Lifelong Generative Modelling Using Dynamic Expansion Graph Model

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

Abstract
Variational Autoencoders (VAEs) suffer from degenerated performance, when learning several successive tasks. This is caused by catastrophic forgetting. In order to address the knowledge loss, VAEs are using either Generative Replay (GR) mechanisms or Expanding Network Architectures (ENA). In this paper we study the forgetting behaviour of VAEs using a joint GR and ENA methodology, by deriving an upper bound on the negative marginal log-likelihood. This theoretical analysis provides new insights into how VAEs forget the previously learnt knowledge during lifelong learning. The analysis indicates the best performance achieved when considering model mixtures, under the ENA framework, where there are no restrictions on the number of components. However, an ENA-based approach may require an excessive number of parameters. This motivates us to propose a novel Dynamic Expansion Graph Model (DEGM). DEGM expands its architecture, according to the novelty associated with each new dataset, when compared to the information already learnt by the network from previous tasks. DEGM training optimizes knowledge structuring, characterizing the joint probabilistic representations corresponding to the past and more recently learned tasks. We demonstrate that DEGM guarantees optimal performance for each task while also minimizing the required number of parameters.

Introduction
The Variational Autoencoder (VAE) (Kingma and Welling 2013) is a popular generative deep learning model with remarkable successes in learning unsupervised tasks by inferring probabilistic data representations (Chen et al. 2018), for disentangled representation learning (Higgins et al. 2017; Ye and Bors 2021d) and for image reconstruction tasks (Ye and Bors 2020c, 2021b,c,d). Training a VAE model involves maximizing the marginal log-likelihood \( \log \ p_\theta (x) = \int p_\theta (x | z) p(z) dz \) which is intractable during optimization due to the integration over the latent space defined by the variables \( z \). VAEs introduce using a variational distribution \( q_\omega (z | x) \) to approximate the posterior while trained by maximizing a lower bound, called Evidence Lower Bound (ELBO), (Kingma and Welling 2013):

\[
\log \ p_\theta (x) \geq \mathbb{E}_{q_\omega (z | x)} [ \log \ p_\theta (x | z)] - KL [q_\omega (z | x) \parallel p (z)] := \mathcal{L}_{\text{ELBO}} (x; \{ \theta, \omega \})
\]

\( \log \ p_\theta (x) \) and \( p(z) = \mathcal{N}(0, I) \) are the decoding and prior distribution, respectively, while \( KL [\cdot | \cdot] \) represents the Kullback–Leibler divergence. Defining a tighter ELBO to the marginal log-likelihood, achieved by using a more expressive posterior (Kim and Pavlovic 2020; Maaloe et al. 2016), importance sampling (Burda, Grosse, and Salakhutdinov 2015; Domke and Sheldon 2018) or through hierarchical variational models (Molchanov et al. 2019; Vahdat and Kautz 2020), has been successful for improving the performance of VAEs. However, these approaches can only guarantee a tight ELBO for learning a single domain and do not work under the lifelong learning (LLL) setting, which involves learning sequentially several tasks associated with different databases. VAEs, similarly to other deep learning methods (Guo et al. 2020), suffer from catastrophic forgetting (French 1999), when learning new tasks, leading to degenerate performance on the previous tasks. One direct way to enable VAEs for LLL is by using Generative Replay (GR) processes (Ramapuram, Gregorova, and Kalousis 2020).

Let us consider a VAE model trained on a sequence of \( t \) tasks. After the \( i \)-th task learning is finished, the GR process allows the model to generate a pseudo dataset \( X^i \) which will be mixed with the incoming data set \( X_{\text{new}} \) to form a joint dataset for the \((i+1)\)-th task learning. Usually, the distribution of \( \{X^i, X_{\text{new}}\} \) does not match the real data distribution exactly and the optimal parameters \( \{\theta^*, \omega^*\} \) are estimated by maximizing ELBO, on samples \( x^* \) drawn from \( \{X^i, X_{\text{new}}\} \). \( \mathcal{L}_{\text{ELBO}}(\cdot) \) is not a tight ELBO in Eq. (1) by using the model’s parameters \( \{\theta^*, \omega^*\} \) which are not optimal for the real sample log-likelihood \( \log p_\theta (x) \) (See Proposition 6 in Appendix-I from SM^2). In this paper, we aim to evaluate the tightness between \( \log p_\theta (x) \) and \( \mathcal{L}_{\text{ELBO}} (x^*; \theta_*, \omega_*) \), by developing an upper bound to the negative marginal log-likelihood, called Lifelong ELBO (LELBO). LELBO involves the discrepancy distance (Mansour, Mohri, and Rostamizadeh 2009) between the target and the evolved source distributions, as well as the resulting accumulated errors when learning each new task. This analysis provides insights into how a VAE model is losing previously learnt knowledge during LLL. We also generalize the proposed theoretical analysis to ENA models, which leads to a novel dynamic expansion graph model (DEGM) enabled with generating graph structures linking the existing components and a newly created component, benefiting on the
transfer learning and the model size reduction. We list our contributions as:

- This is the first research study to develop a novel theoretical framework for analyzing VAE’s forgetting behaviour during LLL.
- We develop a novel generative latent variable model which guarantees the trade-off between the optimal performance for each task and the model’s size during LLL.
- We propose a new benchmark for the probability density estimation task under the LLL setting.

Supplementary materials (SM) and source code are available.

Related Works

Recent efforts in LLL would focus on regularization based methods (Jung, Jung, and Kim 2016; Li and Hoiem 2017), which typically penalize significant changes in the weights when learning new tasks. Other methods rely on memory systems such as using past learned data to guide the optimization (Chaudhry et al. 2018; Guo et al. 2020; Pan et al. 2020), using Generative Adversarial Nets (GANs) or VAEs (Achille et al. 2018; Ramapuram, Gregorova, and Kalousis 2020; Ye and Bors 2021a; Shin et al. 2017; Ye and Bors 2020). We adopt a combination between ENA and GR mechanisms (Rao et al. 2019), or by employing a network expansion mechanism (Rao et al. 2019), or by employing a combination between ENA and GR mechanisms (Ye and Bors 2021a). These methods significantly relieve forgetting but would suffer from informational interference when learning a new task (Riemer et al. 2019).

The tightness on ELBO is key to improving VAE’s performance and one possible way is to use the Importance Weighted Autoencoder (IWELBO) (Burda, Grosse, and Salakhutdinov 2015) in which the tightness is controlled by the number of weighted samples considered. Other approaches would consider as approximate posterior distributions either normalizing flows (Kingma et al. 2016; Rezende and Mohamed 2015), implicit distributions (Mescheder, Nowozin, and Geiger 2017) or hierarchical variational inference (Huang et al. 2019). The IWELBO bound can be used with any of these approaches for further performance improvement (Sobolev and Vetrov 2019). Moreover, online variational inference (Nguyen et al. 2017) has been used in VAEs, but it would require storing past samples for computing the approximate posterior, which is intractable when learning an infinite number of tasks. The tightness of ELBO under LLL was not studied in any of these works.

Preliminary

In this paper, we address a more general lifelong unsupervised learning problem, where task boundaries are provided only during the training. For a given sequence of tasks \(\{T_1, \ldots, T_N\}\) we consider that each \(T_i\) is associated with an unlabeled training set \(Q_i^T\) and an unlabeled testing set \(Q_i^U\). The model only sees a sequence of training sets \(\{Q_1^T, \ldots, Q_N^T\}\) while it is evaluated on \(\{Q_1^U, \ldots, Q_N^U\}\). Let \(\mathcal{X}\) consider the input data space \(\mathcal{X} \in \mathbb{R}^d\) of dimension \(d\), and \(\mathcal{P}\), the probabilistic representation of the testing set \(Q_i^U\). We desire to evaluate the quality of reconstructing data samples \(x \in \mathcal{X}\), by using the square loss (SL) function \(\|x - h(x)\|^2\), where \(h\) is a hypothesis function in a space of hypotheses \(\{h \in \mathcal{H} | \mathcal{H} : \mathcal{X} \rightarrow \mathcal{X}\}\). For the image space, the loss is represented by \(\sum_{i=1}^{d} (x[i] - h(x)[i])^2\), where \([i]\) represents the entry for the \(i\)-th dimension.

Definition 1 (Single model.) Let \(\mathcal{M} = \{f_\omega, g_\theta\}\) be a single model consisting of an encoder \(f_\omega : \mathcal{X} \rightarrow \mathcal{Z}\) for representing \(q_\omega(z|x)\), and a decoder \(g_\theta : \mathcal{Z} \rightarrow \mathcal{X}\) for modelling \(p_\theta(y|x)\). The latent variable \(z = f_\omega^M(x) + f_\gamma^S(x) \odot \gamma\), \(\gamma \sim \mathcal{N}(0, I)\) is reparameterized by the mean \(f_\omega^M(x)\) and variance \(f_\gamma^S(x)\), implemented by a network \(f_\omega(x)\). \((\omega^M, \theta^M)\) are the parameters of the model \(\mathcal{M}^M\), where \(t\) represents the number of tasks considered for training the model. Let \(g_\theta(f_\omega(x)) : \mathcal{X} \rightarrow \mathcal{X}\) be the encoding-decoding process for \(\mathcal{M}\).

Definition 2 (Discrepancy distance.) We implement \(h \in \mathcal{H}\) by \(g_\theta(f_\omega)\) evaluated on the error function \(\mathcal{L} : \mathcal{X} \times \mathcal{X} \rightarrow \mathbb{R}_+\) which is bounded, \(\forall(x, x') \in \mathcal{X}^2, \mathcal{L}(x, x') \leq U\) for some \(U > 0\). We define the error function as the SL function \(\mathcal{L}(x, x') = \|x - x'\|^2\), \(x, x' \in \mathcal{X}\). A risk for \(h(\cdot)\) on the target distribution \(\mathcal{P}_i\) of the \(i\)-th domain (task) is defined as \(R_{\mathcal{P}_i}(h, f_{\mathcal{P}_i}) = \mathbb{E}_{x \sim \mathcal{P}_i}[\mathcal{L}(h(x), f_{\mathcal{P}_i}(x))\), where \(f_{\mathcal{P}_i}\) is the true labeling function for \(\mathcal{P}_i\). The discrepancy distance on two domains \(\mathcal{P}, \mathcal{P}\) over \(\mathcal{X}\) is defined as:

\[
\text{disc}_{\mathcal{L}}(\mathcal{P}, \mathcal{P}) = \sup_{(h, h') \in \mathcal{H}} \{\mathbb{E}_{x \sim \mathcal{P}}[\mathcal{L}(h'(x), h(x))] - \mathbb{E}_{x \sim \mathcal{P}}[\mathcal{L}(h(x), h(x))]\}.
\]

Definition 3 (Empirical discrepancy distance.) In practice, we consider the sample populations of size \(m_{\mathcal{P}}\) and \(m_{\mathcal{P}}\) from the empirical distributions \(\mathcal{P}\) and \(\mathcal{P}\), and the bounds \(U\) and \(U\), corresponding to \(\mathcal{P}\) and \(\mathcal{P}\), respectively. Then, the discrepancy can be estimated by using finite samples:

\[
\text{disc}_{\mathcal{L}}(\mathcal{P}, \mathcal{P}) \leq \text{disc}_{\mathcal{L}}(\hat{\mathcal{P}}, \mathcal{P}) + 8(\text{Re}_U(\mathcal{H}) + \text{Re}_U(\mathcal{H})) + 3M\left(\sqrt{\frac{\log \left(\frac{4}{\delta}\right)}{2m_{\mathcal{P}}} + \sqrt{\frac{\log \left(\frac{4}{\delta}\right)}{2m_{\mathcal{P}}}}\right),
\]

which holds with probability \(1 - \delta, \delta \in (0, 1)\), where \(M > 0\), and \(\text{Re}_U\) is the Rademacher complexity (See Appendix-H from SM). In the following we use \(\text{disc}_{\mathcal{L}}(\cdot)\) to represent the right-hand side (RHS) of Eq. (3).
The Theoretical Framework

Generalization Bounds for a Single Model

Let us consider \( \mathbb{P} \) the approximation distribution for the generated data by \( g_\theta(\cdot) \) of \( \mathcal{M} \), which was trained on a sequence of domains \( \{Q_1, \ldots, Q_T\} \) and \( \mathcal{P}_i \) represents the probabilistic representation for \( Q_i \). Training \( \mathcal{M}^{1:t} \) using GR, for the \((i+1)\)-th task, requires the minimization of \( \mathcal{L}^* \) (implemented as the negative ELBO): 

\[
\mathcal{M}^{i+1} = \arg \min_{\omega_h^{(i+1)}, \theta^{(i+1)}} \mathcal{L}^* \left( \mathbb{P}^{i+1}, \mathbb{P}^i \otimes \hat{\mathcal{P}}_{i+1} \right),
\]

where \( \otimes \) represents the mixing distribution \( \mathbb{P}^i \otimes \hat{\mathcal{P}}_{i+1} \), formed by samples uniformly drawn from both \( \mathbb{P}^i \) and \( \hat{\mathcal{P}}_{i+1} \), respectively. Eq. (4) can be treated as a recursive optimization problem as \( i \) increases from 1 to \( t \). The learning goal of \( \mathcal{M}^{i+1} \) is to approximate the distribution \( \mathbb{P}^{i+1} \approx \mathbb{P}^i \otimes \hat{\mathcal{P}}_{i+1} \) by minimizing \( \mathcal{L}^* \), when learning \((i+1)\)-th task. During the LLL, the errors when the model is tested on the initial tasks \( Q_i \), \( i < t \), would increase, leading to a degenerated performance on its corresponding unseen domains, defined by its performance on the testing set \( Q_i \). One indicator for the generalization ability of a model \( \mathcal{M} \) is to predict its performance on a testing data set by achieving a certain error rate on a training data set (Kuroki et al. 2019). In this paper, we develop a new theoretical analysis that can measure the generalization changes over time. Before we introduce the Generalization Bound (GB) for -ELBO, we firstly define the GB when considering a VAE learning a single task in Theorem 1 and then when learning several tasks in Theorem 2.

Theorem 1 Let \( \mathcal{P}_i \) and \( \hat{\mathcal{P}}_i \) be two domains over \( \mathcal{X} \). Then for \( h^*_P \) = \( \arg\min_\mathcal{H} \mathcal{R}(h, f_P) \) and \( h^*_P \) = \( \arg\min_{h \in \mathcal{H}} \mathcal{R}(h, f_P) \), where \( f_P \in \mathcal{H} \) is the ground truth function under the encoder-decoding process for \( \mathcal{P}_i \), we can define the GB between \( \mathcal{P}_i \) and \( \hat{\mathcal{P}}_i \): 

\[
\mathcal{R} \left( \mathcal{P}_i, h, f_P \right) \leq \mathcal{R} \left( \mathcal{P}_i, h, h^*_P \right) + \text{dis} \left( \mathcal{P}_i, \hat{\mathcal{P}}_i \right) \]

\[
+ \mathcal{R} \left( \mathcal{P}_i, h^-_P, f_P \right) + \mathcal{R} \left( \mathcal{P}_i, h^+_P, f_P \right),
\]

(5)

where the last two terms represent the optimal combined risk denoted by \( \varepsilon \left( \hat{\mathcal{P}}_i, \mathcal{P}_i \right) \), and we have:

\[
\mathcal{R} \left( \hat{\mathcal{P}}_i, h^+_P \right) = \mathcal{E}_x \mathcal{L} \left( h(x), h^*_P (x) \right),
\]

(6)

See the proof in Appendix-A from SM1. We use \( \mathcal{R} \left( \mathcal{P}_i, \hat{\mathcal{P}}_i \right) \) to represent \( \text{dis} \left( \mathcal{P}_i, \hat{\mathcal{P}}_i \right) + \varepsilon \left( \hat{\mathcal{P}}_i, \mathcal{P}_i \right) \). Theorem 1 explicitly defines the generalization error of \( \mathcal{M} \) trained on the source distribution \( \hat{\mathcal{P}}_i \). With Theorem 1, we can extend this GB to ELBO and the marginal log-likelihood evaluation when the source distribution evolves over time.

Theorem 2 For a given sequence of tasks \( \{\mathcal{T}_1, \ldots, \mathcal{T}_t\} \), we derive a GB between the target distribution and the evolved source distribution during the \( t \)-th task learning:

\[
\sum_{i=1}^{t} \mathcal{R} \left( \mathcal{P}_i, h, f_P \right) \leq \mathcal{R} \left( \mathcal{P}_1 \otimes \hat{\mathcal{P}}_i, h, h^*_P \right) \]

\[
+ \mathcal{R} \left( \mathcal{P}_1^{1:t}, \mathbb{P}^{t-1} \otimes \hat{\mathcal{P}}_t \right),
\]

(7)

where \( \mathcal{P}_1^{1:t} \) is the mixture distribution \( \{\mathcal{P}_1 \otimes \mathcal{P}_2, \ldots, \otimes \mathcal{P}_t \} \). See the proof in Appendix-B from SM1.

Remark. Theorem 2 has the following observations:

- The performance on the target domain depends mainly on the discrepancy term even if \( \mathcal{M} \) minimizes the source risk, from the first term of RHS of Eq. (7).
- In the GR process, \( \mathbb{P}^{t-1} \) is gradually degenerated as \( t \) increases due to the repeated retraining (Ye and Bors 2020a), which leads to a large discrepancy distance term.

We also extend the idea from Theorem 2 to derive GBs for GANs, which demonstrates that the discrepancy distance between the target and the generator’s distribution plays an important role for the generalization performance of GANs under the LLL setting, exhibiting similar forgetting behaviour as VAEs (See details in Appendix-G from SM1). In the following, we extend this GB to \( \mathcal{L}^* \).

Lemma 1 Let us consider the random samples \( x^T_i \sim \mathcal{P}_i \) for \( i = 1, \ldots, t \). The sample log-likelihood and its ELBO for all \( \{\mathcal{P}_1, \ldots, \mathcal{P}_t\} \) can be represented by \( \sum_{i=1}^{t} \log p(x^T_i) \) and \( \sum_{i=1}^{t} \mathcal{L}_{\text{ELBO}}(h, x^T_i) \). Let \( x^T_i \) represent the random sample drawn from \( \mathbb{P}^{t-1} \otimes \hat{\mathcal{P}}_t \). We know that \( K L \left( q_{\omega^*}(z | x^T_i) || p(z) \right) \neq K L \left( q_{\omega^*}(z | x^T) || p(z) \right) \) if \( q_{\omega^*}(z | x^T_i) \neq q_{\omega^*}(z | x^T) \), and we have:

\[
\sum_{i=1}^{t} \mathbb{E}_{x^T_i} KL \left( q_{\omega^*}(z | x^T_i) || p(z) \right) \leq \mathbb{E}_{x^T \sim \mathcal{P}_i} KL \left( q_{\omega^*}(z | x^T) || p(z) \right) + K L_1 - K L_2 ,
\]

(8)

\[
\sum_{i=1}^{t} \mathbb{E}_{x^T_i} [ - \mathcal{L}_{\text{ELBO}}(x^T_i; h) ] \leq \mathbb{E}_{x^T \sim \mathcal{P}_i} \left[ - \mathcal{L}_{\text{ELBO}}(x^T; h) \right] + K L_1 - K L_2 ,
\]

(9)

where \( x^T \sim \mathcal{P}_i \) and \( x^T_i \sim \mathbb{P}^{t-1} \otimes \hat{\mathcal{P}}_t \). See the proof in Appendix-C from SM1. We call the RHS of Eq. (9) as Lifelong ELBO (LELBO), denoted as \( \mathcal{L}_{\text{LEBLO}} \) which is a bound for an infinite number of tasks (\( t \to \infty \)). This bound shows the behaviour of \( \mathcal{M} \) when minimizing -ELBO when learning each task. LELBO is also an upper bound to \( - \sum_{i=1}^{t} \mathbb{E}_{x^T_i} \left[ \log p(x^T_i) \right] / t \), estimated by \( \mathcal{M}^t \).

The generalization of LELBO. From Eq. (9), we can generalize LELBO to other VAE variants under LLL, including the auxiliary deep generative models (Maaløe et al. 2016) and hierarchical variational inference (Sobolev and Vetrov 2019) (See details in Appendix-F from SM1). ISELBO bound (Burda, Grosse, and Salakhutdinov 2015) is an extension of ELBO by generating multiple weighted samples.
under the importance sampling (Domke and Sheldon 2018). We generalize the IWELBO bounds to the LLL setting as:

\[
\frac{1}{t} \sum_{i=1}^{t} \mathbb{E}_{x_i \sim P_t} \left[ -\log p \left( x_i^T \right) \right] \leq \sum_{i=1}^{K} \mathbb{E}_{z_i \sim \mathbb{P}^t \otimes \bar{P}_i} \left[ -\log \frac{1}{K} \sum_{i=1}^{K} q (z_i | x) \right] + |KL_1 - KL_2| + \mathcal{R}_A \left( \mathbb{P}(i,t), \mathbb{P}^{t-1} \otimes \bar{P}_i \right),
\]

(10)

See the derivation in Appendix F.1 from SM². We consider \(K'\) weighted importance samples \(\{z_1, \ldots, z_{K'}\}\) (Domke and Sheldon 2018), and omit the subscript for \(q(\cdot)\). We call RHS of Eq. (10) as \(\mathcal{L}_{LELBO_{K'}}\), and \(\mathcal{L}_{LELBO_{K'\leq t}} = \mathcal{L}_{LELBO}\).

**Remark.** We have several conclusions from Eq. (10):

- Based on the assumption that \(\mathbb{P}^{t-1}\) is fixed and \(|KL_1 - KL_2| = 0\), we have \(\mathcal{L}_{LELBO_{K'\leq t}} \leq \mathcal{L}_{LELBO_{K'}}\).
- The tightness of ELBO on \(\mathbb{P}^{t-1} \otimes \bar{P}_i\) (the second term of RHS of Eq. (10)) can not guarantee a tight bound on the testing data log-likelihood since the RHS of Eq. (10) contains the discrepancy distance term and other error terms.

A tight GB can be achieved by reducing the discrepancy distance term by training a powerful generator that approximates the target distributions well, for example by using the Autoencoding VAE (Cemgil et al. 2020) or adversarial learning (Goodfellow et al. 2014), which would fail when learning several different domains due to the fixed model’s capacity and the mode collapse (Srivastava et al. 2017). In the following section, we show how we can achieve a tight GB by increasing the model’s capacity through an expansion mechanism.

### Generalization Bounds for ENA

For a given mixture model \(M = \{M_1, \ldots, M_K\}\), each component \(M_i\) can be trained with GR. In order to assess the trade-off between performance and complexity, we assume that \(\mathbb{P}^{(i,s)}\) is the generator distribution of the \(s\)-th component which was trained on a number of \(t\) tasks. Suppose that the \(j\)-th task was learnt by the \(s\)-th component of the mixture and its approximation distribution \(\mathbb{P}^{(m,s)}\) is formed by the sampling process \(x \sim \mathbb{P}^{(i,s)}\) if \(I_T(x) = j\), where \(I_T : \mathcal{X} \to \mathcal{T}\) is the function that returns the true task label for the sample \(x\), and \(m\) represents the number of times \(M_s\) was used with GR for the \(j\)-th task. We omit the component index \(s\) for \(\mathbb{P}^{(m,s)}\) for the sake of simplification and let \(\mathbb{P}^{(m)}_j\) represent \(\bar{P}_i\). In the following, we derive a GB for a mixture model \(M\) with \(K\) components.

**Theorem 3** Let \(C = \{c_1, \ldots, c_m\}\) represent a set, where each item \(c_i\) indicates that the \(c_i\)-th component \(\{M_{c_i}\}\) is only trained once during LLL. We use \(A = \{a_1, \ldots, a_m\}\) to represent the task label set for \(C\), where \(a_i\) is associated to \(c_i\). Let \(C' = \{c'_1, \ldots, c'_k\}\) represent a set where \(c'_i\) indicates that the \(c'_i\)-th component \(M_{c'_i}\) is trained more than once and is associated with a task label set \(A'_{c'_i} = \{a(i,1), \ldots, a(i,n)\}\). Let \(\bar{C} = \{c(1), \ldots, c(l)\}\) be a set where \(c(i, j)\) denotes the number of times \(M_{c(i,j)}\) was used for \(a(i,j)\)-th task. We have \(|C| + |C'| = K, |A'_{c'_i}| > 1\), where \(K\) is the number of components in the mixture model and \(|\cdot|\) is the cardinality of a set. Let \(\bar{A} = \{\bar{a}_1, \ldots, \bar{a}_l\}\) represent a set where each \(\bar{a}_i\) denotes the number of tasks modelled by the probabilistic representations of the \(c'_i\)-th component \(\bar{a}_i = |A'_{c'_i}|\). We derive the bound for \(M\) during the \(t\)-th task learning as:

\[
\frac{1}{t} \sum_{i=1}^{C} \left\{ \sum_{j=1}^{\bar{a}_i} \right\} \left\{ R_{P_a(i,j)} \left( h_{c'_i}, f_{P_a(i,j)} \right) \right\} + \frac{1}{t} \sum_{i=1}^{C'} \left\{ R_{P_a(i,j)} \left( h_{c'_i}, f_{P_a(i,j)} \right) \right\} \leq \frac{1}{t} R_C - \frac{1}{t} R_{A'}
\]

(11)

\(R_C = \sum_{i=1}^{C} \left\{ \sum_{j=1}^{\bar{a}_i} \right\} \left\{ R_{P_a(i,j)} \left( h_{c'_i}, f_{P_a(i,j)} \right) \right\} + R_A \left( P_{a(i,j)}, \bar{P}_{a(i,j)} \right),
\]

(12)

and \(R_{A'}\) is the accumulated error evaluated by the components that are trained only once:

\[
R_{A'} = \sum_{i=1}^{C'} \left\{ \sum_{j=1}^{\bar{a}_i} \right\} \left\{ R_{P_a(i,j)} \left( h_{c'_i}, f_{P_a(i,j)} \right) \right\} + R_A \left( P_{a(i,j)}, \bar{P}_{a(i,j)} \right),
\]

(13)

and after decomposing the last term it becomes

\[
R_{A'} = \sum_{i=1}^{C'} \left\{ \sum_{j=1}^{\bar{a}_i} \right\} \left\{ R_{P_a(i,j)} \left( h_{c'_i}, f_{P_a(i,j)} \right) \right\} + \sum_{k=1}^{C(i,j)-1} \left\{ R_A \left( P_{a(i,j)}^{(k)}, \bar{P}_{a(i,j)}^{(k)} \right) \right\}.
\]

(14)

The proof is provided in Appendix D from SM².

**Remark.** We have several observations from **Theorem 3**:

- If \(|C'| = 1\) and \(|C| = 0\), then the term \(R_C\) in Eq. (11) would disappear while \(R_{A'}\) would accumulate additional error terms, according to Eq. (14).
- In contrast, if \(|C| = t\), then the GB from Eq. (11) is reduced to \(R_C\), where the number of components \(K\) is equal to the number of tasks and there are no accumulated error terms, leading to a small gap on GB.
- When \(|C|\) increases, the gap on GB tends to be small and the model’s complexity tends to be large because the accumulated error term will be reduced \(|C'| = K - |C|\) in Eq. (14) while \(K\) increases.
- If a single component learns multiple tasks \(|C'| = 1\), then GB on the initial tasks \(a(i,j)\) is small, tends to have more accumulated error terms compared to the GB on the latest given tasks \(a(i,j)\) is large, shown by the number of accumulated error terms \(R_{A'}(\cdot, \cdot)\) in Eq. (14), controlled by \(c(i, j) = t - a(i,j)\).

In the following, we extend GB from \(L\) to \(L^*\).
Lemma 2 We derive a GB for the marginal log-likelihood during the t-th task learning for M:

\[
\frac{1}{t} \sum_{i=1}^{t} \mathbb{E}_{p_i} \left[ -\log p(x_i^t) \right] \leq \frac{1}{t} \left( R_{\Delta}^{li} + R_{\Delta}^{lj} + D_{\text{diff}}^{a} \right) + \frac{1}{t} \sum_{i=1}^{t} \left\{ \sum_{j=1}^{q} \left[ \mathbb{E}_{p_{a(i,j)}} \left[ -\mathcal{L}_{\text{ELBO}} \left( x_{a(i,j)}^t; h_{ci} \right) \right] \right] \right\} + \sum_{i=1}^{q} \mathbb{E}_{p_{a(i,j)}} \left[ -\mathcal{L}_{\text{ELBO}} \left( x_{a(i,j)}^t; h_{ci} \right) \right]
\]

(15)

where we omit the component’s index for each \( \log p(x_i^t) \) for the sake of simplicity, we use \( R_{\Delta}^{li} \) and \( R_{\Delta}^{lj} \) to represent the second terms in the RHS’s from Eq. (12) and (13), respectively, and \( D_{\text{diff}}^{a} \) represents the absolute difference on the KL divergence (details in Appendix-E from SM). Each \( x_{a(i,j)}^t \) is drawn from \( \mathcal{P}_{a(i,j)}[x_{a(i,j)}^t, \pi_{a(i,j)}] \) modelled by the \( c_t \)-th component in M. \( \mathcal{L}_{\text{ELBO}}(x_{a(i,j)}^t; h_{ci}) \) is the ELBO estimated by the \( c_t \)-th component.

Lemma 2 provides an explicit way to measure the gap between ELBO and the model likelihood for all tasks using the mixture model. When \( |C'| = 0, \) \( D_{\text{diff}}^{a} = 0 \) and the discrepancy \( \text{disc}_{\mathcal{L}}^a(\mathcal{P}_{a(i,j)}, \pi_{a(i,j)}) \) is very small, this bound is tight.

Dynamic Expansion Graph Model (DEGM)

According to Theorem 3, achieving an optimal GB requires each mixture component to model a unique task. However, adding dynamically a new component whenever learning a new task, leads to ever-increasing memory and computation requirements. For addressing the trade-off between task learning effectiveness and memory efficiency, we propose a novel mixture expansion mechanism. This would selectively allow the newly created component to reuse some of the parameters and thus transfer information from existing components, according to a knowledge similarity criterion.

Basic and Specific Nodes in DEGM

A component trained during LLL, with independent parameters, is called a basic node and can be transferred to be used in other tasks. Therefore, a basic node can be seen as a knowledge source for other processing nodes in DEGM. Meanwhile, we also have specific nodes associated with the novel information acquired from a new task \( T_{t+1} \), after also considering reusing the information from the basic nodes.

Let \( q_{\omega_i}(z | x) \) and \( p_{\theta_i}(x | z) \) represent the encoding and decoding distributions, respectively, as in Eq. (1). We implement the basic node using paired sub-models, for encoding and decoding information. We consider two sub-inference models, \( f_{\omega_i}: \mathcal{X} \rightarrow \mathcal{Z} \) and \( f_{\omega_i^t}: \mathcal{Z} \rightarrow \mathcal{Z} \) for modelling \( q_{\omega_i}(z | x) \), expressed by \( f_{\omega_i} \circ f_{\omega_i^t}: \mathcal{X} \rightarrow \mathcal{Z} \), where \( \mathcal{Z} \) is an intermediate latent representation space with the dimension larger than \( |Z| > |X| \). We use two networks, \( g_{\theta_i}: \mathcal{X} \rightarrow \mathcal{X} \) and \( g_{\theta_i^t}: \mathcal{X} \rightarrow \mathcal{X} \), for modelling \( p_{\theta_i}(x | z) \) which is expressed by \( g_{\theta_i} \circ g_{\theta_i^t}: \mathcal{Z} \rightarrow X \), where \( \mathcal{X} \) is an intermediate representation space, \( |\mathcal{X}| < |\mathcal{X}| \). Since a basic node has two connectable sub-models \( \{ f_{\omega_i}, g_{\theta_i} \} \), building a specific \( j \)-th node only requires two separate sub-models \( \{ f_{\omega_j^t}, g_{\theta_j} \} \) which would be connected with the sub-models \( \{ f_{\omega_i}, g_{\theta_i} \} \) of all basic nodes, \( i = 1, \ldots, K \) to form a graph structure in DEGM. In the following section, we describe how DEGM expands its architecture during LLL.

Training Sub-Graph Structures in DEGM

Let us assume that we have trained t nodes after learning \( t \) tasks, where \( K \) nodes, \( K < t \), represent basic nodes \( \mathcal{G} = \{ B_1, \ldots, B_K \} \), and \( t - K \) nodes belong to specific nodes \( S = \{ S_1, \ldots, S_{t-K} \} \). Let \( \mathcal{G}(i) \) and \( \mathcal{S}(i) \) be the functions that return the node index for \( \mathcal{G} \) and \( S \). Each \( B_i \in \mathcal{G} \) has four sub-models \( \{ f_{\omega_i}, f_{\omega_i^t}, g_{\theta_i}, g_{\theta_i^t} \} \) where \( i^* = \mathcal{G}(i), \) and each \( S_i \in \mathcal{S} \) has only two sub-models \( \{ f_{\omega_i}, g_{\theta_i} \} \), where \( i^* = \mathcal{S}(i) \). Let us consider \( V \in \mathbb{R}^{t \times t} \) an adjacency matrix representing the directed graph edges from \( S \) to \( \mathcal{G}, \mathcal{V}(i, j) \) is the directed edge from nodes \( i \) to \( j \), and \( \mathcal{V} \) is used for expanding the architecture whenever necessary. After learning \( t \)-th task, we set a new task \( T_{t+1} \) for training the mixture model with \( Q_{t+1}^S \). We evaluate the efficiency of using each element of \( B_i \in \mathcal{G}, i = 1, \ldots, K \) by calculating \( \mathcal{L}_{\text{ELBO}}(S_i, B_i) \) on \( x_i \sim Q_{t+1}^S, j = 1, \ldots, n \), \( (n = 1000 \text{ in experiments}) \). For assessing the novelty of a given task \( T_{t+1} \), with respect to the knowledge already acquired, we consider the following criterion:

\[
k_{S_i} = |\mathcal{L}_{\text{ELBO}}(B_i) - \mathbb{E}_{x_i \sim \mathcal{Q}_{t+1}^S} \mathcal{L}_{\text{ELBO}}(x_i; B_i)|,
\]

(16)

where \( i = 1, \ldots, K \) and \( \mathcal{L}_{\text{ELBO}}(B_i) \) is the best log-likelihood estimated by \( B_i \) on its previously assigned task and we form \( K = \{ k_{S_1}, \ldots, k_{S_K} \} \). Similar log-likelihood evaluations were used for selecting components in (Lee et al. 2020; Rao et al. 2019). However, in our approach we develop a graph-based structure by defining Basic and Specific nodes based on analyzing \( K \), as explained in the following.

Building a Basic node.

A Basic node is added to the DEGM model when the incoming task is assessed as completely novel. If \( \min(K) > \tau \), where \( \tau \) is a threshold, then we set \( V(t+1, \mathcal{G}(i)) = 0, i = 1, \ldots, K \) and DEGM builds a basic node which is added to \( \mathcal{G} \). During the \( (t + 1) \)-th task learning, we only optimize the parameters of the \( (t + 1) \)-th component by using the loss function from Eq. (1) with the given task’s dataset.

Building a Specific node.

A Specific node is built when the incoming task is related to the already learned knowledge, encoded by the basic nodes. If \( \min(K) \leq \tau \), then we update \( \mathbb{V} \) by calculating the importance weight \( V(t + 1, \mathcal{G}(i)) = (w^* - k_{S_i}) / \sum_{j=1}^{K} (w^* - k_{S_j}), w^* = \sum_{i=1}^{K} k_{S_i}, i = 1, \ldots, K, \) where we denote \( \pi_i = V(t + 1, \mathcal{G}(i)) \) for simplification. According to the updated \( \mathbb{V} \), we build a new sub-inference model \( f_{\omega_{t+1}} \), based on a set of sub-models \( \{ f_{\omega_{i}}, i^* = \mathcal{G}(i) \} \), as \( \sum_{i=1}^{K} \pi_{i} f_{\omega_{i}} \circ f_{\omega^t_{i}}(x_{i}) \), which represents \( z = \sum_{i=1}^{K} \pi_{i} z_{i} \), where each \( z_{i} = f_{\omega_{i}} \circ f_{\omega^t_{i}}(x_{i}) \) is weighted by \( \pi_{i} \). In Fig. 1, we show the structure of the decoder, where an identity function implemented by the input layer distributes the latent variable \( z \) to each \( g_{\theta_{i}}(z), i^* = \mathcal{G}(1), \ldots, \mathcal{G}(K), \) leading to \( \bar{x} = \sum_{i=1}^{K} \pi_{i} g_{\theta_{i}}(z) \), where the intermediate feature

8861
Theorem 4. A Specific node is built for learning the \((t+1)\)-th task, which forms a sub-graph structure and can be trained by using a valid lower bound (ELBO) (See details in Appendix-J.1 from SM1):

\[
\mathcal{L}_\text{ELBO}(\mathbf{x}; M_{(t+1)}) =: \\
\mathbb{E}[Q(z) \left[ \log p_{\theta_{(t+1)}}(\mathbf{x} | z) \right] - \sum_{i=1}^{K} \pi_i K L \left( Q_{\hat{\theta}_{GZ(i)}}(z) \left| \mathbf{x}, z \right\rangle \parallel p(z_i) \right),
\]

where \(q_{\hat{\theta}_{GZ(i)}} \circ q_{\theta_{(t+1)}}(z | x)\) is the density function of \(Q_{\hat{\theta}_{GZ(i)}} \circ q_{\theta_{(t+1)}}(z | x)\).

We implement the variational distribution \(Q(z)\) by \(\sum_{i=1}^{K} \pi_i Q_{\hat{\theta}_{GZ(i)}}(z | x)\), which is a mixture inference model. The difference in Eq. (17) from \(q_{\theta_{(t+1)}}(z | x)\) in Eq. (1) is that \(Q(z)\) is much more knowledge expressive by reusing the transferable information from previously learnt knowledge while also reducing the computational cost, by only updating the components \(\{\theta_{(t+1)}, \pi_{(t+1)}\}\) when learning the \((t+1)\)-th task. The first term in the RHS of Eq. (17) is the negative reconstruction error evaluated by a single decoder. The second term consists of the sum of all KL terms weighted by their corresponding edge values \(\pi_i\).

Model selection. We evaluate the negative marginal log-likelihood, (Eq. (1) for Basic nodes, and Eq. (17) for Specific nodes) for testing data samples after LLL. Then we choose the node with the highest likelihood for the evaluation. This mechanism can allow DEGM to infer an appropriate node without task labels (See details in Appendix-J.3 from SM1).

### Table 1: Results for Split MNIST, Split Fashion and COFMI.

<table>
<thead>
<tr>
<th>Methods</th>
<th>S-M</th>
<th>S-F</th>
<th>COFMI</th>
</tr>
</thead>
<tbody>
<tr>
<td>ELBO-GR</td>
<td>-98.23</td>
<td>-240.58</td>
<td>-177.47</td>
</tr>
<tr>
<td>IWELBO-GR-50</td>
<td>-93.57</td>
<td>-236.66</td>
<td>-172.10</td>
</tr>
<tr>
<td>IWELBO-GR-5</td>
<td>-95.80</td>
<td>-238.08</td>
<td>-176.21</td>
</tr>
<tr>
<td>ELBO-GR*</td>
<td>-98.36</td>
<td>-243.91</td>
<td>-180.50</td>
</tr>
<tr>
<td>IWELBO-GR*-50</td>
<td>-91.23</td>
<td>-236.90</td>
<td>-188.9</td>
</tr>
<tr>
<td>CN-DPM*-IWELBO-50</td>
<td>-95.91</td>
<td>-237.47</td>
<td>-184.19</td>
</tr>
<tr>
<td>LIMix-IWELBO-50</td>
<td>-95.74</td>
<td>-237.48</td>
<td>-184.32</td>
</tr>
<tr>
<td>DEGM-ELBO</td>
<td>-93.51</td>
<td>-238.54</td>
<td>-168.91</td>
</tr>
<tr>
<td>DEGM-IWELBO-50</td>
<td>-88.04</td>
<td>-233.76</td>
<td>-163.27</td>
</tr>
<tr>
<td>DEGM-IWELBO-5</td>
<td>-91.44</td>
<td>-235.93</td>
<td>-164.99</td>
</tr>
</tbody>
</table>

Figure 1: The graph structure in DEGM where an image is firstly processed by \(K\) Basic nodes to which the newly created Specific node connects during the inference process. This procedure is also performed at the decoding process.
using the IWELBO bound, for both S-M and S-F settings. The proposed DEGM also outperforms other baselines on COFML, which represents a more challenging task than S-M/S-F. The detailed results for each task are reported in Appendix-K.2 from SM\(^1\), showing that ELBO-GR\(^*\) and IWELBO-GR\(^*\)-50 tend to degenerate the performance on the early tasks under the cross-domain learning setting when compared with VAEs that do not use two stochastic layers. Details, such as the number of Basic and Specific nodes used are provided in Appendix-K.2 from SM\(^1\).

### Comparing to Lifelong Learning Models

**Baselines.** A very strong LLL baseline, called DEGM-2, consists of dynamically creating a new VAE for each new task, trained for achieving the best performance. Meanwhile, Batch Ensemble (BE) (Wen, Tran, and Ba 2020) is designed for classification tasks. We implement each component of BE as a VAE. DEGM is trained by using ELBO without considering the IWELBO bound, aiming for reducing computational complexity. The number of parameters required by various models are provided in Appendix-L.5 from SM\(^1\).

We train various models under CCCOSCZC lifelong learning setting, where each task is associated with one of the datasets: CelebA (Liu et al. 2015), CACD (Chen, Chen, and Hsu 2014), 3D-Chair (Aubry et al. 2014), Omniglot (Lake, Salakhutdinov, and Tenenbaum 2015), ImageNet\(^*\) (Krizhevsky, Sutskever, and Hinton 2012), Car (Yang et al. 2015), Zappos (Yu and Grauman 2017), CUB (Wah et al. 2010) (detailed dataset setting is provided in Appendix-L.1 from SM\(^1\)). The square loss (SL) is used to evaluate the reconstruction quality and the evaluation of other criteria is in Appendix-L.3 from SM\(^1\). The threshold for adding a new component to DEGM, as explained in Section, is \(\tau = 600\) on CCCOSCZC and the results are provided in Table 2. We can observe that the proposed DEGM outperforms other existing lifelong learning models and achieves a close result to DEGM-2 which trains individual VAEs for each task and requires more parameters. Visual results are shown in Appendix-L.6 from SM\(^1\).

### Empirical Results for Theoretical Analysis

We train a VAE on the binarized Caltech 101 database and use it to generate a set of images probabilistically consistent with the dataset. Then, ELBO-GR, IWELBO-GR\(-\Kappa\), where \(\Kappa' \in \{5, 50\}\) corresponds to the number of importance samples used for training on the joint dataset consisting of the generated and a training set from a second task (Fashion).

We evaluate the average target risk (LHS of Eq. (10)) for these models in order to investigate the tightness between the negative log-likelihood (NLL) and LELBO, since NLL is a lower bound to LELBO. The IWELBO bounds considering 5000 weighted importance samples are shown in Fig. 2a. Although, Lemma 1 considers the Gaussian decoder, VAEs with a Bernoulli decoder, corresponding to the IWELBO bound, indicates that IWELBO-GR-50 is a lower bound to IWELBO-GR-5 and ELBO-GR, which empirically proves \(\mathcal{L}_{\text{ELBO}_{50}} \leq \mathcal{L}_{\text{ELBO}_{5}}\) when the generator distribution is fixed and \(|K_{L1} - K_{L2}| = 0\), as discussed in Lemma 1.

We also train a VAE whose decoder outputs the mean vector of a Gaussian distribution with the Identity matrix as its covariance, under MNIST, Fashion and IFashion LLL, where images are greylevel with pixels in \([0, 255]\). The reconstruction error ELBO is normalized by dividing with the image size \((28 	imes 28)\), as in (Park, Kim, and Kim 2019). We evaluate the risk and the discrepancy distance for each training epoch, according to Eq. (9) from Lemma 1 with the results provided in Fig. 2b, where the source risk (the first term in RHS of Eq. (9)) keeps stable and the discrepancy distance \(\text{disc}_{\mathcal{L}}\), Eq. (2), represented within \(\mathcal{R}_{\mathcal{A}}\), increases while learning more tasks. The ‘KL divergence,’ calculated as \(|K_{L1} - K_{L2}|\), shown in Fig. 2b increases slowly. This demonstrates that the discrepancy distance plays an important role on shrinking the gap for the GB. An ablation study, demonstrating the effectiveness of the proposed expansion mechanism, is provided in Appendix-L.4 from SM\(^1\).

### Conclusion

In this paper we analyze the forgetting behaviour of VAEs by finding an upper bound on the negative marginal log-likelihood, called LELBO. This provides insights into the generalization performance on the target distribution when the source distribution evolves continuously over time during lifelong learning (LLL). We further develop a Dynamic Expansion Graph Model (DEGM), which adds new Basic and Specific components to the network, depending on a knowledge novelty criterion during LLL. DEGM can significantly reduce the accumulated errors caused by the forgetting process. The empirical and theoretical results verify the effectiveness of the proposed DEGM methodology.
References


