R-DFCIL: Relation-Guided Representation Learning for Data-Free Class Incremental Learning

R-DFCIL: Relation-Guided Representation Learning for Data-Free Class Incremental Learning (Supplementary Material)

Qiankun Gao[1], Chen Zhao[2], Bernard Ghanem[2], and Jian Zhang[1]*

[1] Peking University Shenzhen Graduate School, Shenzhen, China
gqk@stu.pku.edu.cn, zhangjian.sz@pku.edu.cn
[2] King Abdullah University of Science and Technology (KAUST), Thuwal, Saudi Arabia
{chen.zhao,bernard.ghanem}@kaust.edu.sa

In the supplementary materials, we further validate the proposed approach by providing the following:

- Section A: Relational Knowledge Distillation in Detail.
- Section B: Details of Adaptive Scale Factors in RRL Loss.
- Section C: Additional Experimental Details.
- Section D: Additional Experimental Results.
- Section E: Feature Analysis and Bottlenecks in Prior Approaches.

A Relational Knowledge Distillation in Detail

We described the relation knowledge distillation (RKD) in Sec. 3.2 of the main paper. Here we give some more details of RKD, by illustrating the process in Fig. 1. We calculate the RKD loss for a triplet of new samples ($\mathbf{x}_a$,$\mathbf{x}_b$,$\mathbf{x}_c$) in several steps. **First**, we extract their representations on the old model (f_i) and the current model (f), respectively. **Then**, we transform the representations by two linear layers ϕ and ψ, and we get two groups of transformed representations $\mathbf{t}_*$ and $\mathbf{s}_*$, all of which are points ($\mathbb{R}^{2d}$ vectors) in the feature space. **Next**, from the point $\mathbf{t}^b$ we construct edges $\mathbf{e}^{ab} = \mathbf{t}^a - \mathbf{t}^b$ and $\mathbf{e}^{cb} = \mathbf{t}^c - \mathbf{t}^b$ in the feature space, and obtain the angle $\angle \mathbf{t}_a \mathbf{t}_b \mathbf{t}_c$ between the two edges. Similarly, we can get the angle $\angle \mathbf{s}_a \mathbf{s}_b \mathbf{s}_c$ by constructing edges $\mathbf{h}^{ab} = \mathbf{s}^a - \mathbf{s}^b$ and $\mathbf{h}^{cb} = \mathbf{s}^c - \mathbf{s}^b$. **Finally**, we compute the RKD loss about the sample $\mathbf{x}_b$ by Eq. (5) in the main paper. The RKD loss about the other two samples $\mathbf{x}_a$ and $\mathbf{x}_c$ are computed in the same way, which are omitted in the figure.

B Details of Adaptive Scale Factors in RRL Loss

In Sec. 3.2 of the main paper, we elaborated our relation-guide representation learning (RRL), in which hard knowledge distillation (HKD) prevents forgetting of previous knowledge, local cross-entropy loss (LCE) improves the model's

[*] Corresponding author: Jian Zhang.

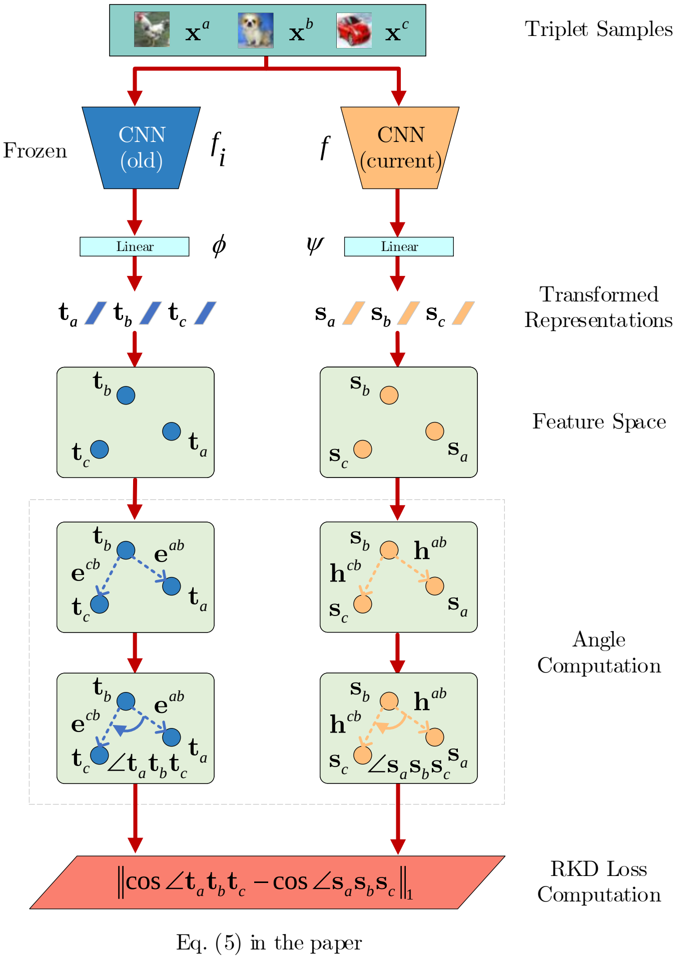

Triplet Samples
$\mathbf{x}^a$ $\mathbf{x}^b$ $\mathbf{x}^c$
Frozen
CNN (old)
f_i
f
CNN (current)
Linear
ϕ
ψ
Linear
$\mathbf{t}_a$ $\mathbf{t}_b$ $\mathbf{t}_c$
$\mathbf{s}_a$ $\mathbf{s}_b$ $\mathbf{s}_c$
Transformed Representations
Feature Space
$\mathbf{e}^{ab}$ $\mathbf{e}^{cb}$
$\mathbf{h}^{ab}$ $\mathbf{h}^{cb}$
Angle Computation
$\angle \mathbf{t}_a \mathbf{t}_b \mathbf{t}_c$
$\angle \mathbf{s}_a \mathbf{s}_b \mathbf{s}_c$
$\|\cos \angle \mathbf{t}_a \mathbf{t}_b \mathbf{t}_c - \cos \angle \mathbf{s}_a \mathbf{s}_b \mathbf{s}_c\|_1$
RKD Loss Computation
Eq. (5) in the paper

Fig. 1: **Relational Knowledge Distillation in Detail.** We first extract samples' representations, then transform them by linear layers. The angle is computed in the feature space and RKD loss is calculated by Eq. (5) in the paper.

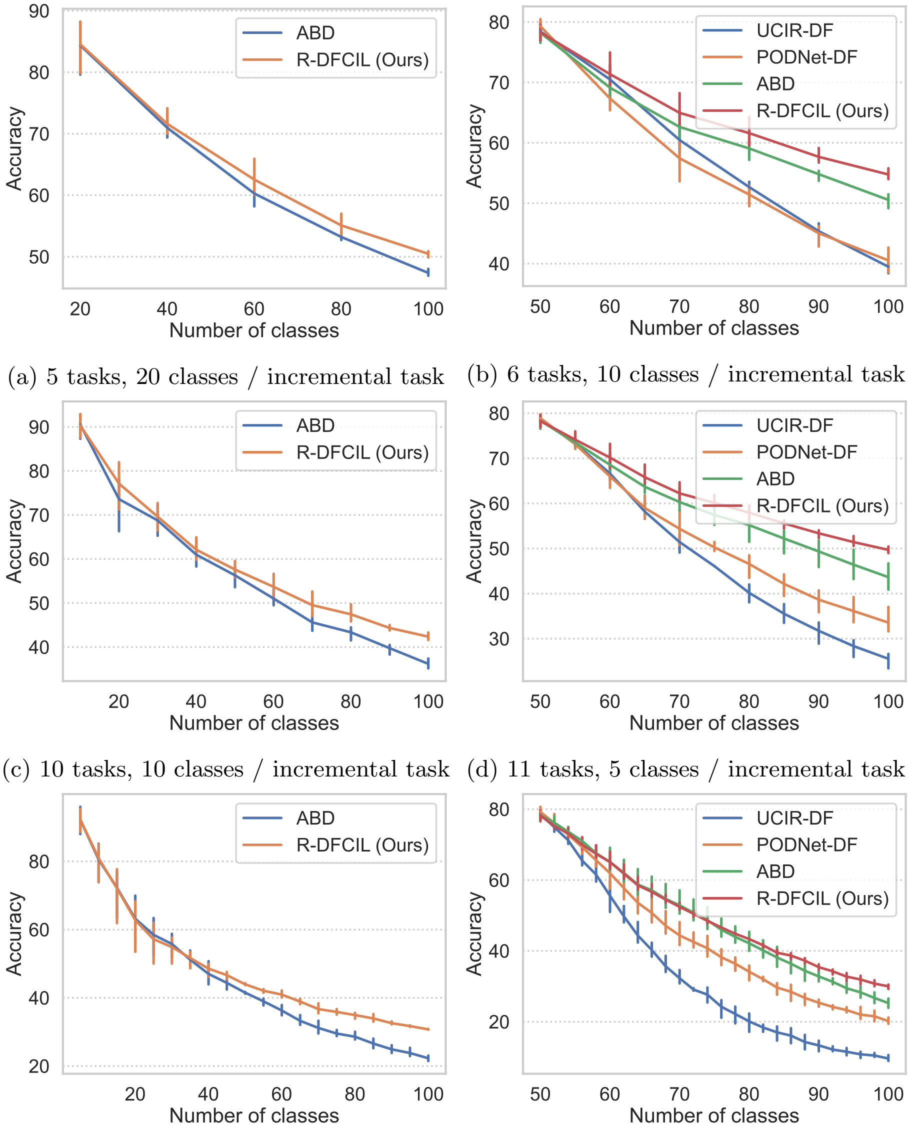
plasticity, and relational knowledge distillation (RKD) alleviates the conflict between them. However, the scale of these three components should adapt to the situations in practice since the number of classes the model has learned, and the number of classes in the new task vary at different times. **For one thing**, the larger the number of previous classes is compared to that of new classes $\frac{|\mathcal{T}_{1:i}|}{|\mathcal{T}_{i+1}|}$, the relatively more previous knowledge the model has to maintain, *i.e.*, the more difficult it is to prevent forgetting. Therefore, in Eq. (9) of the main paper, we scaled up the losses for HKD and RKD $\mathcal{L}_{hkd}$, $\mathcal{L}_{rkd}$ and scaled down the loss for LCE $\mathcal{L}_{lce}$ by $\beta = \sqrt{\frac{|\mathcal{T}_{1:i}|}{|\mathcal{T}_{i+1}|}}$. **For another**, the effect of $\mathcal{L}_{lce}$ becomes stronger as the number of new classes grows, which also increases the difficulty of preserving previous knowledge. For this reason, we scaled up $\mathcal{L}_{hkd}$, $\mathcal{L}_{rkd}$ by $\alpha = \log_2(\frac{|\mathcal{T}_{i+1}|}{2} + 1)$, which was carefully selected from a group of functions that are positively correlated to $|\mathcal{T}_{i+1}| \geq 2$ and start from 1 (when $|\mathcal{T}_{i+1}|$=2). In addition, the LCE loss $\mathcal{L}_{lce}$ gets weak when the number of classes is very small due to the reduction in the number of negative classes (*i.e.*, $|\mathcal{T}_{i+1}| - 1$), so we compensate the $\mathcal{L}_{lce}$ by $\frac{1}{\alpha}$. This adaptive strategy makes our RRL work better in various complex incremental learning situations.

C Additional Experimental Details

Data Augmentation. We follow prior works [2,1,5] to augment data in all of our experiments. For CIFAR100 [3], we first normalize images with means 0.5071, 0.4867, 0.4408 and standard variations 0.2675, 0.2565, 0.2761, then perform random crop with padding 4 and random horizontal flip with probability 0.5. For Tiny-ImageNet200 [4], the normalization means and standard variations are 0.4803, 0.4481, 0.3976 and 0.2764, 0.2688, 0.2816, and the others are same with CIFAR100. As for ImageNet100 [2], we first resize images into 256 × 256, then crop the images into 224 × 224 from the center and normalize them with means 0.485, 0.456, 0.406 and standard variations 0.229, 0.224, 0.225. After that, we apply the same random crop and random horizontal flip augmentations as for the other two datasets.

Image Synthesis Implementation Details. As we presented in Sec. 3.4 of the main paper, there are four optimization objectives (*i.e.*, label diversity, data content, stat alignment and image prior) for training the synthesizer, scale factors of which are set to 1,1,5 and 0.001 in all experiments. The temperature parameter α_{temp} in label diversity loss is set to 1000. The number of training steps are 5000 for CIFAR100 [3] and Tiny-ImageNet200 [4], and 10000 for ImageNet100 [2].

Hyperparameter Tuning. We empirically set λ_{lce} to 0.5 and tuned the other two parameters by a simple grid search from $\{0.05, 0.5, 5.0\}$ on CIFAR100 with $N = 5$. Then, we found that the performance was insensitive to the value of λ_{rkd}, but the performance collapsed with the largest $\lambda_{hkd} = 5.0$. So, we further searched λ_{hkd} from $\{0.25, 0.15\}$, and the model performed better with 0.15. Finally, we conducted all other experiments on three datasets with $\lambda_{lce} = 0.5$, $\lambda_{hkd} = 0.15$ and $\lambda_{rkd} = 0.5$, which verified their effectiveness and generality.



(a) 5 tasks, 20 classes / incremental task

(b) 6 tasks, 10 classes / incremental task

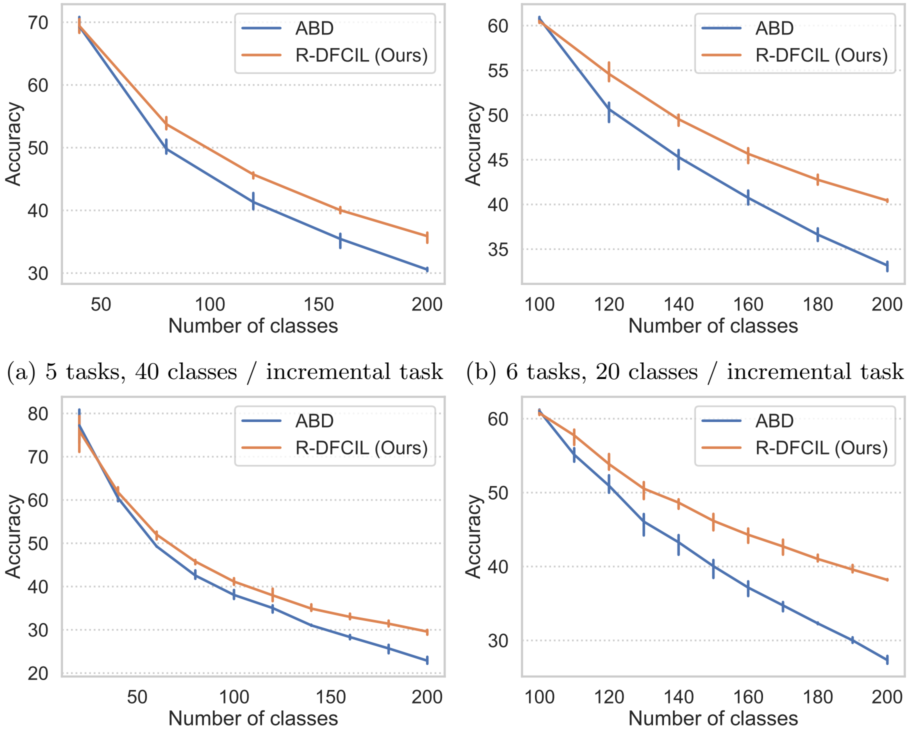
(c) 10 tasks, 10 classes / incremental task

(d) 11 tasks, 5 classes / incremental task

(e) 20 tasks, 5 classes / incremental task

(f) 26 tasks, 2 classes / incremental task

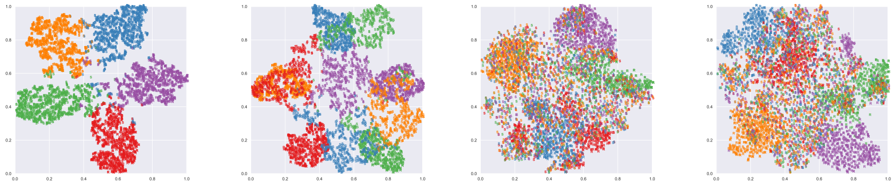
Fig. 2: **Incremental Accuracy on CIFAR100**. The margins between our R-DFCIL and other approaches gradually increase as the number of learned classes grows. The popular CIL approaches (UCIR, PODNet) work badly with synthetic data (UCIR-DF, PODNet-DF). The means and standard deviations are reported of three runs with random class orders.



(a) 5 tasks, 40 classes / incremental task (b) 6 tasks, 20 classes / incremental task

(c) 10 tasks, 20 classes / incremental task (d) 11 tasks, 10 classes / incremental task

Fig. 3: **Incremental Accuracy on Tiny-ImageNet200**. The differences between our R-DFCIL and ABD are more sigificant on Tiny-ImageNet200 than on CIFAR100, though Tiny-ImageNet200 is more challenging. This may because the model is more prone to forgetting when there are more classes in incremental task, and our R-DFCIL is better at mitigating forgetting. The means and standard deviations are reported of three runs with random class orders.



(a) Joint (b) Independent (c) ABD (d) Ours

Fig. 4: **Visualization of t-SNE on synthetic data and real data**. "R" and "S" represent real and synthetic. Please zoom in. Same class have same color. The classes are randomly sampled. (a) jointly and (b) independently reduce dimensions of synthetic data and real data of the first task, respectively, showing impacts of semantic and domain features, respectively. (c) and (d) are feature distributions of ABD's and ours final models, respectively.

D Additional Experimental Results

Additional Incremental Accuracy Plots. We depicted the 20-tasks and 26-tasks task-by-task incremental accuracy in Fig. 2 of the main paper. Here we provide more task-by-task incremental accuracy plots in Fig. 2, 3 of this supplementary document. For the evaluation protocol introduced by Hou *et al*. [2], we implement the Data-Free UCIR (UCIR-DF) [2] and PODNet (PODNet-DF) [1], and present their performances in Fig. 2b, 2d, 2f.

Ablation of Knowledge Distillation. In the Fig. 3 of the main paper, we ablate the RKD and HKD by directly removing them. We also conducted experiments that replace one with the other. The $\bar{A}_{20}/A_{20}$ drops from 49.47%/30.92% to 40.17%/21.35% for a single run when RKD is replaced with HKD, and decreases to 22.19%/5.81% when HKD is taken place by RKD. All these ablation studies demonstrate the importance of applying appropriate knowledge distillation methods to synthetic old and real new data.

Plasticity and Stability Comparison with ABD. In 5-tasks CIFAR100 experiments, we tuned the KD weights to compare model's plasticity (based on the last accuracy) with controlled stability. Our R-DFCIL is higher than ABD by 7% in last accuracy on the new task (better plasticity), with 4% higher last accuracy on old tasks (also better stability).

Newer Relational Knowledge Distillation. We also investigated some newer KD methods, such as CRD [6] and CRCD [7]. These methods are more complex and do not solve the DFCIL problem as effectively as RKD. In our experiments, using RKD achieves an average accuracy gain of 5% (CRCD) and 10% (CRD).

E Feature Analysis and Bottlenecks in Prior Approaches

In Sec. 1 of the main paper, we identified two bottlenecks in previous DFCIL approaches according to our study on feature analysis. Here we define semantic features as the discriminate features between classes and domain features as the features shared by a kind of data (real or synthetic). As shown in Fig. 4a, the synthetic and real data of the same class cluster together because the semantic features dominate domain features when jointly reducing dimensions. On the contrary, semantic features are dominated by domain features when independently reducing the dimensions of synthetic and real data (Fig. 4b), resulting in the mixing of the synthetic data that belong to different classes. Figure 4c and 4d show that the overlap between real and synthetic data in ABD is more severe than ours, and the clustering of real data in ABD is worse than ours, indicating that our R-DFCIL learned more semantic features, *i.e.*, we address the first bottleneck with local classification loss during representation learning.

The figures also illustrate the different decision boundaries between classes in synthetic and real data. Our R-DFCIL is more robust to disturbed decision boundaries in synthetic data because we solve the second bottleneck by applying different knowledge distillation on real and synthetic data, effectively alleviating the conflict between the model's plasticity and stability.

References

1. Douillard, A., Cord, M., Ollion, C., Robert, T., Valle, E.: Podnet: Pooled outputs distillation for small-tasks incremental learning. In: Proceedings of the European Conference on Computer Vision (ECCV) (2020)
2. Hou, S., Pan, X., Loy, C.C., Wang, Z., Lin, D.: Learning a unified classifier incrementally via rebalancing. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2019)
3. Krizhevsky, A., Hinton, G., et al.: Learning multiple layers of features from tiny images. Technical Report (2009)
4. Le, Y., Yang, X.: Tiny imagenet visual recognition challenge. CS 231N (2015)
5. Smith, J., Hsu, Y.C., Balloch, J., Shen, Y., Jin, H., Kira, Z.: Always be dreaming: A new approach for data-free class-incremental learning. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV) (2021)
6. Tian, Y., Krishnan, D., Isola, P.: Contrastive representation distillation. In: Proceedings of the International Conference on Learning Representations (ICLR) (2020)
7. Zhu, J., Tang, S., Chen, D., Yu, S., Liu, Y., Rong, M., Yang, A., Wang, X.: Complementary relation contrastive distillation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2021)