Supplementary Material for Preserving Linear Separability in Continual Learning by Backward Feature Projection

Qiao Gu
University of Toronto
qgu@cs.toronto.edu

Dongsub Shim
LG AI Research
dongsub.shim@lgresearch.ai

Florian Shkurti
University of Toronto
florian@cs.toronto.edu

A. Acknowledgements

The authors thank Jongseong Jang, Yizhou (Philip) Huang, Kevin Xie and Nikita Dhawan for discussions and useful feedback.

B. Implementation Details

B.1. Complete Algorithm

In Algorithm 1, we provide the pseudocode of continual learning with the proposed BFP method. Note that following [8], we sample training datapoints from the memory buffer for each loss independently. We empirically find this results in better performance than using the same set of replayed samples for all losses. The images without augmentation x_o are pushed into the memory and replayed images are augmented on the fly. The classification model is trained using an SGD optimizer (*sgd*) and the projection matrix A is trained using an SGD+Momentum optimizer (*sgdm*).

B.2. Class-balanced Reservoir Sampling

We adopt the class-balanced reservoir sampling (BRS) [9] for memory buffer management. The detail of this algorithm is described in Algorithm 2. Compared to regular Reservoir Sampling (RS), BRS ensures that every class has an equal number of examples stored in the memory buffer. All experiments are incorporated with this change. Empirically we find that BRS does not bring significant changes compared to RS, but it helps to reduce variance in the results.

B.3. Training details

Image sizes are 32×32 in Split-CIFAR10 and Split-CIFAR100 and 64×64 in Split-TinyImageNet. All experiments use the same data augmentation procedure, applied on input from both the current task and the memory buffer independently. Data augmentation includes a full-size random crop with a padding of 4 pixels, a random horizontal flip, and normalization.

For all experiments involving BFP, the optimizer for the matrix A is an SGD+Momentum optimizer with a learning rate of 0.1 and momentum of 0.9. The weighting term γ in Equation 12 is 1. Empirically we find that the BFP performance is not sensitive to these hyperparameters, and we use this one set of hyperparameters for BFP loss in all experiments.

B.4. Hyperparameters

In this section, we list the best hyperparameters used for the compared baselines mentioned in Section 4.2 and their results are reported Table 1. These hyperparameters are adopted from [8] and [7], where they were selected by a hy-

Algorithm 1 - Continual Learning with BFP

```
Input: dataset $\{\mathcal{D}_1, \cdots, \mathcal{D}_T\}$, parameters $\theta = \{\phi, \psi\}$, scalars
  $\alpha$, $\beta$ and $\gamma$, optimizer sgd, sgdm,
  $M \leftarrow \{\}$
for $t$ from $1$ to $T$ do
    $A \leftarrow$ random-init()
    sgdm $\leftarrow$ reinit(sgdm)
    for $(x_o, y_o)$ in $\mathcal{D}_t$ do
        $x, y \leftarrow$ augment$(x_o, y_o)$
        $L \leftarrow$ cross-entropy$(y, f_\theta(x))$                      {Eq. 8}
        if $t > 1$ then
            $x, y \leftarrow$ augment(sample$(M)$)
            $L_{\text{rep-ce}} \leftarrow$ cross-entropy$(y, f_\theta(x))$             {Eq. 9}
            $x, y \leftarrow$ augment(sample$(M)$)
            $L_{\text{rep-logits}} \leftarrow \|f_\theta(x) - f_{\text{old}}(x)\|_2$            {Eq. 10}
            $x, y \leftarrow$ augment(sample$(M)$)
            $L_{\text{BFP}} \leftarrow \|Ah_\psi(x) - h_{\text{old}}(x)\|_2$               {Eq. 11}
            $L = L + L_{\text{rep-ce}} + L_{\text{rep-logits}} + L_{\text{BFP}}$          {Eq. 12}
        end if
        $\theta \leftarrow$ sgd$(\theta, \nabla_\theta L)$
        $A \leftarrow$ sgdm$(A, \nabla_A L)$
        $M \leftarrow$ balanced-reservoir$(M, (x_o, y_o))$                {Alg. 2}
    end for
    $f_{\text{old}} =$ freeze$(f_\theta)$
end for
```

Algorithm 2 Balanced Reservoir Sampling [9]

```
Input: replay buffer $M$, exemplar $(x, y)$,
        number of seen examples $N$.
if $|M| > N$ then
    $M[N] \leftarrow (x, y)$
else
    $j \leftarrow \mathrm{RandInt}([0, N])$
    if $j < |M|$ then
        Reservoir Sampling
        $M[j] \leftarrow (x, y)$
        Balanced Reservoir Sampling
        $\tilde{y} \leftarrow \text{argmax } \mathrm{ClassCounts}(M, y)$
        $k \leftarrow \mathrm{RandChoice}(\{\tilde{k}; M[\tilde{k}] = (x, y), y = \tilde{y}\})$
        $M[k] \leftarrow (x, y)$
    end if
end if
```

perparameter search conducted on a held-out 10% training set for validation. Please refer to [7, 8] for further details.

The proposed BFP only introduced a single hyperparameter γ, which is set to a constant value of 1 throughout all experiments and does not need extra tuning. Other hyperparameters like α and β are inherited from ER and DER++ [8] and we simply adopt the same set of hyperparameters from [8]. We do not further tune or modify them.

B.4.1 Split CIFAR-10

FT: $lr = 0.1$
JT: $lr = 0.1$

Buffer size = 200

iCaRL: $lr = 0.1$, $wd = 10^{-5}$
FDR: $lr = 0.03$, $\alpha = 0.3$
LUCIR: $\lambda_{\text{base}} = 5$, $mom = 0.9$, $k = 2$, $\text{epoch}_{\text{fitting}} = 20$, $lr = 0.03$, $\text{lr}_{\text{fitting}} = 0.01$, $m = 0.5$
BiC: $\tau = 2$, $\text{epochs}_{\text{BiC}} = 250$, $lr = 0.03$
ER-ACE: $lr = 0.03$
ER: $lr = 0.1$
DER++: $lr = 0.03$, $\alpha = 0.1$, $\beta = 0.5$

Buffer size = 500

iCaRL: $lr = 0.1$, $wd = 10^{-5}$
FDR: $lr = 0.03$, $\alpha = 1$
LUCIR: $\lambda_{\text{base}} = 5$, $mom = 0.9$, $k = 2$, $\text{epoch}_{\text{fitting}} = 20$, $lr = 0.03$, $\text{lr}_{\text{fitting}} = 0.01$, $m = 0.5$
BiC: $\tau = 2$, $\text{epochs}_{\text{BiC}} = 250$, $lr = 0.03$
ER-ACE: $lr = 0.03$
ER: $lr = 0.1$
DER++: $lr = 0.03$, $\alpha = 0.2$, $\beta = 0.5$

B.4.2 Split CIFAR-100

FT: $lr = 0.03$
JT: $lr = 0.03$

Buffer size = 500

iCaRL: $lr = 0.3$, $wd = 10^{-5}$
FDR: $lr = 0.03$, $\alpha = 0.3$
LUCIR: $\lambda_{\text{base}} = 5$, $mom = 0.9$, $k = 2$, $\text{epoch}_{\text{fitting}} = 20$, $lr = 0.03$, $\text{lr}_{\text{fitting}} = 0.01$, $m = 0.5$
BiC: $\tau = 2$, $\text{epochs}_{\text{BiC}} = 250$, $lr = 0.03$
ER-ACE: $lr = 0.03$
ER: $lr = 0.1$
DER++: $lr = 0.03$, $\alpha = 0.2$, $\beta = 0.5$

Buffer size = 2000

iCaRL: $lr = 0.3$, $wd = 10^{-5}$
FDR: $lr = 0.03$, $\alpha = 1$
LUCIR: $\lambda_{\text{base}} = 5$, $mom = 0.9$, $k = 2$, $\text{epoch}_{\text{fitting}} = 20$, $lr = 0.03$, $\text{lr}_{\text{fitting}} = 0.01$, $m = 0.5$
BiC: $\tau = 2$, $\text{epochs}_{\text{BiC}} = 250$, $lr = 0.03$
ER-ACE: $lr = 0.03$
ER: $lr = 0.1$
DER++: $lr = 0.03$, $\alpha = 0.1$, $\beta = 0.5$

B.4.3 Split TinyImageNet

FT: $lr = 0.03$
JT: $lr = 0.03$

Buffer size = 4000

iCaRL: $lr = 0.03$, $wd = 10^{-5}$
FDR: $lr = 0.03$, $\alpha = 0.3$
LUCIR: λ_{base}=5, mom=0.9, k=2, $\text{epoch}_{\text{fitting}}$=20, lr=0.03, $\text{lr}_{\text{fitting}}$=0.01, m=0.5
BiC: $\tau = 2$, $\text{epochs}_{\text{BiC}} = 250$, $lr = 0.03$
ER-ACE: $lr = 0.03$
ER: $lr = 0.1$
DER++: $lr = 0.1$, $\alpha = 0.3$, $\beta = 0.8$

C. Additional results

C.1. Final Forgetting

Final Forgetting (FF) measures the performance drop between after learning on each task and the end of CL. A CL method with a lower FF has a better ability to retain knowledge during CL and thus better stability. However, higher stability may come with the price of plasticity, and we remind readers that the Final Average Accuracy (FAA) reported in Table 1 can better reflect the trade-off between stability and plasticity. The Final Forgetting for all baselines and our methods can be found in Table 4. As we can see from Table 4, in the class-IL setting, the proposed DER++ w/ BFP method helps reduce FF compared to the base DER++ method by 11% and 12% on S-CIFAR10 with

200 buffer size and S-CIFAR100 with 500 buffer size respectively. DER++ w/ BFP also achieves the lowest FF over all compared methods in the class-IL setting. Final Forgettings in the Task-IL setting are generally much lower than those from the Class-IL setting because Task-IL provides the oracle task identifiers during the testing time and thus becomes a much easier CL scenario. In this setting, the proposed BFP also brings large improvements over the base ER and DER++ methods.

Dataset	Buffer	FD	BFP	BFP-2
S-CIFAR10	200	55.10±1.85	**63.27±1.09**	60.61±2.72
	500	66.37±1.37	**71.51±1.58**	70.25±1.18
S-CIFAR100	500	20.02±0.09	**22.54±1.10**	21.25±0.73
	2000	36.81±0.71	38.92±1.94	**39.42±2.54**
S-TinyImg	4000	23.13±0.77	**26.33±0.68**	25.87±0.86

Table 3. Class-IL Final Average Accuracy using different types of layer for backward feature projection. The base method is ER.

C.2. Ablation Study based on Experience Replay

We conduct the same ablation study as that in Section 4.5, on different types of the projection layer used in ER w/ BFP. The results are reported in Table 3. From Table 3, we can draw the same conclusion as in Section 4.5. BFP uses learnable linear transformation when distilling features and thus results in better plasticity during CL compared to the simple FD method. Results show that BFP outperforms FD by a significant margin and has better performance than BFP-2 in most cases. This further shows that enforcing a linear relationship between the new and old features could better preserve linear separability and result in less forgetting in CL.

C.3. Linear Probing

We conduct the same linear probing analysis as Section 4.4 Figure 3 on Split-CIFAR100 and Split-TinyImageNet, and the results are reported in Figure 5. On these two datasets, while FD and BFP result in similar linear probing performance when based on DER++, BFP still leads to better linear probing accuracies when based on FT, especially when a large subset of training data is used for linear probing. FT w/ BFP (without the memory buffer) has a similar or even better performance than DER++ (with the memory buffer). This shows that BFP help learns a better feature space from CL, where features from different class are more linearly separable.

C.4. Feature Similarity Analysis

We perform the same feature similarity analysis as Section 4.6 and Figure 4 on Split-CIFAR100 and Split-TinyImageNet, and the results are reported in Figure 6. From Figure 6, although the curves have high variance throughout continual learning, we can see that BFP has feature similarities that are higher than the DER++ baseline but lower than the naive FD, and thus achieve a better trade-off between stability and plasticity.

C.5. Experiments on Split-ImageNet100

To demonstrate that the proposed BFP method scales to large datasets, we conduct experiments on ImageNet100 [23, 41]. We split ImageNet100 into 10 tasks with 10 classes per task and use a memory buffer of size 2000. The model is trained for 65 epochs on each task using an SGD optimizer with an initial learning rate of 0.1 and weight decay of 5×10^{-4}. Within each task, the learning rate goes through a linear warm-up scheduler for the first 5 epochs and then decays with a 0.1 rate after 15, 30, 40, and 45 epochs. The results are reported in Table 5, which shows that the proposed BFP method still gives a significant improvement (over 5% in Class-IL setting) over the DER++ baseline, confirming our existing results.

C.6. Effect of γ on Plasticity and Stability

In continual learning, the weight of regularization loss controls how closely and strictly the model should resemble the old checkpoints. Therefore the weight serves as a control knob on the trade-off between stability and plasticity: with a stronger regularization loss, the model forgets old tasks less but instead has a hard time learning new tasks.

Although we did not perform an extensive hyperparameter search on γ for individual combinations of datasets and buffer sizes, we are still interested in how the varying γ affects the trade-off between stability and plasticity in continual learning. Therefore, we train DER++ w/ BFP on S-CIFAR10 with different γ and report the performance in Table 6. Besides FAA and FF, we also report the Average Learning Accuracy (ALA) [42], which measures the learning ability on new tasks in continual learning and thus reflects the plasticity. Using the notation from Sec. 4.1, ALA is defined as

$$ALA = \frac{1}{T}\sum_{i=1}^{T} a_i^i. \tag{18}$$

From Table 6, we can see that the effect of γ aligns with our intuition. A higher γ poses a strong regularization on the feature space, resulting in a lower FF (more stable) but also a lower ALA (less plastic). Also, we can observe that the final performance (FAA) remains robust to the value of γ within a considerable range.

D. More Related Work

There has been some recent work that also employs PCA computation in continual learning. Note that the proposed BFP does not require PCA computation during training and the feature directions are learned implicitly when optimizing matrix A. However, to provide a complete understand-

Setting	Method	S-CIFAR10		S-CIFAR100		S-TinyImageNet
	Buffer Size	200	500	500	2000	4000
Class-IL	Joint Training	-		-		-
	Finetune	96.44±0.28		89.54±0.16		78.54±0.45
	iCaRL [41]	27.75±0.82	25.31±4.35	30.13±0.28	24.72±0.66	16.82±0.51
	FDR [5]	76.08±4.87	83.16±4.72	73.71±0.68	60.90±1.41	57.01±0.59
	LUCIR [23]	46.36±3.17	29.11±0.63	53.24±0.56	34.16±1.19	25.50±1.86
	BiC [50]	44.36±2.73	20.88±2.17	51.86±1.57	41.42±1.61	61.67±1.42
	ER-ACE [10]	21.59±1.19	15.07±0.99	38.32±1.29	28.69±0.87	30.83±0.23
	ER [42]	42.19±5.19	26.64±6.33	47.62±33.70	44.03±18.79	49.61±16.47
	ER w/ BFP (Ours)	32.23±5.58 (-9.96)	22.67±6.64 (-3.97)	47.69±30.30 (-0.07)	37.49±18.06 (-6.54)	41.59±20.77 (-8.02)
	DER++ [8]	28.28±1.06	20.16±1.49	42.58±2.03	26.29±1.66	16.03±1.20
	DER++ w/ BFP (Ours)	**16.69±0.28** (-11.59)	**13.25±0.64** (-6.91)	**29.85±0.97** (-12.73)	**20.91±0.86** (-5.39)	**9.42±1.04** (-6.28)
Task-IL	Joint Training	-		-		-
	Finetune	39.72±6.27		60.46±2.74		67.04±1.27
	iCaRL [41]	4.29±1.00	1.91±2.12	3.67±0.40	1.82±0.32	3.56±0.46
	FDR [5]	7.03±1.38	4.47±0.45	16.63±0.20	9.17±0.33	13.73±0.30
	LUCIR [23]	2.83±0.99	2.04±0.27	**2.61±0.17**	**1.08±0.13**	4.95±0.61
	BiC [50]	**0.81±0.77**	**0.24±0.25**	3.95±0.35	2.36±0.40	7.08±3.74
	ER-ACE [10]	6.10±0.72	3.64±0.29	13.95±0.45	7.36±0.43	10.67±0.41
	ER [42]	5.71±0.60	3.54±1.15	11.55±6.31	6.12±2.49	11.77±2.06
	ER w/ BFP (Ours)	1.38±0.29 (-4.34)	0.77±0.38 (-2.77)	5.63±1.56 (-5.92)	2.95±0.75 (-3.16)	**3.31±1.19** (-8.46)
	DER++ [8]	3.88±0.51	1.65±0.17	11.68±0.55	4.80±0.45	6.73±0.41
	DER++ w/ BFP (Ours)	1.04±0.23 (-2.84)	0.53±0.23 (-1.12)	6.36±0.43 (-5.32)	3.26±0.15 (-1.54)	4.17±0.37 (-2.49)

Table 4. Final Forgetting (FF, in %, lower is better) in Class-IL and Task-IL setting of baselines and our methods on various datasets and buffer sizes. The green numbers in parentheses show the absolute improvements over the corresponding ER or DER++ baselines brought by BFP.

Method	DER++	w/ FD	w/ BFP	w/ BFP-2
Class-IL	49.20±1.99	51.89±3.42	**54.45**±0.86	52.88±1.86
Task-IL	69.01±2.01	71.23±2.80	**72.05**±1.04	70.56±1.47

Table 5. Final Average Accuracy on ImageNet-100. (mean±std over 3 runs)

ing of the literature, we briefly review the related work that also uses PCA for continual learning.

Doan *et al*. [17] proposed PCA-OGD, which combines PCA analysis with Orthogonal Gradient Descent (OGD). PCA-OGD projects the gradients onto the residuals subspace to reduce the interference of gradient updates from the new tasks on the old tasks. Zhu *et al*. [56] decomposed the learned features during CL using PCA. They showed that feature directions with larger eigenvalues have larger similarities (corresponding angles) before and after learning a task. They proposed that these feature directions are more transferable and less forgettable. They showed that their dual augmentation method can encourage learned features to have more directions with large eigenvalues. GeoDL [46] constructs low-dimensional manifolds for the features extracted by the online model and the old checkpoints and performs knowledge distillation on the manifolds. PCA computation is explicitly conducted on the learned features for the manifold construction. SPACE [44] used PCA analysis for network compression and pruning in continual learning. Similar to our analysis, they use PCA to split the learned filters in a network into Core, which is important for the current task, and Residual, which can be compressed and freed up to learn future tasks. In their work, PCA computation is required during continual learning on every layer of the network in order to do pruning, This poses a significant computational overhead in CL compared to our BFP. Instead of applying PCA analysis in continual learning, Zhang *et al*. [54] designed a modified PCA algorithm based on EWC [30] so that it has continual learning ability. They aim to reduce the forgetting problem in monitoring multimode processes.

γ	0.1	0.3	1.0	3.0	10.0
FAA	74.56	75.77	76.68	76.00	73.54
FF	16.11	14.63	13.16	13.07	12.69
ALA	87.45	87.32	87.21	86.45	83.70

Table 6. Results on CIFAR10 (buffer size 500) with different γ.

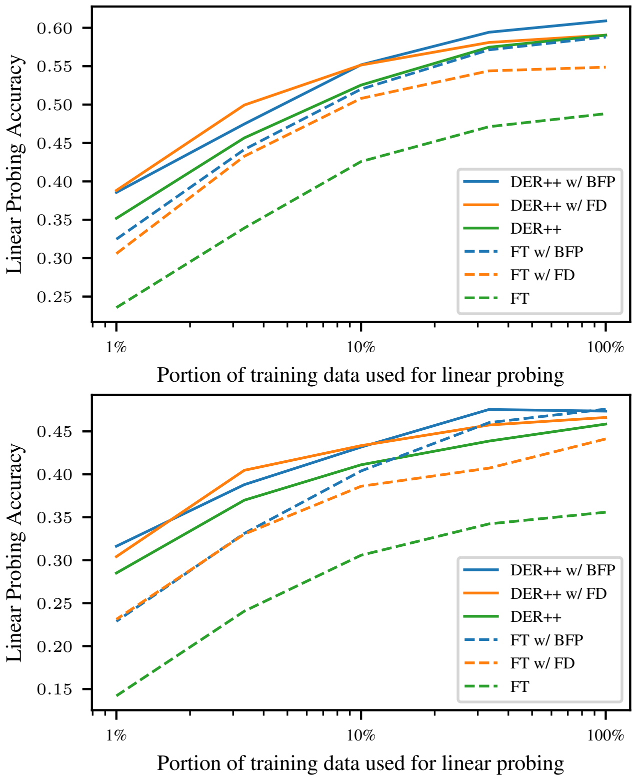

Figure 5. Linear probing accuracies on the fixed feature extractor obtained after training on Split-CIFAR100 (top) and TinyImageNet (bottom). DER++ and its variants use a buffer size of 500 for CIFAR100 and 4000 for TinyImageNet.

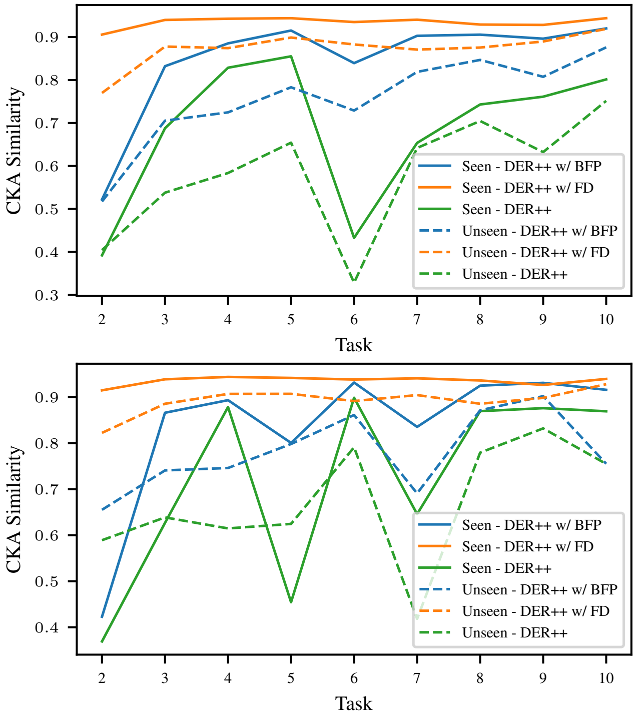

Figure 6. Feature similarity at different tasks of training on Split-CIFAR100 with buffer size 500 (top) and Split-TinyImageNet with buffer size 4000 (bottom), using different CL methods.

References

[1] Rahaf Aljundi, Francesca Babiloni, Mohamed Elhoseiny, Marcus Rohrbach, and Tinne Tuytelaars. Memory aware synapses: Learning what (not) to forget. In *ECCV*, 2018. 2

[2] Rahaf Aljundi, Eugene Belilovsky, Tinne Tuytelaars, Laurent Charlin, Massimo Caccia, Min Lin, and Lucas Page-Caccia. Online continual learning with maximal interfered retrieval. In *NeurIPS*, 2019. 2

[3] Elahe Arani, Fahad Sarfraz, and Bahram Zonooz. Learning fast, learning slow: A general continual learning method based on complementary learning system. In *ICLR*, 2022. 5

[4] Tommaso Barletti, Niccoló Biondi, Federico Pernici, Matteo Bruni, and Alberto Del Bimbo. Contrastive supervised distillation for continual representation learning. In *ICIAP*, pages 597–609. Springer, 2022. 1

[5] Ari S. Benjamin, David Rolnick, and Konrad P. Körding. Measuring and regularizing networks in function space. In *ICLR*, 2019. 6, 12

[6] Magdalena Biesialska, Katarzyna Biesialska, and Marta R Costa-Jussa. Continual lifelong learning in natural language processing: A survey. *arXiv preprint arXiv:2012.09823*, 2020. 1

[7] Matteo Boschini, Lorenzo Bonicelli, Pietro Buzzega, Angelo Porrello, and Simone Calderara. Class-incremental continual learning into the extended der-verse. *arXiv preprint arXiv:2201.00766*, 2022. 5, 6, 9, 10

[8] Pietro Buzzega, Matteo Boschini, Angelo Porrello, Davide Abati, and Simone Calderara. Dark experience for general continual learning: a strong, simple baseline. *NeurIPS*, 2020. 1, 2, 5, 6, 7, 9, 10, 12

[9] Pietro Buzzega, Matteo Boschini, Angelo Porrello, and Simone Calderara. Rethinking experience replay: a bag of tricks for continual learning. In *ICPR*, 2021. 5, 9, 10

[10] Lucas Caccia, Rahaf Aljundi, Nader Asadi, Tinne Tuytelaars, Joelle Pineau, and Eugene Belilovsky. New insights on reducing abrupt representation change in online continual learning. In *ICLR*, 2022. 3, 6, 12

[11] Hyuntak Cha, Jaeho Lee, and Jinwoo Shin. Co2l: Contrastive continual learning. In *ICCV*, 2021. 1

[12] Arslan Chaudhry, Puneet K Dokania, Thalaiyasingam Ajanthan, and Philip HS Torr. Riemannian walk for incremental learning: understanding forgetting and intransigence. In *ECCV*, 2018. 2

[13] Arslan Chaudhry, Marc'Aurelio Ranzato, Marcus Rohrbach, and Mohamed Elhoseiny. Efficient lifelong learning with a-gem. In *ICLR*, 2019. 2

[14] Arslan Chaudhry, Marcus Rohrbach, Mohamed Elhoseiny, Thalaiyasingam Ajanthan, Puneet K Dokania, Philip HS Torr, and Marc'Aurelio Ranzato. Continual learning with tiny episodic memories. In *ICML*, 2019. 5

[15] MohammadReza Davari, Nader Asadi, Sudhir Mudur, Rahaf Aljundi, and Eugene Belilovsky. Probing representation forgetting in supervised and unsupervised continual learning. In *CVPR*, 2022. 7, 8

[16] Prithviraj Dhar, Rajat Vikram Singh, Kuan-Chuan Peng, Ziyan Wu, and Rama Chellappa. Learning without memorizing. In *CVPR*, 2019. 1, 2

[17] Thang Doan, Mehdi Abbana Bennani, Bogdan Mazoure, Guillaume Rabusseau, and Pierre Alquier. A theoretical analysis of catastrophic forgetting through the ntk overlap matrix. In *AISTATS*, pages 1072–1080. PMLR, 2021. 12

[18] Arthur Douillard, Matthieu Cord, Charles Ollion, Thomas Robert, and Eduardo Valle. Podnet: Pooled outputs distillation for small-tasks incremental learning. In *ECCV*, pages 86–102. Springer, 2020. 1, 2

[19] Laura N Driscoll, Lea Duncker, and Christopher D Harvey. Representational drift: Emerging theories for continual learning and experimental future directions. *Current Opinion in Neurobiology*, 76:102609, 2022. 3

[20] Enrico Fini, Victor G Turrisi da Costa, Xavier Alameda-Pineda, Elisa Ricci, Karteek Alahari, and Julien Mairal. Self-supervised models are continual learners. In *CVPR*, 2022. 2, 4, 7

[21] Alex Gomez-Villa, Bartlomiej Twardowski, Lu Yu, Andrew D Bagdanov, and Joost van de Weijer. Continually learning self-supervised representations with projected functional regularization. In *CVPR Workshop*, 2022. 2, 4

[22] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 5

[23] Saihui Hou, Xinyu Pan, Chen Change Loy, Zilei Wang, and Dahua Lin. Learning a unified classifier incrementally via rebalancing. In *ICCV*, 2019. 6, 11, 12

[24] Joel Jang, Seonghyeon Ye, Sohee Yang, Joongbo Shin, Janghoon Han, Gyeonghun Kim, Stanley Jungkyu Choi, and Minjoon Seo. Towards continual knowledge learning of language models. *arXiv preprint arXiv:2110.03215*, 2021. 1

[25] Heechul Jung, Jeongwoo Ju, Minju Jung, and Junmo Kim. Less-forgetting learning in deep neural networks. *arXiv preprint arXiv:1607.00122*, 2016. 2

[26] Minsoo Kang, Jaeyoo Park, and Bohyung Han. Class-incremental learning by knowledge distillation with adaptive feature consolidation. In *CVPR*, 2022. 1, 2

[27] James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, et al. Overcoming catastrophic forgetting in neural networks. *Proceedings of the national academy of sciences*, 114(13):3521–3526, 2017. 2

[28] Simon Kornblith, Mohammad Norouzi, Honglak Lee, and Geoffrey Hinton. Similarity of neural network representations revisited. In *ICML*, pages 3519–3529. PMLR, 2019. 8

[29] Alex Krizhevsky. Learning multiple layers of features from tiny images. Technical report, Citeseer, 2009. 5

[30] Sang-Woo Lee, Jin-Hwa Kim, Jaehyun Jun, Jung-Woo Ha, and Byoung-Tak Zhang. Overcoming catastrophic forgetting by incremental moment matching. In *NeurIPS*, 2017. 1, 12

[31] Timothée Lesort, Vincenzo Lomonaco, Andrei Stoian, Davide Maltoni, David Filliat, and Natalia Díaz-Rodríguez. Continual learning for robotics: Definition, framework,

learning strategies, opportunities and challenges. *Information Fusion*, 2020. 1

[32] Zhizhong Li and Derek Hoiem. Learning without forgetting. *PAMI*, 2017. 1, 2

[33] Xialei Liu, Chenshen Wu, Mikel Menta, Luis Herranz, Bogdan Raducanu, Andrew D Bagdanov, Shangling Jui, and Joost van de Weijer. Generative feature replay for class-incremental learning. In *CVPR Workshops*, 2020. 2

[34] David Lopez-Paz and Marc'Aurelio Ranzato. Gradient episodic memory for continual learning. In *NeurIPS*, 2017. 1, 2

[35] Arun Mallya and Svetlana Lazebnik. Packnet: Adding multiple tasks to a single network by iterative pruning. In *CVPR*, 2018. 1

[36] Marc Masana, Xialei Liu, Bartlomiej Twardowski, Mikel Menta, Andrew D Bagdanov, and Joost van de Weijer. Class-incremental learning: survey and performance evaluation on image classification. *arXiv preprint arXiv:2010.15277*, 2020. 5

[37] Marc Masana, Xialei Liu, Bartłomiej Twardowski, Mikel Menta, Andrew D Bagdanov, and Joost van de Weijer. Class-incremental learning: survey and performance evaluation on image classification. *PAMI*, 2022. 1

[38] Michael McCloskey and Neal J Cohen. Catastrophic interference in connectionist networks: The sequential learning problem. In *Psychology of learning and motivation*. 1989. 1

[39] Karl Pearson. Liii. on lines and planes of closest fit to systems of points in space. *The London, Edinburgh, and Dublin philosophical magazine and journal of science*, 2(11):559–572, 1901. 3

[40] Vinay V Ramasesh, Ethan Dyer, and Maithra Raghu. Anatomy of catastrophic forgetting: Hidden representations and task semantics. *arXiv preprint arXiv:2007.07400*, 2020. 8

[41] Sylvestre-Alvise Rebuffi, Alexander Kolesnikov, Georg Sperl, and Christoph H Lampert. icarl: Incremental classifier and representation learning. In *CVPR*, 2017. 2, 6, 11, 12

[42] Matthew Riemer, Ignacio Cases, Robert Ajemian, Miao Liu, Irina Rish, Yuhai Tu, and Gerald Tesauro. Learning to learn without forgetting by maximizing transfer and minimizing interference. In *ICLR*, 2019. 6, 11, 12

[43] Andrei A Rusu, Neil C Rabinowitz, Guillaume Desjardins, Hubert Soyer, James Kirkpatrick, Koray Kavukcuoglu, Razvan Pascanu, and Raia Hadsell. Progressive neural networks. *arXiv*, 2016. 1

[44] Gobinda Saha, Isha Garg, Aayush Ankit, and Kaushik Roy. Space: Structured compression and sharing of representational space for continual learning. *IEEE Access*, 9:150480–150494, 2021. 12

[45] Hanul Shin, Jung Kwon Lee, Jaehong Kim, and Jiwon Kim. Continual learning with deep generative replay. In *NeurIPS*, 2017. 2

[46] Christian Simon, Piotr Koniusz, and Mehrtash Harandi. On learning the geodesic path for incremental learning. In *CVPR*, pages 1591–1600, 2021. 12

[47] Stanford. Tiny imagenet challenge, cs231n course., CS231N. 5

[48] Gido M Van de Ven and Andreas S Tolias. Generative replay with feedback connections as a general strategy for continual learning. *arXiv preprint arXiv:1809.10635*, 2018. 2

[49] Jianren Wang, Xin Wang, Yue Shang-Guan, and Abhinav Gupta. Wanderlust: Online continual object detection in the real world. In *ICCV*, pages 10829–10838, 2021. 1

[50] Yue Wu, Yinpeng Chen, Lijuan Wang, Yuancheng Ye, Zicheng Liu, Yandong Guo, and Yun Fu. Large scale incremental learning. In *CVPR*, 2019. 5, 6, 12

[51] Shipeng Yan, Jiangwei Xie, and Xuming He. Der: Dynamically expandable representation for class incremental learning. In *CVPR*, pages 3014–3023, 2021. 1

[52] Lu Yu, Bartlomiej Twardowski, Xialei Liu, Luis Herranz, Kai Wang, Yongmei Cheng, Shangling Jui, and Joost van de Weijer. Semantic drift compensation for class-incremental learning. In *CVPR*, 2020. 3

[53] Friedemann Zenke, Ben Poole, and Surya Ganguli. Continual learning through synaptic intelligence. In *ICML*, 2017. 1, 2

[54] Jingxin Zhang, Donghua Zhou, and Maoyin Chen. Monitoring multimode processes: A modified pca algorithm with continual learning ability. *Journal of Process Control*, 103:76–86, 2021. 12

[55] Xiao Zhang, Dejing Dou, and Ji Wu. Feature forgetting in continual representation learning. *arXiv preprint arXiv:2205.13359*, 2022. 7

[56] Fei Zhu, Zhen Cheng, Xu-Yao Zhang, and Cheng-lin Liu. Class-incremental learning via dual augmentation. *NeurIPS*, 34:14306–14318, 2021. 12