1 Supplementary

1.1 Theoretical Derivation of L_{net} and L_{data}

According to equation (1), to achieve the goal of continual learning in the domain-incremental classification setting, $\sum_{j=1}^{|\mathbf{D}_i|} L_{CE}(F(\mathbf{\Theta}_t; I_{i,j}), c_{i,j})$ should be minimized for each task i. Since $\mathbf{D}_t$ and $\mathbf{\Theta}_t$ will be available when learning task t, we can directly calculate its loss, L_{CE}^t. Thus, we have:

$$\begin{aligned}
&\sum_{i=1}^{t}[\sum_{j=1}^{|\mathbf{D}_i|} L_{CE}(F(\mathbf{\Theta}_t; I_{i,j}), c_{i,j})] \\
=&L_{CE}^t + \sum_{i=1}^{t-1}[\sum_{j=1}^{|\mathbf{D}_i|} L_{CE}(F(\mathbf{\Theta}_t; I_{i,j}), c_{i,j})] \\
\triangleq&L_{CE}^t + \sum_{i=1}^{t-1} E_{(I,c)\sim\mathbf{D}_i}[L_{CE}(F(\mathbf{\Theta}_t; I), c)] \\
\triangleq&L_{CE}^t + \sum_{i=1}^{t-1} E_{I\sim\mathbf{D}_i}[D_{KL}(F(\mathbf{\Theta}_t; I)||F(\mathbf{\Theta}_i; I))]
\end{aligned} \tag{7}$$

Where I and c each represent a particular image and its corresponding class label, D_{KL} means the KL divergence. As L_{CE}^t can be easily calculated, we only need to estimate:

$$\sum_{i=1}^{t-1} E_{I\sim\mathbf{D}_i}[D_{KL}(F(\mathbf{\Theta}_t; I)||F(\mathbf{\Theta}_i; I))] \tag{8}$$

Since $(I, c) \sim \mathbf{D}_i, i = 1, 2, ..., t-1$ is not available in task t, we can estimate through existing data. The first alternative is to use buffer samples on behalf of former samples. We have buffer $\mathbf{B} \subset \bigcup_{i=1}^{t-1} \mathbf{D}_i$. Thus, euqation (8) can be replaced by:

$$\begin{aligned}
&E_{I\sim\mathbf{B}}[D_{KL}(F(\mathbf{\Theta}_t; I)||F(\mathbf{\Theta}_{I_d}; I))] \\
\triangleq&E_{I\sim\mathbf{B}}(||F(\mathbf{\Theta}_t; I) - F(\mathbf{\Theta}_{I_d}; I)||_2^2) \\
\triangleq&E_{(I,z)\sim\mathbf{B}}(||F(\mathbf{\Theta}_t; I) - z||_2^2) \\
=&\sum_{b=1}^{|\mathbf{B}|} ||F(\mathbf{\Theta}_t; I_b) - z_b||_2^2
\end{aligned} \tag{9}$$

Where $||.||_2$ is the $l2$ norm, I_d represents the task(domain) number of image I, z represents the logits, I_b and z_b is a stored pair of image and corresponding logits. Therefore, L_{data} in equation (5) can be a substitute for equation (1).

Another alternative is to use knowledge distillation to control the network changes when training the model. As long as current data $\mathbf{D}_t$ is available, we may replace equation (8) by:

$$\begin{aligned} & E_{I\sim\mathbf{D_t}}[D_{KL}(F(\mathbf{\Theta}_t;I)||F(\mathbf{\Theta}_{t-1};I))] \\ \triangleq & -\sum_{j=1}^{|\mathbf{D}_t|}(F(\mathbf{\Theta}_{t-1};I_{t,j})/Tem)\log(F(\mathbf{\Theta}_t;I_{t,j})/Tem) \end{aligned} \tag{10}$$

Where $|\mathbf{D}_t|$ represents the number of samples in current task t, $\mathbf{\Theta}_{t-1}$ and $\mathbf{\Theta}_t$ respectively represents the model before and after training task t, and Tem is the tempurate. Consequently, L_{net} also contributes to the minimization of equation (1).

1.2 Additional Information of Benchmarks

Digits. The Digits benchmark consists of (1)MNIST, (2)MNIST-M, (3)SVHN, (4)Synthesis. MNSIT is a handwritten dataset of digits, while MNIST-M is the modified MNIST with a randomly generated background. SVHN is the realistic street view, and the Sythesis dataset is generated colorful images. All datasets contain ten classes of digits, from '0' to '9'. We resize all images to be 32 * 32 to ensure the same scale in each task when training. Specific numbers of training and testing samples are listed in Table 1.

Pictures. PACS dataset contains sketch, cartoon, art painting and photo images of ['Dog', 'Elephant', 'Giraffe', 'Guitar', 'Horse', 'House', 'Person'] seven categories. Images have the size of 224 * 224. We divide the dataset and design the continual learning benchmark with the sequence of (1)Sketch, (2)Cartoon, (3)Art painting, (4)Photo. Besides, since the original dataset is used in domain adaptation and has no specific testing set for each domain, we split the last 10 % of samples in each domain as the testing set, numbers of which is listed in Table 1.

Processing. STL10 dataset is a processed subset of ImageNet, with 500 training samples and 800 testing samples in the size of 96 *96 for each of the ten categories - ['airplane', 'bird', 'car', 'cat', 'deer', 'dog', 'horse', 'monkey', 'ship', 'truck']. Here, we re-arrange the 13000 images for the Processing benchmark into four balanced parts, each performing different image processing strategies to create different domains. We set four sequential tasks (1)brightness (with a factor of 0.5), (2)grayscale, (3)sharpness (with a factor of 10), (4)contrast (with a factor of 3) changes with the transform function in PyTorch. Specific numbers of training and testing images with class-balanced samples are shown in Table 1.

Table 1. Number of Training and Testing Images

		MNIST	MNIST-M	SVHN	Synthetic
Digits	train	60000	60000	73257	10000
	test	10000	10000	26032	2000
		Sketch	Cartoon	Art Painting	Photo
Pictures	train	3534	2107	1840	1500
	test	395	237	208	170
		Brightness	Grayscale	Sharpness	Saturation
Processing	train	2750	2750	2750	2750
	test	500	500	500	500