Self-Sustaining Representation Expansion for Non-Exemplar Class-Incremental Learning (Supplementary Materials)

Kai Zhu[1] Wei Zhai[1] Yang Cao[1,3,†] Jiebo Luo[2] Zheng-Jun Zha[1]
[1] University of Science and Technology of China [2] University of Rochester
[3] Institute of Artificial Intelligence, Hefei Comprehensive National Science Center
{zkzy@mail., wzhai056@mail., forrest@}ustc.edu.cn jluo@cs.rochester.edu zhazj@ustc.edu.cn

A. Additional Explanation

A.1. The Details of Evaluation Metrics

Following [1, 14], we report average incremental accuracy and average forgetting. Average incremental accuracy A_N is computed as the average accuracy of all the incremental phases a_k (including the first phase), which compares the overall incremental performance of different methods fairly,

$$A_N = \frac{1}{N+1}\sum_{k=0}^{N} a_k. \tag{1}$$

Average forgetting is computed as the average forgetting of different tasks throughout the incremental process, which directly measures the ability of different methods to resist catastrophic forgetting. The forgetting at phase k (k > 0) is calculated as $F_k = \frac{1}{k}\sum_{j=0}^{k-1} f_j^k$, where f_j^k denotes the performance drop of classes first learned in phase j after the model has been incrementally trained up to phase $k > j$ as:

$$f_j^k = \underset{i\in\{j,\cdots k-1\}}{max} a_{i,j} - a_{k,j}, \tag{2}$$

where $a_{i,j}$ represents the accuracy of classes first in phase j after the model has been incrementally trained up to phase i.

A.2. Related Work on Distillation

Knowledge distillation [4] was first applied to the field of model compression, where complex models (teacher) are utilized to provide soft labels for the training of lightweight models (student), thus facilitating deployment. Large models are over-parameterized and tend to have better generalization. The key problem is how to efficiently transfer the knowledge contained therein to small models. Various solutions have been proposed to enhance its effectiveness, such as response-based distillation [3], feature-based distillation [8] and relation-based distillation [6]. At the same time, because of its similarity to human learning style, it is gradually applied to mutual learning [13], continual learning and other fields. In class-incremental learning, various response-based [7], feature-based [5] and even attention-based distillation [2] techniques are used to pass discriminative features of the old network to the newly learned network, which has become an indispensable part. Unlike the common exemplar-based class-incremental learning methods, we consider the calibration of distillation techniques in the absence of old samples.

A.3. Detailed Values of the Curves

To facilitate the fair comparison of subsequent work, we report the detailed values (*i.e.* Figure 7 in the main text) of all the accuracy curves in Table 1, 2 and 3.

Dataset	Phase																				
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
CIFAR-100	82.12	76.21	71.91	70.59	67.02	67.45	64.47	65.03	62.31	60.30	60.40	58.99	57.50	56.18	55.15	54.27	53.73	53.85	53.45	52.86	51.92
TinyImageNet	63.32	56.67	54.53	52.85	51.45	50.24	49.42	48.93	48.16	47.79	47.77	46.89	46.23	45.56	45.55	44.43	43.63	43.25	42.23	41.60	41.03

Table 1. Detailed values of classification accuracy under the setting of 20 phases.

Dataset	Phase										
	0	1	2	3	4	5	6	7	8	9	10
CIFAR-100	80.00	74.09	70.53	67.58	65.63	63.52	61.41	59.62	58.33	58.16	56.57
TinyImageNet	63.32	55.93	52.53	50.15	48.86	48.03	46.44	45.45	43.77	42.59	41.18
ImageNet-Subset	82.80	73.38	71.40	69.54	68.00	67.23	65.48	64.47	61.82	61.16	59.32

Table 2. Detailed values of classification accuracy under the setting of 10 phases.

Dataset	Phase					
	0	1	2	3	4	5
CIFAR-100	80.00	71.30	66.03	61.90	59.00	56.97
TinyImageNet	63.32	55.43	50.37	47.38	44.41	41.45

Table 3. Detailed values of classification accuracy under the setting of 5 phases.

A.4. Standard Deviation of the Average Incremental Accuracy.

All results of the average incremental accuracy (*i.e.* Table 3 in the main text) are evaluated on three different runs. To show the stability of our method, we report its standard deviation on three runs. As shown in Table 4, random factors have little impact on our scheme.

	Methods	**CIFAR-100**			**TinyImageNet**			**ImageNet-Subset**
		P=5	*P=10*	*P=20*	*P=5*	*P=10*	*P=20*	*P=10*
N=0	LwF_MC [7]	45.93	27.43	20.07	29.12	23.10	17.43	31.18
	MUC [12]	49.42	30.19	21.27	32.58	26.61	21.95	35.07
	PASS [14]	63.47	61.84	58.09	49.55	47.29	42.07	61.80
	Ours	**65.88**±0.04	**65.04**±0.39	**61.70**±0.04	**50.39**±0.25	**48.93**±0.03	**48.17**±0.04	**67.69**±0.02

Table 4. Comparisons of the average incremental accuracy (%) with other methods on CIFAR-100, TinyImageNet, and ImageNet-Subset. The red footnotes in the last row represent the standard deviation on three different runs.

A.5. More Visualization.

To better demonstrate the role of DSR and MBD during optimization, we show more visualization results with t-SNE. As shown in Fig. 1 (a), although the old classes have slightly changed in the representation after multi-phase optimization, their distances in the embedding space almost do not decline with our DSR. As shown in Fig. 1 (b), newly incremental classes are easily closed to some of the old classes. Owing to our MBD, the novel features are promoted to differentiate from the old class, thus improving the seperation of novel clusters.

To further evaluate performance of both old and new classes during training, we compare their accuracy on more setting (*i.e.* 5 and 20 phases). As shown in Fig. 2, our method achieves similar performance between the old and new classes without favoring one side due to overfitting, which is a prerequisite for a good incremental learning system.

To verify the role of the PSM, we conduct more data statistics on the incremental samples of 5 phases and 10 phases setting. As shown in Fig. 3 and 4, the new classes have a large difference in similarity. And the intra-class fluctuations are also large, so different classes and samples involved in the optimization process will bring different changes. Therefore it is important to reasonably place them in the two potentially conflicting processes of old feature distillation and new feature learning. At the same time, the values are also different at different phases and settings, so it is also necessary to adapt this changes in different situations, which will serve as a direction for our future research.

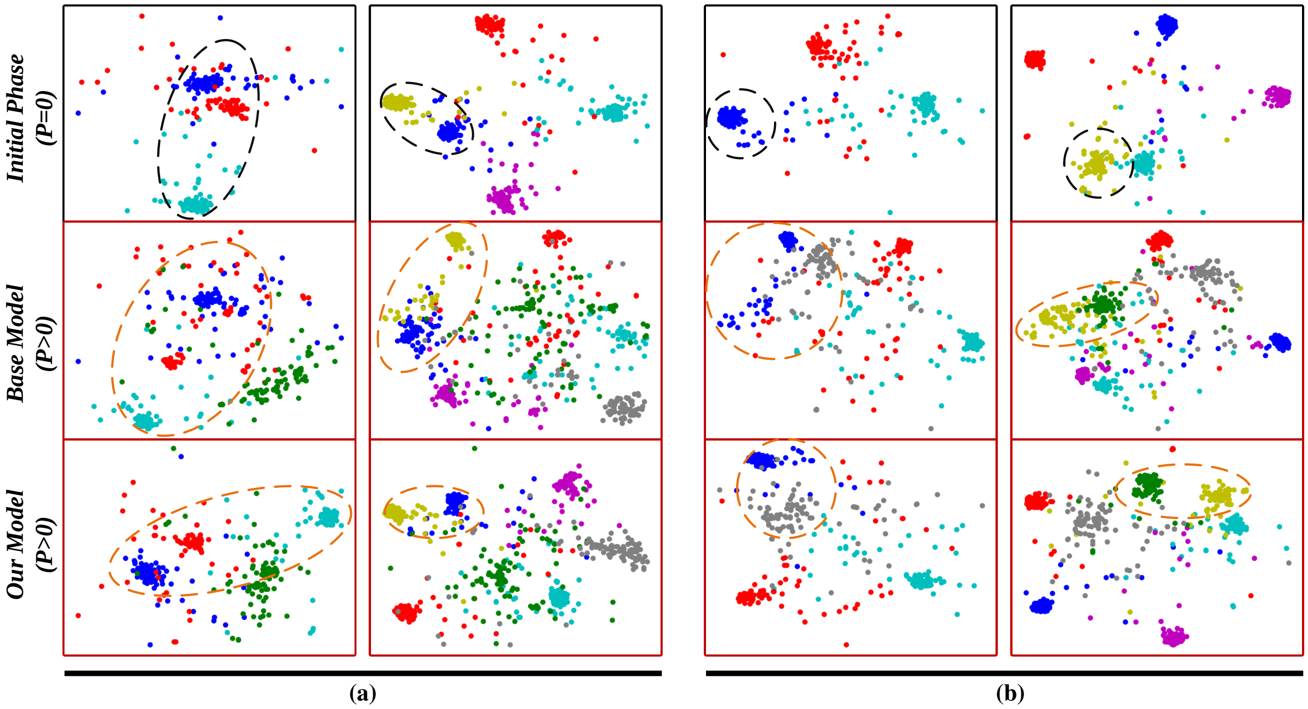
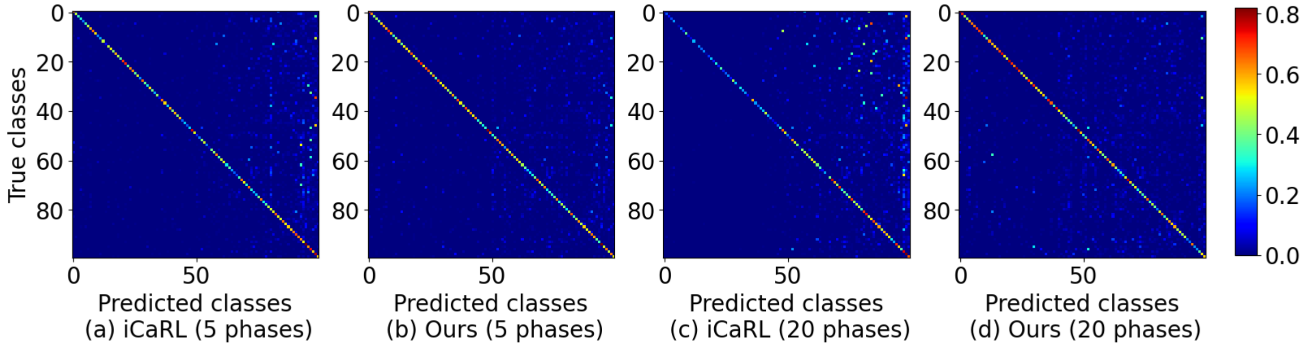


Figure 1. More visualization results on the representation. (a) DSR maintains the discriminative features and inter-relations of old classes, thus enhancing the clustering and separation of the distribution of old classes. (b) MBD results in a better distinction between similar classes.

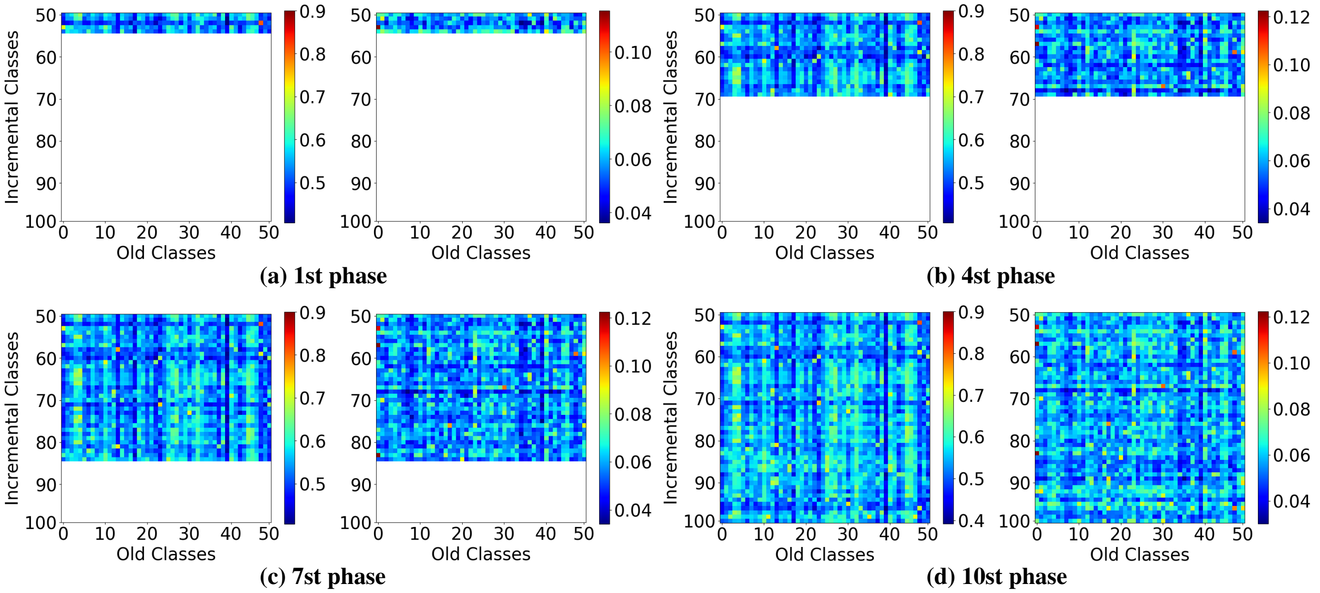


(a) iCaRL (5 phases)
(b) Ours (5 phases)
(c) iCaRL (20 phases)
(d) Ours (20 phases)

Figure 2. More confusion matrices of different methods on CIFAR-100. 5 phases and 20 phases settings are considered to further evaluate the stability of our method on the old and novel classes.

A.6. Limitations.

While promising, we admit our work also has some limitations as follows. First, the focus of this paper is the exploration of the self-sustaining representation expansion scheme, which is implemented with some simple and effective techniques, such as residual adapter and structural reparameterization. More knowledge on the dynamic architectures can be taken into account to further improve the retention and refinement of the old features. Second, although our method also works well



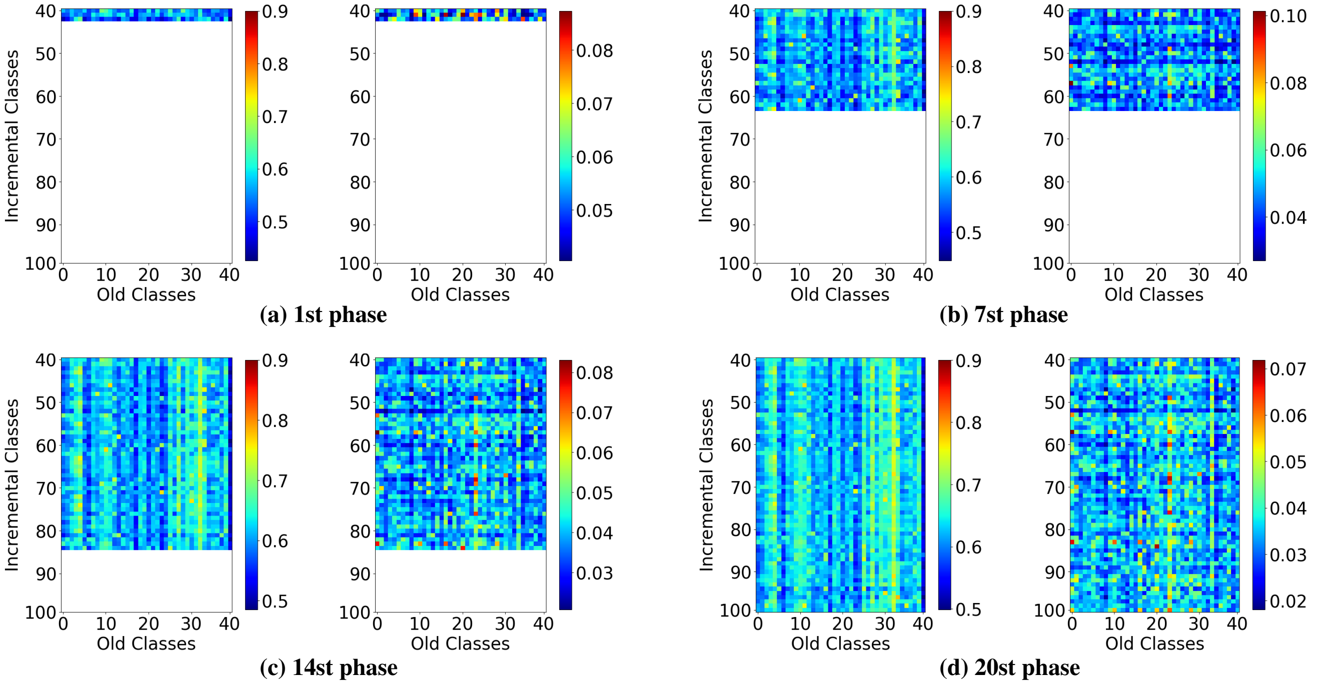
(a) 1st phase

(b) 4st phase

(c) 7st phase

(d) 10st phase

Figure 3. More statistics of similarity on the setting of 10 phases.



(a) 1st phase

(b) 7st phase

(c) 14st phase

(d) 20st phase

Figure 4. More statistics of similarity on the setting of 20 phases.

in the CIL setting (*i.e.* Section B.3), it lacks exploration of the case where the number of exemplars changes. We believe attempting to address above limitations would help to better explore the direction of NECIL and to adapt our scheme to more interesting settings, such as CIL.

B. Additional Results

B.1. Results of Different NECIL Settings and Methods

To prove the superiority of our proposed method, we conduct more comparative experiments under different settings on CIFAR-100. [9] proposes a date-free CIL setting, which is similar to the setting of NECIL [14]. To the best of our knowledge, this is the most relevant work up to the time of submission, which has been accepted by ICCV2021. Despite the similarities in the task, the experimental setup is completely different. B0 setting is adopted in [9], where all 100 classes are divided into 5, 10, and 20 phases equally. B50 setting is adopted in [14], where the initial model is trained on 50 classes, and the remaining 50 classes are divided into 5, 10, and 20 phases. Thanks to their release of the code, we compare our method with [9] in both B0 and B50 settings. As shown in Table 5 and 6, our method achieves average improvement of 2 and 3 points on B0 and B50 settings, respectively. At the same time, as [9] modify the DeepInversion [11] technique to synthesize samples for old classes at each incremental phase, the training time (Time) of our method is much shorter. It can be seen that our method is reliable in terms of effectiveness and efficiency. At the same time, to compare the two methods at all the phases in the incremental process, we plot the detailed curves in Fig. 5. It can be seen that our method is superior in almost at all the settings and phases, especially in the long-term setting.

	CIFAR-100 (B0)					
Method	5 phases		10 phases		20 phases	
	Acc(%)	Time(s)	Acc(%)	Time(s)	Acc(%)	Time(s)
ABD [11]	43.90	1270	33.70	695	20.00	625
Ours	44.60	756	34.39	467	23.12	209

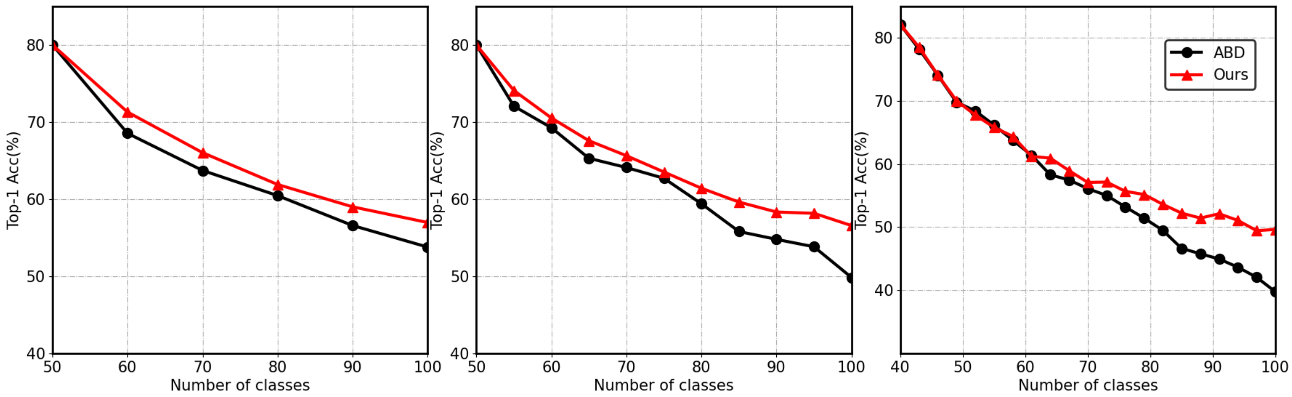
Table 5. Comparisons of the final performance (%) under different settings (*i.e.* 5, 10 and 20 phases). 'Acc' represents the average incremental accuracy and 'Time' represents the training time during each phase. B0 represents the number of base classes is zero, where all 100 classes are divided into 5, 10, and 20 phases equally. For comparison, we conduct all the experiments in the same setup and hardware environment.

	CIFAR-100 (B50)		
Method	5 phases	10 phases	20 phases
ABD [11]	63.85	62.46	57.40
Ours	65.88	65.04	61.70

Table 6. Comparisons of the final performance (%) under different settings. B50 represents the number of base classes is 50, where the initial model is trained on 50 classes, and the remaining 50 classes are divided into 5, 10, and 20 phases. B50 setting is consistent with our experimental setup in the main text.

B.2 Further Analysis of the DSR Strategy

To further explore the impact of DSR strategy on the expandable representation during training, we design more experiments. First, we remove the structural reparameterization part for comparison, where the main branch is not updated in the whole incremental process. As shown in Table 7, the accuracy of model without structural reparameterization (*i.e.* w/o R) is nearly one point lower. It demonstrates the necessity of updating of the base model, especially in the long-term incremental setting. Second, we replace the residual adapter with the non-exemplar dynamic expandable network (*i.e.* NDER), which is similar with [10]. As shown in the top two rows of Table 7, due to the lack of old samples, it is difficult to perform effective optimization with such large expanding parameters. Because of the feature inconsistency between the old and new model in the channel dimension, feature distillation can not be added to maintain the features of old classes (*i.e.* w/o D). For this purpose we add the response-based distillation to align the output of the old and new models. It can be seen in the table that the result has improved but there is still a big gap with other methods. It can be seen that our DSR strategy has the potential to help CIL methods get rid of the dependence on exemplars. Combined with methods such as regularization-based or structure-based strategies could promote our approach to be better. We will analyze and upgrade our DSR strategy in the future in more complex settings.



(a) 5 phases on CIFAR-100 **(b) 10 phases on CIFAR-100** **(c) 20 phases on CIFAR-100**

Figure 5. Classification accuracy on CIFAR-100 (B50), which contains the complete curves of the latest method from different settings (*i.e.* 5, 10 and 20 phases). The red curve represents our method, and the black one represents ABD [11]. All the experiments for different methods are performed in the same hardware and software environment.

	CIFAR-100		
Method	5 phases	10 phases	20 phases
NDER(w/o D)	25.58	16.70	10.65
NDER	29.08	21.13	13.10
Ours(w/o R)	65.11	64.21	60.87
Ours	65.87	65.12	61.60

Table 7. Further analysis of the DSR strategy. 'w/o D' represents the meaning of without distillation and 'w/o R' represents the meaning of without reparameterization.

B.3 Generalization to the CIL

To prove the effectiveness and extensibility of our method, we introduce it into the CIL setting. As [2] is one of the SOTA methods in CIL setting, we modify its implementation with our DSR strategy directly. As shown in Table 8, our method achieves average improvement of 2 points in the normal setting. Even if distillation is removed, the reslut of our method is still comparative with [2]. It can be seen that our method has great potential for CIL settings, which will serve as our future work.

	CIFAR-100 (B50)	
Method	5 phases	10 phases
Podnet	64.88	63.05
Ours(w/o D)	63.88	61.62
Ours	66.65	64.99

Table 8. Comparisons of the average incremental accuracy (%) under the CIL setting.

References

[1] Arslan Chaudhry, Puneet K Dokania, Thalaiyasingam Ajanthan, and Philip HS Torr. Riemannian walk for incremental learning: Understanding forgetting and intransigence. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 532–547, 2018.

[2] Arthur Douillard, Matthieu Cord, Charles Ollion, Thomas Robert, and Eduardo Valle. Podnet: Pooled outputs distillation for small-tasks incremental learning. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XX 16*, pages 86–102. Springer, 2020.

[3] Tommaso Furlanello, Zachary Lipton, Michael Tschannen, Laurent Itti, and Anima Anandkumar. Born again neural networks. In *International Conference on Machine Learning*, pages 1607–1616. PMLR, 2018.

[4] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015.

[5] Saihui Hou, Xinyu Pan, Chen Change Loy, Zilei Wang, and Dahua Lin. Learning a unified classifier incrementally via rebalancing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 831–839, 2019.

[6] Wonpyo Park, Dongju Kim, Yan Lu, and Minsu Cho. Relational knowledge distillation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3967–3976, 2019.

[7] Sylvestre-Alvise Rebuffi, Alexander Kolesnikov, Georg Sperl, and Christoph H Lampert. icarl: Incremental classifier and representation learning. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 2001–2010, 2017.

[8] Adriana Romero, Nicolas Ballas, Samira Ebrahimi Kahou, Antoine Chassang, Carlo Gatta, and Yoshua Bengio. Fitnets: Hints for thin deep nets. *arXiv preprint arXiv:1412.6550*, 2014.

[9] James Smith, Yen-Chang Hsu, Jonathan Balloch, Yilin Shen, Hongxia Jin, and Zsolt Kira. Always be dreaming: A new approach for data-free class-incremental learning. *arXiv preprint arXiv:2106.09701*, 2021.

[10] Shipeng Yan, Jiangwei Xie, and Xuming He. Der: Dynamically expandable representation for class incremental learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3014–3023, 2021.

[11] Hongxu Yin, Pavlo Molchanov, Jose M Alvarez, Zhizhong Li, Arun Mallya, Derek Hoiem, Niraj K Jha, and Jan Kautz. Dreaming to distill: Data-free knowledge transfer via deepinversion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8715–8724, 2020.

[12] L. Yu, S. Parisot, G. Slabaugh, J. Xu, and T. Tuytelaars. More classifiers, less forgetting: A generic multi-classifier paradigm for incremental learning. *European Conference on Computer Vision*, 2020.

[13] Ying Zhang, Tao Xiang, Timothy M Hospedales, and Huchuan Lu. Deep mutual learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4320–4328, 2018.

[14] Fei Zhu, Xu-Yao Zhang, Chuang Wang, Fei Yin, and Cheng-Lin Liu. Prototype augmentation and self-supervision for incremental learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5871–5880, 2021.