Dual-consistency Model Inversion for Non-exemplar Class Incremental Learning

Zihuan Qiu[1] Yi Xu[2] Fanman Meng[1*] Hongliang Li[1] Linfeng Xu[1] Qingbo Wu[1]

[1]University of Electronic Science and Technology of China [2]Dalian University of Technology

{zihuanqiu@std.,fmmeng@,hlli@,lfxu@,qbwu@}uestc.edu.cn yxu@dlut.edu.cn

Abstract

Non-exemplar class incremental learning (NECIL) aims to continuously assimilate new knowledge without forgetting previously acquired ones when historical data are unavailable. One of the generative NECIL methods is to invert the images of old classes for joint training. However, these synthetic images suffer significant domain shifts compared with real data, hampering the recognition of old classes. In this paper, we present a novel method termed Dual-Consistency Model Inversion (DCMI) to generate better synthetic samples of old classes through two pivotal consistency alignments: (1) the semantic consistency between the synthetic images and the corresponding prototypes, and (2) domain consistency between synthetic and real images of new classes. Besides, we introduce Prototypical Routing (PR) to provide task-prior information and generate unbiased and accurate predictions. Our comprehensive experiments across diverse datasets consistently showcase the superiority of our method over previous state-of-the-art approaches.

1. Introduction

When applying a trained deep neural classification network to new classes, the strategy of fine-tuning on new images often leads to catastrophic forgetting of old classes [23]. Class incremental learning (CIL) aims to continuously adapt to new classes without forgetting the learned ones. Exemplar-based methods [2, 10, 19, 41, 48] have shown promise by storing a subset of old class data as exemplars and retraining with new class data in the future. While effective, storing exemplars can be challenging in practice, due to concerns about data privacy or limited storage space. Non-exemplar class incremental learning (NECIL) [23, 30, 50], also known as exemplar-free CIL, has gained increasing attention recently. Compared to their exemplar-based counterparts, it offers advantages in training and storage efficiency, as well as addressing concerns related to data privacy and copyright [42, 56]. Meanwhile, it is more challenging to train the model in the absence of exemplars continuously.

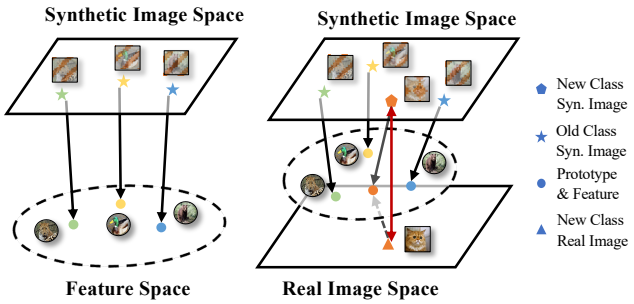


Figure 1. Left: Previous methods align synthetic old class images with their prototypes (black arrow) for semantic consistency but ignore domain consistency. Right: Our method ensures both semantic and domain consistency. 1) Real new class images are projected into the old feature space (black dotted arrow) and undergo semantic alignment in the feature space for both new and old classes (black arrow). 2) The domain gap between synthetic and real image space is minimized by matching real and synthetic pairs of new classes (red arrow).

Existing non-exemplar methods mainly employ knowledge distillation (KD) [18] to prevent the updating model, referred to as the 'new model', from forgetting the learned knowledge by enforcing its output to be consistent with the old model. Nevertheless, distilling only with new class samples weakens the effectiveness of KD due to the significant distribution discrepancy between old and new classes, resulting in cumulative errors. To address this problem, generative-based methods [13, 42] attempt to synthesize samples of old classes using model inversion [4, 32, 36], and subsequently distill the new model using both the real and synthetic samples. The synthetic samples, which resemble the semantic distribution of the old classes, are generated by yielding high-confidence classification probabilities from the old model. It is noteworthy that, unlike conventional generative methods that train a generator in advance, inversion methods do not rely on historical data of old classes, making them more practical for NECIL.

*Corresponding author

Nonetheless, a significant domain gap still exists between the generated images and the real ones, as probability maximization primarily produces abstract patterns that differ significantly from real images in terms of visual and content features [15]. Without surprise, distilling with such synthetic samples fails to provide a promising guarantee for the performance of old classes. Therefore, the desirable synthetic samples should not only possess semantic consistency but also share the same domain information as real images.

In fact, under the CIL scenario, new class samples should share the same domain distribution as the old ones, which inspires us to leverage new class data to authenticate the domain of synthetic samples. To this end, we employ adversarial learning [12, 14] to align the underlying distributions of the synthetic and real data. *i.e.*, a discriminator aims to distinguish distribution discrepancy, while the generator strives to deceive the discriminator. By minimizing the discrepancy between the two distributions, we can align the synthetic distribution of old classes, denoted as $P_{(syn,old)}$, with the real distribution of new classes $P_{(real,new)}$. However, directly aligning these distributions does not guarantee domain consistency due to the distinct semantic shift between P_{old} and P_{new} [33]. Additionally, it is important to guard against overfitting the generated data, as it can lead to incorrect decision boundaries and a significant bias towards recent tasks in joint predictions [42].

In this paper, we propose a novel approach for class incremental learning, named Dual-Consistency Model Inversion (DCMI). The goal of DCMI is to generate consistent old class samples in terms of semantics and domain. On the one hand, DCMI ensures semantic consistency by aligning the features of synthetic samples to the corresponding prototypes in the feature space. On the other hand, DCMI maintains domain consistency by disentangling the domain distribution P_{real} from $P_{(real,new)}$, and supervises the generation of $P_{(syn,old)}$ to approximate $P_{(real,old)}$. The concept of our method is illustrated on the right side of Fig. 1. To address the task-recency bias, we draw inspiration from the unbiased nature of prototype matching [44, 50] and propose Prototypical Routing (PR) as a complement to the strong intra-task discriminative power of the linear classifier. We assess the effectiveness of our method through extensive experiments on various datasets, including CIFAR-100 [24], Tiny-ImageNet [25], ImageNet-Subset [9], and ImageNet-Full [9]. The results demonstrate the substantial superiority of DCMI over the previous state-of-the-art approaches. Our main contributions are as follows:

1) We propose a novel approach for class incremental learning termed dual-consistency model inversion. This approach allows for synthesizing old class samples with both semantic and domain consistency.

2) To mitigate the issue of task-recency bias in predictions, we introduce prototypical routing, offering a robust task-prior that guides predictions across tasks.

3) Extensive experiments on CIFAR-100, Tiny-ImageNet, ImageNet-Subset, and ImageNet-Full datasets demonstrate the superiority of our method over the non-exemplar state-of-the-art.

2. Related Work

Class Incremental Learning. Exemplar-based methods store representative data as exemplars for future training. iCaRL [41] first propose to save exemplars in CIL, and the rehearsal strategy has since become a common practice in subsequent works. Knowledge distillation (KD) is widely used to prevent forgetting. UCIR [19] introduces a less-forget constraint on feature distribution, and PODNet [10] employs KD for intermediate features. ISM-Net [40] introduces model queue distillation to enhance long-range performance. Addressing the task-recency bias as a key challenge in CIL, EEIL [2] propose an additional balanced fine-tuning session, and UCIR replaces the softmax layer with cosine normalization. More recently, some works [45, 52] aim to improve CIL performance through model expansion. Despite exemplar-based methods saving only a subset of old data, concerns about privacy risks and storage needs persist. Non-exemplar methods have recently gained attention. Some methods [23, 47, 51] propose parameter importance estimation to prevent significant changes in critical parameters, while LwF [27] introduces KD to constraint the output logits, where the previous model serves as a teacher. ABD [42] introduces model inversion [4, 32, 36] to improve the effectiveness of KD by synthesizing old class data. PASS [55] explores self-supervised learning in CIL and proposes prototype augmentation for classifier learning. SSRE [56] introduces a self-sustaining expansion scheme and prototype selection. FeTrIL [39] proposes a feature translation technique.

Model Inversion. MI is a vital technique for Data-Free Knowledge Distillation (DFKD), transferring knowledge from a trained teacher model to a compact student model when original training data is unavailable. DFKD finds wide applications in solving problems like model compression [8, 31], transfer learning [4], and incremental learning [42, 49]. Trained models encapsulate data information, allowing reverse generation through noise optimization or a generator. Lopes *et al.* [32] pioneer DFKD using activation summaries, while Nayak *et al.* [36] optimize noise for softmax-like output calculated by class similarity. Chen *et al.* [4] integrate generative adversarial networks for efficiency, and Bhardwaj *et al.* [1] store activation statistics as metadata. DeepDream [35] introduces an image prior term, and DeepInversion [49] focuses on minimizing the distance between feature maps and batch normalization statistics.

Knowledge Distillation. KD was first proposed by Hinton *et al.* [18] for model compression and transfer learning. This technique involves transferring knowledge from complex teacher models to lightweight student models by having the students mimic the outputs of their teachers. Knowledge distillation methods can be broadly categorized into logit-based distillation [7, 11, 18, 34, 46], feature-based distillation [16, 17, 21, 37, 38], and relation-based distillation [6, 28, 29, 37]. Logit-based distillation aims to convey implicit information present within label distribution, while feature-based distillation seeks guidance from intermediate features to facilitate student learning. Relation-based distillation establishes structural relationships among samples or contextual relations to guide the student network.

The most related work to ours is ABD [42], which identifies a significant domain shift that adversely affects performance when utilizing synthetic images. They address this issue by introducing a local CE loss and importance-weighted feature distillation. In our work, we enhance the domain consistency of synthetic images by incorporating authentication from new class data—a dimension overlooked in prior studies.

3. Motivation

In CIL, training data flows in separate tasks, with each task holding disjoint classes $\mathcal{C}_t (0 \leq t \leq N)$. Let $\mathcal{D}_n$ and $\mathcal{D}_o$ denote the underlying distribution of new and old tasks, respectively. The goal of CIL is to learn a model with low disagreement between the hypothesis (h_n and h_o) and the labeling function (f_n and f_o) for both new and old classes:

$$\epsilon_{\mathcal{D}_n}(h_n, f_n) + \epsilon_{\mathcal{D}_o}(h_n, f_o) = \mathrm{E}_{\mathbf{x}\sim\mathcal{D}_n}[|h_n(\mathbf{x}) - f_n(\mathbf{x})|] + \mathrm{E}_{\mathbf{x}\sim\mathcal{D}_o}[|h_n(\mathbf{x}) - f_o(\mathbf{x})|]. \quad (1)$$

The new task error term $\epsilon_{\mathcal{D}_n}(h_n, f_n)$ can be minimized by using a typical classification loss on the new task data. However, minimizing $\epsilon_{\mathcal{D}_o}(h_n, f_o)$ is challenging because the old class distribution $\mathcal{D}_o$ as well as old class labels f_o are inaccessible when learning new classes. To solve this, knowledge distillation (KD) is commonly used in CIL, which ensures that h_n stays close to the previous state h_o (*i.e.*, $\epsilon_{\mathcal{D}_o}(h_o, f_o)$ is small) while learning $\mathcal{D}_n$. Here, we provide a comprehensive understanding of applying KD in CIL by deriving a bound on the error of new hypotheses for seen classes, considering the error of old hypotheses for old classes, the error of new hypotheses for new classes, the discrepancy between new and old hypotheses, and the variation divergence between distillation data and old class data:

$$\epsilon_{\mathcal{D}_n}(h_n, f_n) + \epsilon_{\mathcal{D}_o}(h_n, f_o) \leq \epsilon_{\mathcal{D}_n}(h_n, f_n) + \epsilon_{\mathcal{D}_o}(h_o, f_o) + \epsilon_{\hat{\mathcal{D}}_o}(h_n, h_o) + d_1(\hat{\mathcal{D}}_o, \mathcal{D}_o). \quad (2)$$

Proof See supplementary material.

In this bound, the first and second terms on the right-hand side of the inequality are expected to be small, achieved by empirical risk minimization. The third term describes the difference between h_n and h_o under the distillation data distribution $\hat{\mathcal{D}}_o$, which is guaranteed by KD. The last term is the divergence between the distributions of old class $\mathcal{D}_o$ and distillation data $\hat{\mathcal{D}}_o$. This bound reveals that a better CIL performance can be achieved by designing better KD techniques and using distillation data that resemble the distribution of the old classes. In the absence of old class data, many non-exemplar methods resort to using new class data D_n as a substitute for $\hat{\mathcal{D}}_o$. However, there is a notable discrepancy between $\mathcal{D}_n$ and $\mathcal{D}_o$, which causes the model to deviate from its previous minima and results in significant forgetting of the old classes. Some other methods tackle this problem by introducing synthetic samples that resemble the semantic distribution of the old classes into the KD process, which alleviates forgetting to some extent. Nonetheless, these methods ignore the domain gap between the synthetic and real distribution, leading to a substantial discrepancy between the synthetic and real distribution of the old classes. In this work, we solve this issue by generating better synthetic samples that possess both semantic and domain consistency.

4. Proposed Method

4.1. Dual-consistency Inverting for Old Classes

Recent generative NECIL approaches [13, 42] utilize model inversion [4, 32, 36] to synthesize old class samples. However, the significant domain gap negatively impacts the efficacy of KD. In the following, we introduce a novel model inversion approach for old class synthesis that ensures both semantic and domain consistency.

4.1.1 Semantic Consistency

The overview of our method is depicted in Fig. 2. To disentangle classes, we utilize a conditional generator G with learnable embedding vectors $\mathcal{E} = \{e_i\}_{i=1}^{m}$, where $m = \sum_{i=0}^{t-1}|\mathcal{C}_i|$ is the number of old classes at phase t. The embedding space is formulated as the linear span of the embedding vectors $\mathcal{E}$. For each old class k $(1 \leqslant k \leqslant m)$, the input embedding $\hat{e}_k$ of G is calculated as a linear combination of $\mathcal{E}$ weighted by α_i^k:

$$\hat{e}_k = \sum_{i=1}^{m} \alpha_i^k e_i, \quad (3)$$

where α_i^k represents the cosine similarity between class i and k, followed by the softmax layer:

$$\alpha_i^k = \frac{\exp(\cos(p_i, p_k))}{\sum_{j=1}^{m}\exp(\cos(p_j, p_k))}, \quad (4)$$

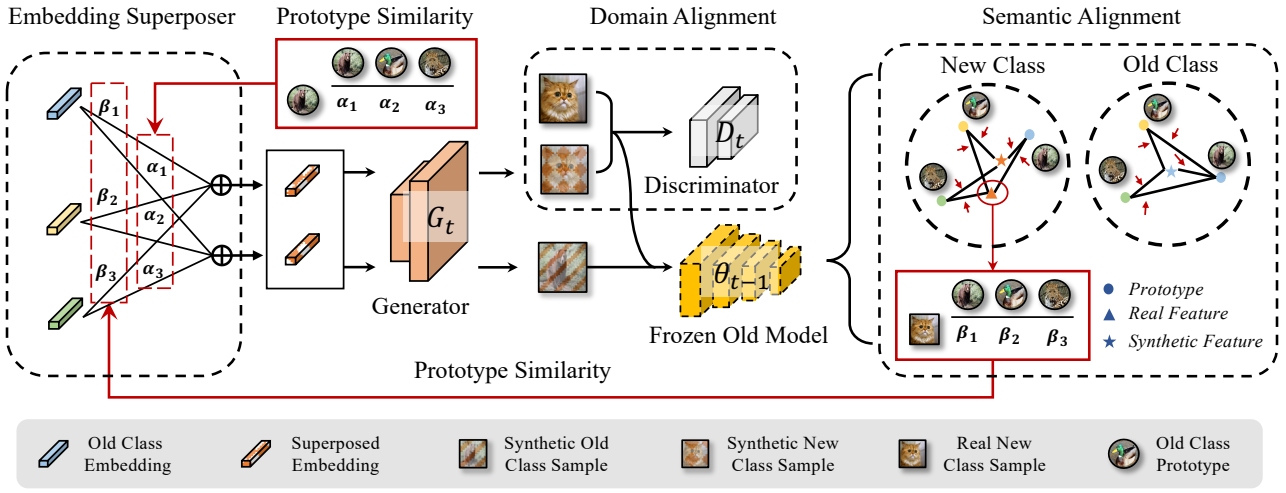


Figure 2. The overview of DCMI. The superposed embedding (lower) of old classes is formed by the linear combination of old class embeddings, weighted by α. The ones (upper) of new classes is weighted by β which is the similarity between new class features (triangle) and the prototypes of old classes. These embeddings are input to the generator, and the resulting features are aligned within the feature space for semantic consistency. The discriminator ensures domain consistency by aligning synthetic and real images of the new classes.

where p_i represents the mean feature of class i, referred to as the class prototype, and $\cos(p_i, p_k) = \frac{p_i^T p_k}{\|p_i\|\|p_k\|}$. Then, the synthetic sample for the old classes is generated as follows:

$$\hat{x}_k = G\left(\hat{e}_k \oplus z\right), z \sim \mathcal{N}(0,1), \tag{5}$$

where $\oplus$ denotes dimensional concatenation.

Then the synthetic samples are projected into the deep feature space of the old model θ_{t-1}. Subsequently, the softmax similarity between $\theta_{t-1}\left(\hat{x}_k\right)$ and the prototypes is calculated as follows:

$$\hat{y}_i\left(\hat{x}_k\right) = \frac{\exp(\cos(p_i, \theta_{t-1}\left(\hat{x}_k\right)))}{\sum_{j=1}^{m}\exp(\cos(p_j, \theta_{t-1}\left(\hat{x}_k\right)))}. \tag{6}$$

If $\theta_{t-1}\left(\hat{x}_k\right)$ captures the semantics of class k, the similarity $\hat{y}_i\left(\hat{x}_k\right)$ should align with α_i^k. Thus we minimize the cross-entropy loss to ensure semantic consistency:

$$\mathcal{L}_{SC}^{o}(\hat{x}_k) = -\sum_{i=1}^{m}\alpha_i^k \log \hat{y}_i\left(\hat{x}_k\right). \tag{7}$$

Similarly, the new class images x_l $(m < l \leqslant n, n = \sum_{i=0}^{t}|\mathcal{C}_i|)$ are projected into the feature space to derive prototype similarity:

$$\beta_i^{x_l} = \frac{\exp(\cos(p_i, \theta_{t-1}\left(x_l\right)))}{\sum_{j=1}^{m}\exp(\cos(p_j, \theta_{t-1}\left(x_l\right)))}. \tag{8}$$

By linearly combining e_i with weights $\beta_i^{x_l}$, the superposed embedding vector of new classes can be obtained:

$$\hat{e}_{x_l} = \sum_{i=1}^{m}\beta_i^{x_l} e_i. \tag{9}$$

To further enhance the semantic consistency, we minimize the cross-entropy loss for new classes:

$$\mathcal{L}_{SC}^{n}(\hat{x}_l) = -\sum_{i=1}^{m}\beta_i^{x_l}\log \hat{y}\left(\hat{x}_l\right), \tag{10}$$

where $\hat{x}_l = G\left(\hat{e}_{x_l} \oplus z\right)$. The overall semantic consistency loss is written as:

$$\begin{aligned}\min_G \mathcal{L}_{SC}(G) =& \mathbb{E}_{\hat{x}_k \sim G(\hat{e}_k \oplus z)}\left[\mathcal{L}_{SC}^{o}(\hat{x}_k)\right] \\ &+ \mathbb{E}_{\hat{x}_l \sim G\left(\hat{e}_{x_l} \oplus z\right)}\left[\mathcal{L}_{SC}^{n}(\hat{x}_l)\right].\end{aligned} \tag{11}$$

4.1.2 Domain Consistency

To address the issue of domain shift, we employ an adversarial learning framework [14] to align the distribution of synthetic data $\hat{x}_l$ with that of real data x_l of the new classes:

$$\begin{aligned}\min_G \max_D \mathcal{L}_{DC}(G,D) =& \mathbb{E}_{x_l}\left[\log D\left(x_l\right)\right] \\ &+ \mathbb{E}_{\hat{x}_l \sim G\left(\hat{e}_{x_l} \oplus z\right)}[\log(1 - D(\hat{x}_l))],\end{aligned} \tag{12}$$

where D is a discriminator that distinguishes between the domains of $\hat{x}_l$ and x_l, while G generates the domain distribution that deceives the discriminator D. The objective of Eq. 12 is equivalent to minimizing the Jensen–Shannon divergence between the distributions P_{x_l} and $P_{\hat{x}_l}$ [14]. Since the semantic distribution is already consistent, the domain distribution of P_{x_l} becomes disentangled and learned. In summary, the overall optimization objective is updated alternately in a min-max fashion:

$$\min_G \max_D \mathcal{L}_{Syn} = \mathcal{L}_{SC}(G) + \lambda \mathcal{L}_{DC}(G, D). \tag{13}$$

We empirically set $\lambda = 0.5$ to balance the consistency between semantics and domain.

4.2. Adapting Network for New Classes

To learn new classes $\mathcal{X}_t$, we initialize a new classifier φ_t, inheriting parameters from the old classifier: $\varphi_t^{1:m} = \varphi_{t-1}^{1:m}$, where the superscript represents the class index. As suggested by [42], to prevent overfitting the generated samples and potentially causing incorrect decision boundaries, we employ local cross-entropy to learn the new classes:

$$\mathcal{L}_{LCE} = -\mathcal{Y}_t \log \left(P_{\theta_t,\varphi_t}^{m+1:n} \left(\mathcal{X}_t \right) \right), \tag{14}$$

where $P_{\theta_t,\varphi_t}^{m+1:n}$ represents the softmax probabilities computed among the new classes.

Using the synthetic data of the old classes $\hat{\mathcal{X}}_t = \{\hat{x}_i\}_{i=1}^m$, generated by Eq. 5, we apply knowledge distillation to the penultimate layer features and the classifier output logits:

$$\begin{aligned} \mathcal{L}_{KD} = &\left[1 - \cos \left(\theta_t(\mathcal{X}_t \cup \hat{\mathcal{X}}_t), \theta_{t-1}(\mathcal{X}_t \cup \hat{\mathcal{X}}_t) \right) \right] \\ &+ \left\| P_{\theta_t,\varphi_t}^{1:m}(\mathcal{X}_t \cup \hat{\mathcal{X}}_t) - P_{\theta_{t-1},\varphi_{t-1}}^{1:m}(\mathcal{X}_t \cup \hat{\mathcal{X}}_t) \right\|. \end{aligned} \tag{15}$$

The first term (referred to as $\mathcal{L}_{FKD}$) prevents class prototypes from drifting and stabilizes the feature distribution. The second term (referred to as $\mathcal{L}_{CKD}$) maintains the discriminative power of the classifier. The overall adapting loss is written as:

$$\min_{\theta_t,\varphi_t} \mathcal{L}_{Ad} = \mathcal{L}_{LCE} + \gamma \mathcal{L}_{KD}, \tag{16}$$

where γ is empirically set to 10. It's worth noting that both the generator G and the embeddings $\mathcal{E}$ are discarded after the adaptation process. Adhering strictly to the NECIL setting, no synthetic or real data is stored for future tasks.

4.3. Prototypical Routing for Unbiased Predictions

The classifier is trained separately for the old and new classes (refer to $\mathcal{L}_{LCE}$ and $\mathcal{L}_{CKD}$), which prevents the establishment of a joint prediction space. Consequently, biased predictions across tasks occur, leading to inferior performance. To tackle this, we propose Prototypical Routing (PR) to eliminate bias in predictions. Specifically, the linear classifier φ_t is partitioned into multiple heads, with each head corresponding to a specific task and responsible only for the classes within that task. To obtain an unbiased task-prior, we apply prototypical matching [44]:

$$\hat{u} = \arg\min_{\mathcal{T}(i)} d\left(\theta_t(x), p_i \right), \tag{17}$$

where $d(\cdot)$ denotes cosine distance and $\mathcal{T}(\cdot)$ represents the mapping from class to task. $\hat{u}$ is the task label of the most similar prototype, which is used as the task prior for predictions. Then, the corresponding head is activated by task-prior $\hat{u}$, and the linear classifier outputs the task-post class predictions:

$$\hat{y} = \arg\min_{y} \varphi_t^{\mathcal{T}'(\hat{u}):\mathcal{T}'(\hat{u}+1)} \left(\theta_t(x) \right). \tag{18}$$

where $\mathcal{T}'(\hat{u})$ represents the first class index in task $\hat{u}$.

The algorithm of DCMI in incremental task t ($1 \leq t \leq N$) is described in Algo. 1:

Algorithm 1 Dual-Consistency Model Inversion for NECIL

Input: Data of new classes $\mathcal{X}_t$; old model θ_{t-1}; old classifier φ_{t-1}; old class prototypes $\{p_i \mid 1 \leqslant i \leqslant m\}$
Output: New model θ_t; new classifier φ_t; new class prototypes $\{p_i \mid m < i \leqslant n\}$

1. Synthesizing images of old classes:
for number of epochs **do**
  Generate $\hat{x}_k$ by Eq. 5;
  Update G and D alternately by Eq. 12;
end for
2. Adapting new classes:
for number of epochs **do**
  Generate $\hat{\mathcal{X}}_t = \{\hat{x}_i\}_{i=1}^m$ by Eq. 5;
  Forward $\mathcal{X}_t$ to $\{\theta_t, \varphi_t\}$ and compute $\mathcal{L}_{LCE}$ by Eq. 14;
  Forward $\mathcal{X}_t \cup \hat{\mathcal{X}}_t$ to $\{\theta_t, \varphi_t, \theta_{t-1}, \varphi_{t-1}\}$ and compute $\mathcal{L}^{KD}$ by Eq. 15;
  Update $\{\theta_t, \varphi_t\}$ by minimizing Eq. 16;
end for
Compute new class prototypes;
3. Inference with prototypical routing:
Predict $\hat{y}$ by Eq. 17 and Eq. 18;

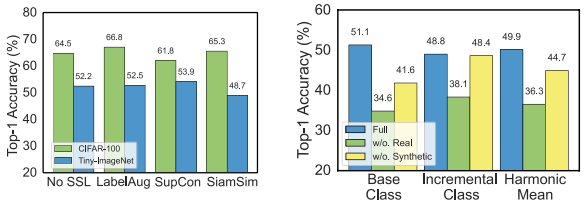
5. Experiments

5.1. Dataset and Settings

Benchmark. To achieve a comprehensive study, we conduct extensive experiments on the CIFAR-100 [24], Tiny-ImageNet [25], ImageNet-Subset [9], and ImageNet-Full [9]. The protocol is consistent with [55] where half of the classes are involved in the initial phase (except for CIFAR-100 and ImageNet-Subset 20 phases), and the rest are evenly distributed across subsequent incremental phases. The class order is shuffled with random seed 1993. We report the standard metrics to measure the CIL performance, including average accuracy A_N, and average forgetting F_N [3]. A desirable CIL approach should effectively learn new classes (high A_N) while minimizing the forgetting of learned knowledge (low F_N).

Setting	P=5		P=10		P=20	
	$A_N(\uparrow)$	$F_N(\downarrow)$	$A_N(\uparrow)$	$F_N(\downarrow)$	$A_N(\uparrow)$	$F_N(\downarrow)$
Ablate PR	58.1	0.9	56.8	0.9	49.5	1.0
Ablate SSL	65.1	7.7	64.5	7.4	60.9	8.6
Ablate $\mathcal{L}_{CKD}$	22.6	73.8	10.0	73.1	10.6	72.5
Ablate $\mathcal{L}_{FKD}$	67.2	9.0	65.1	10.8	56.3	24.7
Full Method	67.9	7.8	66.8	7.3	64.0	9.8

Table 1. Results (%) of ablating components on CIFAR-100 dataset. PR and SSL stand for prototypical routing and self-supervised learning, respectively.



(a) (b)

Figure 3. (a) Comparison of different self-supervised learning approaches. (b) Stability and plastic analysis conducted on CIFAR-100.

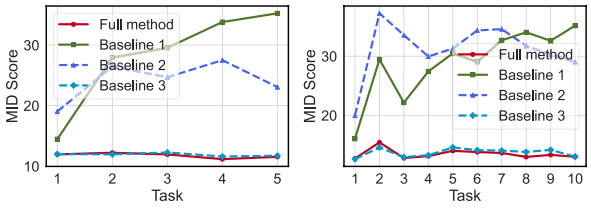
Phases			P=5		P=10		P=20	
Data	SC	DC	$A_N(\uparrow)$	$F_N(\downarrow)$	$A_N(\uparrow)$	$F_N(\downarrow)$	$A_N(\uparrow)$	$F_N(\downarrow)$
(1)	✓	✗	28.9	80.4	22.4	84.5	17.5	86.5
	✗	✓	54.6	29.0	44.0	48.1	28.4	68.3
	✓	✓	67.3	11.4	63.4	18.4	51.7	34.8
(2)	✗	✗	67.6	8.1	65.5	8.8	60.2	13.3
(3)	✓	✓	67.9	7.8	66.8	7.3	64.0	9.8

Table 2. Results (%) of ablating KD data on CIFAR-100 dataset with (1) only synthetic data, (2) only real data, and (3) real + synthetic data. SC and DC denote Semantic Consistency and Domain Consistency, respectively.

Implementation Details. To consist with most previous works [39, 54–56], we follow the ResNet-18 backbone in NECIL to ensure fair and comprehensive comparisons. Similarly, following [55, 56], we apply self-supervised learning (SSL) to the initial task to enable the learning of more generalizable features, while the standard supervised training process is performed for the incremental tasks. LabelAug [26] and SupCon [20] are employed on CIFAR-100 and ImageNet, respectively. Our model is trained from scratch for 100 epochs using the Adam optimizer [22] with an initial learning rate of 0.001 (except 0.0001 on ImageNet-Full). The batch size is set to 128 and the learning rate is reduced based on the cosine annealing schedule. See supplementary material for more details.

5.2. Ablation Study and Analysis

Prototypical Routing. As shown in Tab. 1, the removal of PR leads to a discernible performance decline, primarily attributed to the introduction of task-recency bias. This is illustrated in Fig. 6a and 6b, where a pronounced bias is observed in the absence of PR, resulting in substantial confusion among the base classes. The incorporation of PR proves effective in mitigating this bias by furnishing the correct task prior.



(a) CIFAR-100 5 phases. (b) CIFAR-100 10 phases.

Figure 4. Representational distance analysis

Method	CIFAR-100			Tiny-ImageNet		
	P=5	P=10	P=20	P=5	P=10	P=20
iCaRL-FC	42.1	45.7	43.5	36.9	36.7	45.1
iCaRL-NCM	24.9	28.3	35.5	27.2	28.9	37.4
EEIL	23.4	26.7	32.4	25.6	25.9	35.0
UCIR	21.0	25.1	28.7	20.6	22.3	33.7
LwF-MC	44.2	50.5	55.5	54.3	54.4	63.5
MUC	40.3	47.6	52.7	51.5	50.2	58.0
PASS	25.2	30.3	30.6	18.0	23.1	30.6
SSRE	18.4	19.5	19.0	9.2	14.1	14.2
DCMI	7.8	7.3	9.8	7.5	6.9	8.3

Table 3. Results of average forgetting on 5, 10, and 20 phases.

Self-Supervised Learning. Tab. 1 clearly demonstrates that the incorporation of self-supervised learning (SSL) into the initial task significantly enhances CIL performance. As highlighted in [55], SSL contributes to the acquisition of more generalizable and transferable features, proving advantageous for subsequent tasks. We assess the effectiveness of various SSL methods on CIFAR-100 and Tiny-ImageNet 10 phases. Three commonly used SSL methods are examined: LabelAug [26], SupCon [20], and SimSiam [5]. Fig. 3a illustrates that applying SSL on the initial task yields substantial improvements, emphasizing the importance of SSL in CIL. However, the effectiveness of SSL methods varies across datasets. On CIFAR-100, LabelAug proves the most effective result, achieving 2.3% higher than not using SSL. On Tiny-ImageNet, SupCon emerges as the most effective SSL method, yielding improvements of 1.7% over not using SSL.

Knowledge Distillation. As shown in Tab. 1, both $\mathcal{L}_{CKD}$ and $\mathcal{L}_{FKD}$ are crucial components. Removing $\mathcal{L}_{CKD}$ impairs decision boundaries of old classes, resulting in a significant drop in accuracy and an extremely high forgetting rate. Removing $\mathcal{L}_{FKD}$ induces a shift in the output feature distribution, undermining the effectiveness of the class prototypes [50]. Furthermore, we explore the role of semantic and domain consistency in synthetic data during KD. As shown in Tab. 2, relying solely on semantic or domain consistency fails to guarantee satisfactory performance. Better accuracy and lower forgetting are achieved when synthetic data exhibit both semantic and domain consistency. The best results emerge when KD is simultaneously applied to the real data of new classes and the synthetic data of old

	Method	CIFAR-100			Tiny-ImageNet			ImageNet-Subset			ImageNet-Full
		P=5	*P=10*	*P=20*	*P=5*	*P=10*	*P=20*	*P=5*	*P=10*	*P=20*	*P=10*
E=20	iCaRL-FC* [41]	51.1	48.7	44.4	34.6	31.2	27.9	-	50.5	-	38.4
	iCaRL-NCM* [41]	58.6	54.2	50.5	45.9	43.3	38.0	-	60.8	-	46.7
	EEIL* [2]	60.4	56.1	52.3	47.1	45.0	40.5	-	63.3	-	-
	UCIR* [19]	63.8	62.4	59.1	49.2	48.5	42.8	-	66.2	-	61.3
	PODNet† [10]	65.6	62.8	59.7	42.3	36.7	32.1	-	70.6	-	-
E=0	EWC* [23]	24.5	21.2	15.9	18.8	15.8	12.4	-	20.4	-	-
	LwF-MC* [41]	45.9	27.4	20.1	29.1	23.1	17.4	-	31.2	-	-
	MUC* [30]	49.4	30.2	21.3	32.6	26.6	22.0	-	35.1	-	-
	SDC [50]	56.8	57.0	58.9	-	-	-	-	61.2	-	-
	ABD* [42]	63.9	62.5	57.4	-	-	-	-	-	-	-
	PASS [55]	63.5	61.8	58.1	49.6	47.3	42.1	64.4	61.8	51.3	55.9
	IL2A [54]	66.0	60.3	57.9	47.3	44.7	40.0	-	-	-	-
	SSRE [56]	65.9	65.0	61.7	50.4	48.9	48.2	-	67.7	-	58.1
	FeTrIL$_{fc}$ [39]	64.7	63.4	57.4	52.9	51.7	49.7	69.6	68.9	62.5	**64.4**
	SOPE [57]	66.6	65.8	61.8	53.7	52.9	51.9	-	69.2	-	60.2
	DCMI	**67.9**	**66.8**	**64.0**	**54.8**	**53.9**	**52.5**	**70.5**	**70.0**	**65.5**	61.9

Table 4. Average accuracy (Top-1 accuracy %) on CIFAR-100, TinyImageNet, and ImageNet-Subset. The previous state-of-the-art results among non-exemplar methods are underlined. The best results among non-exemplar methods are in bold. Methods with * represent the reproduced results in [55, 56]. Methods with † stand for our reproduced results. Other results are reported in the original paper.

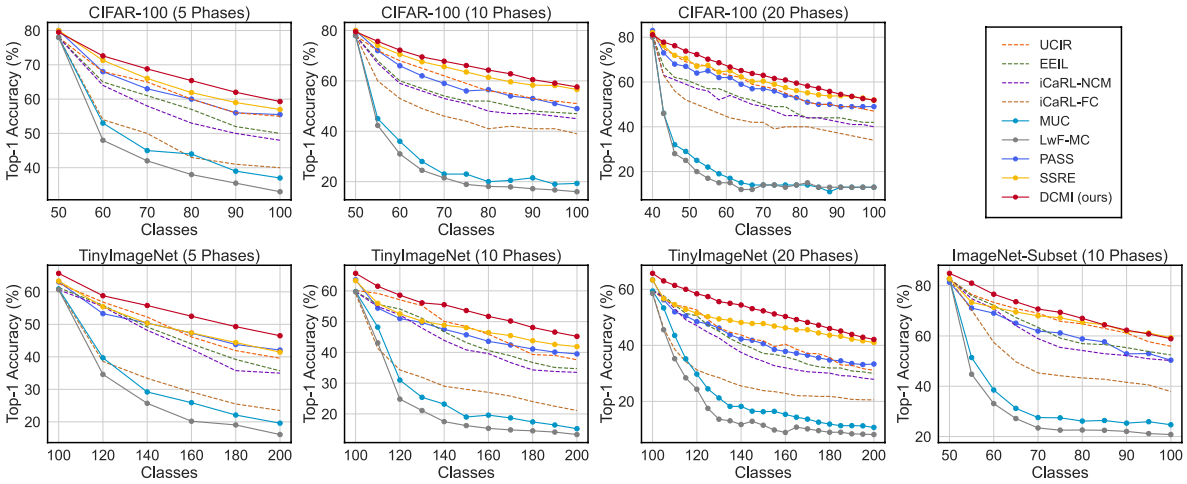
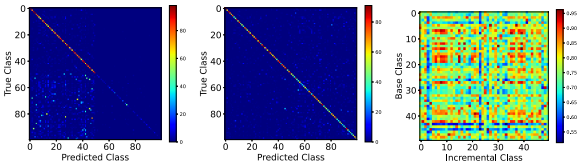


Figure 5. Accuracy for each phase on CIFAR-100, Tiny-ImageNet, and ImageNet-Subset.

classes. Additionally, we assess the impact of integrating synthetic data into KD on model stability and plasticity. Experimental results in Fig. 3b indicate that distillation with a combination of synthetic and real data yields higher accuracy for both base and incremental classes, striking a more favorable balance between stability and plasticity.

Representational Distance. We investigate the representational distance scores (MID) [42] between the synthetic and real samples of the old classes. As depicted in Fig. 4, we designate the settings of Group 1 in Tab. 2 as baselines 1 $\sim$ 3, from top to bottom. The results reveal that baseline 3 and the full method achieve comparably lower MID scores, highlighting the significance of dual consistency in generating distributions closely resembling those of the real samples from the old classes. Conversely, the exclusive alignment of semantics or domain proves inadequate in achieving a low representational distance. Furthermore, this distance tends to increase as the tasks expand.

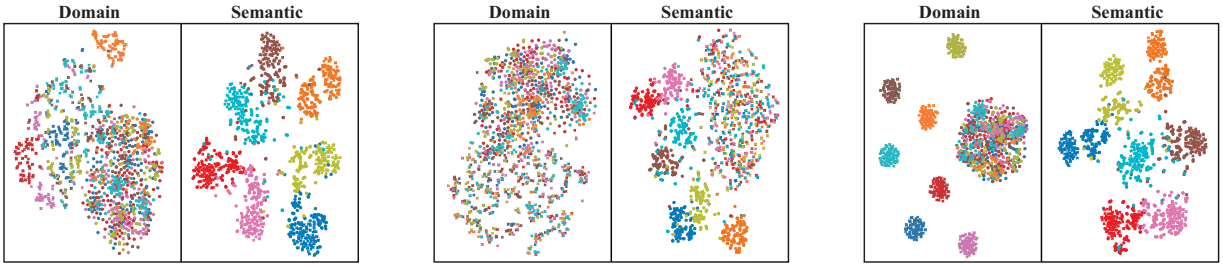


(a) (b) (c)

Figure 6. Visualization on CIFAR-100: Confusion matrix (a) without and (b) with prototypical routing. (c) Similarity statistics matrix.

5.3. Comparative Results

In this section, we conduct a comprehensive comparison with previous state-of-the-art NECIL methods and some classical exemplar-based methods. Tab. 4 illustrates that



(a) DC + SC (b) w/o SC (c) w/o DC

Figure 7. Visualization using t-SNE on CIFAR-100. Synthetic data is denoted by crosses and real data by dots. Features with the same class label share a common color.

the proposed method outperforms all previous state-of-the-art non-exemplar methods. Meanwhile, our method exhibits superior performance compared to certain exemplar-based methods even without utilizing exemplars, underscoring its reliability in retaining old knowledge. Specifically, our method significantly improves upon state-of-the-art results on CIFAR-100, surpassing previous best results by 1.3%, 1.0%, and 2.2% under 5, 10, and 20 phases, respectively. On Tiny-ImageNet, our method outperforms the most recent state-of-the-art method SOPE, by margins of 1.1%, 1.0%, and 0.6% under 5, 10, and 20 phases, respectively. For larger-scale datasets, our method attains an average accuracy of 70.5%, 70.0%, and 65.5% on ImageNet-Subset, and 61.9% on ImageNet-Full. In Fig. 5, we present comprehensive classification accuracy curves, clearly illustrating the superiority of our method over competitors across various phases. To estimate the forgetting of the model, we compare the average forgetting with previous methods, as shown in Tab. 3. Our method consistently achieves significantly lower forgetting than previous methods, demonstrating its effectiveness in mitigating catastrophic forgetting.

5.4. Visualization

Visualization of Feature Space. We employ t-SNE [43] to visualize the feature distribution of synthetic and real data of the old classes. In each subfigure of Fig. 7, the left half showcases shallow features that excel at distinguishing different domains [53], while the right half displays deep features that are semantically discriminative. Fig. 7a reveals that synthetic data forms distinct clusters alongside real data of the same class, indicating high semantic consistency. Simultaneously, the synthetic data exhibits a minor domain discrepancy from real data, demonstrating domain consistency. In contrast, Fig. 7b fails to align with corresponding classes, and Fig. 7c primarily highlights a significant domain discrepancy.

Visualization of Similarity Statistics. We evaluate the similarities between the base and incremental classes on CIFAR-100 by computing the cosine similarity of the class prototypes, as depicted in Fig. 6c. The results reveal a relatively high similarity between the base and incremental classes, suggesting the feasibility of representing new class semantics through the combination of old class concepts.

Visualization of Synthetic Samples. Here we provide visualizations of synthetic samples sourced from ImageNet. When both semantic and domain constraints are applied (refer to Fig. 8b), the generated samples closely resemble the real ones. However, when only the domain consistency is applied, no discernible class-related patterns are observed (see Fig. 8c). Similarly, when only the semantic consistency is applied, the synthetic images visually differ significantly from the real images (see Fig. 8d).

(a) Real (b) $\mathcal{L}_{SC}+\mathcal{L}_{DC}$ (c) Only $\mathcal{L}_{DC}$ (d) Only $\mathcal{L}_{SC}$

Figure 8. Visualization of synthetic data.

6. Conclusion

This paper proposes a novel generative approach, DCMI, for NECIL. DCMI is capable of synthesizing images that maintain consistency on both semantics and domain. The integration of these synthetic images into knowledge distillation yields significantly improved results. Furthermore, to ensure unbiased predictions, we introduce prototypical routing, providing accurate task priors to guide predictions across tasks. Extensive experiments conducted on CIFAR-100, Tiny-ImageNet, ImageNet-Subset, and ImageNet-Full consistently demonstrate the superior performance of our method compared to previous state-of-the-art approaches.

Acknowledgments Supported in part by National Science and Technology Major Project (2021ZD0112001), National Natural Science Foundation of China (No.62271119, U23A20286, 62071086, and 08120002), and Independent Research Project of Civil Aviation Flight Technology and Flight Safety Key Laboratory (FZ2022ZZ06).

References

[1] Kartikeya Bhardwaj, Naveen Suda, and Radu Marculescu. Dream distillation: A data-independent model compression framework. In *International Conference on Machine Learning*, 2019. 2

[2] Francisco Manuel Castro, Manuel J. Marín-Jiménez, Nicolás Guil Mata, Cordelia Schmid, and Alahari Karteek. End-to-end incremental learning. *ArXiv*, abs/1807.09536, 2018. 1, 2, 7

[3] Arslan Chaudhry, Puneet K Dokania, Thalaiyasingam Ajanthan, and Philip HS Torr. Riemannian walk for incremental learning: Understanding forgetting and intransigence. In *Proceedings of the European conference on computer vision (ECCV)*, pages 532–547, 2018. 5

[4] Hanting Chen, Yunhe Wang, Chang Xu, Zhaohui Yang, Chuanjian Liu, Boxin Shi, Chunjing Xu, Chao Xu, and Qi Tian. Data-free learning of student networks. *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 3513–3521, 2019. 1, 2, 3

[5] Xinlei Chen and Kaiming He. Exploring simple siamese representation learning. *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 15745–15753, 2020. 6

[6] Zailiang Chen, Xianxian Zheng, Hailan Shen, Ziyang Zeng, Yukun Zhou, and Rongchang Zhao. Improving knowledge distillation via category structure. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXVIII 16*, pages 205–219. Springer, 2020. 3

[7] Jang Hyun Cho and Bharath Hariharan. On the efficacy of knowledge distillation. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 4794–4802, 2019. 3

[8] Yoojin Choi, Jihwan P. Choi, Mostafa El-Khamy, and Jungwon Lee. Data-free network quantization with adversarial knowledge distillation. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 3047–3057, 2020. 2

[9] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, K. Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255, 2009. 2, 5

[10] Arthur Douillard, Matthieu Cord, Charles Ollion, Thomas Robert, and Eduardo Valle. Podnet: Pooled outputs distillation for small-tasks incremental learning. In *European Conference on Computer Vision*, 2020. 1, 2, 7

[11] Tommaso Furlanello, Zachary Lipton, Michael Tschannen, Laurent Itti, and Anima Anandkumar. Born again neural networks. In *International Conference on Machine Learning*, pages 1607–1616. PMLR, 2018. 3

[12] Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario Marchand, and Victor Lempitsky. Domain-adversarial training of neural networks. *The journal of machine learning research*, 17(1):2096–2030, 2016. 2

[13] Qiankun Gao, Chen Zhao, Bernard Ghanem, and Jian Zhang. R-dfcil: Relation-guided representation learning for data-free class incremental learning. In *European Conference on Computer Vision*, pages 423–439. Springer, 2022. 1, 3

[14] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron C. Courville, and Yoshua Bengio. Generative adversarial nets. In *NIPS*, 2014. 2, 4

[15] Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*, 2014. 2

[16] Byeongho Heo, Jeesoo Kim, Sangdoo Yun, Hyojin Park, Nojun Kwak, and Jin Young Choi. A comprehensive overhaul of feature distillation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1921–1930, 2019. 3

[17] Byeongho Heo, Minsik Lee, Sangdoo Yun, and Jin Young Choi. Knowledge transfer via distillation of activation boundaries formed by hidden neurons. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 3779–3787, 2019. 3

[18] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *In Advances in Neural Information Processing Systems (NeurIPS) Workshop*, 2015. 1, 3

[19] Saihui Hou, Xinyu Pan, Chen Change Loy, Zilei Wang, and Dahua Lin. Learning a unified classifier incrementally via rebalancing. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 831–839, 2019. 1, 2, 7

[20] Prannay Khosla, Piotr Teterwak, Chen Wang, Aaron Sarna, Yonglong Tian, Phillip Isola, Aaron Maschinot, Ce Liu, and Dilip Krishnan. Supervised contrastive learning. *Advances in neural information processing systems*, 33:18661–18673, 2020. 6

[21] Jangho Kim, SeongUk Park, and Nojun Kwak. Paraphrasing complex network: Network compression via factor transfer. *Advances in neural information processing systems*, 31, 2018. 3

[22] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980, 2014. 6

[23] James Kirkpatrick, Razvan Pascanu, Neil C. Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A. Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, Demis Hassabis, Claudia Clopath, Dharshan Kumaran, and Raia Hadsell. Overcoming catastrophic forgetting in neural networks. *Proceedings of the National Academy of Sciences*, 114:3521 – 3526, 2016. 1, 2, 7

[24] Alex Krizhevsky. Learning multiple layers of features from tiny images. 2009. 2, 5

[25] Ya Le and Xuan S. Yang. Tiny imagenet visual recognition challenge. 2015. 2, 5

[26] Hankook Lee, Sung Ju Hwang, and Jinwoo Shin. Self-supervised label augmentation via input transformations. In *International Conference on Machine Learning*, 2019. 6

[27] Zhizhong Li and Derek Hoiem. Learning without forgetting. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40:2935–2947, 2016. 2

[28] Li Liu, Qingle Huang, Sihao Lin, Hongwei Xie, Bing Wang, Xiaojun Chang, and Xiaodan Liang. Exploring inter-channel correlation for diversity-preserved knowledge distillation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8271–8280, 2021. 3

[29] Yufan Liu, Jiajiong Cao, Bing Li, Chunfeng Yuan, Weiming Hu, Yangxi Li, and Yunqiang Duan. Knowledge distillation via instance relationship graph. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7096–7104, 2019. 3

[30] Yu Liu, Sarah Parisot, Gregory G. Slabaugh, Xu Jia, Ale Leonardis, and Tinne Tuytelaars. More classifiers, less forgetting: A generic multi-classifier paradigm for incremental learning. In *European Conference on Computer Vision*, 2020. 1, 7

[31] Yuang Liu, Wei Zhang, and Jun Wang. Zero-shot adversarial quantization. *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1512–1521, 2021. 2

[32] Raphael Gontijo Lopes, Stefano Fenu, and Thad Starner. Data-free knowledge distillation for deep neural networks. In *Advances in Neural Information Processing Systems*, 2017. 1, 2, 3

[33] Massimiliano Mancini, Zeynep Akata, Elisa Ricci, and Barbara Caputo. Towards recognizing unseen categories in unseen domains. In *European Conference on Computer Vision*, pages 466–483. Springer, 2020. 2

[34] Seyed Iman Mirzadeh, Mehrdad Farajtabar, Ang Li, Nir Levine, Akihiro Matsukawa, and Hassan Ghasemzadeh. Improved knowledge distillation via teacher assistant. In *Proceedings of the AAAI conference on artificial intelligence*, pages 5191–5198, 2020. 3

[35] A. Mordvintsev, Christopher Olah, and Mike Tyka. Inceptionism: Going deeper into neural networks. 2015. 2

[36] Gaurav Kumar Nayak, Konda Reddy Mopuri, Vaisakh Shaj, Venkatesh Babu Radhakrishnan, and Anirban Chakraborty. Zero-shot knowledge distillation in deep networks. In *International Conference on Machine Learning*, pages 4743–4751. PMLR, 2019. 1, 2, 3

[37] Wonpyo Park, Dongju Kim, Yan Lu, and Minsu Cho. Relational knowledge distillation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 3967–3976, 2019. 3

[38] Baoyun Peng, Xiao Jin, Jiaheng Liu, Dongsheng Li, Yichao Wu, Yu Liu, Shunfeng Zhou, and Zhaoning Zhang. Correlation congruence for knowledge distillation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5007–5016, 2019. 3

[39] Gregoire Petit, Adrian Popescu, Hugo Schindler, David Picard, and Bertrand Delezoide. Fetril: Feature translation for exemplar-free class-incremental learning. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 3911–3920, 2023. 2, 6, 7

[40] Zihuan Qiu, Linfeng Xu, Zhichuan Wang, Qingbo Wu, Fanman Meng, and Hongliang Li. Ism-net: Mining incremental semantics for class incremental learning. *Neurocomputing*, 523:130–143, 2023. 2

[41] Sylvestre-Alvise Rebuffi, Alexander Kolesnikov, G. Sperl, and Christoph H. Lampert. icarl: Incremental classifier and representation learning. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5533–5542, 2016. 1, 2, 7

[42] James Smith, Yen-Chang Hsu, John C. Balloch, Yilin Shen, Hongxia Jin, and Zsolt Kira. Always be dreaming: A new approach for data-free class-incremental learning. *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 9354–9364, 2021. 1, 2, 3, 5, 7

[43] Laurens van der Maaten and Geoffrey E. Hinton. Visualizing data using t-sne. *Journal of Machine Learning Research*, 9: 2579–2605, 2008. 8

[44] Oriol Vinyals, Charles Blundell, Timothy Lillicrap, Daan Wierstra, et al. Matching networks for one shot learning. 2016. 2, 5

[45] Shipeng Yan, Jiangwei Xie, and Xuming He. Der: Dynamically expandable representation for class incremental learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3014–3023, 2021. 2

[46] Chenglin Yang, Lingxi Xie, Chi Su, and Alan L Yuille. Snapshot distillation: Teacher-student optimization in one generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2859–2868, 2019. 3

[47] Yang Yang, Da-Wei Zhou, De-Chuan Zhan, Hui Xiong, and Yuan Jiang. Adaptive deep models for incremental learning: Considering capacity scalability and sustainability. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 74–82, 2019. 2

[48] Yang Yang, Zhen-Qiang Sun, Hengshu Zhu, Yanjie Fu, Yuanchun Zhou, Hui Xiong, and Jian Yang. Learning adaptive embedding considering incremental class. *IEEE Transactions on Knowledge and Data Engineering*, 35(3):2736–2749, 2021. 1

[49] Hongxu Yin, Pavlo Molchanov, Zhizhong Li, José Manuel Álvarez, Arun Mallya, Derek Hoiem, Niraj Kumar Jha, and Jan Kautz. Dreaming to distill: Data-free knowledge transfer via deepinversion. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8712–8721, 2019. 2

[50] Lu Yu, Bartlomiej Twardowski, Xialei Liu, Luis Herranz, Kai Wang, Yongmei Cheng, Shangling Jui, and Joost van de Weijer. Semantic drift compensation for class-incremental learning. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6980–6989, 2020. 1, 2, 6, 7

[51] Friedemann Zenke, Ben Poole, and Surya Ganguli. Continual learning through synaptic intelligence. In *International conference on machine learning*, pages 3987–3995. PMLR, 2017. 2

[52] Da-Wei Zhou, Qi-Wei Wang, Han-Jia Ye, and De-Chuan Zhan. A model or 603 exemplars: Towards memory-efficient class-incremental learning. In *The Eleventh International Conference on Learning Representations*, 2022. 2

[53] Kaiyang Zhou, Yongxin Yang, Yu Qiao, and Tao Xiang. Domain generalization with mixstyle. In *International Conference on Learning Representations*, 2020. 8

[54] Fei Zhu, Zhen Cheng, Xu-Yao Zhang, and Cheng-Lin Liu. Class-incremental learning via dual augmentation. In *Neural Information Processing Systems*, 2021. 6, 7

[55] Fei Zhu, Xu-Yao Zhang, Chuan Wang, Fei Yin, and Cheng-Lin Liu. Prototype augmentation and self-supervision for incremental learning. *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5867–5876, 2021. 2, 5, 6, 7

[56] Kai Zhu, Wei Zhai, Yang Cao, Jiebo Luo, and Zhengjun Zha. Self-sustaining representation expansion for non-exemplar class-incremental learning. *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9286–9295, 2022. 1, 2, 6, 7

[57] Kai Zhu, Kecheng Zheng, Ruili Feng, Deli Zhao, Yang Cao, and Zheng-Jun Zha. Self-organizing pathway expansion for non-exemplar class-incremental learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 19204–19213, 2023. 7