Supplementary Material for "DER: Dynamically Expandable Representation for Class Incremental Learning"

1. Hyperparameters

Representation learning stage For CIFAR-100, we use SGD to train the model with batch size 128, weight decay 0.0005. We adopt the warmup strategy with the ending learning rate 0.1 for 10 epochs. After the warmup, we run the SGD with 160 epochs and the learning rate decays at 100, 120 epochs with 0.1. For ImageNet100 and ImageNet1000, we adopt SGD with batch size 256, weight decay 0.0005. We also use the same warmup strategy as in CIFAR100. After the warmup, model is trained for 120 epochs. Learning rate starts from 0.1 and decays by 0.1 rates after 30, 60, 80, and 90 epochs.

For the coefficients in the loss function, λ_a is set to 1 for all experiments in the paper. λ_s is tuned to ensure our learned model has comparable number of parameters to other methods for fair comparison. Moreover, for simplicity, we set the same λ_s for every steps in each experiment. For experiments of CIFAR100-B0, λ_s is set as 0.75. For experiments of CIFAR100-B50, λ_s is set as 0.25. For experiments on ImageNet100 and ImageNet, λ_s is set as 0.75 for ImageNet100-B0, 0.5 for ImageNet100-B50 and 0.75 for ImageNet.

Classifier learning stage We adopt SGD optimizer with weight decay 0.0005 to update the classifier only for 30 epochs with SGD optimizer. Learning rate is 0.1 and decays with 0.1 rate at 15 epochs. The temperature of cross-entropy loss are set as $\delta = 5$ for CIFAR-100 and $\delta = 1$ for ImageNet-100 and ImageNet-1000.

2. Sensitive Study of Hyper-parameters

We conduct a sensitive study of our method on CVIFAR100-B0 10 steps with different λ_a. The results are shown in Table 1, which demonstrates our method is robust to λ_a. We also conduct experiments for different λ_s, which is shown in the Figure 1 in the main body.

λ_a	0.1	0.5	1	5	10
Avg	$74.12_{\pm 0.06}$	$74.41_{\pm 0.16}$	$74.64_{\pm 0.28}$	$74.52_{\pm 0.33}$	$73.54_{\pm 0.22}$

Table 1: Sensitive study on effects of λ_a

3. The Quality of Decision Boundary

In this section, our goal is to verify that the high-quality linear classifier can be obtained even re-learning the old classes' decision boundary with memory $\mathcal{M}$. Specifically, we compare our method with an 'ideal' strategy that uses all the previous data to train the classifier in the second stage. Such an upper bound achieves $76.14 \pm 0.80\%$ on the CIFAR100-B0 10 steps, which is only slightly higher than our method ($74.64 \pm 0.28\%$). We also observed similar results on the other benchmarks, which show the efficacy of our second-stage learning.

4. Latency

Regarding inference latency, we conduct an experimental comparison on the ImageNet with GTX 1080Ti. Our method achieves 1.07ms/image, which is comparable to other baseline methods, such as BiC and WA, which are 0.99ms/image.

5. Results for modified 32-layer ResNet

Most works use a modified 32-layer ResNet following iCaRL[3]. We also compare the results of our method with other methods based on the modified 32-layer ResNet. The results on CIFAR100-B0 are shown in Table 2 and the results on CIFAR100-B50 are shown in Table 3. The results of other methods are reported in their papers. It can be found that our method still outperforms other methods on both CIFAR100-B0 and CIFAR100-B50 even with a small network like modified ResNet-32.

6. More detailed results on CIFAR100

Figure 1 shows the performance with respect to steps on CIFAR100-B0 with 5 incremental steps and 10 incremental

steps and CIFAR100-B0 with 2 incremental steps. This also illustrates the superiority of our method.

7. Detailed results on ImageNet

We also show the curves of performance with respect to steps on ImageNet100-B0, ImageNet100-B50 and ImageNet1000-B0 in Figure 2, which proves the effectiveness of our method on complex datasets.

References

[1] Arthur Douillard, Matthieu Cord, Charles Ollion, Thomas Robert, and Eduardo Valle. Podnet: Pooled outputs distillation for small-tasks incremental learning. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2020. 3

[2] Saihui Hou, Xinyu Pan, Chen Change Loy, Zilei Wang, and Dahua Lin. Learning a unified classifier incrementally via rebalancing. In *Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR)*, 2019. 3

[3] 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 (CVPR)*, 2017. 1, 3

[4] Tao Xiaoyu, Chang Xinyuan, Hong Xiaopeng, Wei Xing, and Gong Yihong. Topology-preserving class-incremental learning. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2020. 3

[5] Bowen Zhao, Xi Xiao, Guojun Gan, Bin Zhang, and Shu-Tao Xia. Maintaining discrimination and fairness in class incremental learning. In *Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR)*, 2020. 3

Methods	5 steps		10 steps		20 steps		50 steps	
	#Paras	Avg	#Paras	Avg	#Paras	Avg	#Paras	Avg
iCaRL [3]	0.46	67.20	0.46	64.04	0.46	61.16	0.46	57.00
BiC [2]	0.46	68.92	0.46	66.15	0.46	63.80	0.46	-
WA [5]	0.46	70.00	0.46	67.25	0.46	64.33	0.46	-
Ours(w/o P)	1.38	**73.00**(+3.00)	2.53	**71.29**(+4.04)	4.83	**71.07**(+6.74)	11.73	**70.58**(+13.58)
Ours	0.42	**72.31**(+2.31)	0.52	**69.41**(+2.16)	0.45	**68.82**(+4.49)	0.70	**67.29**(+10.29)

Table 2: Results on CIFAR100-B0 benchmark using modified 32-layer ResNet. *#Paras* means the average number of parameters used during inference over steps, which is counted by million. *Avg* means the average accuracy (%) over steps. *Ours(w/o P)* means our method without pruning.

Methods	2Steps		5Steps		10Steps	
	#Paras	Avg	#Paras	Avg	#Paras	Avg
UCIR [2]	0.46	66.76	0.46	63.42	0.46	60.18
PODNet [1]	-	-	0.46	64.83	0.46	64.03
TPCIL [4]	-	-	0.46	65.34	0.46	63.58
Ours(w/o P)	0.92	**70.18**(+3.42)	1.61	**68.52**(+3.18)	2.76	**67.09**(+3.06)
Ours	0.32	**69.52**(+2.76)	0.59	**67.60**(+2.26)	0.61	**66.36**(+2.33)

Table 3: Results on CIFAR100-B50 using modified 32-layer ResNet. *#Paras* means the average number of parameters used during inference over steps, which is counted by million. *Avg* means the average accuracy (%) over steps. *Ours(w/o P)* means our method without pruning.

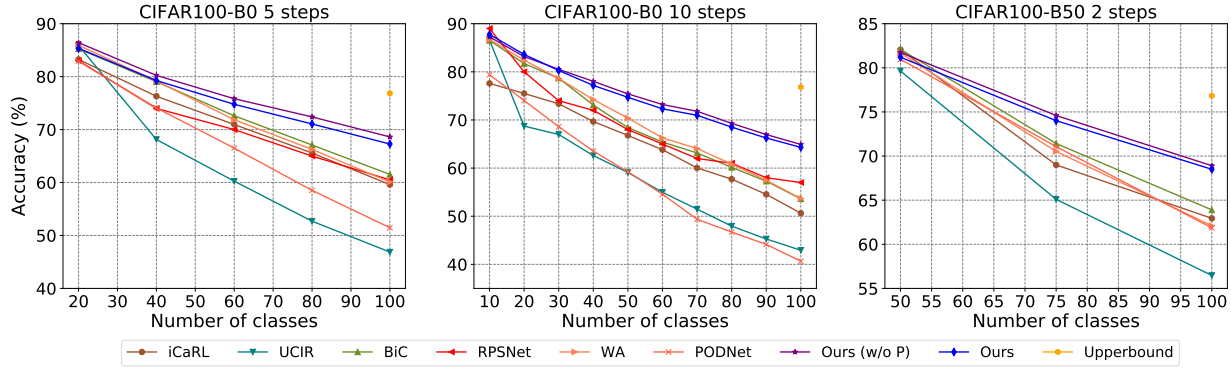

Figure 1: The performance for each step. **Left** is evaluated on CIFAR100-B0 of 5 steps. **Middle** is evaluated on CIFAR100-B0 of 10 steps. **Right** is evaluated on CIFAR100-B50 of 2 steps.

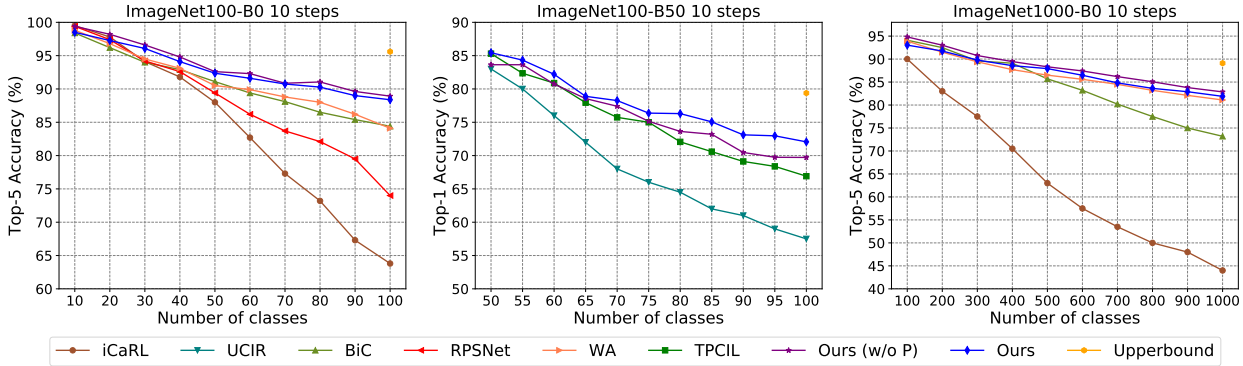

Figure 2: The performance for each step. **Left** is evaluated on ImageNet100-B0 of 10 steps. **Middle** is evaluated on ImageNet100-B50 of 10 steps. **Right** is evaluated on ImageNet1000-B0 of 10 steps.