Class Incremental Learning with Multi-Teacher Distillation

Supplementary Material

7. Additional Related Work

7.1. Weight Permutation

Weight permutation refers to permuting the order of neurons. The output of the neural network has the property that it is invariant to a group of weight permutations, i.e., the neural network has permutation symmetry [10, 43]. Therefore, weight permutation will generate a group of parameters that are equivalent to the original parameters in terms of output. [43] uses weight permutation to study the number of symmetry-induced critical points and the connectivity of global minima. [10] finds that SGD solutions will be in the same low-loss basin if applied with appropriate permutations. Similarly, [36] adopts weight permutation to remove symmetries and finds that flatter minima are closer to each other and minima found by different optimizers can be connected with zero-error paths. Furthermore, [3] proposes three model-matching methods based on permutation to teleport SGD solutions into a single basin. In addition, [35] makes the solving process of permutation differentiable via the Sinkhorn operator [1] and proposes a connectivity-based continual learning method for the task and domain incremental learning [45].

7.2. Discussion

We use weight permutation, feature perturbation, and diversity regularization to generate multiple teachers. Compared with existing multi-teacher distillation work [18, 24, 51, 53], teachers are generated from a basic model and do not need repetitive training models from scratch. Besides, the three techniques used for finding teachers are different from [32, 41]. We adopt the structure of multiple branches for teachers, which is similar to [15, 44]. Compared with existing weight permutation work [3, 10, 36], which aims to teleport minima into a single loss basin, we distribute the basic parameters to different loss basins in opposite directions. Our proposed MTD can be easily adapted to existing CIL work [8, 17, 19] and achieve significant performance improvement. We make a detailed discussion about the existing dual-teacher distillation method DT-CIL [4] in Section 11.1.

8. Loss Landscape

The loss landscapes in Figures 4 and 7 are colored 2-dimensional planes, the color of each point in a plane reflects the loss of the model evaluated on training samples of previous tasks. To visualize the loss landscape on the subspace spanned by parameters W_1, W_2, and W_3, we take the following procedures:

- Compute the basis of the plane, $\boldsymbol{U} = \boldsymbol{W}_2 - \boldsymbol{W}_1$ and $\boldsymbol{V} = \boldsymbol{W}_3 - \boldsymbol{W}_1$, where $\boldsymbol{W}_i$ is the vectorization of W_i;
- Orthogonalize $\boldsymbol{U}$ and $\boldsymbol{V}$ via $\boldsymbol{V} = \boldsymbol{V} - \frac{\boldsymbol{V}^{\mathrm{T}}\boldsymbol{U}}{\|\boldsymbol{U}\|^2}\boldsymbol{U}$ and Normalize them via $\boldsymbol{U} = \frac{\boldsymbol{U}}{\|\boldsymbol{U}\|}$, $\boldsymbol{V} = \frac{\boldsymbol{V}}{\|\boldsymbol{V}\|}$;
- The corresponding parameters of position (x, y) in the plane can be obtained via $\boldsymbol{W} = \boldsymbol{W}_1 + \boldsymbol{U}x + \boldsymbol{V}y$;
- Iteratively evaluate the training loss of parameters $\boldsymbol{W}$ corresponding to each point in the plane on previous tasks.

For the projection of parameters $\boldsymbol{W}$ onto the plane, we take the following step,

$$x = (\boldsymbol{W} - \boldsymbol{W}_1)^{\mathrm{T}}\boldsymbol{U}, \quad y = (\boldsymbol{W} - \boldsymbol{W}_1)^{\mathrm{T}}\boldsymbol{V}.$$

Due to the parameters $\boldsymbol{W}$ being obtained through linear combination, this will result in mismatches with the statistics in batch normalization (BN) layers of the model. We deal with this problem by forwarding data $\cup_{i=1}^{t}\mathcal{D}_i$ in the model for one epoch in the state of training.

9. Statistics between Intermediate Features

Figure 9 shows the cosine similarities between intermediate features of teachers found by "PFT" and "Oracle". It can be seen that the relationship between intermediate features remains consistent with the relationship between embeddings, i.e., the intermediate features of teachers found by "PFT" have high consistency in direction in feature space, and the intermediate features of teachers found by "Oracle" tend to be mutually orthogonal. In addition, the orthogonality between shallow features (at Stage 1) of teachers found by "Oracle" is not very obvious, and features of teachers found by "PFT" maintain high similarity at all stages. These statistics between intermediate features can guide us to further find diverse teachers.

10. Implementation Details

We use PyTorch [34] to reimplement CIL methods iCaRL [37], LUCIR [17], PODNet [8], AANet [27], and AFC [19] in the same environment for fair comparisons. The model architectures are the same as existing work [8, 19, 37], i.e., ResNet-32 [14] for CIFAR-100 and ResNet-18 for ImageNet. All hyperparameters of baseline methods and compared methods are the same as their original implementations. Same as existing work, the memory $\mathcal{M}_t$ stores 20 representative samples for each old class in all experiments. We use MTD to obtain one additional teacher in all comparison experiments, i.e., the total number of teachers is $n = 2$. The last two stages of layers in the model are copied for

Method	DTD			CUB200		
$\mathcal{A}(\%)\uparrow$	B7-5S	B7-10S	B7-20S	B100-5S	B100-10S	B100-20S
PODNet [8]	49.52($\pm$0.92)	48.42($\pm$0.89)	47.40($\pm$1.29)	47.27($\pm$0.56)	46.97($\pm$0.69)	46.45($\pm$0.45)
w/ MTD-S	49.89($\pm$0.11)	49.05($\pm$0.17)	48.78($\pm$0.32)	47.81($\pm$0.06)	47.53($\pm$0.06)	47.34($\pm$0.06)
w/ MTD-T	50.39($\pm$0.02)	49.27($\pm$0.08)	48.80($\pm$0.46)	48.25($\pm$0.11)	47.87($\pm$0.06)	47.77($\pm$0.12)
λ, β, η	50, 50, 50	10, 10, 10	10, 10, 10	1000, 1000, 1000		

Table 4. Comparison results on DTD and CUB200. "MTD-S" denotes the student $\mathcal{F}_t$. "MTD-T" denotes teachers $\overline{\mathcal{B}}_t$. "B" means the number of classes in the basic (first) task. "S" means the number of steps for learning the remaining classes. λ, β, η are the distillation coefficients in Equations (2) and (20). Experiments are run with 3 random seeds and results are reported with mean and standard deviation.

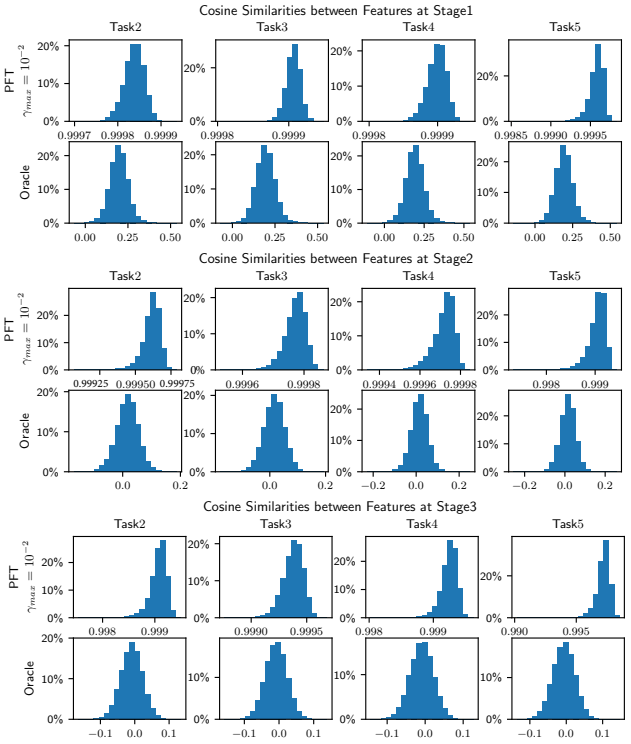

Figure 9. Cosine similarities between intermediate features of teachers found by "PFT" and "Oracle". The horizontal axis is the value of similarity, and the vertical axis is the percentage of embeddings. "Features at Stage 1" denotes features are final outputs of the Stage 1 layers in the ResNet-32 model on the CIFAR-100 B50-5S benchmark.

$n-1$ times to generate $n-1$ branches. The strength of the feature perturbation $\alpha \in \{0.03, 0.07\}$. The coefficient for diversity regularization $\rho \in \{0.7, 1.0, 2.0\}$. The coefficients for distillation $\lambda, \beta, \eta \in \{1, 100, 300, 1000, 1500\}$. The reason why the coefficients for distillation have such a large numerical range is that MTD needs to be applied to different CIL methods and different settings of benchmarks. The learning rate of branches $\gamma \in \{10^{-3}, 10^{-2}\}$. Experiments on CIFAR-100 are run 3 times with random seeds $\{1993, 1994, 1995\}$ to show statistical improvement. Experiments on ImageNet are run with the random seed 1993.

11. Additional Comparison Experiments

11.1. Comparison with DT-CIL

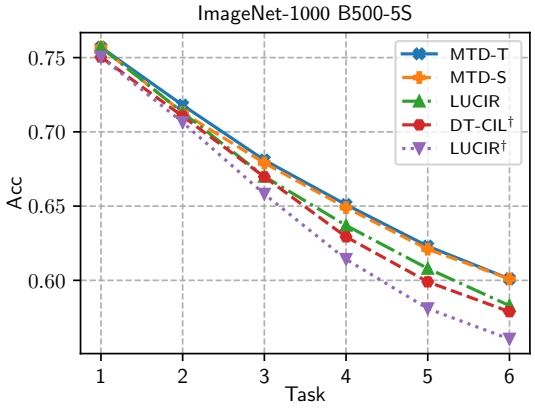

Figure 10. Testing accuracy curves of DT-CIL [4] and MTD taking LUCIR [17] as the baseline on the ImageNet-1000 B500-5S benchmark. † denotes results are referenced from the original paper of DT-CIL. We reimplement LUCIR and take it as the baseline of MTD for fair comparisons. "MTD-S" denotes the student $\mathcal{F}_t$ and "MTD-T" denotes teachers $\overline{\mathcal{B}}_t$ (Equation 16).

In this section, we compare the proposed MTD with the existing dual-teacher distillation method DT-CIL [4]. There are two teachers in DT-CIL, one is the model learned in the previous task, and one is the model individually trained on the new task. To achieve data-free replay, a conditional generator is learned for generating samples of previous tasks. Finally, two teachers with the generator supervise the model of the new task via information distillation. Therefore, DT-CIL involves two additional training procedures for the generator and the teacher regarding the new task.

Compared with DT-CIL, MTD only uses the accumulated memory $\mathcal{M}_t$ to find diverse teachers from a basic model based on the properties of "Oracle", which can be seamlessly embedded in the phase of class-balanced fine-tuning and does not involve additional training procedures. In addition, teachers in MTD are represented as branches and share most of the feature extraction layers. Therefore, MTD reduces additional time and memory consumption compared with DT-CIL. Figure 10 shows the accuracy

curves of DT-CIL and MTD taking LUCIR [17] as the baseline on ImageNet-1000 B500-5S benchmark. It can be seen that MTD can achieve higher performance than DT-CIL.

11.2. Results on Other Datasets

We choose two other datasets for further evaluation of our method. **DTD** contains 47 classes with image sizes ranging from 300×300 to 640×640 and each class has 40 training, validation, and testing samples [5]. **CUB200** contains 200 classes and each class has about 30 training samples and 30 testing samples [48]. We take PODNet as the baseline and use the same hyperparameter settings as those on ImageNet. The hyperparameters of MTD are $n = 2, \gamma = 10^{-3}, \rho = 1$, $\alpha = 0.03$. Table 4 shows the results of MTD adapting to PODNet on DTD and CUB200, and the settings of hyperparameters λ, β, η under different benchmarks. It can be seen that MTD can still be effective on other datasets. MTD-T improves PODNet by 0.87%, 0.85%, and 1.40% for 5, 10, and 20 steps of learning on DTD respectively. MTD-T improves PODNet by 0.98%, 0.90%, and 1.32% for 5, 10, and 20 steps of learning on CUB200 respectively.

11.3. Generalization to Settings without Exemplar

To evaluate MTD in this setting, we follow existing continual learning methods to construct the FIVE dataset, which contains CIFAR10, MNIST, SVHN, FashionMNIST, and notMNIST sub-datasets. We take LwF [25] as the baseline. After learning each task, we sample 5 samples for each class from the current task to optimize teachers instead of using $\mathcal{M}_t$. The distillation coefficient of LwF is set to 1 and the models are 2-layer MLP (MNIST) and ResNet32 (FIVE). The hyperparameters of MTD are $n = 2, \gamma = 10^{-3}$ (MNIST), 10^{-4} (FIVE), $\rho = 1$, and $\alpha = 0.03$. Table 5 shows the results of MTD adapting to LwF on MNIST and FIVE. It can be seen that MTD-T improves LwF by 1.11% and 1.95% on MNIST and FIVE respectively, which demonstrates the generality of MTD in settings without exemplar. The reason for setting the coefficient β to 1000 in MNIST experiments is that we want to inherit more knowledge from previous teachers to reduce forgetting.

$\mathcal{A}(\%)\uparrow$	MNIST B0-5T	FIVE B0-5T
LwF [25]	79.07($\pm$1.53)	41.72($\pm$1.43)
w/ MTD-S	80.34($\pm$0.96)	43.02($\pm$0.66)
w/ MTD-T	80.18($\pm$0.81)	43.67($\pm$0.90)
λ, β, η	1, 1000, 1	1, 1, 1

Table 5. Comparison results on MNIST and FIVE without exemplar. Each benchmark contains 5 tasks. Results are obtained by 3 runs and reported with mean and standard deviation.

11.4. Adapting to Other CIL Methods

In this part, we choose two additional CIL methods for the adaptation of MTD. SS-IL [2] applies the separated softmax output layer and task-wise knowledge distillation to resolve the classification score bias caused by the data imbalance between old and new data. ANCL [20] leverages an additional auxiliary network to pre-learn the new task and takes it together with the previous model as dual teachers for balance between stability and plasticity. We want to emphasize that MTD finds multiple teachers from a basic model based on the properties of "Oracle", which is different from ANCL in simply retraining the auxiliary network on the new task. The hyperparameters in ANCL are $\lambda = 1.0, \lambda_a = 0.5$. The hyperparameters in MTD are $n = 2, \gamma = 10^{-2}, \rho = 1.0, \alpha = 0.03$. Table 6 shows the results of MTD adapting to SS-IL and ANCL on CIFAR-100 benchmarks. It can be seen that MTD-T improves SS-IL by 1.48%, 0.95%, and 0.87% for 5, 10, and 25 steps of incremental learning respectively. MTD-T can improve ANCL by 2.11%, 2.60%, and 1.42% for 5, 10, and 25 steps of incremental learning respectively.

Method	CIFAR-100		
$\mathcal{A}(\%)\uparrow$	B50-5S	B50-10S	B50-25S
SS-IL [2]	59.83($\pm$0.25)	55.32($\pm$0.26)	49.68($\pm$0.07)
w/ MTD-S	59.93($\pm$0.10)	55.44($\pm$0.21)	50.35($\pm$0.25)
w/ MTD-T	61.31($\pm$0.17)	56.27($\pm$0.19)	50.55($\pm$0.29)
λ, β, η	4, 4 ,4,$n = 3$	3, 3, 3	1, 1, 1
ANCL [20]	66.03($\pm$0.19)	63.63($\pm$0.12)	59.97($\pm$0.13)
w/ MTD-S	67.00($\pm$0.15)	64.84($\pm$0.21)	60.11($\pm$0.19)
w/ MTD-T	68.14($\pm$0.29)	66.23($\pm$0.09)	61.39($\pm$0.21)
λ, β, η		1500, 1500, 1500	

Table 6. Comparison results of MTD adapting to SS-IL and ANCL. Results are obtained by 3 runs and reported with mean and standard deviation. We use 3 teachers when MTD adapting to SS-IL in the setting of B50-5S for better performance.

11.5. Repeated Experiments on ImageNet-100

To further demonstrate the statistical effectiveness of MTD, we provide the results of repeated experiments on ImageNet-100 in Table 7. It can be seen that MTD can still improve the performance of baselines, such as improving PODNet by 1.68%, 1.85%, and 3.71% for 5, 10, and 25 steps of incremental learning respectively.

Method	ImageNet-100		
$\mathcal{A}(\%)\uparrow$	B50-5S	B50-10S	B50-25S
PODNet [8]	76.47($\pm$0.14)	73.52($\pm$0.06)	64.86($\pm$1.31)
w/ MTD-S	77.52($\pm$0.17)	74.54($\pm$0.09)	67.55($\pm$0.48)
w/ MTD-T	78.15($\pm$0.23)	75.37($\pm$0.05)	68.57($\pm$0.53)
AFC [19]	77.25($\pm$0.05)	75.45($\pm$0.01)	72.65($\pm$0.23)
w/ MTD-S	77.67($\pm$0.11)	76.44($\pm$0.14)	73.71($\pm$0.13)
w/ MTD-T	77.82($\pm$0.03)	76.27($\pm$0.10)	72.87($\pm$0.13)

Table 7. Comparison results of MTD on ImageNet-100. Experiments are run 3 times with different random seeds and results are reported with mean and standard deviation.