CD^2: Constrained Dataset Distillation for Few-Shot Class-Incremental Learning

Kexin Bao[1,2], **Daichi Zhang**[1,2], **Hansong Zhang**[1,2], **Yong Li**[1], **Yutao Yue**[3] and **Shiming Ge**[1]
[1]Institute of Information Engineering, Chinese Academy of Sciences
[2]School of Cyber Security, University of Chinese Academy of Sciences
[3]Hong Kong University of Science and Technology (Guangzhou)
{baokexin, zhangdaichi, zhanghansong, liyong}@iie.ac.cn, yutaoyue@hkust-gz.edu.cn,
geshiming@iie.ac.cn

Abstract

Few-shot class-incremental learning (FSCIL) receives significant attention from the public to perform classification continuously with a few training samples, which suffers from the key catastrophic forgetting problem. Existing methods usually employ an external memory to store previous knowledge and treat it with incremental classes equally, which cannot properly preserve previous essential knowledge. To solve this problem and inspired by recent distillation works on knowledge transfer, we propose a framework termed **C**onstrained **D**ataset **D**istillation (**CD^2**) to facilitate FSCIL, which includes a dataset distillation module (**DDM**) and a distillation constraint module (**DCM**). Specifically, the DDM synthesizes highly condensed samples guided by the classifier, forcing the model to learn compacted essential class-related clues from a few incremental samples. The DCM introduces a designed loss to constrain the previously learned class distribution, which can preserve distilled knowledge more sufficiently. Extensive experiments on three public datasets show the superiority of our method against other state-of-the-art competitors.

1 Introduction

In real-world applications such as robotics, healthcare, and remote sensing, dealing with sequential data streams and few-shot data arise frequently and often simultaneously [Sun *et al.*, 2024; Li *et al.*, 2024]. While deep neural models demonstrate impressive performance in static and data-abundant settings [Shen *et al.*, 2024; Wei *et al.*, 2024], it is still significantly challenging for them to learn new concepts continually with a very restricted quantity of labeled samples. In this situation, few-shot class-incremental learning (FSCIL) has emerged as a promising solution and has attracted much research attention [Tao *et al.*, 2020]. In FSCIL, the model acquires extensive knowledge with sufficient labeled samples in the base session, and continually learns new knowledge from a few samples while retaining previously learned concepts in subsequent incremental sessions. Within FSCIL, the primary challenge lies in preventing catastrophic forgetting [Kang *et al.*, 2023b; Babakniya *et al.*, 2023], a phenomenon where models tend to overwrite previously acquired concepts when confronted with new data due to the unavailability of data from previous sessions, especially learning with limited data.

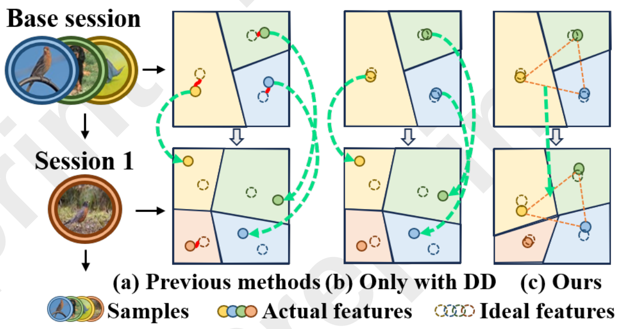

Figure 1: Red lines are the gap between actual extracted features and ideal features with all critical knowledge. Green arrows are the covariate shift. (a) Previous methods build a memory that dilutes critical knowledge, and treat the memory as equal to the real data, causing the covariate shift. (b) Using dataset distillation, the method can build a memory with more critical knowledge, but also faces the challenge of the covariate shift. (c) Our method builds and uses a memory incorporating distillation, which gains more critical knowledge and reduces the covariate shift.

Recently, mainstream methods tend to freeze the backbone and adopt an external memory as an auxiliary tool during incremental sessions to preserve previously learned knowledge [Kukleva *et al.*, 2021; Yang *et al.*, 2023a; Ahmed *et al.*, 2024]. The memory stores a small amount of data from previous sessions by replaying real samples [Kukleva *et al.*, 2021; Zhu *et al.*, 2022], generating samples [Liu *et al.*, 2022; Agarwal *et al.*, 2022] or calculating pseudo features [Hersche *et al.*, 2022; Ji *et al.*, 2023; Yang *et al.*, 2023b; Liu *et al.*, 2023a], enabling quick recall of old knowledge in subsequent sessions. Compared to only freezing the backbone, using a memory significantly boosts performance in incremental sessions, which illustrates that the memory enhances the ability of the model to revisit old knowledge and improves its information discrimination capacity. By leveraging prior experience effectively, using the memory offers a robust solution to catastrophic forgetting in an FSCIL setting.

However, we find that existing methods may still learn redundant features and cannot preserve the previous distribution in incremental sessions. As shown by red lines in Figure 1 (a), when building the memory, previous methods dilute critical class-related knowledge due to mixing discriminative and redundant knowledge, which hinders the recall of old knowledge in subsequent sessions. Recently, Dataset Distillation (DD) is proposed to obtain a few highly condensed and informative samples from a dataset [Zhao *et al.*, 2021; Zhang *et al.*, 2024], which aligns with our goal of learning essential class-related knowledge. Therefore, we introduce DD for building a memory to obtain critical knowledge and synthesize highly informative samples as shown in Figure 1 (b). Further, there is a distribution gap between data in the memory and the new dataset in incremental sessions [Yang *et al.*, 2024]. Treating them equally prompts the model to focus on distribution differences, leading to the covariate shift of old classes as illustrated by green arrows in Figure 1 (a) and (b). Facing the gap, we incorporate the core idea of knowledge distillation (KD) [Hinton *et al.*, 2015] during training. KD transfers knowledge from a teacher model (usually a well-trained, high-performing model) to a student model. And we constrain the change of distribution across continual sessions meticulously by putting the previously trained model as the teacher for the current model, which can mitigate the covariate shift and catastrophic forgetting. Combined with the core idea of distillation, our method can extract and curate critical knowledge as shown in Figure 1 (c).

Specifically, we propose a **C**onstrained **D**ataset **D**istillation framework **CD**^{2} to support FSCIL, which contains a more refined memory strategy and a more effective retention method. Firstly, we analyze existing popular memory strategies, including sample replay (reusing past training samples during incremental learning) and prototype computing (calculating the mean of intermediate features for each class) Then, to retain more essential knowledge, we propose a dataset distillation module (**DDM**) inspired by DD [Zhao *et al.*, 2021]. DDM synthesizes samples for each class to condense critical class-related knowledge, which retains knowledge stably and prevents performance degradation. Further, we employ a distillation constraint module (**DCM**) to transfer knowledge in a stable and flexible manner. The DCM promotes features and structures of old classes in the memory to align seamlessly between previous and current sessions, thereby enabling the model to constrain the distribution gap and utilize the memory more precisely and efficiently.

Our main contributions can be summarized as follows:

- We propose a framework termed Constrained Dataset Distillation (**CD**^{2}) to facilitate the FSCIL task. To the best of our knowledge, we are the first to introduce dataset distillation to FSCIL, which could effectively captures critical knowledge.
- We propose a dataset distillation module (**DDM**) to build a memory with more critical knowledge and a distillation constraint module (**DCM**) to reduce the covariate shift and maintain the stability of knowledge transmission, ensuring effective leveraging of dataset distillation.
- Extensive and comprehensive experiments on three benchmark datasets are conducted, where all results demonstrate the effectiveness of our framework.

2 Related Works

2.1 Few-Shot Class-Incremental Learning

Few-shot class-incremental learning (FSCIL) enables the model to incrementally learn new knowledge while effectively retaining old knowledge with a few new samples [Yang *et al.*, 2023a; Ahmed *et al.*, 2024]. Some early methods update the whole model using the topology [Tao *et al.*, 2020] and knowledge distillation [Dong *et al.*, 2021] in incremental sessions, which suffer from catastrophic forgetting and overfitting due to finetuning a large number of parameters.

Based on this, mainstream methods finetune part of the parameters with the assistance of a memory during incremental learning, which can reduce overfitting while retaining certain old knowledge. Some methods select random [Kukleva *et al.*, 2021] or adaptive [Zhu *et al.*, 2022] samples from old classes as the memory. However, replaying a few samples cannot obtain a representative and high generalization representation, which is also prone to exposing privacy. Some generative methods effectively discourage forgetting and protect privacy by generating synthetic data with a GAN-like model [Liu *et al.*, 2022; Agarwal *et al.*, 2022]. However, there are still deviations between generated and real samples. And some methods store pseudo-features to mitigate catastrophic forgetting under privacy protection, calculating vectors from real features [Hersche *et al.*, 2022], introducing virtual prototypes [Zhou *et al.*, 2022a; Yang *et al.*, 2023b; Liu *et al.*, 2023a], and compressing knowledge into a small number of quantized reference vectors [Chen and Lee, 2021; Ji *et al.*, 2023], which can retain knowledge quickly. However, these methods compress knowledge crudely and treat them equally with real data, affecting the model performance. Based on this, we explore a more refined memory strategy and a higher-utilization method to retain more old knowledge.

2.2 Dataset Distillation

Dataset Distillation (DD) aims to learning compressed data to enhance learning efficiency [Wang *et al.*, 2018; Cazenavette *et al.*, 2022], which is widely used in federated learning [Liu *et al.*, 2023b] and neural architecture search [Such *et al.*, 2020]. DD retains the critical information needed to train a model effectively, which can reduce the amount of data without significantly reducing the performance of the model. Some optimization methods incorporate meta-learning into the surrogate image updating [Zhou *et al.*, 2022c]. And other methods optimize the synthetic images by matching the training gradients [Kim *et al.*, 2022], feature distribution [Zhao *et al.*, 2023a], or training trajectories [Du *et al.*, 2023]. Different from traditional data compression, DD preserves adequate task-useful information so that the model trained on it can generalize well to other unseen data in subsequent tasks.

3 Preliminaries

3.1 Formulation of FSCIL

Few-shot class-incremental learning (FSCIL) [Tao *et al.*, 2020] consists of a base session and multiple incremental sessions, which trains a model incrementally on a time sequence of training sets $\mathcal{D} = \{\mathcal{D}^{(t)}\}_{t=0}^{T}$. The base (0-th) session dataset $\mathcal{D}^{(0)}$ consists of a large label space $C^{(0)}$ with

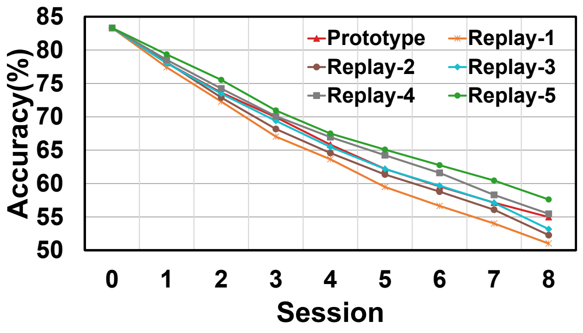

Figure 2: The downtrend of accuracy for all sessions on CIFAR100 as an example. "Replay-1", "Replay-2", "Replay-3", "Replay-4", and "Replay-5" denote using sample replay to build a memory with 1, 2, 3, 4, and 5 images per class when training a model, which trains the model by optimizing Eq. 2. And "Prototype" means using prototypical features as a memory as Eq. 3 when training a model.

sufficient samples per class. And the t-th incremental session dataset $\mathcal{D}^{(t)}(t > 0)$ has limited data with its label space $C^{(t)}(t > 0)$, which has N classes and K training examples per class (N-*way* K-*shot*). Any two label spaces are disjoint, meaning $C^{(i)} \cap C^{(j)} = \varnothing$ for all $i, j (i \neq j)$. Among them, $\mathcal{D}^{(t)} = \{(\mathbf{x}_i, y_i)\}_{i=0}^{|\mathcal{D}^{(t)}|}$, where $\mathbf{x}_i$ is an example (e.g, image), $y_i \in C^{(t)}$ denotes its target. While evaluating the performance in session t, the model is assessed on the current and all previous validation datasets $\{\mathcal{T}^{(i)}\}_{i=0}^{t}$.

During training, we train a model $\phi(x)$ in all sessions, which consists of a backbone $\phi_b(\mathbf{x}_i)$ with the intermediate feature $\mathbf{f}_i = \phi_b(\mathbf{x}_i)$, and a classifier $\phi_c(\mathbf{f}_i)$ with the final output $\mathbf{v}_i = \phi_c(\mathbf{f}_i)$. The classifier contains an MLP block to project intermediate features and a fully connected layer for classification. We train the model on dataset $\mathcal{D}^{(0)}$ in the base session, then finetune the classifier with dataset $\mathcal{D}^{(t)}(t > 0)$ and an extra memory $\mathcal{M}^{(t)}$ in t-th incremental session. Thus, the FSCIL task is formulated into two steps as follows:

$$\min \mathbb{E}_B(\mathcal{D}^{(0)}; \phi^{(0)}(\mathbf{x}_i)), \tag{1a}$$

$$\min \mathbb{E}_I(\mathcal{D}^{(t)}, \mathcal{M}^{(t)}; \phi^{(t)}(\mathbf{x}_i)), \tag{1b}$$

where Eq. (1a) and Eq. (1b) minimizes the empirical risk $\mathbb{E}_B$ in the base session ($t = 0$) and $\mathbb{E}_I$ in incremental sessions ($t > 0$). In incremental sessions, the model only has access to the dataset of the current session and not the training set of previous sessions, causing catastrophic forgetting. In this context, the memory plays an important role in mitigating this issue. Previous methods always build and use memory crudely, which cannot properly preserve critical old knowledge and further affects the distinguishing ability of the model. Therefore, we are devoted to exploring a more refined memory strategy and a higher-utilization method, which preserves more old knowledge during incremental learning and improves the continuous learning ability of the model.

3.2 Explore the Previous Memory Strategies

Sample replay. To achieve high performance in the joint space of old and new classes, the sample replay technique is used in FSCIL. Some methods store some real samples from previous sessions in a memory. In t-th session ($t > 0$), few samples from dataset $\mathcal{D}^{(t-1)}$ are sent to the memory $\mathcal{M}^{(t)}$, which alongside new added samples to finetune the model as

$$\mathcal{L} = \mathcal{L}_{ce}(\phi(\mathbf{x}_i), y_i), \ (\mathbf{x}_i, y_i) \in \{\mathcal{D}^{(t)} \cup \mathcal{M}^{(t)}\}. \tag{2}$$

These methods re-expose previous samples to the model during training, which is like a reminder to help the model retain previous knowledge when it learns new knowledge.

Prototype computing. Although using sample replay can recall old knowledge, it may develop biases for the model due to limited knowledge and is prone to exposing privacy, which restricts the ability to mitigate catastrophic forgetting. Inspired by ProtoNet [Snell *et al.*, 2017], some methods store prototypes as a memory for subsequent tasks as:

$$\bar{\mathbf{f}}_c = \frac{1}{|\mathcal{D}(y_i = c)|} \sum\nolimits_{\mathbf{x}_i \in \mathcal{D}(y_i = c)} \phi_b(\mathbf{x}_i), c \in C^{(t)}. \tag{3}$$

Each prototype is the mean of intermediate features, condensing the common pattern of each class within an embedding vector. The prototypes remain constant across incremental sessions to alleviate catastrophic forgetting.

Analyse. Figure 2 shows the performance of building a memory by sample replay and prototype computing. When using sample replay, the performance gradually increases with the number of samples, which indicates that more samples cover more knowledge. However, the selection problem of samples, resource consumption, and privacy issues cannot be ignored. When outlier samples are chosen for replay, it affects the model to generalize and retain old knowledge properly. Prototype computing preserves privacy and integrates knowledge efficiently. However, averaging features may cause the dilution of critical class-related information, which fails to capture the knowledge best suited for model training. Besides, they may be influenced by a few outlier samples with extreme values, especially when facing scarce data in incremental sessions. These outliers introduce biases and noise that can compromise the model performance. To address that, we further introduce a more available strategy to build the memory and a more efficient method to use the memory.

4 Methodology

As shown in Figure 3, our CD^2 framework for FSCIL begins with training a model with sufficient data in the base session. In incremental sessions, we finetune the classifier with the assistance of a memory. Before each incremental session, we generate a set by a dataset distillation module (DDM) as close as possible to critical knowledge and store it in the memory, which can be trained together with the current dataset to achieve the purpose of learning new concepts and preserving old concepts. In addition to the cross-entropy loss, we also employ a distillation constraint module (DCM) to mitigate catastrophic forgetting while balancing stability and flexibility to make better use of the generated data.

4.1 Dataset Distillation Module

Although previous memory strategies mitigate catastrophic forgetting effectively, they compress knowledge crudely.

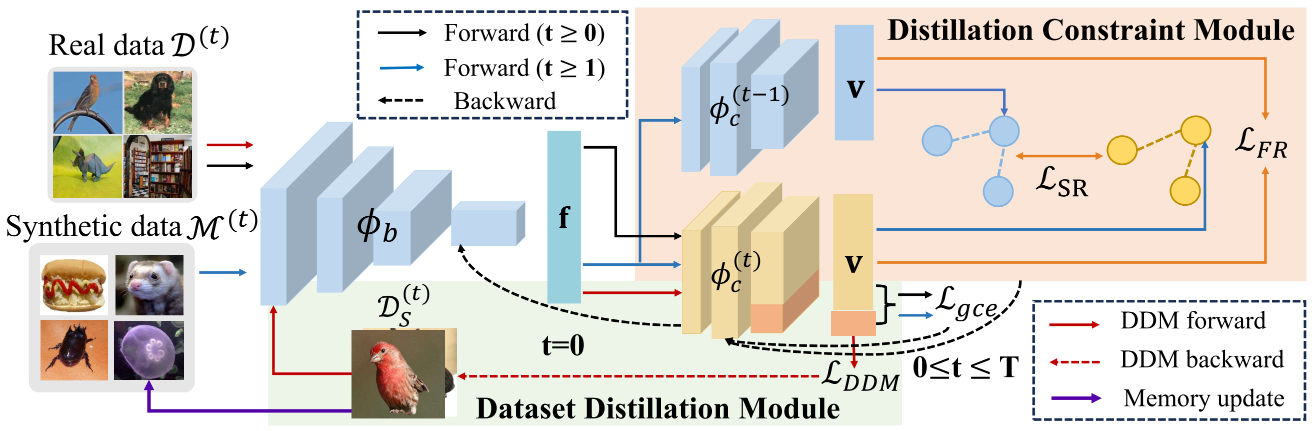

Figure 3: The framework **CD**^{2} generates memory by a **dataset distillation module** (DDM) and finetunes the model with a **distillation constraint module** (DCM), containing a backbone ϕ_b, a classifier ϕ_c, and an extra memory $\mathcal{M}^{(t)}$. During model training, we train the model on $\mathcal{D}^{(0)}$ in the base session ($t = 0$), then freeze the backbone and finetune the classifier with DCM on dataset $\mathcal{D}^{(t)}$ and the memory $\mathcal{M}^{(t)}$ in t-th incremental session ($t > 0$). And after model training for each session ($t < T$), we synthetic samples as $\mathcal{D}_S^{(t)}$ by DDM and send them to the memory $\mathcal{M}^{(t+1)}$.

These strategies lump together critical knowledge that enhances the discriminative ability of the model and redundant knowledge that contributes less to the model performance. Due to a lack of guidance in obtaining critical knowledge, these methods inadvertently downplay critical class-related knowledge and even overlook essential details. The dilution of critical knowledge not only impairs the model performance and discrimination ability, but also limits its potential for continuous learning and self-optimization from experience.

Based on this, we are oriented to effectively retain more critical old knowledge and propose a dataset distillation module (DDM). The core of the strategy is ***to choose critical class-related knowledge that is valid for incremental sessions***, enhancing the adaptability and recognition ability of the model under FSCIL settings. In this way, the model can keep and update knowledge that is critical to performance, while effectively weakening less important knowledge.

Inspired by dataset distillation [Zhao *et al.*, 2021; Liu and Wang, 2023], we synthesize samples for each class to compress critical knowledge. We randomly select K samples from $\mathcal{D}^{(t)}$ for each class and initialize the synthetic set $\mathcal{D}_S^{(t)} = \{(\mathbf{x}_m, y_m)|y_m \in C^{(t)}\}_{m=0}^{|C^{(t)}|\times K}$. While synthesizing, we freeze the whole model and update the synthetic set by class. The maximum mean discrepancy (**MMD**$(\cdot)$) [Gretton *et al.*, 2012] is used to estimate the distance between the real and synthetic data distribution as:

$$\begin{aligned}\mathbf{MMD}(\phi_b, \mathcal{D}_S, \mathcal{D}^{(t)}) &= \sup(E(\mathcal{D}_S) - E(\mathcal{D}^{(t)})) \\ &= (\frac{1}{|\mathcal{D}_S^{(t)}|}\sum \phi_b(\mathbf{x}_m) - \frac{1}{|\mathcal{D}^{(t)}|}\sum \phi_b(\mathbf{x}_i))^2,\end{aligned} \tag{4}$$

where sup is taking an upper bound and E is the expectation.

To synthesize samples more finely, we convert Eq. (4) to estimate the distance of synthetic set $\mathcal{D}_S^{(t)}$ and real dataset $\mathcal{D}^{(t)}$ and optimize the loss for each class as follows

$$\begin{aligned}\mathcal{L}_{DDM} &= \sup(\mathcal{D}_S^{(t)}(y_m = c) - \mathcal{D}^{(t)}(y_i = c)) \\ &= (\frac{1}{K}\sum_{y_m=c} \phi(\mathbf{x}_m) - \frac{1}{|\mathcal{D}^{(t)}(y_i = c)|}\sum_{y_i=c} \phi(\mathbf{x}_i))^2,\end{aligned} \tag{5}$$

where $c \in C^{(t)}$. The learned synthetic set can gain complementary information and summarize important knowledge from different samples of the same class, which helps improve the power of statistical analyses and ignores sampling errors. Meanwhile, it reduces the possibility of privacy leakage because it is difficult to recover the source data. Further, to reduce training consumption while gaining critical knowledge, we replace the dataset $\mathcal{D}^{(t)}$ with dataset $\mathcal{D}_R^{(t)}$ sampled from dataset $\mathcal{D}^{(t)}$ to perform the synthesis and calculation.

4.2 Distillation Constraint Module

Facing catastrophic forgetting, we involve a memory generated by DDM during training to recall old knowledge. Generally, a synthetic sample in the memory has a different distribution and contains more abundant knowledge than a real sample. The same treatment brings the covariate shift of old classes when adding new classes, and the classification loss cannot take full advantage of rich knowledge, impacting the model performance and the discrimination ability.

Based on this, we incorporate a distillation constraint module (DCM) in incremental sessions ***to limit the distribution change across continual sessions, thereby bridging the distribution gap between data in the memory and the dataset***, which facilitates the discriminative ability between old and new classes. The DCM consists of feature retention loss and structure retention loss, which ensure that old knowledge is as accurately retained as possible.

Feature retention loss (FR loss): The feature retention (FR) loss guarantees the consistency of features for old classes between the previous and the current session, thereby maintaining the integrity and coherence of processing data across different sessions. While calculating the FR loss, we get pairs of vectors $\{\mathbf{v}_m^{(t-1)}, \mathbf{v}_m^{(t)}\}_{m=0}^{|\mathcal{M}^{(t)}|}$ from the memory $\mathcal{M}^{(t)}$, where $\mathbf{v}_m^{(t-1)} = \phi_c^{(t-1)}(\mathbf{f}_m)$ and $\mathbf{v}_m^{(t)} = \phi_c^{(t)}(\mathbf{f}_m)$. And we constrain the first $C = |\sum_{i=0}^{t-1} C^{(i)}|$ elements of each pair of vectors to be consistent, which is formulated as

$$\mathcal{L}_{FR} = \frac{1}{\mathcal{M}^{(t)}} \sum_{(x_m,y_m)\in\mathcal{M}^{(t)}} \left|\mathbf{v}_m^{(t-1)} - \mathbf{v}_m^{(t)}[: C]\right|. \quad (6)$$

With the FR loss, the pairs of vectors in the previous model $\phi_c^{(t-1)}$ and the current model $\phi_c^{(t)}$ exhibit a tendency to convergence. And the FR loss constrains the location information, which enables the gap between real data and synthetic data as constant as possible to maintain the coherence and stability of knowledge transmission.

Structure retention loss (SR loss): Further, we hope to retain old knowledge more flexibly and elastically. Inspired by relational knowledge distillation (RKD) [Park *et al.*, 2019], we employ the structure retention (SR) loss to ensure the consistency of the structure distribution of synthetic samples in the previous session and the current session. RKD transfers structural information among a set of samples from the teacher model to the student model, endowing the student model with more flexibility to learn new knowledge. It constrains structural information by constraining the angle between triplets of samples $\{(x_a, x_b, x_c)\}$, which is as follows

$$\mathcal{L}_{RKD} = \sum_{\{(x_a,x_b,x_c)\}} |\cos\angle t_a t_b t_c - \cos\angle s_a s_b s_c|, \quad (7)$$

where $t_i = \psi(x_i)$ and $s_i = \zeta(x_i)$. Here, $\psi(x_i)$ is the teacher model and $\zeta(x_i)$ is the student model.

We incorporate the core idea of RKD into our CD^2, transferring the structural information of the memory (synthetic samples) in the feature space from the old model $\phi_c^{(t-1)}$ to the current model $\phi_c^{(t)}$. To constrain the feature structure, we formulate the SR loss based on a linear transformation between the features, which can be expressed as:

$$\mathcal{L}_{SR} = \frac{1}{\mathcal{M}^{(t)}} \sum_{(x_m,y_m)\in\mathcal{M}^{(t)}} \left|\mathbf{P}^{(t-1)} - \mathbf{P}^{(t)}[: C]\right|, \quad (8)$$

where $\mathbf{P}^{(t)} = \mathbf{v}_m \mathbf{V}^T$ is a vector of linear transformation and $\mathbf{V} = \{\phi_c^{(t)}(\mathbf{f}_m)|(x_m, y_m) \in \mathcal{M}^{(t)}\}$ is a matrix of output vectors in the memory $\mathcal{M}^{(t)}$. By minimizing this loss, we not only ensure the stability of the model when dealing with previous knowledge but also maintain its capacity for exploration and adaptation to new data and knowledge.

Distillation constraint module (DCM). The above two components form the distillation constraint module (DCM), maintaining a stable identification of old classes while being more flexible. The total loss can be summarized as follows:

$$\mathcal{L}_{DCM} = \alpha\mathcal{L}_{SR} + \beta\mathcal{L}_{FR}, \quad (9)$$

where α and β are factors. The FR loss functions to constrain location information, while the SR loss aims to preserve structural information. And these dual constraints help to stabilize the distribution changes across continuous learning sessions when incorporating new classes. The DCM is designed to subtly account for the two core requirements of stability and flexibility, which reduces the covariate shift.

4.3 Model Training & Memory

Fig. 3 illustrates the training process of our CD^2 in t-th session. According to different inputs, we divide the whole training into two stages: base learning and incremental learning.

Base learning: *During model training,* we train a model with dataset $\mathcal{D}^{(0)}$ with a sufficient amount of samples. For each training example $\mathbf{x}_i$, the model $\phi(\mathbf{x}_i)$ represents it into a final vector $\mathbf{v}_i$. In the base session, both the backbone $\phi_b(\mathbf{x}_i)$ and the classifier $\phi_c^{(0)}(\mathbf{f}_i)$ are jointly trained on $\mathcal{D}^{(0)}$ by minimizing the empirical risk loss as

$$\mathcal{L}_b = \frac{1}{|\mathcal{D}^{(0)}|} \sum_{(\mathbf{x}_i,y_i)\in\mathcal{D}^{(0)}} \mathcal{L}_{ce}(\phi^{(0)}(\mathbf{x}_i), y_i). \quad (10)$$

After model training, we build a small buffer as a memory to participate in subsequent training. We select the initial samples from dataset $\mathcal{D}^{(0)}$ for each class and generate samples as Eq. (5) to gain a set $\mathcal{D}_S^{(0)}$. And we send the set $\mathcal{D}_S^{(0)}$ to the memory $\mathcal{M}^{(1)}$.

Incremental learning: *During model training,* we freeze the backbone and finetune the classifier with dataset $\mathcal{D}^{(t)}$ and the memory $\mathcal{M}^{(t)}$. In t-th incremental session, we minimize the learning objective function as:

$$\begin{aligned}\mathcal{L}_i &= \mathcal{L}_{gce} + \mathcal{L}_{DCM} \\ &= \mathcal{L}_{gce} + \alpha\mathcal{L}_{SR} + \beta\mathcal{L}_{FR},\end{aligned} \quad (11)$$

where $\mathcal{L}_{gce}$ is the global classification loss as

$$\begin{aligned}\mathcal{L}_{gce} = &\frac{1}{|\mathcal{D}^{(t)}|} \sum_{(\mathbf{x}_i,y_i)\in\mathcal{D}^{(t)}} \mathcal{L}_{ce}(\phi^{(t)}(\mathbf{x}_i), y_i) \\ &+ \frac{1}{|\mathcal{M}^{(t)}|} \sum_{(\mathbf{x}_m,y_m)\in\mathcal{M}^{(t)}} \mathcal{L}_{ce}(\phi^{(t)}(\mathbf{x}_m), y_m).\end{aligned} \quad (12)$$

As the number of sessions increases, the ratio of previous classes to new classes gets larger, gradually making it more difficult to retain previous knowledge. The model should allocate more energy to the old knowledge as the session grows. Meanwhile, considering that the excessive proportion of the FR loss will affect the flexibility of the model, we only constrain the SR loss adaptively, which factors are set as $\alpha = \ln((-50/|\sum_{i=0}^{(t)} C^{(i)}|)^3 + 2)$.

After model training, we generate a synthetic set $\mathcal{D}_S^{(t)}$ from dataset $\mathcal{D}^{(t)}$ in a similar way as in base learning and add it to the memory $\mathcal{M}^{(t+1)}$ for subsequent fine-tuning.

5 Experiments

5.1 Experimental Setup

Datasets. We evaluate our CD^2 on three FSCIL benchmark datasets: CIFAR100 [Krizhevsky and Hinton, 2009], mini-ImageNet [Russakovsky *et al.*, 2015], and CUB200 [Wah

Method	Accuracy in each session(%)↑									Average	Average
	0	1	2	3	4	5	6	7	8	accuracy	improvement
iCaRl [Rebuffi *et al.*, 2017]	64.10	53.28	41.69	34.13	27.93	25.06	20.41	15.48	13.73	32.87	+35.69
EEIL [Castro *et al.*, 2018]	64.10	53.11	43.71	35.15	28.96	24.98	21.01	17.26	15.85	33.79	+34.77
SoftNet [Kang *et al.*, 2023a]	72.62	67.31	63.05	59.39	56.00	53.23	51.06	48.83	46.63	57.57	+11.10
MCNet [Ji *et al.*, 2023]	73.30	69.34	65.72	61.70	58.75	56.44	54.59	53.01	50.72	60.40	+8.27
GKEAL [Zhuang *et al.*, 2023]	74.01	70.45	67.01	63.08	60.01	57.30	55.50	53.39	51.40	61.35	+7.32
FACT[Zhou *et al.*, 2022b]	74.60	72.09	67.56	63.52	61.38	58.36	56.28	54.24	52.10	62.24	+6.43
C-FSCIL [Hersche *et al.*, 2022]	77.47	72.40	67.47	63.25	59.84	56.95	54.42	52.47	50.47	61.64	+7.03
MICS [Kim *et al.*, 2024]	78.18	73.49	68.97	65.01	62.25	59.34	57.31	55.11	52.94	63.62	+5.05
ALICE [Peng *et al.*, 2022]	79.00	70.50	67.10	63.40	61.20	59.20	58.10	56.30	54.10	63.21	+5.46
CABD [Zhao *et al.*, 2023b]	79.45	75.38	71.84	67.95	64.96	61.95	60.16	57.67	55.88	66.14	+2.53
OrCo [Ahmed *et al.*, 2024]	80.08	68.16	66.99	60.97	59.78	58.60	57.04	55.13	52.19	62.11	+6.56
WaRP [Kim *et al.*, 2023]	80.31	75.86	71.87	67.58	64.39	61.34	59.15	57.10	54.74	65.82	+2.85
NC-FSCIL [Yang *et al.*, 2023b]	82.52	76.82	73.34	69.68	66.19	62.85	60.96	59.02	56.11	67.50	+1.17
Revisting-FSCIL [Tang *et al.*, 2024]	82.90	76.30	72.90	67.80	65.20	62.00	60.70	58.80	56.60	67.02	+1.65
CD^2 (Ours)	**83.32**	**79.42**	**74.96**	**70.33**	**67.28**	**64.21**	**61.35**	**59.81**	**57.36**	**68.67**	-

Table 1: FSCIL performance on CIFAR100. "Average accuracy" means the average accuracy of all sessions and "Average improvement" calculates the improvement of our approach over other approaches. These approaches include class-incremental learning with FSCIL setting and FSCIL methods. The best results are in bold.

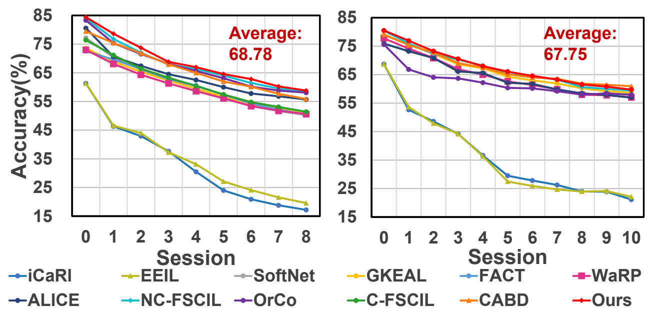

Figure 4: Performance curves of our method compared to recent SOTA methods on mini-ImageNet and CUB200. Left: mini-ImageNet. Right: CUB200. "Average" denotes the average accuracy of all sessions. Please refer to the appendix for more details.

et al., 2011] following previous settings [Liu *et al.*, 2022; Yang *et al.*, 2023b]. On CIFAR100 and mini-ImageNet, the 100 classes are organized into 60 base classes and 40 incremental classes. 60 base classes contain 500 training images for the training model. And 40 incremental classes are further structured in 8 different sets with a 5-*way* 5-*shot* setting. And 200 classes of CUB200 are organized into 100 base classes and 100 incremental classes in a 10-*way* 5-*shot* FSCIL setting for 10 incremental sessions.

Network. Following previous work [Liu *et al.*, 2022; Yang *et al.*, 2023b; Ahmed *et al.*, 2024], we employ ResNet [He *et al.*, 2016] as a backbone. We use ResNet12 without pretraining for CIFAR100 and MiniImageNet, and ResNet18 pre-trained on ImageNet for CUB200. We adopt a three-layer MLP block and a fully connected layer as a classifier.

Implementation details. Our model is optimized using SGD with momentum and adopts a cosine annealing strategy for the learning rate during training. In the base session, we train for 100 to 200 epochs while initializing a learning rate of 0.25 for CIFAR100 and mini-ImageNet, and 0.01 for CUB200. In each incremental session, we train for 100 to 300 iterations initializing a learning rate of 0.25 for CIFAR100 and MiniImageNet, and 0.001 for CUB200. Augmentations include random resizing, random flipping, Mixup [Zhang *et al.*, 2018], and CutMix [Yun *et al.*, 2019]. And we set $\beta = 0.1$ for model training. When using DDM, we train 1000 iterations initializing a learning rate of 0.2.

5.2 State-of-the-art Comparison

We evaluate our CD^2 on three public datasets and conduct a comparative analysis with some class-incremental learning methods with FSCIL setting and some state-of-the-art FSCIL methods using different memory strategies. Table 1 presents the results obtained on the CIFAR100, while Figure 4 illustrates the evaluation results on the mini-ImageNet and CUB200. The results demonstrate our method has superior performance across all three datasets, surpassing previous state-of-the-art methods.

Notably, our method achieves the highest average accuracy in three datasets. Firstly, our method outperforms class-incremental learning (CIL) methods(such as iCaRL [Rebuffi *et al.*, 2017], EEIL [Castro *et al.*, 2018]), which primarily focus on mitigating catastrophic forgetting in the context of CIL rather than FSCIL. In contrast to these methods, our approach is tailored specifically to tackle the challenges of catastrophic forgetting in FSCIL. And secondly, compared with some FSCIL methods, we obtain the best accuracy in all sessions on CIFAR100 and MiniImageNet and the best average accuracy on three datasets, maintaining the accuracy advantage. Our accuracy curve declines more slowly than other methods on CIFAR100 and MiniImageNet, particularly achieving improvements of 0.80% in the base session and $2.60 - 0.39\%$ in incremental sessions than NC-FSCIL [Yang *et al.*, 2023b] on CIFAR100. Although the accuracy curve appears dented on several sessions on CUB200, we still achieve the highest average accuracy. The excellent performance indicates the efficacy of our CD^2, and the memory facilitates the preservation of previous knowledge for subsequent tasks effectively.

$\mathcal{L}_{SR}$	$\mathcal{L}_{FR}$	Base	Prototype		DCM	
			First	Final	First	Final
			78.07	55.28	79.01	56.31
✓		83.32	78.33	55.72	79.24	56.83
	✓		78.57	55.45	79.37	56.06
✓	✓		**78.98**	**56.03**	**79.42**	**57.36**

Table 2: The effect of different memory strategies and the DCM on CIFAR100. "Base" denotes the accuracy of the base session. "First" and "Final" refer to the accuracy of the first and last incremental session, respectively. The best results are in bold.

5.3 Ablation Study

To validate the effectiveness of each component, we further study the effect of different memory strategies and the DCM based on CIFAR100. More studies are in the appendix.

The effect of different memory strategies. We compare the impact of different memory strategies on accuracy as shown in Table 2. Compared with prototype computing, CD^2 achieves higher performance in incremental sessions, which improves more than 0.44% in the first incremental session and more than 1.33% in the last incremental session. Different memory strategies give different knowledge to the model, which directly affects the update direction of the parameters and the final accuracy. DDM can capture the key knowledge and is different from real data, which gives the model a good foundation to learn knowledge continually.

The effect of the DCM. The DCM plays an important role in the whole training. As shown in Table 2, the model achieves higher accuracy when both SR loss and FR loss are applied. When using the SR loss alone, the model retains old knowledge stably, but may lead to displacement. When only using the FR loss, the model can retain old knowledge more accurately, but strict requirements limit the adaptation of the model to new classes. When SR and FR losses are used together, the model can subtly pursue both flexibility and accuracy, resulting in optimal overall performance.

5.4 Further Discussion

For a detailed analysis, we observe the number of synthetic sets and the visualization. More studies are in the appendix.

The effect of K. As shown in Table 3, the model gets an improvement in all sessions as K increases, which means the number of generated samples affects the accuracy of the model. As the sample size increases, the model can be exposed to more diverse data and situations, thus accumulating richer knowledge, which is directly reflected in the improved accuracy in all sessions. However, when $K \geqslant 2$, the number of samples has a negligible impact on the model performance. Considering the resource consumption and the performance, we set $K = 2$.

Representation visualization. Fig. 5 clearly shows the aggregation of classes by t-SNE visualization [Laurens and Hinton, 2008]. Firstly, the samples of the same class can be aggregated well except for some special cases, where the model can effectively capture common patterns of the same class and distinguish the intrinsic properties of the different classes. Secondly, novel classes have closer inter-class distances than

K	Base	First	Final	Average
1		78.85	56.67	67.74
2		79.42	57.36	68.67
3	83.32	79.36	57.13	68.59
4		79.45	56.92	68.23
5		**79.48**	**57.41**	**68.70**

Table 3: The effect of the number in synthetic samples, which is controlled by K. "Base" denotes the accuracy of the base session. "First" and "Final" refer to the accuracy of the first and last incremental session. "Average" is the average accuracy of all sessions. The best results are in bold.

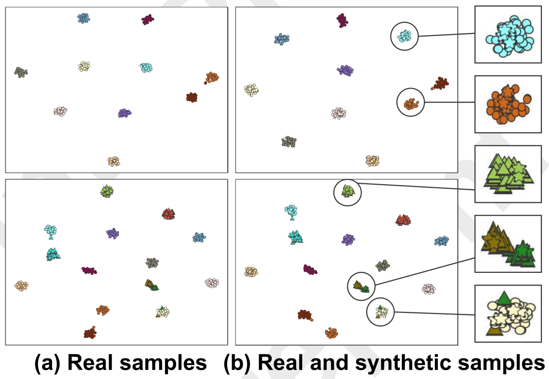

Figure 5: The t-SNE visualization of representations, which uses the base session and an incremental session on CIFAR-100 as an example. We randomly select 50 examples over 10 base classes and 5 incremental classes to show the effect. And we visualize all synthetic samples from all selected classes. Symbols like '●' and '▲' represent samples of base classes and incremental classes. '★' represents synthetic samples. (a) is visual features of real samples, and (b) is visual features of both real and synthetic samples. The top row shows the visualization of the base session, and the bottom row shows the visualization of the incremental session.

base classes, which means learning new data is more difficult than learning base data, resulting in the model being less sensitive to the differences between novel classes. Finally, synthetic samples are well aggregated in their classes, where the model is capable of learning from real data and effectively transferring this learning to the synthetic set. Meanwhile, this also indicates that the model successfully captures the key characteristics of the real data and saves them in simulated samples for subsequent tasks.

6 Conclusion

In this paper, we propose a framework CD^2 to support FSCIL, which contains a dataset distillation module (DDM) and a distillation constraint module (DCM). Inspired by dataset distillation, DDM synthesizes highly condensed samples with class-related clues, which can capture essential information. Furthermore, we introduce DCM to constrain distribution in incremental sessions, enabling more precise memory utilization by constraining feature alignment and structure preservation flexibly and stably, which mitigates catastrophic forgetting. Extensive experiments show that our approach achieves and even surpasses state-of-the-art performances.

Acknowledgements

This work was supported by the Pioneer R&D Program of Zhejiang Province (2024C01024) and the Open Research Project of National Key Laboratory of Science and Technology on Space-Born Intelligent Information Processing (TJ-02-22-01).

Contribution Statement

Daichi Zhang is the project lead of this work. Shiming Ge is the corresponding author of this work.

References

[Agarwal *et al.*, 2022] Aishwarya Agarwal, Biplab Banerjee, Fabio Cuzzolin, and Subhasis Chaudhuri. Semantics-driven generative replay for few-shot class incremental learning. In *ACM MM*, pages 5246–5254, 2022.

[Ahmed *et al.*, 2024] Noor Ahmed, Anna Kukleva, and Bernt Schiele. Orco: Towards better generalization via orthogonality and contrast for few-shot class-incremental learning. In *CVPR*, pages 28762–28771, 2024.

[Babakniya *et al.*, 2023] Sara Babakniya, Zalan Fabian, Chaoyang He, Mahdi Soltanolkotabi, and Salman Avestimehr. A data-free approach to mitigate catastrophic forgetting in federated class incremental learning for vision tasks. In *NeurIPS*, pages 66408–66425, 2023.

[Castro *et al.*, 2018] Francisco M. Castro, Manuel J. Marín-Jiménez, Nicolás Guil, Cordelia Schmid, and Karteek Alahari. End-to-end incremental learning. In *ECCV*, pages 241–257, 2018.

[Cazenavette *et al.*, 2022] George Cazenavette, Tongzhou Wang, Antonio Torralba, Alexei A. Efros, and Jun-Yan Zhu. Dataset distillation by matching training trajectories. In *CVPR*, pages 4749–4758, 2022.

[Chen and Lee, 2021] Kuilin Chen and Chi-Guhn Lee. Incremental few-shot learning via vector quantization in deep embedded space. In *ICLR*, 2021.

[Dong *et al.*, 2021] Songlin Dong, Xiaopeng Hong, Xiaoyu Tao, Xinyuan Chang, Xing Wei, and Yihong Gong. Few-shot class-incremental learning via relation knowledge distillation. In *AAAI*, pages 1255–1263, 2021.

[Du *et al.*, 2023] Jiawei Du, Yidi Jiang, Vincent Y. F. Tan, Joey Tianyi Zhou, and Haizhou Li. Minimizing the accumulated trajectory error to improve dataset distillation. In *CVPR*, pages 3749–3758, 2023.

[Gretton *et al.*, 2012] Arthur Gretton, Karsten M. Borgwardt, Malte J. Rasch, Bernhard Schölkopf, and Alexander J. Smola. A kernel two-sample test. *JMLR*, 13:723–773, 2012.

[He *et al.*, 2016] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, pages 770–778, 2016.

[Hersche *et al.*, 2022] Michael Hersche, Geethan Karunaratne, Giovanni Cherubini, Luca Benini, Abu Sebastian, and Abbas Rahimi. Constrained few-shot class-incremental learning. In *CVPR*, pages 9047–9057, 2022.

[Hinton *et al.*, 2015] Geoffrey E. Hinton, Oriol Vinyals, and Jeffrey Dean. Distilling the knowledge in a neural network. *arXiv*, abs/1503.02531, 2015.

[Ji *et al.*, 2023] Zhong Ji, Zhishen Hou, Xiyao Liu, et al. Memorizing complementation network for few-shot class-incremental learning. *IEEE TIP*, 32:937–948, 2023.

[Kang *et al.*, 2023a] Haeyong Kang, Jaehong Yoon, Sultan Rizky Hikmawan Madjid, Sung Ju Hwang, and Chang D. Yoo. On the soft-subnetwork for few-shot class incremental learning. In *ICLR*, 2023.

[Kang *et al.*, 2023b] Mengxue Kang, Jinpeng Zhang, Jinming Zhang, Xiashuang Wang, Yang Chen, Zhe Ma, and Xuhui Huang. Alleviating catastrophic forgetting of incremental object detection via within-class and between-class knowledge distillation. In *ICCV*, pages 18848–18858, 2023.

[Kim *et al.*, 2022] Jang-Hyun Kim, Jinuk Kim, Seong Joon Oh, Sangdoo Yun, Hwanjun Song, Joonhyun Jeong, Jung-Woo Ha, and Hyun Oh Song. Dataset condensation via efficient synthetic-data parameterization. In Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvári, Gang Niu, and Sivan Sabato, editors, *ICML*, 2022.

[Kim *et al.*, 2023] Do-Yeon Kim, Dong-Jun Han, Jun Seo, and Jaekyun Moon. Warping the space: Weight space rotation for class-incremental few-shot learning. In *ICLR*, 2023.

[Kim *et al.*, 2024] Solang Kim, Yuho Jeong, Joon Sung Park, and Sung Whan Yoon. MICS: midpoint interpolation to learn compact and separated representations for few-shot class-incremental learning. In *WACV*, pages 2225–2234, 2024.

[Krizhevsky and Hinton, 2009] Alex Krizhevsky and Geoffrey Hinton. Learning multiple layers of features from tiny images. *Handbook of Systemic Autoimmune Diseases*, 1, 2009.

[Kukleva *et al.*, 2021] Anna Kukleva, Hilde Kuehne, and Bernt Schiele. Generalized and incremental few-shot learning by explicit learning and calibration without forgetting. In *ICCV*, pages 9000–9009, 2021.

[Laurens and Hinton, 2008] Van Der Maaten Laurens and Geoffrey Hinton. Visualizing data using t-sne. *JMLR*, 9:2579–2605, 2008.

[Li *et al.*, 2024] Dasen Li, Zhendong Yin, Yanlong Zhao, Jiqing Li, and Hongjun Zhang. Rehearsal-based class-incremental learning approaches for plant disease classification. *Comput. Electron. Agric.*, 224:109211, 2024.

[Liu and Wang, 2023] Songhua Liu and Xinchao Wang. Few-shot dataset distillation via translative pre-training. In *ICCV*, pages 18608–18618, 2023.

[Liu *et al.*, 2022] Huan Liu, Li Gu, Zhixiang Chi, Yang Wang, Yuanhao Yu, Jun Chen, and Jin Tang. Few-shot class-incremental learning via entropy-regularized data-free replay. In *ECCV*, pages 146–162, 2022.

[Liu *et al.*, 2023a] Binghao Liu, Boyu Yang, Lingxi Xie, Ren Wang, Qi Tian, and Qixiang Ye. Learnable distribution calibration for few-shot class-incremental learning. *IEEE TPAMI*, 45:12699–12706, 2023.

[Liu *et al.*, 2023b] Ping Liu, Xin Yu, and Joey Tianyi Zhou. Meta knowledge condensation for federated learning. In *ICLR*, 2023.

[Park *et al.*, 2019] Wonpyo Park, Dongju Kim, Yan Lu, and Minsu Cho. Relational knowledge distillation. In *CVPR*, pages 3967–3976, 2019.

[Peng *et al.*, 2022] Can Peng, Kun Zhao, Tianren Wang, Meng Li, and Brian C. Lovell. Few-shot class-incremental learning from an open-set perspective. In *ECCV*, pages 382–397, 2022.

[Rebuffi *et al.*, 2017] Sylvestre-Alvise Rebuffi, Alexander Kolesnikov, Georg Sperl, and Christoph H. Lampert. iCaRL: Incremental classifier and representation learning. In *CVPR*, pages 5533–5542, 2017.

[Russakovsky *et al.*, 2015] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael S. Bernstein, Alexander C. Berg, and Li Fei-Fei. Imagenet large scale visual recognition challenge. *IJCV*, 115:211–252, 2015.

[Shen *et al.*, 2024] Xiaobo Shen, Peizhuo Song, Yun-Hao Yuan, and Yuhui Zheng. Distributed manifold hashing for image set classification and retrieval. In *AAAI*, pages 4802–4810, 2024.

[Snell *et al.*, 2017] Jake Snell, Kevin Swersky, and Richard Zemel. Prototypical networks for few-shot learning. In *NeurIPS*, pages 4080–4090, 2017.

[Such *et al.*, 2020] Felipe Petroski Such, Aditya Rawal, Joel Lehman, Kenneth O. Stanley, and Jeffrey Clune. Generative teaching networks: Accelerating neural architecture search by learning to generate synthetic training data. In *ICML*, 2020.

[Sun *et al.*, 2024] Le Sun, Mingyang Zhang, Benyou Wang, and Prayag Tiwari. Few-shot class-incremental learning for medical time series classification. *IEEE J-BHI*, 28(4):1872–1882, 2024.

[Tang *et al.*, 2024] Yu-Ming Tang, Yi-Xing Peng, Jingke Meng, and Wei-Shi Zheng. Rethinking few-shot class-incremental learning: Learning from yourself. In *ECCV*, 2024.

[Tao *et al.*, 2020] Xiaoyu Tao, Xiaopeng Hong, Xinyuan Chang, Songlin Dong, Xing Wei, and Yihong Gong. Few-shot class-incremental learning. In *CVPR*, pages 12180–12189, 2020.

[Wah *et al.*, 2011] Catherine Wah, Steve Branson, Peter Welinder, et al. The caltech-ucsd birds-200-2011 dataset. Technical Report CNS-TR-2011-001, California Institute of Technology, 2011.

[Wang *et al.*, 2018] Tongzhou Wang, Jun-Yan Zhu, Antonio Torralba, and Alexei A. Efros. Dataset distillation. *arXiv*, abs/1811.10959, 2018.

[Wei *et al.*, 2024] Tong Wei, Bo-Lin Wang, and Min-Ling Zhang. EAT: towards long-tailed out-of-distribution detection. In *AAAI*, pages 15787–15795, 2024.

[Yang *et al.*, 2023a] Boyu Yang, Mingbao Lin, Yunxiao Zhang, Binghao Liu, Xiaodan Liang, Rongrong Ji, and Qixiang Ye. Dynamic support network for few-shot class incremental learning. *IEEE TPAMI*, 45:2945–2951, 2023.

[Yang *et al.*, 2023b] Yibo Yang, Haobo Yuan, Xiangtai Li, Zhouchen Lin, Philip Torr, and Dacheng Tao. Neural collapse inspired feature-classifier alignment for few-shot class-incremental learning. In *ICLR*, 2023.

[Yang *et al.*, 2024] William Yang, Ye Zhu, Zhiwei Deng, and Olga Russakovsky. What is dataset distillation learning? In *ICML*, 2024.

[Yun *et al.*, 2019] Sangdoo Yun, Dongyoon Han, Sanghyuk Chun, Seong Joon Oh, Youngjoon Yoo, and Junsuk Choe. Cutmix: Regularization strategy to train strong classifiers with localizable features. In *ICCV*, pages 6022–6031, 2019.

[Zhang *et al.*, 2018] Hongyi Zhang, Moustapha Cissé, Yann N. Dauphin, and David Lopez-Paz. Mixup: Beyond empirical risk minimization. In *ICLR*, 2018.

[Zhang *et al.*, 2024] Hansong Zhang, Shikun Li, Pengju Wang, Dan Zeng, and Shiming Ge. M3D: dataset condensation by minimizing maximum mean discrepancy. In *AAAI*, pages 9314–9322, 2024.

[Zhao *et al.*, 2021] Bo Zhao, Konda Reddy Mopuri, and Hakan Bilen. Dataset condensation with gradient matching. In *ICLR*, 2021.

[Zhao *et al.*, 2023a] Ganlong Zhao, Guanbin Li, Yipeng Qin, and Yizhou Yu. Improved distribution matching for dataset condensation. In *CVPR*, pages 7856–7865, 2023.

[Zhao *et al.*, 2023b] Linglan Zhao, Jing Lu, Yunlu Xu, et al. Few-shot class-incremental learning via class-aware bilateral distillation. In *CVPR*, pages 11838–11847, 2023.

[Zhou *et al.*, 2022a] Da-Wei Zhou, Fu-Yun Wang, Han-Jia Ye, et al. Forward compatible few-shot class-incremental learning. In *CVPR*, pages 9036–9046, 2022.

[Zhou *et al.*, 2022b] Da-Wei Zhou, Fu-Yun Wang, Han-Jia Ye, Liang Ma, Shiliang Pu, and De-Chuan Zhan. Forward compatible few-shot class-incremental learning. In *CVPR*, pages 9036–9046, 2022.

[Zhou *et al.*, 2022c] Yongchao Zhou, Ehsan Nezhadarya, and Jimmy Ba. Dataset distillation using neural feature regression. In *NeurIPS*, pages 9813–9827, 2022.

[Zhu *et al.*, 2022] Juntao Zhu, Guangle Yao, Wenlong Zhou, Guiyu Zhang, Wang Ping, and Wei Zhang. Feature distribution distillation-based few shot class incremental learning. In *PRAI*, pages 108–113, 2022.

[Zhuang *et al.*, 2023] Huiping Zhuang, Zhenyu Weng, Run He, Zhiping Lin, and Ziqian Zeng. GKEAL: gaussian kernel embedded analytic learning for few-shot class incremental task. In *CVPR*, pages 7746–7755, 2023.