Supplementary for Multi-Centroid Task Descriptor for Dynamic Class Incremental Inference

1. Visualization of Bipartite Results

As shown in Fig. 1, we visualize the correspondence between class prototypes and centroids under different class combinations for the task [lion, tiger, plate]. Since we sample a few classes randomly such that the class number is less than the centroids number, the categories inside a batch can be of three possible combinations [tiger, plate], [lion, plate], [tiger, lion]. The correspondence relationship can be determined by solving a bipartite problem using hungarian algorithm. For [tiger, plate], [lion, plate], because of the high similarity between tiger and lion, they are assigned to the same centroid, which is able to represent the two similar classes as a group. However, for [lion, tiger], because the bipartite problem requires that no two classes share the same centroids, one of these categories is assigned to the wrong centroid. However, this problem will be mitigated by other correct match results.

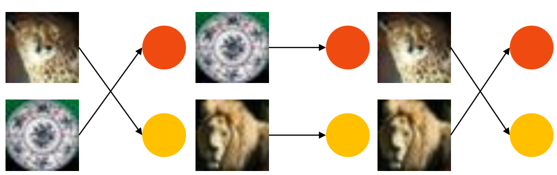

Figure 1. The visualization of the class-centroid correspondence relationship under different class combinations in a batch. Left: [tiger, plate]. Mid: [lion, plate]. Right: [tiger, lion]

2. Hyperparameters

We provide the hyperparameters in Tab. 1. The parenthesizes are used to distinguish different hyperparameters used for supernet and gate training. Due to the scale of different datasets, we tune the epoch number, learning rate scheduler, and batch size for better convergence.

3. Results on Modified Resnet-32

Some old works like iCaRL [2] use the 32-layer ResNet as the backbone. However, DER [3] uses ResNet-18 instead of ResNet-32 and they claim that ResNet-18 can achieve much higher accuracy than ResNet-32. Moreover, they provide quantitative results with ResNet-32 and so do we. The results on CIFAR100-B0 are shown in Tab. 3 and the results on CIFAR100-B50 are shown in Tab. 4. It appears that our method still outperforms DER even with a small network.

4. Results on Small-Scale Datasets

As shown in Tab. 2, we also conduct experiments on small-scale benchmarks used by CCCG [1]. The evaluation protocols used in CCGN are 1) Split MNIST, Split SVHN, and Split CIFAR-10. These incremental protocols are constructed by dividing **MNIST**, **SVHN**, and **CIFAR-10** datasets into 5 tasks with each containing 2 classes. 2) ImageNet-50. A subset of the **ImageNet-1000** containing 50 randomly selected classes. We divide the selected 50 classes into 5 tasks with each containing 10 classes and resize the images to a resolution of 32×32 pixels. Note that CCGN uses a generative model to synthesize data from old tasks, we use a memory of 1000 images for rehearsal as a replacement. It turns out that even on small-scale datasets, our model still surpasses CCGN by a large margin.

5. Visualization of Task Prediction

To better show the effectiveness of our gate network, we visualize the task ID prediction results for images from different classes on CIFAR100-B0S10. As shown in Fig. 2, each panel represent classes from one tasks. We can see that some samples have the first and second largest task probabilities the similar value, and the relative relationship should be taken into consideration for dynamic inference.

References

[1] Davide Abati, Jakub Tomczak, Tijmen Blankevoort, Simone Calderara, Rita Cucchiara, and Babak Ehteshami Bejnordi. Conditional channel gated networks for task-aware continual learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3931–3940, 2020. 1

[2] 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

	CIFAR100	ImageNet100 B0S10	ImageNet100 B50S10	ImageNet1000 B0S10
epochs	130	130	120	130
lr scheduler	[70, 110]	[70, 110]	[50, 70, 90]	[40, 70, 100]
batch size	64(supernet) 256(gate)	128(supernet) 256(gate)	128(supernet) 256(gate)	128(supernet) 1024(gate)
warmup	10	10	5	10
lr	0.1			
warmup lr	0.01			
momentum	0.9			
weight decay	5e-4			
ζ	12			

Table 1. Hyperparameters for different protocols.

	Split MNIST	Split SVHN	Split CIFAR10	ImageNet-50
CCGN	97.27	83.41	70.06	35.24
Ours	**98.28**	**84.49**	**85.01**	**63.72**

Table 2. Quantitative results on small datasets including Split MNIST, Split SVHN, Split CIFAR-10, and ImageNet-50.

[3] Shipeng Yan, Jiangwei Xie, and Xuming He. Der: Dynamically expandable representation for class incremental learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021. 1

Methods	5 steps		10 steps		20 steps		50 steps	
	#P	Avg	#P	Avg	#P	Avg	#P	Avg
iCaRL	0.46	67.20	0.46	64.04	0.46	61.16	0.46	57.00
BiC	0.46	68.92	0.46	66.15	0.46	63.80	0.46	-
WA	0.46	70.00	0.46	67.25	0.46	64.33	0.46	-
DER(ResNet-32)	1.38	73.00	2.53	71.29	4.83	71.07	11.73	70.58
Ours(ResNet-32)	+0.46	**74.73**	+0.46	**74.67**	+0.46	**74.37**	+0.46	**72.41**

Table 3. Quantitative results on CIFAR100-B0 (average over 3 runs). $\#P$ means the average number of parameters in millions. *Avg* means the average accuracy (%) over steps. $+0.46$ indicates the extra number of parameters brought by the gate network.

Methods	2Steps		5Steps		10Steps	
	#P	Avg	#P	Avg	#P	Avg
UCIR	0.46	66.76	0.46	63.42	0.46	60.18
PODNet	-	-	0.46	64.83	0.46	64.03
TPCIL	-	-	0.46	65.34	0.46	63.58
DER(ResNet-32)	0.92	70.18	1.61	68.52	2.76	67.09
Ours(ResNet-32)	+0.46	**72.04**	+0.46	**71.31**	+0.46	**70.66**

Table 4. Quantitative results on CIFAR100-B50 (average over 3 runs). $\#P$ means the average number of parameters in millions. *Avg* means the average accuracy (%) over steps. $+0.46$ indicates the extra number of parameters brought by the gate network.

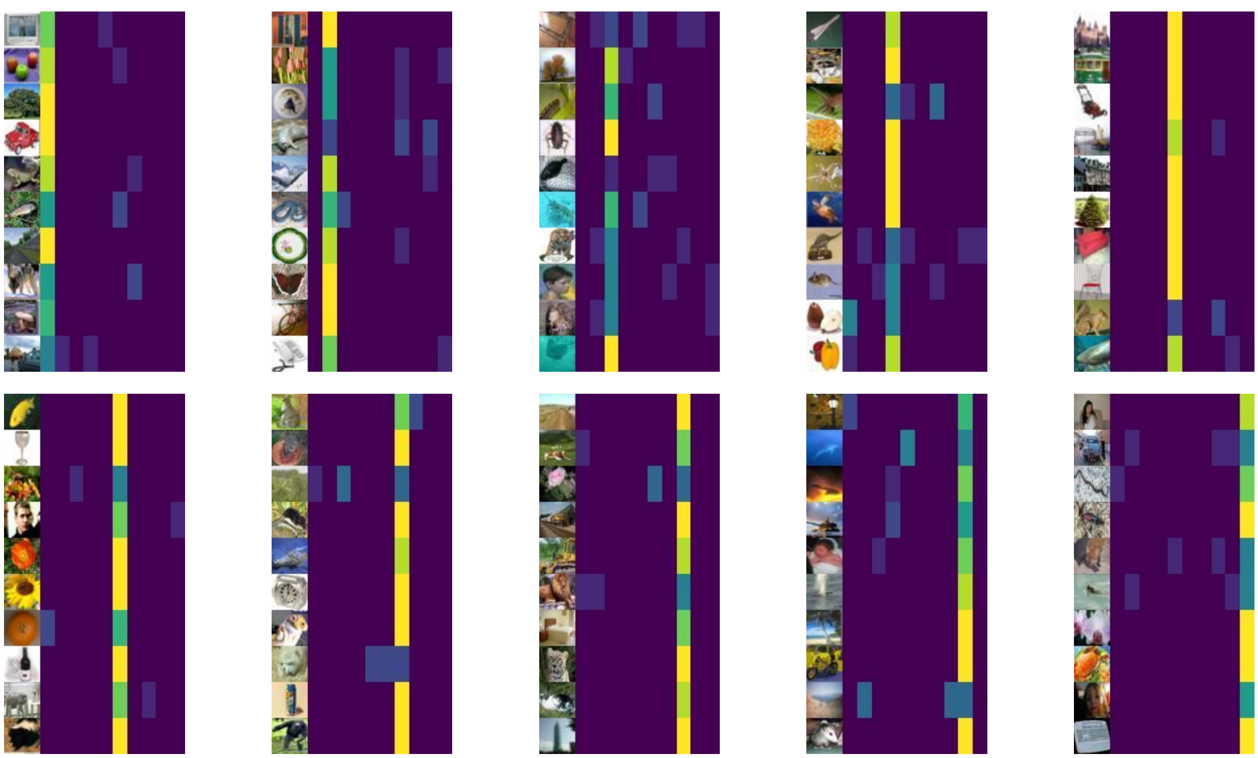

Figure 2. Samples from different classes and their corresponding task ID prediction results on CIFAR100-B0S10. Each panel represents a task.