CODE-CL: Conceptor-Based Gradient Projection for Deep Continual Learning

Supplementary Material

A. Conceptor Implementation Details

We implement the conceptor operations following the equations presented in Section 2, with one exception: the AND operation (4).

The operation defined in (4) is only valid when the conceptor matrices are invertible. However, in practice, since we use a limited number of samples to compute the conceptors, the resulting matrices are often not full rank. To address this, we adopt a more general version of the AND operation, as proposed in [8]:

$$\boldsymbol{C} \wedge \boldsymbol{B} = \boldsymbol{D}(\boldsymbol{D}^\top(\boldsymbol{C}^\dagger + \boldsymbol{B}^\dagger - \boldsymbol{I})\boldsymbol{D})^{-1}\boldsymbol{D}^\top, \tag{11}$$

Here, $\boldsymbol{C}^\dagger$ and $\boldsymbol{B}^\dagger$ denote the pseudo-inverses of $\boldsymbol{C}$ and $\boldsymbol{B}$, respectively. The matrix $\boldsymbol{D}$ consists of columns that form an arbitrary orthonormal basis for the intersection of the column spaces of $\boldsymbol{C}$ and $\boldsymbol{B}$.

The procedure for computing $\boldsymbol{D}$ is outlined in Algorithm 2.

Algorithm 2 Computation of matrix $\boldsymbol{D}$ in (11)

Input: $\boldsymbol{C}$, $\boldsymbol{B}$, β (threshold), N (dimension of $\boldsymbol{C}$ and $\boldsymbol{B}$)
Output: $\boldsymbol{D}$

$\boldsymbol{U_C}, \boldsymbol{S_C} \leftarrow \text{SVD}(\boldsymbol{C})$ ▷ Singular value decomposition
$\boldsymbol{U_B}, \boldsymbol{S_B} \leftarrow \text{SVD}(\boldsymbol{B})$
$k_{\boldsymbol{C}} \leftarrow \text{num\_elements}(\boldsymbol{S_C} > \beta)$ ▷ # of elements $> \beta$
$k_{\boldsymbol{B}} \leftarrow \text{num\_elements}(\boldsymbol{S_B} > \beta)$
$\boldsymbol{U}'_{\boldsymbol{C}} \leftarrow \boldsymbol{U_C}[:, k_{\boldsymbol{C}} :]$ ▷ Last $N - k_{\boldsymbol{C}}$ columns
$\boldsymbol{U}'_{\boldsymbol{B}} \leftarrow \boldsymbol{U_B}[:, k_{\boldsymbol{B}} :]$
$\boldsymbol{U}, \boldsymbol{S} \leftarrow \text{SVD}(\boldsymbol{U}'_{\boldsymbol{C}}\boldsymbol{U}'^\top_{\boldsymbol{C}} + \boldsymbol{U}'_{\boldsymbol{B}}\boldsymbol{U}'^\top_{\boldsymbol{B}})$
$k \leftarrow \text{num\_elements}(\boldsymbol{S} > \beta)$
$\boldsymbol{D} \leftarrow \boldsymbol{U}[:, k :]$

B. Additional Ablation Studies

In this section, we present additional ablation studies to evaluate the impact of the number of free dimensions (K) and aperture (α) on the 5-Datasets benchmark, as well as the effect of the threshold parameter (ϵ) across all three benchmarks.

Tables 5 and 6 summarize the results on the 5-Datasets benchmark. We observe that increasing α leads to a reduction in BWT, consistent with the findings in Section 4. Similarly, increasing K improves final accuracy, further validating trends observed in the other datasets.

Regarding the threshold parameter (ϵ), results suggest that lower values of ϵ enhance performance by allowing more directions in the intersection of input spaces across tasks to be freed. However, this also increases memory requirements. Therefore, selecting an appropriate ϵ involves a trade-off between performance and computational resources.

Table 5. Ablation studies on the aperture (α) hyperparameter on the 5-Datasets benchmark. Results are reported as mean ± standard deviation over five trials. Other hyperparameters are constant as reported in Section 4.

α	ACC (%)	BWT (%)
4	93.32 ± 0.13	-0.25 ± 0.02
8	$\mathbf{93.51 \pm 0.13}$	-0.11 ± 0.01
16	93.46 ± 0.16	-0.04 ± 0.00

Table 6. Ablation studies on the number of free dimensions (K) parameter on the 5-Datasets benchmark. Results are reported as mean ± standard deviation over five trials. Other hyperparameters are constant as reported in Section 4.

K	ACC (%)	BWT (%)
0	91.67 ± 0.31	-1.36 ± 0.07
20	92.70 ± 0.07	-0.43 ± 0.01
40	93.08 ± 0.08	-0.33 ± 0.09
60	93.22 ± 0.16	-0.28 ± 0.00
80	$\mathbf{93.32 \pm 0.13}$	-0.25 ± 0.00

Table 7. Ablation studies on the threshold (ϵ) across the four benchmarks. Results are reported as mean ± standard deviation over five trials. Other hyperparameters are constant as reported in Section 4.

	ϵ	ACC (%)	BWT (%)
S-CIFAR100	0.2	$\mathbf{77.51 \pm 0.18}$	-0.84 ± 0.24
	0.5	77.21 ± 0.32	-1.10 ± 0.28
	0.8	75.71 ± 0.40	-0.93 ± 0.36
S-MiniImageNet	0.2	68.61 ± 0.94	-1.30 ± 0.18
	0.5	$\mathbf{68.83 \pm 0.41}$	-1.10 ± 0.30
	0.8	66.57 ± 0.24	-0.56 ± 0.18
5-Datasets	0.2	$\mathbf{93.42 \pm 0.11}$	-0.20 ± 0.06
	0.5	93.32 ± 0.13	-0.25 ± 0.02
	0.8	92.28 ± 0.24	-0.71 ± 0.18

C. Experimental Setup

This section provides details on the architecture of all models used in this work, the dataset statistics, the hyperparameters for each experiment, and the compute resources employed.

Table 8. 5-Datasets statistics.

Dataset	CIFAR10	MNIST	SVHN	Fashion MNIST	notMNIST
Number of classes	10	10	10	10	10
Training samples	47500	57000	69595	57000	16011
Validation samples	2500	3000	3662	3000	842
Test samples	10000	10000	26032	10000	1873

Table 9. List of hyperparameters used in our experiments.

Dataset	Split CIFAR100	Split miniImageNet	5-Datasets
Learning rate (η)	0.01	0.1	0.1
Batch size (b)	64	64	64
Batch size for conceptor comp. (b_s)	125	125	125
Min. learning rate (η_{th})	10^{-5}	10^{-5}	10^{-3}
Learning rate decay factor	$1/2$	$1/2$	$1/3$
Patience	6	6	5
Number of epochs (E)	200	100	100
Aperture (α)	6	8	4
Threshold (ϵ)	0.5	0.5	0.5

Table 10. Split CIFAR100 and Split miniImageNet datasets statistics.

Dataset	Split CIFAR100	Split miniImageNet
Number of tasks (T)	10	20
Sample dimensions	$3 \times 32 \times 32$	$3 \times 84 \times 84$
Number of classes per task	10	5
Training samples per task	4750	2375
Validation samples per task	250	125
Test samples per task	1000	500

C.1. Model Architecture

In this work, we utilize two models: an AlexNet-like architecture, as described in [26], and a Reduced ResNet18 [17].

The AlexNet-like model incorporates batch normalization (BN) in every layer except the classifier layer. The BN layers are trained during the first task and remain frozen for subsequent tasks. The model consists of three convolutional layers with 64, 128, and 256 filters, using kernel sizes of 4×4, 3×3, and 2×2, respectively. These are followed by two fully connected layers, each containing 2048 neurons. ReLU activation functions are used throughout, along with 2×2 max-pooling layers after each convolutional layer. Dropout is applied with rates of 0.2 for the first two layers and 0.5 for the remaining layers.

The Reduced ResNet18 follows the architecture detailed in [24]. For the Split miniImageNet experiments, the first layer uses a stride of 2, while for the 5-Datasets benchmark, it uses a stride of 1.

For all models and experiments, cross-entropy loss is employed as the loss function.

C.2. Dataset Statistics

The statistics for the four benchmarks used in this work for continual image classification are summarized in Table 10 and Table 8. For all benchmarks, we follow the same data partitions as those used in [15, 23, 24].

For the 5-Datasets benchmark, grayscale images are replicated across all RGB channels to ensure compatibility with the architecture. Additionally, all images are resized to 32×32 pixels, resulting in an input size of $3 \times 32 \times 32$ for this benchmark.

C.3. Hyperparameters

The hyperparameters used in our experiments are detailed in Table 9.

C.4. Compute resources

All experiments were conducted on a shared internal Linux server equipped with an AMD EPYC 7502 32-Core Processor, 504 GB of RAM, and four NVIDIA A40 GPUs, each with 48 GB of GDDR6 memory. Additionally, code was implemented using Python 3.9 and PyTorch 2.2.1 with CUDA 11.8.

D. ViT and task-agnostic evaluation

While our main results use CNNs, CODE-CL is architecture-agnostic. For instance, when fine-tuning ViTs with LoRA (i.e., $\boldsymbol{W}^{\text{new}} = \boldsymbol{W}^{\text{old, fixed}} + \boldsymbol{BA}$), CODE-CL can be applied to $\nabla_{\boldsymbol{A}}\mathcal{L}$ as $\nabla_{\boldsymbol{A}}\mathcal{L} := \nabla_{\boldsymbol{A}}\mathcal{L} - \nabla_{\boldsymbol{A}}\mathcal{L}\boldsymbol{C}^{t-1}$ to mitigate forgetting. Using the setup from [14], we extended CODE-CL to ViTs. Initial results (Table 11) show that CODE-CL outperforms [14] in a task-agnostic class-incremental setting. These findings highlight the potential of CODE-CL to extend to ViTs and task-agnostic CL.

Table 11. Class-Incremental Learning with ViT on Split CIFAR100.

Methods	InfLoRA[14]	CODE-CL (Ours)
Accuracy	87.06 ± 0.25	$\mathbf{88.23 \pm 0.20}$

References

[1] Antonio Carta, Lorenzo Pellegrini, Andrea Cossu, Hamed Hemati, and Vincenzo Lomonaco. Avalanche: A PyTorch Library for Deep Continual Learning. *Journal of Machine Learning Research*, 24(363):1–6, 2023. 5

[2] Arslan Chaudhry, Marcus Rohrbach Facebook, A I Research, Mohamed Elhoseiny, Thalaiyasingam Ajanthan, Puneet K Dokania, Philip H S Torr, and Marc ' Aurelio Ranzato. On Tiny Episodic Memories in Continual Learning. *arXiv:1902.10486*, 2019. 1, 3, 6

[3] Arslan Chaudhry, Marc'Aurelio Ranzato, Marcus Rohrbach, and Mohamed Elhoseiny. Efficient Lifelong Learning with A-GEM. *International Conference on Learning Representations*, 2019. 1, 3, 6

[4] Sayna Ebrahimi, Franziska Meier, Roberto Calandra, Trevor Darrell, and Marcus Rohrbach. Adversarial Continual Learning. In *Computer Vision – ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XI*, pages 386–402, Berlin, Heidelberg, 2020. Springer-Verlag. 5

[5] Mehrdad Farajtabar, Navid Azizan, Alex Mott, Ang Li, Deepmind Caltech, and Deepmind Deepmind. Orthogonal Gradient Descent for Continual Learning. In *Proceedings of the Twenty Third International Conference on Artificial Intelligence and Statistics*, pages 3762–3773. PMLR, 2020. 1

[6] Raia Hadsell, Dushyant Rao, Andrei A. Rusu, and Razvan Pascanu. Embracing Change: Continual Learning in Deep Neural Networks. *Trends in Cognitive Sciences*, 24(12): 1028–1040, 2020. 1

[7] Xu He and H. Jaeger. Overcoming Catastrophic Interference using Conceptor-Aided Backpropagation. *International Conference on Learning Representations*, 2018. 3

[8] Herbert Jaeger. Controlling Recurrent Neural Networks by Conceptors. *arXiv:1403.3369*, 2014. 1, 2, 3

[9] Zixuan Ke, Bing Liu, Nianzu Ma, Hu Xu, and Lei Shu. Achieving forgetting prevention and knowledge transfer in continual learning. In *Proceedings of the 35th International Conference on Neural Information Processing Systems*, Red Hook, NY, USA, 2021. Curran Associates Inc. 1

[10] James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A. Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, Demis Hassabis, Claudia Clopath, Dharshan Kumaran, and Raia Hadsell. Overcoming catastrophic forgetting in neural networks. *Proceedings of the National Academy of Sciences of the United States of America*, 114 (13):3521–3526, 2017. 1, 3, 6

[11] Alex Krizhevsky. Learning Multiple Layers of Features from Tiny Images. 2009. 5

[12] Dhireesha Kudithipudi, Mario Aguilar-Simon, Jonathan Babb, Maxim Bazhenov, Douglas Blackiston, Josh Bongard, Andrew P. Brna, Suraj Chakravarthi Raja, Nick Cheney, Jeff Clune, Anurag Daram, Stefano Fusi, Peter Helfer, Leslie Kay, Nicholas Ketz, Zsolt Kira, Soheil Kolouri, Jeffrey L. Krichmar, Sam Kriegman, Michael Levin, Sandeep Madireddy, Santosh Manicka, Ali Marjaninejad, Bruce McNaughton, Risto Miikkulainen, Zaneta Navratilova, Tej Pandit, Alice Parker, Praveen K. Pilly, Sebastian Risi, Terrence J. Sejnowski, Andrea Soltoggio, Nicholas Soures, Andreas S. Tolias, Darío Urbina-Meléndez, Francisco J. Valero-Cuevas, Gido M. van de Ven, Joshua T. Vogelstein, Felix Wang, Ron Weiss, Angel Yanguas-Gil, Xinyun Zou, and Hava Siegelmann. Biological underpinnings for lifelong learning machines. *Nature Machine Intelligence 2022 4:3*, 4(3):196–210, 2022. 1

[13] Yan-Shuo Liang and Wu-Jun Li. Adaptive plasticity improvement for continual learning. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023. 3, 6

[14] Yan-Shuo Liang and Wu-Jun Li. Inflora: Interference-free low-rank adaptation for continual learning. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 23638–23647, 2024. 3

[15] Sen Lin, Li Yang, Deliang Fan, and Junshan Zhang. TRGP: Trust Region Gradient Projection for Continual Learning. *International Conference on Learning Representations*, 2022. 1, 3, 5, 6, 7, 8, 2

[16] Sen Lin, Li Yang, Deliang Fan, and Junshan Zhang. Beyond not-forgetting: continual learning with backward knowledge transfer. In *Proceedings of the 36th International Conference on Neural Information Processing Systems*, Red Hook, NY, USA, 2022. Curran Associates Inc. 1, 3, 6, 8

[17] David Lopez-Paz and Marc ' Aurelio Ranzato. Gradient Episodic Memory for Continual Learning. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, 2017. 1, 3, 5, 2

[18] Arun Mallya and Svetlana Lazebnik. PackNet: Adding Multiple Tasks to a Single Network by Iterative Pruning. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7765–7773, 2017. 1, 3

[19] Qi Qin, Wenpeng Hu, Han Peng, Dongyan Zhao, and Bing Liu. BNS: Building Network Structures Dynamically for Continual Learning. *Advances in Neural Information Processing Systems*, 34:20608–20620, 2021. 1, 3

[20] Sylvestre Alvise Rebuffi, Alexander Kolesnikov, Georg Sperl, and Christoph H. Lampert. iCaRL: Incremental Classifier and Representation Learning. *2017 IEEE Conference*

on Computer Vision and Pattern Recognition (CVPR), 2017-January:5533–5542, 2017. 1, 3

[21] H. Ritter, Aleksandar Botev, and D. Barber. Online Structured Laplace Approximations For Overcoming Catastrophic Forgetting. *Neural Information Processing Systems*, 2018. 3

[22] Andrei A. Rusu, Neil C. Rabinowitz, Guillaume Desjardins, Hubert Soyer, James Kirkpatrick, Koray Kavukcuoglu, Razvan Pascanu, and Raia Hadsell. Progressive Neural Networks. *arXiv preprint arXiv:1606.04671*, 2016. 1, 3

[23] Gobinda Saha and Kaushik Roy. Continual Learning with Scaled Gradient Projection. *Proceedings of the 37th AAAI Conference on Artificial Intelligence, AAAI 2023*, 37:9677–9685, 2023. 1, 2, 3, 5, 6, 8

[24] Gobinda Saha, Isha Garg, and K. Roy. Gradient Projection Memory for Continual Learning. *International Conference on Learning Representations*, 2021. 1, 2, 3, 5, 6, 8

[25] Jonathan Schwarz, Wojciech M. Czarnecki, Jelena Luketina, A. Grabska-Barwinska, Y. Teh, Razvan Pascanu, and R. Hadsell. Progress & Compress: A scalable framework for continual learning. *International Conference on Machine Learning*, 2018. 3

[26] J. Serrà, Dídac Surís, M. Miron, and Alexandros Karatzoglou. Overcoming catastrophic forgetting with hard attention to the task. *International Conference on Machine Learning*, 2018. 1, 3, 6, 2

[27] Yujun Shi, Li Yuan, Yunpeng Chen, and Jiashi Feng. Continual Learning via Bit-Level Information Preserving. *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 16669–16678, 2021. 1

[28] O. Vinyals, C. Blundell, T. Lillicrap, K. Kavukcuoglu, and Daan Wierstra. Matching Networks for One Shot Learning. *Neural Information Processing Systems*, 2016. 5

[29] Liyuan Wang, Xingxing Zhang, Hang Su, and Jun Zhu. A Comprehensive Survey of Continual Learning: Theory, Method and Application. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 46(08):5362–5383, 2024. 1, 3, 4

[30] Shipeng Wang, Xiaorong Li, Jian Sun, and Zongben Xu. Training Networks in Null Space of Feature Covariance for Continual Learning. *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021. 1

[31] Ju Xu and Zhanxing Zhu. Reinforced Continual Learning. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, pages 907–916, 2018. 1, 3

[32] Enneng Yang, Li Shen, Zhenyi Wang, Shiwei Liu, Guibing Guo, and Xingwei Wang. Data augmented flatness-aware gradient projection for continual learning. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 5607–5616, 2023. 3, 6

[33] Jaehong Yoon, Eunho Yang, Jeongtae Lee, and Sung Ju Hwang. Lifelong Learning with Dynamically Expandable Networks. *International Conference on Learning Representations*, 2018. 1, 3

[34] Jaehong Yoon, Saehoon Kim, Eunho Yang, and Sung Ju Hwang. Scalable and Order-robust Continual Learning with Additive Parameter Decomposition. *International Conference on Learning Representations*, 2020. 1

[35] Guanxiong Zeng, Yang Chen, Bo Cui, and Shan Yu. Continual learning of context-dependent processing in neural networks. *Nature Machine Intelligence 2019 1:8*, 1(8):364–372, 2019. 1, 3, 6

[36] Friedemann Zenke, Ben Poole, and Surya Ganguli. Continual Learning Through Synaptic Intelligence. *Proceedings of machine learning research*, 70:3987, 2017. 1, 3

[37] Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. Understanding deep learning requires rethinking generalization. In *International Conference on Learning Representations*, 2017. 2, 3

[38] Zhen Zhao, Zhizhong Zhang, Xin Tan, Jun Liu, Yanyun Qu, Yuan Xie, and Lizhuang Ma. Rethinking gradient projection continual learning: Stability/plasticity feature space decoupling. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023. 3, 6