Article:
Ye, Fei and Bors, Adrian Gheorghe orcid.org/0000-0001-7838-0021 (2025) Evolving Ensemble Model based on Hilbert Schmidt Independence Criterion for task-free continual learning. Neurocomputing. 129370. ISSN: 0925-2312

https://doi.org/10.1016/j.neucom.2025.129370

Evolving Ensemble Model Based on Hilbert Schmidt Independence Criterion for Task-Free Continual Learning

Fei Ye[a], and Adrian G. Bors[b]

[a]School of Information and Software Engineering, University of Electronic Science and Technology of China, Chengdu, China,
[b]University of York, York YO10 5GH, York, UK,

ARTICLE INFO

Keywords:
Lifelong learning
Variational Autoencoders (VAE)
Hilbert Schmidt Independence Criterion
Representation learning

ABSTRACT

Continual Learning (CL) aims to extend the abilities of deep learning models for continuously acquiring new knowledge without forgetting. However, most CL studies assume that task identities and boundaries are known, which is not a realistic assumption in a real scenario. In this work, we address a more challenging and realistic situation in CL, namely the Task-Free Continuous Learning (TFCL), where an ensemble of experts model is trained on non-stationary data streams without having any task labels. To deal with TFCL, we introduce the Evolving Ensemble Model (EEM), which can dynamically build new experts into a mixture of experts for adapting to the changing data distributions while continuously learning new data sets. To ensure a compact network architecture for EEM during training, we propose a novel expansion mechanism that considers the Hilbert-Schmidt Independence Criterion (HSIC) for evaluating the feature space statistical consistency between the knowledge learned by each expert and the given data. This expansion mechanism does not require storing all previous samples and is more efficient as it performs statistical evaluations in the low-dimensional feature space inferred by a deep network. We also propose a new dropout mechanism for selectively removing unimportant stored samples from the memory buffer used for storing the continuously incoming data before being used for training. The proposed dropout mechanism ensures the diversity of information being learnt by the experts from our model. We perform extensive TFCL tests which show that the proposed approach achieves the state of the art. The source code is available in https://github.com/dtuzi123/HSCI-DEM.

1. Introduction

Continual learning, also called lifelong learning, is one of the essential functions in an artificial intelligence system, representing the ability to continuously remember all the previously learned experiences from a sequence of tasks Parisi, Kemker, Part, Kanan and Wermter (2019). Each task in practice is defined by a given data set. Such abilities are inherited in humans and animals enabling them to survive when changing their living conditions and environments, throughout their lives. However, modern deep learning systems usually perform well on single tasks, Erhan, Szegedy, Toshev and Anguelov (2014); Ren, He, Girshick and Sun (2015); Ye and Bors (2021a,b), but suffer dramatic performance loss when trained on several different tasks in a succession, Aljundi, Chakravarty and Tuytelaars (2017); Chen, Ma and Liu (2015); Fagot and Cook (2006); Rannen, Aljundi, Blaschko and Tuytelaars (2017); Tessler, Givony, Zahavy, Mankowitz and Mannor (2017); Yoon, Yang, Lee and Hwang (2017). The reason for the performance loss is the catastrophic forgetting caused by the fact that network's parameters are recalculated when training on new data, Parisi et al. (2019).

Continual Learning (CL) defines a learning paradigm in which a sequence of tasks is streamed, while artificial learning model accesses only the samples characteristic of the currently given task. Given that usually task boundaries are predetermined, there are three types of approaches to relieve forgetting in CL. First, by regularizing the optimization behaviour of the model by adding a penalty term in the objective function. Such approaches require storing some previously learnt samples in a memory buffer to evaluate the importance of the parameters from the previously trained network with respect to the new task. The second type of approaches in CL relies on memory buffers or on training a generator such as a Generative Adversarial Network (GAN) Goodfellow, Pouget-Abadie, Mirza, Xu, Warde-Farley, Ozair, Courville and Bengio (2014), or Variational Autoencoders (VAEs) Kingma and Welling (2013) as a generative replay network. The memory-based approaches prevent forgetting by replaying the samples from the memory buffer or by using generated data. However, due to the

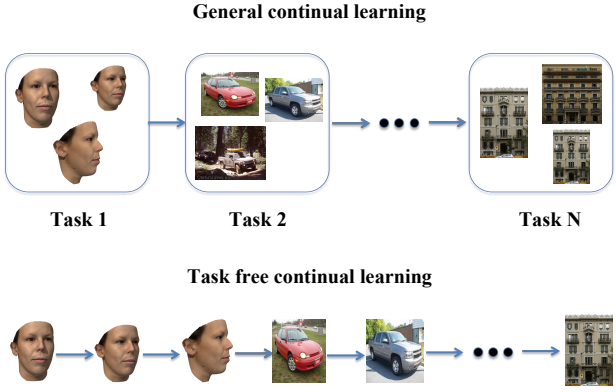


Figure 1: Continual learning under the Task-Free Continual Learning (TFCL) assumption.

ORCID(s):

fixed memory capacity, such approaches are not scalable when the aim is to learn an infinite number of tasks. The dynamic expansion model Ye and Bors (2021c) could solve this problem by dynamically building new layers and hidden nodes to adapt to the new task. However, these approaches still require knowing the task label for creating task-specific modules Ye and Bors (2021c).

In this paper, we address a more challenging and realistic learning environment in CL, namely the Task-Free Continual Learning (TFCL), which assumes that task identities and boundaries are not available during training. The main difference between general continual learning and TFCL is illustrated in Fig. 1, where we can observe from the bottom diagram that under the TFCL the model is trained on non-stationary data streams, with no task labels. A TFCL learning model aims to continuously and effectively adapt to the changing data representing a more realistic artificial intelligence paradigm than other CL assumptions. A popular approach for TFCL is to use a small buffer memory to store incoming samples at each training time, Aljundi, Kelchtermans and Tuytelaars (2019b). Such an approach works well for TFCL when the memory buffer contains a diversity of data samples, Bang, Kim, Yoo, Ha and Choi (2021). However, there are two major drawbacks in the memory-based approaches :1) The model suffers from the negative backward transfer when the memory buffer stores incoming samples which are sufficiently different from the previously learnt ones, Lopez-Paz and Ranzato (2017); 2) It can not address infinite data streams due to the fixed memory capacity.

The drawbacks of existing memory-based approaches are addressed by introducing a new TFCL framework, namely the Evolving Ensemble Model (EEM) which can dynamically build new experts for adapting to the statistical changes in the data while continuously training. First, we implement each expert using a classifier for the prediction task while a VAE is used for the model selection and generative modelling tasks. We then introduce three different dropout mechanisms that regularize the memory buffer by selectively removing certain stored samples to avoid memory overload. In order to promote expert diversity in the EEM, one of the proposed dropout mechanisms introduces using sample log-likelihood evaluation to select the appropriate data to be stored in a memory buffer. In this case the main objective is to ensure that the model has access to a memory buffer whose samples are diverse and novel with respect to the already learned knowledge. Moreover, in order to balance the model complexity and the generalization performance, we introduce a new expansion mechanism that evaluates the Hilbert Schmidt Independence Criterion (HSIC) between the information stored by each expert and the current memory, Gretton, Bousquet, Smola and Schölkopf (2005); Ye and Bors (2022b). HSIC is usually used to evaluate the statistical dependence between a pair of variables while in this paper is employed for evaluating the discrepancy between each trained expert and the data from the current memory buffer by comparing their inferred variables extracted by corresponding VAEs. A new expert is created when the current memory is sufficiently different from the knowledge stored in each trained expert.

The following contributions are brought in this research study:

- We introduce a new TFCL framework, namely the Evolving Ensemble Model (EEM), which learns an infinite number of data streams without forgetting.
- We introduce three different dropout mechanisms that selectively remove certain stored samples from the memory buffer in order to avoid its overload and for promoting statistical diversity among the data generated by the experts.
- We introduce a new expansion mechanism that utilizes the Hilbert Schmidt Independence Criterion to detect data distribution shifts, providing better expansion signals.
- A new theoretical framework for TFCL is developed, which provides new insights into the forgetting behaviour of the continual learning models and theoretical guarantees for the proposed EEM. This is the first work to propose theoretical analysis for TFCL.

The rest of the paper is organized as in the following. Section 2 outlines the main existing approaches in the area of continual learning, while Section 3 introduces the proposed Evolved Ensemble Model (EEM). In Section 4, we provide the theoretical analysis in order to understand the forgetting behaviour of the proposed approach. The experimental results are discussed in Section 5 and the conclusions of this study are drawn in Section 6.

2. Related work

2.1. General continual learning

The usual assumption in continual learning is that after learning a sequence of tasks, the model must recognize all their characteristics. Research studies in the area of continual or lifelong learning can be divided into three branches : regularisation-based such as it was explored in Polikar, Upda, Upda and Honavar (2001); Hinton, Vinyals and Dean (2014); Jung, Ju, Jung and Kim (2018); Kirkpatrick, Pascanu, Rabinowitz, Veness, Desjardins, Rusu, Milan, Quan, Ramalho, Grabska-Barwinska, Hassabis, Clopath, Kumaran and Hadsell (2017); Li and Hoiem (2017); Ren, Wang, Li and Gao (2017); Dai, Yang, Xue and Yu (2007); Nguyen, Li, Bui and Turner (2018); Sun, Yang, Liu, Liu, Xu and Yu (2019); Chen, Chen, Liu, Cao, Zhao, Zhang and Tian (2022), generative replay mechanisms, Achille, Eccles, Matthey, Burgess, Watters, Lerchner and Higgins (2018); Rao, Visin, Rusu, Teh, Pascanu and Hadsell (2019b); Ramapuram, Gregorova and Kalousis (2020); Shin, Lee, Kim and Kim (2017); Zhai, Chen, Tung, He, Nawhal and Mori (2019); Ye and Bors (2021d, 2022a, 2020a,b); Le, Lei, Mu, Zhang, Zeng and Liao (2021) and dynamic extension approaches, Ye and Bors (2021c, 2023, 2022c).

A typical regularization-based approach aims to minimize the changes in those network weights important for past

tasks, when training for a new task. These approaches still require the management of a small memory buffer in which some past samples are stored during training to preserve some of the previously learned information when regularizing the updating of the model. However, such approaches require a significant computational effort when learning a long sequence of tasks.

Generative Replay Mechanisms (GRMs) represent another type of replay approaches, where a generator such as a Variational Autoencoder (VAE) Kingma and Welling (2013) or a Generative Adversarial Network (GAN) Goodfellow et al. (2014) is trained as a generative replay network. After completing the learning of each task, GRMs are able to generate data which are statistically consistent with the previously learned samples. These generative replay samples are then merged with new samples characteristic of the next task to form a joint dataset which is then used for training the model.

The dynamic models usually add new layers with processing nodes Cortes, Gonzalvo, Kuznetsov, Mohri and Yang (2017) within the structure of a neural network or would add a task-specific module into a mixture system, Ye and Bors (2021c); Rao et al. (2019b); Lee, Ha, Zhang and Kim (2020); Wen, Tran and Ba (2020). While the approach from Cortes et al. (2017) is suitable for learning a sequence of tasks from a single domain, the model from Ye and Bors (2021c) is good for learning a sequence of different domains. The study from Wang, Zhou, Ye and Zhan (2022) introduces a new dynamic expansion model that firstly creates a trainable new feature extractor to learn the new task and then eliminate redundant parameters through a feature boosting mechanism. The proposed approach has three main differences from Wang et al. (2022). Firstly, the model from Wang et al. (2022) relies heavily on the task identity information. In contrast, the proposed approach does not require any task information and therefore can be used in task-free continual learning. Secondly, the model from Wang et al. (2022) performs the model expansion when seeing a new task, given its label. In contrast, the proposed model employs an HSIC-based dynamic expansion mechanism, which detects the data distribution shift as the signal for the model expansion. Thirdly, this paper provides a theoretical analysis for continual learning, which is lacking in Wang et al. (2022). The study from Zhou, Wang, Ye and Zhan (2023) introduces a simple but effective approach to deal with continual learning, which can dynamically extend specialized layers to capture new information. Similarly to Wang et al. (2022), the model from Zhou et al. (2023) requires accessing the task information and therefore can not detect data distribution shifts in TFCL.

2.2. Task-free continual learning

Task-free continual learning (TFCL) can be considered a particular form of continual learning that assumes that there are no task boundaries during training, Aljundi, Caccia, Belilovsky, Caccia, Lin, Charlin and Tuytelaars (2019a). TFCL is a more challenging framework because the model accesses only one or a few samples during each training session while all previously visited samples are not available. One of the widely used approaches for TFCL employs a small memory buffer that aims to preserve a few past samples to relieve forgetting Aljundi et al. (2019b). This approach was also extended to using GRM to train both the classifier and a VAE network. In the Maximal Interfered Retrieval (MIR) Aljundi et al. (2019a), a retrieval mechanism is used to select and preserve the most important samples into a memory buffer. The performance of the memory-based approaches in TFCL depends heavily on the quality of the samples stored in memory, and a suitable sample selection criterion is crucial for its performance. The Gradient Sample Selection (GSS) defines the sample selection problem as a constrained optimization reduction, Aljundi, Lin, Goujaud and Bengio (2019c). The Continual Prototype Evolution (CoPE), De Lange and Tuytelaars (2021), represents a learner-evaluator framework, which selects samples with equal probabilities for each category. The Gradient-based Memory EDiting (GMED) Jin, Sadhu, Du and Ren (2021), besides selecting samples also modifies the stored samples to create more appropriate data for training the model.

Another approach to TFCL is based on extending the network architecture or by using mixture models, Ye and Bors (2022b, 2020c). Rao et al. (2019b) proposed the Continual Unsupervised Representation Learning (CURL), Rao, Visin, Rusu, Pascanu, Teh and Hadsell (2019a), which dynamically constructs new inference models in a VAE framework when detecting data distribution shifts, while a GRM is employed to alleviate forgetting. However, CURL still suffers from forgetting due to the frequent updating of the generator. This problem was solved by Lee et al. (2020), by using a pure expansion mechanism called the Continual Neural Dirichlet Process Mixture (CN-DPM), where Dirichlet processes are used to add new VAE components whenever necessary. Although these expansion-based approaches show promising results in TFCL, they do not take into account the previously learned knowledge when evaluating the expansion criterion.

2.3. Generative modelling in continual learning

Generative modelling in CL aims to learn meaningful data representations from a sequence of tasks without forgetting. Most existing continual learning models fail because of lacking an inference mechanism, Ye and Bors (2020a). Using generative modelling in CL was firstly proposed in, Achille et al. (2018), which introduced a new lifelong VAE framework aiming to learn disentangled representations across domains over time. However, alleviating forgetting relies on data samples generated by a VAE module, whose performance would eventually degrade on past tasks, as VAEs tend to produce blurry images. This problem is addressed by introducing a hybrid framework in which GANs are used to generate high-quality generative replay samples, while inference models are trained to learn cross-domain representations from both generative replay samples and the new data, which is referred to as the Lifelong VAE-GAN, Ye and Bors (2020a). Although GANs can provide

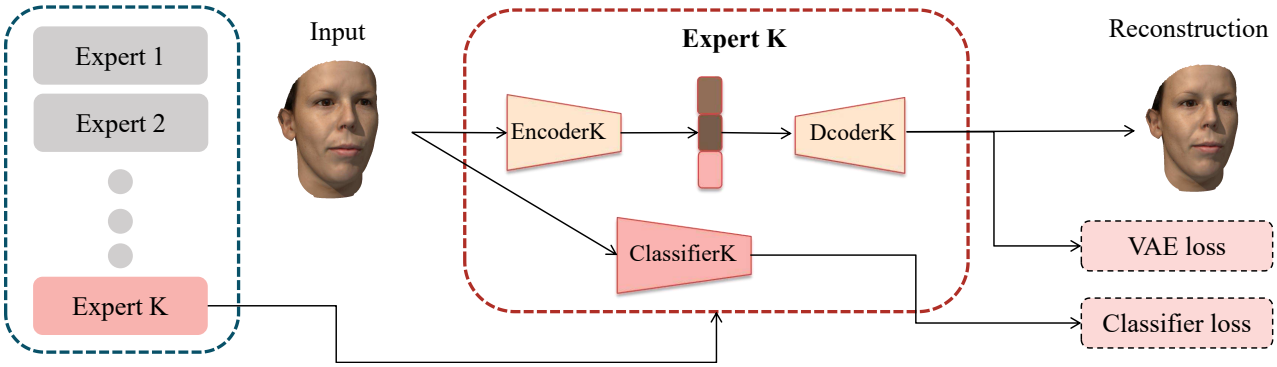


Figure 2: The network architecture for the proposed ensemble model. We assume that we have K experts while only updating the last expert for continuously learning from the given data stream. Each expert consists of a VAE model, that aims to reconstruct images, and a classifier.

high-quality generative replay patterns, their effectiveness is limited when aiming to learn an infinite number of tasks, due to the mode collapse, Srivastava, Valkov, Russell, Gutmann and Sutton (2017). Dynamic expansion models such as CURL, Rao et al. (2019a) and CN-DPM, Lee et al. (2020) are suitable to solve this problem as they can expand their capacity for adapting to new tasks. In this study, we also extend our model for the generative modelling of several tasks under TFCL.

3. The evolving continual learning model

In this section, we introduce our approach in detail. Firsly, we introduce the ensemble model and its network architecture in Section 3.1. Then in Section 3.2 we provide a a new model expansion criterion using the Hilbert Schmith Independence Criterion (HSIC), which also considers a buffer for storing new incoming data. Finally in Section 3.3 we introduce a dropout mechanism for selecting and removing non-essential samples from the memory buffer to avoid memory overload.

3.1. The framework

Before introducing the proposed approach, we first provide the definition of TFCL as follows. Let $\mathcal{D}$ represent a data stream and $\mathcal{T} = \{\mathcal{T}_1, \mathcal{T}_2, \cdots, \mathcal{T}_n\}$ be a set of training steps. We consider n training steps for learning $\mathcal{D}$. Let $\mathcal{D}_i = \{\mathbf{x}_{i,j}, \mathbf{y}_{i,j}\}_{j=1}^{b}$ represent a batch of samples drawn from $\mathcal{D}$, where b is the batch size. The data stream $\mathcal{D}$ is represented by combining n given data batches :

$$\mathcal{D} = \bigcup_{i=1}^{n} \mathcal{D}_i \,. \tag{1}$$

At the j-th training step ($\mathcal{T}_j$), the model only accesses the incoming data $\mathcal{D}_j$ while all previously visited data batches $\{\mathcal{D}_1, \cdots, \mathcal{D}_{j-1}\}$ are not available. After finishing all training steps, we evaluate the model's performance on the entire testing dataset. The maximum number of training steps n depends on the dataset and batch sizes.

The ensemble model has lately become a popular deep learning systems, Hinton et al. (2014); Heo, Lee, Yun and Choi (2019), by considering the predictions by multiple individual components to achieve better generalization performance. However, most ensemble models are only applied in a single domain and cannot handle the task-free continual learning, Phuong and Lampert (2019). In this section, we introduce an ensemble model that can deal with data streams without knowing the task boundaries.

We present the whole network architecture of the proposed ensemble model in Fig. 2. Each expert consists of a VAE and a classifier. The VAE associated with each expert is used for the selection process based on the sample log-likelihood estimated by the VAE, aiming to choose the most appropriate expert for each testing data sample. VAEs also provide appropriate latent codes which can be used for the evaluation of the proposed HSIC-based expansion criterion. Let $f_{\omega_K} : \mathcal{X} \to \mathcal{Z}$ be a neural network parameterized by ω_K, aiming to map an image $\mathbf{x}$ from the input space $\mathcal{X}$ into a low-dimensional feature space $\mathcal{Z}$. Let $f_{\theta_K} : \mathcal{Z} \to \mathcal{X}$ be a neural network parameterized by θ_K, which is used to recover the data from the latent code. Let $f_{\xi_K} : \mathcal{X} \to \mathcal{Y}$ be a classifier with the trainable parameters ξ_K, where $\mathcal{Y}$ is the output space of the model's prediction. In order to avoid building frequently new experts during the training, we introduce a memory buffer, denoted as $\mathcal{G}_j$, updated at the training step ($\mathcal{T}_j$). Then, we evaluate the HSIC-based expansion criterion when the memory buffer is full. The loss function for the VAE aims to maximize the marginal log-likelihood, Kingma and Welling (2013), defined as :

$$\begin{aligned}\mathcal{L}^K_{VAE}(\mathcal{G}_j) \triangleq\ & \mathbb{E}_{q_{\omega_K}(\mathbf{z}\,|\,\mathbf{x})}\Big[\log p_{\theta_K}(\mathbf{x}\,|\,\mathbf{z})\Big] \\ & - D_{KL}\Big[q_{\omega_K}(\mathbf{z}\,|\,\mathbf{x})\,||\,p(\mathbf{z})\Big]\,,\end{aligned} \tag{2}$$

where $p_{\theta_K}(\mathbf{x}\,|\,\mathbf{z})$ and $q_{\omega_K}(\mathbf{z}\,|\,\mathbf{x})$ represent the decoding and encoding distributions, implemented by $f_{\theta_K}(\mathbf{z})$ and $f_{\omega_K}(\mathbf{x})$, respectively, where K represents the number of experts from the ensemble model. We also define the loss for training the classifier on $\mathcal{G}_j$, at the training step ($\mathcal{T}_j$) as :

$$\mathcal{L}^K_{class}(\mathcal{G}_j) \triangleq \frac{1}{|\mathcal{G}_j|}\sum_{t=1}^{|\mathcal{G}_j|}\Big\{\mathcal{L}_{CE}\big(f_{\xi_K}(\mathbf{x}_t), y_t\big)\Big\}\,, \tag{3}$$

where $\mathcal{L}_{CE}(\cdot)$ is the cross-entropy loss function and $|\mathcal{G}_j|$ represents the size of the memory buffer $\mathcal{G}_j$.

3.2. Model expansion

Let $\mathcal{Z}_1$ and $\mathcal{Z}_2$ represent two domains and $\mathbb{P}_{\mathbf{z}_1,\mathbf{z}_2}$ be a joint distribution from which we draw a pair of samples $\{\mathbf{z}_1, \mathbf{z}_2\}$ over $\mathcal{Z}_1 \times \mathcal{Z}_2$. The main goal of HSIC, Gretton et al. (2005), as defined in the Reproducing Kernel Hilbert Space (RKHS), Wang and Li (2018), is to measure the dependence between the domains of the two variables, $\mathbf{z}_1$ and $\mathbf{z}_2$ by evaluating the norm of the cross-covariance operator over the domain $\mathcal{Z}_1 \times \mathcal{Z}_2$. Let Q and B be the RKHSs on $\mathcal{Z}_1$ and $\mathcal{Z}_2$ and $f_Q : \mathcal{Z}_1 \to Q$, and $f_B : \mathcal{Z}_2 \to B$, their feature mappings, respectively. We define the associated reproducing kernels as $k(\mathbf{z}_1, \mathbf{z}_1') = \left\langle f_Q(\mathbf{z}_1), f_Q(\mathbf{z}_1') \right\rangle$ and $l(\mathbf{z}_2, \mathbf{z}_2') = \left\langle f_B(\mathbf{z}_2), f_B(\mathbf{z}_2') \right\rangle$, where $\mathbf{z}_1, \mathbf{z}_1' \in \mathcal{Z}_1$ and $\mathbf{z}_2, \mathbf{z}_2' \in \mathcal{Z}_2$. The cross-covariance operator between f_Q and f_B is defined as :

$$\begin{aligned} C_{\mathbf{z}_1\mathbf{z}_2} =& \mathbb{E}_{\mathbf{z}_1\mathbf{z}_2}\Big\{ \Big(f_Q(\mathbf{z}_1) - \mathbb{E}_{\mathbf{z}_1}\left[f_Q(\mathbf{z}_1) \right] \Big) \otimes \\ & \Big(f_B(\mathbf{z}_2) - \mathbb{E}_{\mathbf{z}_2}\left[f_B(\mathbf{z}_2) \right] \Big) \Big\} \end{aligned} \tag{4}$$

where $\otimes$ is the tensor product. HSIC is defined as the square of the Hilbert-Schmidt norm of $C_{\mathbf{z}_1,\mathbf{z}_2}$:

$$\begin{aligned} \mathcal{L}_{HSIC}(Q, B, \mathbb{P}_{\mathbf{z}_1,\mathbf{z}_2}) &= \left\| C_{\mathbf{z}_1,\mathbf{z}_2} \right\|_{HS}^2 \\ &= \mathbb{E}_{\mathbf{z}_1,\mathbf{z}_1',\mathbf{z}_2,\mathbf{z}_2'}[k(\mathbf{z}_1, \mathbf{z}_1') l(\mathbf{z}_2, \mathbf{z}_2')] \\ &+ \mathbb{E}_{\mathbf{z}_1,\mathbf{z}_1'}[k(\mathbf{z}_1, \mathbf{z}_1')] \mathbb{E}_{\mathbf{z}_2,\mathbf{z}_2'}[l(\mathbf{z}_2, \mathbf{z}_2')] \\ &- 2\mathbb{E}_{\mathbf{z}_1,\mathbf{z}_2}[\mathbb{E}_{\mathbf{z}_1'}[k(\mathbf{z}_1, \mathbf{z}_1')] \mathbb{E}_{\mathbf{z}_2}[l(\mathbf{z}_2, \mathbf{z}_2')]] , \end{aligned} \tag{5}$$

where $\mathbb{E}_{\mathbf{z}_1,\mathbf{z}_1',\mathbf{z}_2,\mathbf{z}_2'}$ represents the expectation over paired samples $\{\mathbf{z}_1, \mathbf{z}_2\}$ and $\{\mathbf{z}_1', \mathbf{z}_2'\}$ drawn from $\mathbb{P}_{\mathbf{z}_1,\mathbf{z}_2}$.

The expansion mechanism is illustrated in Fig. 3, where we assume that we have trained K experts for the ensemble model at $\mathcal{T}_j$. The main idea of the proposed expansion criterion is that if the probabilistic representation of the current memory buffer is different from the probabilistic representations of the trained experts, then we should add a new expert to the ensemble. The new expert is then trained with the data associated with the newly given task. Such a mechanism encourages each component to learn a different underlying data distribution. For this aim, we compare the probabilistic representation of the current memory buffer $\mathcal{G}_j$ with the previously learnt knowledge when evaluating the expansion criterion. Since we can not access all previously learnt samples, we generate pseudo samples using the VAE's decoder from each expert, as a statistical data representation consistent with the previously learnt knowledge. Directly comparing the generated pseudo data and the stored samples in the memory buffer requires substantial computational costs because of the high-dimensionality of the input space. In order to reduce the computation burden we apply the HSIC in the feature space representative of the probabilistic representations. Let $\mathbb{P}_{\tilde{\mathbf{x}}_i}$ represent the distribution of generative replay samples drawn from the VAE model corresponding to the i-th expert. Let $\mathbb{P}_{\tilde{\mathbf{z}}_i}$ be the distribution of the latent variables inferred using the inference model of the i-th expert when considering the samples drawn from $\mathbb{P}_{\tilde{\mathbf{x}}_i}$ and let $\mathbb{P}_{\mathbf{z}_{j,i}}$ represent the distribution of the latent variables inferred by the i-th expert when considering the stored samples from memory $\mathcal{G}_j$ as well. Let $\mathbb{P}_{\tilde{\mathbf{z}}_i,\mathbf{z}_{j,i}}$ represent the joint distribution with marginals $\mathbb{P}_{\mathbf{z}_{j,i}}$ and $\mathbb{P}_{\tilde{\mathbf{z}}_i}$. Then we estimate the HSIC between the knowledge learnt by the i-th expert and the probabilistic representation of the data from the memory buffer at $\mathcal{T}_j$ by $\mathcal{L}_{HSIC}(Q, B, \mathbb{P}_{\tilde{\mathbf{z}}_i,\mathbf{z}_{j,i}})$. The expansion criterion for the ensemble model at $\mathcal{T}_j$ is defined as :

$$\begin{aligned} \min\{ & \mathcal{L}_{HSIC}(Q, B, \mathbb{P}_{\tilde{\mathbf{z}}_1,g_{j,i}}), \cdots, \\ & \mathcal{L}_{HSIC}(Q, B, \mathbb{P}_{\tilde{\mathbf{z}}_{K-1},\mathbf{z}_{j,K-1}}) \} > \lambda, \end{aligned} \tag{6}$$

where λ is a pre-defined threshold controlling the model's expansion. If Eq. (6) holds, we build a new expert into the ensemble model. A large HSIC measure means that the current memory buffer data differs from the already learnt knowledge. In contrast, a small HSIC measure indicates that the current buffer memory is related to the already accumulated knowledge and there is no need to add a new expert.

3.3. Buffer sample dropout

The incoming samples are continuously added to the memory buffer whose statistics is then used to decide whether to train and add new components to the ensemble model. In the following we consider a mechanism for managing the size of the memory buffer, by removing some of the samples from the buffer, according to a sample novelty criterion, in order to free space for other data.

In the following we introduce three different sample dropout mechanisms to regularize the memory capacity. The first sample dropout mechanism consists of using a sliding window for progressively streaming the data samples on a first come first served basis, which removes the earliest stored batch of samples while simply adding the incoming samples to the memory buffer. We call this dropout mechanism as the Evolving Ensemble Model with Sliding Window (EEM-SW). The second dropout mechanism, called EEM-Rand, randomly drops out a batch of samples from the memory buffer $\mathcal{G}_j$ when learning a new task, $\mathcal{T}_j$. In the third sample dropout approach we encourage the diversity among the data samples from the buffer. This approach selectively removes the least diverse stored samples, when considering the already learnt knowledge represented by the statistical representation stored by each EEM component. For this we introduce a discrepancy score for each stored sample, calculated by evaluating the output of each trained expert from the EEM, as :

$$\mathcal{L}_{score}(\tilde{\mathbf{x}}_m) = \sum_{j=1}^{K} \{ -\mathcal{L}_{VAE}^{j}(\tilde{\mathbf{x}}_m) \} , \tag{7}$$

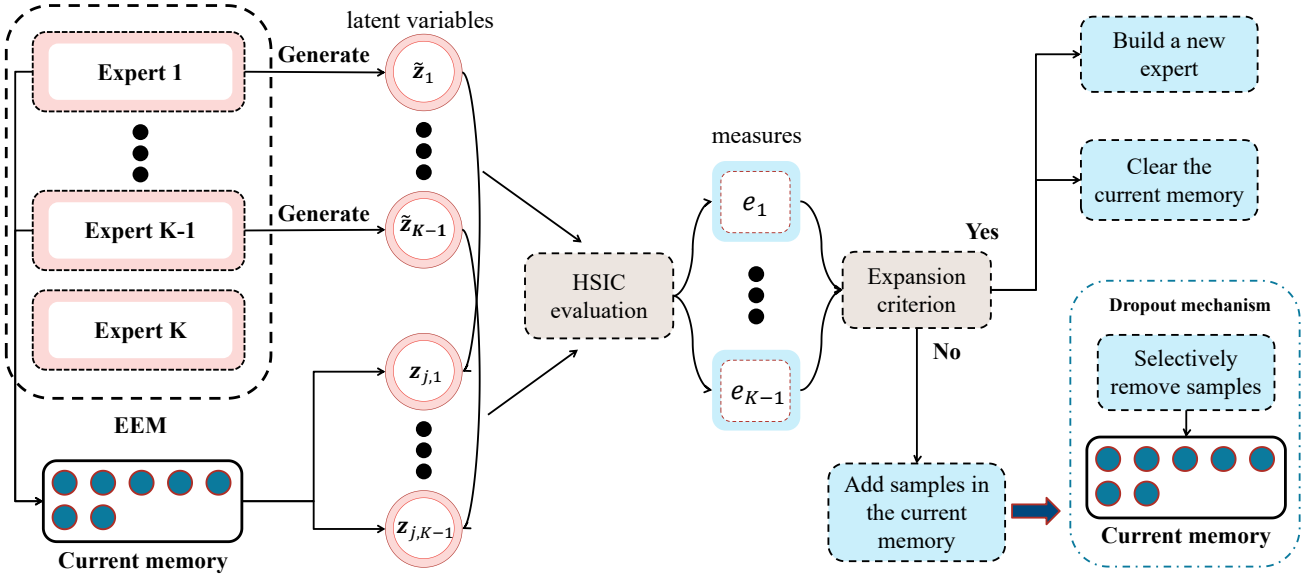


Figure 3: Diagram illustrating the expansion mechanism for the Evolving Ensemble Model (EEM) using the HSIC measure. We assume that the ensemble model has already trained K experts, while considering the VAE decoder of each expert to produce the pseudo samples and then employ the VAE encoder to map them to the latent variable $\mathbf{z}_i$, $i = 1, \ldots, K$. Then we use the VAE encoder of each expert to map the stored samples from the current memory to the latent variables $\tilde{\mathbf{z}}_i$, $i = 1, \ldots, K$. The HSIC evaluation from Eq. (5) is calculated on two sets of latent variables resulting in the evaluations e_i, $i = 1, \ldots, K$. Once all evaluations are finished, we consider $\{e_1, \cdots, e_{k-1}\}$ for Eq. (6) to decide whether the ensemble model should be expanded with a new espert.

where $\{\tilde{\mathbf{x}}_m \in \mathcal{G}_j \,|\, m = 1, \ldots, |\mathcal{G}_j|, |\mathcal{G}_j| \leq |\mathcal{G}|_{max}\}$ are samples stored in the memory buffer, where $|\mathcal{G}|_{max}$ represents the maximum buffer size. Then we use Eq. (7) to rank the data from the buffer according to their novelty for the EEM ensemble model :

$$\mathcal{G}_j^{New} = \{\tilde{\mathbf{x}}_i \in \mathcal{G}_j^{New} \,| \, \mathcal{L}_{score}(\tilde{\mathbf{x}}_1) < \mathcal{L}_{score}(\tilde{\mathbf{x}}_2) < \ldots < \mathcal{L}_{score}(\tilde{\mathbf{x}}_{|\mathcal{G}_j|-N})\}, \tag{8}$$

where we consider that we remove N samples from the memory buffer $\mathcal{G}_j$, creating a new memory buffer $\mathcal{G}_j^{New}$ keeping $|\mathcal{G}_j^{New}| = |\mathcal{G}_j| - N$ samples. Then N new data samples can be added to the memory buffer up to $|\mathcal{G}|_{max}$. We call this method when controlling the memory buffer size as EEM with Selective Dropout (EEM-SD).

3.4. Implementation

In this section, we integrate the overall proposed HSIC-based expansion and dropout mechanisms into a unified algorithm framework aiming to learn a compact Evolving Ensemble Model (EEM) for TFCL. The whole training and testing procedure can be divided into four main steps :

Step 1 (Training phase.) During this stage we train the current expert (K-th expert) on the data samples from the memory buffer $\mathcal{G}_j$. If EEM has only one expert, after having processed $\mathcal{T}_{K_{Max}}$ tasks, where $K_{Max} = 100$ in the experiments, we automatically create a new expert while freezing the weights for the first expert. Then, the first expert is used in the evaluation of the expansion criterion, as in Eq. (6).

Step 2 (Check the model's expansion.) In order to avoid frequently evaluating Eq. (6), we only check the model's expansion when the current memory buffer is full, $|\mathcal{G}_j| = |\mathcal{G}|_{max}$. To check the expansion, we calculate the HSIC measure between the statistics of the current memory buffer and those of the trained experts, according to Eq. (5). Then the mixture model will add a new expert if Eq. (6) is fulfilled. To allow the new expert to learn statistically non-overlapping data, we also clear up the current memory buffer each time when the mixture model is expanded.

Step 3 (Dropout mechanism.) This step aims to avoid memory overload. If the current memory buffer $\mathcal{G}_j$ at $\mathcal{T}_j$ is full, we drop out the least important N samples (in the experiments $N = 10$) from the memory buffer.

Step 4 (Classification at the testing phase.) Once all training steps have been finished, we make the prediction for each testing sample $\mathbf{x}'_j$ using the expert selection process :

$$\hat{s} = \arg\min_{i=1,\cdots,K} \left\{ \mathcal{L}_{VAE}^i(\mathbf{x}'_j) \right\} . \tag{9}$$

Then we make the prediction using the selected expert, expressed as $f_{\xi_{\hat{s}}}(\mathbf{x}'_j)$. We provide the pseudocode in Algorithm 1. During the testing phase, we select and evaluate only the expert having the highest sample log-likelihood for the given testing data sample.

4. Theoretical analysis

In this section, we study the forgetting behaviour of the EEM by defining its underlying learning framework

Algorithm 1: The training algorithm for EEM

Input: All training databases
Output: The model's parameters
for $j \leq n$ **do**
  Sampling process ;
  $\mathcal{D}_j \sim \mathcal{D}$;
  $\mathcal{G}_j = \mathcal{G}_j \cup \mathcal{D}_j$;
  Training step;
  Train the ensemble model on $\mathcal{G}_j$ using Eq. (2) and Eq. (3) ;
  Check the model's expansion ;
  if $|\mathcal{G}_j| \geq |\mathcal{G}|_{max}$ **then**
    Calculate HSIC measures using Eq. (5) ;
    if *satisfy Eq.* (6) **then**
      Build a new expert into the ensemble model ;
    end
  end
  Dropout mechanism ;
  if $|\mathcal{G}_j| \geq |\mathcal{G}|_{max}$ **then**
    Drop out the samples from $\mathcal{G}_j$ according to the dropout mechanism ;
  end
end
Testing phase ;
for $j \leq n$ **do**
  Get the testing sample $\mathbf{x}'_j$;
  $\hat{s} = \arg\min_{i=1,\cdots,K} \left\{ \mathcal{L}^i_{VAE}(\mathbf{x}'_j) \right\}$;
  Make the prediction using the selected expert $f_{\xi_{\hat{s}}}(\mathbf{x}'_j)$;
end

using the Wasserstein distance, Redko, Habrard and Sebban (2017), as a probabilistic measure of comparing two data representations. After introducing some notations and definitions in Section 4.1 we derive the generalization bounds for analyzing the forgetting behaviour for a single model and for the dynamic expansion model in Sections 4.2 and 4.3, respectively.

4.1. Preliminary

Definition 1. ***(Dataset.)*** *Let $\mathcal{X}$ be the n-dimensional input space and $\mathcal{Y}$ represent the output space which is defined within $[0,1]^n$ for the reconstruction task. Let $\mathcal{A}_i^T = \{\mathbf{x}_j^T, y_i^T\}_{j=1}^{N_i^T}$ and $\mathcal{A}_i^S = \{\mathbf{x}_j^S, y_i^S\}_{j=1}^{N_i^S}$ represent the training and testing sets for the i-th dataset, where $\mathbf{x}_j^T \in \mathcal{X}$ and $y_j^T \in \mathcal{Y}$ represent the data (images) and the associated ground truth label, respectively. N_i^T and N_i^S are the total number of samples for $\mathcal{A}_i^T$ and $\mathcal{A}_i^S$, respectively.*

Definition 2. ***(Target distributions.)*** *For a given $\mathcal{A}_i^T$, let us consider that it contains C_i^T data categories, where a category represents a set of data with certain characteristics, and we use $\mathcal{A}_{i,j}^T$ to represent the samples from the j-th category in $\mathcal{A}_i^T$. Let $\mathbb{P}_{\mathcal{A}_{i,j}^T}$ represent the target distribution, characterizing the samples drawn from $\mathcal{A}_{i,j}^T$.*

Definition 3. ***(Memory.)*** *Let $\mathcal{G}_j$ represent a memory buffer updated at the training step ($\mathcal{T}_j$), and $\mathbb{P}_{\mathcal{G}_j}$ the empirical distribution of the stored samples from $\mathcal{G}_j$.*

Definition 4. ***(Model risk.)*** *Let $h \in \mathcal{H}$ be a hypothesis, implemented by a classifier, from a $\mathcal{H}$ class of hypotheses. For a given domain P_i over $\mathcal{X} \times \mathcal{Y}$, we can define the risk as :*

$$\mathcal{R}(h, P_i) = \frac{1}{m} \sum_{k=1}^{m} \mathcal{L}(h(\mathbf{y}_k), \mathbf{x}_k), \tag{10}$$

where $\mathbf{x}_k \sim P_i$, and $\mathcal{L}$ is the loss function implemented by the classification error and $h(\cdot)$ returns the prediction of the given data $\mathbf{x}_k$.

4.2. Forgetting analysis for a model having a single component

In this section, we provide the theoretical framework for analysing the forgetting behaviour when having a model with a single component (expert). The main motivation of the theoretical analysis is to formulate the forgetting problem as a generalization error in the context of domain adaptation, Tzeng, Hoffman, Saenko and Darrell (2017). Unlike the classical domain adaptation theory, which assumes a static source distribution, our analysis deals with dynamically changing source distributions and provides the generalization bounds for each training step during continuous learning.

Theorem 1. *Let $\mathcal{D}$ represent a data stream consisting of the samples drawn from the categories $\mathcal{A}_i^S$, denoted as $\mathcal{D} = \bigcup_{t=1}^{C_i^S} \mathcal{A}_{i,t}^S$. Let $\mathcal{P}_j$ represent the distribution of all visited samples from the data stream $\mathcal{D}$ at $\mathcal{T}_j$. Let U_{x1} and U_{x2} be two sample populations of sizes N_S and N_T, drawn from $\mathcal{P}_j$ and $\mathbb{P}_{\mathcal{G}_j}$, respectively. Let $\tilde{\mathcal{P}}_j$ and $\tilde{\mathbb{P}}_{\mathcal{G}_j}$ represent the empirical probability densities for U_{x1} and U_{x2}, respectively. With the probability of $1-u$, we have the following generalization bound (GB) :*

$$\begin{aligned}\mathcal{R}\left(h, \mathcal{P}_j\right) \leq & \mathcal{R}\left(h, \mathbb{P}_{\mathcal{G}_j}\right) + \mathrm{W}_1\left(\tilde{\mathcal{P}}_j, \tilde{\mathbb{P}}_{\mathcal{G}_j}\right) + \sqrt{2\log\left(\frac{1}{u}\right)/\varsigma'} \\ & \left(\sqrt{\frac{1}{N_S}} + \sqrt{\frac{1}{N_T}}\right) + \mathcal{R}_\lambda,\end{aligned} \tag{11}$$

where $\sqrt{2} > \varsigma' > 0$, and $\mathcal{R}_\lambda$ represents the combined error :

$$\mathcal{R}_\lambda = \mathcal{R}(h^*, \mathcal{P}_j) + \mathcal{R}(h^*, \mathbb{P}_{\mathcal{G}_j}) \tag{12}$$

achieved by the optimal hypothesis h^* that minimizes this error when considering the Wasserstein distance $\mathrm{W}_1(\cdot)$as a metric between two statistical representations. The detailed proof is provided in Redko et al. (2017).

Proof. We prove Theorem 1 by the following derivations :

$$\begin{aligned}
\mathcal{R}\left(h,\mathcal{P}_j\right) &\leq \mathcal{R}\left(h^*,\mathcal{P}_j\right) + \mathcal{R}\left(h^*,h,\mathcal{P}_j\right)\\
\mathcal{R}\left(h,\mathcal{P}_j\right) &\leq \mathcal{R}\left(h^*,\mathcal{P}_j\right) + \mathcal{R}\left(h^*,h,\mathbb{P}_{\mathcal{G}_j}\right) + \mathcal{R}\left(h^*,h,\mathcal{P}_j\right)\\
&\quad - \mathcal{R}\left(h^*,h,\mathbb{P}_{\mathcal{G}_j}\right)\\
\mathcal{R}\left(h,\mathcal{P}_j\right) &\leq \mathcal{R}\left(h^*,\mathcal{P}_j\right) + \mathcal{R}\left(h^*,h,\mathbb{P}_{\mathcal{G}_j}\right) + \mathrm{W}_1\left(\mathcal{P}_j,\mathbb{P}_{\mathcal{G}_j}\right)\\
\mathcal{R}\left(h,\mathcal{P}_j\right) &\leq \mathcal{R}\left(h^*,\mathcal{P}_j\right) + \mathcal{R}\left(h^*,\mathbb{P}_{\mathcal{G}_j}\right) + \mathcal{R}\left(h,\mathcal{P}_{\mathcal{G}_j}\right)\\
&\quad + \mathrm{W}_1\left(\mathcal{P}_j,\mathbb{P}_{\mathcal{G}_j}\right)\\
\mathcal{R}\left(h,\mathcal{P}_j\right) &\leq \mathcal{R}\left(h,\mathbb{P}_{\mathcal{G}_j}\right) + \mathrm{W}_1\left(\mathcal{P}_j,\mathbb{P}_{\mathcal{G}_j}\right) + \mathrm{W}_1\left(\tilde{\mathcal{P}}_j,\tilde{\mathbb{P}}_{\mathcal{G}_j}\right)\\
&\quad + \mathcal{R}_\lambda\\
\mathcal{R}\left(h,\mathcal{P}_j\right) &\leq \mathcal{R}\left(h,\mathbb{P}_{\mathcal{G}_j}\right) + \mathrm{W}_1\left(\tilde{\mathcal{P}}_j,\tilde{\mathbb{P}}_{\mathcal{G}_j}\right)\\
&\quad + \sqrt{2\log\left(\frac{1}{u}\right)/\varsigma'}\left(\sqrt{\frac{1}{N_S}} + \sqrt{\frac{1}{N_T}}\right) + \mathcal{R}_\lambda.
\end{aligned} \tag{13}$$

Similarly to Redko et al. (2017), the first and fourth rows are derived by using the property of the triangular inequality for the error function $\mathcal{R}(\cdot)$. Meanwhile, the following derivations are obtained according to the properties of the Wasserstein metric.

From Theorem 1, we observe that the model would have a tight GB during the initial learning stages since the memory buffer $\mathcal{G}_s$ can store all previously visited samples at $\mathcal{T}_s$ (s is small). However, as the number of training steps increases, the GB would increase since the memory buffer discards past samples loosing their associated information. This phenomenon corresponds to the forgetting process of the model. In the following, we extend Theorem 1 to derive GB for analyzing the generalization performance of a model on the testing dataset.

Theorem 2. *Let $\mathcal{D}$ be a data stream consisting of samples drawn from the categories $\mathcal{A}_i^S$ and let $\mathbb{P}_{\mathcal{A}_{i,t}^T}$ be their associated target distribution. Let $\mathbb{P}_{\mathcal{G}_j}$ represent the distribution of the stored samples from the memory buffer $\mathcal{G}_j$, updated at $\mathcal{T}_j$. Let $U_{\mathcal{A}_{i,t}^T}$ and $U_{\mathcal{G}_j}$ be two sample populations of sizes $N_{\mathcal{A}_{i,t}^T}$ and $N_{\mathcal{G}_j}$, drawn from $\mathbb{P}_{\mathcal{A}_{i,t}^T}$ and $\mathbb{P}_{\mathcal{G}_j}$, respectively. Let $\tilde{\mathcal{P}}_{\mathcal{A}_{i,t}^T}$ and $\tilde{\mathbb{P}}_{\mathcal{G}_j}$ represent the probability distributions for $U_{\mathcal{A}_{i,t}^T}$ and $U_{\mathcal{G}_j}$, respectively. With the probability of $1-u$, we have the following GB :*

$$\begin{aligned}
\mathcal{R}(h,\mathbb{P}_{\mathcal{A}_{i,t}^T}) \leq\ & \mathcal{R}\left(h,\mathbb{P}_{\mathcal{G}_j}\right) + \mathrm{W}_1\left(\tilde{\mathbb{P}}_{\mathcal{A}_{i,t}^T},\tilde{\mathbb{P}}_{\mathcal{G}_j}\right)\\
&+ \sqrt{2\log\left(\frac{1}{u}\right)/\varsigma'}\left(\sqrt{\frac{1}{N_{\mathcal{A}_{i,t}^T}}} + \sqrt{\frac{1}{N_{\mathcal{G}_j}}}\right)\\
&+ \mathcal{R}_\lambda\left(\mathbb{P}_{\mathcal{A}_{i,t}^T},\mathbb{P}_{\mathcal{G}_j}\right),
\end{aligned} \tag{14}$$

where $\mathcal{R}_\lambda(\mathbb{P}_{\mathcal{A}_{i,t}^T},\mathbb{P}_{\mathcal{G}_j})$ is the optimal combined error :

$$\mathcal{R}_\lambda(\mathbb{P}_{\mathcal{A}_{i,t}^T},\mathbb{P}_{\mathcal{G}_j}) = \mathcal{R}(h^*,\mathbb{P}_{\mathcal{A}_{i,t}^T}) + \mathcal{R}(h^*,\mathbb{P}_{\mathcal{G}_j}), \tag{15}$$

achieved by the optimal hypothesis h^*.

The proof is similar to that for Theorem 1. From Theorem 2, we observe that the generalization performance of the model on the target distribution relies on the Wasserstein distance between the memory distribution $\mathbb{P}_{\mathcal{G}_j}$ and the target distribution $\mathbb{P}_{\mathcal{A}_{i,t}^T}$. In a continual learning process defined by a class-incremental setting, we usually have multiple target distributions, each representing the distribution of samples from a certain category. In the following, we derive a GB to analyze the generalization performance on multiple target distributions.

Lemma 1. *For a given data stream $\mathcal{D}$ consisting of samples drawn from $\mathcal{A}_i^S$, let $\{\mathbb{P}_{\mathcal{A}_{i,1}^T},\cdots,\mathbb{P}_{\mathcal{A}_{i,C_i^T}^T}\}$ represent a set of C_i^T target distributions. The GB for a single model at $\mathcal{T}_j$ is expressed as :*

$$\begin{aligned}
\sum_{t=1}^{C_i^T}\left\{\mathcal{R}\left(h,\mathbb{P}_{\mathcal{A}_{i,t}^T}\right)\right\} \leq \sum_{t=1}^{C_i^T}\Bigg\{ & \mathcal{R}\left(h,\mathbb{P}_{\mathcal{G}_j}\right) + \mathrm{W}_1\left(\tilde{\mathbb{P}}_{\mathcal{A}_{i,t}^T},\tilde{\mathbb{P}}_{\mathcal{G}_j}\right)\\
&+ \sqrt{2\log\left(\frac{1}{u}\right)/\varsigma'}\left(\sqrt{\frac{1}{N_{\mathcal{A}_{i,t}^T}}} + \sqrt{\frac{1}{N_{\mathcal{G}_j}}}\right)\\
&+ \mathcal{R}_\lambda\left(\mathbb{P}_{\mathcal{A}_{i,t}^T},\mathbb{P}_{\mathcal{G}_j}\right)\Bigg\},
\end{aligned} \tag{16}$$

The proof simply sums up the GBs between each target and the memory distributions according to Theorem 1.

Remark. We have several observations from Lemma 1 :

- A single model can improve its performance by minimizing the Wasserstein distance between the target and the memory buffer distributions.
- The diversity in the memory buffer plays an important role for the generalization performance of a single model. For instance, if the memory buffer $\mathcal{G}_j$ loses all initial samples, the GB is likely to be large, leading to a degenerated performance on $\mathbb{P}_{\mathcal{A}_{i,t}^T}$, since $\mathrm{W}_1(\tilde{\mathbb{P}}_{\mathcal{A}_{i,t}^T},\tilde{\mathbb{P}}_{\mathcal{G}_j})$ is increased.
- A single component model would anyway suffer from the negative backward transfer even if the memory buffer stores diverse samples.

In conclusion, a model with a single component can not address well the modelling of multiple target distributions due to its limited model capacity and fixed memory size. In the following section, we address these shortcomings by means of the proposed dynamic expansion model in the context of the Task Free Continual Learning (TFCL).

4.3. Forgetting analysis in a dynamic expansion model

In this section, we derive the GB for analyzing the forgetting behaviour of the dynamic expansion model. Firstly, we provide the necessary notations and definitions as follows.

Definition 5. ***(Dynamic expansion model.)*** *We define a dynamic expansion model as* $\mathrm{M} = \{\mathcal{M}_1, \cdots, \mathcal{M}_K\}$ *which consists of* K *experts, where* $\mathcal{M}_i$, $i = 1, \ldots, K$ *is an expert component in a VAE mixture model.*

Definition 6. ***(Memory distributions.)*** *When adding a new expert to the proposed EEM model, the current memory buffer* $\mathcal{G}$ *will be used for training the new expert and afterwards will be emptied. The aim is to accumulate diverse data in* $\mathcal{G}$*, for encouraging the assimilation of a variety of information when training the new expert. Let* $\mathrm{T} = \mathcal{T}_{k_1}, \cdots, \mathcal{T}_{k_K}\}$ *be a set of training steps, each associated with a task, corresponding to a set of buffer memories* $\{\mathcal{G}_{k_1}, \cdots, \mathcal{G}_{k_K}\}$ *at different times, where each buffer is loaded with data during each training step* $\mathcal{T}_{k_i}$*. We consider that the* i*-th expert is trained on the memory buffer* $\mathcal{G}_{k_i}$*.*

Theorem 3. *Let* $\mathcal{D}$ *represent a data stream consisting of data categories (each category represents a set of samples with certain characteristics) from a dataset* $\mathcal{A}_i^S$*. Let* $\mathcal{P}_j$ *represent the distribution of all visited samples from the data stream* $\mathcal{D}$ *at the training step* $\mathcal{T}_j$*. We assume that we have trained* K *experts at* $\mathcal{T}_j$*. With the probability of* $1-u$*, we have the following GB at* $\mathcal{T}_j$ *:*

$$\mathcal{R}\left(h, \mathcal{P}_j\right) \leq \mathrm{F}_{Sel}(\mathcal{P}_j), \tag{17}$$

where $\mathrm{F}_{Sel}(\cdot)$ *is the selection function defined as :*

$$\begin{aligned}\mathrm{F}_{Sel}(\mathcal{P}_j) = \min_{i=1}^{K} \Big\{ & \mathcal{R}\left(h, \mathbb{P}_{\mathcal{G}_j}\right) + \mathrm{W}_1\left(\tilde{\mathcal{P}}_j, \tilde{\mathbb{P}}_{\mathcal{G}_{k_i}}\right) \\ & + \sqrt{2\log\left(\frac{1}{u}\right)/\varsigma'}\left(\sqrt{\frac{1}{N_{\mathcal{P}_j}}} + \sqrt{\frac{1}{N_{\mathcal{G}_{k_i}}}}\right) \\ & + \mathcal{R}_\lambda\left(\mathcal{P}_j, \mathbb{P}_{\mathcal{G}_{k_i}}\right)\Big\}.\end{aligned} \tag{18}$$

Usually, we perform the model selection for a given batch of samples instead of the whole training set, allowing a quicker online training procedure for our model. Therefore, we divide $\mathcal{P}_j$ *into* j *parts* $\{\mathcal{P}_j^1, \cdots, \mathcal{P}_j^j\}$*, where each part represents the distribution of a batch of samples with certain characteristics (for example category-like information), learnt at* $\mathcal{T}_j$*. The,n we rewrite Eq.* (17) *as :*

$$\sum_{t=1}^{i}\left\{\mathcal{R}\left(h, \mathcal{P}_j^t\right)\right\} \leq \sum_{t=1}^{i}\left\{\mathrm{F}_{Sel}(\mathcal{P}_j^t)\right\}. \tag{19}$$

Eq. (19) provides a tighter GB in a mixture model, when compared to that of a model with a single expert component, whose GB is provided in Eq. (11) as defined by Theorem 1, because of its selection process that always selects a tight GB when training a new component at $\mathcal{T}_j$. In the following, we derive the GB for the dynamic expansion model in order to assess its generalization performance on multiple target distributions.

Table 1
The classification accuracy when considering five independent runs for various models, where '*' and '†' represent the results cited from De Lange and Tuytelaars (2021) and Jin et al. (2021), 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	-	-
CN-DPM*	93.23 ± 0.09	45.21 ± 0.18	20.10 ± 0.12
EEM-SW	96.79 ± 0.11	58.81 ± 0.12	22.33 ± 0.15
EEM-Rand	96.73 ± 0.12	56.09 ± 0.15	21.78 ± 0.16
EEM-SD	**97.02** ± 0.17	**59.03** ± 0.23	**23.26** ± 0.14

Table 2
The number of parameters of the proposed approach when considering various buffer sample selection approaches.

Methods	Split MNIST	Split CIFAR10	Split CIFAR100	Split M
EEM-SW	21	25	7	7
EEM-Rand	21	25	7	7
EEM-SD	21	25	7	7

Lemma 2. *For a given data stream* $\mathcal{D}$ *consisting of samples drawn from* $\mathcal{A}_i^S$*, let* $\{\mathbb{P}_{\mathcal{A}_{i,1}^T}, \cdots, \mathbb{P}_{\mathcal{A}_{i,C_i^T}^T}\}$ *represent a set of target probability distributions associated with* C_i^T *categories of data. The GB for the dynamic expansion model EEM with several experts, at* $\mathcal{T}_j$*, is expressed as :*

$$\sum_{t=1}^{C_i^T}\left\{\mathcal{R}\left(h, \mathbb{P}_{\mathcal{A}_{i,t}^T}\right)\right\} \leq \sum_{t=1}^{C_i^T}\left\{\mathrm{F}_{Sel}(\mathbb{P}_{\mathcal{A}_{i,t}^T})\right\}. \tag{20}$$

Remark. We have several observations from Lemma 2 :

- To compare with a single model (See Lemma 1), the dynamic expansion model achieves a tighter GB (Eq. (20))

Table 3
The number of parameters of the proposed approach when considering various buffer sample selection approaches.

Methods	Split MNIST	Split CIFAR10	Split CIFAR100	Split M
EEM-SW	2.1×10^7	5.3×10^8	1.5×10^8	1.5×10^8
EEM-Rand	2.1×10^7	5.3×10^8	1.5×10^8	1.5×10^8
EEM-SD	2.1×10^7	5.3×10^8	1.5×10^8	1.5×10^8

since it involves the selection process that ensures a tight GB for each target distribution.

- Information diversity, accumulated by each expert in the dynamic expansion model, plays an important role in the generalization performance. The proposed EEM can achieve this in two ways. First, the proposed expansion mechanism evaluates the probability distance between the incoming samples and the already learned knowledge. A new expert would then be added whenever the information provided is completely new when compared to that already accumulated by the existing experts. Second, we empty the memory buffer after creating a new expert, in order to be filled with new incoming data.
- When compared to the theoretical analysis from Ye and Bors (2021c), Lemma 2 provides a measure of information loss for a dynamic expansion model at each training step under TFCL, bridging the gap between the theoretical analysis and the implementation of TFCL.

5. Experimental results

In the following we provide the empirical results for the proposed expanding mixture model for the Task Free Continual Learning (TFCL).

5.1. Experiment setting

First, we introduce several well-known TFCL benchmarks.

- **Split MNIST** splits MNIST, LeCun, Bottou, Bengio and Haffner (1998), which contains 60k training images, into five tasks according to the category, with each category corresponding to images representing 2 different digits, corresponding to 2 classes, as in De Lange and Tuytelaars (2021).
- **Split CIFAR10** divides CIFAR10, Krizhevsky and Hinton (2009), into five tasks, where each task consists of samples from two different classes, De Lange and Tuytelaars (2021).
- **Split CIFAR100** divides CIFAR100 into 20 tasks, where each task has 2500 examples from five different categories, Lopez-Paz and Ranzato (2017).
- **Split MImageNet** splits MINI-ImageNet into 20 disjoint subsets, where each subset contains images from five classes, Aljundi et al. (2019a).

Table 4
The average classification accuracy from 20 runs on complex datasets.

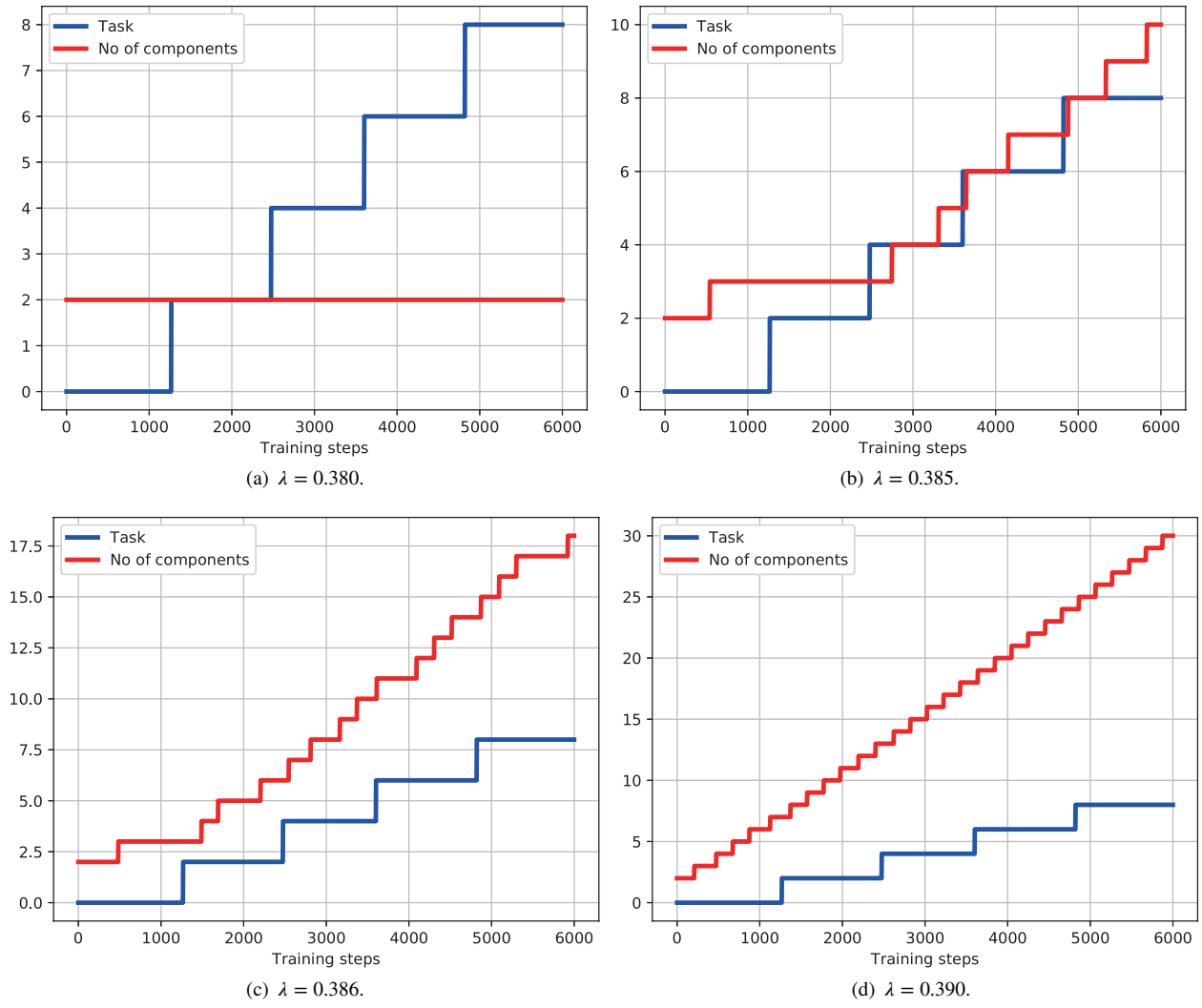
Methods	Split MImageNet	Permuted MNIST
ER_a	25.92 ± 1.2	78.11 ± 0.7
ER + GMED	27.27 ± 1.8	78.86 ± 0.7
MIR+GMED	26.50 ± 1.3	79.25 ± 0.8
MIR	25.21 ± 2.2	79.13 ± 0.7
EEM-SW	**28.90** ± 1.1	**80.32** ± 0.6
EEM-Rand	27.23 ± 1.2	80.28 ± 0.5

Table 5
Average classification accuracy for five independent runs for various models over data streams with fuzzy task boundaries.

Methods	Split MNIST	Split CIFAR10	Split MImageNet
Vanilla	21.53 ± 0.1	20.69 ± 2.4	3.05 ± 0.6
ER	79.74 ± 4.0	37.15 ± 1.6	26.47 ± 2.3
MIR	84.80 ± 1.9	38.70 ± 1.7	25.83 ± 1.5
ER + GMED	82.73 ± 2.6	40.57 ± 1.7	28.20 ± 0.6
MIR+GMED	86.17 ± 1.7	41.22 ± 1.1	26.86 ± 0.7
EEM-SW	**96.78** ± 1.5	57.32 ± 1.3	**28.63** ± 0.8
EEM-Rand	96.63 ± 1.3	**58.27** ± 1,5	28.48 ± 0.9

In the following we introduce several baselines that are used in our experiments.

- **Finetune** trains a single classifier directly on incoming samples under TFCL.
- **Gradient Episodic Memory (GEM)** is a memory-based approach that employs a small memory buffer to store a few past samples, Lopez-Paz and Ranzato (2017).
- **Incremental Classifier and Representation Learning (iCARL)** is a standard memory-based method used in a class incremental setup, Rebuffi, Kolesnikov, Sperl and Lampert (2017).
- **reservoir*, Vitter (1985)**, is a simple memory-based approach that stores the observed samples into a memory buffer $\mathcal{M}$ with the random selection process.
- **Maximally Interfered Retrieval (MIR)**, introduces a retrieval strategy for the sample selection in the memory buffer during TFCL, Aljundi et al. (2019a).
- **Gradient based Sample Selection (GSS)**, employs a specific mechanism for the sample selection which are stored in a buffer requiring class labels, which can not be applied in an unsupervised learning setting, Aljundi et al. (2019c).



(a) $\lambda = 0.380$. (b) $\lambda = 0.385$.

(c) $\lambda = 0.386$. (d) $\lambda = 0.390$.

Figure 4: The investigation of the data distribution shift and the change in the number of components in EEM-SW when changing λ in Eq. (6).

- **Gradient based Memory EDiting (GMED)**, stores examples in the memory buffer via gradient updates, to build more "challenging" samples for replay. GMED can also be combined with other CL methods, Jin et al. (2021).

Network architecture. We consider the Importance Weighted Autoencoder from Burda, Grosse and Salakhutdinov (2015) for implementing the VAE model of each expert. For Split MNIST the inference and generator models are implemented by two fully connected layers, each with 200 hidden units. For CIFAR10 and CIFAR100, we implement the encoder of the VAE model using a fully connected network with six layers of $\{2000, 1500, 1000, 600, 300, 200\}$, while for the decoder of the VAE model we use a fully connected network with six layers of $\{200, 300, 600, 1000, 1500, 2000\}$. The dimension of the latent variable is 200. The GPU used for the experiments is a GeForce GTX 1080 while using Linux Ubuntu 18.04.5 as the operating system.

Network and hyperparameters. We adapt ResNet18, He, Zhang, Ren and Sun (2016), as the classifier for Split CIFAR10 and Split CIFAR100 according to the setting from De Lange and Tuytelaars (2021). For Split MNIST, we consider a Multilayer Perceptron (MLP) network with 2 hidden layers of 400 units each, De Lange and Tuytelaars (2021), as the classifier. We set the maximum memory buffer size for Split MNIST, Split CIFAR10, Split CIFAR100 as 2000, 1000 and 5000, respectively. For each training step, a model would only accesses a batch of 10 samples while all previously batches are not available. For the expansion criterion from Eq. (6), we consider $\lambda = \{0.385, 0.397, 0.4, 0.42\}$ for Split MNIST, Split CIFAR10, Split CIFAR100 and Split MImageNet datasets.

5.2. Results on TFCL benchmarks

Firstly, we evaluate various models under Split MNIST, Split CIFAR10, and Split CIFAR100, and the results are reported in Tab. 1, where we compare with several baselines including : "Finetune" that directly trains a classifier on the data stream, GSS, Aljundi et al. (2019c), MIR, Aljundi et al.

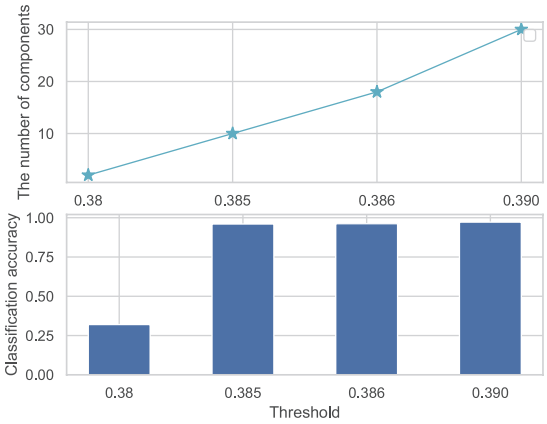


Figure 5: The number of components (top) and performance (bottom) for EMM-SW when tuning λ from Eq. (6) for the lifelong learning of Split MNIST.

(2019a), GEM, Lopez-Paz and Ranzato (2017), iCARL, Rebuffi et al. (2017), reservoir Vitter (1985), CURL, Rao et al. (2019a), CN-DPM, Lee et al. (2020), CoPE, De Lange and Tuytelaars (2021), ER + GMED and ER_a + GMED, Jin et al. (2021) where ER is the experience replay, Rolnick, Ahuja, Schwarz, Lillicrap and Wayne (2019) and ER_a is ER with data augmentation. The number of experts reached by the proposed Evolving Ensemble Model (EEM) model for Split MNIST, Split CIFAR10 and Split CIFAR100 is of 21, 25 and 7, respectively. The results from Tab. 1 show that the proposed approach outperforms not only single component models, including GSS, CoPE, and GMED, but also the dynamic expansion model CN-DPM, Lee et al. (2020), on all three datasets. We provide the number of experts and parameters in Tab. 2 and Tab. 3, respectively.

5.3. Continuous learning results on datasets containing more complex images

We also investigate the performance of various models on the large-scale dataset, MINI-ImageNet, Le and Yang (2015). We follow the setting from Aljundi et al. (2019a), and implement the classifier of each expert by employing a reduced version of ResNet-18, He et al. (2016). The comparison between the proposed approach and the baselines on MINI-ImageNet and Permuted MNIST, which contains images representing different random permutation of the image pixels from the MNIST database are reported in Tab. 4, where the results for the baselines are cited from Jin et al. (2021), and the number of experts for the EEM is of 7. This result shows that EEM Sliding Window (EEM-SW) outperforms the baselines under the continuous learning of a challenging dataset containing complex images.

In the following we test the learning robustness by adopting a more challenging learning setting considering data streams with fuzzy task boundaries, as in Lee et al. (2020), where we contaminate the data from a certain task with outliers corresponding to data from other tasks. We train the proposed approach under this setting and report the results in Tab. 5. These results show that the proposed EEM outperforms other baselines by a large margin on datasets characterized by fuzzy task boundaries, when the data stream is non-stationary and contains images from different data categories, demonstrating its robustness.

5.4. Ablation study

We perform an ablation study to investigate the performance of the proposed EEM under different hyperparameter configurations. For testing the expansion mechanism we consider various λ values in Eq. (6), which controls the mixture's expansion, when training the EEM Sliding Window (EEM-SW) on the Split MNIST. The data distributions (tasks) are indicated at each training step in the results from Fig. 4 (a)-(d) for $\lambda = \{0.3, 0.385, 0.386, 0.39\}$. We can observe that $\lambda = 0.380$ leads to a lower number of components for EEM-SW and when increasing λ, the EEM-SW adds and trains additional experts for the ensemble. The optimal choice of λ should satisfy a trade-off between a compact network architecture and the optimal results for EEM. We can observe from Fig. 4 that $\lambda = 0.385$ represents the best choice since it does not create more experts while maintaining a good performance.

In Fig. 5 we show the results and the EEM-SW model's complexity when varying the threshold λ, while continuously adding and dropping data samples from the memory buffer $\mathcal{G}$ under Split MNIST. These results indicate that a small threshold λ leads to degenerated performance. According to the results from Fig. 5, by increasing λ leads to a linear increase in the number of components for EMM-SW while also improving the performance.

5.5. Visual results

In Fig. 6 (f)-(j) we show the generated images, when aiming to reconstruct the images from Fig. 6 (a)-(e), after training EEM-SW after the continuous learning of Split MNIST. These results show that the proposed approach provides accurate image reconstructions for each task after the continual learning of Split MNIST. We also present the generation results by the selected experts after training with EEM-SW, in Fig. 7 (a)-(e), where each selected expert models the underlying distribution of a unique task. These results show that the proposed EEM-SW can represent well the data, while also capturing the disentangled knowledge characteristics of each task without requiring task labels.

6. Conclusion

In this paper, we address the challenging Task Free Continual Learning (TFCL) paradigm by proposing a novel approach, namely the Evolved Ensemble Model (EMM), which learns infinite data streams without forgetting and without requiring task labels. A memory buffer is used for streaming the training data. To address the data distribution shift in TFCL, we introduce a new statistically driven expansion mechanism based on the Hilbert-Schmidt Independence Criterion (HSIC) which provides appropriate expansion signals for adding new components when the incoming training data statistics changes. Furthermore, the proposed expansion

(a) Testing samples of Task 1. (b) Testing samples of Task 2. (c) Testing samples of Task 3. (d) Testing samples of Task 4. (e) Testing samples of Task 5.

(f) Reconstructions of Task 1. (g) Reconstructions of Task 2. (h) Reconstructions of Task 3. (i) Reconstructions of Task 4. (j) Reconstructions of Task 5.

Figure 6: Results of EEM-SW showing the test images in (a)-(e) and their reconstructions in (f)-(j) after continuous learning on Split MNIST.

(a) Generations of expert 2. (b) Generations of expert 7. (c) Generations of expert 16. (d) Generations of expert 17. (e) Generations of expert 23.

Figure 7: Image generation results by a selected expert from the EEM-SW when considering the continuous learning on Split MNIST.

mechanism does neither require substantial computational costs nor significant additional memory, given that HSIC is evaluated from the statistics of the feature latent space. In order to avoid the memory overload, we propose various dropout mechanisms that can remove unnecessary samples from the memory buffer. We perform several experiments on TFCL benchmarks, which demonstrate that the proposed approach achieves state of the art performance. By addressing the TFCL challenge we develop a novel approach to the streaming artificial intelligence, where tasks characterized by probabilistic distributions with non-stationary characteristics are continuously available for training.

References

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

Aljundi, R., Caccia, L., Belilovsky, E., Caccia, M., Lin, M., Charlin, L., Tuytelaars, T., 2019a. Online continual learning with maximal interfered retrieval, in: Advances in Neural Information Processing Systems (NeurIPS), pp. 11872–11883.

Aljundi, R., Chakravarty, P., Tuytelaars, T., 2017. Expert gate: Lifelong learning with a network of experts, in: Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR), pp. 3366–3375.

Aljundi, R., Kelchtermans, K., Tuytelaars, T., 2019b. Task-free continual learning, in: Proc. od IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR), pp. 11254–11263.

Aljundi, R., Lin, M., Goujaud, B., Bengio, Y., 2019c. Gradient based sample selection for online continual learning, in: Advances in Neural Inf. Proc. Systems (NeurIPS), pp. 11817–11826.

Bang, J., Kim, H., Yoo, Y., Ha, J.W., Choi, J., 2021. Rainbow memory: Continual learning with a memory of diverse samples, in: Proc. of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR), pp. 8218–8227.

Burda, Y., Grosse, R., Salakhutdinov, R., 2015. Importance weighted autoencoders. arXiv preprint arXiv:1509.00519 .

Chen, W., Chen, B., Liu, Y., Cao, X., Zhao, A., Zhang, H., Tian, L., 2022. Max-margin deep diverse latent Dirichlet allocation with continual learning. IEEE Trans. on Cybernetics 52, 5639–5653.

Chen, Z., Ma, N., Liu, B., 2015. Lifelong learning for sentiment classification, in: Proc. of the Annual Meeting of the Assoc. for Comp. Linguistics and Int. Joint Conf. on Natural Language Processing, pp. 750–756.

Cortes, C., Gonzalvo, X., Kuznetsov, V., Mohri, M., Yang, S., 2017. AdaNet: Adaptive structural learning of artificial neural networks, in: Proc. of Int. Conf. on Machine Learning (ICML), vol. PMLR 70, pp. 874–883.

Dai, W., Yang, Q., Xue, G.R., Yu, Y., 2007. Boosting for transfer learning, in: Proc. Int Conf. on Machine Learning (ICML), pp. 193–200.

De Lange, M., Tuytelaars, T., 2021. Continual prototype evolution: Learning online from non-stationary data streams, in: Proc. of the IEEE/CVF Int. Conf. on Computer Vision (ICCV), pp. 8250–8259.

Erhan, D., Szegedy, C., Toshev, A., Anguelov, D., 2014. Scalable object detection using deep neural networks, in: Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR), pp. 2147–2154.

Fagot, J., Cook, R.G., 2006. Evidence for large long-term memory capacities in baboons and pigeons and its implications for learning and the evolution of cognition. Proc. of the National Academy of Sciences (PNAS) 103, 17564–17567.

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y., 2014. Generative adversarial nets, in: Proc. Advances in Neural Inf. Proc. Systems (NIPS), pp. 2672–2680.

Gretton, A., Bousquet, O., Smola, A., Schölkopf, B., 2005. Measuring statistical dependence with Hilbert-Schmidt norms, in: Proc. Int. Conf. on Algorithmic Learning Theory, vol. Lecture Notes in Artif. Intell. (LNAI) 3734, pp. 63–77.

He, K., Zhang, X., Ren, S., Sun, J., 2016. Deep residual learning for image recognition, in: Proc. of IEEE Conf. on Computer Vision and Pattern Recognition (CVPR), pp. 770–778.

Heo, B., Lee, M., Yun, S., Choi, J.Y., 2019. Knowledge distillation with adversarial samples supporting decision boundary, in: Proc. of the AAAI Conf. on Artificial Intelligence, pp. 3771–3778.

Hinton, G., Vinyals, O., Dean, J., 2014. Distilling the knowledge in a neural network, in: Proc. NIPS Deep Learning Workshop, arXiv preprint arXiv:1503.02531.

Jin, X., Sadhu, A., Du, J., Ren, X., 2021. Gradient-based editing of memory examples for online task-free continual learning, in: Advances in Neural Information Processing Systems (NeurIPS), pp. 1–13.

Jung, H., Ju, J., Jung, M., Kim, J., 2018. Less-forgetful learning for domain expansion in deep neural networks, in: Proc. of the AAAI Conf. on Artificial Intelligence, pp. 3358–3365.

Kingma, D.P., Welling, M., 2013. Auto-encoding variational Bayes. arXiv preprint arXiv:1312.6114 .

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

Krizhevsky, A., Hinton, G., 2009. Learning multiple layers of features from tiny images. Technical Report.

Le, J., Lei, X., Mu, N., Zhang, H., Zeng, K., Liao, X., 2021. Federated continuous learning with broad network architecture. IEEE Trans. on Cybernetics 51, 3874–3888.

Le, Y., Yang, X., 2015. Tiny ImageNet visual recognition challenge. CS 231N 7, 3.

LeCun, Y., Bottou, L., Bengio, Y., Haffner, P., 1998. Gradient-based learning applied to document recognition. Proc. of the IEEE 86, 2278–2324.

Lee, S., Ha, J., Zhang, D., Kim, G., 2020. A neural Dirichlet process mixture model for task-free continual learning, in: Proc. Int. Conf. on Learning Representations (ICLR), arXiv preprint arXiv:2001.00689.

Li, Z., Hoiem, D., 2017. Learning without forgetting. IEEE Trans. on Pattern Analysis and Machine Intelligence 40, 2935–2947.

Lopez-Paz, D., Ranzato, M., 2017. Gradient episodic memory for continual learning, in: Advances in Neural Information Processing Systems (NeurIPS), pp. 6467–6476.

Nguyen, C.V., Li, Y., Bui, T.D., Turner, R.E., 2018. Variational continual learning, in: Proc. of Int. Conf. on Learning Representations (ICLR), arXiv preprint arXiv:1710.10628.

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

Phuong, M., Lampert, C., 2019. Towards understanding knowledge distillation, in: Proc. Int. Conf. on Machine Learning (ICML), vol. PMLR 97, pp. 5142–5151.

Polikar, R., Upda, L., Upda, S.S., Honavar, V., 2001. Learn++: An incremental learning algorithm for supervised neural networks. IEEE Trans. on Systems Man and Cybernetics, Part C 31, 497–508.

Ramapuram, J., Gregorova, M., Kalousis, A., 2020. Lifelong generative modeling. Neurocomputing 404, 381–400.

Rannen, A., Aljundi, R., Blaschko, M., Tuytelaars, T., 2017. Encoder based lifelong learning, in: Proc. of the IEEE Int. Conf. on Computer Vision (ICCV), pp. 1320–1328.

Rao, D., Visin, F., Rusu, A., Pascanu, R., Teh, Y.W., Hadsell, R., 2019a. Continual unsupervised representation learning, in: Advances in Neural Information Processing Systems (NeurIPS), pp. 7645–7655.

Rao, D., Visin, F., Rusu, A.A., Teh, Y.W., Pascanu, R., Hadsell, R., 2019b. Continual unsupervised representation learning, in: Proc. Neural Inf. Proc. Systems (NeurIPS), pp. 7645–7655.

Rebuffi, S.A., Kolesnikov, A., Sperl, G., Lampert, C.H., 2017. iCaRL: Incremental classifier and representation learning, in: Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR), pp. 2001–2010.

Redko, I., Habrard, A., Sebban, M., 2017. Theoretical analysis of domain adaptation with optimal transport, in: Porc. Joint European Conf. on Machine Learning and Knowledge Discovery in Databases (ECML PKDD), vol LNCS 10535, pp. 737–753.

Ren, B., Wang, H., Li, J., Gao, H., 2017. Life-long learning based on dynamic combination model. Applied Soft Computing 56, 398–404.

Ren, S., He, K., Girshick, R., Sun, J., 2015. Faster R-CNN: Towards real-time object detection with region proposal networks, in: Proc. Advances in Neural Inf. Proc. Systems (NIPS), pp. 91–99.

Rolnick, D., Ahuja, A., Schwarz, J., Lillicrap, T.P., Wayne, G., 2019. Experience replay for continual learning, in: Advances in Neural Information Processing Systems (NeurIPS), pp. 348–358.

Shin, H., Lee, J.K., Kim, J., Kim, J., 2017. Continual learning with deep generative replay, in: Advances in Neural Information Processing Systems (NIPS), pp. 2990–2999.

Srivastava, A., Valkov, L., Russell, C., Gutmann, M.U., Sutton, C., 2017. Veegan: Reducing mode collapse in gans using implicit variational learning, in: Proc. Advances in Neural Inf. Proc. Systems (NIPS), pp. 3308–3318.

Sun, G., Yang, C., Liu, J., Liu, L., Xu, X., Yu, H., 2019. Lifelong metric learning. IEEE Trans. on Cybernetics 49, 3168–3179.

Tessler, C., Givony, S., Zahavy, T., Mankowitz, D.J., Mannor, S., 2017. A deep hierarchical approach to lifelong learning in Minecraft, in: Proc. AAAI Conf. on Artificial Intelligence, pp. 1553–1561.

Tzeng, E., Hoffman, J., Saenko, K., Darrell, T., 2017. Adversarial discriminative domain adaptation, in: Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR), pp. 7167–7176.

Vitter, J.S., 1985. Random sampling with a reservoir. ACM Transactions on Mathematical Software (TOMS) 11, 37–57.

Wang, F.Y., Zhou, D.W., Ye, H.J., Zhan, D.C., 2022. FOSTER: Feature boosting and compression for class-incremental learning, in: Proc. European Conference on Computer Vision (ECCV), vol. LNCS 13685, pp. 398–414.

Wang, T., Li, W., 2018. Kernel learning and optimization with Hilbert–Schmidt independence criterion. Int. Jour. of Machine Learning and Cyber. 9, 1707–1717.

Wen, Y., Tran, D., Ba, J., 2020. BatchEnsemble: an alternative approach to efficient ensemble and lifelong learning, in: Proc. Int. Conf. on Learning Representations (ICLR), arXiv preprint arXiv:2002.06715.

Ye, F., Bors, A., 2022a. Lifelong teacher-student network learning. IEEE Transactions on Pattern Analysis and Machine Intelligence 44, 6280–6296.

Ye, F., Bors, A.G., 2020a. Learning latent representations across multiple data domains using lifelong VAEGAN, in: Proc. European Conf. on Computer Vision (ECCV), vol. LNCS 12365, pp. 777–795.

Ye, F., Bors, A.G., 2020b. Lifelong learning of interpretable image representations, in: Proc. Int. Conf. on Image Processing Theory, Tools and Applications (IPTA), pp. 1–6.

Ye, F., Bors, A.G., 2020c. Mixtures of variational autoencoders, in: Proc. Int. Conf. on Image Processing Theory, Tools and Applications (IPTA),

pp. 1–6.
Ye, F., Bors, A.G., 2021a. InfoVAEGAN: Learning joint interpretable representations by information maximization and maximum likelihood, in: Proc. IEEE Int. Conf. on Image Processing (ICIP), pp. 749–753.
Ye, F., Bors, A.G., 2021b. Learning joint latent representations based on information maximization. Information Sciences 567, 216–236.
Ye, F., Bors, A.G., 2021c. Lifelong infinite mixture model based on knowledge-driven Dirichlet process, in: Proc. of the IEEE Int. Conf. on Computer Vision (ICCV), pp. 10695–10704.
Ye, F., Bors, A.G., 2021d. Lifelong twin generative adversarial networks, in: Proc. IEEE Int. Conf. on Image Processing (ICIP), pp. 1289–1293.
Ye, F., Bors, A.G., 2022b. Deep mixture generative autoencoders. IEEE Transactions on Neural Networks and Learning Systems 33, 5789–5803.
Ye, F., Bors, A.G., 2022c. Lifelong generative modelling using dynamic expansion graph model, in: Proc. of the AAAI Conf. on Artificial Intelligence, pp. 8857–8865.
Ye, F., Bors, A.G., 2023. Lifelong mixture of variational autoencoders. IEEE Transactions on Neural Networks and Learning Systems 34, 461–474.
Yoon, J., Yang, E., Lee, J., Hwang, S.J., 2017. Lifelong learning with dynamically expandable networks, in: Proc. of the International Conference on Learning Representations (ICLR), arXiv preprint arXiv:1708.01547.
Zhai, M., Chen, L., Tung, F., He, J., Nawhal, M., Mori, G., 2019. Lifelong GAN: Continual learning for conditional image generation, in: Proc. of the IEEE Int. Conf. on Computer Vision (ICCV), pp. 2759–2768.
Zhou, D.W., Wang, Q.W., Ye, H.J., Zhan, D.C., 2023. A model or 603 exemplars: Towards memory-efficient class-incremental learning, in: Proc. of the International Conference on Learning Representations (ICLR), arXiv preprint arXiv:2205.13218.

Fei Ye received the PhD in computer science from the University of York, UK. He is currently a postdoctoral associate at the Mohamed bin Zayed University of Artificial Intelligence, UAE. He received the bachelor degree from Chengdu University of Technology, China, in 2014 and the master degree in computer science and technology from Southwest Jiaotong University, China, in 2018. His research topics includes deep generative image models, lifelong learning and mixture models.

Adrian G. Bors received the PhD degree in Informatics from the Aristotle University of Thessaloniki, Greece, in 1999 and the MSc degree in Electronics Engineering from the Polytechnic University of Bucharest, Romania, in 1992. He is currently an Associate Professor at the University of York, UK, and a Visiting Associate Professor at the Mohamed bin Zayed University of Artificial Intelligence, Abu Dhabi, UAE. He was a Visiting Scholar at the Univ. of California at San Diego (UCSD), and an Invited Professor at the Univ. of Montpellier, France and a Research Scientist at the Tampere University, Finland. Dr. Bors has authored and co-authored more than 170 research papers, including 42 in journals. His research interests include computer vision, computational intelligence, pattern recognition, and image processing. He was a member of the organizing committees for *IEEE* WIFS 2021, *IEEE ICIP* 2018 and 2001, *BMVC* 2016, *IPTA* 2020 and 2014 and *CAIP* 2013. He has been an Associate Editor of the *IEEE Transactions on Image Processing* from 2010 to 2014 and the *IEEE Transactions on Neural Networks* from 2001 to 2009. He was a Co-Guest Editor for a special issue for the *International Journal for Computer Vision* in 2018 as well as for the *Journal of Pattern Recognition* in 2015. He is a Senior IEEE member.