Appendix for Self-Evolved Dynamic Expansion Model for Task-Free Continual Learning

August 18, 2023

Contents

A Theoretical framework

In this section, we propose a novel theoretical framework for analyzing the forgetting behaviour of the model under TFCL. First, we give the problem definition and necessary notations :

A.1 Preliminary

Definition 1 ***(The distribution of the data stream.)*** *For a given data stream* $\mathcal{V} = \bigcup_{j=1}^{n} \mathcal{B}_j^r$, *let* $\mathbb{P}_{\mathbf{x}^r}$ *represent the probabilistic representation of* $\mathcal{D}_r^S$. *Let* $\mathbb{P}_i$ *represent the distribution of all previously learnt data batches* $\{\mathbb{B}_1^r, \cdots, \mathbb{B}_i^r\}$ *drawn from* $\mathcal{V}$ *at* $\mathcal{S}_i$.

Definition 2 ***(The model risk and*** $d_{\mathcal{H}\triangle\mathcal{H}}$ ***distance)*** *Let* $\mathcal{H}$ *be a hypothesis space with* d *Vapnik–Chervonenkis (VC) dimension. For a given distribution* $\mathbb{P}_{\mathbf{x}^r}$, *the risk of a model* $h \in \mathcal{H}$ *is defined as* $\mathcal{E}\big(h, \mathbb{P}_{\mathbf{x}^r}\big) \triangleq \mathbb{E}_{\{\mathbf{x},y\}\sim\mathbb{P}_{\mathbf{x}^r}}\big[\tau\big(y, h(\mathbf{x})\big)\big]$. *For two given distributions* $\mathbb{P}_{\mathbf{x}^r}$ *and* $\mathbb{P}_i$, *the* $d_{\mathcal{H}\triangle\mathcal{H}}$ *distance between them is defined as :*

$$d_{\mathcal{H}\triangle\mathcal{H}}\big(\mathbb{P}_{\mathbf{x}^r}(\mathbf{x}), \mathbb{P}_i(\mathbf{x})\big) \triangleq \sup_{(h,h')\in\mathcal{H}^2} \Big| \mathcal{E}\big(h, h', \mathbb{P}_{\mathbf{x}^r}(\mathbf{x})\big) - \mathcal{E}\big(h, h', \mathbb{P}_i(\mathbf{x})\big) \Big| , \tag{1}$$

where $\{h, h'\} \in \mathcal{H}$ *and* $\mathcal{E}\big(h, h', \mathbb{P}_{\mathbf{x}^r}\big)$ *is defined as :*

$$\mathcal{E}\big(h, h', \mathbb{P}_{\mathbf{x}^r}\big) \triangleq \mathbb{E}_{\{\mathbf{x},y\}\sim\mathbb{P}_{\mathbf{x}^r}}\big[\tau\big(h'(\mathbf{x}), h(\mathbf{x})\big)\big] \tag{2}$$

where $|\cdot|$ *is the absolute value and* $\mathbb{P}_{\mathbf{x}^r}(\mathbf{x})$ *is the marginal of* $\mathbb{P}_{\mathbf{x}^r}$.

A.2 Theoretical guarantees

Learning more components into a dynamic expansion model would improve the performance since it may capture more underlying data distributions. However, learning many overlapping components would not improve the performance too much but lead to unnecessary parameters. In this section, we study how to find a good trade-off between the model's size (the number of components) and generalization performance. One solution to induce a good trade-off is to promote the knowledge diversity among components during the expansion. The primary motivation for this solution is that maintaining the knowledge diversity among components can allow to capture more underlying data distributions with a minimized number of parameters. The proposed

SEDEM can satisfy the above condition by two approaches : (1) The proposed dynamic expansion mechanism compares the knowledge similarity between each previously learnt component and the current component, which guides to expand the network architecture if the current component learns sufficiently novel knowledge. Such a mechanism can promote the information diversity among components. (2) The proposed novelty-aware sample selection approach encourages the current component to learn novel samples, which further promotes the knowledge diversity among components.

In the following, we provide the theoretical analysis to show why the knowledge diversity among components can lead to a good trade-off between the model's size and generalization performance.

Assumption 1 *Let $\mathbf{Q} = \{\mathcal{Q}_1, \cdots, Q_c\}$ be a dynamic expansion model with c components at the training step (T_i). Let $\mathcal{S}_{a_j}$ be the training step that $\mathcal{Q}_j$ was trained on and $\mathcal{C}_{a_j}$ was the associated memory buffer. We assume that (Eq.(7) of the paper) is the optimal component selection criterion. Then we can view the dynamic expansion model $\mathbf{Q}$ as a single model h trained on all previously learnt memories $\{\mathcal{C}_{a_1}, \cdots, \mathcal{C}_{a_{c-1}}\}$ and the current memory $\mathcal{C}_i$ at $\mathcal{S}_i$, where $\mathcal{C}_{a_c} = \mathcal{C}_i$. Let $\mathbb{P}_{\mathcal{C}_{a_1, \cdots, a_{c-1}} \otimes \mathcal{C}_i}$ represent the distribution of all finished memories $\{\mathcal{C}_{a_1}, \cdots, \mathcal{C}_{a_{c-1}}\}$ and the current memory $\mathcal{C}_i$ at $\mathcal{S}_i$.*

Theorem 1. Let $\mathbb{P}_i$ represent the distribution of all previously learnt data batches drawn from $\mathcal{V}$ at $\mathcal{S}_i$. Based on Assumption 1. we derive a GB with probability (at least $1-\delta$) at $\mathcal{S}_i$:

$$\begin{aligned}
\mathcal{E}\big(h, \mathbb{P}_i\big) \leq \mathcal{E}\big(&h, h_{\mathcal{C}_{a_1, \cdots, a_{c-1}} \otimes \mathcal{C}_i}, \mathbb{P}_{\mathcal{C}_{a_1, \cdots, a_{c-1}} \otimes \mathcal{C}_i}\big) \\
&+ \frac{1}{2} d_{\mathcal{H}\triangle\mathcal{H}}(\mathcal{R}_{\mathbb{P}_i}, \mathcal{R}_{\mathcal{C}_{a_1, \cdots, a_{c-1}} \otimes \mathcal{C}_i}) \\
&+ 4\sqrt{\frac{2d\log(2m') + \log(\frac{2}{\delta})}{m'}} \\
&+ \mathcal{L}_{\text{Error}}(\mathbb{P}_i, \mathbb{P}_{\mathcal{C}_{a_1, \cdots, a_{c-1}} \otimes \mathcal{C}_i}) \, ,
\end{aligned} \tag{3}$$

where $\mathcal{L}_{Error}(\mathbb{P}_i, \mathbb{P}_{\mathcal{C}_{a_1, \cdots, a_{c-1}} \otimes \mathcal{C}_i})$ is the optimal error defined as :

$$\mathcal{L}_{\text{Error}}(\mathbb{P}_i, \mathbb{P}_{\mathcal{C}_{a_1, \cdots, a_{c-1}} \otimes \mathcal{C}_i}) = \min\big\{\mathcal{E}(h^\star, \mathbb{P}_i) + \mathcal{E}(h^\star, \mathbb{P}_{\mathcal{C}_{a_1, \cdots, a_{c-1}} \otimes \mathcal{C}_i})\big\} \tag{4}$$

where $h^\star$ is the optimal classifier that minimizes the joint risk :

$$h^\star = \arg\min_{h \in \mathcal{H}} \left\{ \mathcal{E}(h, \mathbb{P}_i) + \mathcal{E}(h, \mathbb{P}_{\mathcal{C}_{a_1,\cdots,a_{c-1}} \otimes \mathcal{C}_i}) \right\} \tag{5}$$

The detailed proof can be found in [2].

Remark. We have several observations from Theorem 4 :

- The $d_{\mathcal{H}\triangle\mathcal{H}}$ distance between $\mathbb{P}_i$ and $\mathbb{P}_{\mathcal{C}_{a_1,\cdots,a_{c-1}} \otimes \mathcal{C}_i}$ plays an important role for the forgetting behaviour of h. As $d_{\mathcal{H}\triangle\mathcal{H}}$ distance increases in Eq. (3), h would suffer from a significant degeneration in performance since RHS of Eq. (3) increases. $\mathbb{P}_{\mathcal{C}_{k_1,\cdots,k_{c-1}} \otimes \mathcal{C}_i}$ represents the information from all learnt memories and the current memory, where each memory is learnt by the associated component. Therefore, encouraging the knowledge diversity among components can allow each $\mathbb{P}_{\mathcal{C}_{a_j}}$ to capture a different underlying data distribution, resulting in learning more underlying data distributions with a suitable number of components. In contrast, if several components learn the overlapping knowledge and ignore other underlying data distributions, $\mathbb{P}_{\mathcal{C}_{k_1,\cdots,k_{c-1}} \otimes \mathcal{C}_i}$ would not capture more underlying data distributions of $\mathbb{P}_i$ and thus lead to forgetting during the training.
- This theorem theoretically proves that the probabilistic diversity between trained components in a dynamic expansion model is crucial for relieving forgetting using a minimized number of parameters.

In the following, we provide theoretical analysis to show that the knowledge diversity among trained components can also improve the generalization performance.

Theorem 2 For a given data stream $\mathcal{V} = \bigcup_{j=1}^{n} \mathcal{B}_j^r$, we assume that $\mathcal{V}$ contains t different underlying data distributions. Let $\mathbb{P}_j^{\mathcal{V}}$ represent a certain underlying data distribution. Based on Assumption 1, we derive a GB with probability (at least $1-\delta$) at $\mathcal{S}_i$:

$$\begin{aligned}
\sum_{j=1}^{t} \left\{ \mathcal{E}\big(h, \mathbb{P}_j^{\mathcal{V}}\big) \right\} \leq\ & \mathcal{E}\big(h, h_{\mathcal{C}_{a_1,\cdots,a_{c-1}} \otimes \mathcal{C}_i}, \mathbb{P}_{\mathcal{C}_{a_1,\cdots,a_{c-1}} \otimes \mathcal{C}_i}\big) \\
& + \frac{1}{2} d_{\mathcal{H}\triangle\mathcal{H}}(\mathcal{R}_{\mathbb{P}_j^{\mathcal{V}}}, \mathcal{R}_{\mathcal{C}_{a_1,\cdots,a_{c-1}} \otimes \mathcal{C}_i}) + 4\sqrt{\frac{2d\log(2m') + \log(\frac{2}{\delta})}{m'}} \\
& + \mathcal{L}_{\mathrm{Error}}\big(\mathbb{P}_j^{\mathcal{V}}, \mathbb{P}_{\mathcal{C}_{a_1,\cdots,a_{c-1}} \otimes \mathcal{C}_i}\big)\,,
\end{aligned} \tag{6}$$

From Eq. (6), it observes that as the proposed model learns more components over time, RHS of Eq. (6) would be reduced since the model gains more knowledge from the data stream. Since the data stream has several different underlying data distributions, encouraging the knowledge diversity among components in the proposed model can help to capture these underlying data distributions with a fair number of parameters. The existing dynamic expansion models [8, 6, 11] fail to achieve the optimal trade-off between the model's size and generalization performance since they do not take into account the diversity of components when performing the expansion.

A.3 The theoretical analysis for the expansion threshold

In this section, we provide the theoretical analysis for the expansion threshold (Eq.(1) of the paper). As the expansion threshold β increases, we tend to employ less experts for learning, which can be explained by the following analysis.

$$\begin{aligned}\sum_{j=1}^{t}\big\{\mathcal{E}\big(h,\mathbb{P}^{T}_{t,j}\big)\big\} \leq \sum_{j=1}^{t}\big\{\mathcal{E}\big(h,h_{\mathcal{C}_{a_1,\cdots,a_{c-1}}\otimes\mathcal{C}_i},\mathbb{P}_{\mathcal{C}_{a_1,\cdots,a_{c-1}}\otimes\mathcal{C}_i}\big) \\ +\frac{1}{2}d_{\mathcal{H}\triangle\mathcal{H}}(\mathcal{R}_{\mathbb{P}^{T}_{t,j}},\mathcal{R}_{\mathcal{C}_{a_1,\cdots,a_{c-1}}\otimes\mathcal{C}_i})+4\sqrt{\frac{2d\log(2m')+\log(\frac{2}{\delta})}{m'}} \\ +\mathcal{L}_{\text{Error}}(\mathbb{P}^{T}_{t,j},\mathbb{P}_{\mathcal{C}_{a_1,\cdots,a_{c-1}}\otimes\mathcal{C}_i})\big\}\,,\end{aligned} \tag{7}$$

We assume that $\mathcal{D}^{T}_{t,j}$ has t number of underlying data distribution and each one is denoted as $\mathbb{P}^{T}_{t,j}$. A small number of experts would allow $\mathbb{P}_{\mathcal{C}_{a_1,\cdots,a_{c-1}}\otimes\mathcal{C}_i}$ to capture fewer knowledge and thus would lose the knowledge corresponding to several target distributions. In contrast, when we decrease the expansion threshold β, $\mathbb{P}_{\mathcal{C}_{a_1,\cdots,a_{c-1}}\otimes\mathcal{C}_i}$ can capture more knowledge and can reduce the $d_{\mathcal{H}\triangle\mathcal{H}}$ distance term, leading to a reduction in RHS of Eq. (7). Although, a small expansion threshold β can improve the generalization performance of the proposed model, it also leads to a large number of experts where some of them would capture the same underlying data distribution. An appropriate threshold β can allow the proposed model to employ fewer experts to learn more underlying data distributions, ensuring a good trade-off between the model's size and generalization performance.

B Additional information for the proposed SEDEM

In this section, we provide the pseudocode of the proposed SEDEM in Algorithm 1, which can summarized into four steps :

Step 1. Sample selection : We continually add the incoming data batches $\mathcal{B}_i^r$ to $\mathcal{C}_i$, as $\mathcal{C}_i = \mathcal{C}_{i-1} \bigcup \mathcal{B}_i^r$ at $\mathcal{S}_i$. If the memory buffer size is larger than λ, we perform the sample selection by using Eq.(4) of the paper and then we perform **Step 2**.

Step 2. Training SEDEM : We build the first expert $\mathcal{Q}_1$ into $\mathbf{Q}$ in the beginning of the training phase, and train it until $\mathcal{S}_\lambda$ in order to preserve the initial information of a data stream. The subsequent learning is described in Fig.2 of the paper, where we suppose that we have already trained k experts and added them into $\mathbf{Q}$ at $\mathcal{S}_i$. We only optimize the current expert $\mathcal{Q}_k$ by using the two loss functions :

$$\mathcal{L}_{cl} = -\frac{1}{\lambda} \sum\nolimits_{j=1}^{\lambda} \Big\{ \sum\nolimits_{t=1}^{C} \left\{ y_j^m(t) \log(p_j^k(t)) \right\} \Big\} \tag{8}$$

$$\mathcal{L}_{Vl} = -\frac{1}{\lambda} \sum\nolimits_{j=1}^{\lambda} \Big\{ \mathcal{L}_{VAE}(\mathbf{z}_j ; G_{(\phi_k, \varphi_k)}) \Big\} \,, \tag{9}$$

where $p_j^k(t)$ is the SoftMax probability for the t-th class, predicted by using $f_{\omega_k} \circ k_{\gamma_k}(\mathbf{x}_j^m)$. $\mathbf{z}_j$ is the j-th feature vector extracted by using the feature extractor f_{ω_k} of $\mathcal{Q}_k$. Eq. (8) and Eq. (9) are employed to train the classifier $f_{\omega_k} \circ C_{\gamma_k}(\mathbf{x}_j^m)$ with the mask parameters and the expert selector $G_{(\phi_k, \varphi_k)}$ on $\mathcal{C}_i$ at $\mathcal{S}_i$. Then we perform **Step 3**.

Step 3 . Dynamic expansion : To avoid the frequently checking the model expansion, we only evaluate Eq.(1) of the paper if and only if the memory buffer is full $|\mathcal{C}_i| = \lambda$ where $|\mathcal{C}_i|$ is the number of memorized samples, If Eq.(1) of the paper is satisfied, we add a new expert $\mathcal{Q}_{k+1}$ to $\mathbf{Q}$ and clear up the memory buffer $\mathcal{C}_i$ in order to allow $\mathcal{Q}_{k+1}$ to learn statistically non-overlapping samples. Then we return back to **Step 1**.

Step 4 . Testing phase : Once all $\{\mathcal{S}_1, \cdots, \mathcal{S}_n\}$ are completed, we perform the expert selection by using Eq.(7) of the paper to select an appropriate expert for evaluating a given input.

B.1 Additional information for the difference between SEDEM and related works

In this section, we discuss the difference between the proposed SEDEM and several related works. The first work related to this paper is proposed in [11], called Online Cooperative Memorization (OCM), which manages two memory buffers to store the short and long-term information from a data stream. OCM can also be combined with the dynamic expansion mechanism to further enhance its generalization performance. There are several differences between OCM and SEDEM. First, OCM employs

Algorithm 1 Training algorithm for SEMOE

(Input:The data stream);
for $i < n$ **do**
  $\mathcal{B}_i^r \sim \mathcal{S}$
  $\mathcal{C}_i = \mathcal{C}_{i-1} \bigcup \mathcal{B}_i^r$
  Sample selection
  if $|\mathcal{C}_i| > \lambda$ **then**
    for $c < |\mathcal{C}_i|$ **do**
      $\mathbf{x}_j^m \sim \mathcal{C}_i$
      $\mathcal{L}_s(\mathbf{x}_j^m) \triangleq -\frac{1}{k-1}\sum_{h=1}^{k-1}\left\{\sum_{t=1}^{C}\left\{y_j^m(t)\log(p_j^h(t))\right\}\right\}$
    end for
    $\mathcal{C}_i = \{\mathbf{x}_j^m \mid \mathcal{L}_s(\mathbf{x}_j^m) < \mathcal{L}_s(\mathbf{x}_{j+1}^m), j = 1, \cdots, \lambda\}$
  end if
  Training the SEMOE
  if $|\mathbf{Q}| = 1$ and $i > \lambda$ **then**
    $\mathbf{Q} = \mathcal{Q}_2 \bigcup \mathbf{Q}$ Add the second expert.
  end if
  $k = |\mathbf{Q}|$
  Train the classifier of $\mathcal{Q}_k$ on $\mathcal{C}_i$ using $\mathcal{L}_{cl}$
  Train the expert selector of $\mathcal{Q}_k$ on $\mathcal{C}_i$ using $\mathcal{L}_{Vl}$
  Dynamic expansion
  if $|\mathcal{C}_i| > \lambda$ **then**
    if $\min\left\{\mathcal{L}_b(\mathcal{Q}_1, \mathcal{Q}_k), \cdots, \mathcal{L}_b(\mathcal{Q}_{k-1}, \mathcal{Q}_k)\right\} \geq \beta$ **then**
      $\mathbf{Q} = \mathcal{Q}_{k+1} \bigcup \mathbf{Q}$ Add the second expert.
    end if
  end if
end for
Testing phase
Perform the expert selection $s^\star = \arg\max_{s=1,\cdots,k}\{\mathcal{L}_{VAE}(f_{\omega_s}(\mathbf{x}); G_{(\phi_s,\varphi_s)})\}$
Perform the evaluation

a dual memory system while SEDEM uses a single memory buffer. Second, OCM proposes a kernel-based sample selection approach that transfers necessary samples from short-term to long-term memory. The sample selection in SEDEM is based on the cross-entropy evaluation, which encourages the newly added component to learn novel knowledge. Finally, OCM detects the loss change as the expansion signal, which does not have theoretical guarantees. In contrast, the proposed SEDEM evaluates knowledge diversity among experts as the expansion signal, ensuring a compact model structure and having theoretical guarantees.

The second related work is called the Online Discrepancy Distance Learning (ODDL) [12] which introduces to estimate the discrepancy distance between the already learnt knowledge and incoming samples and uses this result for the model expansion and sample selection. There are several differences between ODDL and SEDEM. First, the discrepancy-based expansion mechanism in ODDL requires performing the generation (sampling) process for each component, leading to more computational costs. In contrast, the proposed expansion mechanism (Eq.(1) of the paper) directly estimates the knowledge diversity among experts using the memorized samples, as the expansion signal, which is more efficient than ODDL. Second, as similar to the expansion mechanism, the sample selection in ODDL also needs the sampling process for each component. In contrast, the proposed SEDEM employs the cross-entropy evaluation without the generation process for the sample selection, which is more efficient. Finally, ODDL considers learning a VAE model on the image space while the proposed SEDEM trains each VAE model to learn the feature representation from each expert. Consequently, the proposed SEDEM enjoys faster inference at the testing phase than ODDL.

C Additional information for experiment

C.1 Additional information for the setting

Network architecture and hyperparameter : We adapt a small CNN network instead of ResNet-18 [4], used as the classifier for Split CIFAR10 and Split CIFAR100 in order to reduce the whole model size. We also use an MLP network with 2 hidden layers of 200 units [3] as the classifier for Split MNIST. We set the maximum memory size λ as 2000, 1000, and 5000 for Split MNIST, Split CIFAR10, and Split CIFAR100, respectively.

GPU hardware. The GPU used for the experiments was GeForce GTX 1080. The operating system considered for experiments was Ubuntu 18.04.5.

Split MNIST. We divide MNIST which contains 60k training samples into five tasks, each consisting of images from two classes, in consecutive order of their displayed digits, while increasing the numbers represented in the images [3].

Split CIFAR10. We split CIFAR10 into five tasks where each task consists of samples from two different classes [3].

Split CIFAR100. We split CIFAR100 into 20 tasks where each task has 2500 examples from five different classes [7].

We adapt ResNet 18 [4] for Split CIFAR10 and Split CIFAR100. We use an MLP network with 2 hidden layers of 400 units each [3] for Split MNIST.

C.2 Additional information for baselines

In this section, we introduce several baselines in detail.

Finetune is a simple model, implemented by a classifier, which is directly trained on a new batch of images during TFCL.

Gradient Episodic Memory (GEM) [7] is a memory-based approach that would use the memory to store past samples. GEM is also required to access both the task label and class label during the training.

Incremental Classifier and Representation Learning (iCARL) [9] is a standard memory-based method used in a class incremental setup.

reservoir* [10] is a memory-based approach that stores the observed sample into a memory buffer $\mathcal{C}$ with probability $|\mathcal{C}|/n$ where n is the number of stored samples, and $|\cdot|$ represents the cardinality of a set.

Dynamic-OCM [11] is a dynamic expansion model which proposes an online cooperative memorization (OCM) approach. OCM manages two memory buffers, aiming to store short- and long-term knowledge during training. In addition, Dynamic-OCM detects the change of the loss value as expansion signals, which does not have theoretical guarantees.

MIR [7] introduces a retrieval strategy for the sample selection in the memory during the Online Continual Learning (OCL). However, the retrieval strategy in MIR requires evaluating the loss in each training session. This means that MIR requires modifying the retrieval strategy for different tasks such as classification or generation tasks. The proposed OCM does not change the sample selection strategy for different tasks since we evaluate the sample similarity in the given feature space using the kernel function from Eq. (16) from the paper.

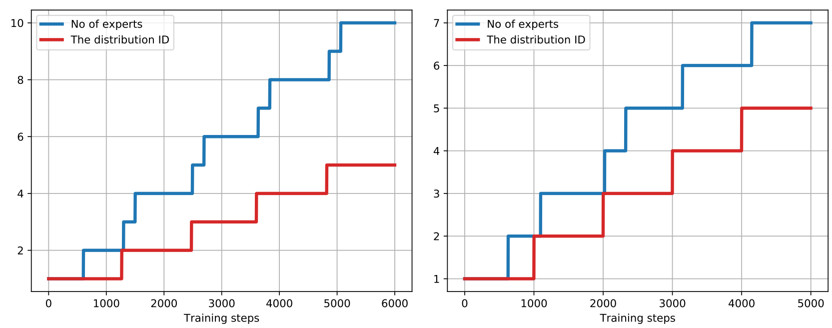
GSS [1] formulates the sample selection process as a constraint reduction problem. GSS stores samples in a buffer based on the gradient information which requires to access the class labels and can not be applied in the unsupervised learning setting.

D Additional results for the ablation study

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

Methods	Split MNIST	Split CIFAR10	Split CIFAR100
SEDEM-CoPE	97.63	50.82	23.75
SEDEM-MIR	97.65	50.38	23.62
SEDEM-reservoir	97.98	50.35	22.97
SEDEM-NoRS	97.29	50.14	22.85
SEDEM-B1	97.42	52.98	22.74
SEDEM	**98.35**	**55.27**	**24.85**

Table 1: The effectiveness of the proposed sample selection in SEDEM.



(a) Split MNIST (b) Split CIFAR10

Figure 1: The number of experts of SEDEM and the distribution shift during the training.

D.1 Dynamic expansion

In this section, we investigate the performance of the proposed model when changing the expansion threshold. First, we train the proposed model on Split MNIST and Split CIFAR100 with different thresholds and the results are reported in Fig. 2. It observes that a small threshold allows SEDEM to use fewer experts, which leads to degenerated performance. In contrast, as increase the threshold, SEDEM creates more experts while improving performance.

D.2 Memory buffer size

In this section, we train various models using different memory configurations. We report the performance of various models in Fig. 3. It observes that the dynamic expansion model outperforms most static models on all memory configurations. Furthermore, the proposed approach outperforms other baselines under different memory sizes

Methods	Split MNIST	Split CIFAR10	Split CIFAR100
finetune*	19.75 $\pm$ 0.05	18.55 $\pm$ 0.34	3.53 $\pm$ 0.04
MIR*	93.20 $\pm$ 0.36	42.80 $\pm$ 2.22	20.00 $\pm$ 0.57
GEM*	93.25 $\pm$ 0.36	24.13 $\pm$ 2.46	11.12 $\pm$ 2.48
iCARL*	83.95 $\pm$ 0.21	37.32 $\pm$ 2.66	10.80 $\pm$ 0.37
ER + GMED†	82.67 $\pm$ 1.90	34.84 $\pm$ 2.20	20.93 $\pm$ 1.60
ER_a + GMED†	82.21 $\pm$ 2.90	47.47 $\pm$ 3.20	19.60 $\pm$ 1.50
reservoir*	92.16 $\pm$ 0.75	42.48 $\pm$ 3.04	19.57 $\pm$ 1.79
GSS*	92.47 $\pm$ 0.92	38.45 $\pm$ 1.41	13.10 $\pm$ 0.94
CoPE-CE*	91.77 $\pm$ 0.87	39.73 $\pm$ 2.26	18.33 $\pm$ 1.52
CoPE*	93.94 $\pm$ 0.20	48.92 $\pm$ 1.32	21.62 $\pm$ 0.69
CURL*	92.59 $\pm$ 0.66	-	-
CNDPM	95.36 $\pm$ 0.18	48.76 $\pm$ 0.28	22.52 $\pm$ 1.26
WGF-SVGD	-	47.90 $\pm$ 2.50	19.90 $\pm$ 2.30
Dynamic-OCM	94.02 $\pm$ 0.23	49.16 $\pm$ 1.52	21.79 $\pm$ 0.68
SEDEM-NoRS	**97.29**	**50.14**	**22.85**

Table 2: Classification accuracy, representing the average of five independent runs, for the continuous learning of three datasets. * and † denote the results cited from [3] and [5], respectively.

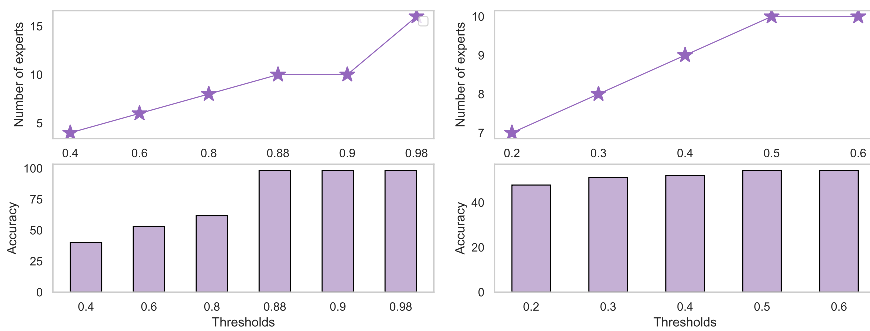
for each dataset. These results show that the proposed model is robust to the memory size change.

D.3 Effects of the proposed sample selection

We investigate the effectiveness of the proposed sample selection by comparing with SEDEM that adopts other sample selection strategies, including CoPE, MIR and reservoir, resulting in several baselines such as SEDEM-CoPE, SEDEM-MIR and SEDEM-reservoir. We also create a baseline, SEDEM-NoRS, which does not employ the sample selection. We report the classification accuracy in Tab. 1. It observes that the proposed sample selection approach can allow SEDEM to perform better than other sample selection approaches. This is because the other sample selection approach does not encourage storing novel samples, which would learn the overlapping knowledge.

Methods	Split MNIST	Split CIFAR10	Split CIFAR100	Split MiniImageNet
Dynamic-OCM	4.2M	68.0M	81.8M	70.0M
CNDPM	4.6M	72.5M	86.6M	78.2M
SEDEM	3.5M	66.8M	79.2M	69.2M

Table 3: The number of parameters of various models under Split MNIST, Split CIFAR10 and Split CIFAR100



(a) Split MNIST (b) Split CIFAR10

Figure 2: The performance of the proposed model when changing the expansion threshold.

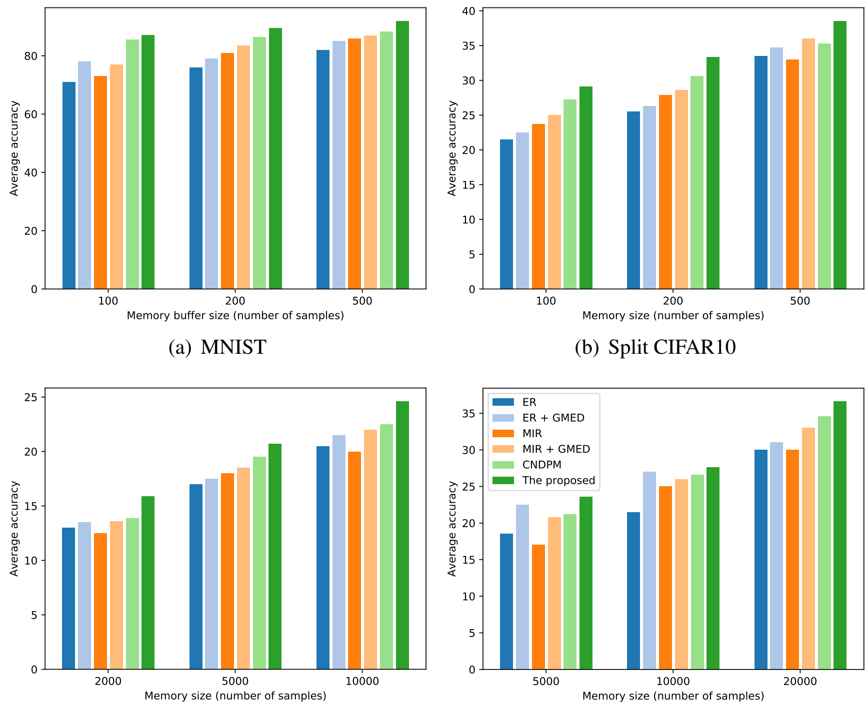
D.4 Effects of the proposed DEKMM

In this section, we evaluate the effectiveness of the proposed DEKMM. We create a baseline that does not use DEKMM, called SEDEM-B1. We report the results in Tab. 1. The results show that the proposed DEKMM can further improve the performance of SEDEM compared with the baseline.

D.5 The knowledge diversity among experts

We show the dynamic expansion of SEDEM trained on Split CIFAR10 in Fig. (1). It observes that SEDEM can accurately detects the data distribution shift. In addition, a single expert almost captures a unique underlying data distribution, demonstrating the knowledge diversity among experts in SEDEM.

In addition, we also record the expansion signals (Left-Hand-Side (LHS) of Eq.(1) of the paper) in each training step where we record the zero when SEDEM has only a single expert. We train the proposed SEDEM under Split CIFAR10 and plot the results in Fig. 4. It observes that the proposed SEDEM gives the low score (LHS of Eq.(1) of the paper) when facing the data distribution shift. Such a low score indicates that



(a) MNIST

(b) Split CIFAR10

(c) Split CIFAR100

(d) Split MinImageNet

Figure 3: The performance of various models under different memory configurations.

the SEDEM performs the expansion to adapt to the data distribution shift, ensuring the knowledge diversity among the trained experts.

D.6 The effects of batch size

In this section, we investigate the performance of the proposed SEDEM when changing the batch size. We train the proposed SEDEM by using the different batch size configurations and the results are reported in Fig. 5. These results show that the proposed SEDEM can maintain a stable performance when changing the batch size.

D.7 Computational costs

In this section, we investigate the computational costs (training times) of various models for the classification task. We report the training times of various models in Tab. 4. It observes that the proposed SEDEM requires fewer training times than Dynamic-

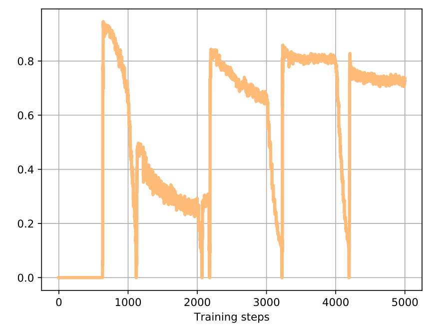


Figure 4: The expansion criterion of the proposed SEDEM under Split CIFAR10.

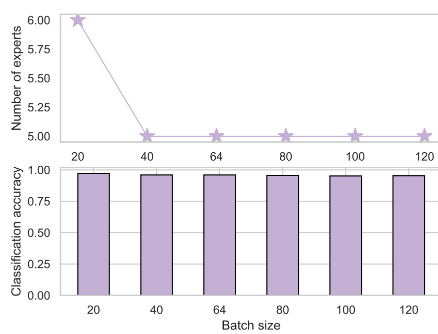


Figure 5: The performance of the proposed SEDEM on Split MNIST when changing the batch size.

OCM, which is also based on the dynamic expansion mechanism. In addition, SEDEM requires more training times than CNDPM since the proposed sample selection in SEDEM requires some computational costs. Furthermore, the SEDEM-NoRS, which does not use the sample selection, requires less training times and perform better than CNDPM, as shown in Tab. 2 and Tab. 4. These results indicate that the proposed SEDEM still outperforms other baselines even if the proposed sample selection is not used.

E The comparison for the model's complexity

We report the number of experts of the proposed model and other existing dynamic expansion models in Tab. 3. It observes that the proposed model achieves better performance and employ fewer parameters compared with CNDPM and Dynamic-OCM.

Methods	Split MNIST	Split CIFAR10	Split CIFAR100
Dynamic-OCM	10.2	42.3	47.8
CNDPM	0.9	18.6	30.2
SEMOE	5.6	32.5	38.9
SEDEM-NoRS	0.8	16.9	26.5

Table 4: The training time of various models for the classification task.

References

[1] R. Aljundi, M. Lin, B. Goujaud, and Y. Bengio. Gradient based sample selection for online continual learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 11817–11826, 2019. 9

[2] Shai Ben-David, John Blitzer, Koby Crammer, Alex Kulesza, Fernando Pereira, and Jennifer Wortman Vaughan. A theory of learning from different domains. *Machine learning*, 79(1):151–175, 2010. 4

[3] Matthias De Lange and Tinne Tuytelaars. Continual prototype evolution: Learning online from non-stationary data streams. In *Proc. of the IEEE/CVF International Conference on Computer Vision (CVPR)*, pages 8250–8259, 2021. 8, 9, 11

[4] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proc. of IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016. 8

[5] Xisen Jin, Arka Sadhu, Junyi Du, and Xiang Ren. Gradient-based editing of memory examples for online task-free continual learning. In *Advances in Neural Information Processing Systems (NeurIPS), arXiv preprint arXiv:2006.15294*, 2021. 11

[6] Soochan Lee, Junsoo Ha, Dongsu Zhang, and Gunhee Kim. A neural Dirichlet process mixture model for task-free continual learning. In *Int. Conf. on Learning Representations (ICLR), arXiv preprint arXiv:2001.00689*, 2020. 5

[7] David Lopez-Paz and Marc'Aurelio Ranzato. Gradient episodic memory for continual learning. In *Advances in Neural Information Processing Systems*, pages 6467–6476, 2017. 8, 9

[8] Dushyant Rao, Francesco Visin, Andrei A. Rusu, Yee Whye Teh, Razvan Pascanu, and Raia Hadsell. Continual unsupervised representation learning. In *Proc. Neural Inf. Proc. Systems (NIPS)*, pages 7645–7655, 2019. 5

[9] Sylvestre-Alvise Rebuffi, Alexander Kolesnikov, Georg Sperl, and Christoph H Lampert. iCaRL: Incremental classifier and representation learning. In *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 2001–2010, 2017. 9

[10] Jeffrey S Vitter. Random sampling with a reservoir. *ACM Transactions on Mathematical Software (TOMS)*, 11(1):37–57, 1985. 9

[11] Fei Ye and Adrian G. Bors. Continual variational autoencoder learning via online cooperative memorization, 2022. 5, 6, 9

[12] Fei Ye and Adrian G Bors. Task-free continual learning via online discrepancy distance learning. *Advances in Neural Information Processing Systems*, 35:23675–23688, 2022. 7