Appendix for Continual Variational Autoencoder via Continual Generative Knowledge Distillation

Anonymous Author(s)
Affiliation
Address
email

Contents

Submitted to 36th Conference on Neural Information Processing Systems (NeurIPS 2022). Do not distribute.

A Proof of Theorem 1

Our proof depends on the theoretical results from [17]. Firstly we assume that $\mathcal{X}$ is a metric space that satisfies $\mathcal{L}(a,b) \leq \mathcal{L}(a,c) + \mathcal{L}(c,b)$, where the loss function $\mathcal{L}(\cdot)$ is a metric and $a, b, c \in \mathcal{X}$.

Based on this assumption, we provide the detailed proof as follows.

Let $\mathbb{P}_{\mathbf{x}'(1:i)}$ and $\mathbb{P}_{\mathcal{M}_i}$ be two domains over $\mathcal{X}$. Let $h^{\star}_{\mathbb{P}_{\mathbf{x}'(1:i)}} = \arg\min_{h\in\mathcal{H}} \mathcal{E}_{\mathbb{P}_{\mathbf{x}'(1:i)}}(h, f_{\mathbb{P}_{\mathbf{x}'(1:i)}})$ and $h^{\star}_{\mathbb{P}_{\mathcal{M}_i}} = \arg\min_{h\in\mathcal{H}} \mathcal{E}_{\mathbb{P}_{\mathcal{M}_i}}(h, f_{\mathbb{P}_{\mathcal{M}_i}})$ represent the optimal models for $\mathbb{P}_{\mathbf{x}'(1:i)}$ and $\tilde{\mathcal{C}}_i$, respectively. $f_{\mathbb{P}_{\mathcal{M}_i}} \in \mathcal{H}$ denotes the identity function for $\mathbb{P}_{\mathcal{M}_i}$.

Based on the triangle inequality property of $\mathcal{L}$, we have :

$$\mathcal{E}_{\mathcal{C}_i}\big(h, f_{\mathbb{P}_{\mathbf{x}'(1:i)}}\big) \leq \mathcal{E}_{\mathcal{C}_i}\big(h, h^{*}_{\mathbb{P}_{\mathcal{M}_i}}\big) + \mathcal{E}_{\mathcal{C}_i}\big(h^{*}_{\mathbb{P}_{\mathcal{M}_i}}, h^{*}_{\mathbb{P}_{\mathbf{x}'(1:i)}}\big) + \mathcal{E}_{\mathcal{C}_i}\big(h^{*}_{\mathbb{P}_{\mathbf{x}'(1:i)}}, f_{\mathbb{P}_{\mathbf{x}'(1:i)}}\big) \tag{1}$$

Eq. (1) holds because, after applying twice the triangle inequality, $\mathcal{L}(a,b) \leq \mathcal{L}(a,c) + \mathcal{L}(c,d) + L(d,b)$ where a, b, c, d are $h(\mathbf{x})$, $f_{\mathbb{P}_{\mathbf{x}'(1:i)}}(\mathbf{x})$, $h^{*}_{\mathbb{P}_{\mathcal{M}_i}}(\mathbf{x})$, $h^{*}_{\mathbb{P}_{\mathbf{x}'(1:i)}}(\mathbf{x})$ and $\mathbf{x}$ is sampled from the same domain $\mathcal{C}_i$. According to the definition of discrepancy distance (See Definition 2 of the paper), we define the discrepancy distance between $\mathcal{C}_i$ and $\tilde{\mathcal{C}}_i$ as :

$$\mathcal{L}_{\text{disc}}\big(\mathbb{P}_{\mathbf{x}'(1:i)}, \mathbb{P}_{\mathcal{M}_i}\big) = \sup_{(h,h')\in\mathcal{H}} \Big| \mathbb{E}_{\mathbf{x}\sim\mathbb{P}_{\mathbf{x}'(1:i)}} \left[\mathcal{L}\left(h'\left(\mathbf{x}\right), h\left(\mathbf{x}\right)\right)\right] - \mathbb{E}_{\mathbf{x}\sim\mathbb{P}_{\mathcal{M}_i}} \left[\mathcal{L}\left(h'\left(\mathbf{x}\right), h\left(\mathbf{x}\right)\right)\right] \Big|. \tag{2}$$

We rewrite the above equation as :

$$\mathcal{L}_{\text{disc}}\big(\mathbb{P}_{\mathbf{x}'(1:i)}, \mathbb{P}_{\mathcal{M}_i}\big) = \sup_{(h,h')\in\mathcal{H}} \big| \mathcal{E}_{\mathbb{P}_{\mathbf{x}'(1:i)}}\left(h, h'\right) - \mathcal{E}_{\mathbb{P}_{\mathcal{M}_i}}\left(h, h'\right) \big|. \tag{3}$$

We consider h' to be $h^{*}_{\mathbb{P}_{\mathcal{M}_i}}$ in Eq. (3) and we have :

$$\sup_{(h,h')\in H} \Big| \mathcal{E}_{\mathbb{P}_{\mathbf{x}'(1:i)}}\left(h, h'\right) - \mathcal{E}_{\mathbb{P}_{\mathcal{M}_i}}\left(h, h'\right) \Big| \geq \Big| \mathcal{E}_{\mathbb{P}_{\mathbf{x}'(1:i)}}\big(h, h^{*}_{\mathbb{P}_{\mathcal{M}_i}}\big) - \mathcal{E}_{\mathbb{P}_{\mathcal{M}_i}}\big(h, h^{*}_{\mathbb{P}_{\mathcal{M}_i}}\big) \Big| \tag{4}$$

We also know that :

$$\mathcal{E}_{\mathbb{P}_{\mathbf{x}'(1:i)}}\big(h, h^{*}_{\mathbb{P}_{\mathcal{M}_i}}\big) - \mathcal{E}_{\mathbb{P}_{\mathcal{M}_i}}\big(h, h^{*}_{\mathbb{P}_{\mathcal{M}_i}}\big) \leq \Big| \mathcal{E}_{\mathbb{P}_{\mathbf{x}'(1:i)}}\big(h, h^{*}_{\mathbb{P}_{\mathcal{M}_i}}\big) - \mathcal{E}_{\mathbb{P}_{\mathcal{M}_i}}\big(h, h^{*}_{\mathbb{P}_{\mathcal{M}_i}}\big) \Big| \tag{5}$$

$$\mathcal{E}_{\mathbb{P}_{\mathbf{x}'(1:i)}}\big(h, h^{*}_{\mathbb{P}_{\mathcal{M}_i}}\big) \leq \Big| \mathcal{E}_{\mathbb{P}_{\mathbf{x}'(1:i)}}\big(h, h^{*}_{\mathbb{P}_{\mathcal{M}_i}}\big) - \mathcal{E}_{\mathbb{P}_{\mathcal{M}_i}}\big(h, h^{*}_{\mathbb{P}_{\mathcal{M}_i}}\big) \Big| + \mathcal{E}_{\mathbb{P}_{\mathcal{M}_i}}\big(h, h^{*}_{\mathbb{P}_{\mathcal{M}_i}}\big) \tag{6}$$

Therefore, we can replace the first term of the right hand side of Eq. (1) by the right hand side of Eq. (6), resulting in :

$$\begin{aligned} \mathcal{E}_{\mathbb{P}_{\mathbf{x}'(1:i)}}\Big(h, f_{\mathbb{P}_{\mathbf{x}'(1:i)}}\Big) \leq\ & \mathcal{E}_{\mathbb{P}_{\mathcal{M}_i}}\big(h, h^{*}_{\mathbb{P}_{\mathcal{M}_i}}\big) + \Big| \mathcal{E}_{\mathbb{P}_{\mathbf{x}'(1:i)}}\big(h, h^{*}_{\mathbb{P}_{\mathcal{M}_i}}\big) - \mathcal{E}_{\mathbb{P}_{\mathcal{M}_i}}\big(h, h^{*}_{\mathbb{P}_{\mathcal{M}_i}}\big) \Big| \\ & + \mathcal{E}_{\mathbb{P}_{\mathbf{x}'(1:i)}}\big(h^{*}_{\mathbb{P}_{\mathcal{M}_i}}, h^{*}_{\mathbb{P}_{\mathbf{x}'(1:i)}}\big) + \mathcal{E}_{\mathbb{P}_{\mathbf{x}'(1:i)}}\big(h^{*}_{\mathbb{P}_{\mathbf{x}'(1:i)}}, f_{\mathbb{P}_{\mathbf{x}'(1:i)}}\big) \end{aligned} \tag{7}$$

Then the second term, representing the absolute value of the difference, in RHS of Eq. (7) can be replaced by $\mathcal{L}_{\text{disc}}(\mathbb{P}_{\mathbf{x}'(1:i)}, \mathbb{P}_{\mathcal{M}_i})$ from Eq. (4)), since the discrepancy distance between two distributions is an upper bound to this absolute value, resulting in :

$$\begin{aligned} \mathcal{E}_{\mathbb{P}_{\mathbf{x}'(1:i)}}\big(h, f_{\mathbb{P}_{\mathbf{x}'(1:i)}}\big) \leq\ & \mathcal{E}_{\mathbb{P}_{\mathcal{M}_i}}\big(h, h^{*}_{\mathbb{P}_{\mathcal{M}_i}}\big) + \mathcal{L}_{\text{disc}}(\mathbb{P}_{\mathbf{x}'(1:i)}, \mathbb{P}_{\mathcal{M}_i}) \\ & + \mathcal{E}_{\mathbb{P}_{\mathbf{x}'(1:i)}}\big(h^{*}_{\mathbb{P}_{\mathcal{M}_i}}, h^{*}_{\mathbb{P}_{\mathbf{x}'(1:i)}}\big) + \mathcal{E}_{\mathbb{P}_{\mathbf{x}'(1:i)}}\big(h^{*}_{\mathbb{P}_{\mathbf{x}'(1:i)}}, f_{\mathbb{P}_{\mathbf{x}'(1:i)}}\big) \end{aligned} \tag{8}$$

In the following, we introduce how to estimate the discrepancy distance term using finite samples. First, we define the Rademacher complexity.

Definition 4 (Rademacher complexity). Let $\mathcal{H}$ represent a hypothesis class, For a given unlabeled sample $U = \{\mathbf{x}_i\}_{i=1}^m$, the Rademacher complexity of $\mathcal{H}$ with respect to the sample U is defined as follows :

$$\mathrm{Re}_U(\mathcal{H}) = \mathop{\mathbb{E}}_{\mathcal{K}}\left[\sup_{h\in\mathcal{H}} \frac{2}{m}\sum_{i=1}^{m} \mathcal{K}_i h(\mathbf{x}_i)\right] \tag{9}$$

where $\mathcal{K}_i$ is an independent uniform random variable within $\{-1, +1\}$. The Rademacher complexity for the whole hypothesis class is defined as :

$$\mathrm{Re}_n(\mathcal{H}) = \mathbb{E}_{U\sim(D)^n}\mathrm{Re}_U(\mathcal{H}) \tag{10}$$

Then we derive $\widehat{\mathcal{L}}_{\mathrm{disc}}(\cdot)$ (see Definition 3 of the paper) based on the Rademacher complexity.

From **Definition 3** from the paper, we know that $\mathcal{L}_{\mathrm{disc}}(\mathbb{P}_{\mathbf{x}'(1:i)}, \mathbb{P}_{\mathcal{M}_i}) \leq \widehat{\mathcal{L}}_{\mathrm{disc}}(\mathbb{P}_{\mathbf{x}'(1:i)}, \mathbb{P}_{\mathcal{M}_i})$. This allows use to replace $\mathcal{L}_{\mathrm{disc}}(\mathbb{P}_{\mathbf{x}'(1:i)}, \mathbb{P}_{\mathcal{M}_i})$ by using $\widehat{\mathcal{L}}_{\mathrm{disc}}(\mathcal{C}_i, \mathbb{P}_{\mathcal{M}_i})$ in Eq. (8), resulting in :

$$\begin{aligned}\mathcal{E}_{\mathbb{P}_{\mathbf{x}'(1:i)}}\left(h, f_{\mathbb{P}_{\mathbf{x}'(1:i)}}\right) \leq\ & \mathcal{E}_{\mathbb{P}_{\mathcal{M}_i}}\left(h, h^*_{\mathbb{P}_{\mathcal{M}_i}}\right) + \widehat{\mathcal{L}}_{\mathrm{disc}}\left(\mathbb{P}_{\mathbf{x}'(1:i)}, \mathbb{P}_{\mathcal{M}_i}\right) \\ & + \mathcal{E}_{\mathcal{C}_i}\left(h^*_{\mathbb{P}_{\mathcal{M}_i}}, h^*_{\mathbb{P}_{\mathbf{x}'(1:i)}}\right) + \mathcal{E}_{\mathbb{P}_{\mathbf{x}'(1:i)}}\left(h^*_{\mathbb{P}_{\mathbf{x}'(1:i)}}, f_{\mathbb{P}_{\mathbf{x}'(1:i)}}\right)\end{aligned} \tag{11}$$

The proof of Eq. (11) can be found in found in Theorem 1 from [17].

Based on the results from Eq. (11), we have the following proof procedure. According to the bound on the KL divergence :

$$\begin{aligned}\mathbb{E}_{\mathbb{P}_{\mathbf{x}'(1:i)}} KL(q_{\phi^i}(\mathbf{z} \mid \mathbf{x}'(1:i)) \mid\mid p(\mathbf{z})) \leq\ & \mathbb{E}_{\mathbb{P}_{\mathcal{M}_i}} KL(q_{\phi^i}(\mathbf{z} \mid \mathbf{x}_{\mathcal{M}_i}) \mid\mid p(\mathbf{z})) \\ & + |\mathbb{E}_{\mathbb{P}_{\mathcal{M}_i}} KL(q_{\phi^i}(\mathbf{z} \mid \mathbf{x}_{\mathcal{M}_i}) \mid\mid p(\mathbf{z})) \\ & - \mathbb{E}_{\mathbb{P}_{\mathbf{x}'(1:i)}} KL(q_{\phi^i}(\mathbf{z} \mid \mathbf{x}'(1:i)) \mid\mid p(\mathbf{z}))|\end{aligned} \tag{12}$$

We also know that $\mathcal{L}_{ELBO}(\mathbf{x}; \{\theta, \phi\})$ is expressed as :

$$\mathcal{L}_{ELBO}(\mathbf{x}; \{\theta, \phi\}) := \mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})}\left[\log p_\theta(\mathbf{x} \mid \mathbf{z})\right] - KL\left[q_\phi(\mathbf{z} \mid \mathbf{x}) \mid\mid p(\mathbf{z})\right], \tag{13}$$

When the decoder models a Gaussian distribution, $\log p_\theta(\mathbf{x} \mid \mathbf{z})$ can be represented as :

$$\log p_\theta(\mathbf{x} \mid \mathbf{z}) = -\frac{1}{2\sigma_\theta^2(\mathbf{z})}\|\mathbf{x} - \mu_\theta(\mathbf{z})\|^2 - \frac{1}{2}\log 2\pi\sigma_\theta^2(\mathbf{z}) \tag{14}$$

where $\sigma_\theta(\mathbf{z})$ and $\mu_\theta(\mathbf{z})$ are the variance and mean of Gaussian distribution, obtained by the decoder. $\|\cdot\|^2$ represents the reconstruction error (square loss). We implement the decoder by a Gaussian distribution with the fixed variance $\mathcal{N}(\mu_\theta(\mathbf{z}), \sigma\mathbf{I})$ where $\mu_\theta(\mathbf{z})$ is a deep convolutional neural network and $\mathbf{I}$ is the identity matrix. Therefore, Eq. (14) is represented by the fixed variance σ :

$$\log p_\theta(\mathbf{x} \mid \mathbf{z}) = -\frac{1}{2\sigma^2}\|\mathbf{x} - \mu_\theta(\mathbf{z})\|^2 - \frac{1}{2}\log 2\pi\sigma^2 \tag{15}$$

Since h is the hypothesis of the model, implemented as an encoding-decoding process, we have

$$\begin{aligned}\mathcal{L}_{ELBO}(\mathbf{x}'(1:i); h) = & -\frac{1}{2\sigma^2}\mathcal{L}(h(\mathbf{x}'(1:i)), f_{\mathbb{P}_{\mathbf{x}'(1:i)}}(\mathbf{x}'(1:i))) \\ & - \frac{1}{2}\log 2\pi\sigma^2 - KL(q_{\phi^i}(\mathbf{z} \mid \mathbf{x}'(1:i)) \mid\mid p(\mathbf{z}))\end{aligned} \tag{16}$$

Then we focus on the negative ELBO :

$$\begin{aligned}-\mathcal{L}_{ELBO}(\mathbf{x}'(1:i); h) = & \frac{1}{2\sigma^2}\mathcal{L}(h(\mathbf{x}'(1:i)), f_{\mathbb{P}_{\mathbf{x}'(1:i)}}(\mathbf{x}'(1:i))) \\ & + \frac{1}{2}\log 2\pi\sigma^2 + KL(q_{\phi^i}(\mathbf{z} \mid \mathbf{x}'(1:i)) \mid\mid p(\mathbf{z}))\end{aligned} \tag{17}$$

And we know that $\mathcal{R}_{\mathbb{P}_{\mathbf{x}'(1:i)}}(h, f_{\mathbb{P}_{\mathbf{x}'(1:i)}}) = \mathbb{E}_{\mathbf{x}\sim\mathbb{P}_{\mathbf{x}'(1:i)}}\mathcal{L}(h(\mathbf{x}'(1:i)), f_{\mathbb{P}_{\mathbf{x}'(1:i)}}(\mathbf{x}'(1:i)))$ and we have :

$$\mathbb{E}_{\mathbf{x}\sim\mathbb{P}_{\mathbf{x}'(1:i)}}\left[-\mathcal{L}_{ELBO}(\mathbf{x}'(1:i);h)\right] = \mathbb{E}_{\mathbf{x}'(1:i)\sim\mathbb{P}_{\mathbf{x}'(1:i)}}\left\{\frac{1}{2\sigma^2}\mathcal{L}(h(\mathbf{x}'(1:i)), f_{\mathbb{P}_{\mathbf{x}'(1:i)}}(\mathbf{x}'(1:i))) + KL(q_{\phi^i}(\mathbf{z}\mid\mathbf{x}'(1:i))\mid\mid p(\mathbf{z}))\right\} + \frac{1}{2}\log 2\pi\sigma^2 \tag{18}$$

We observe that $\frac{1}{2}\log 2\pi\sigma^2$ and $\frac{1}{2\sigma^2}$ are constants. In order to simplify the notations, we set $\sigma = \frac{1}{\sqrt{2}}$. Therefore, Eq. (18) is rewritten as :

$$\mathbb{E}_{\mathbf{x}'(1:i)\sim\mathbb{P}_{\mathbf{x}'(1:i)}}\left[-\mathcal{L}_{ELBO}(\mathbf{x}'(1:i);h)\right] = \mathbb{E}_{\mathbf{x}'(1:i)\sim\mathbb{P}_{\mathbf{x}'(1:i)}}\{\mathcal{L}(h(\mathbf{x}'(1:i)), f_{\mathbb{P}_{\mathbf{x}'(1:i)}}(\mathbf{x}'(1:i))) + KL(q_{\phi^i}(\mathbf{z}\mid\mathbf{x}'(1:i))\mid\mid p(\mathbf{z}))\} + \frac{1}{2}\log\pi \tag{19}$$

We then combine Equations (12) and (11) and we have :

$$\begin{aligned}&\mathbb{E}_{\mathbf{x}'(1:i)\sim\mathbb{P}_{\mathbf{x}'(1:i)}}\{\mathcal{L}(h(\mathbf{x}'(1:i)), f_{\mathbb{P}_{\mathbf{x}'(1:i)}}(\mathbf{x}'(1:i))) + KL(q_{\phi^i}(\mathbf{z}|\mathbf{x}'(1:i))\mid\mid p(\mathbf{z}))\} \leq \\ &\mathbb{E}_{\mathbf{x}_{\mathcal{M}_i}\sim\mathbb{P}_{\mathcal{M}_i}}\left\{\mathcal{L}\left(h(\mathbf{x}_{\mathcal{M}_i}), h^*_{\mathbb{P}_{\mathcal{M}_i}}(\mathbf{x}_{\mathcal{M}_i})\right) + KL(q_{\phi^i}(\mathbf{z}\mid\mathbf{x}_{\mathcal{M}_i})\mid\mid p(\mathbf{z}))\right\} \\ &+ |KL_1 - KL_2| + \mathcal{E}_A\left(\mathbb{P}_{\mathbf{x}'(1:i)}, \mathbb{P}_{\mathcal{M}_i}\right),\end{aligned} \tag{20}$$

where KL_1 and KL_2 are defined in Eq. (14) from the paper. We assume that $h^*_{\mathbb{P}_{\mathcal{M}_i}}(\mathbf{x}_{\mathcal{M}_i})$ is a perfect model for $\mathbb{P}_{\mathcal{M}_i}$. Then we have $h^*_{\mathbb{P}_{\mathcal{M}_i}}(\mathbf{x}_{\mathcal{M}_i}) = f^*_{\mathbb{P}_{\mathcal{M}_i}}(\mathbf{x}_{\mathcal{M}_i})$. It notes that we can add the constant $\frac{1}{2}\log\pi$ in both sides of Eq. (20). According to Eq. (19), we can rewrite Eq. (20) as :

$$\begin{aligned}\frac{1}{t}\sum_{i=1}^{t}\mathbb{E}_{\mathbf{x}'(1:i)\sim\mathbb{P}_{\mathbf{x}'(1:i)}}[-\mathcal{L}_{ELBO}\left(\mathbf{x}'(1:i);h\right)] &\leq \mathbb{E}_{\mathbf{x}_{\mathcal{M}_i}\sim\mathbb{P}_{\mathcal{M}_i}}\left[-\mathcal{L}_{ELBO}\left(\mathbf{x}_{\mathcal{M}_i};h\right)\right] \\ &+ |KL_1 - KL_2| + \mathcal{E}_A\left(\mathbb{P}_{\mathbf{x}'(1:i)}, \mathbb{P}_{\mathcal{M}_i}\right)\end{aligned} \tag{21}$$

□

This corresponds to Eq. (12) and proves Theorem 1 from the paper.

B Lemma 1

In the following, we analyse the generalization performance of a single model on the test sets.

Lemma 1. *Suppose that the data stream to learn has involved t different data domains. Let $\{D_1^T, \cdots, D_t^T\}$ be the testing sets and $\mathbb{P}_{D_j}$ be the probability distribution of the statistical representation for D_j^T. Let $\mathbb{P}_{D_{(1:t)}}$ denote the distribution for $\{D_1^T, \cdots, D_t^T\}$. Then, we can derive a risk bound to analyze the generalization performance of a single model trained on $\mathcal{M}_i$ at $\mathcal{T}_i$:*

$$\begin{aligned}\sum_{j=1}^{t}\left\{\mathbb{E}_{\mathbb{P}_{D_j}}\left[-\mathcal{L}_{ELBO}\big(\mathbf{x}_{D_j};h\big)\right]\right\} &\leq \mathcal{E}_A\big(\mathbb{P}_{D_{(1:t)}}, \mathbb{P}_{\mathcal{M}_i}\big) \\ &+ \mathbb{E}_{\mathbb{P}_{\mathcal{M}_i}}\left[-\mathcal{L}_{ELBO}\big(\mathbf{x}_{\mathcal{M}_i};h\big)\right] + \left|KL_{D_{(1:t)}} - KL_2\right|,\end{aligned} \tag{22}$$

At the beginning of training, the model h gradually gains knowledge and improves its generalization performance while increasing the number of training steps, given that the memory $\mathcal{M}_i$ can store all previously seen samples. However, after a while, given the amount of data to be learnt, a single model cannot guarantee an optimal performance for the target distribution. This is because a given fixed-capacity memory cannot capture the full information from an ever increasing stream of training data samples.

Proof. First, we have the following inequality :

$$\frac{1}{t}\sum_{j=1}^{t}\Big\{\mathbb{E}_{\mathbb{P}_{D_j}} KL(q_{\phi^i}(\mathbf{z} \mid \mathbf{x}_{D_j}) \mid\mid p(\mathbf{z}))\Big\} \leq \mathbb{E}_{\mathbb{P}_{\mathcal{M}_i}} KL(q_{\phi^i}(\mathbf{z} \mid \mathbf{x}_{\mathcal{M}_i}) \mid\mid p(\mathbf{z})) + \left|\frac{1}{t}\sum_{j=1}^{t}\mathbb{E}_{\mathbb{P}_{D_j}} KL(q_{\phi^i}(\mathbf{z} \mid \mathbf{x}_{D_j}) \mid\mid p(\mathbf{z})) - \mathbb{E}_{\mathbb{P}_{\mathcal{M}_i}} KL(q_{\phi^i}(\mathbf{z} \mid \mathbf{x}_{\mathcal{M}_i}) \mid\mid p(\mathbf{z}))\right| \tag{23}$$

We use $KL_{D_{(1:t)}}$ to represent the Left-Hand Side (LHS) of Eq. (23). We then take Eq. (23) in Eq. (21) from Theorem 1, where the target distribution $\mathbb{P}_{\mathbf{x}'(1:i)}$ is replaced by $\mathbb{P}_{D_j}$, and then we have :

$$\begin{aligned}&\frac{1}{t}\sum_{j=1}^{t}\Big\{\mathbb{E}_{\mathbf{x}_{D_j}\sim\mathbb{P}_{D_j}}\{\mathcal{L}(h(\mathbf{x}_{D_j}), f_{\mathbb{P}_{D_j}}(\mathbf{x}_{D_j})) + KL(q_{\phi^i}(\mathbf{z}|\mathbf{x}_{D_j}) \mid\mid p(\mathbf{z}))\}\Big\} \leq \\ &\mathbb{E}_{\mathbf{x}_{\mathcal{M}_i}\sim\mathbb{P}_{\mathcal{M}_i}}\Big\{\mathcal{L}\left(h(\mathbf{x}_{\mathcal{M}_i}), h^*_{\mathbb{P}_{\mathcal{M}_i}}(\mathbf{x}_{\mathcal{M}_i})\right) + KL(q_{\phi^i}(\mathbf{z} \mid \mathbf{x}_{\mathcal{M}_i}) \mid\mid p(\mathbf{z}))\Big\} \\ &+ \left|KL_{D_{(1:t)}} - KL_2\right| + \mathcal{E}_A\left(\mathbb{P}_{D_{(1:t)}}, \mathbb{P}_{\mathcal{M}_i}\right)\end{aligned} \tag{24}$$

where $\mathbb{P}_{D_{(1:t)}}$ is the empirical distribution formed by samples uniformly drawn from $\{\mathbb{P}_{D_1}, \cdots, \mathbb{P}_{D_t}\}$. Then according to the definition of ELBO (Eq. (13)), we can rewrite Eq. (24) as :

$$\sum_{j=1}^{t}\Big\{\mathbb{E}_{\mathbb{P}_{D_j}}\Big[-\mathcal{L}_{ELBO}\big(\mathbf{x}_{D_j}; h\big)\Big]\Big\} \leq \mathcal{E}_A\big(\mathbb{P}_{D_{(1:t)}}, \mathbb{P}_{\mathcal{M}_i}\big) + \mathbb{E}_{\mathbb{P}_{\mathcal{M}_i}}\Big[-\mathcal{L}_{ELBO}\big(\mathbf{x}_{\mathcal{M}_i}; h\big)\Big] + \big|KL_{D_{(1:t)}} - KL_2\big| \tag{25}$$

□

Eq. (25) proves Lemma 1.

C Lemma 2

In this section, we show that the Generalization Bound (GB) from Theorem 1 and Lemma 1 are also an upper bound to the real sample log-likelihood.

Lemma 2 Let $\mathcal{S}$ be a data stream and $\mathbb{P}_{\mathbf{x}'(1:i)}$ be a probability distribution for all previously seen batches $\{\{\mathbf{x}_{1,j}\}_{j=1}^{b}, \cdots, \{\mathbf{x}_{t,j}\}_{j=1}^{b}\}$ from $\mathcal{S}$ at $\mathcal{T}_i$. We can derive an upper bound for the sample log-likelihood of the target distribution :

$$\mathbb{E}_{\mathbb{P}_{\mathbf{x}'(1:i)}}\Big[-\log p_h(\mathbf{x}'(1:i))\Big] \leq \mathcal{E}_A\big(\mathbb{P}_{\mathbf{x}'(1:i)}, \mathbb{P}_{\mathcal{M}_i}\big) + \mathbb{E}_{\mathbb{P}_{\mathcal{M}_i}}\Big[-\mathcal{L}_{ELBO}\big(\mathbf{x}_{\mathcal{M}_i}; h\big)\Big] + \big|KL_1 - KL_2\big|, \tag{26}$$

where $\log p_h(\mathbf{x}'(1:i))$ is the sample log-likelihood estimated by the model h. We can observe that the GB (Eq. (26)) is tight when $\mathbb{P}_{\mathcal{M}_i}$ approximates $\mathbb{P}_{\mathbf{x}'(1:i)}$ exactly in which the discrepancy term in Eq. (26) is closed to 0. In contrast, as $\mathbb{P}_{\mathcal{M}_i}$ is gradually far away from $\log p_h(\mathbf{x}'(1:i))$, the gap in the GB (Eq. (26)) is growing, resulting in poor performance (low sample log-likelihood estimation). We also generalize Eq. 26 to analyze the generalization performance of a model at $\mathcal{T}_i$ as :

$$\frac{1}{t}\sum_{i=1}^{t}\Big\{\mathbb{E}_{\mathbf{x}_{D_j}\sim\mathbb{P}_{D_j}}[-\log p_h(\mathbf{x}_{D_j})]\Big\} \leq \mathbb{E}_{\mathbf{x}_{\mathcal{M}_i}\sim\mathbb{P}_{\mathcal{M}_i}}\left[-\mathcal{L}_{ELBO}\left(\mathbf{x}_{\mathcal{M}_i}; h\right)\right] + \left|KL_{D_{(1:t)}} - KL_2\right| + \mathcal{E}_A\left(\mathbb{P}_{D_{(1:t)}}, \mathbb{P}_{\mathcal{M}_i}\right). \tag{27}$$

The first term in RHS of Eq. (27) would be gradually increased as the number of testing sets t grows, which leads to a drop in generalization performance (the sample log-likelihood on the training sets).

Proof First, according to the definition of ELBO, we have :

$$\mathbb{E}_{\mathbb{P}_{\mathbf{x}'(1:i)}}\Big[\log p_h(\mathbf{x}'(1:i))\Big] \geq \mathbb{E}_{\mathbb{P}_{\mathbf{x}'(1:i)}}\Big[\mathcal{L}_{ELBO}\big(\mathbf{x}'(1:i); h\big)\Big] \tag{28}$$

Then we multiply -1 in both sides of Eq. (28), resulting in :

$$\mathbb{E}_{\mathbb{P}_{\mathbf{x}'(1:i)}}\Big[-\log p_h(\mathbf{x}'(1:i))\Big] \leq \mathbb{E}_{\mathbb{P}_{\mathbf{x}'(1:i)}}\Big[-\mathcal{L}_{ELBO}\big(\mathbf{x}'(1:i);h\big)\Big] \tag{29}$$

Then we also know that the Right-Hand Side (RHS) of Eq. (12) of the paper, which indicates that :

$$\begin{aligned}\mathbb{E}_{\mathbb{P}_{\mathbf{x}'(1:i)}}\Big[-\mathcal{L}_{ELBO}\big(\mathbf{x}'(1:i);h\big)\Big] \leq\ & \mathcal{E}_A\big(\mathbb{P}_{\mathbf{x}'(1:i)},\mathbb{P}_{\mathcal{M}_i}\big) \\ & + \mathbb{E}_{\mathbb{P}_{\mathcal{M}_i}}\Big[-\mathcal{L}_{ELBO}\big(\mathbf{x}_{\mathcal{M}_i};h\big)\Big] + \big|KL_1 - KL_2\big|,\end{aligned} \tag{30}$$

Then it results

$$\begin{aligned}\mathbb{E}_{\mathbb{P}_{\mathbf{x}'(1:i)}}\Big[-\log p_h(\mathbf{x}'(1:i))\Big] \leq\ & \mathcal{E}_A\big(\mathbb{P}_{\mathbf{x}'(1:i)},\mathbb{P}_{\mathcal{M}_i}\big) \\ & + \mathbb{E}_{\mathbb{P}_{\mathcal{M}_i}}\Big[-\mathcal{L}_{ELBO}\big(\mathbf{x}_{\mathcal{M}_i};h\big)\Big] + \big|KL_1 - KL_2\big|\end{aligned} \tag{31}$$

□

which proves Lemma 2.

D Proof of Theorem 2

let $\mathbb{P}_{\mathcal{M}_i\otimes\theta_{(1:c)}}$ represent a probability distribution formed by the samples uniformly drawn from $\{\mathbb{P}_{\theta_1},\cdots,\mathbb{P}_{\theta_c},\mathcal{P}_{\mathcal{M}_i}\}$. We treat the knowledge distillation procedure as a training process in which a student module h_s is trained to approximate $\mathbb{P}_{\mathcal{M}_i\otimes\theta_{(1:c)}}$. The we consider $\mathbb{P}_{\mathbf{x}'(1:i)}$ and $\mathbb{P}_{\mathcal{M}_i\otimes\theta_{(1:c)}}$ to be the target and source distribution, respectively. Let $\mathbf{x}'(1:i)$ and $\mathbf{x}''$ represent the latent variables drawn from $\mathbb{P}_{\mathcal{M}_i\otimes\theta_{(1:c)}}$ and $\mathcal{P}_{\mathcal{M}_i}$, respectively. First, we have the following inequality :

$$\begin{aligned}\mathbb{E}_{\mathbb{P}_{\mathbf{x}'(1:i)}} KL(q_{\phi^i}(\mathbf{z}\mid\mathbf{x}'(1:i))\ ||\ p(\mathbf{z})) \leq\ & \mathbb{E}_{\mathbb{P}_{\mathcal{M}_i\otimes\theta_{(1:c)}}} KL(q_{\phi^i}(\mathbf{z}\mid\mathbf{x}'')\ ||\ p(\mathbf{z})) \\ & + |\mathbb{E}_{\mathbb{P}_{\mathcal{M}_i\otimes\theta_{(1:c)}}} KL(q_{\phi^i}(\mathbf{z}\mid\mathbf{x}'')\ ||\ p(\mathbf{z})) \\ & - \mathbb{E}_{\mathbb{P}_{\mathbf{x}'(1:i)}} KL(q_{\phi^i}(\mathbf{z}\mid\mathbf{x}'(1:i))\ ||\ p(\mathbf{z}))|\end{aligned} \tag{32}$$

We use $KL_{\mathcal{M}_i\otimes\theta_{(1:c)}}$ to represent the first term in RHS of Eq. (32). We then take Eq. (32) in Eq. (11) where the source distribution $\mathbb{P}_{\mathcal{M}_i}$ is replaced by $\mathbb{P}_{\mathcal{M}_i\otimes\theta_{(1:c)}}$, we have :

$$\begin{aligned}&\mathbb{E}_{\mathbf{x}'(1:i)\sim\mathbb{P}_{\mathbf{x}'(1:i)}}\{\mathcal{L}(h(\mathbf{x}'(1:i)), f_{\mathbb{P}_{\mathbf{x}'(1:i)}}(\mathbf{x}'(1:i))) + KL(q_{\phi^i}(\mathbf{z}|\mathbf{x}'(1:i))\ ||\ p(\mathbf{z}))\} \leq \\ &\mathbb{E}_{\mathbf{x}''\sim\mathbb{P}_{\mathcal{M}_i}}\Big\{\mathcal{L}\left(h(\mathbf{x}''), h^*_{\mathbb{P}_{\mathcal{M}_i\otimes\theta_{(1:c)}}}(\mathbf{x}'')\right) + KL(q_{\phi^i}(\mathbf{z}\mid\mathbf{x}'')\ ||\ p(\mathbf{z}))\Big\} \\ &+ \big|KL_1 - KL_{\mathcal{M}_i\otimes\theta_{(1:c)}}\big| + \mathcal{E}_A\left(\mathbb{P}_{\mathcal{M}_i\otimes\theta_{(1:c)}},\mathbb{P}_{\mathbf{x}'(1:i)}\right)\end{aligned} \tag{33}$$

Then we can rewrite Eq. (33) as :

$$\begin{aligned}\mathbb{E}_{\mathbb{P}_{\mathbf{x}'(1:i)}}\Big[-\mathcal{L}_{ELBO}\big(\mathbf{x}'(1:i);h\big)\Big] \leq\ & \mathcal{E}_A\big(\mathbb{P}_{\mathbf{x}'(1:i)},\mathbb{P}_{\mathcal{M}_i\otimes\theta_{(1:c)}}\big) \\ & + \mathbb{E}_{\mathbb{P}_{\mathcal{M}_i\otimes\theta_{(1:c)}}}\Big[-\mathcal{L}_{ELBO}\big(\mathbf{x}'';h\big)\Big] \\ & + \big|KL_1 - KL_{\mathcal{M}_i\otimes\theta_{(1:c)}}\big|\end{aligned} \tag{34}$$

□

Eq. (34) proves Theorem 2.

E Lemma 3

In the following, we analyse the generalisation performance of a single model on the test sets.

Lemma 3 Let $\mathcal{S}$ be a data stream that has involved t different data domains. Let $\{D_1^T, \cdots, D_t^T\}$ be the testing sets and $\mathbb{P}_{D_j}$ be the probability distribution for D_j^T. Let $\mathbb{P}_{D_{(1:t)}}$ denote the distribution for $\{D_1^T, \cdots, D_t^T\}$. We assume that the Teacher module has already trained c experts $\mathbf{A} = \{\mathcal{A}_1, \cdots, \mathcal{A}_c\}$ on $\mathcal{M}_i$ at $\mathcal{T}_i$. Let h be a Student module which is implemented by a VAE and we derive an upper bound to the negative ELBO at $\mathcal{T}_i$ as :

$$\begin{aligned}\sum_{j=1}^{t}\Big\{\mathbb{E}_{\mathbb{P}_{D_j}}\Big[-\mathcal{L}_{ELBO}\big(\mathbf{x}_{D_j};h\big)\Big]\Big\} \leq \mathcal{E}_A\big(\mathbb{P}_{D_{(1:t)}},\mathbb{P}_{\mathcal{M}_i\otimes\theta_{(1:c)}}\big)\\ + \mathbb{E}_{\mathbb{P}_{\mathcal{M}_i\otimes\theta_{(1:c)}}}\Big[-\mathcal{L}_{ELBO}\big(\mathbf{x}'';h\big)\Big]\\ + \big|KL_{D_{(1:t)}} - KL_{\mathcal{M}_i\otimes\theta_{(1:c)}}\big|.\end{aligned} \tag{35}$$

The proof is similar to the proof of Theorem 2.

F Theoretical analysis for the existing approaches under TFCL

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

F.1 Importance sampling

Importance Weighted Autoencoder (IWELBO) [2] is an improved version of VAE, which can achieve a tight ELBO than VAE by allowing the inference network to generate multiple samples during the optimization that leads to better modelling of the posterior probabilities. The definition of ELBO can be extended for IWELBO as :

$$\mathcal{L}_{ELBO_{W'}}(\mathbf{x};h) = \mathbb{E}_{\mathbf{z}_1,..,\mathbf{z}_{K'}\sim q(\mathbf{z}|\mathbf{x})}\left[\log\frac{1}{W'}\sum_{i=1}^{W'}\frac{p(\mathbf{x},\mathbf{z}_i)}{q(\mathbf{z}_i|\mathbf{x})}\right] \tag{36}$$

where W' is the number of weighted samples. $\mathcal{L}_{ELBO_{W'}}$ is reduced to $\mathcal{L}_{ELBO}$ if $W' = 1$. h is a certain model. We also know that $-\log p_h(\mathbf{x}) \leq -\mathcal{L}_{ELBO_{W'}}(\mathbf{x};h) \leq -\mathcal{L}_{ELBO}(\mathbf{x};h)$. We assume that an optimal model $h^\star$ in $-\mathcal{L}_{ELBO}(\mathbf{x};h^\star)$ satisfies $-\mathcal{L}_{ELBO}(\mathbf{x};h^\star) \approx -\log p_{h^\star}(\mathbf{x})$ and we have $-\mathcal{L}_{ELBO}(\mathbf{x};h^\star) \leq -\mathcal{L}_{ELBO_{W'}}(\mathbf{x};h)$. Then we consider $-\mathcal{L}_{ELBO}(\mathbf{x};h^\star) \leq -\mathcal{L}_{ELBO_{W'}}(\mathbf{x};h)$ into Eq. (26), resulting in :

$$\begin{aligned}\mathbb{E}_{\mathbb{P}_{\mathbf{x}'(1:i)}}\Big[-\log p_{h^\star}(\mathbf{x}'(1:i))\Big] \leq \mathcal{E}_A\big(\mathbb{P}_{\mathbf{x}'(1:i)},\mathbb{P}_{\mathcal{M}_i}\big)\\ + \mathbb{E}_{\mathbb{P}_{\mathcal{M}_i}}\Big[-\mathbb{E}_{\mathbf{z}_1,..,\mathbf{z}_{K'}\sim q(\mathbf{z}|\mathbf{x}_{\mathcal{M}_i})}\Big[\log\frac{1}{W'}\sum_{i=1}^{W'}\frac{p(\mathbf{x}_{\mathcal{M}_i},\mathbf{z}_i)}{q(\mathbf{z}_i|\mathbf{x}_{\mathcal{M}_i})}\Big]\Big]\\ + \big|KL_1 - KL_2\big|,\end{aligned} \tag{37}$$

It observes that a tight $\mathcal{L}_{ELBO_{W'}}(\mathbf{x};h)$ can not ensure a tight GB in Eq. (37) because Eq. (37) also involves the discrepancy distance term between the source and target distribution. We can also find that Eq. (37) can be further extended to analyze the forgetting behaviour of hierarchical variational inference technologies [8, 10, 11].

F.2 Dynamic expansion model

In this section, we extend the proposed theoretical analysis for the dynamic expansion model, which usually trains a growing mixture system and does not attempt to train a lightweight Student model.

Let $\mathbf{A} = \{\mathcal{A}_1, \cdots, \mathcal{A}_c\}$ be a dynamic expansion model that assumes to train c experts at $\mathcal{T}_i$. We assume that each expert $\mathcal{A}_s$ is implemented by an independent VAE model. Let k'_s be the index of the training step $\mathcal{T}_{k'_s}$ that the s-th expert ($\mathcal{A}_s$) finished the training. As a result, we consider $\mathbb{P}_{\mathcal{M}_{k'_s}}$ for representing a probability distribution for the memory buffer that the s-th expert $\mathcal{A}_s$ has finished its training on $\mathcal{M}_{k'_s}$. Firstly, we derive the GB to analyze the forgetting behaviour of the dynamic expansion model during training.

Theorem 3 Let $\mathcal{S}$ be a given data stream and $\mathbb{P}_{\mathbf{x}'(1:i)}$ be a probability distribution for all previously seen batches $\{\{\mathbf{x}_{1,j}\}_{j=1}^{b}, \cdots, \{\mathbf{x}_{t,j}\}_{j=1}^{b}\}$ from $\mathcal{S}$ at $\mathcal{T}_i$. Let $\mathbb{P}_{\mathbf{x}'(i)}$ be a probability distribution for the i-th batch of samples $\{\mathbf{x}_{t,j}\}_{j=1}^{b}$. We derive a GB for $\mathbf{A}$ on $\mathcal{M}_i$ at $\mathcal{T}_i$:

$$\sum_{j=1}^{i} \Big\{ \mathbb{E}_{\mathbb{P}_{\mathbf{x}'(j)}} \Big[-\mathcal{L}_{ELBO}\big(\mathbf{x}'(j); h\big) \Big] \Big\} \leq \sum_{j=1}^{i} \Big\{ F_{op}\big(\mathbb{P}_{\mathbf{x}'(j)}, \mathbf{A}\big) \Big\}, \tag{38}$$

where $F_{op}(\cdot,\cdot)$ is an optimal selection function that selects the best expert and returns the minimum risk value, expressed as :

$$\begin{aligned} F_{op}\big(\mathbb{P}_{\mathbf{x}'(j)}, \mathbf{A}\big) = \min_{\mathcal{A}_s \sim \mathbf{A}, s=1,\cdots,c} \Big\{ & \mathcal{E}_A\big(\mathbb{P}_{\mathbf{x}'(j)}, \mathbb{P}_{\mathcal{M}_{k'_s}}\big) \\ & + \mathbb{E}_{\mathbb{P}_{\mathcal{M}_{k'_s}}} \Big[-\mathcal{L}_{ELBO}\big(\mathbf{x}_{\mathcal{M}_{k'_s}}; \mathcal{A}_{k'_s}\big) \Big] + \big| KL'_1(\mathbb{P}_{\mathbf{x}'_j}, \mathcal{A}_s) - KL'_2(\mathcal{A}_s) \big| \Big\} \end{aligned} \tag{39}$$

where $q_{\phi_s^{k'_s}}(\cdot,\cdot)$ is the inference model of the expert $\mathcal{A}_{k'_s}$. $KL'_1(\mathbb{P}_{\mathbf{x}'_j}, \mathcal{A}_s)$ and $KL'_2(\mathcal{A}_s)$ are defined as :

$$KL'_1(\mathbb{P}_{\mathbf{x}'_j}, \mathcal{A}_s) = \mathbb{E}_{\mathbb{P}_{\mathbf{x}'(j)}} KL\big(q_{\phi_s^{k'_s}}(\mathbf{z} \,|\, \mathbf{x}'(j)) \,||\, p(\mathbf{z})\big), \tag{40}$$

$$KL'_2(\mathcal{A}_s) = \mathbb{E}_{\mathbb{P}_{\mathcal{M}_{k'_s}}} KL\big(q_{\phi_s^{k'_s}}(\mathbf{z} \,|\, \mathbf{x}_{\mathcal{M}_{k'_s}}) \,||\, p(\mathbf{z})\big). \tag{41}$$

The proof sums the GB of each seen batch of samples using Theorem 1 of the paper. We can observe that Eq. (38) is tighter than Eq. (12) of the paper. Moreover, Eq. (38) can explain the forgetting behaviour of certain dynamic expansion models such as CN-DPM [7]. As the number of training examples increases, the dynamic expansion model can remember most of the previous information by gradually increasing its modelling capabilities through the expansion mechanism. However, this also leads to a large network architecture for the dynamic expansion model. Therefore, the statistical diversity among the experts in a dynamic expansion mixture model plays an important role in balancing the complexity of the model and the generalisation performance. However, existing dynamic expansion models such as CN-DPM and CURL would lead to training experts defined by statistically overlapping knowledge. This happens because they do not compare the similarity of knowledge between the incoming samples and the learned knowledge when performing the expansion mechanism. In practice, the dynamic expansion model usually has a certain expert selection criterion at the testing phase. In the following, we analyze the forgetting behaviour of the dynamic expansion model when using an expert selection criterion.

Lemma 4 Let F_{select} be a certain expert selection function. Then we derive a GB for $\mathbf{A}$ at $\mathcal{T}_i$ as :

$$\sum_{j=1}^{i} \Big\{ \mathbb{E}_{\mathbb{P}_{\mathbf{x}'(j)}} \Big[-\mathcal{L}_{ELBO}\big(\mathbf{x}'(j); h\big) \Big] \Big\} \leq \sum_{j=1}^{i} \Big\{ F_{select}\big(\mathbb{P}_{\mathbf{x}'(j)}, \mathbf{A}\big) \Big\}, \tag{42}$$

where F_{select} is defined as :

$$\begin{aligned} F_{\text{select}}\big(\mathbb{P}_{\mathbf{x}'(j)}, \mathbf{A}\big) = & \; \mathcal{E}_A\big(\mathbb{P}_{\mathbf{x}'(j)}, \mathbb{P}_{\mathcal{M}_{k'_s}}\big) \\ & + \mathbb{E}_{\mathbb{P}_{\mathcal{M}_{k'_s}}} \Big[-\mathcal{L}_{ELBO}\big(\mathbf{x}_{\mathcal{M}_{k'_s}}; \mathcal{A}_{k'_s}\big) \Big] + \big| KL'_1(\mathbb{P}_{\mathbf{x}'_j}, \mathcal{A}_s) - KL'_2(\mathcal{A}_s) \big| \\ & , s = F_{\text{criterion}}\big(\mathbb{P}_{\mathbf{x}'(j)}, \mathbf{A}\big) \end{aligned} \tag{43}$$

where $F_{\text{criterion}}$ is the pre-defined expert selection criterion, expressed as :

$$\begin{aligned} F_{\text{criterion}}\big(\mathbb{P}_{\mathbf{x}'(j)}, \mathbf{A}\big) = \arg\min_{\mathcal{A}_s \sim \mathbf{A}, s=1,\cdots,c} \Big\{ & \mathcal{E}_A\big(\mathbb{P}_{\mathbf{x}'(j)}, \mathbb{P}_{\mathcal{M}_{k'_s}}\big) \\ & + \mathbb{E}_{\mathbb{P}_{\mathcal{M}_{k'_s}}} \Big[-\mathcal{L}_{ELBO}\big(\mathbf{x}_{\mathcal{M}_{k'_s}}; \mathcal{A}_{k'_s}\big) \Big] + \big| KL'_1(\mathbb{P}_{\mathbf{x}'_j}, \mathcal{A}_s) - KL'_2(\mathcal{A}_s) \big| \Big\} \end{aligned} \tag{44}$$

Eq. (44) is an optimal component selection function, which, however is computational intractable because the second term in RHS of Eq. (44) require to access all previously learnt memory buffers which are not available. In practice, CURL and other dynamic expansion models usually consider the sample log-likelihood evaluation as the component selection criterion. Therefore, we define a practical component selection criterion as :

$$F_{\text{pcriterion}}\left(\mathbb{P}_{\mathbf{x}'(j)}, \mathbf{A}\right) = \arg\min_{\mathcal{A}_s \sim \mathbf{A}, s=1,\cdots,c} \left\{ \mathbb{E}_{\mathbb{P}_{\mathbf{x}'(j)}} \left[-\mathcal{L}_{ELBO}\left(\mathbf{x}'(j); \mathcal{A}_{k'_s}\right) \right] \right\} \tag{45}$$

The component selection function associated with Eq. (45) is defined as :

$$\begin{aligned} F_{\text{pselect}}\left(\mathbb{P}_{\mathbf{x}'(j)}, \mathbf{A}\right) &= \mathcal{E}_A\left(\mathbb{P}_{\mathbf{x}'(j)}, \mathbb{P}_{\mathcal{M}_{k'_s}}\right) \\ &+ \mathbb{E}_{\mathbb{P}_{\mathcal{M}_{k'_s}}} \left[-\mathcal{L}_{ELBO}\left(\mathbf{x}_{\mathcal{M}_{k'_s}}; \mathcal{A}_{k'_s}\right) \right] + \left| KL'_1(\mathbb{P}_{\mathbf{x}'_j}, \mathcal{A}_s) - KL'_2(\mathcal{A}_s) \right| \\ &, s = F_{\text{pcriterion}}\left(\mathbb{P}_{\mathbf{x}'(j)}, \mathbf{A}\right) \end{aligned} \tag{46}$$

Since $F_{\text{pcriterion}} \geq F_{\text{criterion}}$, we have $F_{\text{pselect}} \geq F_{\text{select}}$. Therefore, using Eq. (46) for the component selection in Eq. (42) (replace F_{select} by F_{pselect}) would lead more errors which are expressed as :

$$err = \sum_{j=1}^{i} \left\{ F_{\text{pselect}}\left(\mathbb{P}_{\mathbf{x}'(j)}, \mathbf{A}\right) - F_{\text{select}}\left(\mathbb{P}_{\mathbf{x}'(j)}, \mathbf{A}\right) \right\}, \tag{47}$$

From Eq. (47), we can observe that the errors caused by the expert selection (F_{pselect}) also significantly influence the performance of the existing dynamic expansion models such as CURL and CNDPM. In contrast to these models, the proposed CGKD has several advantages : 1) The proposed CGKD does not require the component selection process at the testing phase and thus does not have additional errors caused by Eq. (47). 2) The proposed CGKD is computationally efficient in the inference phase since it transfers all knowledge into a lightweight student model that is used as the evaluation. 3) The proposed CGKD can model the correlation between different data domains into a single latent space, benefiting many tasks, including cross-domain image reconstruction and interpolation.

G Additional information for the proposed approach

We provide the learning procedure of the proposed model in Fig. 1.

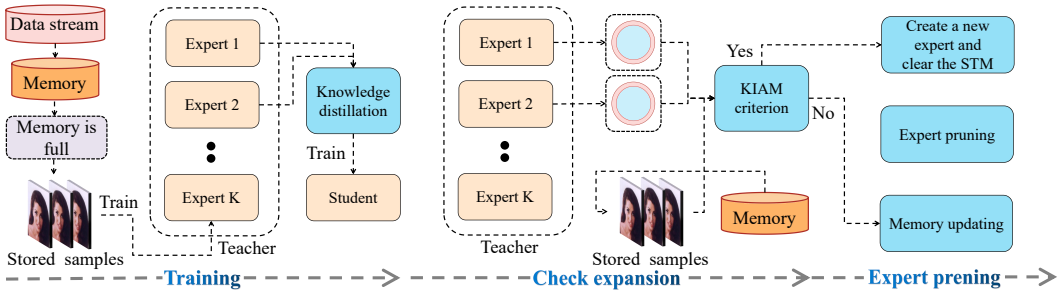


Figure 1: The learning procedure of the proposed model. We assume that the teacher module has learnt t experts. During the training, we update the current expert (Expert K) while all previously learnt experts are frozen to preserve past knowledge. We also train the student module using Eq.(4) of the paper, which involves the knowledge distillation term. We then check the model's expansion by using the KIAM criterion. If this criterion is satisfied, we create a new expert for the teacher module and clear the memory buffer. Otherwise, we update the memory buffer by adding incoming samples. We also perform the expert pruning if the number of experts is large than the predefined n.

G.1 A new expert pruning approach for the KD procedure

Section 3.4 of the paper has introduced a new expert pruning approach that uses n as a threshold to control the total number of experts in the proposed framework. However, such an approach is

limited to learn infinite data streams due the fixed maximum number of experts (n). In this section, we introduce a new threshold λ_2 to replace n, which can allow the proposed framework to have arbitrary number of experts.

The motivation of the proposed threshold λ_2 is to determine whether the relevance between two experts is significant enough. Then we only remove one of the paired experts if their discrepancy score is small than λ_2. To implement this goal, we follow the same procedure described in Section 3.4 of the paper. Once the matrix $\mathbf{Q}$ is evaluated, we identity a pair of overlapped experts by searching the minimal discrepancy score in $\mathbf{Q}$:

$$a^\star, b^\star = \arg \min_{a,b=1,\cdots,c} \left\{ \mathbf{Q}(a,b) \right\}, \tag{48}$$

where $a^\star$ and $b^\star$ are the indexes of the selected experts. We then propose to use λ_2 to determine whether we remove one of the paired experts :

$$\min \left\{ \sum\nolimits_{j=1}^{c-1} \left\{ \mathbf{Q}(a^\star, j), j \neq a^\star \right\}, \sum\nolimits_{j=1}^{c-1} \left\{ \mathbf{Q}(b^\star, j), j \neq b^\star \right\} \right\} < \lambda_2 \,, \tag{49}$$

If Eq. (49) is satisfied, we then remove one of the paired experts according to the criterion :

$$c^\star = \arg \min_{c=a^\star, b^\star} \left\{ \sum\nolimits_{j=1}^{c-1} \left\{ \mathbf{Q}(a^\star, j), j \neq a^\star \right\}, \sum\nolimits_{j=1}^{c-1} \left\{ \mathbf{Q}(b^\star, j), j \neq b^\star \right\} \right\}, \tag{50}$$

We use Eq. (50) to remove the $c^\star$-th expert from the teacher module. To remove multiple experts, we can repeat the above procedure until the criterion (Eq .(49)) is no longer satisfied. We refer to the threshold λ_2 used in our framework as CGKD-GAN-λ_2 and CGKD-VAE-λ_2 when the teacher uses GANs and VAEs, respectively.

In the next section, we provide the detailed pseudo code for both two expert pruning approaches.

G.2 Implementation for the algorithm

We present the training algorithm of CGKD*-GAN and CGKD-GAN-λ_2 in Algorithm 1, where the CGKD*-GAN uses the threshold n to avoid extra experts while the CGKD-GAN-λ_2 employs the threshold λ_2 to gradually remove unnecessary experts which can be used in infinite data streams. In addition, to evaluate Eq.(1) of the paper, each $\mathbb{P}_{\theta_j}$ is formed by the same number of samples with respect to the memory buffer $\mathcal{M}_i$. Moreover, to evaluate Eq.(5) of the paper, we generate 1000 samples using the VAE generator of each component and map these samples using the inference model $q_\phi(\mathbf{z} \,|\, \mathbf{x})$ of the student.

H Additional information and results for the experiment

H.1 Additional Information for the experiment setting

Network Architecture. We introduce the details for the network architecture. For the generator of each expert in the teacher module and the decoder of the Student, we adopt the same CNN network that consists of five convolution layers $\{256, 256, 256, 256, 3\}$. For the inference model $q(\mathbf{z} \mid \mathbf{x})$ of the Student, we use the network consisting of four convolution layers and two fully connected layers that returns the hyperparameters of Gaussian distribution. The kernel size is 3×3. The number of kernels in each convolution layer is of $\{64, 128, 256, 512\}$. The number of hidden nodes in the fully connected layer is 256. For the input size of $64 \times 64 \times 3$, The generator for each expert consists of 6 convolution layer $\{256, 256, 256, 256, 128, 3\}$. The decoder of the Student also uses the same network architecture with the generator. The inference model $q(\mathbf{z} \mid \mathbf{x})$ of the Student and each expert consists of four convolution layers $\{64, 128, 256, 512\}$, one hidden layers $\{1024\}$ and two separate layers $\{256\}$ which are used to outputs the hyperparameter of Gaussian distribution.

For all experiments, we use the Adam optimization algorithm [5] with a learning rate of 0.0002 and the hyperparameter $\beta = 0.5$. The number of training epochs for each training step is set to 1.

GPU. Following from [22, 25, 23, 21, 19, 1, 24, 20, 13, 14, 17, 16, 18, 15], we consider to adopt Tesla V100-SXM2 (32GB) GPU and RHEL 8 operating system for our experiment.

Algorithm 1: Training algorithm of CGKD*-GAN and CGKD-GAN-λ_2

Input:$\mathcal{S}$, t (Number of training steps);

for $i < t$ **do**
  $\{\mathbf{x}_{m,j}\}_{j=1}^{b} \sim \mathcal{S}$;
  Updating of the memory;
  if ($|\mathcal{M}_i| \geq |\mathcal{M}_i|_{max}$) **then**
    Remove earliest samples from $\mathcal{M}_i$;
  end if
  $\mathcal{M}_i = \mathcal{M}_i \bigcup \{\mathbf{x}_{m,j}\}_{j=1}^{b}$;
  Teacher learning;
  if (i == 100) **then**
    Build a new expert $\mathcal{A}_2$ while fixing $\mathcal{A}_1$.;
  else
    Train the latest expert on $\mathcal{M}_i$ using either Eq.(8) or Eq.(9);
  end if
  Checking the expansion;
  if ($|\mathcal{M}_i| \geq |\mathcal{M}_i|_{max}$) **then**
    If Eq.(1) of the paper is satisfied, we add a new expert $\mathcal{A}_{c+1}$ to the teacher module;
    Cleaning up $\mathcal{M}_i$;
  end if
  Expert pruning for KD (CGKD*-GAN);
  while True **do**
    if ($|\mathcal{G}| > n$) **then**
      Calculate $\mathbf{Q}$ using Eq.(5) of the paper;
      Find a pair of experts by $a^\star, b^\star = \arg\min_{a,b=1,\cdots,c} \{\mathbf{Q}(a,b)\}$;
      $c^\star = \arg\min_{c=a^\star,b^\star} \left\{ \sum_{j=1}^{c-1} \{\mathbf{Q}(a^\star,j), j \neq a^\star\}, \sum_{j=1}^{c-1} \{\mathbf{Q}(a^\star,j), j \neq b^\star\} \right\}$;
      Remove $c^\star$-th expert from the teacher;
    else
      break;
    end if
  end while
  Expert pruning for KD (CGKD-GAN-λ_2);
  while True **do**
    Calculate $\mathbf{Q}$ using Eq.(5) of the paper;
    if ($\min \left\{ \sum_{j=1}^{c-1} \{\mathbf{Q}(a^\star,j), j \neq a^\star\}, \sum_{j=1}^{c-1} \{\mathbf{Q}(a^\star,j), j \neq b^\star\} \right\} < \lambda_2$) **then**
      $c^\star = \arg\min_{c=a^\star,b^\star} \left\{ \sum_{j=1}^{c-1} \{\mathbf{Q}(a^\star,j), j \neq a^\star\}, \sum_{j=1}^{c-1} \{\mathbf{Q}(a^\star,j), j \neq b^\star\} \right\}$;
      Remove $c^\star$-th expert from the teacher;
    else
      break;
    end if
  end while
  Perform the expert pruning to remove necessary experts;
  Student learning;
  Train the student using Eq.(4) of the paper;
end for

H.2 Additional results for lifelong generative modelling

In this section, we provide more results for the MSFIRC and CI-MSFIRC settings. The FID and IS results for MSFIRC and CI-MSFIRC are listed in Tab. 1 and Tab. 2.

H.3 Additional visual results

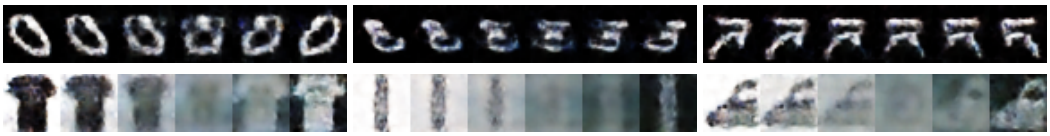
In this section, we provide the additional visual results. First, the image interpolation results for MSFIRC are presented in Fig. 2, where we compare the proposed CGKD-GAN with the Reservoir.

Table 1: The inception score and FID of various models after the MSFIRC lifelong learning.

Methods	MNIST	SVHN	Fashion	IFashion	RMNIST	CIFAR10	Average	No
Fréchet Inception Distance								
finetune	174.1	148.3	237.0	229.1	159.2	216.4	194.0	1
Reservoir [12]	127.2	159.3	213.4	201.6	110.2	113.3	154.2	1
LTS [20]	44.8	62,9	92.9	83.1	41.8	80.3	67.7	1
LGM [9]	104.8	134.3	194.3	168.1	94.8	91.5	131.3	1
CN-DPM [7]	118.7	73.4	120.7	120.3	97.9	97.6	104.8	18
CGKD-GAN	11.6	70.6	101.9	29.9	11.41	68.6	49.0	7
CGKD-VAE	122.9	73.6	109.2	104.3	119.1	86.4	102.6	11
CGKD*-GAN	12.0	74.6	69.8	22.3	11.4	68.5	43.1	7
CGKD*-VAE	82.6	82.5	127.0	132.9	88.8	86.3	100.0	7
Inception score								
finetune	1.8	1.7	2.0	2.0	1.9	2.0	1.9	1
Reservoir [12]	2.7	3.2	2.9	3.7	2.7	3.8	3.2	1
LTS [20]	2.2	3.4	3.7	3.9	2.1	4.9	3.4	1
LGM [9]	2.7	3.6	3.1	4.1	2.7	4.2	3.4	1
CN-DPM [7]	2.6	3.1	3.4	4.0	2.6	4.3	3.3	18
CGKD-GAN	1.9	3.7	3.8	3.9	1.9	5.1	3.4	7
CGKD-VAE	2.7	3.2	3.6	4.1	2.7	4.6	3.5	11
CGKD*-GAN	1.9	3.7	4.2	3.5	1.9	5.0	3.4	7
CGKD*-VAE	2.5	3.3	3.5	4.4	2.6	4.7	3.5	7

These visual results show that the proposed CGKD-GAN can produce better image interpolation results.

(a) The results of CGKD-GAN.



(b) The results of Reservoir.

Figure 2: Image interpolation results of CGKD-GAN under MSFIRC setting.

Table 2: The inception score and FID of various models under the class-incremental setting.

Methods	MNIST	SVHN	Fashion	IFashion	RMNIST	CIFAR10	Average	No
Fréchet Inception Distance								
finetune	158.1	167.6	246.2	233.3	138.6	229.4	195.6	1
Reservoir [12]	141.7	163.6	220.0	200.1	127.1	115.5	161.3	1
LTS [20]	101.9	99.4	140.6	139.5	99.9	95.5	112.8	1
LGM [9]	108.5	122.1	189.5	175.9	96.6	92.4	130.9	1
CN-DPM [7]	90.9	62.0	109.0	95.0	77.9	95.5	88.4	18
CGKD-GAN	16.7	65.1	44.5	43.9	27.9	85.2	47.2	11
CGKD-VAE	102.6	69.9	117.1	99.5	113.0	82.7	97.5	11
CGKD*-GAN	13.51	72.73	89.91	52.12	12.49	71.98	52.12	7
CGKD*-VAE	131.0	70.3	106.7	92.2	126.5	87.7	102.4	7
Inception score								
finetune	1.5	1.6	2.1	2.0	1.5	1.9	1.8	1
Reservoir [12]	2.8	3.2	2.8	3.4	2.8	3.8	3.1	1
LTS [20]	2.5	3.4	3.2	4.0	2.5	4.3	3.3	1
LGM [9]	2.8	3.6	3.0	3.9	2.8	4.2	3.4	1
CN-DPM [7]	2.6	3.1	3.7	4.0	2.5	4.3	3.4	18
CGKD-GAN	1.9	3.1	3.9	3.9	1.9	4.5	3.2	11
CGKD-VAE	2.7	3.2	3.7	4.5	2.7	4.7	3.6	11
CGKD*-GAN	1.9	3.8	4.1	4.1	1.9	5.0	3.5	7
CGKD*-VAE	2.7	3.2	3.9	4.4	2.7	4.7	3.6	7

Figure 3: Image interpolation results of CGKD-GAN under CelebA-Chair setting.

I Additional results for ablation study

In this section, we provide additional ablation studies in order to investigate the effectiveness of each component in the proposed framework.

I.1 The effect of the memory size

In this section, we examine the performance and number of experts for the proposed CGKD-GAN and CN-DPM with different memory configurations under the MSFIRC settings. The configuration for maximum memory size (the maximum number of samples stored in memory) is 1000, 2000, 3000, 4000, 5000 and 6000 respectively. The results are given in Fig. 4. It can be observed that both CGKD-GAN and CN-DPM gradually increase the number of experts as the maximum memory size decreases. A large memory buffer would also increase the performance and reduce the complexity of the model for CGKD-GAN as it can store more training samples. In contrast, reducing the maximum memory size leads to learn more experts for CGKD-GAN. This result indicates that the proposed expansion mechanism can automatically increases the model's capacity to combat the small-scale memory. Compared to CN-DPM, the proposed CGKD-GAN achieves better results even though it uses fewer experts with different memory configurations.

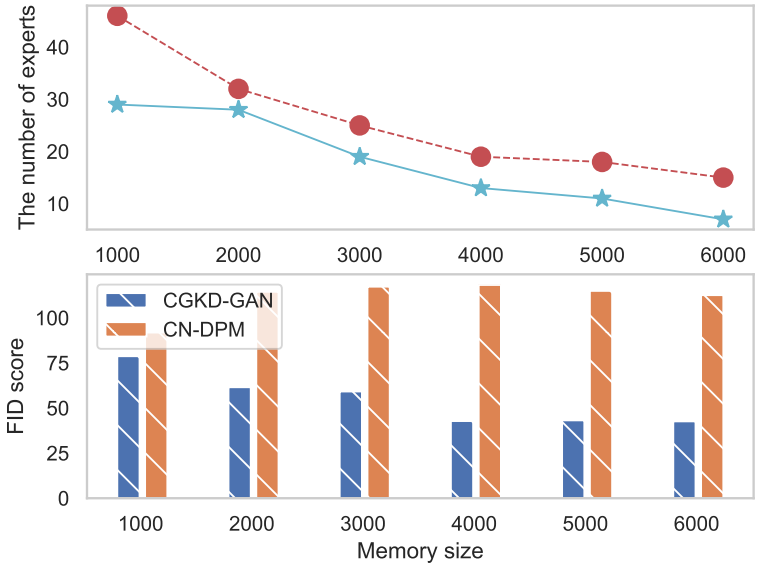


Figure 4: The performance and the number of experts for CGKD-GAN and CN-DPM with different memory configurations under the MSFIRC setting.

I.2 The effect of varying n in the proposed expert pruning approach

In this section, we investigate the impact of the threshold n for the proposed CGKD*- GAN. We train CGKD*- GAN under the MSFIRC setting with different n and the results are shown in Fig. 5. It can be seen that the proposed CGKD*- GAN would lose performance when n is very small. On the other hand, when n is equal to or larger than 5, the proposed CGKD*- GAN achieves stable performance.

I.3 The effect of varying λ_2 in CGKD-GAN-λ_2

In this section, we investigate the performance of the proposed CGKD-GAN-λ_2 when changing the threshold λ_2. We train CGKD-GAN-λ_2 under the MSFIRC setting with the different threshold λ_2 and the empirical results are shown in Fig. 6. It observes that a small threshold λ_2 leads to more experts for CGKD-GAN-λ_2. In contrast, a large threshold λ_2 makes CGKD-GAN-λ_2 to use fewer experts after the training. However, varying the threshold λ_2 in CGKD-GAN-λ_2 does not change the final performance too much, as observed in Fig. 6.

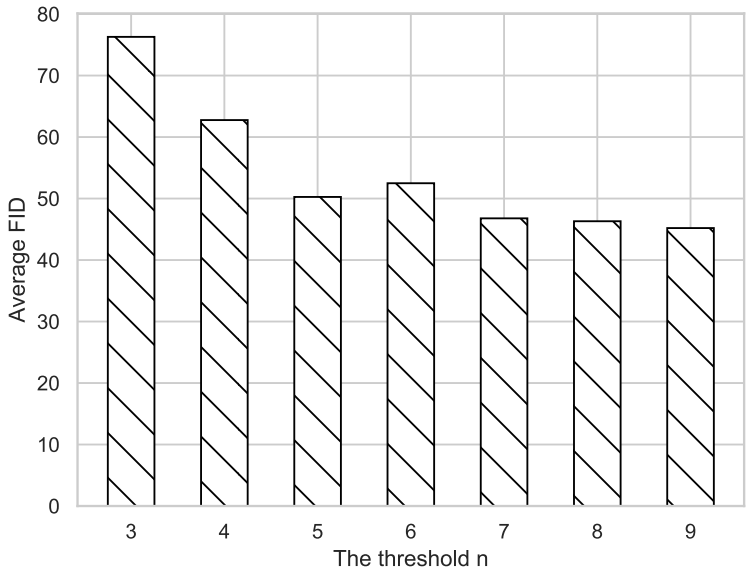


Figure 5: The performance for CGKD*-GAN with the different threshold n under the MSFIRC setting.

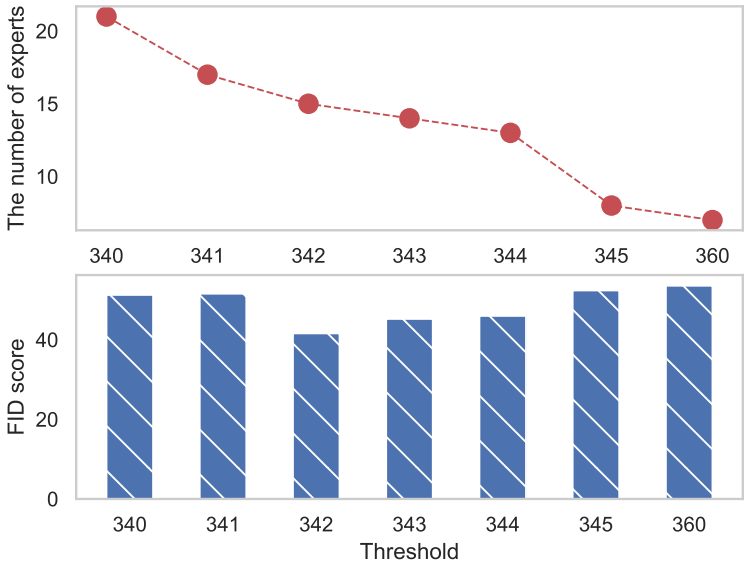


Figure 6: The performance for CGKD-GAN-λ_2 with the different threshold λ_2 under the MSFIRC setting.

I.4 Reducing computational costs using the offline learning for the student module

In this section, we explore how we can further reduce training time for CGKD-GAN without sacrificing performance. We create a baseline, called CGKD-GAN-Offline, which trains the student only when the number of experts in the teacher module exceeds a certain number (6 in our experiments). In this way, CGKD-GAN-Offline updates the teacher module only in most training times and can reduce the overall computational cost. We train CGKD-GAN and CGKD-GAN-Offline with the same setting under MSFIRC and the empirical results are shown in Fig. 7. From the results, CGKD-GAN-Offline can significantly reduce the total training time while the performance remains competitive with CGKD-GAN.

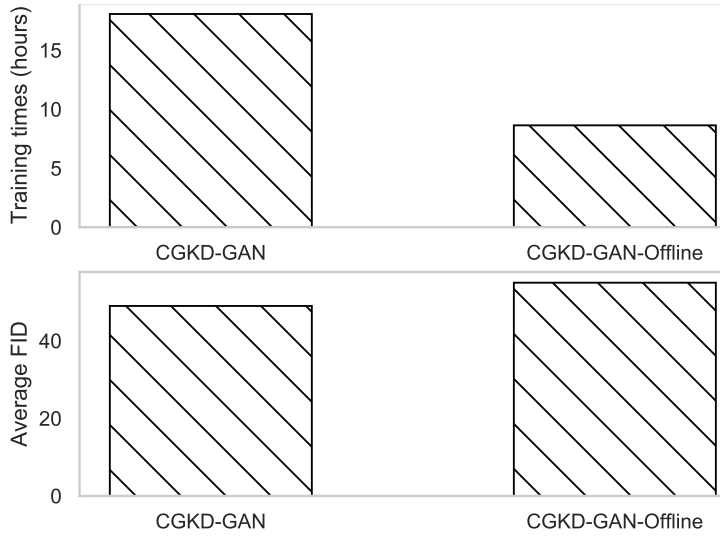


Figure 7: The results of CGKD-GAN and CGKD-GAN-Offline under the MSFIRC setting.

I.5 The change of the order of data domains

In this section, we investigate the performance of various models when changing the order of data domains in a data stream $\mathcal{S}$. First, we consider creating a data stream consisting of Fashion, SVHN, IFashion, RMNIST, CIFAR10 and MNIST, namely FSIRCM. We train various models under FSIRCM and report the results in Tab. 3. These results show that the proposed CGKD still outperforms other baselines under FSIRCM with a suitable number of components.

We also consider a data stream consists of IMNIST, Fashion, IFashion, MNIST, SVHN and CIFAR10, namely IFIMSC. We report the results in Tab. 4. The proposed CGKD still outperforms other baselines under the IFIMSC setting. Together with results from Tab. 4 and Tab. 3, we demonstrate that the proposed CGKD is robust to the change of the order of data domains in a data stream.

Table 3: The inception score and FID of various models after the FSIRCM lifelong learning.

Methods	MNIST	SVHN	Fashion	IFashion	RMNIST	CIFAR10	Average	No
Fréchet Inception Distance								
finetune	30.5	127.0	185.4	179.9	28.9	206.1	126.3	1
Reservoir [12]	5.7	213.8	137.1	216.2	5.0	197.5	129.2	1
LTS [20]	6.7	155.1	109.3	167.8	6.2	230.5	112.6	1
LGM [9]	4.4	213.2	164.7	227.8	4.0	192.4	134.4	1
CN-DPM [7]	6.5	76.3	99.8	112.6	5.6	128.7	71.6	26
CGKD-GAN	6.1	79.2	16.1	118.2	13.8	110.4	57.3	10
CGKD-VAE	4.1	75.5	94.5	139.7	3.7	135.8	75.6	11
CGKD*-GAN	5.0	60.6	15.1	139.7	8.6	98.1	**54.5**	7
CGKD*-VAE	4.3	77.0	75.6	101.7	3.9	109.1	61.9	7
Inception scrore								
finetune	2.0	2.6	2.6	2.6	2.0	2.7	2.4	1
Reservoir [12]	1.9	2.2	3.1	2.7	2.0	2.2	2.3	1
LTS [20]	1.9	2.0	3.1	2.7	1.9	2.2	2.3	1
LGM [9]	2.0	3.1	3.1	2.8	2.0	2.9	2.6	1
CN-DPM [7]	1.9	3.1	3.0	3.5	1.9	4.2	3.0	26
CGKD-GAN	1.9	2.8	3.8	4.4	2.0	3.9	3.1	10
CGKD-VAE	1.9	3.0	3.3	3.4	2.0	4.0	2.9	11
CGKD*-GAN	2.0	2.7	3.8	4.4	2.0	3.8	3.1	7
CGKD*-VAE	1.9	3.0	3.5	3.4	1.9	4.7	3.1	7

I.6 The complicated dataset

In this section, we also consider to learn the complicated image dataset, ImageNet [6]. We choose 50000 number of training samples from CelebA and ImageNet, respectively, resulting in a total 10000 training samples. We then create a data stream consisting of these training samples, namely CelebA-ImageNet. We also select 10000 images from each dataset as the testing set. We report the results in Table. 5. It observes that the proposed CGKD-GAN still outperforms other baselines on the complicated dataset.

I.7 The similarity criterion in expert pruning

In Eq.(5) of the paper, we evaluate the knowledge similarity on the feature space, which is computational efficient. The other criterion can be considered in Eq.(5). First, we can replace the square loss in Eq.(5) of the paper by using the cosine similarity, expressed as :

$$\mathcal{L}_{ks}(\mathcal{A}_a, \mathcal{A}_b) = \mathbb{E}_{\mathbf{z}_a \sim \mathcal{A}_a, \mathbf{z}_b \sim \mathcal{A}_b} D_{cs}(\mathbf{z}_a, \mathbf{z}_b)\,, \tag{51}$$

where $D_{cs}(\cdot,\cdot)$ is the cosine similarity :

$$D_{cs}(\cdot,\cdot) = \frac{\sum_s^{d_z}\{\mathbf{z}_a[s]\} \times \mathbf{z}_b[s]\}}{\sqrt{\sum_s^{d_z}(\mathbf{z}_a[s])^2} \times \sqrt{\sum_s^{d_z}(\mathbf{z}_b[s])^2}}\,, \tag{52}$$

where d_z is the dimension of the latent space and $\mathbf{z}_a[s]$ represents the s-th dimensional value of $\mathbf{z}_a$. We replace Eq.(5) of the paper with Eq. (52), used for the expert pruning, namely CGKD*-GAN-CS. Then we train CGKD*-GAN-CS under the MSFIRC lifelong learning and report the results in Tab. 6. It observes that CGKD*-GAN-CS achieves similar performance when compared with other baselines.

Table 4: The inception score and FID of various models after the IFIMSC lifelong learning.

Methods	MNIST	SVHN	Fashion	IFashion	RMNIST	CIFAR10	Average	No
Fréchet Inception Distance								
finetune	206.1	142.2	265.8	236.3	227.1	204.6	213.7	1
Reservoir [12]	185.5	120.8	223.9	193.5	191.1	110.4	170.9	1
LTS [20]	7.1	186.0	122.9	193.7	6.5	256.8	128.8	1
LGM [9]	215.7	109.2	250.7	202.2	223.0	119.6	186.7	1
CN-DPM [7]	89.7	87.2	135.9	125.4	103.7	98.6	106.7	26
CGKD-GAN	16.4	57.0	75.9	28.8	15.4	88.3	47.0	12
CGKD-VAE	58.8	64.3	96.3	93.9	38.8	97.3	74.9	10
CGKD*-GAN	10.5	49.0	33.2	20.2	9.7	76.2	33.1	7
CGKD*-VAE	112.5	77.9	140.2	123.3	104.0	90.4	108.0	7
Inception score								
finetune	1.7	1.8	1.9	2.1	1.6	2.2	1.9	1
Reservoir [12]	2.8	3,4	2.7	3.6	2.8	3.9	3.2	1
LTS [20]	1.9	2.1	3.1	2.6	1.9	2.4	2.3	1
LGM [9]	2.7	3.4	2.5	3.6	2.7	3.5	3.1	1
CN-DPM [7]	2.4	3.2	3.3	4.2	2.4	4.2	3.3	26
CGKD-GAN	1.9	3.0	3.8	3.7	2.0	4.5	3.1	12
CGKD-VAE	2.3	3.3	3.4	4.2	2.1	4.3	3.3	10
CGKD*-GAN	1.9	3.3	3.9	3.6	1.9	4.8	3.2	7
CGKD*-VAE	2.6	3.1	3.4	4.4	2.5	4.3	3.4	7

Table 5: The FID of various models under the CelebA-ImageNet learning setting.

Methods	CelebA	ImageNet	Average	No
finetune	64.0	186.9	125.5	1
Reservoir [12]	25.3	129.7	77.5	1
LTS [20]	34.2	123.2	78.7	1
LGM [9]	47.9	122.5	85.2	1
CN-DPM [7]	22.5	128.2	75.3	11
CGKD-GAN	19.3	112.9	66.1	2
CGKD-VAE	53.8	145.1	99.4	4

I.8 Extension for classification task

In this section, we try to extend our model for classification task. We adopt the latest TFCL benchmark from [22]. Similar to [22], we only consider a teacher module where each expert is implemented by a VAE model. We also train a single classifier for each expert. Therefore, each expert $\mathcal{A}_i$ in the teacher module consists of a VAE model $\{p_{\theta_i}(\mathbf{x} \mid \mathbf{z}), q_{\eta_i}(\mathbf{z} \mid \mathbf{x})\}$ and a classifier (for example ResNet-18) C_{δ_i}, where δ_i denotes the classifier's parameters of the i-th expert.

We train the current classifier and the VAE using the samples drawn from the memory buffer. We also use the proposed Knowledge Incremental Assimilation Mechanism to dynamically add new experts during the training. At the testing phase, the VAE is used for the expert selection by comparing the sample log-likelihood. We provide the pseudo code of the classification task in Algorithm 2.

Table 6: The inception score and FID of various models after the MSFIRC lifelong learning.

Methods	MNIST	SVHN	Fashion	IFashion	RMNIST	CIFAR10	Average	No
Fréchet Inception Distance								
CGKD*-GAN-CS	7.7	68.3	34.0	129.9	9.7	75.6	54.2	7
CGKD-GAN	11.6	70.6	101.9	29.9	11.41	68.6	49.0	7
CGKD-VAE	122.9	73.6	109.2	104.3	119.1	86.4	102.6	11
CGKD*-GAN	12.0	74.6	69.8	22.3	11.4	68.5	43.1	7
CGKD*-VAE	82.6	82.5	127.0	132.9	88.8	86.3	100.0	7
Inception score								
CGKD*-GAN-CS	1.9	3.6	3.9	4.6	2.0	4.9	3.5	7
CGKD-GAN	1.9	3.7	3.8	3.9	1.9	5.1	3.4	7
CGKD-VAE	2.7	3.2	3.6	4.1	2.7	4.6	3.5	11
CGKD*-GAN	1.9	3.7	4.2	3.5	1.9	5.0	3.4	7
CGKD*-VAE	2.5	3.3	3.5	4.4	2.6	4.7	3.5	7

We employ ResNet 18 [4] as the classifier for Split CIFAR10 and Split CIFAR100. We use a fully connect network with 2 hidden layers of 400 units each [3] for Split MNIST. The maximum memory size for Split MNIST, Split CIFAR10, Split CIFAR100 are 2000, 1000 and 5000, respectively.

We adopt the same setting and datasets from [22, 3] and report the results in Table 7, where the performance of all baselines are cited by [22, 3]. These results show that our model is applicable for the classification task with better performance than other baselines.

Table 7: Classification accuracy of five independent runs for various models on three datasets. respectively.

Methods	Split MNIST	Split CIFAR10	Split CIFAR100
finetune	19.75 ± 0.05	18.55 ± 0.34	3.53 ± 0.04
GEM	93.25 ± 0.36	24.13 ± 2.46	11.12 ± 2.48
iCARL	83.95 ± 0.21	37.32 ± 2.66	10.80 ± 0.37
reservoir	92.16 ± 0.75	42.48 ± 3.04	19.57 ± 1.79
MIR	93.20 ± 0.36	42.80 ± 2.22	20.00 ± 0.57
GSS	92.47 ± 0.92	38.45 ± 1.41	13.10 ± 0.94
CoPE-CE	91.77 ± 0.87	39.73 ± 2.26	18.33 ± 1.52
CoPE	93.94 ± 0.20	48.92 ± 1.32	21.62 ± 0.69
ER + GMED†	82.67 ± 1.90	34.84 ± 2.20	20.93 ± 1.60
ER_a + GMED†	82.21 ± 2.90	47.47 ± 3.20	19.60 ± 1.50
CURL	92.59 ± 0.66	-	-
CNDPM	93.23 ± 0.09	45.21 ± 0.18	20.10 ± 0.12
Dynamic-OCM	94.02 ± 0.23	49.16 ± 1.52	21.79 ± 0.68
The proposed model	$\mathbf{95.78} \pm 0.27$	$\mathbf{53.98} \pm 1.27$	$\mathbf{26.92} \pm 1.17$

I.9 The comparison of computational costs

In this section, we introduce three approaches to further reduce the computational costs of the proposed model.

Algorithm 2: Training algorithm for the classification task

Input:$\mathcal{S}$, t (Number of training steps);

for $i < t$ **do**
  $\{\mathbf{x}_{m,j}, y_{m,j}\}_{j=1}^{b} \sim \mathcal{S}$;
  Updating of the memory;
  if ($|\mathcal{M}_i| \geq |\mathcal{M}_i|_{max}$) **then**
    Remove earliest samples from $\mathcal{M}_i$;
  end if
  $\mathcal{M}_i = \mathcal{M}_i \bigcup \{\mathbf{x}_{m,j}\}_{j=1}^{b}$;
  Teacher learning;
  if (i == 100) **then**
    Build a new expert $\mathcal{A}_2$ while fixing $\mathcal{A}_1$.;
  else
    Train the VAE model of the latest expert on $\mathcal{M}_i$ using the VAE loss;
    Train the classifier of the latest expert on $\mathcal{M}_i$ using the cross-entropy;
  end if
  Checking the expansion;
  if ($|\mathcal{M}_i| \geq |\mathcal{M}_i|_{max}$) **then**
    If Eq.(1) of the paper is satisfied, we add a new expert $\mathcal{A}_{c+1}$ to the teacher module;
    Cleaning up $\mathcal{M}_i$;
  end if
end for
Testing phase;
for $j < count$ **do**
  $\mathbf{x} \sim \mathcal{D}^T$ testing sample;
  $s* = \arg\max_{i=1,\cdots,K}\{\mathcal{L}_{ELBO}(\mathbf{x}; \mathcal{A}_i)\}$ Component selection;
  $y = C_{\delta_{s*}}(\mathbf{x})$ make prediction;
end for

(1) We empirically find that we do not require a significant number of samples for the evaluation of Eq.(5) and Eq.(1) of the paper. Therefore, we only use each expert to generate a batch, considered as 64 in the experiments, of samples for evaluating Eq.(5) and Eq.(1) of the paper.

(2) We use a predefined threshold $n \in [3, 10]$ to control the total number of experts. Therefore, when the number of experts of the proposed CGKD-GAN is larger than n, we can perform the expert pruning process. Thus the expert pruning process would not happen frequently during the training.

(3) We can transfer the teacher's knowledge to the student module after the teacher module has finished the training, which can further reduce the computational costs. During each training step, only the current teacher's component is updated while all components are frozen. Therefore, as the number of components increases, the computational costs for the proposed CGKD-GAN during the training is not increased.

We report the training time of the proposed CGKD-GAN with the mentioned three approaches and the baseline (LTS) in Tab. 8. The results show that the proposed model requires less training time while achieving better performance than the baseline (LTS).

Table 8: The training times required by various models under the MSFIRC setting.

Methods	**CGKD-GAN**	**LTS**
Training time (hours)	4.07	4.23

I.10 Exploring other expansion criterion

In this section, we employ the student module as a pre-trained model for training and evaluation. Specifically, we consider that the student module can accumulate knowledge over time and can thus

be used to evaluate the novelty of incoming samples. Once the student finishes the training in a time, we treat the student module as a pre-trained evaluator which aims to detect the data distribution shift. In the following, we provide the detailed implementation. Since the student module, which is a VAE, can estimate the sample log-likelihood, we can replace the FID metric in (Eq.(1) of the paper) by considering the difference on the sample log-likelihood (D_s).

$$D_s(\mathbb{P}_{\theta_j}, \mathbb{P}_{\mathcal{M}_i}) = \mathbb{E}_{\mathbf{x} \sim \mathbb{P}_{\theta_j}, \mathbf{x}' \sim \mathbb{P}_{\mathcal{M}_i}} \left| \mathcal{L}_{ELBO}(\mathbf{x}; Stu) - \mathcal{L}_{ELBO}(\mathbf{x}'; Stu) \right| \tag{53}$$

where $\mathcal{L}_{ELBO}(\mathbf{x}'; Stu)$ is the sample log-likelihood estimated by using the student module and $|\cdot|$ denotes the absolute value. Therefore, the dynamic expansion criterion (Eq.(1) of the paper) can be replaced by:

$$D_s(\mathbb{P}_{\theta_j}, \mathbb{P}_{\mathcal{M}_i}) \geq \nu \tag{54}$$

Similar images usually tend to have close sample log-likelihood and therefore, $D_s(\cdot)$ can be used to measure the knowledge similarity between two distributions. Furthermore, the new dynamic expansion criterion does not uses FID and IS and is easy to be implemented. We report the results of the proposed framework using the new dynamic expansion criterion in Tab. 9, where "CGKD-GAN-New" denotes that the proposed CGKD-GAN uses the new dynamic expansion mechanism. These results show that the proposed framework still achieves competitive performance compared with the OGKD-GAN and OGKD-VAE.

Table 9: FID for various models under the MSFIRC setting.

Methods	**MNIST**	**SVHN**	**Fashion**	**IFashion**	**RMNIST**	**CIFAR10**	**Average**	**No**
CGKD-GAN-New	21.2	62.6	102.1	30.5	23.2	84.0	53.9	14
CGKD-VAE-New	125.9	75.5	108.4	101.5	120.6	86.7	103.1	12

J Negative societal impact and limitation

One negative societal impact of this work is that the proposed model would produce unsuitable image generations when it is applied in the privacy dataset.

The main limitation of this work is that the proposed KIAM still involves a threshold which controls the number of generators for the teacher module. An inappropriate threshold would lead to creating more experts which learn overlapped knowledge. This issue is addressed by the proposed expert pruning approach that removes unnecessary experts that store the overlapped knowledge.

References

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

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

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

[4] 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.

[5] 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.

[6] 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.

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

[8] L. Maale, C. K. Snderby, S. K. Snderby, and O. Winther. Auxiliary deep generative models. In *Proc. Int. Conf. on Machine Learning (ICML) vol. PMLR 48*, pages 1445–1453, 2016.

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

[10] Artem Sobolev and Dmitry Vetrov. Importance weighted hierarchical variational inference. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 32, pages 601–613, 2019.

[11] Casper Kaae Sønderby, Tapani Raiko, Lars Maaløe, Søren Kaae Sønderby, and Ole Winther. Ladder variational autoencoders. In *Advances in neural information processing systems*, pages 3738–3746, 2016.

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

[13] 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.

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

[15] Fei Ye and Adrian G Bors. Mixtures of variational autoencoders. In *2020 Tenth International Conference on Image Processing Theory, Tools and Applications (IPTA)*, pages 1–6. IEEE, 2020.

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

[17] Fei Ye and Adrian G Bors. Lifelong generative modelling using dynamic expansion graph model. *arXiv preprint arXiv:2112.08370*, 2021.

[18] 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)*, pages 10695–10704, 2021.

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

[20] Fei Ye and Adrian G. Bors. Lifelong teacher-student network learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021.

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

[22] Fei Ye and Adrian G Bors. Continual variational autoencoder learning via online cooperative memorization. *arXiv e-prints*, pages arXiv–2207, 2022.

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

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

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