Supplementary Material
Continual Learning for Visual Search with Backward Consistent Feature Embedding

Timmy S. T. Wan[1] Jun-Cheng Chen[2] Tzer-Yi Wu[3] Chu-Song Chen[1,*]
National Taiwan University[1] Academia Sinica[2] Ucfunnel Co. Ltd.[3]
{r08944004,chusong}@csie.ntu.edu.tw, pullpull@citi.sinica.edu.tw, kenny.wu@ucfunnel.com

A. Contribution Review

General Incremental Setup: Unlike previous works, we investigate a general case for CL (Fig. A.1). Our setup considers the incremental classes of the disjoint setup and also covers overlapped classes of the blurry setup. In a broad sense, it offers a more common situation for the CL research community on both classification and retrieval.

Backward Consistent Feature Space Learning: We propose a novel continual learner for visual search allowing acquiring knowledge for unseen classes and making both the previous and the current feature space comparable without backfilling (*i.e.*, re-extraction) of the previously processed gallery images. We bridge this gap in three loss terms:

- An inter-session data coherence loss learns from the history of all sessions by taking the extensible replayed embedding as a free supervision signal for guidance.
- A neighbor-session model coherence loss preserves the distance metric for the seen classes in both new and old sessions; it leverages a revised triplet loss with a new sampling strategy for distillation.
- An intra-session discrimination loss grasps knowledge from the novel categories using pointwise metric learning without loss of flexibility.

An enlarged Fig. 2 of the main paper is provided for a more precise illustration, as shown in Fig. A.2. In the following, we complement more implementation details in Section B and ablation studies in Section C.

B. Hyperparameter Details

As presented in Section 4.1 of the main paper, we show the hyperparameter details as follows. We implement all models with Pytorch [13] using NVIDIA V100 GPUs. To control the embedding size, we insert the fully connected layer with dimension 128 before the final softmax layer given the network architecture. The full experiments on the general incremental setup are shown in Tables C.1 and C.2.

* indicates corresponding author.

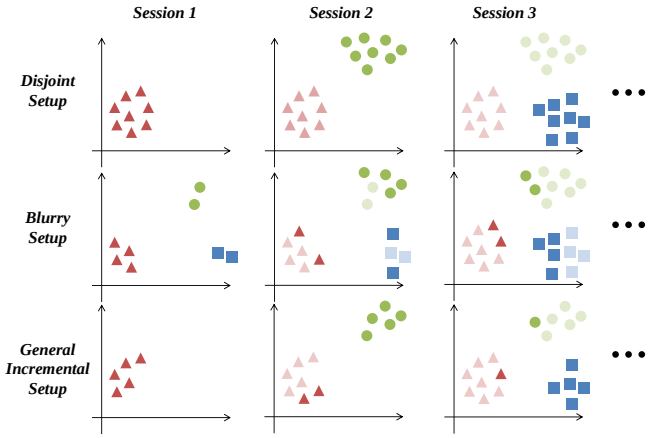

Figure A.1. The widely adopted Disjoint setup (upper row) assumes the image categories mutually disjoint among sessions. The middle row shows the recent Blurry setup, where different sessions allow overlapping classes but all the classes are given initially; every session has a specific data distribution over the known classes. The bottom row shows our General Incremental setup, where the classes in a new session can be either old or novel.

B.1. Details on the Coarse-grained Datasets

The hyperparameters almost follow those in Rainbow [1] but with different batch sizes for Tiny ImageNet. We train ResNet-18 [6] over 256 epochs with the batch size of 16 and 64 for CIFAR100 and Tiny ImageNet, respectively. The networks are optimized using SGD with an initial learning rate of 0.03, a momentum of 0.9, and a weight decay of 0.0001. We adjust the learning rate in the range between 0.03 and 0.0003 by the cosine learning rate scheduler [10]. About data augmentation, training images from CIFAR100 are padded by 4 pixels on all borders and then preprocessed through randomly cropping at 32×32, randomly horizontal flipping followed by AutoAug [4]. For Tiny ImageNet, we follow the similar augmentation process but use randomly cropping at 64×64 and RandAug [5] instead.

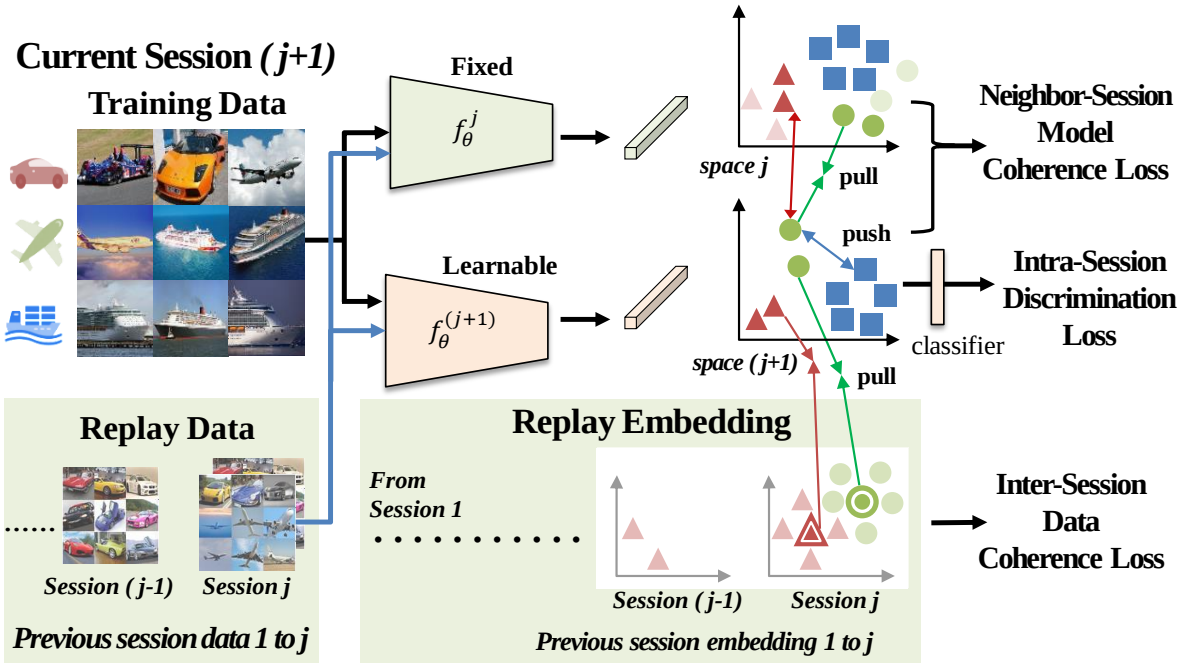

Figure A.2. Overview of our CVS method for CL in General-Incremental setup with long-term backward embedding consistency. In the current session $j+1$, the training data of the session (together with the replayed data under a budget control) are used in the three loss terms. In addition, the replayed embedding summarized from previous sessions $1:j$ serves as historically concentrated attractors to guide the inter-session data-coherence training; it acts as an extension of cross-batch memory [17, 18] to cross-session memory in CL. We introduce a 2-sample-3-embedding strategy in a triplet for distillation learning across neighbor sessions to enforce the model coherence. Note that we omit the replayed data to simplify the illustration. We use a L2-normalized embedding in classification [20] to provide the intra-session discriminating capability, and normalized embedding is adopted in all three loss terms. Our approach is simple but effective in all three CL setups, and we provide the first study on general-incremental setup in CL.

Figure A.3. Sample images of fine-grained datasets. The first pair is from Stanford Dog, the latter pair is from iNaturalist 2017, and the final four images are from Product-10K.

B.2. Details on the Fine-grained Datasets

Some fine-grained samples (Stanford Dog, iNat-M, and Product-M datasets) are shown in Fig. A.3. They have only subtle changes between classes and are more demanding for retrieval. Following similar experimental settings mentioned in [11, 15], we finetune the ImageNet-pretrained ResNet-50 for 100 epochs using SGD with a small fixed learning rate of 0.0001. The batch size is 64 by default but 32 for Product-M. We follow [15] to preprocess the training images by randomly resizing and cropping them to 224×224 with random horizontal flipping. At testing time, we emphasize the object by central cropping of 224×224 from the 256×256 resized image for feature extraction.

B.3. Reimplementation details

We re-implement the LWF [9], MMD [2], and BCT [16] in our experiments. All loss terms are equal weighting to meet the balance between previously learned information and new knowledge. For MMD, the maximum mean discrepancy loss is solely used for blurry setup, and an additional knowledge distillation loss is applied to the novel class data according to the original definition under the disjoint and general-incremental setups. For BCT, we make some modifications to suit for different setups as it is not a CL solution and requires all old class samples collected so far. We use all the old samples from the seen classes for the blurry and general-incremental setups, and employ the replayed data mined by iCaRL [14] for the disjoint setup.

Figure A.4. **Qualitative comparison** of the top-4 results using our method (CVS) and Finetune on the Product-M dataset. The correct and incorrect matches are highlighted in green and red, respectively.

B.4. Qualitative Study

We demonstrate some qualitative results in Fig. A.4. Our method maintains the backward consistency and captures fine-grained characteristics of the particular object. *E.g.*, the first row shows that our method can retrieve visually similar bottles, but Fine-tune yields the results with perturbation.

C. Ablation study on different losses for neighbor-session model coherence

As referenced in Section 4.3 of the main paper, we provide a complete experiment result as follows.

Comparisons to Other Embedding Distillations: We perform an in-depth analysis of different metric losses for the loss term $\mathbf{L}^{\mathbf{m}}_{j:j+1}$, as shown in Table 3. Dark Knowledge [8] minimizes the KL divergence on the classifier side. Absolute MLKD [19] performs distillation at the penultimate layer output. By estimating relational structural information given a mini-batch, RKD [12] reduces the Huber loss using pairwise euclidean distance difference between two models (i.e., Distancewise RKD) and angle from three points (i.e., Anglewise RKD). Following the same spirit, Relative MLKD [19] uses the difference of Frobenius norm instead, and DarkRank [3] re-estimates the similarity ranks using listwise relationships. Except for Dark Knowledge, we use their official Github implementation for fair comparisons. We tune hyperparameter α at $\{1, 10\}$ for the best AR@1 value at the validation phase for importance weighting. As can be seen in Table C.3, our method provides the most favorable results at AR@k when k is small (1 or 2) on all setups, and only slightly inferior to Relative MLKD and DarkRank at AR@4 on the blurry and general-incremental setups, respectively.

Comparisons to Other Sample Mining Strategy: We examine different mining techniques because $\mathbf{L}^{\mathbf{m}}_{j;j+1}$ is computed based on the sampled triplets. Instead of our easiest positive mining, we use the hardest positive mining (*i.e.*, **BatchHard**, a hardest-positive-hardest-negative online mining strategy mentioned in [7]) for fair comparisons (Table. C.4). We implement BatchHard with a balanced batch sampler to enforce each batch containing at least two samples per class for forming sufficient valid triplets. Unfortunately, BatchHard obtains the worst result or even lower than the one with $\mathbf{L}^{\mathbf{m}}_{j;j+1}$ disabled. We attribute this result to the misleading guidance due to a large variation between feature spaces; thus, mining triplets according to the cross-session distance is unreliable. On the other hand, our strategy gains the best result by forming the positive samples using the outputs from two models without explicit mining. Therefore, our triplet mining design is simple but effective, and easy to implement.

D. More Technical Analysis

Results with different memory budgets: The experiment about the influence of the buffer size is provided in Table. C.5. We observe that CVS consistently beats other replay-based methods on AR@1 using half the budget under the general-incremental setup.

Multiple run results: We present the results of averaging 3 runs on all selected datasets. With Table. C.6, most methods show no significant deviation ($< 1\%$) except for RWalk in Tiny ImageNet. But, this doesn't change any conclusions in our main paper.

References

[1] Jihwan Bang, Heesu Kim, YoungJoon Yoo, Jung-Woo Ha, and Jonghyun Choi. Rainbow memory: Continual learning with a memory of diverse samples. In *CVPR*, pages 8218–8227, 2021.

[2] Wei Chen, Yu Liu, Weiping Wang, Tinne Tuytelaars, Erwin M Bakker, and Michael Lew. On the exploration of incremental learning for fine-grained image retrieval. In *BMVC*, 2020.

[3] Yuntao Chen, Naiyan Wang, and Zhaoxiang Zhang. Darkrank: Accelerating deep metric learning via cross sample similarities transfer. In *AAAI*, pages 2852–2859, 2018.

[4] Ekin D. Cubuk, Barret Zoph, Dandelion Mane, Vijay Vasudevan, and Quoc V. Le. Autoaugment: Learning augmentation strategies from data. In *CVPR*, pages 113–123, 2019.

[5] Ekin D. Cubuk, Barret Zoph, Jonathon Shlens, and Quoc V. Le. Randaugment: Practical automated data augmentation with a reduced search space. In *CVPRW*, pages 3008–3017, 2020.

[6] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, pages 770–778, 2016.

[7] Alexander Hermans, L. Beyer, and B. Leibe. In defense of the triplet loss for person re-identification. *arXiv preprint arXiv:1703.07737*, 2017.

[8] Geoffrey Hinton, Oriol Vinyals, and Jeffrey Dean. Distilling the knowledge in a neural network. In *NeurIPSW*, 2015.

[9] Zhizhong Li and Derek Hoiem. Learning without forgetting. *PAMI*, 40:2935–2947, 2018.

[10] Ilya Loshchilov and Frank Hutter. Sgdr: Stochastic gradient descent with warm restarts. In *ICLR*, 2017.

[11] Kevin Musgrave, Serge J. Belongie, and Ser-Nam Lim. A metric learning reality check. In *ECCV*, pages 681–699, 2020.

[12] Wonpyo Park, Dongju Kim, Yan Lu, and Minsu Cho. Relational knowledge distillation. In *CVPR*, pages 3962–3971, 2019.

[13] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in pytorch. In *NeurIPSW*, 2017.

[14] Sylvestre-Alvise Rebuffi, Alexander Kolesnikov, Georg Sperl, and Christoph H. Lampert. icarl: Incremental classifier and representation learning. In *CVPR*, pages 5533–5542, 2017.

[15] Karsten Roth, Timo Milbich, Samarth Sinha, Prateek Gupta, Bjorn Ommer, and Joseph Paul Cohen. Revisiting training strategies and generalization performance in deep metric learning. In *ICML*, pages 8242–8252, 2020.

[16] Yantao Shen, Yuanjun Xiong, Wei Xia, and Stefano Soatto. Towards backward-compatible representation learning. In *CVPR*, pages 6367–6376, 2020.

[17] Xun Wang, Haozhi Zhang, Weilin Huang, and Matthew R. Scott. Cross-batch memory for embedding learning. In *CVPR*, pages 6387–6396, 2020.

[18] Huei-Fang Yang, Kevin Lin, and Chu-Song Chen. Cross-batch reference learning for deep classification and retrieval. In *ACM MM*, pages 1237—-1246, 2016.

[19] Lu Yu, Vacit Oguz Yazici, Xialei Liu, Joost van de Weijer, Yongmei Cheng, and Arnau Ramisa. Learning metrics from teachers: Compact networks for image embedding. In *CVPR*, pages 2902–2911, 2019.

[20] Andrew Zhai and Hao-Yu Wu. Classification is a strong baseline for deep metric learning. In *BMVC*, 2019.

	CIFAR100			Tiny ImageNet			Dog			iNat-M			Product-M		
	AR@1	AR@2	AR@4	AR@1	AR@2	AR@4	AR@1	AR@2	AR@4	AR@1	AR@2	AR@4	AR@1	AR@2	AR@4
Joint Train	81.97	84.6	86.82	47.45	51.94	55.99	86.98	91.08	93.99	75.85	79.97	83.65	79.36	83.83	87.73
Finetune	60.79	64.73	68.14	31.11	35.87	40.8	82.7	88.32	92.23	67.75	72.68	77	70.99	76.26	81.16
BCT	58.31	62.2	65.66	30.17	34.64	39.29	81.73	87.49	91.55	67.34	72.07	75.99	70.48	75.67	80.47
LWF	65.53	70.91	75.31	32.22	38.23	44.56	83.25	89.29	93.04	68.51	73.71	78.12	72.95	78.38	83.1
MMD	65.51	70.22	74.33	33.35	38.21	43.06	83.2	88.98	92.71	68.58	73.48	77.79	72.89	78.21	82.96
EWC	60.89	64.86	68.2	27.86	32.67	37.78	81.64	88.7	92.82	66.3	71.4	75.82	66.01	72.53	78.24
RWalk	69.9	73.37	76.39	33.83	38.43	42.97	82.41	88.31	92.43	68.77	73.82	78.16	68.71	74.64	80.13
Rainbow	68.4	71.56	74.2	37.53	41.64	45.44	82.78	89.04	**93.23**	68.68	73.22	77.21	69.39	75.19	80.34
Ours (CVS)	**73.95**	**76.73**	**78.84**	**38.78**	**42.38**	**45.89**	**84.71**	**89.4**	92.61	**72.57**	**76.39**	**79.87**	**75.47**	**80.36**	**84.68**
CVS w/o replay	67.61	71.3	74.39	36.39	40.32	44.2	83.51	88.96	92.45	71.31	75.25	78.56	74.19	78.95	83.27

Table C.1. Results on our general incremental setup. The champion is highlighted in bold and the runner-up is underlined in red.

	CIFAR100			Tiny ImageNet			Dog			iNat-M			Product-M		
	AR@1	AR@2	AR@4	AR@1	AR@2	AR@4	AR@1	AR@2	AR@4	AR@1	AR@2	AR@4	AR@1	AR@2	AR@4
CVS: $\mathbf{L^c + L^m + L^d}$	**73.95**	**76.73**	**78.74**	**38.78**	**42.38**	**45.89**	**84.71**	**89.4**	**92.61**	72.57	**76.39**	**79.87**	75.47	**80.36**	**84.68**
CVS w/o replay	67.61	71.3	74.39	36.39	40.32	44.2	83.51	88.96	92.45	71.31	75.25	78.56	74.19	78.95	83.27
$\mathbf{L^c + L^d}$	72.16	74.67	76.7	38.4	41.22	43.87	84.37	89.07	92.46	**72.61**	76.27	79.45	**75.6**	80.27	84.33
$\mathbf{L^c + L^d}$ w/o replay	64.5	68.12	71.14	32.82	36.2	39.55	83.29	88.77	92.38	70.94	74.62	77.86	73.96	78.69	83.02
$\mathbf{L^c + L^m}$	63.79	68.13	72.02	32.1	37.15	42.78	82.65	88.43	92.29	67.79	72.64	76.97	71.77	76.92	81.78
$\mathbf{L^c}$	60.79	64.73	68.14	31.11	35.87	40.8	82.7	88.32	92.23	67.75	72.68	77	70.99	76.26	81.16

Table C.2. Ablation results on our general incremental setup. The champion is highlighted in bold and the runner-up is underlined in red.

	Disjoint			Blurry			General		
	AR@1	AR@2	AR@4	AR@1	AR@2	AR@4	AR@1	AR@2	AR@4
None	69.85	72.67	74.58	44.33	46.14	47.72	72.16	74.67	76.7
Dark Knowledge	69.41	73.09	76.05	45.89	47.76	49.5	73.18	76.34	78.68
Absolute MLKD	66.04	70.36	74.27	46.9	48.89	50.77	72.43	75.72	78.47
Relative MLKD	66.64	71.2	75.35	44.1	48.25	**52.33**	70.91	75.19	78.92
Anglewise RKD	70.45	72.65	74.67	45.07	46.99	48.67	72.65	74.67	76.33
Distancewise RKD	70.1	72.29	74.11	45.09	47.01	48.81	72.83	75.17	77.14
Hard DarkRank	66.45	71.03	75.11	44.8	48.2	51.35	72.33	76.15	**79.56**
Ours (CVS)	**71.47**	**74.8**	**77.51**	**47.47**	**49.86**	52.17	**73.95**	**76.73**	78.84

Table C.3. Replace $\mathbf{L^m_{j;j+1}}$ with different metric distillation losses on CIFAR100. *None* disables $\mathbf{L^m_{j;j+1}}$ for a simple baseline.

	Disjoint			Blurry			General		
	AR@1	AR@2	AR@4	AR@1	AR@2	AR@4	AR@1	AR@2	AR@4
None	69.85	72.67	74.58	44.33	46.14	47.72	72.16	74.67	76.7
BatchHard	60.83	63.91	66.67	39.24	42.59	45.7	68.1	71.02	73.35
Ours (CVS)	**71.47**	**74.8**	**77.51**	**47.47**	**49.86**	**52.17**	**73.95**	**76.73**	**78.84**

Table C.4. Compute $\mathbf{L^m_{j;j+1}}$ with different mining strategies on CIFAR100.

	CIFAR100			Tiny ImageNet		
	AR@1	AR@2	AR@4	AR@1	AR@2	AR@4
RWalk (0.5× budget)	66.9	70.32	73.59	31.42	36.39	41.41
Rainbow (0.5× budget)	65.21	69.33	72.85	34.78	39.8	44.83
CVS (0.5× budget)	71.99	74.94	77.25	37.72	41.58	45.59
RWalk (1× budget)	69.9	73.37	76.39	33.83	38.43	42.97
Rainbow (1× budget)	68.4	71.56	74.2	37.53	41.64	45.44
CVS (1× budget)	73.95	76.73	78.84	38.78	42.38	45.89

Table C.5. Results with different memory budgets on our general incremental setup. 1× budget represents the continual learner with a memory buffer size of 2000 samples in CIFAR100 and 4000 samples in TinyImageNet. 0.5× indicates using half the budget.

	CIFAR100			Tiny ImageNet			Dog			iNat-M			Product-M		
	AR@1	AR@2	AR@4	AR@1	AR@2	AR@4	AR@1	AR@2	AR@4	AR@1	AR@2	AR@4	AR@1	AR@2	AR@4
BCT	58.01±0.26	62.13±0.12	65.84±0.17	30.11±0.43	34.89±0.36	39.72±0.51	81.77±0.07	87.33±0.23	91.34±0.24	67.17±0.31	71.88±0.22	75.95±0.16	70.55±0.1	75.83±0.17	80.67±0.4
LWF	65.39±0.24	70.6±0.36	75.06±0.29	32.17±0.59	38.3±0.41	44.81±0.26	83.45±0.33	89.19±0.21	93.01±0.15	68.34±0.3	73.71±0.04	78.2±0.07	73.11±0.14	78.42±0.04	83.01±0.19
MMD	65.15±0.32	69.98±0.22	74.18±0.21	33.3±0.3	38.47±0.27	43.57±0.46	83.53±0.32	89.01±0.15	92.65±0.06	68.63±0.31	73.6±0.26	77.85±0.06	72.81±0.27	78.16±0.17	82.84±0.23
EWC	60.9±0.14	65.1±0.55	68.76±0.92	28.01±0.88	33.21±0.66	38.55±0.66	82.17±0.77	89.1±0.68	93.21±0.54	66.18±0.1	71.53±0.12	76.02±0.17	66.33±0.28	72.53±0.04	78.04±0.17
RWalk	69.69±0.27	73.16±0.19	76.23±0.14	34.05±1.17	38.77±1.07	43.48±0.98	82.53±0.27	88.67±0.33	92.81±0.34	68.85±0.17	73.9±0.11	78.15±0.05	69.05±0.31	74.45±0.18	79.36±0.67
Rainbow	68.18±0.2	71.32±0.27	73.96±0.23	37.47±0.51	41.86±0.38	45.89±0.44	83.06±0.29	89.1±0.1	93.15±0.09	68.85±0.17	73.43±0.33	77.41±0.36	69.66±0.25	74.91±0.3	79.76±0.56
Ours (CVS)	73.81±0.15	76.48±0.26	78.58±0.27	38.56±0.33	42.03±0.07	46.01±0.23	84.64±0.07	89.5±0.09	92.85±0.21	72.7±0.13	76.47±0.13	79.74±0.23	75.77±0.35	80.49±0.21	84.7±0.19

Table C.6. Average results of 3 runs on our general incremental setup.