

Appendix for Online Task-Free Continual Learning via Dynamic Expansionable Memory Distribution

Contents

A	Additional Information for the Related Work	2
B	The Additional Information for the Experiment Settings	3
B.1	The hyperparameter configuration and GPU hardware.	3
B.2	The description of the datasets	4
C	Additional Ablation Studies	5
C.1	The memory expansion process	5
C.2	The dynamic signals	5
C.3	Different memory sizes	6
C.4	The visual results	6
C.5	The hyperparameter selection	7
C.6	The computational costs	7

A Additional Information for the Related Work

In this section, we provide additional information about the related work, which was presented in Section 2 from the paper. Most existing studies are primarily focused on addressing network forgetting in a popular and general continual learning scenario [1, 2, 4, 5, 6, 13, 14, 15, 16], where the class information and task boundaries are always given during the training and testing phases. However, these studies can not deal with a more challenging and realistic continual learning scenario in which both task and class information are not available. This paper addresses this challenging learning scenario by developing a novel memory approach, called the Dynamic Expansionable Memory Distribution (DEMD) approach, which can automatically preserve critical past examples over time without knowing class or task information. Unlike existing memory-based methods [1, 2, 4, 5, 6, 13, 14, 15, 16], which can easily ensure the category balance into the memory buffer by utilizing task or class labels, the proposed approach compares the sample similarity between the memorized and incoming samples as the signal for implementing the memory expansion process, which can also ensure the preservation of the diversity of data samples over all categories.

A related approach, to the one proposed in the paper, is called the Dynamic Cluster Memory (DCM), proposed in [19], which can also deal with task-free unsupervised continual learning. The proposed approach differs from the DCM in three aspects: (1) The DCM evaluates the data similarity on the high-dimensional data space, such as the image information given by pixels. In contrast, the proposed approach evaluates the data similarity on the low-dimensional latent space, which is faster and more memory efficient; (2) The DCM chooses a sample as the centre point for each memory cluster. In contrast, the proposed approach forms an explicit memory distribution (Gaussian distribution) in a low-dimensional feature space as a compact representation for every sub-memory buffer; (3) This paper proposes a novel theoretical framework and provides theoretical guarantees for the proposed approach while the DCM from [19] does not have any theoretical results.

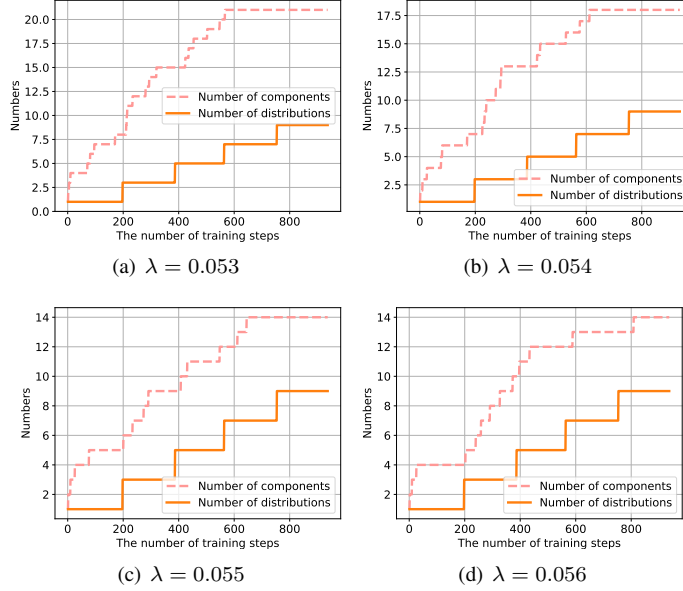


Figure 1: The number of memory distributions and data distributions at each training time.

B The Additional Information for the Experiment Settings

B.1 The hyperparameter configuration and GPU hardware.

In all experiments performed and whose results are reported in this paper, we use Adam [7] as the training algorithm for various learning models. For training the Adam optimization algorithm we consider a learning rate of 0.0001. The number of training epochs for each training time during continual learning is five. Our experiments are performed using one Tesla V100 GPU and running on the Ubuntu 18.04.5 operating system.

In the experiments, we set the hyperparameter $\lambda = 0.057$, for the memory updating in Eq. (5) and (6) from page 4 of the paper, for all datasets.

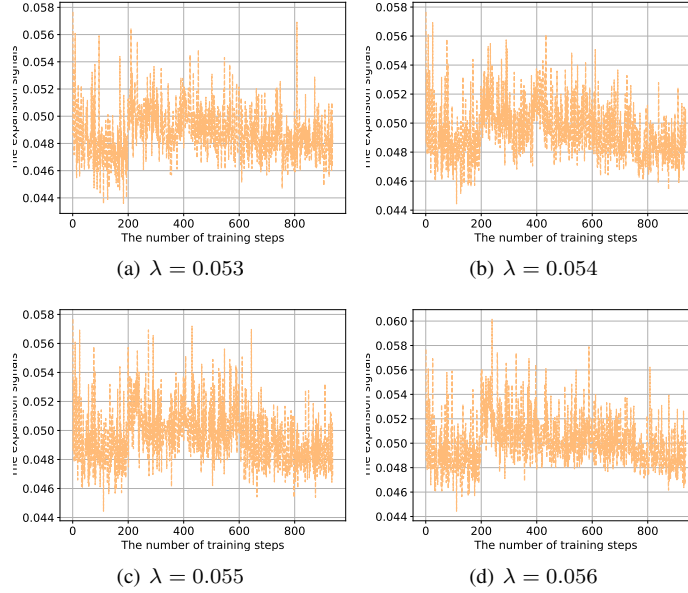


Figure 2: The expansion signals produced by the proposed approach using different λ configurations.

B.2 The description of the datasets

In this section, we provide the detailed information about the datasets used in the experiments from the paper.

Split MNIST. The MNIST dataset [10] comprises images of handwritten digits, featuring 60,000 samples for training and 10,000 samples for testing. We construct Split MNIST by partitioning the MNIST dataset into five segments as outlined in [3], with each segment containing samples from two distinct classes.

Split Fashion. The Fashion dataset [18] consists of images depicting clothing items, including a training set of 60,000 samples and a testing set of 10,000 examples. We generate Split Fashion by dividing the Fashion dataset into five segments in accordance with [3], where each segment encompasses samples from two different classes.

Split CIFAR10. The CIFAR10 dataset [8] is composed of natural images, containing 60,000 training samples and 10,000 testing samples. We create Split CIFAR10 by segmenting CIFAR10 into five parts based on the methodology described in [3], with

each part including samples from two distinct classes.

Split SVHN. The SVHN dataset [12] is a real-world image dataset featuring digits cropped from pictures of house number plates, consisting of 73,257 training images and 26,032 testing samples. We develop Split SVHN by dividing SVHN into five segments as per [3], where each segment contains samples from two different classes.

We resize all images from Split MNIST, Split Fashion, Split SVHN, and Split CIFAR10 to dimensions of $32 \times 32 \times 3$. Additionally, we resize the images from CelebA [11], MINImageNet [17], and ImageNet [9] to $64 \times 64 \times 3$.

C Additional Ablation Studies

In this section, we provide additional ablation studies in the following sections.

C.1 The memory expansion process

In order to investigate the memory expansion process of the proposed approach, we record the number of memory distributions and data distributions (task IDs) at each training time. The results of the proposed approach are plotted in Fig. 1 for 4 different λ values in Eq. (5) and (6) from page 4 of the paper. From these results we can observe that using a small threshold λ encourages the proposed approach to frequently create more memory distributions. On the other hand, using an appropriate λ configuration enables the creation of an appropriate number of memory distributions.

C.2 The dynamic signals

To investigate the expansion process of the memory buffer in the proposed approach during the training, we propose to estimate and record the expansion signals using the left-hand side of Eq. (6) from the page 4 of the paper at each training time. We provide the results in Fig. 2a-d for $\lambda \in \{0.053, 0.054, 0.055, 0.056\}$. These results show that different threshold λ configurations can lead to small differences in the expansion processes. In addition, using a large threshold λ configuration promotes the proposed

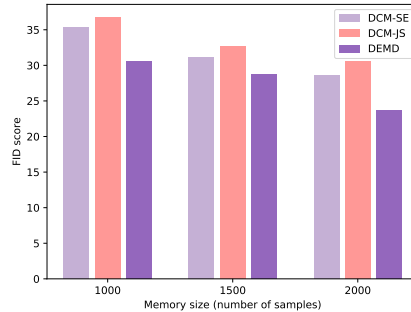


Figure 3: The FID score of various models in continual learning.

approach to yield more peak expansion signals, resulting in creating more memory distributions. In contrast, a small threshold λ configuration leads to a few peak expansion signals.

C.3 Different memory sizes

In this Appendix, we evaluate the performance of various models when considering different memory configurations. In the plots from Fig. 3 we provide the results for training various models when considering the memory buffers of $\rho = \{1000, 1500, 2000\}$ samples, where $\rho = |\mathcal{M}(j)|^{Max}$, represents its maximum buffer capacity. From the results, we find that using a large memory capacity can increase the FID score. In addition, the proposed approach outperforms the current state-of-the-art on all memory configurations.

C.4 The visual results

In this Appendix, we provide additional visual results, which are shown in Fig. 4a,b,c and d for Split MNIST, Split Fashion, Split SVHN and Split CIFAR10. The visual results show that DEMD can produce diverse high-quality generation results following the continual learning of these datasets.

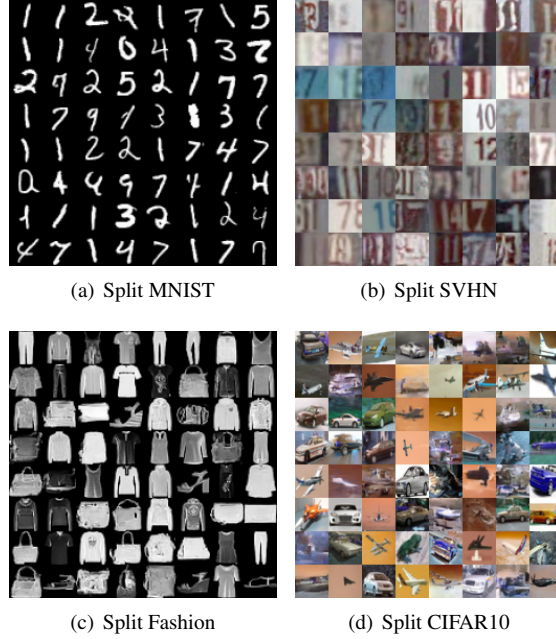


Figure 4: The visual results generated by the proposed approach after learning the Split MNIST, Split SVHN, Split Fashion and Split CIFAR10, respectively.

C.5 The hyperparameter selection

In this section, we provide the detailed information for the hyperparameter selection for DEDM. Specifically, for a given data stream, we record the number of memory distributions over times of training during the continual learning. In addition, we also consider 300 training samples as the validation dataset. We initially search the threshold λ , from Eq. (6) on page 4 of the paper, from 0.01 to 0.1 and then narrow the search space from 0.04 to 0.06. The best λ is determined when the model achieves the best performance on the validation dataset.

C.6 The computational costs

Since the proposed memory approach can be optimized independently, we can only perform the memory optimization process without involving the model’s training procedure. Specifically, we perform the proposed memory approach on the Split MNIST,

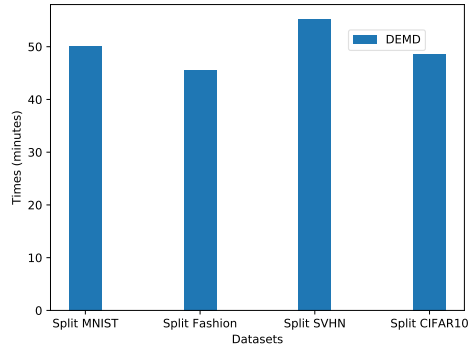


Figure 5: The computational costs required for the proposed memory approach without the model training procedure.

Split Fashion, Split SVHN and Split CIFAR10 and the results are presented in Fig. 5. The empirical results show that the optimization time for the proposed approach is less than one hour, which is computationally efficient.

References

- [1] Jihwan Bang, Hyunseo Koh, Seulki Park, Hwanjun Song, Jung-Woo Ha, and Jonghyun Choi. Online continual learning on a contaminated data stream with blurry task boundaries. In *Proc. of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9275–9284, June 2022. 2
- [2] Hyuntak Cha, Jaeho Lee, and Jinwoo Shin. Co2l: Contrastive continual learning. In *Proc. of the IEEE/CVF International Conference on Computer Vision (CVPR)*, pages 9516–9525, 2021. 2
- [3] Matthias De Lange and Tinne Tuytelaars. Continual prototype evolution: Learning online from non-stationary data streams. In *Proc. of the IEEE/CVF International Conference on Computer Vision*, pages 8250–8259, 2021. 4, 5
- [4] Yanan Gu, Xu Yang, Kun Wei, and Cheng Deng. Not just selection, but exploration: Online class-incremental continual learning via dual view consistency. In *Proc. of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7442–7451, June 2022. 2

- [5] Yiduo Guo, Bing Liu, and Dongyan Zhao. Online continual learning through mutual information maximization. In *International Conference on Machine Learning (ICLR)*, vol. *PMLR 162*, pages 8109–8126, 2022. 2
- [6] Fushuo Huo, Wenchao Xu, Jingcai Guo, Haozhao Wang, and Yunfeng Fan. Non-exemplar online class-incremental continual learning via dual-prototype self-augment and refinement. In *Proc. of the AAAI Conference on Artificial Intelligence*, volume 38, pages 12698–12707, 2024. 2
- [7] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations (ICLR)*, *arXiv preprint arXiv:1412.6980*, 2015. 3
- [8] Alex Krizhevsky and Geoffrey Hinton. Learning multiple layers of features from tiny images. Technical report, Univ. of Toronto, 2009. 4
- [9] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in Neural Inf. Proc. Systems (NIPS)*, pages 1097–1105, 2012. 5
- [10] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proc. of the IEEE*, 86(11):2278–2324, 1998. 4
- [11] Z. Liu, P. Luo, X. Wang, and X. Tang. Deep learning face attributes in the wild. In *Proc. of IEEE Int. Conf. on Computer Vision (ICCV)*, pages 3730–3738, 2015. 5
- [12] Y. Netzer, T. Wang, A. Coates, A. Bissacco, B. Wu, and A. Y. Ng. Reading digits in natural images with unsupervised feature learning. In *NIPS Workshop on Deep Learning and Unsupervised Feature Learning*, 2011. 5
- [13] R. Polikar, L. Upda, S. S. Upda, and Vasant Honavar. Learn++: An incremental learning algorithm for supervised neural networks. *IEEE Trans. on Systems Man and Cybernetics, Part C*, 31(4):497–508, 2001. 2
- [14] B. Ren, H. Wang, J. Li, and H. Gao. Life-long learning based on dynamic combination model. *Applied Soft Computing*, 56:398–404, 2017. 2
- [15] Hippolyt Ritter, Aleksandar Botev, and David Barber. Online structured Laplace approximations for overcoming catastrophic forgetting. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 31, pages 3742–3752, 2018. 2

- [16] Rishabh Tiwari, Krishnateja Killamsetty, Rishabh Iyer, and Pradeep Shenoy. GCR: Gradient coreset based replay buffer selection for continual learning. In *Proc. of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 99–108, 2022. 2
- [17] Oriol Vinyals, Charles Blundell, Timothy Lillicrap, Koray Kavukcuoglu, and Daan Wierstra. Matching networks for one shot learning. *Advances in Neural Information Processing Systems (NIPS)*, 29:3637–3645, 2016. 5
- [18] H. Xiao, K. Rasul, and R. Vollgraf. Fashion-MNIST: a novel image dataset for benchmarking machine learning algorithms. *arXiv preprint arXiv:1708.07747*, 2017. 4
- [19] Fei Ye and Adrian G Bors. Online task-free continual generative and discriminative learning via dynamic cluster memory. In *Proc. of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 26202–26212, 2024. 2