component. Let $\mathbf{q} = \{q_1, \cdots, q_k\}$ represent the training steps that each component converged on. For instance, h_i converged on $\mathcal{M}'_{q_i}$ at t_{q_i}, is not updated in the following training steps. Since CURL uses the generative replay to avoid forgetting, $\mathcal{M}'_{q_i}$ is defined as the memory buffer to store the replayed samples. Then $\mathbb{P}_{\mathrm{G}_{q_i}}$ and $\mathbb{P}_{m_{q_i}}$ represent the generator distribution and the distribution of samples drawn from $\mathcal{M}'_{q_i}$. For a given set of target domains $\{\mathbb{P}_{\mathbf{x}^1}, \cdots, \mathbb{P}_{\mathbf{x}^n}\}$, the bound on ELBO for CURL is defined according to Lemma 2 of the paper.

$$\sum_{j=1}^{n} \mathbb{E}_{\mathbb{P}_{\mathbf{x}^j}}[\mathcal{L}_{ELBO}(\mathbf{x}; \theta, \omega)] \leq \sum_{i=1}^{n} \{\mathrm{F}^{\star}(\mathbb{P}_{\mathbf{x}^i})\} \ . \tag{46}$$

where $\mathrm{F}^{\star}(\mathbb{P}_{\mathbf{x}^j})$ is the selection function, defined as :

$$\begin{aligned} \mathrm{F}^{\star}(\mathbb{P}_{\mathbf{x}^i}) = \max_{j=1,\cdots,k} \Big\{ & \mathbb{E}_{\mathbb{P}_{m_{q_j}}}[\mathcal{L}_{ELBO}(\mathbf{x}; \theta, \omega)] \\ & + 2\mathrm{W}^{\star}_{\mathcal{L}}(\mathbb{P}_{m_{q_j}}, \mathbb{P}_{\mathrm{G}_i}) - \mathrm{W}^{\star}_{\mathcal{L}}(\mathbb{P}_{\mathbf{x}^i}, \mathbb{P}_{m_{q_j}}) \\ & + \tilde{\mathrm{F}}(\mathbb{P}_{\mathrm{G}_{q_j}}, \mathbb{P}_{m_{q_j}}) \Big\} . \end{aligned} \tag{47}$$

Since $\mathcal{M}'_{q_i}$ stores the replayed samples from the generator, $\mathbb{P}_{m_{q_j}}$ does not represent the real training samples. The term $\mathrm{W}^{\star}_{\mathcal{L}}(\mathbb{P}_{\mathbf{x}^i}, \mathbb{P}_{m_{q_j}})$ would be enlarged if $\mathcal{M}'_{q_i}$ does not capture the information for the i-th target set. Additionally, CURL continually updates the generator's parameters in the whole training process, which would lead to forgetting prior information. The proposed dynamic expansion approach does not require generative replay and only updates the part of the whole network architecture when learning novel samples.

CN-DPM [14] is a dynamic expansion framework which introduces the Dirichlet process for the expansion of the network architecture. Different from CURL, CN-DPM does not use the generative replay and preserves the knowledge into the frozen components when building a new component. The forgetting behaviour of CN-DPM can be explained in Eq. (46). $\mathcal{M}'_{q_i}$ is implemented as a memory buffer which is used to train the i-th component for CN-DPM. To compare with CN-DPM, the proposed OCM employ a criterion which makes each $\{\mathcal{M}'_{q_i}; , i = 1, \cdots, k\}$ more diverse, which would lead to a better performance, empirically demonstrated in the results from the main paper.

F.2 Task labels are available

In this section, we apply the proposed theoretical framework for analyzing the forgetting behaviour of the model in a more general continual learning scenario where the task identity is provided during the training.

LGM [17] is a generative replay approach that trains a Teacher-Student framework where both the Teacher and Student are implemented by the VAE model. We first define the generative replay process of LGM.

Definition (**Generative replay for LGM.**) Let $\mathbb{P}^t_{\tilde{\mathbf{x}}}$ represent the distribution of samples drawn from the generator, which can be either the Teacher or Student, of LGM trained at the t-th task. Let $f_t \colon \mathcal{X} \to \mathcal{T}$ be the true labelling function that returns the task label for the data sample. Let $\mathbb{P}_{\tilde{\mathbf{x}}^{(i,m)}}$ be the distribution of samples drawn from the process $\mathbf{x} \sim \mathbb{P}^t_{\tilde{\mathbf{x}}}$ if $f_t(\mathbf{x}) = i$ where m represents that $\mathbb{P}_{\tilde{\mathbf{x}}^{(i,0)}}$ evolves to $\mathbb{P}_{\tilde{\mathbf{x}}^{(i,m)}}$ through m generative replay processes. Let $\mathbb{P}_{\tilde{\mathbf{x}}^{(i,0)}}$ and $\mathbb{P}_{\tilde{\mathbf{x}}^{(i,-1)}}$ represent $\mathbb{P}_{\mathbf{x}^i}$ and $\mathbb{P}_{\hat{\mathbf{x}}^i}$ for simplicity. Let $\mathbb{P}_{\mathrm{G}_i}$ represent the generator distribution of LGM trained at the i-th task.

Lemma 3. For a certain task (i), the bound on ELBO for LGM at the N-th task learning is defined as :

$$\begin{aligned}\mathbb{E}_{\mathbb{P}_{\hat{\mathbf{x}}^i}}[\mathcal{L}_{ELBO}(\mathbf{x};\theta,\omega)] \leq\ & \mathbb{E}_{\mathbb{P}_{\tilde{\mathbf{x}}^{(i,i-1)}}}[\mathcal{L}_{ELBO}(\mathbf{x};\theta,\omega)] \\ & + \sum_{s=0}^{N-i} \big\{ 2\mathrm{W}^{\star}_{\mathcal{L}}(\mathbb{P}_{\tilde{\mathbf{x}}^{(i,s)}}, \mathbb{P}_{\mathrm{G}_i}) + \tilde{\mathrm{F}}(\mathbb{P}_{\mathrm{G}_i}, \mathbb{P}_{\tilde{\mathbf{x}}^{(i,s)}}) \\ & - \mathrm{W}^{\star}_{\mathcal{L}}(\mathbb{P}_{\hat{\mathbf{x}}^{(i,s-1)}}, \mathbb{P}_{\tilde{\mathbf{x}}^{(i,s)}}) \big\}.\end{aligned} \tag{48}$$

The proof is similar to that for Theorem 3. Then we extend Lemma 3 to multiple tasks, as defined in the following Lemma 4.

Lemma 4. For a given set of disjoint tasks $\{\mathcal{T}_1, \cdots, \mathcal{T}_N\}$, the bound on ELBO for LGM is defined as :

$$\begin{aligned}\sum_{i=1}^{N} \Big\{ \mathbb{E}_{\mathbb{P}_{\hat{\mathbf{x}}^i}}[\mathcal{L}_{ELBO}(\mathbf{x};\theta,\omega)] \Big\} \leq\ & \sum_{i=1}^{N} \Big\{ \mathbb{E}_{\mathbb{P}_{\tilde{\mathbf{x}}^{(i,i-1)}}}[\mathcal{L}_{ELBO}(\mathbf{x};\theta,\omega)] \\ & + \sum_{s=0}^{N-i+1} \big\{ 2\mathrm{W}^{\star}_{\mathcal{L}}(\mathbb{P}_{\tilde{\mathbf{x}}^{(i,s)}}, \mathbb{P}_{\mathrm{G}_i}) + \tilde{\mathrm{F}}(\mathbb{P}_{\mathrm{G}_i}, \mathbb{P}_{\tilde{\mathbf{x}}^{(i,s)}}) \\ & - \mathrm{W}^{\star}_{\mathcal{L}}(\mathbb{P}_{\hat{\mathbf{x}}^{(i,s-1)}}, \mathbb{P}_{\tilde{\mathbf{x}}^{(i,s)}}) \big\} \Big\}.\end{aligned} \tag{49}$$

From Eq. (49), it can be observed that as progressing with learning more tasks, LGM's training would be affected from increasing forgetting because of the accumulated errors (the last term in the RHS of Eq. (49)) increase. Lemma 3 and Lemma 4 can describe the forgetting behaviour of most existing generative replay approaches including Lifelong VAEGAN [21]. In the following, we apply Theorem 3 of the paper to analyze the forgetting behaviour of dynamic expansion approaches.

LIMix [23] aims to learn an infinite mixture model by automatically expanding its network architecture when seeing a novel task while reusing a selected component to model a related work. The generative replay mechanism is used when LIMix chooses

an existing component for learning a new task. The bound on ELBO for LIMix is defined according to Theorem 3 of the paper.

$$\sum_{i=1}^{|\mathcal{A}|}\Big\{\mathbb{E}_{\mathbb{P}_{\tilde{\mathbf{x}}^{\tilde{a}_i}}}[\mathcal{L}_{ELBO}(\mathbf{x};\theta,\omega)]\Big\}+\sum_{i=1}^{|\mathcal{B}|}\Big\{\sum_{q=1}^{|\tilde{b}_i|}\big\{\mathbb{E}_{\mathbb{P}_{\tilde{\mathbf{x}}^{\tilde{b}_i^q}}}[\mathcal{L}_{ELBO}(\mathbf{x};\theta,\omega)]\big\}\Big\}\leq\mathcal{R}_S+\mathcal{R}_M \tag{50}$$

From Eq. (50), it can be observed that the number of components plays an important role in improving the performance of LIMix. For instance, the optimal performance can be achieved when the number of components in LIMix matches the number of tasks, meaning that there is no degenerated performance caused by the generative replay process.

Regularisation based approaches with episodic memory. Gradient Episodic Memory (GEM) [15] is a popular regularization approach that introduces using a small memory buffer to store a few past samples. In order to apply the proposed theoretical analysis for GEM, we assume we use GEM for training a VAE model instead of a classifier. Let $\mathbb{P}_{\mathrm{G}}$ represent the generator distribution of GEM and $\mathcal{M}_i$ represents a subset of the memory buffer, which stores the training samples from the i-th task. Since these stored samples represent the statistical information for each task, $\mathcal{M}_i$ can form an empirical distribution, denoted as $\mathbb{P}_{\tilde{\mathbf{x}}^i}$. The bound on ELBO for GEM can be defined as :

$$\begin{aligned}\sum_{i=1}^{N}\Big\{\mathbb{E}_{\mathbb{P}_{\tilde{\mathbf{x}}^i}}[\mathcal{L}_{ELBO}(\mathbf{x};\theta,\omega)]\Big\}\leq\sum_{i=1}^{N}\Big\{&\mathbb{E}_{\mathbb{P}_{\tilde{\mathbf{x}}^i}}[\mathcal{L}_{ELBO}(\mathbf{x};\theta,\omega)]\\&+2\mathrm{W}^{\star}_{\mathcal{L}}(\mathbb{P}_{\tilde{\mathbf{x}}^i},\mathbb{P}_{\mathrm{G}})-\mathrm{W}^{\star}_{\mathcal{L}}(\mathbb{P}_{\mathbf{x}^i},\mathbb{P}_{\tilde{\mathbf{x}}^i})\\&+\tilde{\mathrm{F}}(\mathbb{P}_{\mathrm{G}},\mathbb{P}_{\tilde{\mathbf{x}}^i})\Big\},\end{aligned} \tag{51}$$

From Eq. (51), it can be observed that the generalization performance of GEM is relying on the quality of stored samples. For instance, if each $\mathcal{M}_i$ captures the full information of the associated data distribution (the probabilistic representation) of the i-th task, then the term $-\mathrm{W}^{\star}_{\mathcal{L}}(\mathbb{P}_{\mathbf{x}^i},\mathbb{P}_{\tilde{\mathbf{x}}^i})$ is small and thus the ELBO on $\mathbb{P}_{\tilde{\mathbf{x}}^i}$ is close to the ELBO on the target domain $\mathbb{P}_{\mathbf{x}^i}$. Additionally, existing methods using episodic memory [5, 9] can also be explained by using Eq. (51) through which the algorithm finds the optimal updated direction in the parameter space by taking into account all subsets $\mathcal{M}_i$, $i=1,\ldots,t$. The other method [16], considers identifying the influential samples that are used to form the influential memory $\mathcal{M}_i$ corresponding to each task. Such influential memory data can reduce the computational complexity while improving the generalization performance on the target domains.

Mixture/Ensemble models. The other type of continual learning approaches are focused on deploying a multi-head framework in which a shared feature extractor is connected with several components and each component models a unique task only [24]. The advantage of such an approach is that it can achieve the optimal performance for each task since it does not lose the performance on previously learnt tasks when

learning novel tasks.

$$\sum_{i=1}^{N}\Big\{\mathbb{E}_{\mathbb{P}_{\hat{\mathbf{x}}^i}}[\mathcal{L}_{ELBO}(\mathbf{x};\theta,\omega)]\Big\} \leq \sum_{i=1}^{N}\Big\{\mathbb{E}_{\mathbb{P}_{\mathbf{x}^i}}[\mathcal{L}_{ELBO}(\mathbf{x};\theta,\omega)] + 2\mathrm{W}^{\star}_{\mathcal{L}}(\mathbb{P}_{\mathbf{x}^i},\mathbb{P}_{\mathrm{G}}) - \mathrm{W}^{\star}_{\mathcal{L}}(\mathbb{P}_{\mathbf{x}^i},\mathbb{P}_{\mathbf{x}^i}) + \mathrm{F}(\mathbb{P}_{\mathrm{G}},\mathbb{P}_{\mathbf{x}^i})\Big\}, \tag{52}$$

Since the shared feature extractor is only updated at the first task learning, learning a new task does not degenerate the performance on prior tasks. From Eq. (52), it can be observed that the generalization performance of the model is relying on the distance between the source domain and the target domain in each task.

F.3 Theoretical analysis when changing the order of tasks

The performance of a model on all testing sets would be affected by order in which the tasks are learned. This is usually caused by changing the number of generative replay processes for each task. Since each task has a different associated data complexity, the accumulated errors corresponding to each task are not equal. In this section, we apply the proposed theoretical framework for analyzing the model's performance when changing the order in which the tasks are learned.

Assumption 1 *We assume that each task has a different data complexity, in our case defined by the image representations over the entire database. The accumulated errors for each task are not equal even if using the same number of generative replay processes. The formulation for this assumption is defined as :*

$$\sum_{s=0}^{c}\big\{2\mathrm{W}^{\star}_{\mathcal{L}}(\mathbb{P}_{\tilde{\mathbf{x}}^{(q,s)}},\mathbb{P}_{\mathrm{G}^1}) - \mathrm{W}^{\star}_{\mathcal{L}}(\mathbb{P}_{\hat{\mathbf{x}}^{(q,s-1)}},\mathbb{P}_{\tilde{\mathbf{x}}^{(q,s)}}) + \tilde{\mathrm{F}}(\mathbb{P}_{\mathrm{G}^1},\mathbb{P}_{\tilde{\mathbf{x}}^{(q,s)}})\big\} \neq$$
$$\sum_{s=0}^{c}\big\{2\mathrm{W}^{\star}_{\mathcal{L}}(\mathbb{P}_{\tilde{\mathbf{x}}^{(d,s)}},\mathbb{P}_{\mathrm{G}^1}) - \mathrm{W}^{\star}_{\mathcal{L}}(\mathbb{P}_{\hat{\mathbf{x}}^{(d,s-1)}},\mathbb{P}_{\tilde{\mathbf{x}}^{(d,s)}}) + \tilde{\mathrm{F}}(\mathbb{P}_{\mathrm{G}^1},\mathbb{P}_{\tilde{\mathbf{x}}^{(d,s)}})\big\} \ \text{ for } \ \mathrm{d} \neq \mathrm{s}. \tag{53}$$

Lemma 5. For a given single VAE model, let $\mathcal{T} = \{\mathcal{T}_1, \cdots, \mathcal{T}_N\}$ be a task order where each $\mathcal{T}_i$ is associated with the dataset $\mathcal{D}_i^S$. When changing the order of $\mathcal{T}$, the performance of the model on all testing datasets is also changed.

Proof. Firstly, we derive the bound on ELBO for a single VAE model according to

Theorem 3 from the paper :

$$\sum_{q=1}^{N}\left\{\mathbb{E}_{\mathbb{P}_{\hat{\mathbf{x}}^q}}[\mathcal{L}_{ELBO}(\mathbf{x};\theta,\omega)]\right\} \leq \sum_{q=1}^{N}\Big\{\mathbb{E}_{\mathbb{P}_{\tilde{\mathbf{x}}^{(q,N-q)}}}[\mathcal{L}_{ELBO}(\mathbf{x};\theta,\omega)] + \sum_{s=0}^{N-q}\big\{2\mathrm{W}^{\star}_{\mathcal{L}}(\mathbb{P}_{\tilde{\mathbf{x}}^{(q,s)}},\mathbb{P}_{\mathrm{G}^1}) - \mathrm{W}^{\star}_{\mathcal{L}}(\mathbb{P}_{\hat{\mathbf{x}}^{(q,s-1)}},\mathbb{P}_{\tilde{\mathbf{x}}^{(q,s)}}) + \tilde{\mathrm{F}}(\mathbb{P}_{\mathrm{G}^1},\mathbb{P}_{\tilde{\mathbf{x}}^{(q,s)}})\big\}\Big\} \tag{54}$$

We name the right hand side of Eq. (54) as R_{risk1}. Let $\mathrm{F}_{order}(\mathcal{T}_i, j)$ be a function that changes the dataset corresponding to the task $\mathcal{T}_i$ with $\mathcal{D}_j^S$, but this function still returns the same task labels $\mathcal{T}_i$. Then we derive the risk bound for the case that the dataset $\mathcal{D}_i$ associated with $\mathcal{T}_i$ is changed by $\mathrm{F}_{order}(\mathcal{T}_i, j)$:

$$\sum_{q=1}^{N}\left\{\mathbb{E}_{\mathbb{P}_{\hat{\mathbf{x}}^q}}[\mathcal{L}_{ELBO}(\mathbf{x};\theta,\omega)]\right\} \leq \sum_{q=1}^{N}\Big\{\mathbb{E}_{\mathbb{P}_{\tilde{\mathbf{x}}^{(\mathrm{F}_{order}(q,N-q+1),N-q)}}}[\mathcal{L}_{ELBO}(\mathbf{x};\theta,\omega)] + \sum_{s=0}^{N-q}\big\{2\mathrm{W}^{\star}_{\mathcal{L}}(\mathbb{P}_{\tilde{\mathbf{x}}^{(\mathrm{F}_{order}(q,N-q),s)}},\mathbb{P}_{\mathrm{G}^1}) - \mathrm{W}^{\star}_{\mathcal{L}}(\mathbb{P}_{\hat{\mathbf{x}}^{(\mathrm{F}_{order}(q,N-q),s-1)}},\mathbb{P}_{\tilde{\mathbf{x}}^{(\mathrm{F}_{order}(q,N-q),s)}}) + \tilde{\mathrm{F}}(\mathbb{P}_{\mathrm{G}^1},\mathbb{P}_{\tilde{\mathbf{x}}^{(\mathrm{F}_{order}(q,N-q),s)}})\big\}\Big\} \tag{55}$$

We name the RHS of Eq. (55) as R_{risk2}. Since we use the function $\mathrm{F}_{order}(\cdot,\cdot)$ to change the associated dataset, which is the same with changing the learning order of the given tasks. We have $\mathrm{R}_{risk1} \neq \mathrm{R}_{risk2}$ because the the accumulated error terms satisfy Assumption 1.

F.4 Theoretical analysis for the importance weighted autoencoder

In this section, we extend the proposed theoretical framework for the importance weighted autoencoder (IWVAE) [4]. Firstly, we derive the bound on ELBO for IWVAE in a general continual learning setting where the task information is given.

Lemma 6. We assume that the task information is provided. For a given IWVAE model, the bound on ELBO for IWVAE is defined as :

$$\sum_{q=1}^{N}\left\{\mathbb{E}_{\mathbb{P}_{\hat{\mathbf{x}}^q}}[\mathcal{L}_{ELBO}(\mathbf{x};\theta,\omega)]\right\} \leq \sum_{q=1}^{N}\Big\{\mathbb{E}_{\mathbb{P}_{\tilde{\mathbf{x}}^{(q,N-q)}}}[\mathcal{L}^{m}_{IW}(\mathbf{x};\theta,\omega)] + \sum_{s=0}^{N-q}\big\{2\mathrm{W}^{\star}_{\mathcal{L}}(\mathbb{P}_{\tilde{\mathbf{x}}^{(q,s)}},\mathbb{P}_{\mathrm{G}}) - \mathrm{W}^{\star}_{\mathcal{L}}(\mathbb{P}_{\hat{\mathbf{x}}^{(q,s-1)}},\mathbb{P}_{\tilde{\mathbf{x}}^{(q,s)}}) + \tilde{\mathrm{F}}(\mathbb{P}_{\mathrm{G}},\mathbb{P}_{\tilde{\mathbf{x}}^{(q,s)}})\big\}\Big\}, \tag{56}$$

where $\mathbb{P}_{\mathrm{G}}$ represents the generator distribution of a IWVAE model. To compare with Eq. (31), Eq. (56) would improve the performance when the source distributions in Eq. (56) and Eq. (31) are equal. However, in practice, the source distributions would differ in each run and IWVAE bound can not guarantee better performance than ELBO.
Proof. Firstly, we have the IWVAE bound (See Eq.(2) of the paper) :

$$\mathcal{L}_{IW}^{m}(\mathbf{x};\theta,\omega) := \mathbb{E}_{z_1,\cdots,z_m\sim q_\omega(\mathbf{z}\,|\,\mathbf{x})}\left[\log\frac{1}{m}\sum_{i=1}^{m} w_i\right]. \tag{57}$$

Since $\mathcal{L}_{IW}^{m}(\mathbf{x};\theta,\omega) > \mathcal{L}_{ELBO}^{m}(\mathbf{x};\theta,\omega)$ for $m > 1$, we replace the first term in RHS of Eq, (31) by IWVAE bound, resulting in

$$\begin{aligned}\sum_{q=1}^{N}\left\{\mathbb{E}_{\mathbb{P}_{\tilde{\mathbf{x}}^q}}[\mathcal{L}_{ELBO}(\mathbf{x};\theta,\omega)]\right\} \leq \sum_{q=1}^{N}\Big\{\mathbb{E}_{\mathbb{P}_{\tilde{\mathbf{x}}^{(q,N-q)}}}[\mathcal{L}_{IW}^{m}(\mathbf{x};\theta,\omega)] + \sum_{s=0}^{N-q}\big\{2\mathrm{W}_{\mathcal{L}}^{\star}(\mathbb{P}_{\tilde{\mathbf{x}}^{(q,s)}},\mathbb{P}_{\mathrm{G}}) \\ - \mathrm{W}_{\mathcal{L}}^{\star}(\mathbb{P}_{\hat{\mathbf{x}}^{(q,s-1)}},\mathbb{P}_{\tilde{\mathbf{x}}^{(q,s)}}) + \tilde{\mathrm{F}}(\mathbb{P}_{\mathrm{G}},\mathbb{P}_{\tilde{\mathbf{x}}^{(q,s)}})\big\}\Big\},\end{aligned} \tag{58}$$

In the following, we analyze the forgetting behaviour of a IWVAE model under OCL.
Lemma 7. Let $\mathbb{P}_{m_i}$ and $\mathbb{P}_{\mathbf{x}}$ be the source and target domains. From Eq.(7) of the paper, we derive the bound on ELBO for a IWVAE model at the training step t_i :

$$\begin{aligned}\mathbb{E}_{\mathbb{P}_{\mathbf{x}}}[\mathcal{L}_{ELBO}(\mathbf{x};\theta,\omega)] \leq \mathbb{E}_{\mathbb{P}_{m_i}}[\mathcal{L}_{IW}(\mathbf{x};\theta,\omega)] \\ + 2\mathrm{W}_{\mathcal{L}}^{\star}(\mathbb{P}_{m_i},\mathbb{P}_{\mathrm{G}_i}) - \mathrm{W}_{\mathcal{L}}^{\star}(\mathbb{P}_{\mathbf{x}},\mathbb{P}_{m_i}) \\ + \widetilde{\mathrm{F}}(\mathbb{P}_{\mathrm{G}_i},\mathbb{P}_{m_i})\,.\end{aligned} \tag{59}$$

To compare with Eq.(8) of the paper, Eq. (59) can lead to better performance (RHS of Eq. (59) is larger than the RHS of Eq.(8) of the paper) when the source distribution for the VAE and IWVAE are equal to each other.
Proof. Because IWVAE bound is larger than ELBO for $m > 1$, we replace the first term in the RHS of Eq.(8) of the paper, which proves Lemma 7.

F.5 Theoretical analysis for lifelong VAEGAN

Lifelong VAEGAN [21] is one of the GRM based models, which combines GANs and VAEs into a unified framework. GAN in the Lifelong VAEGAN is used as the generative replay network, while additional inference models in lifelong VAEGAN are introduced to capture latent representations across domains over time. In this section, we can employ the proposed theoretical framework for analyzing the forgetting behaviour of Lifelong VAEGAN.

Lemma 8. Let $\mathbb{P}_{\mathrm{G}}$ represent the generator distribution of Lifelong VAEGAN. Then the bound on ELBO for lifelong VAEGAN when learning a sequence of N tasks is defined

as :

$$\sum_{q=1}^{N}\big\{\mathbb{E}_{\mathbb{P}_{\tilde{\mathbf{x}}^q}}[\mathcal{L}_{ELBO}(\mathbf{x};\theta,\omega)]\big\} \leq \sum_{q=1}^{N}\Big\{\mathbb{E}_{\mathbb{P}_{\tilde{\mathbf{x}}^{(q,N-q)}}}[\mathcal{L}_{ELBO}(\mathbf{x};\theta,\omega)] + \sum_{s=0}^{N-q}\big\{2\mathrm{W}^{\star}_{\mathcal{L}}(\mathbb{P}_{\tilde{\mathbf{x}}^{(q,s)}},\mathbb{P}_{\mathrm{G}}) - \mathrm{W}^{\star}_{\mathcal{L}}(\mathbb{P}_{\tilde{\mathbf{x}}^{(q,s-1)}},\mathbb{P}_{\tilde{\mathbf{x}}^{(q,s)}}) + \tilde{\mathrm{F}}(\mathbb{P}_{\mathrm{G}},\mathbb{P}_{\tilde{\mathbf{x}}^{(q,s)}})\big\}\Big\} \tag{60}$$

As shown from Eq. (60), Lifelong VAEGAN still suffers from degenerated performance when learning a growing number of tasks. Additionally, the performance of lifelong VAEGAN is unstable when changing the order of tasks (See details in Lemma 5). To compare with approaches that employ the VAE as the generative replay network, Lifelong VAEGAN would lead to better generalization performance because the GAN used as the generative replay network can produce high-quality generative replay samples.

G The algorithm for a single VAE and the dynamic expansion model

G.1 Additional details for a single VAE with OCM

The training process for a single VAE model with OCM, consists of three main stages and is illustrated in Fig. 1. We also provide the pseudo-code of a single VAE with OCM in Algorithm 1.

G.2 Additional details for the Dynamic Expansion Model (DEM) with OCM

In the following we consider the dynamic expansion model with OCM. We provide the detailed learning process for the dynamic expansion model with OCM in Fig. 3 where we evaluate the sample similarity by utilizing all learned encoders from the DEM. We also provide the pseudo-code of the dynamic expansion model with OCM in Algorithm 2.

Exploring the mixture of kernels used in the sample selection. In the proposed dynamic expansion mechanism, the sample selection combines the feature vector extracted by each trained inference model into an augmented feature vector which is used to calculate the sample similarity. However, as the number of components is growing, the dimension of the augmented feature vector will also increase infinitely. In order to address this issue We explore a new way to utilize the entire previously learned knowledge when making the sample selection. The procedure for evaluating the sample similarity between $\mathbf{x}^{e}_{(i,j)}$ and $\mathbf{x}^{l}_{(i,j)}$ by using a mixture of kernels is presented in Fig.2 where each learned inference model extracts the feature vectors from a pair of samples and then the kernel function is used to evaluate the sample similarity. Finally, we average all

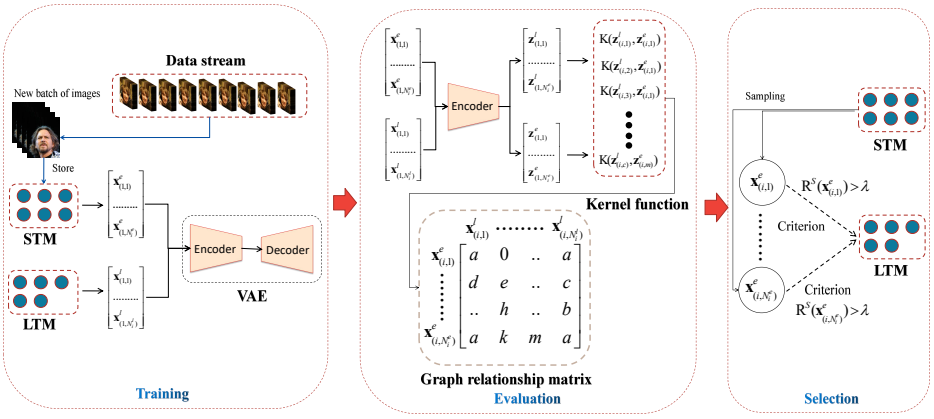


Figure 1: The detailed learning process of the proposed OCM when training a single VAE model. **(Learning.)** At a training step, STM stores a new batch of images while the model is trained to adapt both LTM and STM; If STM is full, we perform the evaluation and selection steps, otherwise, we continually perform the learning process at the next training step. **(Evaluation.)** We obtain feature vectors $\{\mathbf{z}^e_{(i,1)}, \cdots, \mathbf{z}^l_{(i,N_i^l)}\}$ from inputs $\{\mathbf{x}^e_{(i,1)}, \cdots, \mathbf{x}^l_{(i,N_i^l)}\}$ by using the encoder of a VAE model, which is used for the evaluation of the sample similarity using the given kernel (Eq.(16) from the paper). This similarity information is preserved in the graph relationship matrix $\mathbf{S}_i$. **(Selection.)** We transfer the samples from STM to LTM using the proposed criterion (Eq.(19) from the paper) using $\mathbf{S}_i$.

Algorithm 1 Training a single model with OCM

Input:$\mathcal{D}^S$ (Training dataset);

for $t_i < t_N$ **do**
  Step 1 (Learning:)
  $\mathbf{X}_b^i \sim \mathcal{D}^S$;
  $\mathbf{X}_b^i \in \mathcal{M}_i^e$;
  Train the model using samples from $\mathcal{M}_i^e$ and $\mathcal{M}_i^l$;
  if $\mathrm{Count}(\mathcal{M}_i^e) \geq \mathcal{M}_c^{Max}$ **then**
    Step 2 (Evaluation:)
    $\mathbf{S}_i = \mathrm{F}_{\exp}\Big(-(\mathbf{Z}_i^e(-1\mathbf{Z}_i^l)^{\mathrm{T}}) \odot (\mathbf{Z}_i^e(-1\mathbf{Z}_i^l)^{\mathrm{T}})/2\alpha^2\Big)$. Calculate the graph relationship matrix;
    Step 3 (Selection:)
    for $j < N_i^e$ **do**
      $\mathrm{R}^S(\mathbf{x}_{i,j}^e) = \frac{1}{N_i^l}\sum_{k=1}^{N_i^l}\mathbf{S}_i(j,k)$.; Calculate the average similarity score from $\mathbf{x}_{i,j}^e$ to LTM based on $\mathbf{S}_i$;
      if $\mathrm{R}^S(\mathbf{x}_{i,j}^e) > \lambda$ **then**
        $\mathcal{M}_i^l = \mathcal{M}_i^l \cup \mathbf{x}_{i,j}^e$; Add $\mathbf{x}_{i,j}^e$ into LTM memory;
      end if
    end for
    $\mathcal{M}_i^e = \emptyset$; Clear the STM memory;
  end if
end for

similarity scores calculated by each inference model as the sample similarity between $\mathbf{x}_{(i,j)}^e$ and $\mathbf{x}_{(i,j)}^l$.

We perform the experiment for the mixture of kernels used in the dynamic expansion model (DEM) under Split MNIST and the results are reported in Table 1, where "Dynamic-ELBO-OCM-MixKernel" represents the DEM using the mixture of kernels and the results show that the Dynamic-ELBO-OCM-MixKernel outperforms the Dynamic-ELBO-OCM while using more components. The performance of the Dynamic-ELBO-OCM-MixKernel on more tasks will be investigated in future work.

Methods	Log	Memory	N
Dynamic-ELBO-OCM	-115.89	1.1K	5
Dynamic-ELBO-OCM-MixKernel	-113.24	0.8K	10

Table 1: The results of the dynamic expansion model under Split MNIST.

G.3 Additional information for the motivation of using the kernel

The main motivation behind using the kernel as the criterion for the sample selection is summarized in two aspects. Firstly, the Gaussian kernel is a non-parametric way to measure the sample similarity while enjoying rich theoretical properties [8]. We apply

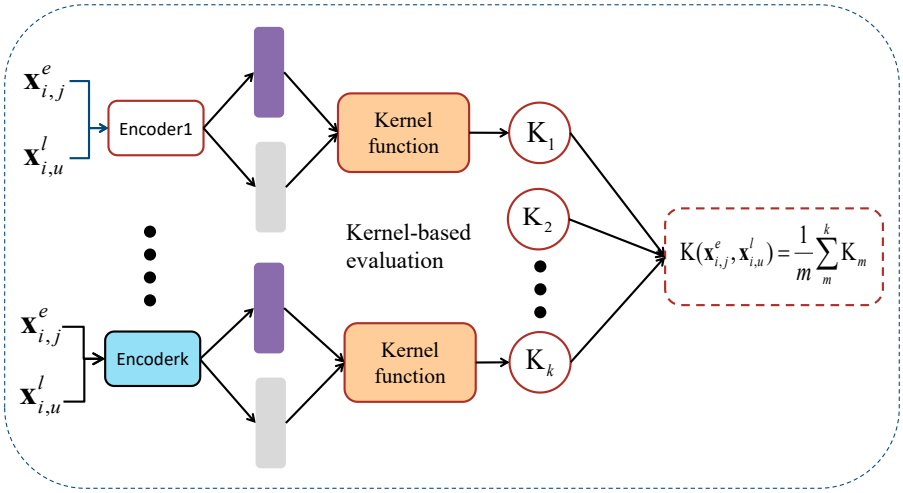


Figure 2: The mixture of kernels for the sample selection.

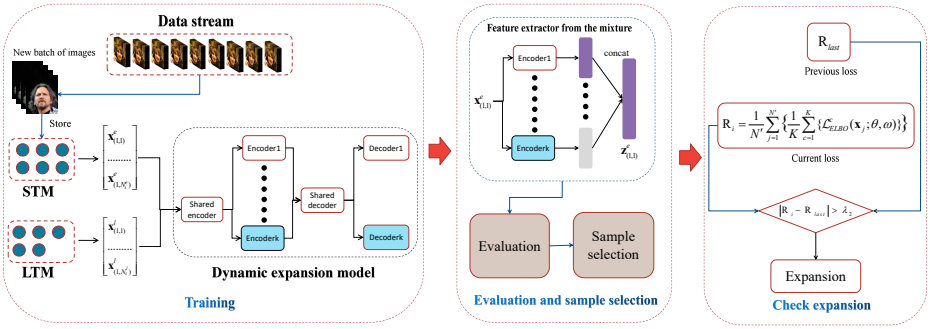


Figure 3: The learning process of the dynamic expansion model with OCM. We only update the current component (blue colour) at the learning step while fixing other modules to avoid forgetting. At the evaluation step, we use each learned encoder to extract the feature vector and then incorporate these feature vectors, resulting in a more expressive feature vector. Finally, we use the kernel function to calculate the sample similarity by using these expressive feature vectors at the sample selection step. We are also required to check the expansion at the sample selection stage.

Algorithm 2 Training a dynamic expansion model with OCM

Input:$\mathcal{D}^S$ (Training dataset);

for $t_i < t_N$ **do**
  Step 1 (Learning:)
  $\mathbf{X}_b^i \sim \mathcal{D}^S$;
  $\mathbf{X}_b^i \in \mathcal{M}_i^e$;
  Train the model using samples from $\mathcal{M}_i^e$ and $\mathcal{M}_i^l$;
  if $\mathrm{Count}(\mathcal{M}_i^e) \geq \mathcal{M}_c^{Max}$ **then**
    Step 2 (Evaluation:)
    $\mathbf{S}_i = \mathrm{F}_{\exp}\Big(-(\mathbf{Z}_i^e(-1\mathbf{Z}_i^l)^{\mathrm{T}}) \odot (\mathbf{Z}_i^e(-1\mathbf{Z}_i^l)^{\mathrm{T}})/2\alpha^2\Big)$. Calculate the graph relationship matrix;
    Step 3 (Selection:)
    for $j < N_i^e$ **do**
      $\mathrm{R}^S(\mathbf{x}_{i,j}^e) = \frac{1}{N_i^l}\sum_{k=1}^{N_i^l}\mathbf{S}_i(j,k)$.; Calculate the average similarity score from $\mathbf{x}_{i,j}^e$ to LTM based on $\mathbf{S}_i$;
      if $\mathrm{R}^S(\mathbf{x}_{i,j}^e) > \lambda$ **then**
        $\mathcal{M}_i^l = \mathcal{M}_i^l \cup \mathbf{x}_{i,j}^e$; Add $\mathbf{x}_{i,j}^e$ into LTM memory;
      end if
    end for
    Check the expansion:
    R_i Calculated by Eq.(22) from the paper;
    if $|\mathrm{R}_i - \mathrm{R}_{last}| > \lambda_2$ **then**
      Build a new component;
      $\mathcal{M}_i^l = \emptyset$; Clear the LTM memory;
      $\mathrm{R}_{last} = \mathrm{R}_i$;
    end if
    $\mathcal{M}_i^e = \emptyset$; Clear the STM memory;
  end if
end for

the kernel from Eq. (16) from the paper on the low dimensional feature space avoiding substantial computational costs. Secondly, to our best knowledge, this paper is the first to explore sample selection by the kernel-based criterion for the temporary memory under OCL. In a general continual learning [7] the kernel was shown to be effective for regression. However, the approach from [7] still requires both the task and class labels during the training. The proposed OCM does not require any supervised signals nor the task information during the training, and can be used in both supervised and unsupervised learning frameworks. Additionally, the kernel defines the operation on an inner product space, allowing us to evaluate the sample similarity more efficiently using the matrix operation as in Eq. (17) from the paper. Furthermore, we also explore other criteria for the sample selection and report the results in Section H.4. These results show that the kernel-based criterion performs well when compared to other measures.

G.4 Additional information for the motivation of considering OCM.

We provide the summary for the main motivation of the proposed OCM in two aspects. Firstly, existing approaches for OCL would usually use a single memory to store the most important data samples. However, such approached ignore the information about the future data streams given for training when making the sample selection. Additionally, they tend to have more computational costs than OCM because they perform the sample selection during each training step. The proposed OCM can overcome these two drawbacks by involving an STM memory that stores more recent samples. This STM can provide future information about the data stream, which benefits the sample selection, empirically demonstrated in Fig. 4a. Additionally, OCM only performs the sample selection when STM is full, which can significantly reduce the computational costs compared with existing approaches. Secondly, different from existing methods, the proposed OCM evaluates the sample similarity in the feature space using the kernel-based criterion. There are two advantages over existing approaches : 1) The sample selection in the proposed OCM does neither require the task information nor class labels, and consequently it can be used in unsupervised learning. 2) The sample selection in the proposed OCM does not rely on the loss function. This means that the proposed OCM does not require changing the selection criterion when it is used in a wide range of applications. Additionally, the proposed OCM can be used in any VAE variant with minimal modifications.

G.5 Additional information for the connection between the proposed OCM and the theoretical analysis

This section provides additional information for the connection between the proposed OCM and the theoretical analysis from the paper. Previous approaches have proposed to learn a diverse memory according to the category information. However, these approaches do not provide a theoretical guarantee for the accumulated memory's diversity. To our best knowledge, this paper is the first to provide theoretical guarantees and forgetting analysis for existing OCL models (See details in Section F). Additionally, the proposed theoretical framework demonstrates that the diversity of memory content can be achieved without knowing the category information (Lemma 1 of the paper). This motivates us to develop a new memory buffering approach that does not rely on the task information and class labels, which can be used in both supervised and unsupervised learning.

Furthermore, Lemma 2 of the paper shows that by considering the dynamic expansion mechanism, we improve the performance of the model over when considering a single VAE. The dynamic expansion mechanism reduces the negative transfer when each component learns different underlying data distributions (See detailed analysis in Appendix C). This theoretical analysis guides our dynamic expansion mechanism from two aspects. Firstly, we introduce a criterion to detect the data distribution shift by comparing the loss value between the previously learnt samples and the novel samples (Eq. (21) and (22) of the paper). Secondly, in order to encourage each component to learn different underlying data distributions, we would clear both STM and LTM when dynamically adding a new component to the mixture model. This can also avoid the

negative transfer by preserving the previously learnt knowledge into the frozen network structure, satisfying the conclusion of Lemma 2 of the paper (See detailed analysis in Appendix C).

H Additional information for the experimental configuration

The release of the code. We have provided the detailed implementation of the proposed Online Cooperative Memory (OCM) model. We will organize the source code of the OCM 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.1 Experiment setting

The hyperparameter configuration and GPU hardware. To perform the density estimation task, we use Adam [12] with a learning rate of 0.0001 and its default hyperparameters. To perform the generative modelling task, we use the Adam with a learning rate of 0.00005. We set the batch size and the number of epochs for each training step as 64 and 100, respectively. The GPU used for the experiments was GeForce GTX 1080. The operating system considered for experiments was Ubuntu 18.04.5.
The configuration of the network architecture for log-likelihood estimation task. We adapt the network architecture from [4] 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.
The configuration of the network architecture for the generative modelling task. The shared encoder is implemented using a fully connected network with three layers of processing with $[2000, 1500, 1000]$ units, and the component encoder uses a fully connected network with three layers of $[600, 300, 200]$ units. The shared decoder is implemented by a fully connected network with three layers $[200, 300, 600]$ and the component encoder is implemented by a fully connected network with three layers $[1000, 1500, 2000]$. The dimension of the latent variable is 200.
Hyperparameter setting. The batch size is of 64 images, and we consider 100 epochs for each training stage. The maximum memory size of STM is 0.5K and the optimal $\lambda = 0.6$ and $\lambda_2 = 10$ in Eq. (19) and (21) of the paper, respectively.
Additional information for the evaluation. All results reported in the paper are evaluated on the testing datasets after Online Continual Learning (OCL).
Additional information for Tiny-ImageNet under the generative modelling task. We divide Tiny-ImageNet into ten parts, Each part in Split Tiny-ImageNet has samples from 20 segregated categories.

H.2 The configuration for the classification task.

First, we introduce the details about the datasets used in our classification task as follows.

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 [6].

Split CIFAR10. We split CIFAR10 into five tasks where each task consists of samples from two different classes [6].

Split CIFAR100. We split CIFAR100 into 20 tasks where each task has 2500 examples from five different classes [15].

We adapt ResNet 18 [10] for Split CIFAR10 and Split CIFAR100. We use an MLP network with 2 hidden layers of 400 units each [6] for Split MNIST. The maximimum memory size (LTM + STM) for Split MNIST, Split CIFAR10, Split CIFAR100 are 2000, 1000 and 5000, respectively. For Dynamic-OCM, we build a new classifier when the proposed mixture model creates a new component where the classifier is trained on the labelled samples drawn from LTM and STM. For the inference process, the classifier with the associated selected component is used to make predictions for the given samples.

After the convergence following the training, the final number of components in Dynamic-OCM is 7, 8, 13 for Split MNIST, Split CIFAR10 and Split CIFAR100, respectively.

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) [15] 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.

Incremental Classifier and Representation Learning (iCARL) [19] is a standard memory-based method used in a class incremental setup.

reservoir* [20] 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 [1] 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 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 [2] 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.

H.3 Additional information for the reconstruction task

In this section, we investigate the performance of the reconstruction task when learning multiple domains. We create a data stream consisting of samples from Split CIFAR10, Split Tiny-ImageNet and CelebA, where the samples from CelebA are not ordered and we name this setting as CTC. The batch size is set to 64 for this setting and other hyperparameters are the same as for Split CIFAR10. The maximum number of components are restricted to not more than 20. We present the results in Table 2, where Inception Score (IS) and Fréchet Inception Distance (FID) are calculated on testing samples from Split CIFAR10 and Split Tiny-ImageNet in order to evaluate the reconstruction quality on the previously learnt datasets.

Methods	IS	FID	Memory	N
VAE-ELBO-Random	4.12	103.55	3K	1
CNDPM [14]	4.15	97.49	2K	20
LIMix [23]	3.84	129.32	2K	20
VAE-ELBO-OCM	4.25	89.40	2K	1
Dynamic-ELBO-OCM	4.36	80.25	1.3K	5

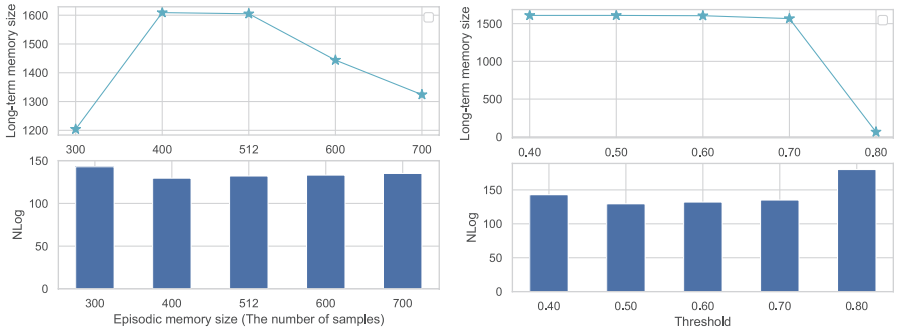
Table 2: IS and FID scores under CTC.

Configuration for ImageNet under OCL. We train a single VAE model with OCM on the ImageNet training set wherein each training step, we only access a batch of samples. We use β-VAE [11] loss for training and $\beta = 0.01$ relieves the over-regularization problem. In order to avoid growing LTM fast at the initial learning stage, the sample selection approach in this setting only chooses a single sample that has the largest distance from LTM and adds this sample into LTM. The number of training epochs for each training step t_i is 5. We adapt the network architecture from [22], the generator (decoder) consists of 6 convolution layers with $\{256, 256, 256, 256, 128, 3\}$ units. The inference model of the VAE consists of four convolution layers with $\{64, 128, 256, 512\}$ units, one hidden layer with $\{1024\}$ units and two separate layers with $\{256\}$ units each, which are used to output the hyperparameters of the Gaussian distribution. The dimension of the latent variable is 256.

H.4 Additional results for the ablation study

Ablation study for the hyperparameters. We firstly investigate the performance change when varying the size of STM and the threshold λ from Eq. (19) of the paper for a single VAE model under Split MNIST. The results are provided in Fig. 4 where the model faces degenerated performance when the size of STM is very small (300). This demonstrates that by condidering additional future samples can improve the performance. The results when changing the threshold λ are reported in Fig. 4b. These indicate that a large λ leads to smaller memory sizes and consequently a drop in the performance.

Ablation study for the dynamic expansion. Firstly, we investigate the performance of Dynamic-ELBO-OCM when changing the threshold $\lambda_2 = \{5, 10, 15, 20, 25, 30\}$ from eq. (21) from the paper, while $\lambda = 0.6$ from eq. (19) and the maximum memory



(a) Changing STM size. (b) Changing λ.

Figure 4: Assessment of the STM size and when changing the threshold λ for a single VAE model.

size for STM is of 500. From the results reported in Fig. 5 we can observe that by increasing λ_2 we can reduce the number of components. If λ_2 is very small, such as $\lambda_2 = 5$, the model has more components and consequently its performance would improve significantly.

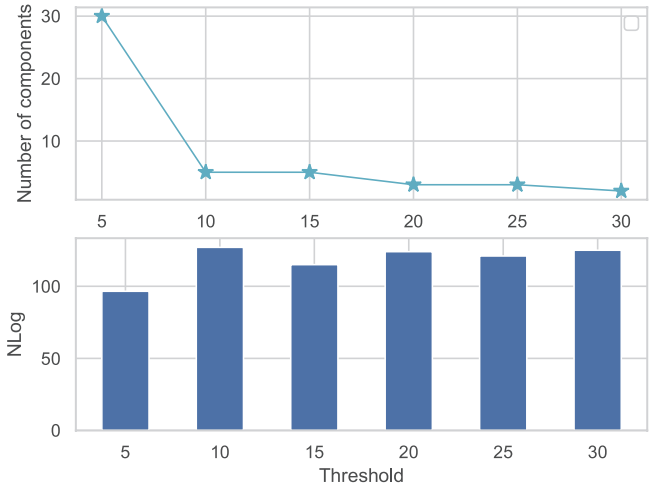


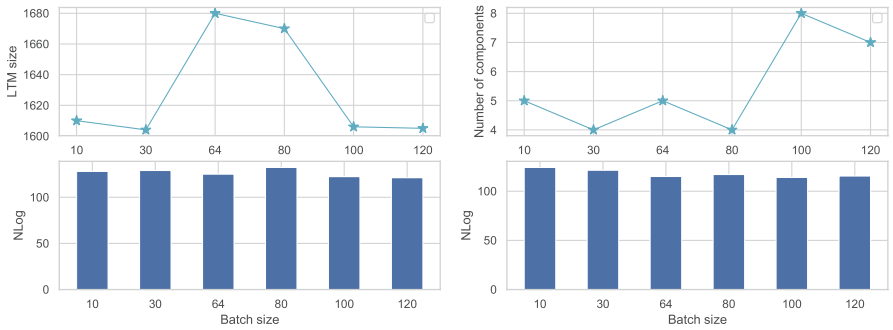
Figure 5: The performance when changing the threshold λ_2 for Dynamic-ELBO-OCM under Split MNIST.

Changing the batch size. We investigate the performance and memory change when varying the batch size. We consider batch sizes of 10, 30, 64, 80, 100, 120 for training a single VAE model with OCM under Split MNIST, and the results are reported in Fig. 7a. From this plot we can observe that the change of batch size has only a minor change in the LTM size and in the performance.

In the following, we evaluate the performance change of a dynamic expansion

model with OCM under Split MNIST with different batch sizes and the results are reported in Fig. 7b. It can be observed that the change of batch size does not have a significant influence, neither on the performance nor on the number of components. We present some memorized samples, from MNIST database and stored in the LTM in Fig. 6. The result shows that the proposed approach can encourage LTM to store diverse data samples during OCL.

Figure 6: Memorized samples draw from LTM.



(a) Single VAE with OCM. (b) Dynamic expansion model with OCM.

Figure 7: The change of LTM size and performance when training a single VAE model with OCM under Split MNIST with different batch sizes.

Unsupervised forward/backward transfer. To investigate the results for the proposed forward/backward transfer criteria, we train a VAE model that randomly selects samples, as a baseline. We calculate the backward transfer score for each training step by :

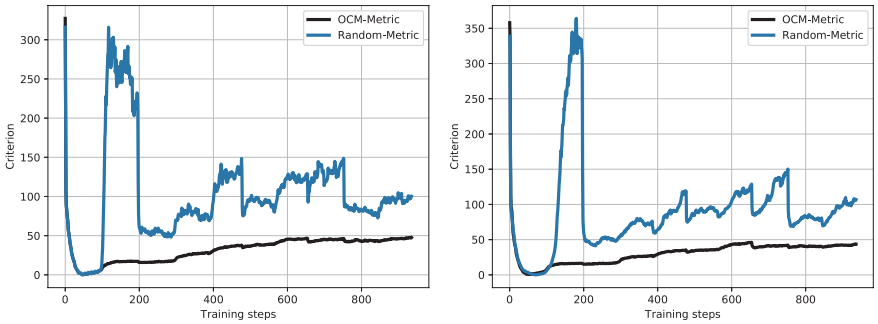
$$s_i = |\mathrm{E}_{(t_i,i)} - \mathrm{E}'_{t_i}| \tag{61}$$

where s_i represents the score calculated by the model at the training step t_i and other notations are defined in Section E. We also define the forward transfer score for each

training step by :

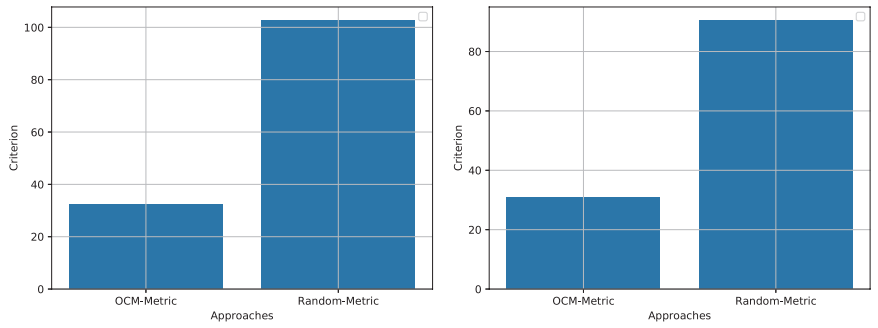
$$s'_i = |\mathrm{E}'_{t_{i+1}} - \mathrm{E}_{(t_i, i+1)}| \tag{62}$$

We provide the curve of the proposed forward/backward transfer criteria for both the baseline and a single VAE with the proposed OCM in Fig. 8 where "OCM-Metric" and "Random-Metric" represent the scores achieved by the baseline and the proposed approach, respectively. We can observe from these results that the proposed approach achieves better results than the random selection approach in each training step. We also report the results for the criteria (See BWT from Eq. (44)) and FWT from Eq. (45)) in Fig. 9. These results show that the proposed OCM outperforms the baseline by a large margin in both the forward and backward transfer criteria.



(a) Forward transfer results. (b) Backward transfer results.

Figure 8: Training curves for the proposed forward/backward transfer criteria, calculated for each training step under Split MNIST.



(a) Forward transfer results. (b) Backward transfer results.

Figure 9: The results for the proposed forward/backward transfer criteria from Eq. (44)) and Eq. (45), under Split MNIST.

The scale hyperparameter of the RBF kernel used in the proposed OCM. We investigate the performance of the proposed OCM framework when changing the hyperparameters of RBF kernel in Eq. (16) from the paper. We vary the hyperparameter $\alpha =$

$\{5, 10, 20, 30, 50, 70, 100\}$ for training a VAE model with OCM under Split MNIST, We present the results in Fig. 10. These results show that OCM with $\alpha = 10$ achieves the best results.

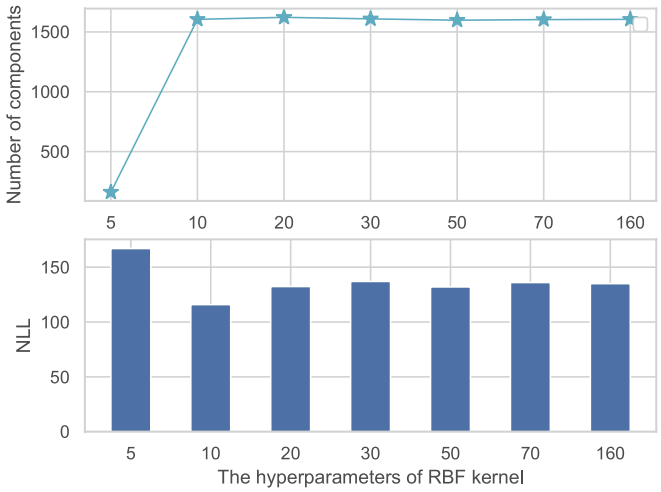


Figure 10: The results when varying the hyperparameter α for the RBF kernel from Eq. (16) from the paper, under Split MNIST.

The cosine distance used in the sample selection. We employ the cosine distance used for the proposed sample selection approach, defined as :

$$\mathrm{S}_C(\mathbf{x}^e_{i,j}, \mathbf{x}^l_{i,u}) := \frac{\mathbf{z}^e_{i,j} \cdot \mathbf{z}^l_{i,u}}{\left\|\mathbf{z}^e_{i,j}\right\| \left\|\mathbf{z}^l_{i,u}\right\|} = \frac{\sum_{i=1}^{d_z} \mathbf{z}^e_{i,j}(i)\mathbf{z}^l_{i,u}(i)}{\sqrt{\sum_{i=1}^{d_z} \left(\mathbf{z}^e_{i,j}(i)\right)^2} \sqrt{\sum_{i=1}^{d_z} \left(\mathbf{z}^l_{i,u}(i)\right)^2}} \tag{63}$$

We use "VAE-ELBO-OCM-COS" to represent a VAE model with OCM, where the cosine distance is used as the criterion for the sample selection. Since a small measure in Eq. (63) means that $\mathbf{x}^e_{i,j}$ is far away from $\mathbf{x}^l_{i,u}$, we modify Eq. (19) of the paper by :

$$\mathrm{R}^S(\mathbf{x}^e_{i,j}) < \lambda \Rightarrow \mathcal{M}^l_i = \mathcal{M}^l_i \cup \mathbf{x}^e_{i,j} \,. \tag{64}$$

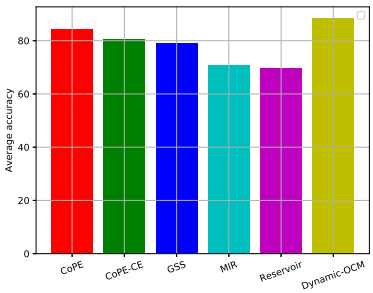
where λ is set to 0 in our experiment.

We report the results of various models under Split MNIST in Table 3. These results show that the kernel used as the criterion for the sample selection outperforms the cosine distance.

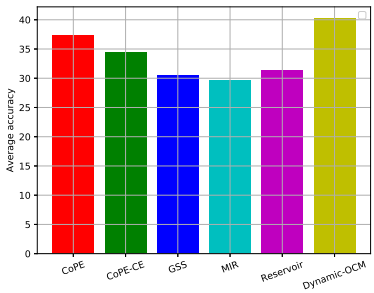
Imbalanced Benchmark Results. We follow the imbalanced data stream setting from [6], where several selected tasks have more samples while the remaining tasks have fewer data samples (See the detailed setting in [6]). The network architecture for the imbalanced benchmark is the same as for the balanced setting except for the Split

Methods	Log	Memory	N
VAE-ELBO-OCM-COS	-137.92	1.6K	1
VAE-ELBO-OCM	-132.07	1.6K	1
VAE-IWVAE50-OCM	-127.11	1.6K	1
Dynamic-ELBO-OCM	**-115.89**	1.1K	5

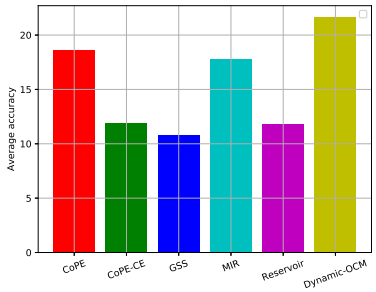
Table 3: The estimation of log-likelihood on all testing samples by using the IWVAE bound with 1000 importance samples.



(a) Split MNIST.



(b) Split CIFAR10.



(c) Split CIFAR100.

Figure 11: The results for the imbalanced benchmark where the results of baselines are cited from [6].

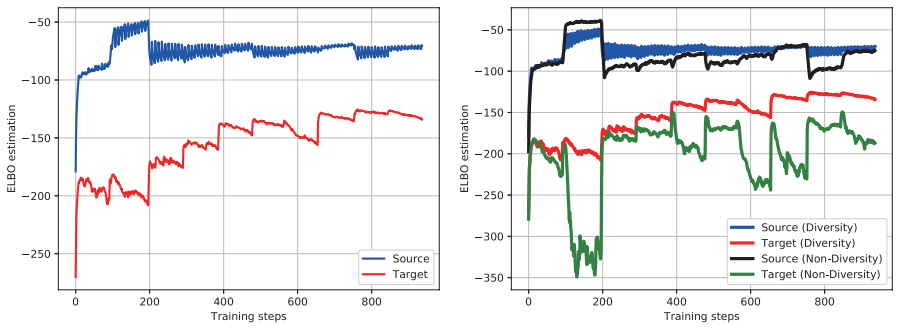
MNIST where we use an MLP with two hidden layers of [100 100] units, with the memory size of 3K.

We report the imbalanced benchmark results in Fig. 11 where the number of components for Split MNIST, Split CIFAR10 and Split CIFAR100 is 7,6 and 10, respectively. These results show that the proposed OCM with the dynamic expansion mechanism outperforms the state of the art approaches on the imbalanced data stream setting.

Why combine the OCM and dynamic expansion mechanism? The primary motivation of the combination between the OCM and the dynamic expansion mechanism is provided in Section G.5. Additionally, the empirical results (Table 1 and Table 2 of the paper) indicate that the proposed OCM with the dynamic expansion mechanism outperforms a single VAE model with OCM for all datasets. Furthermore, we also provide the empirical results of theoretical analysis to show the importance of the proposed dynamic expansion mechanism, according to the explanations from the following Section H.5.

H.5 Analysis of the theoretical results

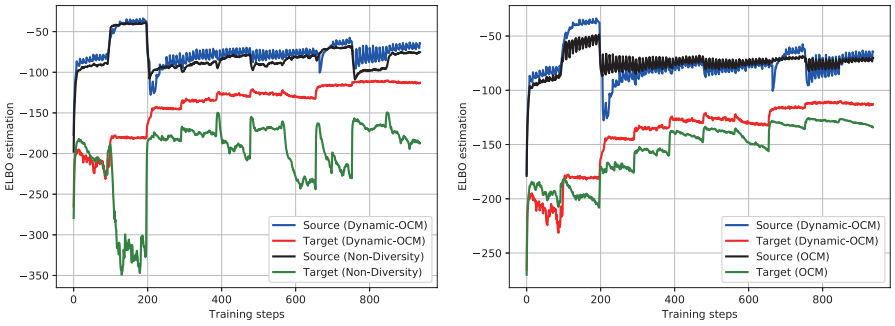
Theoretical results for a single VAE model. We train VAE-ELBO-OCM under Split MNIST and evaluate ELBO on $\mathbb{P}_{\mathbf{x}}$ (Training set) and $\mathcal{P}_{m_i}$ (Memory). We plot the results in Fig. 12 where "Target" and "Source" represent the ELBO evaluated on $\mathbb{P}_{\mathbf{x}}$ and $\mathbb{P}_{m_i}$, respectively. It can be observed that ELBO on $\mathbb{P}_{\mathbf{x}}$ is very small as the model is trained with few training steps and the model continually learns novel samples, as it is explained by Theorem 2 of the paper. We also investigate how the diversity can relieve forgetting, explained in Lemma 1. We train a baseline that only stores the more recent samples during the learning and we plot the result in Fig. 12b where "Source (Diversity)" represent ELBO on $\mathbb{P}_{m_i}$ evaluated by VAE-ELBO-OCM and "Target (Non-Diversity)" represent ELBO on $\mathbb{P}_{\mathbf{x}}$ evaluated by the baseline. We can observe that VAE-ELBO-OCM has rather stable performance according to its ELBO on $\mathbb{P}_{m_i}$, while the performance of the baseline on the target domain tends to degenerate as the training step increases, because of forgetting.



(a) ELBO on $\mathbb{P}_{m_i}$ and $\mathbb{P}_{\mathbf{x}}$. (b) Diversity analysis.

Figure 12: Empirical results for theoretical analysis.

The theoretical results of the dynamic expansion model. We also investigate the theoretical results of the dynamic expansion model. First, we train Dynamic-ELBO-OCM under Split MNIST where we evaluate ELBO on $\mathbb{P}_{\mathbf{x}}$ (Training set) and $\mathcal{P}_{m_i}$ (Memory), We also train a baseline that randomly selects samples during OCL. We then plot the results for each training step in Fig. 13a where "Target (Dynamic-OCM)" represents the ELBO evaluated on the target domain $\mathbb{P}_{\mathbf{x}}$, achieved by Dynamic-ELBO-OCM. The results show that although the baseline has a similar ELBO on the source domain (memory) with Dynamic-ELBO-OCM, the ELBO on the target domain, achieved by Dynamic-ELBO-OCM, is an upper bound for the ELBO on the target domain, performed by the baseline. This shows that a higher ELBO on the source domain can not guarantee good performance on the target domain, as demonstrated in **Remark** of **Theorem 2** of the paper. Additionally, we also compare the ELBO provided by Dynamic-ELBO-OCM and VAE-ELBO-OCM and plot the results in Fig. 13b where "Source (OCM)" represents the ELBO on the source domain, estimated by VAE-ELBO-OCM. It shows that ELBO on the target domain, obtained following the dynamic expansion mechanism, is still an upper bound to ELBO on the target domain, performed by a single model. This demonstrates that the dynamic expansion model can achieve a maximum upper bound to ELBO on the target domain when compared with a single model, as described in **Lemma 2** of the paper. Furthermore, as shown in Fig. 13b, Dynamic-ELBO-OCM achieves a similar ELBO on the target domain with VAE-ELBO-OCM at the initial training phase. However, Dynamic-ELBO-OCM gradually outperforms VAE-ELBO-OCM in the following training steps. This demonstrates that the proposed dynamic expansion mechanism can further relieve the negative transfer when compared with a single model, as discussed in Section C and Section G.5. The theoretical results for a general continual learning setting where the task information is given, will be investigated in our future studies.



(a) Diversity analysis

(b) The comparison between Dynamic expansion model and a single model.

Figure 13: Empirical results of theoretical analysis for the dynamic expansion model.

H.6 The model's complexity analysis

Since only CN-DPM [14] reports the number of parameters for the classification task under OCL, we provide the comparison on the number of parameters in Table 4. We can observe from this Table that the proposed approach outperforms CN-DPM while requiring fewer parameters.

Methods	Split MNIST	Split CIFAR10	Split CIFAR100
CN-DPM [14]	524K	4.60M	19.2M
Dynamic-OCM	519K	2.80M	17.87M

Table 4: The number of parameters for the classification task. The number of parameters for CN-DPM is reported in [14].

Time complexity analysis. To compare with the random selection approach, the proposed OCM would require a bit more computational cost in the evaluation step (See details in Eq.(16) of the paper). However, we can accelerate the computations of Eq.(16) of the paper by the matrix operation (Eq.(17) of the paper). Additionally, compared with existing sample selection approaches [2, 6, 3], the proposed OCM requires fewer computations since OCM has an STM to store the more recent samples to avoid frequently performing the sample selection (See Section 5.1 of the paper).

H.7 Visual results

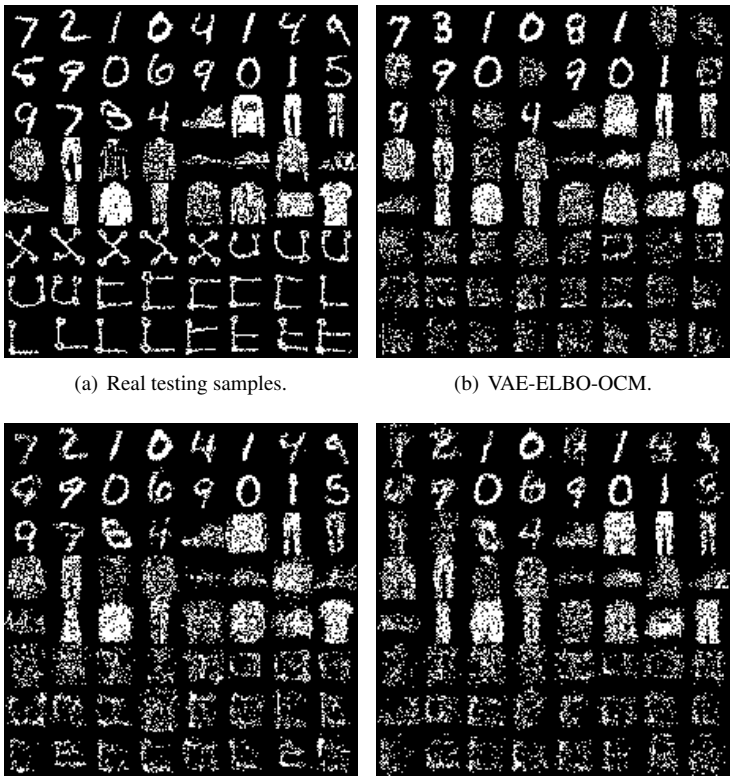
In this section we provide further visual results. In Fig. 14 we provide the generated results by the proposed method and for other methods for comparison. Testing samples and the reconstruction given by various models after the Cross-Domain setting for MNIST, Fashion and Omniglot are provided in Fig. 15. The reconstruction results for various models for CIFAR10 database are provided in Fig. 16 and on ImageNet [13] on Fig. 17.

(a) Real testing samples. (b) VAE-ELBO-OCM.

(c) Dynamic-ELBO-OCM. (d) VAE-ELBO-Random.

(e) LIMix. (f) CN-DPM.

Figure 14: Testing samples and the reconstruction given by various models after CTC.



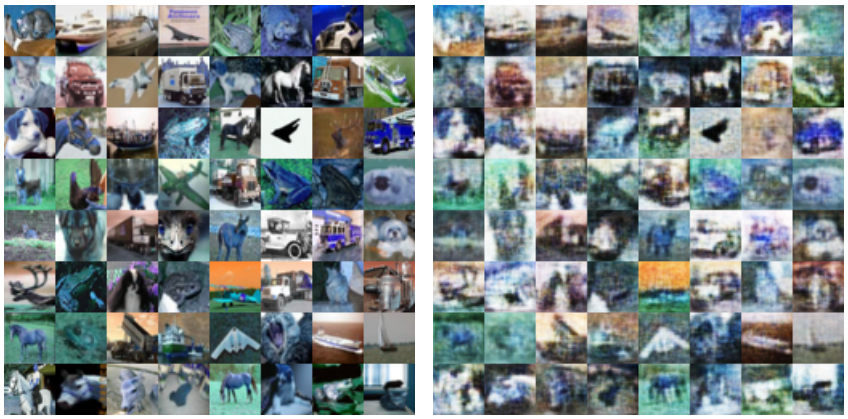
(a) Real testing samples.

(b) VAE-ELBO-OCM.

(c) Dynamic-ELBO-OCM.

(d) VAE-ELBO-Random.

Figure 15: Testing samples and the reconstruction given by various models after the Cross-Domain setting.



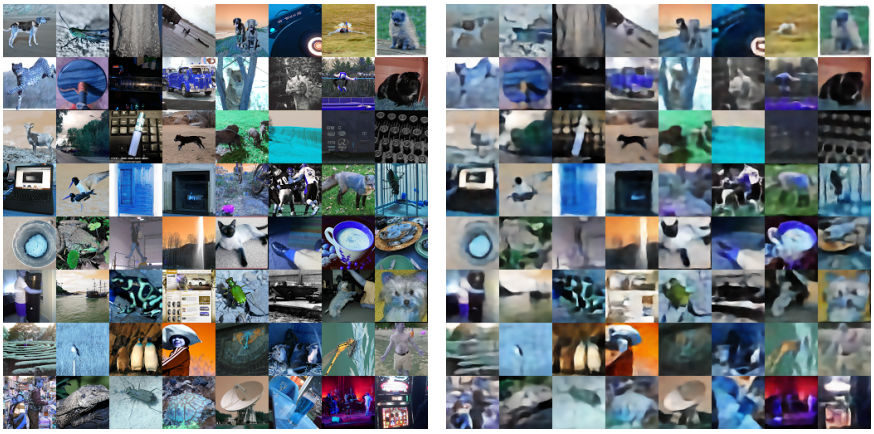
(a) Real testing samples.

(b) VAE-ELBO-OCM.

(c) Dynamic-ELBO-OCM.

(d) VAE-ELBO-Random.

Figure 16: Testing samples and the reconstruction given by various models after Split CIFAR10 setting.



(a) Real testing samples. (b) Reconstruction of VAE-ELBO-OCM

Figure 17: The real testing samples and their reconstruction on ImageNet after OCL.

I Limitations of the proposed OCM

In this section, we discuss the limitation of our work, which will be addressed in our future studies. Since we use the kernel-based diversity criterion for the sample selection in OCL, the choice of the kernels and hyperparameters is vital to decide the upper bound of the model's performance. Additionally, using other criteria for the sample selection for the proposed OCM framework would also be important in order to explore the full advantage of the proposed OCM framework. Therefore, we have investigated several kernel settings and other criteria in Appendix-H.4. More results will be investigated in our future studies.

References

[1] 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)*, volume 33, pages 11872–11883, 2019. 25

[2] R. Aljundi, M. Lin, B. Goujaud, and Y. Bengio. Gradient based sample selection for online continual learning. In *Advances Neural Information Processing Systems (NeurIPS)*, volume 33, pages 11817–11826, 2019. 25, 34

[3] Jihwan Bang, Heesu Kim, YoungJoon Yoo, Jung-Woo Ha, and Jonghyun Choi. Rainbow memory: Continual learning with a memory of diverse samples. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8218–8227, 2021. 34

[4] Yuri Burda, Roger Grosse, and Ruslan Salakhutdinov. Importance weighted autoencoders. *arXiv preprint arXiv:1509.00519*, 2015. 16, 24

[5] Arslan Chaudhry, Marc'Aurelio Ranzato, Marcus Rohrbach, and Mohamed Elhoseiny. Efficient lifelong learning with A-GEM. In *Int. Conf. on Learning Representations (ICLR), arXiv preprint arXiv:1812.00420*, 2019. 14

[6] Matthias De Lange and Tinne Tuytelaars. Continual prototype evolution: Learning online from non-stationary data streams. In *Proc. of the IEEE/CVF Int. Conference on Computer Vision (ICCV)*, pages 8250–8259, 2021. 25, 30, 31, 34

[7] Mohammad Mahdi Derakhshani, Xiantong Zhen, Ling Shao, and Cees Snoek. Kernel continual learning. In *International Conference on Machine Learning*, pages 2621–2631. PMLR, 2021. 22

[8] Pengfei Fang, Mehrtash Harandi, and Lars Petersson. Kernel methods in hyperbolic spaces. In *Proc. of the IEEE/CVF Int. Conference on Computer Vision (ICCV)*, pages 10665–10674, 2021. 20

[9] Yunhui Guo, Mingrui Liu, Tianbao Yang, and Tajana Rosing. Improved schemes for episodic memory-based lifelong learning. In *Advances in Neural Information Processing Systems*, pages 1023–1035, 2020. 14

[10] 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. 25

[11] I. Higgins, L. Matthey, A. Pal, C. Burgess, X. Glorot, M. Botvinick, S. Mohamed, and A. Lerchner. β-VAE: Learning basic visual concepts with a constrained variational framework. In *Proc. Int. Conf. on Learning Representations (ICLR)*, 2017. 26

[12] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *Proc. Int. Conf. on Learning Representations (ICLR), arXiv preprint arXiv:1412.6980*, 2015. 24

[13] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in Neural Inf. Proc. Systems (NIPS)*, pages 1097–1105, 2012. 34

[14] Soochan Lee, Junsoo Ha, Dongsu Zhang, and Gunhee Kim. A neural Dirichlet process mixture model for task-free continual learning. In *Proc. Int. Conf. on Learning Representations (ICLR), arXiv preprint arXiv:2001.00689*, 2020. 12, 26, 34

[15] David Lopez-Paz and Marc'Aurelio Ranzato. Gradient episodic memory for continual learning. In *Advances in Neural Information Processing Systems*, pages 6467–6476, 2017. 10, 14, 25

[16] Pingbo Pan, Siddharth Swaroop, Alexander Immer, Runa Eschenhagen, Richard Turner, and Mohammad Emtiyaz E Khan. Continual deep learning by functional regularisation of memorable past. In *Advances in Neural Information Processing Systems*, volume 33, pages 4453–4464, 2020. 14

[17] J. Ramapuram, M. Gregorova, and A. Kalousis. Lifelong generative modeling. In *Proc. Int. Conf. on Learning Representations (ICLR), arXiv preprint arXiv:1705.09847*, 2017. 13

[18] Dushyant Rao, Francesco Visin, Andrei A. Rusu, Yee Whye Teh, Razvan Pascanu, and Raia Hadsell. Continual unsupervised representation learning. In *Advances Neural Inf. Processing Systems (NeurIPS)*, pages 7645–7655, 2019. 12

[19] 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. 25

[20] Jeffrey S Vitter. Random sampling with a reservoir. *ACM Transactions on Mathematical Software (TOMS)*, 11(1):37–57, 1985. 25

[21] Fei Ye and Adrian G. Bors. Learning latent representations across multiple data domains using lifelong VAEGAN. In *Proc. European Conf. on Computer Vision (ECCV), vol. LNCS 12365*, pages 777–795, 2020. 11, 13, 17

[22] Fei Ye and Adrian G. Bors. Deep mixture generative autoencoders. *IEEE Transactions on Neural Networks and Learning Systems*, pages 1–15, 2021. 26

[23] Fei Ye and Adrian G. Bors. Lifelong infinite mixture model based on knowledge-driven Dirichlet process. In *Proc. of the IEEE Int. Conf. on Computer Vision (ICCV)*, 2021. 11, 13, 26

[24] Fei Ye and Adrian G. Bors. Lifelong mixture of variational autoencoders. *IEEE Transactions on Neural Networks and Learning Systems*, pages 1–14, 2021. 14