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

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

Abstract. Class-Incremental Learning (CIL) struggles with catastrophic forgetting when learning new knowledge, and Data-Free CIL (DFCIL) is even more challenging without access to the training data of previously learned classes. Though recent DFCIL works introduce techniques such as model inversion to synthesize data for previous classes, they fail to overcome forgetting due to the severe domain gap between the synthetic and real data. To address this issue, this paper proposes relation-guided representation learning (RRL) for DFCIL, dubbed R-DFCIL. In RRL, we introduce relational knowledge distillation to flexibly transfer the structural relation of new data from the old model to the current model. Our RRL-boosted DFCIL can guide the current model to learn representations of new classes better compatible with representations of previous classes, which greatly reduces forgetting while improving plasticity. To avoid the mutual interference between representation and classifier learning, we employ local rather than global classification loss during RRL. After RRL, the classification head is refined with global class-balanced classification loss to address the data imbalance issue as well as learn the decision boundaries between new and previous classes. Extensive experiments on CIFAR100, Tiny-ImageNet200, and ImageNet100 demonstrate that our R-DFCIL significantly surpasses previous approaches and achieves a new state-of-the-art performance for DFCIL. Code is available at https://github.com/jianzhangcs/R-DFCIL

Keywords: Incremental Learning, Data-Free, Representation Learning

1 Introduction

Class-Incremental Learning (CIL) is a learning paradigm in which a model (referred to as a solver model) continually learns a sequence of classification tasks. The model suffers from catastrophic forgetting [5,21] since its access to data of previous tasks is restricted when learning a new task. Existing CIL

⋆ Corresponding author: Jian Zhang.

works [20,8,4,2,15] try to overcome the challenge mainly through saving a small proportion of previous training data in memory. Despite their success of mitigating catastrophic forgetting, these approaches may bring issues such as violation of data legality and explosion of storage space. Instead, some works [24,3,9] simultaneously train the solver model and a data generator, which is used to generate data for previous classes at a new task. This usually performs poorly and still causes data privacy concerns because the generator may remember sensitive information in the real data. To address these concerns, researchers start to consider Data-Free CIL (DFCIL) [14,31,25], in which the model incrementally incorporates new information without storing data or generator of previous tasks.

Early DFCIL works, *e.g.*, LwF [14], are often ineffective in overcoming catastrophic forgetting without data of previous tasks [27]. More recently, Yin *et al.* introduce model inversion [31] to DFCIL to synthesize data for previous tasks when learning a new task, the forgetting of previous classes can be mitigated by performing knowledge distillation on these synthetic data. However, the synthetic data have a severe domain gap with the real data, misleading the decision boundaries between new and previous classes. These approaches may come through the first few tasks (*i.e.*, short-term CIL), but they lose the stability-plasticity balance when learning many tasks (*i.e.*, long-term CIL). It is still a great challenge to train a model with both good stability (*i.e.*, not forgetting previous knowledge) and plasticity (*i.e.*, learning new knowledge) in DFCIL.

After a thorough study on DFCIL with synthetic data of previous classes, we identify bottlenecks in prior approaches as follows: **1)** with the existence of domain gap between synthetic and real data, the global classification loss (*i.e.*, the cross-entropy between the model's prediction among all seen classes and the ground truth) leads classifiers to separate new and previous classes by domain features rather than semantic features, which also causes the model to learn more domain features of synthetic data than semantic features of previous classes; **2)** to overcome forgetting, prior works perform the same knowledge distillation method on the synthetic data and the data of new classes, ignoring the difference between them, which actually hurts the model's plasticity and is not helpful in alleviating the conflict between improving plasticity and maintaining stability. Please refer to the supplementary material for more details.

To address the above bottlenecks, we propose **1)** relation-guided representation learning (RRL) with hard knowledge distillation (HKD) for synthetic old data together with the relational knowledge distillation (RKD) for data of new task; **2)** local classification loss (*i.e.*, the cross-entropy between the model's prediction among new classes and ground truth) in place of global classification loss during representation learning, following classification head refinement with global class-balanced classification loss using a small learning rate.

Specifically, our novel approach R-DFCIL consists of three stages: **1)** before learning a new task, we **train an image synthesizer** by inverting the old model through model inversion technique [31], which is used to synthesize image during learning new task; **2)** we design three components to encourage the model to learn the representations of new classes without forgetting learned classes, in

which **local classification loss** improves model's plasticity, **hard knowledge distillation** maintains model's stability, and **relational knowledge distillation** mitigates the conflict between them; **3)** after representation learning, we refine the classification head to address the data imbalance between classes as well as learn the decision boundaries between new and previous classes, in which a **global class-balanced classification loss** is adopted, and the weights of classes are computed by their number of training samples.

We summarize our contributions as follows:

- We propose a novel DFCIL approach R-DFCIL, which strikes a better stability-plasticity balance by relation-guided representation learning (RRL) and classification head refinement (CHR).
- To the best of our knowledge, we are the first to introduce relational knowledge distillation (RKD) to DFCIL, which is critical to mitigate the conflict between learning the representations for new classes and preserving the representations of previously learned classes.
- We conduct extensive experiments on CIFAR100 [10] , Tiny-ImageNet200 [11], and ImageNet100 [8] datasets, on all of which, our R-DFCIL surpasses the previous state-of-the-art ABD [25] with accuracy gains of 8.46%, 9.23%, and 9.88%, respectively, and achieves a new record for DFCIL.

2 Related Work

Class-Incremental Learning (CIL). To overcome catastrophic forgetting, successful approaches [20,2,8,32,4,19,15,1] store representative training data for previously learned classes and replay them when updating the model with the data from new task. Knowledge distillation (KD) [7] techniques are widely used in these approaches to further alleviate forgetting of learned information, *e.g.*, iCaRL [20] conducts KD on the pre-softmax output of the old and new data, UCIR [8] designs a novel feature distillation loss, and PODNet [4] proposes to distill from not only the final embedding output but also the pooled output of the model's intermediate layers. However, these methods are not suitable for synthetic data, so we adopt a hard KD, which directly distills the knowledge from the model's output. PODNet requires another stage to fine-tune the classifier with balanced data, our approach also has a classification head refinement stage, in which the model addresses the data imbalance issue and learns decision boundaries between new and previous classes with a global class-balanced classification loss. The classification head also impacts the incremental performance: iCaRL works better with NME than CNN classifier, UCIR is more compatible with cosine classifier, and PODNet depends on LSC classifier. We remove the bias parameter of the linear classifier to adapt our approach better.

Data-Free Class-Incremental Learning (DFCIL). The earliest DFCIL work is LwF [14], which first introduces knowledge distillation (KD) to incremental learning. Unfortunately, KD has limited effectiveness in overcoming forgetting when using only new data. Some prior works [24,3,9,30,28] train a large generator simultaneously with the training of the solver model, which helps the solver

model remember the knowledge of previous tasks through replaying the generated data. These approaches usually perform poorly [27] due to the domain gap between generated and real data, and they also cause data privacy concerns because the generator may remember sensitive information in the real data [16]. Recent works [31,25] introduce model inversion technique to synthesize data of previous tasks. Although the visual quality is very different from the real images, the synthetic images generally match the statistical distribution of the real data from previous tasks. The synthetic images often mix features from multiple classes, which confuse the decision boundaries between classes, the prior approaches that overcome forgetting with real data may fail with synthetic data. Our approach follows ABD [25] to synthesize data for previous classes by model inversion technique, but we further propose a training framework that separates representation and classifier learning to avoid the mutual interference caused by domain gap between synthetic and real data.

Knowledge Distillation (KD) was first introduced to Deep Learning by Hinton *et al.* [7] to transfer knowledge from a teacher model to a small student model. Since then, various KD methods [22,13,26,18] have been developed. Conventional KD methods extract knowledge from individual data, *i.e.*, keep the hidden activation or the final output of the student model consistent with those of the teacher model for individual training samples. In contrast, Park *et al.* [17] propose Relational KD (RKD) to transfer structural knowledge using mutual relations of data examples in the teacher's output presentation. Their experimental results demonstrate that RKD is superior to conventional individual KD (IKD) methods. KD techniques are also widely used in incremental learning to overcome catastrophic forgetting, but most of them are IKD methods. These IKD methods can improve the model's stability when applied to old data but may hurt the model's plasticity when applied to new data. Inspired by RKD, we propose relation-guided representation learning to address DFCIL problem.

3 Methodology

3.1 Problem formulation and R-DFCIL architecture

Problem formulation. In the problem of Data-Free Class-Incremental Learning (DFCIL), a model sequentially learns a series of tasks, in which the i^{th} task introduces a set of classes $\mathcal{T}_i$ that do not overlap with those in previous tasks. We use $\mathcal{T}_i$ and the i^{th} task interchangeably in this paper, and denote the number of classes in $\mathcal{T}_i$ as $|\mathcal{T}_i|$. At learning phase i, the model can only access the training data of the current task $\mathcal{T}_i$, and predicts for all the data of the tasks $\mathcal{T}_{1:i}$ (*i.e.*, from $\mathcal{T}_1$ to $\mathcal{T}_i$) for inference after the learning is finished. We denote the feature extractor with stacks of convolutional layers as $f : \mathbb{R}^{h\times w\times 3} \rightarrow \mathbb{R}^d$, and the classification head with c linear classifiers as $\theta : \mathbb{R}^d \rightarrow \mathbb{R}^c$, then the model $\theta \circ f$ predicts the class y of input $\mathbf{x}$ via $\hat{y} = \arg\max_{j\in\{0,\ldots,c-1\}} \theta^{(j)}(f(\mathbf{x}))$. For simplicity, we denote the frozen snapshot of $\theta \circ f$ at the end of learning phase i as $\theta_i \circ f_i$, which means $\theta_i \circ f_i$ has learned $\mathcal{T}_{1:i}$. The training data and test data

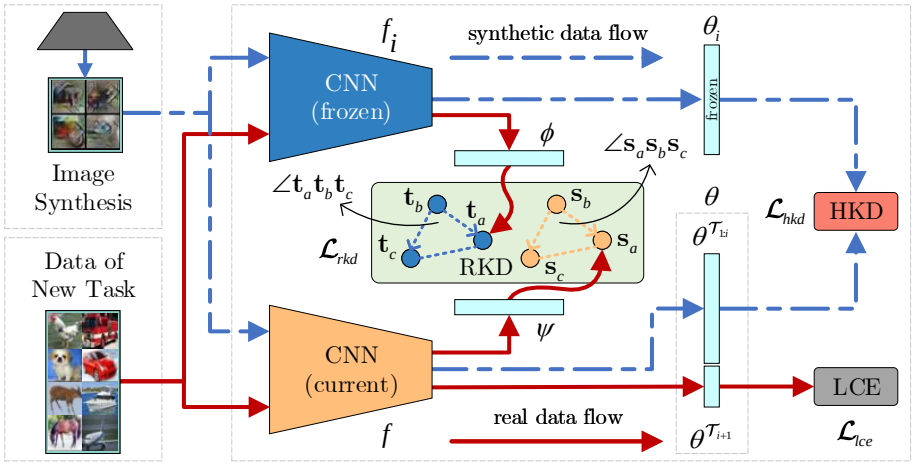


Fig. 1: Overview of our R-DFCIL. The model $\theta \circ f$ is learning the current task $\mathcal{T}_{i+1}$. The hard knowledge distillation loss $\mathcal{L}_{hkd}$ is applied on synthetic data to alleviate forgetting. The local classification loss $\mathcal{L}_{lce}$ is employed on new data to learn new knowledge. The relation knowledge distillation loss $\mathcal{L}_{rkd}$ transfers the structural relation of new data $\mathcal{D}_{i+1}^{train}$ from the previous model $\theta_i \circ f_i$ to the current model $\theta \circ f$. ϕ and ψ are two linear transform functions.

of $\mathcal{T}_i$ are described by $\mathcal{D}_i^{train}$ and $\mathcal{D}_i^{test}$, respectively. We also refer $\mathcal{D}_{1:i}^{train}$ and $\mathcal{D}_{1:i}^{test}$ to the training and test data of $\mathcal{T}_{1:i}$ for convenience.

R-DFCIL architecture. Fig. 1 illustrates the architecture of our relation-guided representation learning for DFCIL (R-DFCIL). Our R-DFCIL is based on the framework that synthesizes old data when learning a new task, and contains the following three stages: **First**, at the beginning of the learning phase i+1, we train a synthesizer by inverting the old model $\theta_i \circ f_i$ through model inversion technique [31] following ABD [25]. **Then**, the model starts to learn new task $\mathcal{T}_{i+1}$ once the synthesizer training is completed. We temporarily keep the snapshot $\theta_i \circ f_i$ (old model) in memory, and add $|\mathcal{T}_{i+1}|$ new linear classifiers (denoted as $\theta^{\mathcal{T}_{i+1}}$) to θ (the original θ is denoted as $\theta^{\mathcal{T}_{1:i}}$). Then, we randomly sample a batch of training data (X^{new}, Y^{new}) from the new training data $\mathcal{D}_{i+1}^{train}$, and synthesize the same number of data (X^{old}, Y^{old}) by the synthesizer for previous classes, which are passed to the model to learn the representations of new classes without forgetting previous classes by integrating the hard knowledge distillation, local classification loss and relational knowledge distillation. **Last**, after representation learning, we freeze the feature extractor f and refine the classification head θ with global class-balanced classification loss to address the data imbalance issue as well as learn the decision boundaries between new and previous classes.

In the following subsections, we will first describe our core contributions of relation-guided representation learning in Sec. 3.2 and classification head refinement in Sec. 3.3, then review the synthesizer training in Sec. 3.4.

3.2 Relation-Guided Representation Learning

Learning new knowledge will inevitably change the current model, causing the forgetting of previously learned classes. Therefore, on the one hand, how to overcome forgetting is essential in DFCIL. To this end, we provide the technique of hard knowledge distillation (HKD). On the other hand, the model should also have the flexibility to learn knowledge from the classes in the new task, for which we adopt local cross-entropy loss (LCE) on data of new classes. However, the conflict between overcoming forgetting by HKD and learning new knowledge by LCE still can not be well resolved, which motivates us to propose relation-guided representation learning (RRL) via relational knowledge distillation (RKD).

Hard Knowledge Distillation (HKD). Prior works usually take the following knowledge distillation method to keep the model from forgetting previous $1:i$ tasks when learning the $i{+}1^{\text{th}}$ task:

$$\mathcal{L}_{kd} = \frac{1}{|X|} \sum_{\mathbf{x} \in X} \mathcal{D}_{KL} \left(\text{softmax} \left(\theta_i \left(f_i \left(\mathbf{x} \right) \right) / \tau \right), \text{softmax} \left(\theta^{\mathcal{T}_{1:i}} \left(f \left(\mathbf{x} \right) \right) / \tau \right) \right), \quad (1)$$

where τ is a temperature parameter, $\mathcal{D}_{KL}$ is KL divergence, and X is one of X^{new}, X^{old} and $X^{new} \cup X^{old}$. However, we find that it is not hard enough when applied to synthetic data. Instead, we use a harder variant of $\mathcal{L}_{kd}$ and **only apply it on synthetic data** without freezing $\theta^{\mathcal{T}_{1:i}}$. We formulate our HKD as:

$$\mathcal{L}_{hkd} = \frac{1}{|X^{old}| \times |\mathcal{T}_{1:i}|} \sum_{\mathbf{x} \in X^{old}} \| \theta_i \left(f_i \left(\mathbf{x} \right) \right) - \theta^{\mathcal{T}_{1:i}} \left(f \left(\mathbf{x} \right) \right) \|_1. \quad (2)$$

With this hard knowledge distillation, the outputs of old model $\theta_i \circ f_i$ and current model $\theta \circ f$ for the synthetic old data tend to be the same, but the model remains flexible inside to adapt to new knowledge. Next, we focus on learning representations of new classes, which requires the model to learn as many features from new task $\mathcal{T}_{i+1}$ as possible.

Local Classification Loss. In CIL, it's common to use *global* cross-entropy as the base loss that is applied on all available training data at the same time. However, when we use synthetic data to replace the real old data in DFCIL, the domain gap between synthetic and real data leads the model to separate new and old classes by the difference of domain rather than semantics, as pointed out in ABD [25]. We also observe that the decision boundaries within synthetic data are different from the ones within the real data. For instance, a synthetic fish image may mix a lot of features of a bird, which might confuse the old classifiers. Therefore, we adopt a *local* classification loss, which is the cross-entropy loss computed on the new data and the new classifiers (*i.e.*, $\theta^{\mathcal{T}_{i+1}}$), formulated as:

$$\mathcal{L}_{lce} = \frac{1}{|X^{new}|} \sum_{(\mathbf{x}, y) \in (X^{new}, Y^{new})} \mathcal{L}_{CE} \left(\text{softmax} \left(\theta^{\mathcal{T}_{i+1}} \left(f \left(\mathbf{x} \right) \right) \right), y \right). \quad (3)$$

This local classification loss does not directly affect classifiers of previous classes $\theta^{\mathcal{T}_{1:i}}$, but it changes f to adapt to new task, which may corrupt the representations of previous learned classes.

The conflict between learning representations of new classes and maintaining representations of previously learned classes can only be mitigated by sacrificing one for the other if the representation learning is not properly guided, and finally they compromise each other to achieve a coarse balance. Therefore, we propose to guide the current model to learn representations of new classes by the structural relation of their data in the old model's feature space.

Relational Knowledge Distillation (RKD). The HKD applied on synthetic data prevents changes in the representation of previous classes, since it strictly forces the representation of a single sample to be consistent on the new and old models. However, it limits the model's plasticity when employed on data of new classes. In contrast to HKD, RKD [17] transfers structural information among a set of samples from teacher model to student model, endowing the student model more flexibility to learn new knowledge. The angle-wise RKD defines the relation on a triplet of samples $(\mathbf{x}_a, \mathbf{x}_b, \mathbf{x}_c)$ as the following cosine value:

$$\cos\angle \mathbf{r}_a\mathbf{r}_b\mathbf{r}_c = \left\langle \mathbf{e}^{ab}, \mathbf{e}^{cb} \right\rangle \quad \text{where} \quad \mathbf{e}^{ij} = \frac{\mathbf{r}_i - \mathbf{r}_j}{\left\|\mathbf{r}_i - \mathbf{r}_j\right\|_2}. \tag{4}$$

Here $\mathbf{r}_*$ is sample $\mathbf{x}_*$'s feature representation on the teacher or student model.

We incorporate RKD into our DFCIL framework, transferring the *structural information of the new data* in the feature space of *the old model* $\theta_i \circ f_i$ to current model $\theta \circ f$. Therefore, it can build a bridge between learning representations of new classes and maintaining representations of previously learned classes.

When applying RKD, instead of directly using the data representations from the model to construct the relation, we first transform the representations via a $d \times 2d$ linear layer denoted as ϕ, considering the following. The new classes may not be effectively distinguished by their representations on old model $\theta_i \circ f_i$, and therefore, the structural relation built directly from the old representations may not help improving model's plasticity. The representations of the new classes on current model $\theta \circ f$ are transformed by another linear layer ψ to align to the transformed old representations. Then, we apply the following angle-wise relational knowledge distillation to a triplet $(\mathbf{x}_a, \mathbf{x}_b, \mathbf{x}_c)$ of the new data:

$$\begin{aligned} \mathcal{L}_{rkd} &= \frac{1}{|X^{new}|^3} \sum_{\mathbf{x}_a, \mathbf{x}_b, \mathbf{x}_c \in X^{new}} \left\| \cos\angle \mathbf{t}_a\mathbf{t}_b\mathbf{t}_c - \cos\angle \mathbf{s}_a\mathbf{s}_b\mathbf{s}_c \right\|_1, \\ &\text{where} \quad \mathbf{t}_k = \phi\left(f_i\left(\mathbf{x}_k\right)\right), \mathbf{s}_k = \psi\left(f\left(\mathbf{x}_k\right)\right). \end{aligned} \tag{5}$$

By minimizing this loss, we limit the cases when the relation built from the transformed old representations hinders the improvement of plasticity or when the representation change hurts the model's stability. The two learnable transformation functions ϕ and ψ are optimized with the representation learning to minimize this loss, making the relation distillation flexible. Using this relational knowledge distillation, we mitigate the conflict between improving plasticity by local classification loss and maintaining stability by hard knowledge distillation.

RRL Loss. The above three components form the relation-guided representation learning (RRL). The loss of the RRL at phase $i{+}1$ is formulated as:

$$\mathcal{L}_{rrl} = \lambda_{lce}^{i+1}\mathcal{L}_{lce} + \lambda_{hkd}^{i+1}\mathcal{L}_{hkd} + \lambda_{rkd}^{i+1}\mathcal{L}_{rkd}, \tag{6}$$

where lambdas are corresponding scale factors. Considering the amount of new knowledge increases with the number of new classes, and the difficulty of preserving previous knowledge grows as the ratio of previous classes to new classes gets larger, scale factors at learning phase $i{+}1$ are adaptively set as follows:

$$\lambda_{lce}^{i+1} = \frac{1+1/\alpha}{\beta}\lambda_{lce}, \quad \lambda_{hkd}^{i+1} = \alpha\beta\lambda_{hkd}, \quad \lambda_{rkd}^{i+1} = \alpha\beta\lambda_{rkd} \tag{7}$$

$$\text{where} \quad \alpha = \log_2(\frac{|\mathcal{T}_{i+1}|}{2}+1), \quad \beta = \sqrt{\frac{|\mathcal{T}_{1:i}|}{|\mathcal{T}_{i+1}|}}, \tag{8}$$

in which λ_{lce}, λ_{hkd} and λ_{rkd} are base scale factors that can be configurable, α and β denote the amount of new knowledge and the difficulty of preserving previous knowledge, respectively. We appropriately increase the local classification loss to compensate for its weakening as the number of new classes decreases. The overall loss of the RRL at learning phase $i{+}1$ is finally defined as:

$$\mathcal{L}_{rrl} = \frac{1+1/\alpha}{\beta}\lambda_{lce}\mathcal{L}_{lce} + \alpha\beta\lambda_{hkd}\mathcal{L}_{hkd} + \alpha\beta\lambda_{rkd}\mathcal{L}_{rkd}. \tag{9}$$

3.3 Classification Head Refinement

We achieve better stability-plasticity balance in feature extractor by relation-guided representation learning, but there are still two issues in classification head to address. One is that the decision boundaries between new and previous classes have not been learned by the model, and the other is that the imbalanced training data may cause biased classifiers. ABD attacks these problems concurrently with representation learning by a global task-balanced classification loss [24,12,29]. However, we find that the global classification loss is not beneficial to the representation learning due to the domain gap between synthetic and real data. In addition to the data imbalance between new and previous classes, the data imbalance also exists within previous classes because the label of synthetic images are random. Inspired by prior works [4,29], we fine-tune the classification head with *the feature extractor frozen* after representation learning, in which the $\mathcal{L}_{lce}$ is replaced with the following global class-balanced classification loss:

$$\mathcal{L}_{gce} = \frac{1}{|X|}\sum_{(\mathbf{x},y)\in(X,Y)}\frac{w_y}{\sum_{j=0}^{|\mathcal{T}_{1:i+1}|-1}w_j}\mathcal{L}_{CE}\left(\text{softmax}\left(\theta\left(f\left(\mathbf{x}\right)\right)\right),y\right), \tag{10}$$
$$\text{where} \quad (X,Y) = (X^{new}\cup X^{old}, Y^{new}\cup Y^{old}).$$

The weight w_y of class y is the reciprocal of it's number of samples (*i.e.*, synthetic for previous classes and real for new classes) passed to the model during training.

3.4 Image Synthesis

The model inversion technique was first introduced to DFCIL in DeepInversion [31] to synthesize data for previous classes. DeepInversion iteratively optimizes random noises to images of given classes together with training the classification model, which is time consuming. Instead, ABD [25] trains a synthesizer

before learning new task, speeding up the learning process. Therefore, we follow ABD [25] to train our synthesizer using the following four optimization objectives. **The label diversity loss** forces the synthesizer to produce balanced data for previous classes. **The data content loss** is the cross-entropy loss with a large temperature parameter to scale down the difference between the model's output, so that the synthetic images can be predicted as a certain class with high confidence. **The stat alignment loss** minimizes the KL divergence between the distribution of synthetic data and the distribution in BatchNorm layers of f_i, which record the statistics of the real data during the previous training. **The image prior loss** encourages the synthesizer to produce more realistic images.

By this means, we can obtain synthetic data that mimic the old real data. However, there are still the following issues with the synthetic data, and different techniques of our R-DFCIL addresses these issues accordingly. **1) Class imbalance** is attacked by class-balanced classification loss defined in Sec. 3.3. **2) The domain gap** between synthetic and real data, which misleads classifiers to learn wrong decision boundaries between new and previous classes, is attacked by separating the learning process into representation learning (Sec. 3.2) and classifier learning (Sec. 3.3). **3) The conflict between model's plasticity and stability** is alleviated by relational knowledge distillation (Sec. 3.2), and catastrophic forgetting is effectively overcome by hard knowledge distillation.

4 Experiment

4.1 Datasets and Evaluation Protocol

Datasets. We chose three representative classification datasets CIFAR100 [10], Tiny-ImageNet200 [11] and ImageNet100 [8], in which CIFAR100 and ImageNet100 are two extensively used datasets in CIL, and Tiny-ImageNet200 is considered as a challenging dataset for DFCIL [25]. CIFAR100 contains 100 classes, each class with 500 training images of size 32×32×3 and 100 test images in the same size. ImageNet100 is a subset of ImageNet1000 [23], with 100 randomly sampled classes. It has about 1300 training and 50 test images per class, and the spatial size of images vary. Tiny-ImageNet200 is an ImageNet-like dataset with smaller (64×64×3) images than ImageNet. It has 200 classes in total, with 500 training and 50 test images for each class.

Evaluation Protocol. In the CIL literature, there are two commonly used protocols. The first protocol splits the classes equally into $N = 5, 10, 20$ tasks for simulating short-term and large task incremental learning scenarios, in which $|\mathcal{T}_{1:i}|/|\mathcal{T}_{i+1}|$ is relatively small and the number of classes per task are relatively large. The other protocol introduced by Hou *et al.* [8] takes a half of classes as the first task, and equally divides the rest classes into 5, 10 or 25 tasks (*i.e.*, $N = 6, 11, 26$), which matches the situation of long-term and small task incremental learning. We follow prior works [20,8,4,25] to evaluate approaches by the typical incremental metrics: last incremental accuracy A_N and average incremental accuracy $\bar{A}_N = \frac{1}{N}\sum_{i=1}^{N} A_i$, in which the incremental accuracy A_i

Table 1: Evaluation on CIFAR100 with protocol that equally split 100 classes into N tasks. The means and standard deviations are reported of three runs with random class orders. Approaches with * are reported directly from ABD paper.

Approach	$N = 5$	10	20
	A_N (%)	A_N (%)	A_N (%)
Upper Bound	70.67 ± 0.16	70.67 ± 0.16	70.67 ± 0.16
DGR* [24]	14.40 ± 0.40	8.10 ± 0.10	4.10 ± 0.30
LwF* [14]	17.00 ± 0.10	9.20 ± 0.00	4.70 ± 0.10
DeepInversion* [31]	18.80 ± 0.30	10.90 ± 0.60	5.70 ± 0.30
ABD* [25]	43.90 ± 0.90	33.70 ± 1.20	20.00 ± 1.40
ABD [25]	47.36 ± 0.48	36.19 ± 0.93	22.29 ± 0.65
R-DFCIL (Ours)	**50.47 ± 0.43**	**42.37 ± 0.72**	**30.75 ± 0.12**
	$\bar{A}_N$ (%)	$\bar{A}_N$ (%)	$\bar{A}_N$ (%)
ABD [25]	63.23 ± 1.49	56.61 ± 1.93	45.10 ± 2.01
R-DFCIL (Ours)	**64.85 ± 1.78**	**59.41 ± 1.76**	**48.47 ± 1.90**

is formally defined as:

$$A_i = \frac{1}{|\mathcal{D}_{1:i}^{test}|} \sum_{(\mathbf{x},y)\in\mathcal{D}_{1:i}^{test}} \mathbb{1}\left(\hat{y} = y\right), \text{ where } \hat{y} = \underset{0\le j<|\mathcal{T}_{1:i}|}{\arg\max}\, \theta_i^{(j)}(f_i(\mathbf{x})), \tag{11}$$

in which $\mathbb{1}(\cdot)$ is the indicator function that maps the boolean value to $\{0, 1\}$.

4.2 Implementation Details

All approaches are implemented within the same code base written in PyTorch. We reproduce the current SOTA DFCIL approach ABD [25], and two popular replay-based CIL approaches UCIR [8], PODNet [4]. To fairly comparision, we implement the UCIR-DF and PODNet-DF by replacing the real old data with the synthetic data that used by ABD and our R-DFCIL. For CIFAR100, we follow the prior works [20,8,4] to adopt a modified 32-layer ResNet [6] backbone and train the model with SGD optimizer for 160 epochs, the learning rate is initially set to 0.1 and is divided by 10 after 80 and 120 epochs, the weight decay is set to 0.0005 and batch size is 128. We change the weight decay to 0.0002 for Tiny-ImageNet200 and keep other settings same as CIFAR100. For ImageNet100, we employ a ResNet18 [6] backbone and train the model with SGD optimizer for 90 epochs, the learning rate starts from 0.1 and is divided by 10 after 30 and 60 epochs, the weight decay is set to 0.0001 and batch size is 64. Our R-DFCIL fine-tunes the classification head with a small constant learning rate 0.005 for another 40 epochs for CIFAR100, Tiny-ImageNet200, and 30 epochs for ImageNet100. The hyper parameters of our R-DFCIL are set to $\lambda_{lce} = 0.5$, $\lambda_{hkd} = 0.15$, $\lambda_{rkd} = 0.5$ in all experiments. Please see supplementary material for more details on hyper-parameter tuning.

Table 2: Evaluation on CIFAR100 with the protocol introduced by Hou *et al.* [8]. The results of UCIR, PODNet and their Data-Free implementation UCIR-DF, PODNet-DF (all with CNN classifier) are present here for clearly comparison.

Approach	Data Free	$N = 6$	11	26
		A_N (%)	A_N (%)	A_N (%)
UCIR (CNN) [8]	✗	55.73 ± 0.89	53.22 ± 0.71	50.08 ± 0.35
PODNet (CNN) [4]	✗	56.19 ± 1.00	52.53 ± 0.55	49.14 ± 0.25
UCIR-DF (CNN) [8]	✓	39.49 ± 0.81	25.54 ± 1.51	9.62 ± 0.73
PODNet-DF (CNN) [4]	✓	40.54 ± 1.68	33.57 ± 2.48	20.18 ± 0.76
ABD [25]	✓	50.55 ± 1.14	43.65 ± 2.40	25.27 ± 1.09
R-DFCIL (Ours)	✓	**54.76 ± 0.76**	**49.70 ± 0.61**	**30.01 ± 0.56**
		$\bar{A}_N$ (%)	$\bar{A}_N$ (%)	$\bar{A}_N$ (%)
UCIR (CNN) [8]	✗	65.58 ± 1.00	63.54 ± 1.12	60.32 ± 1.09
PODNet (CNN) [4]	✗	66.82 ± 1.25	63.91 ± 1.07	61.56 ± 1.02
UCIR-DF (CNN) [8]	✓	57.82 ± 0.86	48.69 ± 1.16	33.33 ± 1.18
PODNet-DF (CNN) [4]	✓	56.85 ± 1.40	52.61 ± 1.72	43.23 ± 1.70
ABD [25]	✓	62.40 ± 1.17	58.97 ± 1.87	48.91 ± 1.88
R-DFCIL (Ours)	✓	**64.78 ± 1.58**	**61.71 ± 1.17**	**49.95 ± 0.76**

4.3 Results and Analysis

CIFAR100. We follow ABD [25] to conduct five-, ten-, and twenty-tasks class-incremental experiments, with respectively 20, 10 and 5 classes per task. We run all approaches on three random class orders with the seeds 0, 1, 2 (*i.e.*, consistent with the official ABD code) and report the means and standard deviations of these three runs. In Table 1, we report the results of ABD implemented by us and present the original data reported by ABD paper. Our R-DFCIL surpasses ABD by 3.11/1.62 ($A_N/\bar{A}_N$), 6.18/2.80 and 8.46/3.37 percent points on five-, ten-, and twenty-tasks settings, respectively. Table 2 shows results of the experiments with the protocol introduced by Hou *et al.* [8], in which the first task has 50 classes and 10, 5, 2 classes per incremental task for $N = 6, 11, 26$, respectively. From the comparison between UCIR/PODNet and UCIR-DF/PODNet-DF, we can see a great performance degradation of the popular replay-based approaches when replacing the real old data with synthetic old data. Prior CIL works believe that more tasks imply stronger forgetting. But we find that the initially learned knowledge and the number of classes in incremental tasks also impact forgetting, since both ABD and our R-DFCIL perform better with the second protocol than with the first protocol despite more tasks (Table 2 *vs.* 1).

Tiny-ImageNet200. We compare our R-DFCIL with ABD in the more challenging dataset Tiny-ImageNet200, in which we can observe similar results to the experiments on CIFAR100. From the data presented in Table 3, 4, we can see that there are more performance gains of our R-DFCIL over ABD as the total number of tasks increases (*e.g.*, $N = 5 \rightarrow 20$, $6 \rightarrow 26$). We plot the task-by-task incremental accuracy in Fig. 2, in which we can see the ABD drops faster than

Table 3: Evaluation on Tiny-ImageNet200 with the protocol that equally divides classes into N tasks. The means and standard deviations are reported of three runs with random class orders. ABD* indicates data reported from ABD paper.

Approach	$N = 5$	10	20
	A_N (%)	A_N (%)	A_N (%)
Upper Bound	55.39 ± 0.33	55.39 ± 0.33	55.39 ± 0.33
ABD* [25]	-	-	12.1
ABD [25]	30.56 ± 0.22	22.87 ± 0.67	15.20 ± 1.01
R-DFCIL (Ours)	**35.89 ± 0.75**	**29.58 ± 0.51**	**24.43 ± 0.82**
	$\bar{A}_N$ (%)	$\bar{A}_N$ (%)	$\bar{A}_N$ (%)
ABD [25]	45.30 ± 0.50	41.05 ± 0.54	34.74 ± 0.91
R-DFCIL (Ours)	**48.96 ± 0.40**	**44.36 ± 0.18**	**39.34 ± 0.18**

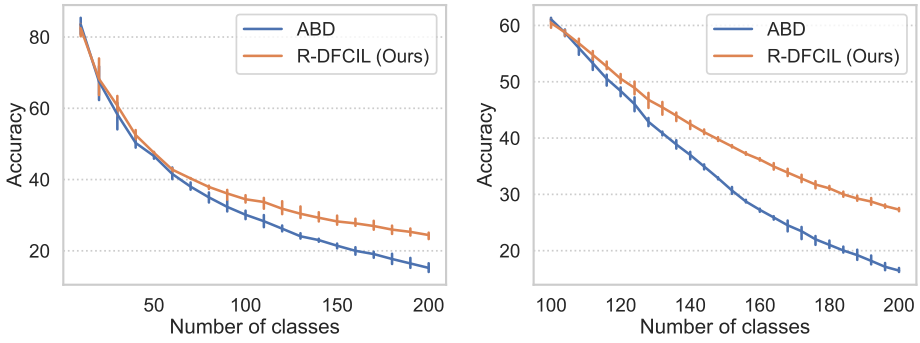
Table 4: Evaluation on Tiny-ImageNet200 with the protocol introduced by Hou *et al*. [8]. The means and standard deviations are reported of three runs with random class orders. The best values are in bold font.

Approach	$N = 6$	11	26
	A_N (%)	A_N (%)	A_N (%)
ABD [25]	33.18 ± 0.46	27.34 ± 0.44	16.46 ± 0.34
R-DFCIL (Ours)	**40.44 ± 0.11**	**38.19 ± 0.08**	**27.29 ± 0.24**
	$\bar{A}_N$ (%)	$\bar{A}_N$ (%)	$\bar{A}_N$ (%)
ABD [25]	44.55 ± 0.13	41.64 ± 0.46	34.47 ± 0.29
R-DFCIL (Ours)	**48.91 ± 0.29**	**47.60 ± 0.50**	**40.85 ± 0.28**

our R-DFCIL as the number of learned classes increases. We can conclude from the above observations that our R-DFCIL solves the forgetting of previously learned classes better than ABD.

ImageNet100. We report the experimental results on ImageNet100 in Table 5. In these experiments, the model is less prone to forgetting than experiments on CIFAR100 and Tiny-ImageNet200 due to the large model capacity (11.0 *vs*. 0.4 million parameters). Although the performance of ABD is close to our R-DFCIL when N=5, the difference becomes significant when N increases to 20.

Ablation Study. We ablate three main components of our R-DFCIL, and display the results in Table 6. The experiments are conducted on CIFAR100 with total N=20 tasks and 5 classes per task. All three components contribute greatly to our R-DFCIL, the last incremental accuracy drops by 9.21, 25.38, 7.00 percent point without relational knowledge distillation (RKD), hard knowledge distillation (HKD), classification head refinement (CHR), respectively. From Fig. 3, we can clearly see that the HKD is necessary for reducing the forgetting of learned



(a) 20 tasks, 10 classes / incremental task (b) 26 tasks, 4 classes / incremental task

Fig. 2: **Incremental Accuracy on Tiny-ImageNet200**. The lines show the phase-by-phase evaluation results of ABD [25] and our F-DFCIL. The means and standard deviations are reported of three runs with random class orders.

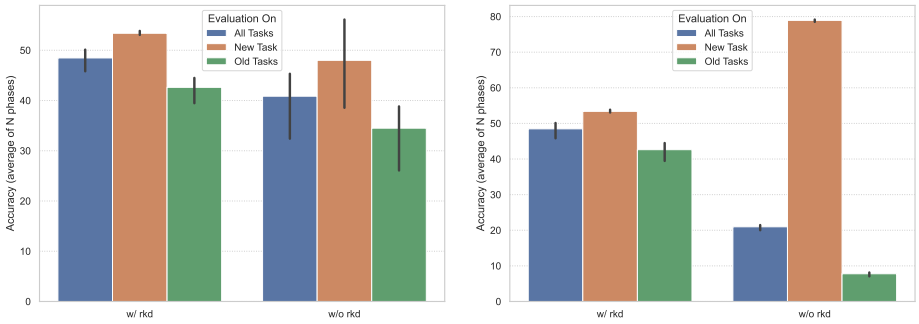
Table 5: Evaluation on ImageNet100 with the protocol that equally split 100 classes into N tasks. We report the evaluation results of a single run.

Approach	$N = 5$	10	20
	A_N (%)	A_N (%)	A_N (%)
Upper Bound	77.46	77.46	77.46
ABD [25]	51.46	35.96	22.40
R-DFCIL (Ours)	**53.10**	**42.28**	**30.28**
	$\bar{A}_N$ (%)	$\bar{A}_N$ (%)	$\bar{A}_N$ (%)
ABD [25]	67.42	57.76	44.89
R-DFCIL (Ours)	**68.15**	**59.10**	**47.33**

classes. We also observe that the RKD boost both plasticity and stability, demonstrating the success of our relation-guided representation learning in alleviating the conflict between improving plasticity and maintaining stability. In fact, our R-DFCIL achieves better plasticity as well as stability than previous approaches, the details are present in supplementary material. It is worth emphasizing that our adaptive design (*i.e.*, introduction of linear transformation functions) contributes about 2% gain in the last incremental accuracy. We also investigated some newer relational KD methods, please see supplementary material.

5 Conclusion

This paper studies the problem of Data-Free Class-Incremental Learning (DFCIL). We propose relation-guided representation learning (RRL) for DFCIL (R-DFCIL) to address the catastrophic forgetting caused by the severe domain gap



(a) relational knowledge distillation (b) hard knowledge distillation

Fig. 3: **Ablation Study about Stability-Plasticity Balance**. The left (a) shows a better balance with RKD (w/ rkd), and the right show the importance of the HKD to mitigate forgetting.

Table 6: Abalation Study on CIFAR100 with $N = 20$. The results show the comparison between our R-DFCIL with all components and without relation knowledege distillation (RKD), hard knowledge distillation (HKD), classification head refinement (CHR, *i.e.*, training process ends with representation learning).

RKD	HKD	CHR	A_N (%)	$\bar{A}_N$ (%)
✗	✓	✓	21.63 ± 5.60	40.86 ± 5.98
✓	✗	✓	5.37 ± 0.35	20.96 ± 0.69
✓	✓	✗	23.75 ± 0.81	43.09 ± 1.53
✓	✓	✓	**30.75 ± 0.12**	**48.47 ± 1.90**

between synthetic and real data. In RRL, the model overcomes forgetting of previous classes by hard knowledge distillation on synthetic data, and learns new knowledge by the local classification loss on new data. The relational knowledge distillation (RKD) can mitigate the conflict between improving plasticity and maintaining stability by transferring structural relation of new data from the old to the current model. After RRL, the classification head is refined with global class-balanced classification loss to address data imbalance issue and learn the decision boundaries between classes. Our R-DFCIL surpasses previous SOTA approach on CIFAR100, Tiny-ImageNet200 and ImageNet100 with 8.46%, 9.23%, and 9.88% accuracy gain, respectively. Our R-DFCIL learns representation and classifier independently in two stages, which constructs a basic framework for future studies to address the domain gap between synthetic and real data in DFCIL. We introduce RKD to DFCIL for the first time, providing a reference for future works to overcome forgetting using structural information.

Acknowledgement This work was supported by the King Abdullah University of Science and Technology (KAUST) Office of Sponsored Research (OSR) under Award No. OSR-CRG2021-4648 and the Shenzhen General Research Project (JCYJ20190808182805919).

References

1. Bang, J., Kim, H., Yoo, Y., Ha, J., Choi, J.: Rainbow memory: Continual learning with a memory of diverse samples. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2021)
2. Castro, F.M., Marín-Jiménez, M.J., Guil, N., Schmid, C., Alahari, K.: End-to-end incremental learning. In: Proceedings of the European Conference on Computer Vision (ECCV) (2018)
3. Cong, Y., Zhao, M., Li, J., Wang, S., Carin, L.: GAN memory with no forgetting. In: Proceedings of the Advances in Neural Information Processing Systems (NeurIPS) (2020)
4. 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)
5. Goodfellow, I.J., Mirza, M., Xiao, D., Courville, A., Bengio, Y.: An empirical investigation of catastrophic forgetting in gradient-based neural networks. arXiv preprint arXiv:1312.6211 (2013)
6. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2016)
7. Hinton, G., Vinyals, O., Dean, J.: Distilling the knowledge in a neural network. In: Proceedings of the Advances in Neural Information Processing Systems (NeurIPS), Deep Learning and Representation Learning Workshop (2015)
8. 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)
9. Kemker, R., Kanan, C.: Fearnet: Brain-inspired model for incremental learning. In: Proceedings of the International Conference on Learning Representations (ICLR) (2018)
10. Krizhevsky, A., Hinton, G., et al.: Learning multiple layers of features from tiny images. Technical Report (2009)
11. Le, Y., Yang, X.: Tiny imagenet visual recognition challenge. CS 231N (2015)
12. Lee, K., Lee, K., Shin, J., Lee, H.: Overcoming catastrophic forgetting with unlabeled data in the wild. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV) (2019)
13. Lee, S.H., Kim, D.H., Song, B.C.: Self-supervised knowledge distillation using singular value decomposition. In: Proceedings of the European Conference on Computer Vision (ECCV) (2018)
14. Li, Z., Hoiem, D.: Learning without forgetting. IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI) (2017)
15. Liu, Y., Schiele, B., Sun, Q.: Adaptive aggregation networks for class-incremental learning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2021)
16. Nagarajan, V., Raffel, C., Goodfellow, I.J.: Theoretical insights into memorization in gans. In: Proceedings of the Advances in Neural Information Processing Systems (NeurIPS) Workshop (2018)
17. Park, W., Kim, D., Lu, Y., Cho, M.: Relational knowledge distillation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2019)

18. Passalis, N., Tefas, A.: Learning deep representations with probabilistic knowledge transfer. In: Proceedings of the European Conference on Computer Vision (ECCV) (2018)
19. Prabhu, A., Torr, P.H.S., Dokania, P.K.: Gdumb: A simple approach that questions our progress in continual learning. In: Proceedings of the European Conference on Computer Vision (ECCV) (2020)
20. Rebuffi, S.A., Kolesnikov, A., Sperl, G., Lampert, C.H.: icarl: Incremental classifier and representation learning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2017)
21. Robins, A.V.: Catastrophic forgetting, rehearsal and pseudorehearsal. Connect. Sci. (1995)
22. Romero, A., Ballas, N., Kahou, S.E., Chassang, A., Gatta, C., Bengio, Y.: Fitnets: Hints for thin deep nets. In: Proceedings of the International Conference on Learning Representations (ICLR) (2015)
23. Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., et al.: Imagenet large scale visual recognition challenge. International Journal of Computer Vision (IJCV) (2015)
24. Shin, H., Lee, J.K., Kim, J., Kim, J.: Continual learning with deep generative replay. In: Guyon, I., von Luxburg, U., Bengio, S., Wallach, H.M., Fergus, R., Vishwanathan, S.V.N., Garnett, R. (eds.) Proceedings of the Advances in Neural Information Processing Systems (NeurIPS) (2017)
25. 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)
26. Srinivas, S., Fleuret, F.: Knowledge transfer with jacobian matching. In: Proceedings of the International Conference on Machine Learning (ICML) (2018)
27. van de Ven, G.M., Siegelmann, H.T., Tolias, A.S.: Brain-inspired replay for continual learning with artificial neural networks. Nature communications (2020)
28. Wu, C., Herranz, L., Liu, X., van de Weijer, J., Raducanu, B., et al.: Memory replay gans: Learning to generate new categories without forgetting. In: Proceedings of the Advances in Neural Information Processing Systems (NeurIPS) (2018)
29. Wu, Y., Chen, Y., Wang, L., Ye, Y., Liu, Z., Guo, Y., Fu, Y.: Large scale incremental learning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2019)
30. Ye, F., Bors, A.G.: Learning latent representations across multiple data domains using lifelong VAEGAN. In: Proceedings of the European Conference on Computer Vision (ECCV) (2020)
31. Yin, H., Molchanov, P., Alvarez, J.M., Li, Z., Mallya, A., Hoiem, D., Jha, N.K., Kautz, J.: Dreaming to distill: Data-free knowledge transfer via deepinversion. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2020)
32. Yu, L., Twardowski, B., Liu, X., Herranz, L., Wang, K., Cheng, Y., Jui, S., van de Weijer, J.: Semantic drift compensation for class-incremental learning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2020)