Proceedings Paper:
Ye, Fei and Bors, Adrian Gheorghe orcid.org/0000-0001-7838-0021 (2023) Dynamic Scalable Self-Attention Ensemble for Task-Free Continual Learning. In: IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 04-10 Jun 2023 IEEE
.

https://doi.org/10.1109/ICASSP49357.2023.10094791

DYNAMIC SCALABLE SELF-ATTENTION ENSEMBLE FOR TASK-FREE CONTINUAL LEARNING

Fei Ye and Adrian G. Bors[*]

Department of Computer Science, University of York, York YO10 5GH, UK

ABSTRACT

Continual learning represents a challenging task for modern deep neural networks due to the catastrophic forgetting following the adaptation of network parameters to new tasks. In this paper, we address a more challenging learning paradigm called Task-Free Continual Learning (TFCL), in which the task information is missing during the training. To deal with this problem, we introduce the Dynamic Scalable Self-Attention Ensemble (DSSAE) model, which dynamically adds new Vision Transformer (ViT) based-experts to deal with the data distribution shift during the training. To avoid frequent expansions and ensure an appropriate number of experts for the model, we propose a new dynamic expansion mechanism that evaluates the novelty of incoming samples as expansion signals. Furthermore, the proposed expansion mechanism does not require knowing the task information or the class label, which can be used in a realistic learning environment. Empirical results demonstrate that the proposed DSSAE achieves state-of-the-art performance in a series of TFCL experiments.

Index Terms— Self-Attention, Continual learning, Mixture model, Task-Free Continual Learning, Visual Transformer

1. INTRODUCTION

One of the most fundamental desirable artificial intelligence systems characteristics is to be able to continually acquire novel knowledge and skills without forgetting. The methods having such an ability are called lifelong learning or continual learning models. Modern deep learning can get impressive performance on individual tasks, including classification [1], representation learning and image generation tasks. However, they suffer from huge performance losses when continually learning several different tasks. In this case, the performance loss for a model is called catastrophic forgetting [2].

Lifelong learning methods can be roughly branched into three categories, according to the principle used : regularization [3, 4], memory buffer [5, 6, 7] and dynamic expansion [8, 9]. The regularization-based approaches normally impose constraints on the objective function during training in order to alleviate catastrophic forgetting [3, 4]. The memory-based approaches either train a generator or use a fixed-length memory buffer for preserving and replaying past examples during training. The dynamic expansion approaches dynamically build new components to deal with the new tasks during the training [9]. However, these approaches rely on task information, which is actually not available in a realistic learning paradigm called the Task-Free Continual Learning (TFCL) [10].

Using a memory buffer is a popular approach in TFCL, which usually designs an efficient sample selection strategy [10, 11] by selectively storing and they replying samples during each training step. However, interference occurs in such methods between the old and new sample learning, resulting in adverse knowledge transfer effects [12]. Dynamic expansion approaches [13, 14, 11] increase the model's capacity to deal with the data distribution shift during the training, in order to address the adverse knowledge transfer effect. Recently, the Vision Transformer (ViT) [15] and its variants [16, 17, 18] have shown impressive capabilities when learning individual tasks, which can be extended in continual learning. The key component of ViT is the self-attention mechanism which models the similarity information between different image patches. The effectiveness of the self-attention mechanism in TFCL has not been investigated so far. Therefore, we develop the Dynamic Scalable Self-Attention Ensemble (DSSAE), which employs the self-attention mechanism to learn a non-stationary data distribution without knowing the task information. To implement this goal, each DSSAE expert employs a self-attention-based feature extractor and a linear classifier. Then, a dynamic expansion mechanism adds new experts when identifying data distribution shifts. Specifically, a memory buffer is used to store the recent samples from a data stream and then evaluate the novelty of the memory buffer as the expansion signal. THIS mechanism enables a compact model, where each expert learns a different underlying data distribution.

We summarize the contributions of this paper as follows :

- We propose the Dynamic Scalable Self-Attention Ensemble (DSSAE) which can learn non-stationary data distributions without requiring any task information.
- A new dynamic expansion mechanism evaluating the memory buffer's novelty as an expansion signal to ensure a compact model structure.
- The proposed DSSAE achieves state-of-the-art performance.

[*] Dr. A. G. Bors acknowledges the partial support from the EPSRC, UK, project COUSIN (EP/V009591/1)

2. DYNAMIC VISION TRANSFORMER ENSEMBLE

In this section, we detail the dynamic scalable vision transformer ensemble. First, we describe each module of DSSAE and the memory updating strategy. Then we propose a novel dynamic expansion mechanism, enabling DSSAE to increase its capacity to deal with the data distribution shift under continual learning. In the final section, we propose a learning algorithm for training DSSAE.

2.1. Experts of DSSAE

First, we introduce the learning paradigm for TFCL and then the detailed implementation of each expert of DSSAE. Let $\mathcal{T} = \{\mathcal{T}_1, \cdots, \mathcal{T}_n\}$ be a set of training steps/times for learning a data stream $\mathcal{D}$. This data stream consists of n data batches $\mathcal{D} = \{\mathcal{D}_1, \cdots, \mathcal{D}_n\}$, where each data batch $\mathcal{D}_i = \{\mathbf{x}_{i,j}, y_{i,j}\}_{j=1}^{b}$ consists of several training examples and b is the batch size. In a certain training step $\mathcal{T}_i$, the model can only access the data batch $\mathcal{D}_i$ while all previous data batches $\{\mathcal{D}_1, \cdots, \mathcal{D}_{i-1}\}$ are not available. After the model has finished all training steps, we evaluate the model's performance on all testing samples. In the following, we describe the implementation of an expert for DSSAE.

Each expert E_i in DSSAE consists of a feature extractor F_{δ_i} and a linear classifier F_{θ_i}. We implement each feature extractor F_{δ_i} by using a ViT, given that it has a robust feature learning ability. Let $\mathbf{x} \in \mathbb{R}^{H\times W\times S}$ be a data sample. where $\{H, W, S\}$ represent the image height, weight and channels, respectively. We split an input $\mathbf{x}$ into a set of image patches $\mathbf{b} \in \mathbb{R}^{R\times(K^2\times S)}$ where each patch $\mathbf{b}_j \in \mathbb{R}^{K^2}$ has the size of K^2 pixels and $R = HW/K^2$ is the number of image patches. Let $\mathbf{W}_p$ be a projection matrix which transfers the image patches into the H-dimensional embedding space :

$$\mathbf{p} = \mathbf{W}_p\mathbf{b}\,. \tag{1}$$

Then we consider combining several self-attention modules into a unified framework called the multi-head attention mechanism [19]. Each self-attention module has three independent trainable matrices $\{\mathbf{W}_K^i, \mathbf{W}_Q^i, \mathbf{W}_V^i\}$, aiming to capture different statistical information from an input. A multi-head attention mechanism, therefore, can have a self-attention modules $\{\mathbf{W}_K^1, \mathbf{W}_Q^1, \mathbf{W}_V^1, \cdots, \mathbf{W}_K^a, \mathbf{W}_Q^a, \mathbf{W}_V^a\}$. First, we calculate the output by using each self-attention module :

$$\begin{aligned}\mathcal{H}_i &= \mathrm{Softmax}(\mathbf{Q}^i(\mathbf{K}^i)^T/\sqrt{r})\mathbf{V}^i\,,\\ \mathbf{Q}^i &= \mathbf{W}_Q^i\mathbf{p}, \mathbf{K}^i = \mathbf{W}_K^i\mathbf{p}, \mathbf{V}^i = \mathbf{W}_V^i\mathbf{p}\,,\end{aligned} \tag{2}$$

where $\sqrt{r}$ is a scaling factor. Then we concentrate all outputs from self-attention modules into one matrix, resulting in :

$$\mathcal{H}^\star = \mathrm{Concat}(\mathcal{H}_1, \cdots, \mathcal{H}_a)\,, \tag{3}$$

where $\mathrm{Concat}(\cdot)$ denotes that we concatenate all matrices into a single one. Then the output of the multi-head attention mechanism, Eq. (3), is fed into a feed-forward Multi-Layer Perceptron (MLP) to produce the feature information for a linear classifier, which is expressed by :

$$\begin{aligned}y &= \varepsilon(\mathbf{W}_M\mathbf{x}_m + b_m)\,,\\ \mathbf{x}_m &= \mathrm{F}_{\mathrm{MLP}}(\mathcal{H}^\star)\,,\end{aligned} \tag{4}$$

where $\mathrm{F}_{\mathrm{MLP}}$ is the MLP and $\varepsilon(\cdot)$ represents the sigmoid activation function. $\{\mathbf{W}^m, b_m\}$ are the trainable parameters of the linear classifier and y is the prediction for an input $\mathbf{x}$. Let $\mathbf{F}_{\theta_i}$ and $\mathbf{F}_{\delta_i}$ represent the feature extractor and classifier for the i-th expert, where θ_i and δ_i represent all parameters of the self-attention module and the classifier, respectively. Since the proposed DSSAE, $\mathbf{E} = \{E_1, \cdots, E_c\}$ would involve multiple experts, we introduce a novel dynamic expansion mechanism which enables $\mathbf{E}$ to continually add new experts during the training, which is described in the following section.

2.2. The dynamic expansion mechanism

In this section, we propose a new dynamic expansion mechanism to enable DSSAE for continual learning. The main motivation is that we want to create a new expert when the data distribution shift occurs during the training with continuous streams of data. In order to implement this, we first consider assigning an autoencoder $\mathcal{V}_i$ for each expert E_i, in order to evaluate the knowledge correlation between the information learnt by the expert through the evaluation of data reconstruction and generation, and the information corresponding to a given data batch. In addition, the autoencoder V_i can be also used with the appropriate expert selector when evaluating a given data batch input during the testing phase.

Let $\mathcal{L}_i$ be a fixed-length memory buffer updated at $\mathcal{T}_i$ and $|\mathcal{L}|^{Max}$ be the maximum data sample capacity for $\mathcal{T}_i$. We consider a simple memory updating mechanism that removes the earliest samples while continuously adding new samples from an incoming data stream. Then we introduce a new dynamic mixture model expansion mechanism evaluating the novelty of the data from the memory buffer as the expansion signal at the training step $\mathcal{T}_i$:

$$\mathrm{F}_s(\mathcal{M}_i, E_j) = \frac{1}{|\mathcal{M}_i|}\sum\nolimits_{u=1}^{|\mathcal{M}_i|}\{\mathrm{F}_{Rec}(\mathbf{x}'_u, V_j(\mathbf{x}'_u))\}\,, \tag{5}$$

where $\mathrm{F}_{Rec}(\cdot,\cdot)$ is the reconstruction error and $V_z(\mathbf{x}'_j)$ is the reconstruction for the j-th memorized sample from the memory buffer, implemented by the autoencoder of the j-th expert. Then, the dynamic expansion mechanism evaluates the memory buffer using all previously trained experts :

$$\min\{\mathrm{F}_s(\mathbf{x}'_j, E_1), \cdots, \mathrm{F}_s(\mathbf{x}'_j, E_{k-1})\} \geq \lambda\,, \tag{6}$$

where we assume that DSSAE has already trained k components and λ is a threshold that controls the model expansion. In addition, the threshold λ can also balance the model's generalization performance and complexity. For instance, a big λ

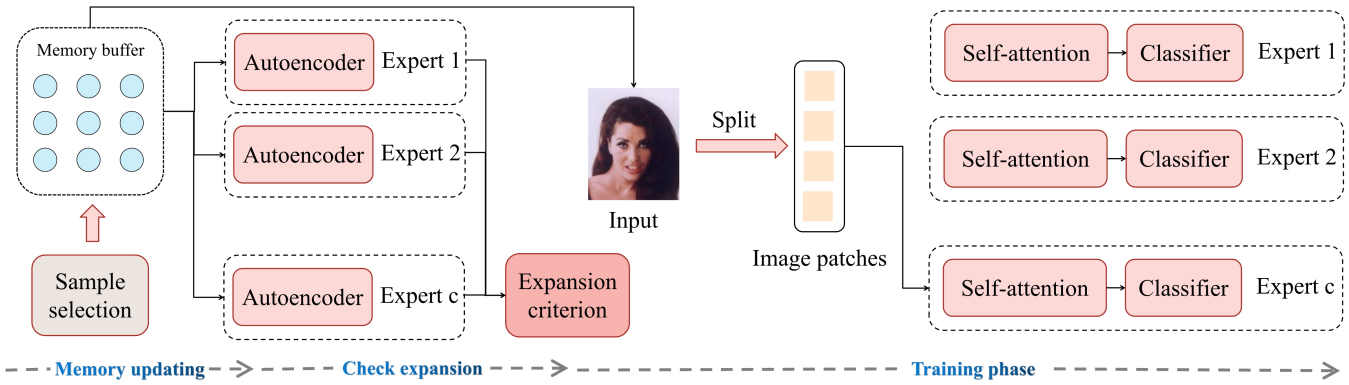

Fig. 1. The learning procedure of the proposed model consists of three steps. The first step is to update the memory buffer by continually adding a new batch of samples. If the memory buffer is overloaded, we then remove the earliest samples from the memory buffer. The second step is to check the model expansion using Eq. (6). Finally, the third step trains the current component on the memory buffer using Eq. (7) and Eq. (8).

leads to more components and thus improves the performance. On the other hand, a small threshold λ tends to build fewer components, resulting in lower performance.

2.3. Implementation

Each E_i expert in DSSAE consists of three modules, a ViT-based feature extractor F_{δ_j}, a linear classifier F_{θ_j} and an autoencoder V_j. For V_j we consider an encoder $Q_j(\mathbf{x})$ and a decoder $S_j(\mathbf{z})$, which are trained by using the reconstruction error loss on the memory buffer at $\mathcal{T}_i$:

$$\mathcal{L}_{Rec} = \frac{1}{|\mathcal{M}_i|} \sum_{u=1}^{|\mathcal{M}_i|} \{\mathrm{F}_{Rec}(\mathbf{x}'_u, V_j(\mathbf{x}'_u))\} \ . \quad (7)$$

Then the classifier of the j-th expert is trained on the memory buffer using the cross-entropy loss :

$$\mathcal{L}_{Class} = \frac{1}{|\mathcal{M}_i|} \sum_{u=1}^{|\mathcal{M}_i|} \{\mathrm{F}_{ce}(\mathrm{F}_{\delta_i} \otimes \mathrm{F}_{\theta_i}(\mathbf{x}'_u), y'_u)\} \ , \quad (8)$$

where $\mathrm{F}_{ce}(\cdot,\cdot)$ is the cross-entropy loss function and $\otimes$ represents the connection between two modules.

Algorithm. We introduce a new algorithm for training the Dynamic Scalable Self-Attention Ensemble (DSSAE), which can be summarized into four steps :

- **Step 1 (Memory updating mechanism).** In the i-th training step $\mathcal{T}_i$, the model sees a new batch of samples from the data stream D which is added to the memory buffer $\mathcal{M}_i$. If the memory buffer $\mathcal{M}_i$ is overloaded $|\mathcal{M}_i| > |\mathcal{M}_i|^{Max}$, we remove the earliest samples from the memory buffer until its size becomes equal to $|\mathcal{M}_i|^{Max}$.
- **Step 2 (Learning process).** In the i-th training step, we only train the current component E_k on the memory buffer $\mathcal{M}_i$ using Eq. (7) and (8), while all previously learnt experts are frozen in order to preserve the prior learnt knowledge.
- **Step 3 (Check the model's expansion).** If the memory buffer is full $|\mathcal{M}_i| = |\mathcal{M}|^{Max}$, we check the model's expansion using Eq. (6). If Eq. (6) is satisfied, we build a new expert E_{k+1} into $\mathbf{E}$ and return back to the step 1.
- **Step 4 (Expert selection).** Once all training steps have been completed, we select a component for a given testing sample $\mathbf{x}_t$ according to :

$$s = \arg \min_{s=1,\cdots,k} \{\mathrm{F}_{reco}(\mathbf{x}_t, V_s(\mathbf{x}_t))\} \,, \quad (9)$$

where s is the selected expert index for evaluating $\mathbf{x}_t$.

3. EXPERIMENTAL RESULTS

3.1. Experiment setting

We adopt the standard TFCL benchmarks from [20]. Split MNIST : we divide MNIST dataset into five tasks, each consisting of samples from two classes. We repeat this on CIFAR10 [21], resulting in Split CIFAR10. Split CIFAR100 : we divide CIFAR100 into 20 tasks where each task consists of 2500 samples belonging to five classes.

Network architecture and hyperparameters for the classifier. Following the setting in [20], the maximum memory size is considerd as 2000, 1000 and 5000 for Split MNIST, Split CIFAR10, and Split CIFAR100, respectively. In each training step, a model would only access a batch of 10 samples. We consider the image patch size of 7×7 pixels for Split MNIST. For Split CIFAR10 and Split CIFAR100, we consider the image patch size of 8×8 pixels.

3.2. Classification task

In Table 1 we evaluate DSSAE on Split MNIST, Split CIFAR10, and Split CIFAR100, and compare the results with several baselines including : Finetune which directly trains a classifier on the data stream, CURL [14], iCARL [22], CoPE

Methods	Split MNIST	Split CIFAR10	Split CIFAR100
finetune*	19.75 ± 0.05	18.55 ± 0.34	3.53 ± 0.04
GEM*	93.25 ± 0.36	24.13 ± 2.46	11.12 ± 2.48
iCARL*	83.95 ± 0.21	37.32 ± 2.66	10.80 ± 0.37
reservoir*	92.16 ± 0.75	42.48 ± 3.04	19.57 ± 1.79
MIR*	93.20 ± 0.36	42.80 ± 2.22	20.00 ± 0.57
GSS*	92.47 ± 0.92	38.45 ± 1.41	13.10 ± 0.94
CoPE-CE*	91.77 ± 0.87	39.73 ± 2.26	18.33 ± 1.52
CoPE*	93.94 ± 0.20	48.92 ± 1.32	21.62 ± 0.69
ER + GMED†	82.67 ± 1.90	34.84 ± 2.20	20.93 ± 1.60
ER_a + GMED†	82.21 ± 2.90	47.47 ± 3.20	19.60 ± 1.50
CURL*	92.59 ± 0.66	-	-
CNDPM*	93.23 ± 0.09	45.21 ± 0.18	20.10 ± 0.12
Dynamic-OCM	94.02 ± 0.23	49.16 ± 1.52	21.79 ± 0.68
DSSAE	**95.23** ± 0.18	**51.36** ± 1.12	**23.35** ± 1.08

Table 1. Classification accuracy of five independent runs for various models on three datasets. * and † denote the results cited from [20] and [23], respectively.

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
DSSAE	**89.72** ± 1.5	**43.27** ± 1.2	**29.87** ± 0.9

Table 2. The classification accuracy of five independent runs for various models over streams with fuzzy task boundaries.

[20], CNDPM, ER + GMED and ER_a + GMED [23], where GMED is Gradient based Memory Editing and ER is the Experience rRplay [24], while ER_a is ER with data augmentation. Those results show that the proposed DSSAE approach outperforms other baselines on all datasets.

In the following, we also evaluate the performance of various models on a more challenging setting called fuzzy task boundaries [13] in which we swap randomly samples between two tasks for each data stream, thus introducing outliers in their probabilistic representations. We report the results in Table 2, where we also compare the performance of various models on Split MINI-ImageNet [25] which divides the MINI-ImageNet [25] into 20 tasks and each task contains samples of five different classes with images of higher complexity. These results show that the proposed approach still outperforms other baselines when learning this more challenging dataset.

3.3. Ablation study

In this section, we perform an ablation study to investigate the effectiveness of each module of the proposed DSSAE. We first evaluate DSSAE when changing the expansion threshold λ from Eq. (6) on Split MNIST and the results are provided in

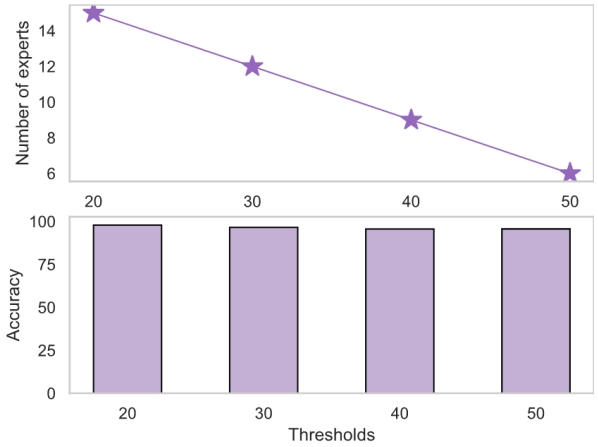

Fig. 2. The performance and the number of experts when changing the threshold λ.

Methods	Split MNIST	Split CIFAR10	Split CIFAR100
DSCNNE	94.12 ± 0.24	50.27 ± 1.02	22.76 ± 1.05
DSSAE	**95.23** ± 0.18	**51.36** ± 1.12	**23.35** ± 1.08

Table 3. Classification accuracy of five independent runs when considering a CNN-based expert against a ViT-based expert.

Fig. 2. We can observe that a small λ encourages the proposed DSSAE to build more experts while a large threshold λ hinders the expansion process. In addition, more experts can lead to better performance while requiring more parameters. We also investigate whether the ViT-based expert is better than a Convolution Neural Network (CNN) based expert. We consider a baseline model where each expert employs a CNN as a feature extractor instead of ViT, called DSCNNE. We train DSCNNE on three datasets and the results are reported in Table 3. These results show that the proposed DSSAE outperforms DSCNNE on three datasets, demonstrating that the ViT-based experts used in DSSAE perform better than the CNN-based expert.

4. CONCLUSION

In this paper, we propose a new model called the Dynamic Scalable Vision Transformer Ensemble (DSSAE), for continual learning. DSSAE dynamically adds new experts, each containing a Visual Transformer (ViT), to deal with the data distribution shift under the continual learning scenario. In order to avoid frequent expansion and ensure the knowledge diversity among the trained components, we propose a new dynamic expansion mechanism evaluating the novelty of incoming samples with respect to the knowledge already aquired by the model. Furthermore, such a mechanism does not require accessing the task information or class label, and can be used in a realistic continual learning setting. We perform a series of experiments, and the empirical results demonstrate that the proposed approach achieves the state of the art performance.

5. REFERENCES

[1] Fei Ye and Adrian G Bors, "Mixtures of variational autoencoders," in *Proc. Int. Conf. on Image Processing Theory, Tools and Applications (IPTA)*, 2020, pp. 1–6.

[2] G. I. Parisi, R. Kemker, J. L. Part, C. Kanan, and S. Wermter, "Continual lifelong learning with neural networks: A review," *Neural Networks*, vol. 113, pp. 54–71, 2019.

[3] F. Zenke, B. Poole, and S. Ganguli, "Continual learning through synaptic intelligence," in *Proc. of Int. Conf. on Machine Learning, vol. PLMR 70*, 2017, pp. 3987–3995.

[4] J. Kirkpatrick, R. Pascanu, N. Rabinowitz, J. Veness, G. Desjardins, A. A. Rusu, K. Milan, J. Quan, T. Ramalho, A. Grabska-Barwinska, D. Hassabis, C. Clopath, D. Kumaran, and R. Hadsell, "Overcoming catastrophic forgetting in neural networks," *Proc. of the National Academy of Sciences (PNAS)*, vol. 114, no. 13, pp. 3521–3526, 2017.

[5] David Lopez-Paz and Marc'Aurelio Ranzato, "Gradient episodic memory for continual learning," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2017, pp. 6467–6476.

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

[7] Fei Ye and Adrian G. Bors, "Learning latent representations across multiple data domains using lifelong VAEGAN," in *Proc. European Conf. on Computer Vision (ECCV), vol. LNCS 12365*, 2020, pp. 777–795.

[8] Fei Ye and Adrian G. Bors, "Lifelong mixture of variational autoencoders," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 34, no. 1, pp. 461–474, 2023.

[9] Fei Ye and Adrian G. Bors, "Lifelong infinite mixture model based on knowledge-driven Dirichlet process," in *Proc. of the IEEE Int. Conf. on Computer Vision (ICCV)*, 2021, pp. 10695–10704.

[10] Rahaf Aljundi, Klaas Kelchtermans, and Tinne Tuytelaars, "Task-free continual learning," in *Proc. of IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 11254–11263.

[11] Fei Ye and Adrian G. Bors, "Continual variational autoencoder learning via online cooperative memorization," in *Proc. European Conf. on Computer Vision (ECCV), vol. LNCS 13683*, 2022, pp. 531–549.

[12] Sebastian Lee, Sebastian Goldt, and Andrew Saxe, "Continual learning in the teacher-student setup: Impact of task similarity," in *Proc. International Conference on Machine Learning, vol. PMLR 139*, 2021, pp. 6109–6119.

[13] 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.

[14] 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 (NeurIPS)*, 2019, pp. 7645–7655.

[15] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby, "An image is worth 16x16 words: Transformers for image recognition at scale," in *Proc. Int. Conf. on Learning Representations (ICLR), arXiv preprint arXiv:2010.11929*, 2021.

[16] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo, "Swin transformer: Hierarchical vision transformer using shifted windows," in *Proc. of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021, pp. 10012–10022.

[17] Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Hervé Jégou, "Training data-efficient image transformers & distillation through attention," in *Proc. International Conferance on Machine Learning (ICML)*. PMLR 139, 2021, pp. 10347–10357.

[18] Stéphane d'Ascoli, Hugo Touvron, Matthew L Leavitt, Ari S Morcos, Giulio Biroli, and Levent Sagun, "ConViT: Improving vision transformers with soft convolutional inductive biases," in *Proc. International Conference on Machine Learning (ICML), vol PMLR 139*, 2021, pp. 2286–2296.

[19] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin, "Attention is all you need," *Advances in Neural Information Processing Systems (NIPS)*, vol. 30, pp. 6000–6010, 2017.

[20] 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 (ICCV)*, 2021, pp. 8250–8259.

[21] Alex Krizhevsky and Geoffrey Hinton, "Learning multiple layers of features from tiny images," Tech. Rep., Univ. of Toronto, 2009.

[22] 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)*, 2017, pp. 2001–2010.

[23] 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.

[24] David Rolnick, Arun Ahuja, Jonathan Schwarz, Timothy P. Lillicrap, and Gregory Wayne, "Experience replay for continual learning," in *Advances in Neural Information Processing Systems 34 (NeurIPS)*, 2019, pp. 348–358.

[25] Ya Le and Xuan Yang, "Tiny imageNet visual recognition challenge," Tech. Rep., Univ. of Stanford, 2015.