Few-shot Class-incremental Learning for Classification and Object Detection: A Survey

Jinghua Zhang , Li Liu, Olli Silvén, Matti Pietikäinen, Dewen Hu

Abstract—Few-shot Class-Incremental Learning (FSCIL) presents a unique challenge in Machine Learning (ML), as it necessitates the Incremental Learning (IL) of new classes from sparsely labeled training samples without forgetting previous knowledge. While this field has seen recent progress, it remains an active exploration area. This paper aims to provide a comprehensive and systematic review of FSCIL. In our in-depth examination, we delve into various facets of FSCIL, encompassing the problem definition, the discussion of the primary challenges of unreliable empirical risk minimization and the stability-plasticity dilemma, general schemes, and relevant problems of IL and Few-shot Learning (FSL). Besides, we offer an overview of benchmark datasets and evaluation metrics. Furthermore, we introduce the Few-shot Class-incremental Classification (FSCIC) methods from data-based, structure-based, and optimization-based approaches and the Few-shot Class- incremental Object Detection (FSCIOD) methods from anchor-free and anchor-based approaches. Beyond these, we present several promising research directions within FSCIL that merit further investigation.

Index Terms—Incremental learning, continual learning, lifelong learning, class-incremental learning, catastrophic forgetting, few-shot learning, few-shot class-incremental learning, deep learning, image classification

1 Introduction

Over the last decade, Deep Neural Networks (DNNs) have gone through several distinct developmental phases: from architectural engineering based on supervised learning as demonstrated by AlexNet [1] and ResNet [2], to the combined strategy of supervised pre-training and fine-tuning, with Transformer-based BERT [3] being a prime example. This progress further extended to a fusion of self-supervised or semi-supervised pre-training with prompt engineering, as demonstrated by the GPT series [4]. These advancements have consistently expanded algorithmic performance boundaries and opened up new application possibilities. However, it's essential to recognize that these DNN achievements have heavily relied on a huge amount of high-quality data, expensive computing hardware, and excellent DNN architectures that are costly to obtain.

DNN learning paradigms are primarily designed for static tasks within a closed-world setting, and it has inherent limitations. Firstly, these models cannot retain previously acquired knowledge and learn new knowledge over time. Specifically, once they are trained on a particular dataset, they often require retraining from scratch when confronted with new tasks or data distributions. Additionally, the process of retraining involves storing vast amounts of old data and updating models, leading to additional computational and storage costs. Such a learning paradigm has at least the following major issues:

- Capability and Application Limitations: These systems are optimized for specific tasks they've been trained on, making them ill-suited for dynamic situations.
- Purely Data-Driven Gap: Unlike humans, who learn efficiently with few examples and exhibit lifelong adaptability, these systems rely heavily on vast data and lack the versatility and retention inherent to human learning.
- Efficiency and Sustainability Issues: These data and energy-intensive systems require frequent retraining for new data or tasks, increasing computational resource strain and carbon footprint.
- Privacy and Security Concerns: The dynamic world exposes these systems to heightened security risks in novel scenarios. Moreover, retaining heightens the risk of data breaches, raising privacy alarms.

IL, also termed continual or lifelong learning, enables systems to learn new tasks over time while maintaining previous knowledge [5, 6, 7], aiming to replicate human learning abilities [5]. This field has seen growing interest recently, prompting numerous studies and surveys [5, 7, 8, 9, 10, 11]. The development trend in IL is summarized by the count of academic papers from major conferences and journals, as shown in our collection and the *Awesome-Incremental-Learning* resource[1], and depicted in Fig. 1. Class-incremental Learning (CIL) is notably prominent, addressing key challenges in real-world scenarios where models should adapt to new classes without forgetting existing ones.

As an important subset of CIL, FSCIL has experienced significant growth over the past four years, shown in Fig. 1. It is specifically designed to address the challenges of learning new classes with limited data. This learning paradigm demands that the model retains previously acquired knowledge while continually incorporating new classes, all while dealing with the constraints of limited annotated samples for each class [12, 13, 14]. Unlike conventional CIL, FSCIL faces more complex challenges, such as preventing catastrophic forgetting and mitigating overfitting due to sample scarcity. FSCIL seeks to emulate human learning efficiency with minimal data and maintain knowledge over time, making it highly relevant for real-world settings with limited,

J. Zhang (zhangjingh@foxmail.com) and Dewen Hu (dwhu@nudt.edu.cn) are with the College of Intelligence Science and Technology, National University of Defense Technology (NUDT), Changsha, China. Jinghua Zhang is also with the Center for Machine Vision and Signal Analysis (CMVS), University of Oulu, Finland. Li Liu (dreamliu2010@gmail.com) is with the College of Electronic Science and Technology, NUDT, Changsha, China. Olli Silvén (olli.silven@oulu.fi) and Matti Pietikäinen (matti.pietikainen@oulu.fi) is with CMVS, University of Oulu, Finland.
Corresponding authors: Dewen Hu and Li Liu
This work was partially supported by National Key Research and Development Program of China No.2021YFB3100800, the Academy of Finland under grant 331883, the National Natural Science Foundation of China under grant 62036013 and 62376283, and the Key Stone grant (JS2023-03) of the NUDT.

1. https://github.com/xialeiliu/Awesome-Incremental-Learning#2023

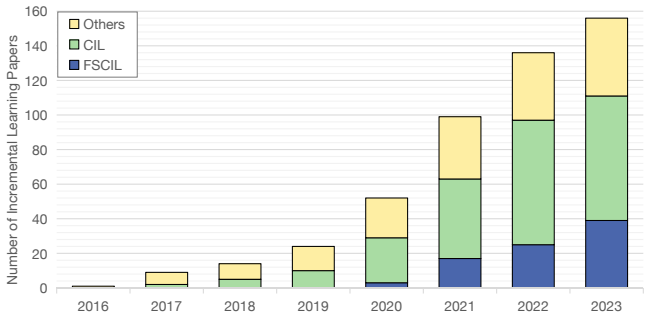


Fig. 1. IL publications from 2016 to 2023. It is observed that CIL research has become predominant in the field of IL over time, due to its practical value. Concurrently, FSCIL shows a steady rise, mirroring the growing requirement of CIL with limited data.

evolving data. To highlight its practical importance, we provide a concise summary of FSCIL's practical significance:

- Adaptation to Dynamic World: FSCIL empowers models to acquire new classes while retaining previous knowledge, a critical capability for effectively adapting to a dynamically changing world.
- High Data Efficiency: FSCIL can mitigate the necessity for extensive sample labeling, providing advantages in situations with limited data and high labeling costs.
- Environmental Sustainability: FSCIL promotes sustainability by requiring fewer computational and storage resources than traditional methods, a crucial benefit in resource-limited environments.
- Data Security and Privacy: FSCIL reduces the need to retain extensive historical data, thereby aligning with data security and privacy requirements.
- Versatile Applications: FSCIL is applicable in various fields, especially where data is limited, labeling is costly, and frequent class updates are needed

Although there has been some progress in the field of FSCIL and some representative works [5, 13, 14, 15, 16] have emerged, it is yet in its development stage. The key milestones from 2020 to the present are illustrated in Fig. 2. Current methods still have a gap to meet the practical applications. Therefore, it is imperative to systematically review the latest developments in this field, identify the core challenges and open questions that hinder its development, and determine the promising future direction. Nevertheless, most of the research on FSCIL is still quite dispersed, and this field needs a systematic and comprehensive survey. It has inspired our survey, which aims to fill the gap. Since it is an ML problem proposed in the field of computer vision in recent years and most of the research work is based on the deep learning algorithm, the scope discussed in our paper is mainly the deep FSCIL algorithm in the field of computer vision, which includes primarily classification and object detection tasks.

Despite existing surveys on FSL [27, 28, 29, 30] and IL [5, 6, 7, 11, 31, 32, 33, 34, 35], there is a clear lack of systematic and comprehensive surveys specifically on FSCIL. Existing FSL surveys primarily focus on tackling ML problems with limited data. While they provide systematic classifications of FSL methods, they do not touch upon the issue of FSCIL. Similarly, IL surveys mostly focus on ML problems in a continual learning scenario. For instance, *De Lange et al.* [7] conducted a systematic review of DR, regularization, and parameter isolation, along with comparative experiments, while *Zhou et al.* [34] summarized the CIL problem from perspectives like DR, Data Regularization, and Dynamic Networks. However, none of these works systematically review FSCIL. Although some surveys briefly mention FSCIL [32, 35], they only provide a short introduction to the concept and a few studies without offering a thorough or systematic analysis. Although *Tian et al.* [36] recently conducted a review of FSCIL, its introduction to FSCIL is not in-depth and comprehensive enough.

In this regard, we summarize existing surveys in Tab. 1 and systematically describe the uniqueness of our paper to highlight its unique contributions. To address the shortcomings in FSCIL research, we systematically summarize the field from various aspects, including definition, challenges, general schemes, related problems, datasets, metrics, methods, performance comparisons, and future directions. Our contributions are:

- Our survey offers a systematic and comprehensive review of classification and object detection methods in FSCIL.
- We cover problem definition, core challenges, general schemes, related ML problems, benchmark datasets, and evaluation metrics in detail.
- A structured taxonomy is offered for FSCIL, discussing classification methods from data, structure, and optimization perspectives, and detection methods from anchor-based and anchor-free perspectives.
- Valuable insights and outlooks in FSCIL are discussed.

The paper structure is as follows: Sec. 2 presents a detailed overview of FSCIL, including its definition, challenges, general frameworks, and its relationship with relevant problems. Sec. 3 discusses popular FSCIL datasets and evaluation metrics. Sec. 4 examines FSCIC methods from data, structure, and optimization perspectives, and Sec. 5 covers FSCIOD methods from anchor-based and anchor-free viewpoints. The paper concludes in Sec. 6 with a summary and future directions.

2 Background

2.1 Problem Definition

The FSCIL aims to learn an ML model that can continuously learn knowledge from a sequence of new classes with only a few labeled training samples while preserving the knowledge gained from previous classes [13, 14, 37, 38]. Taking the classification task as an example, Fig. 3(a) offers an overview of the general setting for FSCIL, including the setting of training data, the model learning process, and the evaluation setting.

Setting: As shown in Fig. 3(a), the data stream used in FSCIL contains a base session and a sequence of new sessions. The training datasets in these sessions can be denoted by $\{D^0_{train}, D^1_{train}, \cdots, D^B_{train}\}$, where B is the number of new sessions. The base training dataset generally contains sufficient labeled samples from the distribution D^0_t, and it can be formulated by $D^0_{train} = \{(x_i, y_i)\}_{i=1}^{n_0}$, where n_0 is the number of training samples in the base session, x_i is a training sample from class $y_i \in Y_0$, and Y_0 is the corresponding label space of D^0_{train}. Differently, the training dataset in each new session is in the form of $N-$way $K-$shot, where $N-$way means that the training set contains N classes and $K-$shot means each class contains K labeled samples. It can be formulated as $\forall$ integer $b \in [1, B], D^b_{train} = \{(x_i, y_i)\}_{i=1}^{N\times K}$. Note that the classes in different sessions do not intersect, *i.e.*, $\forall$ integer $p, q \in [0, B]$ and $p \neq q$, $Y_p \cap Y_q = \emptyset$.

Model: During the training session b, the dataset D^b_{train} is accessible, and the original complete training datasets from

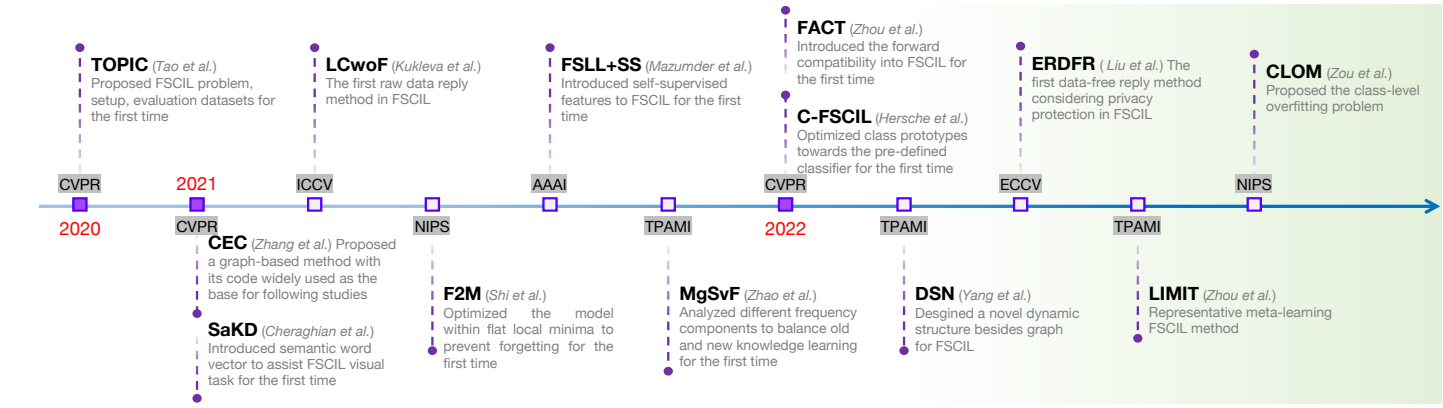


Fig. 2. A chronological overview of some representative FSCIL methods. FSCIL was first carried out by TOPIC [13]. CEC [14] was widely used as a base for subsequent studies. SaKD [17] integrated semantic word vectors into FSCIL, offering a reference for applying language-image models in the future. LCwoF [18] and ERDFR [19] proposed distinct Data Replay (DR) strategies. F2M [20] introduced a novel approach by constraining optimization within flat local minima. FSLL+SS [21] introduced the semi-supervised features to FSCIL for the first time. MgSvF [22] analyzed and utilized different frequency components to balance the old and new knowledge. FACT [23] introduced a fresh perspective by advocating forward compatibility in FSCIL. C-FSCIL [24] pre-defined classifiers to guide model optimization. DSN [25] offered a novel dynamic structure for FSCIL. LIMIT [16] proposed a representative meta-learning paradigm for FSCIL. CLOM [26] pointed out the issue of class-level overfitting made by metric learning in FSCIL.

TABLE 1
A summary and comparison of the primary surveys in the fields of FSL, IL, and FSCIL.

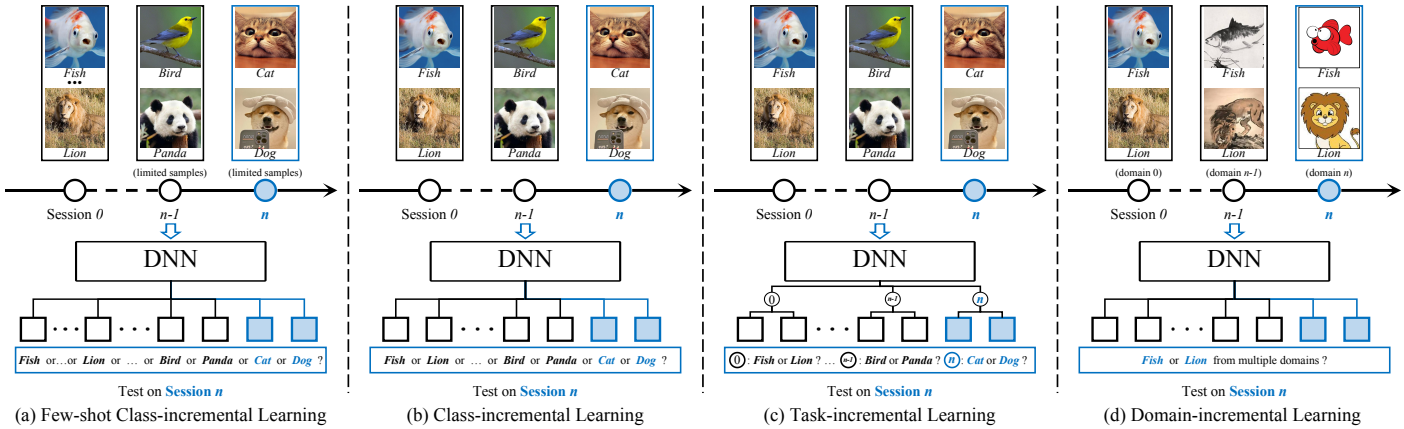
Topic	Year	Venue	Title	Main Content
FSL	2020	ACM CS	Generalizing from a Few Examples: A Survey on Few-shot Learning [27]	This paper clearly introduces FSL, including its definition, challenges, and main methods from data, model, and algorithm perspectives. However, it lacks a summary of datasets and performance.
	2022	ACM SC	Few-Shot Object Detection: A Survey [29]	This paper focuses on few-shot object detection. It reviews related methods based on data augmentation, transfer learning, metric learning, and meta-learning. However, it lacks a summary of challenges and relevant problems.
	2022	TPAMI	A Survey of Self-Supervised and Few-Shot Object Detection [30]	This paper reviews few-shot object detection methods in self-supervised and few-shot object detection, highlighting the potential of technique fusion. However, it lacks a summary of challenges and relevant problems.
	2023	PR	A survey on machine learning from few samples [28]	This paper categorizes FSL approaches into generative and discriminative models. It also discusses emerging FSL topics and its various applications. However, it lacks a summary of challenges and relevant problems.
IL	2019	Neural Networks	Continual Lifelong Learning with Neural Networks: A Review [6]	This paper discusses the challenges and approaches in lifelong learning, which involves continuous knowledge acquisition and application, and emphasizes the need to address catastrophic forgetting. But it lacks a summary of relevant problems and performance.
	2020	Information Fusion	Continual Learning for Robotics: Definition, Framework, Learning Strategies, Opportunities and Challenges [31]	This paper reviews continual learning, highlights the need for stable real-world algorithms, summarizes existing benchmarks and metrics, and proposes a framework for evaluating approaches across both robotics and non-robotics fields. However, it lacks a summary of performance.
	2021	Neural Networks	A Comprehensive Study of Class Incremental Learning Algorithms for Visual Tasks [32]	This paper focuses on CIL, providing the taxonomy of regularization approaches, dynamic architectures, and complementary learning systems and memory replay. However, it lacks a summary of challenges and relevant problems.
	2022	TPAMI	A Continual Learning Survey: Defying Forgetting in Classification Tasks [5]	This paper summarizes Task-incremental Learning (TIL) and compares 11 advanced methods. It also analyzes the impact of model parameters, task order, and resource requirements. However, it lacks a summary of challenges and relevant problems.
	2022	TPAMI	Class-Incremental Learning: Survey and Performance Evaluation on Image Classification [7]	This paper surveys CIL methods for image classification. It provides experimental evaluations on 13 methods and explores various scenarios. However, it lacks a summary of relevant problems.
	2022	Neuro-Computing	Online Continual Learning in Image Classification: An Empirical Survey [33]	This study reviews online continual learning in image classification and compares advanced methods. However, it lacks a summary of challenges and relevant problems.
	2024	TPAMI	A Comprehensive Survey of Continual Learning: Theory, Method and Application [35]	This survey explores continual learning from theoretical foundations, methods, and applications, but it lacks a summary of relevant problems, datasets, and performance.
	2024	TPAMI	Class-Incremental Learning: A Survey [34]	This paper categorizes methods into data-centric, model-centric, and algorithm-centric, evaluating 16 methods and advocating for fair comparisons. However, it lacks a summary of challenges and relevant problems.
FSCIL	2023	Neural Networks	A Survey on Few-Shot Class-Incremental Learning [36]	This survey focuses on FSCIL. However, the taxonomy and summary of relevant problems are not systematic. It also lacks a summary of the challenges. Discussion about object detection and the future direction is not in-depth.
	2023	**Ours**	**Few-shot Class-incremental Learning: A Survey**	**Our paper focuses on FSCIL in classification and object detection tasks, clarifies the challenge and relationship with relevant problems, and provides a systemical taxonomy from data-based, structure-based, and optimization-based approaches. We also summarize the performance of existing methods and point out potential directions.**

previous sessions are unavailable (note: some methods like DR or KD may store a few old samples to revisit old knowledge). The FSCIL model must learn new classes from D^b_{train} while maintaining performance on old classes, *i.e.*, minimizing the expected risk $\mathcal{R}(f, b)$ on all the seen classes [5, 16]. This process is formulated as follows:

$$\mathbb{E}_{(x_i,y_i)\sim D^0_t \cup \cdots D^b_t} \left[L\left(f\left(x_i; \mathcal{D}^b_{train}, \boldsymbol{\theta}^{b-1} \right), y_i \right) \right], \quad (1)$$

where the current FSCIL algorithm f aims to built the new model based on the dataset $\mathcal{D}^b_{train}$ and previous model parameters $\boldsymbol{\theta}^{b-1}$ and minimize the loss L on all seen classes [16]. Because the datasets in FSCIL are continuously updated, the expected risk on every new session should be optimized, *i.e.*, $\sum_{b=1}^{B} \mathcal{R}(f, b)$ should be optimized.

Evaluation: The testing datasets in FSCIL sessions can be de-



	Training data		Test on Session *i*
	Session 0	**Session *i***	
FSCIL	Sufficent base classes with enough samples	Limited samples for each class	Evaluated on all seen classes
CIL (typical)	Base classes with enough samples	Novel classes with enough samples	Evaluated on all seen classes
TIL (typical)	Base classes with enough samples	Novel classes with enough samples	Evaluated on all seen classes with knowing task identification
DIL (typical)	Base classes with enough samples from domain *0*	Base classes with enough samples from domain *i*	Evaluated on base classes with data from multiple domains

(e) Summary of the difference between different incremental learning fields

Fig. 3. The general settings of different IL tasks. Specifically, (a) shows the setting of FSCIL, (b) is the CIL, (c) represents the setting of TIL, and (d) illustrates the Domain-incremental Learning (DIL). FSCIL can be viewed as a subdomain of CIL, where the base session usually has sufficient training data, and the incremental sessions are formed in the $N-$way $K-$shot format. TIL differs from CIL because the session identity is known during model training and testing. In contrast, DIL maintains the same classification tasks, but the data across different sessions comes from different domains. Note that "session" may also be called "task" in other literature.

noted by $\{D^0_{test}, \cdots, D^B_{test}\}$, which shares the same label space as their corresponding training datasets. For the evaluation in session b, the FSCIL model needs to be evaluated by the joint testing datasets, which encompass all the testing datasets from the current and all preceding sessions, denoted as $D^0_{test} \cup \cdots \cup D^b_{test}$. This measure helps quantify the model's performance across all classes it has encountered up to that point.

2.2 Core Challenges

FSCIL faces significant challenges, notably the unreliable empirical risk minimization and the stability-plasticity dilemma. In FSCIL sessions, limited supervised data mean empirical risk fails to accurately represent expected risk, decreasing model generalization and increasing overfitting risks. Moreover, as new classes are continually added, old knowledge can be easily forgotten and overwritten by new knowledge. This leads to catastrophic forgetting. Otherwise, intransigence may occur. Therefore, balancing model stability and plasticity is another core challenge. This section provides the details of these challenges.

2.2.1 Unreliable Empirical Risk Minimization

In FSCIL, unreliable empirical risk minimization, where the model is trained to minimize prediction errors on the training data, poses a major challenge. This approach doesn't ensure strong generalization on test data, especially with limited training samples. In FSCIL, each session's training dataset follows an $N-$way $K-$shot format, often leading to a significant discrepancy between empirical and expected risks due to inadequate samples for new classes. This gap can result in overfitting, where the model excels on training data but underperforms on testing data, compromising its generalization ability [27, 39].

In contrast to conventional FSL, FSCIL not only grapples with the issue of scarce samples but is also confronted with the challenge posed by the continual increase in classes. Continuous unreliable empirical risk minimization in successive sessions may hinder the model's convergence to an ideal state, questioning not only the reliability of the model formed in the current incremental session but also presenting a challenge in maintaining model stability in the subsequent incremental session. This issue becomes particularly pronounced when dealing with multiple incremental classes with limited training samples [20].

To elaborate on this challenge, we introduce essential concepts of empirical risk minimization [27, 40, 41]. For a learning task with dataset $D = \{D_{train}, D_{test}\}$, where $p(x, y)$ denotes the joint probability distribution of data x and label y, and f_o is the optimal hypothesis from x to y, *i.e.*, the function that minimizes the expected risk. Specifically, given a hypothesis f, the expected risk $\mathcal{R}(f)$, which measures the loss concerning $p(x, y)$, is formulated as:

$$\mathcal{R}(f) = \int L(f(x), y)\, dp(x, y) = \mathbb{E}[L(f(x), y)], \tag{2}$$

and f_o can be explained as:

$$f_o = \arg\min_f \mathcal{R}(f). \tag{3}$$

As $p(x, y)$ is unknown, the empirical risk, which is the average loss value obtained on the training dataset D_{train} of I samples, is generally used as a proxy of $\mathcal{R}(f)$ for minimization. Specifically, empirical risk can be formulated as:

$$\mathcal{R}_I(f) = \frac{1}{I}\sum_{i=1}^{I} L(f(x), y). \tag{4}$$

Since D_{train} is deterministic, a hypothesis space $\mathcal{F}$ of hypotheses $f(\boldsymbol{\theta})$ is chosen to optimize the model. The minimization of $\mathcal{R}_I(f)$ can be denoted as:

$$f_e = \arg\min_{f \in \mathcal{F}} \mathcal{R}_I(f). \tag{5}$$

Ideally, f_e approximates f_o as closely as possible. However, since f_o is unknown, it requires some $f \in \mathcal{F}$ to approximate

it. Assume f_b is the best approximation for f_o in $\mathcal{F}$, which can be formulated as:

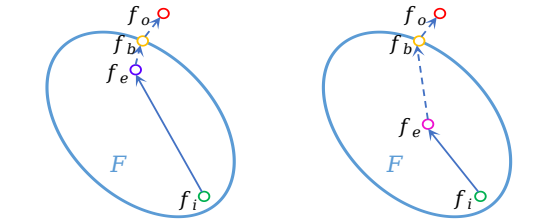
$$f_b = \arg\min_{f \in \mathcal{F}} \mathcal{R}(f). \tag{6}$$

Eclectically, we hope f_e can approximate f_b as closely as possible. For simplicity, we assume that f_o, f_e, and f_b are well-defined and unique. The total error can be decomposed as:

$$\mathbb{E}[\mathcal{R}(f_e) - \mathcal{R}(f_o)] = \underbrace{\mathbb{E}[\mathcal{R}(f_b) - \mathcal{R}(f_o)]}_{\mathcal{E}_{app}} + \underbrace{\mathbb{E}[\mathcal{R}(f_e) - \mathcal{R}(f_b)]}_{\mathcal{E}_{est}}. \tag{7}$$

Here, the expectation concerns the random choice of D_{train}. The approximation error $\mathcal{E}_{app}$ measures how closely functions in $\mathcal{F}$ can approximate the optimal hypothesis f_o, and the estimation error $\mathcal{E}_{est}$ measures the effect of minimizing the empirical risk $\mathcal{R}_I(f)$ instead of the expected risk $\mathcal{R}(f)$ in $\mathcal{F}$. Overall, the hypothesis space $\mathcal{F}$ and the number of examples in D_{train} affect the total error [27].

As illustrated in Fig. 4(a), when the supervised information in D_{train} is sufficient, i.e., I in D_{train} is large enough, the empirical risk minimization function in $\mathcal{F}$ can approximate the best-expected risk minimization function in $\mathcal{F}$ well, i.e., f_e can provide a good approximation to f_b. However, due to the limited number of training samples in each FSCIL incremental session, the best empirical risk minimization function is often a poor approximation to the best-expected risk minimization function in $\mathcal{F}$, i.e., f_e is far from f_b in $\mathcal{F}$, as shown in Fig. 4(b). This discrepancy leads to unreliable empirical risk minimization in the model learning process.



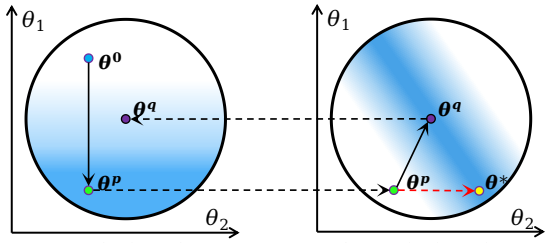
(a) Learning with sufficient samples (b) Learning with few samples

Fig. 4. The illustration of unreliable empirical risk minimization in FSCIL. (a) with sufficient training samples, the empirical risk minimization can approximate the best-expected risk minimization function. (b) when the training samples are insufficient, the best empirical risk minimization function is often a poor approximation to the best-expected risk minimization function.

2.2.2 Stability-plasticity Dilemma

In FSCIL, a central challenge is the stability-plasticity dilemma, which involves balancing the model's consistent performance on learned classes (stability) and its adaptability to new classes with limited samples (plasticity). Traditional deep learning models are typically static and can only handle previously learned classes. FSCIL demands continual learning of new classes with only a few available labeled training samples and without access to the original complete training data of old classes. It requires the model to maintain the stability of previously learned knowledge and plasticity in learning new knowledge. Due to different optimization goals for old and new classes, the decision boundary often shifts toward new classes, leading to catastrophic forgetting. Conversely, focusing too much on old knowledge stability may limit the ability to learn new tasks, a phenomenon known as intransigence. Therefore, balancing stability and plasticity is crucial in FSCIL.

The stability-plasticity dilemma can be illustrated through consecutive sessions p and q. Fig. 5(a) and Fig. 5(b) depict error surfaces for these sessions, with darker areas representing ideal loss values, and the model under consideration has only two parameters, θ_1 and θ_2. It can be observed that the optimization objective of session p is to move downwards, while that of session q is to approach the band center. Suppose the initial model on session p is $\boldsymbol{\theta^0}$, and the optimized is $\boldsymbol{\theta^p}$, which shows promising performance on session p. However, when the model starts learning the next session q, $\boldsymbol{\theta^p}$ obtained from session p is insufficient to meet the requirement of session q. To solve the problem, the model usually adjusts the parameters to minimize the loss towards the center of the loss surface. Assuming the optimized model for session q is $\boldsymbol{\theta^q}$, it can be observed that $\boldsymbol{\theta^q}$ can adapt well to the analysis tasks on session q. However, when we use $\boldsymbol{\theta^q}$ to make predictions on session p, the decision boundary cannot achieve satisfactory performance, indicating the occurrence of forgetting. Nevertheless, if we constrain $\boldsymbol{\theta^p}$ to move towards $\boldsymbol{\theta^\star}$ while learning session q, we can observe that the model can adapt to both session p and session q effectively.



(a) Learning in session p (b) Learning in session q

Fig. 5. The illustration of stability-plasticity dilemma in FSCIL. (a) and (b) are two consecutive sessions. Darker areas indicate optimal loss values. $\boldsymbol{\theta^p}$ performs well in session p but poorly in q. Optimizing $\boldsymbol{\theta^p}$ to $\boldsymbol{\theta^q}$ on session q diminishes its performance on session p. Yet, directing optimization towards $\boldsymbol{\theta^\star}$ ensures good results on both sessions.

To balance model stability and plasticity in a new session, the key approach is distinguishing between critical and non-critical parameters from the previous session, optimizing only the non-critical ones. The loss function for the new session encompasses both the classification task and prevention of catastrophic forgetting. It is formulated as follows:

$$L'(\boldsymbol{\theta}) = L(\boldsymbol{\theta}) + \lambda \sum_i b_i \left(\theta_i - \theta_i^b\right)^2, \tag{8}$$

where L is the partial loss function for the current classification task, θ_i denotes the parameter in the current model $\boldsymbol{\theta}$, θ_i^b represents the corresponding parameter in the previous model $\boldsymbol{\theta^b}$, b_i characterizes the importance of θ_i^b for the previous task, and the hyperparameter λ balances the two parts of the overall loss. Setting $b_i = 0$ imposes no constraint on θ_i, leading to catastrophic forgetting. Conversely, setting $b_i = \infty$ results in intransigence, where θ_i always equals θ_i^b.

2.3 General Schemes

FSCIL has two main frameworks, as shown in Fig. 6. The first uses a feature extractor with a softmax classifier, while the second involves a feature embedding network and the nearest class mean classifier [22, 42, 43]. The entire network is trainable throughout the IL process in the first one. To mitigate catastrophic forgetting, some studies [44, 45, 46] use KD to maintain competent classification capabilities on previous classes while learning new ones. The second framework focuses on training a feature embedding network to map samples into a space where distances represent semantic differences, followed by classification using the nearest class mean classifier. For instance, some

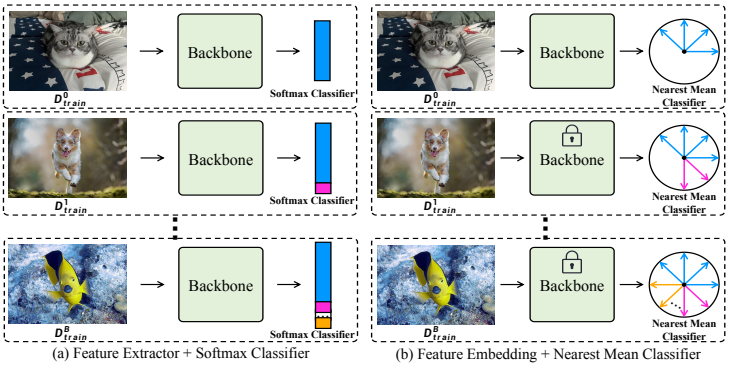


Fig. 6. The general schemes of FSCIL. (a) "Feature Extractor + Softmax Classifier" has a trainable backbone with Knowledge Distillation (KD)-based methods being the most typical method. (b) is "Feature Embedding + Nearest Mean Classifier" with a fixed backbone after base training.

studies [47] employ metric loss for the training of the embedding network, enabling it to learn more discriminative features and better adapt to incremental classes.

2.4 Relevant Problems

2.4.1 Incremental Learning

This section reviews the relationship and distinctions between FSCIL and other IL scenarios, specifically CIL, TIL, and DIL, as outlined by *Van de Ven et al.* [11].

Class-incremental Learning: CIL aims to learn an algorithm that can continuously recognize new classes without forgetting old ones [5, 11, 34]. As FSCIL can be seen as a subdomain of CIL, it can be observed from Fig. 3(a) and Fig. 3(b) that their general settings are very similar. Both require learning new class data as it arrives and maintaining classification abilities on previous classes. However, FSCIL's base session often includes many training samples, while CIL does not have strict restrictions. Additionally, the training samples in the incremental session of FSCIL are limited and exist in the form of $N-$way $K-$shot. In contrast, the training samples in the incremental session of CIL are usually sufficient. The core challenge of CIL lies in solving the stability-plasticity dilemma. At the same time, FSCIL needs to solve this challenge and address the problem caused by unreliable empirical risk minimization due to the lack of training samples and its sustained impact in continuous scenarios.

Task-incremental Learning: TIL aims to learn an algorithm that can progressively learn new tasks without forgetting old ones. As depicted in Fig.3(c), TIL's training data in classification scenarios is split into multiple sessions, each representing a distinct task. During both training and testing, the TIL model is always aware of the specific task identity. To avert catastrophic forgetting, various algorithms[11, 48, 49] employ task-specific components or design separate networks for each task. TIL's primary challenge lies in identifying shared features across tasks to balance performance and computational complexity, using knowledge from one task to enhance performance in others [11].

Domain-incremental Learning: DIL is an ML problem designed to continuously adapt to data distribution from different domains while the structure of the problem is always the same [11]. DIL addresses the variation in data distribution across incremental domains, enabling effective learning and prediction in new domains without forgetting previously acquired knowledge. As depicted in Figure 3(d), DIL involves training data from multiple sessions, each containing identical classes but with distinct data distributions indicative of different domains. The DIL model must continuously adapt to these new domains without losing prior knowledge. Its primary challenge is to identify and leverage shared features across domains, allowing quick adaptation to new domains and learning new knowledge while preserving existing knowledge in old domains.

2.4.2 Few-shot Learning

FSL refers to using very few training samples for model learning [50]. To better understand the correlations and distinctions between FSCIL and FSL, this section presents pertinent concepts, including FSL and general Few-shot Learning (gFSL). For clarity, Tab. 2 is provided, summarizing the distinct attributes of FSL, gFSL, and FSCIL.

TABLE 2
The difference between FSL, gFSL, and FSCIL. Note that the base classes indicate the original complete version of base training data.

Settings	Training Data		Testing Data
	Initial Phrase	Sequent Phrase	
FSL	Base Classes	New Classes	New Classes
gFSL	Base Classes	Base + New Classes	Base + New Classes
FSCIL	Base Classes	New Classes	Base + New Classes

Few-shot Learning: FSL is an ML problem that aims to learn a model capable of classifying and recognizing new classes with very limited training samples [27, 51, 52]. Similar to FSL, FSCIL also employs $N-$way $K-$shot learning for each new class. However, FSCIL's training data comprises multiple incremental sessions, each with several few-shot classes. As Tab. 2 indicates, FSL's main goal is to enable model generalization to new classes using limited training data, without emphasizing base class recognition performance. In contrast, FSCIL aims to continuously learn new classes with limited samples while preserving knowledge of previously learned classes.

General Few-shot Learning: FSL typically doesn't consider base class performance in testing [53]. However, real-world applications often require models to learn new classes from limited samples while maintaining performance on base classes, which often represent high-frequency classes in the real world [13, 54]. This practical need has led to the development of a novel setting, gFSL [15], aimed at enabling learning of new classes with limited samples without compromising performance on previous classes [15, 55, 56]. As highlighted in Tab. 2, unlike FSCIL, gFSL allows access to initial training data of base classes.

2.5 Taxonomy

For a thorough examination of FSCIL research, we propose a taxonomy for current methods. Illustrated in Fig. 7, we analyze existing methods from three angles: data-based, structure-based, and optimization-based approaches for the FSCIC problem. Additionally, for the FSCIOD issue, we assess methods through anchor-based and anchor-free perspectives.

3 Datasets and Evaluation

3.1 Datasets

3.1.1 Datasets for Classification

***mini*ImageNet:** *mini*ImageNet, a diverse and challenging dataset containing object classes from various fields such as animals, plants, daily necessities, and vehicles, was originally proposed by *Vinyals et al.* [57] in 2016 and has been commonly used to evaluate FSL algorithms. The dataset comprises 60,000 images selected from ImageNet [58], with 100 classes and 600 images per class, each sized at 84×84 pixels. In FSCIL, the prevalent data

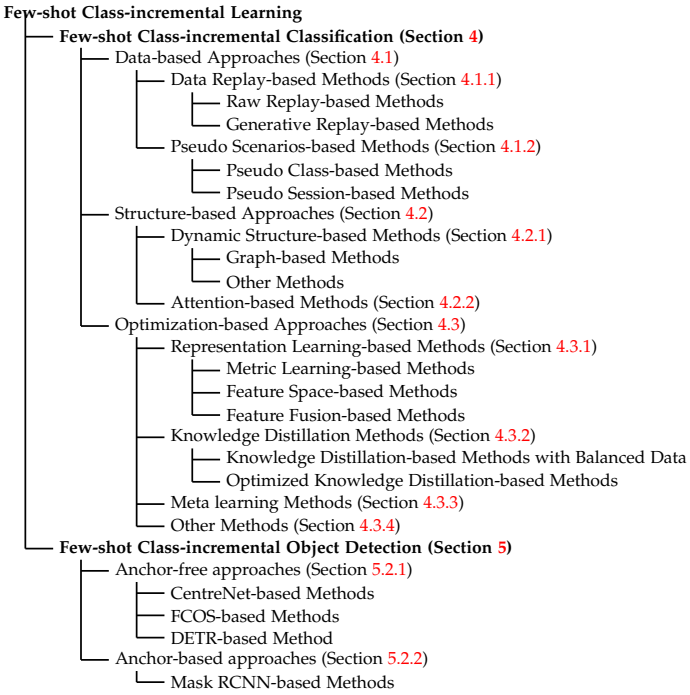


Fig. 7. The taxonomy of representative methods in FSCIL.

partitioning method by *Tao et al.* [13] divides these 100 classes into 60 base and 40 incremental classes. These incremental classes are further segmented into 8 sessions, each with 5 classes and 5 training samples per class, forming a $5-$way $5-$shot setup.

CIFAR-100: CIFAR-100, introduced by *Krizhevsky et al.* [59] in 2009, is widely used in CIL. It features a broad array of image data, covering classes such as plants, humans, and vehicles. The dataset comprises 100 classes, each with 600 32×32 RGB images, allocated into 500 for training and 100 for testing. For FSCIL, the common data partitioning approach by *Tao et al.* [13] divides these 100 classes into 60 base classes and 40 incremental classes. These incremental classes are further split into 8 sessions, each with 5 classes. Every class in these sessions has 5 training samples, establishing a $5-$way $5-$shot format.

CUB-200: The Caltech-UCSD Birds-200-2011 (CUB-200) dataset, created by *Wah et al.* [60] in 2011, is a benchmark dataset for fine-grained classification in computer vision. It comprises 11,788 images across 200 bird species. For FSCIL algorithm evaluation, the data partitioning method by *Tao et al.* [13] is commonly employed. This method splits the 200 classes into 100 base and 100 incremental classes, with these incremental classes further divided into 10 sessions. Each session encompasses 10 classes, with 10 training samples per class, resulting in each session being a $10-$way $10-$shot task. The standard image size in this context is 224×224 pixels.

3.1.2 Datasets for Object Detection

COCO: The Microsoft Common Objects in Context (COCO) dataset, widely used for object detection tasks, comprises 80 object classes including people, animals, vehicles, furniture, and food [61]. It features a diverse and complex array of images that reflect real-world scenarios, complete with detailed annotations such as bounding boxes, class labels, and semantic segmentation masks. For FSCIOD tasks, the data partitioning strategy by *Perez-Rua et al.* [62] is commonly used. This approach utilizes 20 classes overlapping with the PASCAL VOC dataset [63] as new incremental classes and the remaining 60 as base data. FSCIOD models under this setup are evaluated using $K \in 1, 5, 10$ bounding boxes per new class.

PASCAL VOC: The PASCAL Visual Object Classes (VOC) dataset, widely used for object detection tasks, includes 20 common object classes like people, animals, vehicles, and household items [63]. It is frequently utilized for cross-dataset evaluations of FSCIOD algorithms. Notably, the VOC shares 20 classes with the COCO dataset. Thus, the 60 non-overlapping classes in COCO are typically the base training data for cross-dataset evaluations, with the VOC's 20 classes serving as new incremental classes to assess few-shot IL capabilities. The evaluation strategy, proposed by *Perez-Rua et al.* [62], is similar to that used with the COCO dataset, where FSCIOD models are evaluated using $K \in 1, 5, 10$ bounding boxes annotated for each new class.

3.2 Evaluation Metrics

3.2.1 Evaluation Metrics for Classification

In FSCIL, the model needs to learn new classes while retaining previous knowledge. After each session, it is tested on all encountered classes, using accuracy as the primary metric. Additionally, after completing all incremental sessions, the overall performance is evaluated using Average Accuracy (AA) and Performance Dropping (PD) rate. The AA calculates the mean accuracy across the base and all incremental sessions, with higher values indicating superior performance. PD measures the accuracy drop between the base and final incremental sessions, where lower values represent better FSCIL performance. Definitions are shown in Tab. 3.

TABLE 3
Evaluation metric definitions for classification and object detection tasks. A_i represents the accuracy obtained in session i, and B is the number of incremental sessions in classification. P_i and R_i represent the precision and recall obtained in class i, and K is the number of counted classes.

Task	Metric	
Classification	$AA = \frac{1}{B+1}\sum_{i=0}^{B} A_i$	$PD = A_B - A_0$
Object Detection	$mAP = \frac{1}{K}\sum_{i=1}^{K} P_i$	$mAR = \frac{1}{K}\sum_{i=1}^{K} R_i$

3.2.2 Evaluation Metrics for Object Detection

In FSCIOD tasks, two approaches incorporate new incremental data: batch and continuous IL. Batch IL entails learning all new classes at once, while continuous IL adds new classes progressively. Batch IL, similar to single-session FSCIL, is more common. The predominant performance metric is mean Average Precision (mAP), which is the mean of the AP values calculated for all counted classes. mAP is calculated separately for base classes, new classes, and all classes, with higher mAP values across all classes indicating better FSCIOD performance. Additionally, some studies use a similar way to calculate mean Average Recall (mAR) and mAP50 as complementary metrics for a more comprehensive evaluation. Definitions are shown in Tab. 3.

3.3 Summary

The overview of datasets and evaluation methods reveals a scarcity of publicly available datasets for FSCIL tasks, limiting their practical application. Some studies, such as [64], have introduced datasets for various FSCIL scenarios, but there remains significant scope for dataset enhancement. Regarding model evaluation, while current metrics assess the model's learning

ability to some extent, they don't completely capture the detailed performance of FSCIL throughout the continuous learning process [47]. Hence, both datasets and evaluation metrics in FSCIL present substantial opportunities for further development.

4 Few-shot Class-incremental Classification

This section, focusing on FSCIL classification tasks, summarizes existing methods classified into data-based, structure-based, and optimization-based categories, noting some overlap across these domains. The methods are categorized based on their attributes and core innovations, concluding with a performance comparison and key concerns discussion.

4.1 Data-based Approaches

Data-based approaches refer to addressing FSCIL challenges arising from limited or non-reusable data by focusing on the data perspective. Relevant methodologies include DR and pseudo-data construction.

4.1.1 Data Replay-based Methods

Catastrophic forgetting often occurs in FSCIL due to the unavailability of original complete training data from previous sessions. DR is a direct strategy to mitigate this issue by replaying valuable data while adapting to new sessions. Existing methods include raw replay and generative replay, involving the replay of samples or feature representations.

Raw Replay-based Methods: The raw replay methods address catastrophic forgetting by storing a portion of raw samples from previous sessions in auxiliary memory and replaying it during the learning process of a new session to review previous knowledge. As shown in Fig. 8(a), *Kukleva et al.* [18] proposed a multi-stage FSCIL method called LCwoF. It first used the Cross-entropy (CE) loss to train the backbone. In the second stage, it employed the KD loss and base-normalized CE loss to jointly supervise learning new classes and preserve the old knowledge. In the final stage, randomly sampled old and new class data were combined for DR to further calibrate the performance. Differently, *Zhu et al.* [44] proposed a feature distribution distillation-based method, which stored the same number of old samples as each new class to form a joint set during its learning process. Both the old and new models generated the feature representations for this set. A joint function based on CE loss and KD loss was employed to constrain the new model to generate similar representations as the old model to preserve the old knowledge. However, the performance of raw replay methods is influenced by factors such as auxiliary storage space, sample selection, and quantity, which have not been fully addressed.

Generative Replay-based Methods: The generative replay methods train and store a model that generates data, including samples or feature representations of old classes during the new session learning process to review old knowledge. As shown in Fig. 8(b), *Liu et al.* [19] proposed a data-free replay method that used GAN-like ideas to train a generator with an uncertainty constraint based on entropy regularization so that the generated data could get close to the decision boundary. In incremental sessions, the generated and new data fine-tuned the model, giving it a good performance on new and old classes. Different from generating samples, some methods choose to generate features. Specifically, *Shankarampeta and Yamauchi* [65] proposed a framework based on Wasserstein GAN [66] with MAML [67], which mainly consisted of a feature extractor and a feature generator. During the training process, the feature extractor was

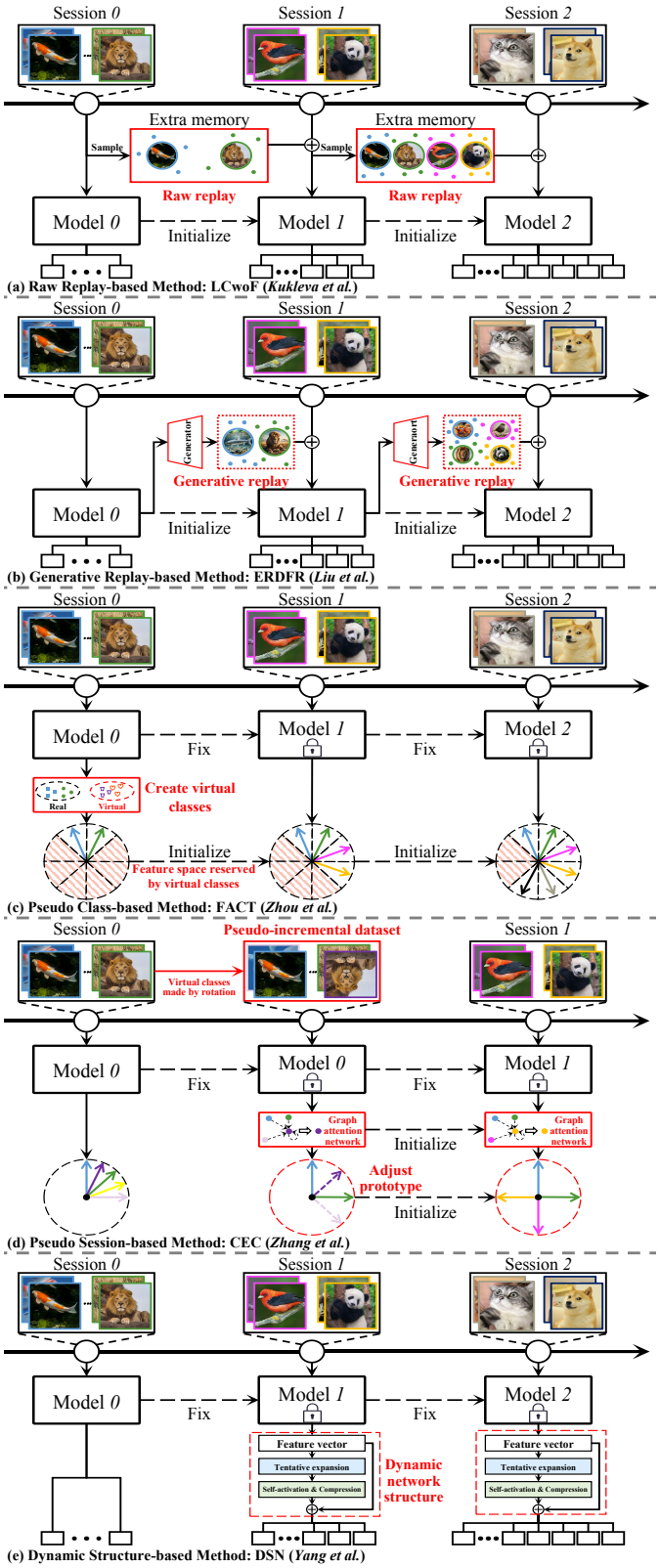


Fig. 8. The representative classification methods in FSCIL, with the core designs highlighted in red. (a) LCwoF stores some old samples for raw replay, jointly calibrating performance with the new classes during new sessions; (b) ERDFR trains a generator using the previous main model and synthesizes virtual old samples for DR during the learning of new classes, aiding in the retention of old knowledge; (c) FACT creates virtual incremental classes from base classes to simulate future scenarios, enabling the model to develop forward compatibility during the base session; (d) CEC constructs a pseudo-incremental session from the base session to train a GAT, which is later used to adjust the relationship between new and old class prototypes during real incremental sessions; (e) DSN designs a dynamic structure alongside the backbone that allows for expansion and autonomous compression, facilitating the learning of new classes while retaining old knowledge.

initialized on base data, and then the feature generator was trained by meta-learning with MAML. In IL, the feature extractor with feature distillation was combined with feature replay at the classifier level to tackle catastrophic forgetting. Similarly, FSIL-GAN proposed by *Agarwal et al.* [68] used a similar framework to perform feature replay. The main contribution of FSIL-GAN was the import of the semantic projection module, which constrained the synthesized features to match with the latent semantic vectors to ensure their diversity and discriminability. In IL, KD ensured knowledge transfer between the old and new generators. Generative replay offers flexibility and safety in sample generation but increases the model complexity.

Discussion: DR is a direct strategy to address catastrophic forgetting in FSCIL. While raw replay methods provide simplicity and convenience, their effectiveness is influenced by factors including auxiliary storage space, the selection and quantity of samples, and the imbalanced distribution of old and new classes. In comparison, generative replay methods exhibit better flexibility and mitigate potential privacy concerns associated with raw replay methods. However, generative replay methods face challenges in continuously generating old samples in the imbalanced and dynamic data stream, generation quality and efficiency, and additional computational costs. These issues require further exploration and research.

4.1.2 Pseudo Scenarios-based Methods

Contrasting backward-compatible methods such as DR that tackle catastrophic forgetting, another prevalent FSCIL strategy is the construction of pseudo-incremental scenarios. These scenarios, acting as preview mechanisms in the dynamic and ever-expanding FSCIL data stream, prepare models for actual incremental sessions, ensuring effective performance. These methods primarily fall into two categories: pseudo-class and pseudo-session construction.

Pseudo Class-based Methods: Pseudo-class construction methods aim to generate synthetic classes and their corresponding samples to facilitate FSCIL models preparing for the real incremental classes. Most current studies employ base sessions to develop these pseudo-classes, training the models using pseudo-data and the original data. This strategy promotes forward compatibility in the FSCIL models. As shown in Fig. 8(c), this approach is the forward-compatible FSCIL framework proposed by *Zhou et al.* [23]. The crux of this framework lay in constraining the real samples during training, enabling them to render their respective categories more compact in the embedding space and reserve some spaces for the constructed virtual categories. In particular, this method promoted the intra-class compactness of the real data and forced the masked features based on the real data to be closed to a pseudo-class. Simultaneously, the framework employed similar techniques to constrain virtual features constructed from a mixture of multiple class features. It ensured the compactness of real categories while reserving some feature space for incremental classes. Similarly, *Peng et al.* [47] generated pseudo-classes by merging two distinct classes from the base session and augmenting the data using techniques such as random cropping, horizontal flipping, and color jittering in the ALICE framework. It used angular penalty loss commonly used in face recognition for feature extractor training based on the joint set of pseudo and real data. The core idea also involved promoting intra-class compactness and reserving feature space for incremental classes.

Pseudo Session-based Methods: Unlike pseudo-class methods that create synthetic classes, pseudo-session construction methods focus more on emulating incremental sessions. Most existing approaches use base sessions to create pseudo-incremental sessions and meta-learning techniques to allow FSCIL models to understand how to handle incremental sessions. The ways to construct pseudo-incremental sessions are various. As shown in Fig. 8(d), the CEC framework proposed by *Zhang et al.* [14] applied the large angle rotation transformation on the base classes to build pseudo-incremental sessions. These pseudo-sessions were then combined with the base session to train the graph attention network by meta-learning strategy so that it could pass context information between prototypes, thus better equipping it to handle the FSCIL task. The FSCIL model by *Zhu et al.* [69] included two innovations: Random Episode Selection (RES) and Dynamic Relation Projection (DRP). RES sampled five classes randomly to create $N-$way $K-$shot pseudo-incremental sessions, masking original class prototypes and using pseudo-incremental data to generate class prototypes by averaging. These prototypes were refined using DRP, which mapped class prototypes from standard and pseudo-IL to shared latent space. It calculated the cosine similarity between old and new classes to obtain a relation matrix. This matrix acted as a transitional coefficient for prototype optimization, enabling dynamic optimization to preserve existing knowledge and boost new classes' discriminative ability.

Discussion: Pseudo-scenario construction is a forward-compatible strategy, synthesizing classes or sessions to train models for future real incremental classes. Pseudo-class construction is a method where pseudo-classes are constructed in conjunction with base classes to train the model, enabling the feature space to reserve certain spaces for upcoming incremental classes. However, reserving space often requires prior knowledge of the total number of incremental classes, which contradicts the real world. Since synthetic and real data often exhibit differences, the suitability of reserved space remains to be discovered. In contrast, pseudo-session construction is more reasonable, as it often combines the pseudo-incremental sessions with meta-learning to train the model that can learn to adapt to incremental sessions. However, the issue of whether pseudo-incremental sessions can effectively simulate real incremental sessions needs further exploration.

4.2 Structure-based Approaches

Structure-based approaches refer to utilizing the model design or its characteristics to address the challenges in FSCIL. These methods mainly involve dynamic structure methods and attention-based methods.

4.2.1 Dynamic Structure-based Methods

Dynamic structure methods aim to achieve FSCIL by dynamically adjusting the model structure or the interrelationships between prototypes. Currently, existing methods can be broadly categorized into graph-based and other methods.

Graph-based Methods: Methods based on graph structures utilize graph topological properties to achieve FSCIL. These methods typically use nodes and edges in the graph to describe the similarity or correlation between different classes from various sessions and adjust the graph structure based on the mutual influences among classes. Some studies employ graph structures to implement FSCIL [13, 14]. For example, *Tao et al.* [13] proposed the TOPIC framework, which utilized the neural gas network for knowledge extraction and representation. TOPIC aimed to address FSCIL by dynamically adjusting the interrelationships

between feature representations. Specifically, the neural gas network defined an undirected graph that mapped the feature space to a finite set of feature vectors and maintained the topological properties of the feature space through competitive Hebbian learning [70]. To achieve FSCIL, they gradually improved the neural gas network by enabling the supervised neural gas model to grow nodes and edges through competitive learning. Additionally, they designed a stability loss to suppress catastrophic forgetting and an adaptability loss to reduce overfitting. In addition, the CEC framework [14] mentioned in Sec. 4.1.2 also utilized graph structures for FSCIL. It first trained a graph attention network using pseudo-incremental sessions to adjust the model. During incremental sessions, the model incorporated an attention mechanism to regulate the interrelationships between nodes, represented by prototypes of old and new classes. This allowed for better context information transfer between sessions, making the class prototypes more robust.

Other Methods: In addition to graph-based methods, some studies employ other dynamic structures to achieve FSCIL [25, 71, 72]. For example, *Yang et al.* proposed a series of works [25, 71]. As shown in Fig. 8(e), they proposed a novel Dynamic Support Network (DSN) [25] to address the challenges of FSCIL. DSN was a self-adaptive updating network with compressed node expansion, aiming to "support" the feature space. In each session, DSN temporarily expanded the network nodes to enhance the feature representation capability for incremental classes. Then, it dynamically compressed the expanded network through node self-activation, pursuing compact feature representation to alleviate overfitting. Moreover, DSN selectively invoked the distribution of old classes during the IL process to support feature distribution and avoid confusion between classes. Furthermore, in the framework proposed by *Ahmad et al.* [72], the output nodes of the model increased with the number of classes involved in the current session. The model parameters related to old classes were kept fixed. The newly added nodes' weights were randomly initialized, and they were trained using the training data from the current session to update the parameters.

Discussion: Dynamic structure methods are important approaches to address the challenges of FSCIL. These methods achieve the learning of new knowledge while preserving old knowledge by dynamically adjusting the model structure or the relationships between prototypes. For example, graph-based methods utilize the topological characteristics of graphs to achieve non-forgettable IL by adjusting nodes and edges to describe the similarity and correlation between different classes. Dynamic structure networks enhances feature representation and alleviates overfitting by temporarily expanding and dynamically compressing network nodes. Dynamic structural methods play a significant role in FSCIL, but further research and exploration are still needed to develop more design methods for dynamic structures.

4.2.2 Attention-based Methods

Attention-based methods in FSCIL adjust the attention allocation of features by introducing attention modules into the model structure. This allows the model to focus on information relevant to the current task, improving its performance and generalization ability. The role of attention modules used in many FSCIL approaches [14, 16, 17, 73, 74] is diverse. For example, in the dual-branch KD framework proposed by *Zhao et al.* [73], which consisted of a base branch and a novel branch, they noted that fine-tuning by novel classes inevitably led to forgetting old knowledge. To further improve the performance of base classes, they proposed an attention-based aggregation module that selectively merges predictions from the base branch and the novel branch. Furthermore, *Cheraghian et al.* [17] employed meta-learning to train a backbone that can incrementally learn new classes with limited samples without forgetting the old classes. However, many existing FSCIL paradigms updated the classifier by concatenating the base classifier with the new class prototypes obtained by averaging the features of each training sample. This approach often led to bias. Therefore, this paper proposed a correction model based on Transformer [75]. With its attention mechanism, the correction model can effectively transmit context information among different classes, making the classifier more efficient and robust. Similar approaches included the graph attention network used in the CEC framework [14].

4.3 Optimization-based Approaches

Optimization-based approaches tackle the challenges in FSCIL by addressing the complexity of optimization problems. The relevant strategies primarily involve representation learning, meta-learning, and KD, according to the existing works.

4.3.1 Representation Learning-based Methods

In FSCIL, representation learning aims to extract meaningful features from a limited stream of samples to form a "representation" of the data [76]. Through effective representation learning, models can identify and utilize underlying patterns within these few samples and generalize them to new, unseen classes. Even in few-shot incremental scenarios, models can perform excellently with efficient representation learning. In FSCIL, there are diverse approaches to performing representation learning, which can be categorized into metric learning-based, feature space-based, feature fusion-based, and other methods, based on the core principles of the respective methods.

Metric Learning-based Methods: Metric learning aims to determine the similarity between objects using an optimal distance metric for learning tasks [77]. It has found extensive application in FSL [78]. Recently, metric learning has also been adopted in FSCIL to learn effective representations. Among the commonly used approaches, triplet loss stands out. As shown in Fig. 9(a), *Mazumder et al.* [21] proposed a novel approach for FSCIL. It incorporated self-supervised learning to enhance the generalization capability of the backbone. Then, this approach analyzed the importance of model parameters and updated only the unimportant parameters for new classes. The update was achieved by combining three loss functions: triplet loss, regularization loss, and cosine similarity loss. The triplet loss aimed to generate discriminative features, while the regularization loss prevented catastrophic forgetting. The cosine similarity loss focused on controlling the similarity between old and new prototypes. Thus, FSCIL achieved effective performance. Moreover, other metric learning methods are applied to FSCIL too. Concretely, *Peng et al.* [47] proposed the ALICE framework, incorporating the angular penalty loss originally used in face recognition to obtain well-clustered features. This loss was employed to train the backbone using both base class data and synthetic data, thereby creating additional space for accommodating incremental classes, and cosine similarity was applied to achieve the classification.

Despite the good performance achieved by these margin-based metric losses, *Zou et al.* [26] pointed out the issue in FSCIL: large margin values can result in good discriminability among base classes but hinder the generalization capability of new classes. Conversely, small or even negative margin values can

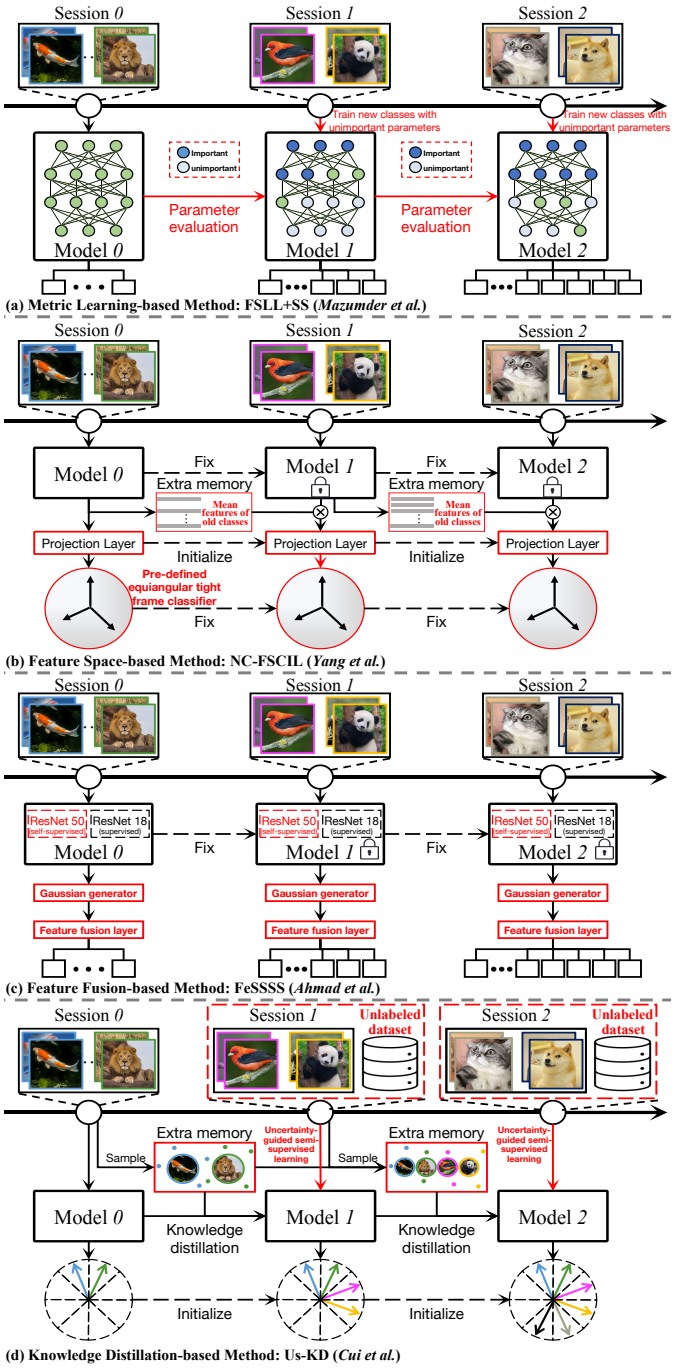


Fig. 9. The representative classification methods in FSCIL, with the core designs highlighted in red. (a) FSLL+SS initially uses self-supervised learning to provide good generalization. Then, it evaluates the importance of model parameters, fixing important ones to maintain old knowledge while using metric learning to learn new classes with unimportant parameters; (b) NC-FSCIL uses neural collapse theory to pre-define a classifier before training, and the model is constrained to learn towards the pre-defined classifier via the projection layer, preventing conflicts between learning new and old classes; (c) FeSSSS introduces a ResNet50 trained through self-supervised learning in addition to the regular ResNet backbone. By merging the supervised and self-supervised features, the framework enhances the generalization capability in FSCIL; (d) Us-KD focuses on semi-supervised FSCIL, first utilizing KD with stored old samples and new data to learn new knowledge while retaining old knowledge. Then, uncertainty quantification is applied to select suitable unlabeled data for labeling, combined with labeled samples for iterative model updates.

lead to poor performance on the base classes but exhibit better generalization on new classes. This phenomenon is known as the class-level overfitting problem. To address this issue, *Zou et al.* [26] proposed the CLOM framework, which combined margin theory with the characteristics of neural network structures. Specifically, since the shallow layers of neural networks are more suitable for learning common features among classes, while the deep layers are better suited to acquiring advanced features, they designed a loss function that constrains shallow feature learning and deep feature learning separately. Furthermore, this framework alleviated the class-level overfitting problem by integrating class relationships.

Feature Space-based Methods: Feature space-based methods are a class of approaches that aim to perform FSCIL by optimizing the feature space. The core idea of these methods is to design the feature space for learning more robust and efficient feature representations. Some related methods address FSCIL by designing subspaces [22, 79, 80]. For example, inspired by the frequency decoupling [81], *Zhao et al.* [22] discussed and utilized the characteristics of different frequency components in features. Specifically, the method first trained a feature extractor using metric learning loss and regularization loss. Then, they decoupled the features based on their frequency and observed the roles of high-frequency and low-frequency information in FSCIL. It was found that low-frequency information may contribute more to preserving old knowledge. Therefore, they designed subspaces with different learning rates to learn features in different frequency domains, where the fast subspace learned new knowledge and the slow subspace preserved old knowledge. Through this subspace combination strategy, the method achieved good performance.

Furthermore, some methods address FSCIL by designing feature spaces of specialized structures. For example, *Hersche et al.* [24] proposed a C-FSCIL framework, which consisted of a feature extractor trained by meta-learning, a trainable fully connected layer, and a rewritable explicit memory. The core idea was introducing hyperdimensional embedding, which has three advantages: (1) the high probability of quasi-orthogonality between random vectors, (2) rich expressive space, and (3) good semantic representation capability. C-FSCIL had three training strategies. The first was based on simple meta-learning, as described in Sec. 4.3.3. The second strategy stored initial prototypes in the globally average activation memory and applied an element-wise sign operation to transform similar feature prototypes into bipolar vectors. These transformed vectors were then supervised to train the fully connected layer, which learned the weights for the final prototypes. The third strategy was similar to the second one but incorporated two losses to constrain the inter-class differences and maintain the relationship with the original prototypes. Besides, as shown in Fig. 9(b), *Yang et al.* [82] proposed an FSCIL framework based on neural collapse [83], which refers to the phenomenon that at the end of training (after 0 training error rate), the last layer features of the same class collapse into a single vertex in the feature space, aligning all class vertices with their classifier prototypes and forming as a simplex equiangular tight frame. Based on this characteristic, the proposed framework predefined a structure similar to neural collapse and directed the model to optimize it. Specifically, a group of prototypes for both the base and incremental sessions was pre-assigned as a simplified form of equiangular tight frame. During training, the prototypes were fixed. They introduced a novel loss function and an additional projection layer to assign each class to its respective prototype separately. Without cumbersome operations, this method achieved superior performance. In addition, in Sec. 4.1.2, the method proposed by *Zhou et al.* [23] also addressed FSCIL by learning the feature space in a way that preserves some space for incremental classes during the learning

of base classes.

Feature Fusion-based Methods: Feature fusion refers to integrating or combining features obtained from different information sources or feature extraction methods to create a more comprehensive and efficient representation that exhibits robustness and generalization capabilities [84]. In the context of FSCIL, various methods employ feature fusion strategies to learn effective feature representations that can adapt to specific task requirements. Notably, a significant focus is on incorporating self-supervised features into the fusion process [21, 72, 85, 86]. For example, as shown in Fig. 9(c), *Ahmad et al.* [72] proposed a framework that combines self-supervised and supervised features. The core structure of this framework included the following components: Firstly, feature extractors obtained through supervised training on base-class data and self-supervised tasks on ImageNet [58] or OpenImages-v6 [87] using methods such as pretext tasks, contrastive loss, or clustering. Secondly, the Gaussian generator synthesized feature for replay in incremental sessions. Lastly, a lightweight model for incremental feature fusion and classification. Additionally, *Kalla and Biswas* [86] proposed S3C, a method for addressing FSCIL based on the stochastic classifier [88] and self-supervision. They introduced a novel self-supervised training approach [89], using image augmentations to generate artificial labels, to train the classification layer. The stochastic classifier weights helped mitigate the impact of limited new samples and the unavailability of old samples. The self-supervision component enabled the learning of features from base classes that generalize well to future unseen classes, effectively reducing catastrophic forgetting.

In addition to feature fusion based on self-supervised features, there are also works that integrate other features to achieve good performance in FSCIL. For example, *Yao et al.* [90] proposed a simple strategy for enhancing the new prototype. Specifically, it first trained a Convolutional Neural Network (CNN) on the base classes and kept it fixed. It was used to generate class prototypes for both base and new classes. Then, the initial class prototypes for new classes were measured for their similarity to base class prototypes. Based on the similarity, the prototypes for new classes were updated by mixing the initial weights with other base prototypes. This fusion enhancement strategy imitated human cognition by guiding new class learning based on existing knowledge.

Discussion: Feature fusion plays a crucial role in FSCIL by integrating multiple information sources and feature extraction methods to provide a comprehensive, efficient, and robust representation. In FSCIL, various feature fusion strategies are employed to learn effective feature representations that adapt to specific tasks. For instance, combining self-supervised and supervised features enables the model to acquire representations with good generalization ability. Additionally, another approach fuses existing features to guide new class learning. These methods highlight the significance of feature fusion in addressing FSCIL challenges, while further exploration of more efficient feature fusion strategies is needed to enhance model performance and generalization ability.

4.3.2 Knowledge Distillation-based Methods

In continuous learning, KD is widely employed to transfer knowledge from an old model, known as the "teacher model," to a new model, referred to as the "student model" [91]. It effectively addresses catastrophic forgetting in continuous learning. However, in FSCIL, the data distribution between the base and incremental sessions is imbalanced, with sufficient samples in the base session and limited samples in the incremental session. Conventional KD methods for continuous learning are prone to overfitting in the incremental session and further biases in future incremental sessions [73]. Nevertheless, many studies have explored the application of KD to FSCIL, focusing on transferring knowledge between sessions using KD. Based on the approach to address the challenges of data imbalance and overfitting in FSCIL, these studies can be classified into two categories: KD by balancing data and optimized KD.

Knowledge Distillation-based Methods with Balanced Data: To address the inadequacy of KD methods in FSCIL due to data imbalance, some approaches [18, 44, 92] address the issue by selecting an equal number of samples from the base and incremental sessions for distillation, avoiding bias. For instance, *Dong et al.* [92] proposed a relation KD framework. They constructed a sample relation graph to represent learned knowledge, ensuring balance by selecting an equal number of samples from each base class. The samples were chosen based on the angles between their feature vectors, removing redundancies until the desired K samples remained. A sample relation loss function was introduced to discover the relationship knowledge among different classes, facilitating the distillation of sample relationships and the propagation of structural information in the graph. Additionally, as introduced in Sec. 4.1.1, *Zhu et al.* [44] addressed overfitting and knowledge transfer in FSCIL by fine-tuning the backbone and sampling base class samples.

Another solution to mitigate data imbalance and limited samples in FSCIL is introducing additional data to prevent overfitting. In the context of FSCIL, *Cui et al.* proposed a series of semi-supervised methods that leverage KD and unlabeled data [45, 46, 93]. As shown in Fig. 9(d), *Cui et al.* [45] introduced the Us-KD framework, which used an uncertainty-guided module to select unlabeled data to mitigate overfitting during knowledge transfer. The framework initially trained the model on base classes and stored some samples. In the incremental session, the model was initialized with the previous model's weights and updated using stored old samples and labeled samples from the current session through classification and distillation losses. Subsequently, the uncertainty-guided module selected and labeled unlabeled samples, combined with labeled samples to update the model iteratively. Finally, the stored data was updated with these samples. In their further research [46], they pointed out that well-learned or easily classifiable classes often have higher prediction probabilities. Thus, they designed a data selection method called "Class Equilibrium," where well-learned categories were assigned fewer samples, and poorly learned categories were assigned more samples. It is worth noting that they highlighted the potential unreliability of KD with unlabeled data. Thus, they introduced an uncertainty-aware distillation approach suitable for semi-supervised FSCIL, consisting of uncertainty-guided refinement and adaptive distillation loss. Refinement involved leveraging uncertainty assessment to filter reliable samples from the augmented dataset, while adaptive distillation adjusted the distillation loss weights based on the sample quantity.

Optimized Knowledge Distillation-based Methods: Some methods have innovatively optimized KD methods to address the issues caused by the characteristics of the FSCIL data stream. For example, *Cheraghian et al.* [17] proposed a semantic-aware KD framework. In the base session training, this framework first mapped the labels to word embeddings using natural language processing models. Then, the backbone was used to convert images into original features. Subsequently, a multi-head attention

model was trained using a super-class aggregation approach to prevent overfitting during the incremental process. Finally, a mapping model was trained to align the image features with the word embeddings. For incremental sessions, the labels of the new classes were first mapped to word embeddings. The mapping model was trained using fine-tuning and KD to further refine the image features. The classification was achieved by assessing the similarity between the image features and word embeddings. Additionally, *Zhao et al.* [73] proposed a class-aware bilateral distillation framework, which consists of two branches: the base branch and the novel branch. Two teacher models guided the learning of the novel branch. One teacher model was trained on base class data and possessed rich general knowledge to alleviate the overfitting of new classes. The other teacher model was updated from the previous incremental session and contained adaptive knowledge of the previous new classes to mitigate their forgetting. Fine-tuning leads to forgetting, and an attention-based aggregation module was inevitably introduced to further improve the performance of the base classes by selectively merging the predictions from the base branch and the novel branch.

Discussion: The applicability of KD in FSCIL depends on resolving challenges of imbalanced data distribution and overfitting with few-shot samples and its exacerbated effects resulting from incremental scenarios. The data-driven approach can address these challenges, including incorporating unlabeled data, establishing a balanced distribution, and employing sample relation distillation. Furthermore, optimizing the KD framework is another strategy. For instance, introducing semantic word embeddings as auxiliary information can be employed to optimize. These approaches aim to alleviate the above challenges and facilitate the effective application of KD in FSCIL.

4.3.3 Meta Learning-based Methods

FSCIL faces challenges of overfitting and catastrophic forgetting due to limited samples for continuous learning. Meta-learning, or "learning to learn," is a prominent approach to address these issues. Meta-learning leverages experiences distilled from multiple learning episodes, encompassing a distribution of related tasks, to enhance future learning performance [94]. In FSCIL, meta-learning is crucial in improving the model's adaptation ability. Building on the description in Sec. 4.1.2, most meta-learning methods in FSCIL are trained by pseudo-incremental tasks sampled from the base session. It proves effective for backbone training, special structure training, feature distribution learning, and various other applications in FSCIL.

One common application of meta-learning is to directly train backbone models by constructing a series of pseudo-incremental scenarios, enabling them to adapt to real incremental scenarios. For instance, in the C-FSCIL framework proposed by *Hersche et al.* [24], empirical evidence demonstrated that training the backbone using the meta-learning strategy effectively can extract robust features. Utilizing the average of these features to create class prototypes surpassed the state-of-the-art methods at that time. Moreover, meta-learning was employed to learn feature distributions in FSCIL. *Zheng and Zhang* [95] introduced meta-learned class structures to regulate the distribution of learned classes in the feature space. Class structures describe the distribution of learned classes in specific directions. They ensured discriminative class prototypes without interference by proposing a class structure regularizer consisting of direction vectors associated with class structures and an alignment kernel aligning sampled embeddings with the class structures. A novel loss function was also introduced to prevent interference between new and old prototypes. The model was trained on a series of constructed meta-learning tasks. Additionally, meta-learning can be utilized to train specially designed structures in FSCIL. In the LIMIT framework proposed by *Zhou et al.* [16], a series of pseudo-incremental tasks were sampled from the base session for meta-learning-based training. To mitigate bias issues caused by direct classification, a corrective model with a transformer as its core was introduced. The corrective model, trained using meta-learning and incorporating self-attention mechanisms, adjusted the biased relationship between old class classifiers and new class prototypes, ensuring that feature embeddings encompass contextual information. Similarly, the CEC framework mentioned in Sec. 4.1.2 combined pseudo-incremental sessions with meta-learning to train a graph attention network for regulating the relationships between prototypes.

4.3.4 Other Methods

In addition to the methods above, some studies focus on learning efficient feature representations to adapt to FSCIL through other approaches. For instance, unlike existing methods that attempt to overcome catastrophic forgetting when learning new tasks, *Shi et al.* [20] proposed a novel strategy to address this issue while learning base classes. The core idea was to identify the flat local minima of the loss function during base training and perform fine-tuning in the flat region during incremental sessions. This approach maximized the preservation of knowledge when conducting fine-tuning on novel classes. Specifically, since directly finding the flat local minima is challenging, they proposed adding random noise to the model parameters to approximate it during base training. In the incremental sessions, FSCIL was achieved through fine-tuning within the flat local range. The experiments showed effectiveness.

4.4 Summary

4.4.1 Performance Comparison

In this section, we summarize the performance of mainstream FSCIC methods. Since not all relevant methods are open-source and their implementation conditions and configurations (such as different backbone networks, feature fusion with other methods, and different learning paradigms) vary, we summarize the performance of mainstream FSCIC methods with similar backbones on three commonly used benchmark datasets in Table 4, including *mini*ImageNet, CIFAR-100, and CUB-200, to enable a fair comparison as much as possible. The performance values are extracted from corresponding papers. To fully demonstrate the characteristics of each method, Tab. 4 provides their types and specific taxonomy categories. In addition, we provide the backbone used by each method in this table. For methods with too many additional factors, we provide a supplementary table in the appendix for reference. Since some methods introduce extra auxiliary factors, we have specially included an "extra factor" column in the table to summarize these factors for each method. The performance of FSCIC methods is primarily evaluated by measuring the accuracy achieved on different incremental sessions, AA across all sessions, and PD values. Given the space limitations, we only provide accuracy for the starting and ending sessions, AA, and PD. Moreover, we summarize the highlights of each method in these tables.

For FSCIC methods, the performance of the backbone achieved on the base session is crucial for subsequent IL. From the analysis of SA across the three datasets in Tab. 4, it can be seen that the top five methods on *mini*ImageNet

TABLE 4
The performance of mainstream FSCIC methods. The data are extracted from the original papers. For *mini*ImageNet and CUB-200, all the methods employ ResNet18 as the backbone. For CIFAR-100, related methods have two different settings, ResNet18 and ResNet20. To provide a comprehensive comparison, we report them in this table together, with the details noted. The accuracy of the first and last sessions is abbreviated as SA and EA. We use "-" to mark the dataset without reporting in the original papers. The best results are bold and underlined, while the second-best are underlined only. (In %)

Type	Taxonomy	Methods	Venue	*mini*ImageNet					CIFAR-100					CUB-200					Highlights
				Backbone (ResNet)	SA	EA	AA	PD↓	Backbone (ResNet)	SA	EA	AA	PD↓	Backbone (ResNet)	SA	EA	AA	PD↓	
Data-based	Data Replay	FSIL-GAN [68]	ACM MM 22	18	69.87	46.14	56.40	23.73	18	70.14	46.61	55.31	23.53	18	81.07	59.13	69.26	21.94	Proposed a semantics-driven generative replay framework
	Data Replay	ERDFR [19]	ECCV 22	18	71.84	48.21	58.02	23.63	20	74.40	50.14	60.78	24.26	18	75.90	52.39	61.52	23.51	Proved the effectiveness of DR in FSCIL and proposed a data-free replay method
	Data Replay Knowledge Distillation	FDD [44]	PRAI 22	18	64.14	42.01	52.44	22.13	18	64.94	41.73	52.52	23.21	18	-	-	-	-	Fixed the shallow layers and fine-tune the deep layers with KD loss and CE loss
	Pseudo Scenarios	SPPR [69]	CVPR 21	18	61.45	41.92	52.76	19.53	18	63.97	43.32	54.45	20.65	18	68.68	37.33	49.32	31.35	Built a randomly episodic training based on a novel self-promoted prototype refinement mechanism
	Pseudo Scenarios	FACT [23]	CVPR 22	18	72.56	50.49	60.70	22.07	20	74.60	52.10	62.24	22.50	18	75.90	56.94	64.42	18.96	Proposed a forward compatible training strategy, which reserves embedding space for new classes during base learning
	Pseudo Scenarios Representation Learning	ALICE [47]	ECCV 22	18	80.60	55.70	63.99	24.90	18	79.00	54.10	63.21	24.90	18	77.40	60.10	65.75	**17.30**	ALICE used angular penalty loss for discriminated and generalized feature learning
	Pseudo Scenarios Representation Learning	SAVC [96]	CVPR 23	18	**81.12**	**57.11**	**67.05**	24.01	20	78.77	53.12	63.63	25.65	18	**81.85**	62.50	69.35	19.35	Used supervised contrast learning and virtual classes to initialize backbone so that it can have good generalization performance
Structure-based	Dynamic Structure	TOPIC [13]	CVPR 20	18	61.31	24.42	39.64	36.89	18	64.10	29.37	42.62	34.73	18	68.68	26.28	43.92	42.40	Introduced FSCIL setting for the first time, and proposed TOPIC framework based on neural gas network
	Dynamic Structure Pseudo Scenarios Attention	CEC [14]	CVPR 21	18	72.00	47.63	57.75	24.37	20	73.07	49.14	59.53	23.93	18	75.85	52.28	61.33	23.57	Proposed CEC framework based on GAT to propagate context information, and it was trained pseudo-incremental sessions
	Dynamic Structure	DSN [25]	TPAMI 22	18	66.95	45.89	54.39	21.06	18	73.00	50.00	60.14	23.00	18	80.86	**63.21**	**71.02**	17.65	Proposed DSN, an adaptively updating network with compressive node expansion to "support" the feature space
Optimization-based	Representation Learning	SFbFSCIL [79]	ICCV 21	18	61.37	42.23	50.73	19.14	18	62.00	41.69	50.86	20.31	18	68.78	43.23	51.84	25.55	Proposed the use of a mixture of subspaces and synthesized features by VAE to reduce the forgetting and overfitting problem
	Representation Learning	FSLL+SS [21]	AAAI 21	18	68.85	43.92	53.92	24.93	18	66.76	39.57	48.71	27.19	18	75.63	55.82	62.62	19.81	FSLL trained only a few selected parameters to limit overfitting, and leverages self-supervision
	Representation Learning	MgSvF [22]	TPAMI 21	18	63.09	45.76	52.87	**17.33**	18	74.24	51.40	61.67	22.84	18	72.29	54.33	62.37	17.96	MgSvF used frequency-aware regularization and feature space composition to balance old and new knowledge
	Representation Learning	CLOM [26]	NeurIPS 22	18	73.08	48.00	58.48	25.08	20	74.20	50.25	60.57	23.95	18	79.57	59.58	67.17	19.99	Interpreted the dilemma of the margin-based classification as a class-level overfitting problem and proposed CLOM to mitigate it
	Representation Learning	RE [90]	JEI 22	18	70.74	45.48	56.30	25.26	18	70.72	47.52	57.77	23.20	18	-	-	-	-	Proposed representation enhancement method by exploring correlations with previously learned classes
	Representation Learning	WaRP [80]	ICLR 23	18	72.99	50.65	59.69	22.34	20	**80.31**	54.74	65.82	25.57	18	77.74	57.01	64.66	20.73	WaRP, a weight space rotation process that compressed old knowledge into key parameters, allowing fine-tuning without forgetting
	Representation Learning	TEEN [97]	NeurIPS 23	18	73.53	52.08	61.45	21.45	20	74.92	52.64	63.10	22.28	18	77.26	59.31	66.63	18.13	TEEN, a training-free calibration strategy, enhanced new class discriminability by fusing new and weighted base class prototypes.
	Knowledge Distillation Attention	SaKD [17]	CVPR 21	18	61.33	38.73	48.20	22.60	18	64.03	34.94	45.53	29.09	18	68.23	32.96	46.13	35.27	Proposed a semantic-aware distillation method and an attention driven alignment strategy to mitigate catastrophic forgetting
	Knowledge Distillation	ERL++ [92]	AAAI 21	18	61.71	40.77	49.84	20.94	18	73.70	48.25	59.31	25.45	18	73.52	52.28	61.18	21.24	Proposed the exemplar relation distillation and degree-based graph construction method to model the exemplar relationship
	Knowledge Distillation Attention	BiDistFSCIL [73]	CVPR 23	18	74.65	52.22	61.42	22.43	18	79.45	**55.88**	**66.14**	23.57	18	79.12	60.93	67.34	18.19	It proposed a KD strategy with two teacher models, designed a two-branch network, and used an attention mechanism to aggregate the predictions from both branches.
	Meta Learning Attention	MetaFSCIL [74]	CVPR 22	18	72.04	49.19	58.85	22.85	20	74.50	49.97	60.79	24.53	18	75.90	52.64	61.93	23.26	Adopted a bi-level meta-learning optimization and bi-directional guided modulation approach
	Meta Learning	CSR [95]	ICDMW 21	18	67.67	44.52	54.11	23.15	18	72.02	49.00	59.07	23.02	18	74.69	55.09	62.32	19.60	Introduced class structures and adopted an alignment kernel, employing meta-learning process
	Meta Learning Attention	LIMIT [16]	TPAMI 22	18	72.32	49.19	59.06	23.13	20	73.81	51.23	61.84	22.58	18	75.89	57.41	65.48	18.48	LIMIT, a meta-learning based paradigm, which synthesized fake tasks to build a generalizable feature space for unseen tasks
	Others	F2M [20]	NeurIPS 21	18	67.28	44.65	54.89	22.63	18	64.71	44.67	53.65	**20.04**	18	81.07	60.26	69.49	20.81	Proposed a method by finding flat local minima during base training and fine-tuning within this region when learning new classes

are: SAVC [96] (81.12%), ALICE [47] (80.60%), BiDistFSCIL [73] (74.65%), TEEN [97] (73.53%), and CLOM [26] (73.08%). On CIFAR-100, the top five methods are: WaRP [80] (80.31%), BiDistFSCIL (79.45%), ALICE (79.00%), SAVC (78.77%), and TEEN (74.92%). On the CUB-200 dataset, the top five methods are: SAVC (81.85%), F2M [20] and FSIL-GAN [68] (81.07%), DSN [25] (80.86%), and BiDistFSCIL (79.12%). It is noteworthy that the SAVC, based on virtual class synthesis, achieved the best initial performance on *mini*ImageNet and CUB-200; the ALICE framework, which leverages metric learning and pseudo-data synthesis, also performed exceptionally well on these datasets. This indicates that the virtual class strategy is highly effective in improving performance in the initial session. Apart from that, it has been found that almost all methods that achieved top five performance on the base session also achieved top five performance in terms of the AA index. It reflects the influence of the performance achieved on the base session.

The performance obtained in the final session reflects the learning capability of the FSCIC model for incremental classes and the stability of keeping old knowledge. However, as the model learned in each incremental session will be tested on all seen classes, and the number of classes involved in the base session is large, the performance obtained on each session cannot fully represent the model's IL ability. In contrast, the PD value can better reflect the model's ability to resist forgetting. In Tab. 4, the top five methods with the lowest PD values on *mini*ImageNet are: MgSvF [22] (17.33%), SFbFSCIL [79] (19.14%), SPPR [69] (19.53%), ERL++ [92] (20.94%), and DSN (21.06%). On CIFAR-100, the top five methods are: F2M [20] (20.04%), SFbFSCIL (20.31%), SPPR (20.65%), TEEN (22.28%), and FACT [23] (22.50%). On CUB-200, the top five methods are: ALICE (17.30%), DSN (17.65%), MgSvF (17.96%), TEEN (18.13%), and BiDistFSCIL(18.19%). It can be seen that SPPR, which constructs pseudo incremental sessions, and SFbFSCIL,

which is based on feature space fusion and VAE feature synthesis, both have achieved good PD values on the first two datasets. DSN, based on dynamic network structure, and MgSvF, based on frequency domain analysis, performed well on the first and last datasets. Furthermore, combining the performance of other methods on the three datasets, it can be found that techniques such as KD, pseudo-incremental scenario construction, dynamic structures, and feature optimization can effectively alleviate the catastrophic forgetting problem.

4.4.2 Main Issues and Facts

In FSCIC, the current issues primarily encompass a lack of comprehensive evaluation metrics, unfairness in experimental conditions, and inconsistencies with real-world scenarios. Most studies use AA or PD values to measure model performance, but they can not reflect the performance details during the continuous learning process [47]. Furthermore, the variability in choosing backbone networks and the introduction of additional data introduce inherent biases when comparing different methodologies. Most importantly, the current setting of FSCIC faces challenges in real-world implementation.

5 Few-shot Class-incremental Object Detection

Since the instance segmentation framework in FSCIL generally has object detection capabilities, this section discusses them together. Firstly, the difference with FSCIC is presented. Then, existing methods are systematically summarized from the perspectives of anchor-free and anchor-based frameworks. Finally, the paper summarizes the entire work, including performance comparisons and discussions of key issues.

5.1 Difference with Classification

In contrast to the classification task in FSCIL, FSCIOD aims to enable the model to continuously learn new classes from limited samples while achieving accurate localization (using bounding box regression or segmentation) and classification of each corresponding individual object in an image [98, 99, 100]. The model is also required to retain the capability of object localization and classification for the old classes.

Similar to the classification setting in FSCIL provided in Sec. 2.1, the training data for FSCIOD can be divided into the base and new training sets. However, there is a difference. In the classification task, the new classes are typically further divided into multiple incremental sessions in the form of $N-$way $K-$shot, while in the current object detection setting, the new classes usually are formed as one incremental session. Specifically, the training sets for FSCIOD can be denoted as $\{D^b_{train}, D^n_{train}\}$, where the base training set D^b_{train} contains a large number of labeled training samples and can be represented as $D^b_{train} = (x_i, y_i)^{n_0} i = 1$, where x_i, y_i, and n_0 represent the training sample, its corresponding ground truth set, and the number of base samples, respectively. Similar to the classification setting, the new training set $D^n_{train} = \{(x_i, y_i)\}^{N\times K}_{i=1}$ is in the form of $N-$way $K-$shot. Note that the classes in the base and new training sets do not intersect. The evaluation process for the object detection task in FSCIL is similar to the classification task. After learning the new training set, the model is evaluated on the performance of all seen classes, *i.e.*, the union of testing data from all seen classes.

It is important to note that in incremental images, even if a single image contains multiple objects of different classes, only the ground truth set for the current class is provided to align with the few-shot class-incremental setup.

5.2 Methods

FSCIOD requires simultaneously localizing and classifying new class objects during IL while not forgetting the old knowledge. This poses a greater challenge compared to classification in FSCIL. Current methods include both anchor-based and anchor-free frameworks. Generally, anchor-based detectors have superior detection performance, but they suffer from lower efficiency and flexibility due to the design of anchors. On the other hand, anchor-free detectors are more efficient and flexible.

5.2.1 Anchor-free Frameworks

Recently, some studies [62, 99, 100, 101, 102] have adopted anchor-free frameworks to perform this task. The reason is that anchor-free frameworks can effectively handle incremental classes without defining anchor boxes. According to their detection framework, these studies can be classified into three categories: CentreNet-based, FCOS-based, and DETR-based methods.

CentreNet-based Methods: CentreNet [103] redefined object detection as a *point+attribute* regression problem. During detection, it divided the input image into different regions, each with a centre point. CentreNet made predictions to determine whether the centre point corresponds to an object. Then, it predicted the class and confidence for this object. CentreNet also adjusted the centre point to obtain the accurate location and regressed the object's width and height. By maintaining independent prediction heatmaps for each class and using activation thresholding for independent object detection, CentreNet supported incremental registration of new classes. Based on CentreNet, *Perez-Rua et al.* [62] proposed the ONCE framework, which incorporated meta-learning for object detection in FSCIL. It decomposed CentreNet into a fixed universal feature extractor trained on base classes and a meta-learned object localizer with class-specific parameters. In the few-shot incremental detection scenario, the model only required forward propagation for registration without model updating or accessing base data. Additionally, *Cheng et al.* [101] also utilized CentreNet as the backbone and introduced meta-learning based on MAML [67]. First, meta-learning provided good initialization for the object localizer based on base data, enabling easy fine-tuning with few-shot samples from new classes. Furthermore, the filter parameters of base classes were retained. The meta-learner determined the remaining parameters of the object localizer. The study also concluded that the main factor limiting the performance of new classes is the overfitting of the feature extractor to base classes, resulting in insufficient generalization.

FCOS-based Methods: Similarly, recent works have adopted it as a backbone due to the strong performance and class-agnostic localization capability of FCOS [104]. For instance, Sylph proposed by *Yin et al.* [99] decomposed the detection framework into a class-agnostic detector and a novel classifier to enable continual learning of new classes. Specifically, FCOS was employed in Sylph for class-agnostic object localization. Since optimizing softmax can lead to catastrophic forgetting [99, 105], Sylph replaced it with multiple binary sigmoid-based classifiers, each independently handling its own set of parameters. When adding new classes, a new set of classifier parameters can be generated with zero interference between predictions of different classes. In addition, *Feng et al.* [102] proposed two modules inspired by the phenomenon of establishing new connections between memory

cells in the brain when new memories appear. The first was called the MCH module, which added a classification branch to predict new classes each time they appeared. The second was called the BPMCH module, which added a new backbone that was initialized with the weights of the base class backbone to transfer more knowledge from the base classes to the new classes. In this work, FCOS and ATSS [106] were employed as the baseline detectors. Training started on the base classes and was then fine-tuned on the new classes, ensuring the retention of knowledge learned from the base classes and transferring that knowledge to the new classes.

DETR-based Method: In anchor-free frameworks, in addition to the methods based on CentreNet and FCOS, another work adopts the DETR framework [107] as the backbone. Specifically, *Dong et al.* [100] proposed the incremental-DETR, which firstly introduced DETR to FSCIOD. This method consisted of two stages: First, the entire network was pre-trained using a large amount of data from the base classes, and the class-specific components of DETR (including the projection layer and classification head for specific classes) were fine-tuned using self-supervision from additional object proposals generated by selective search algorithm [108] as pseudo labels. Then, the CNN backbone, transformer, and regression head were fixed, and an incremental few-shot fine-tuning strategy was introduced to fine-tune and distill knowledge from the class-specific components of DETR. This strategy encouraged the framework to detect new classes without catastrophic forgetting.

5.2.2 Anchor-based Frameworks

In addition to anchor-free frameworks, there have been some studies [98, 109] that adopt the anchor-based framework, Mask R-CNN [110], to address object detection and instance segmentation in FSCIL. Mask R-CNN is a popular framework for the instance segmentation, which extended the Faster R-CNN [111] architecture by incorporating a mask prediction branch. It is a two-stage approach that combines object detection and pixel-level segmentation into one framework. Currently, there is limited research on instance segmentation in FSCIL, and all utilize Mask R-CNN as the backbone. For example, *Ganea et al.* [98] proposed the iMTFA framework while initially introducing the setting of few-shot incremental instance segmentation. Specifically, they added an instance segmentation branch (similar to Mask R-CNN to Faster R-CNN) to the few-shot object detection framework TFA [112], resulting in MTAF. One drawback of MTAF was that it required continual fine-tuning when adding new classes. Thus, they extended MTFA to an incremental method called iMTFA. In this framework, the regression and mask prediction heads were class-agnostic. Additionally, the framework learned a feature extractor that generates discriminative features. The feature extractor was used for new classes to compute the averaged prototype vectors for each class, which were then concatenated with the existing classifier. This enabled few-shot incremental instance segmentation without the need for further training. Furthermore, *Nguyen and Todorovic* [109] extended the Mask R-CNN framework in the second stage: a new object class classifier based on the probit function [113] and a new uncertainty-guided bounding box predictor. The former utilized Bayesian learning to address the scarcity of training examples for new classes. The latter not only predicted object bounding boxes but also estimated the uncertainty of the predictions, which guided the refinement of bounding boxes. Two new loss functions were also specified based on the estimated object-class distribution and bounding-box uncertainty.

5.3 Summary

5.3.1 Performance Comparison

In this section, we summarize the performance of FSCIOD methods. We summarize the performance of relevant methods on COCO and VOC in Tab. 5. To fully elucidate the attributes of each method, Tab. 5 includes their types and specific taxonomy categories. In addition, the backbone employed by each method is furnished in this table. Because some methods can achieve object detection and instance segmentation simultaneously, we have added a "task" column in Tab. 5 to denote performance on related tasks. FSCIOD methods are evaluated in two ways: standard evaluation on COCO and cross-dataset evaluation on COCO and VOC. Tab. 5 presents mAP, mAP50, and mAR values, along with method highlights.

Given that the FSCIOD evaluation is usually conducted under different sample shots, we analyze and summarize based on the overall performance of relevant methods. It can be found from Tab. 5, the top three performance methods for object detection achieved on base classes are Sylph [99], iFS-RCNN [109], and MCH [102]. The top three performance methods for novel COCO classes are iFS-RCNN, Incremental-DETR [100], and iMTFA [98]. The top three methods for overall performance on COCO are iFS-RCNN, Sylph, and MCH. Among all methods for cross-dataset evaluation on VOC, the top three performers are: Incremental-DETR, BPMCH [102], and MCH. Therefore, it can be seen that iFS-RCNN based on complex Mask RCNN yields the best results, and Sylph and MCH, which are based on simple FCOS, also show good performance. In instance segmentation, only anchor-based methods have conducted the relevant evaluation, among which iMTFA has the overall best incremental segmentation ability on novel classes, but iFS-RCNN performs best on base classes. In summary, anchor-based methods are suitable for object detection and instance segmentation scenarios, with excellent performance but more complex structures; anchor-free methods are suitable for application scenarios requiring lower framework complexity and can achieve performance slightly inferior to anchor-based methods.

5.3.2 Main Issues and Facts

The current FSCIOD mainly faces the issue of insufficient research. In addition, the performance of current research is relatively poor compared to supervised learning methods, especially in detecting novel classes, which is far from the level of practical application. Furthermore, similar to FSCIC, FSCIOD also faces the problem of a need for more suitable evaluation metrics. The evaluation metrics used by different works vary slightly and are not yet unified.

6 Conclusion and Outlooks

In this paper, we present a comprehensive and systemic survey of FSCIL, covering its background and significance, problem definition, core challenges, general schemes, relations with related problems, datasets, evaluation protocols, and metrics. We focused on the classification and object detection tasks in FSCIL, summarized the relevant works, analyzed their performance, and summarized the main issues and facts faced by FSCIL. Considering that FSCIL is still in its infancy, we attempt to offer valuable insights and discuss potential directions.

6.1 Human-machine Gap in FSCIL

The memory learning in the human brain can be categorized into three main processes: encoding, storage, and retrieval [114]. In

TABLE 5
The performance of FSCIOD methods. The data are extracted from the original papers. The taxonomy is abbreviated to taxo. We use "-" to mark the dataset without reporting in the original papers. (In %)

Type	Taxo.	Method	Venue	Backbone (ResNet)	Task	Shot	COCO									VOC			Highlights
							Base			Novel			Overall			Novel			
							mAP	mAP50	mAR	mAP	mAP50	mAR	mAP	mAP50	mAR	mAP	mAP50	mAR	
Anchor-free	CentreNet-based	ONCE [62]	CVPR 20	**50**	D	**1**	17.90	-	19.50	0.70	-	6.30	13.60	-	16.20	-	-	-	Proposed FSCIOD setting and introduced the first work, ONCE
						5	17.90	-	19.50	1.00	-	7.40	13.70	-	16.40	2.40	-	12.20	
						10	17.90	-	19.50	1.20	-	7.60	13.70	-	16.50	2.60	-	11.60	
		SS [101]	TCSVT 21	**50**	D	**1**	26.90	-	25.80	0.90	-	4.20	20.40	-	20.40	1.50	2.30	6.10	Proposed new models, redesigning the CenterNet and incorporating a novel meta-learning method, MAML, to perform FSCIOD task
						5	29.20	-	27.30	1.40	-	7.10	22.30	-	22.20	3.10	5.50	12.00	
						10	27.40	-	25.90	1.50	-	7.90	20.90	-	21.40	3.80	6.50	13.50	
		MS [101]				1	30.70	-	27.60	1.50	-	5.50	23.40	-	22.00	2.50	4.50	8.50	
						5	33.30	-	29.10	2.50	-	9.10	25.60	-	24.10	5.00	9.70	14.60	
						10	31.40	-	27.80	2.60	-	9.60	24.20	-	23.30	6.20	11.40	15.80	
	FCOS-based	Sylph [99]	CVPR 22	50	D	1	37.60	-	-	1.10	-	-	28.48	-	-	-	-	-	Introduced FCOS-based Sylph, decoupling object detection into classification and localization
						5	42.40	-	-	1.50	-	-	32.18	-	-	-	-	-	
						10	42.80	-	-	1.70	-	-	32.53	-	-	-	-	-	
		MCH [102]	PRL 22	50	D	1	36.90	-	-	0.40	-	-	27.70	-	-	1.00	-	-	Introduced MCH and BPMCH, human memory-inspired models, outperforming ONCE by effectively transferring knowledge from base to novel classes
						5	36.00	-	-	5.50	-	-	28.30	-	-	14.30	-	-	
						10	35.50	-	-	7.80	-	-	28.60	-	-	18.30	-	-	
		BPMCH [102]				1	29.40	-	-	2.40	-	-	22.60	-	-	6.10	-	-	
						5	36.00	-	-	6.40	-	-	28.60	-	-	16.40	-	-	
						10	35.60	-	-	7.00	-	-	28.50	-	-	17.60	-	-	
	DETR -based	Incremental -DETR [100]	AAAI 23	50	D	1	29.40	47.10	-	1.90	2.70	-	22.50	36.00	-	4.10	6.60	-	Developed Incremental-DETR for FSCID, which uses self-supervised learning and a fine-tuning strategy
						5	30.50	48.40	-	8.30	13.30	-	24.90	39.60	-	16.60	26.30	-	
						10	27.30	44.00	-	14.40	22.40	-	24.10	38.60	-	24.60	38.40	-	
Anchor-based	Mask RCNN-based	iMTFA [98]	CVPR 21	50	D	1	27.81	40.11	-	3.23	5.89	-	21.67	31.55	-	-	-	-	Proposed instance segmentation setting in FSCIL and introduced the first work, iMTFA, which can perform both instance segmentation and object detection
						5	24.13	33.69	-	6.07	11.15	-	19.62	28.06	-	-	-	-	
						10	23.36	32.41	-	6.97	12.72	-	19.26	27.49	-	-	-	-	
					S	1	25.90	39.28	-	2.81	4.72	-	20.13	30.64	-	-	-	-	
						5	22.56	33.25	-	5.19	8.65	-	18.22	27.10	-	-	-	-	
						10	21.87	32.01	-	5.88	9.81	-	17.87	26.46	-	-	-	-	
		iFS-RCNN [109]	CVPR 22	50	D	1	40.08	-	-	4.54	-	-	31.19	-	-	-	-	-	Introduced iFS-RCNN, an extension of Mask-RCNN, leveraging probit function and uncertainty-guided bounding box prediction for instance segmentation and object detection in FSCIL
						5	40.06	-	-	9.91	-	-	32.52	-	-	-	-	-	
						10	40.05	-	-	12.55	-	-	33.02	-	-	-	-	-	
					S	1	36.35	-	-	3.95	-	-	28.45	-	-	-	-	-	
						5	36.33	-	-	8.80	-	-	28.89	-	-	-	-	-	
						10	36.32	-	-	1.06	-	-	30.41	-	-	-	-	-	

the encoding phase, the brain efficiently processes information through associative learning and abstract thinking, effectively encoding features of new categories even with limited samples. During the storage phase, the hippocampus converts short-term memories into long-term memories, forming stable neural networks across different regions of the cerebral cortex. In the retrieval phase, existing memories may be consolidated, updated, or actively forgotten in conjunction with new information, leading to the formation of memories adapted to the current environment. This sequence of processes highlights the brain's efficient knowledge handling capabilities.

Currently, some IL research, such as the method proposed by *ZKudithipudi et al.* [115] that emulates the Drosophila's mushroom body's mechanisms, attempts to enhance model memory capabilities by bio-inspired intelligence. However, a systematic bio-inspired approach in FSCIL is still lacking. Current FSCIL models lack associative learning and abstract thinking in limited sample learning, and there is room for improvement in prior knowledge acquisition. These models typically use one model for storing all knowledge from continual learning, suggesting the need for exploring multi-modular knowledge storage and long-short term memory mechanisms. Additionally, FSCIL requires proactive strategies for knowledge consolidation, updating, and personalized management, such as actively forgetting infrequent knowledge, reinforcing challenging knowledge, and integrating consistent knowledge.

6.2 Practical Settings in FSCIL

The current FSCIL setting, based on *Tao et al.* [13], is idealistic. The real world requires practical settings. Some research has improved the FSCIL setting to better adapt to the real-world, for example: (a) *FSCIL with limited base samples*: Ensuring that the base session has abundant samples is challenging in some situations. Thus, *Kalla and Biswas* [86] suggested the FSCIL-lb setting with fewer required base training samples; (b) *FSCIL with imbalanced sessions*: Considering the practical difficulty in ensuring the $N-$way $K-$shot format, *Kalla and Biswas* [86] proposed the FSCIL-im setting, where the incremental sessions appear with an imbalanced data distribution; (c) *Semi-supervised FSCIL*: Some scenarios have some available unlabeled data. *Cui et al.* [93] leveraged them to propose semi-supervised FSCIL.

Despite some efforts to propose settings that better match real-world situations, some directions are still worth exploring: (a) *Cross-domain FSCIL*: Considering the domain changes in the real world (e.g., changes in imaging condition and environment), FSCIL should be robust under cross-domain conditions; (b) *FSCIL with repetition*: The no repetition constraint of current FSCIL doesn't reflect practical scenarios where class recurrence is common. Researching how to utilize these repetitions (considering that current samples may be scarce, but could increase in the future) can improve the practicality; (c) *Incomplete FSCIL*: In real-world scenarios, where most classes have ample training samples but some are scarce, the assumption of uniformly few-shot data is unrealistic. Hence, investigating incomplete FSCIL, encompassing incremental sessions with classes of varying sample availability, is also meaningful; (d) *Federated FSCIL*: Combining the privacy and distributed features of federated learning with FSCIL's ability to learn from limited data, this method aims to create models that are privacy-aware and adaptable to multiple clients with limited and dynamic data.

6.3 Knowledge Acquisition and Update in FSCIL

FSCIL involves a continuous learning process with base and incremental sessions, so knowledge acquisition and update are analyzed in two parts:

Base Stage: Effective initialization of the backbone is crucial for ensuring base class performance and generalization for fu-

ture incremental classes. Current methods often rely on a large number of base class samples for backbone network initialization, which may not align with reality and whose generalization capabilities are difficult to accurately assess. To enhance generalization, researchers have tried introducing strategies like self-supervised learning and forward compatibility, but these usually depend on sufficient base class data. There is a lack of research on initial knowledge acquisition without specific requirements for the base data. Therefore, exploring methods to enrich initial stage knowledge acquisition is important. From the data perspective, increasing data diversity and improving knowledge learning strategies, such as exploring data augmentation, data generation, introducing unsupervised data, and optimizing backbone learning methods, are essential. Additionally, introducing pre-trained models and other prior knowledge can be considered. For instance, foundation models like CLIP, SAM, and GPT, which combine self-supervised or semi-supervised pre-training with prompt engineering, have shown excellent generalization and transfer capabilities, offering new possibilities for enhancing FSCIL model performance. Some recent works have attempted to incorporate foundation models to address FSCIL challenges. For example, *D'Alessandro et al.* [116] designed a prompt learning strategy for CLIP in FSCIL, and *Zhang et al.* [117] leveraged the RETFound foundation model to enhance feature learning in few-shot class-incremental retinal disease recognition. However, these approaches have not yet become mainstream, and attention to fairness in experimental comparisons is still necessary.

Incremental Stage: In incremental sessions, models typically initialize with weights from prior phases, focusing on learning new classes and preserving existing knowledge. Challenges arise from limited new samples and restricted access to complete old data, making effective learning of new categories and old knowledge retention pivotal. Current solutions include freezing the backbone network and using class prototype averaging, demanding robust generalization and discrimination from the network, yet possibly leading to reduced performance as new classes increase. An alternative is maintaining key parameters for new class learning, though this risks diminishing old class performance and complicates parameter evaluation due to deep learning models' opaque nature. KD is also commonly used, but how to effectively learn new categories and select efficient old samples for distillation is still a direction to be further explored.

6.4 Applications and Safety in FSCIL

Application Scenarios: Current FSCIL research mainly targets image classification, with emerging yet non-systematic studies in visual object detection, natural language processing, lip reading, remote sensing, and robotics. Most work evaluates performance on benchmark datasets, with real-world applications still evolving. In many application scenarios, the demand for the few-shot continuous learning capability is significant. For instance, in applications like video analysis, service robotics in hotels, and autonomous driving, the need for FSCIL technology is evident. These fields often require learning new classes from limited data, maintaining high accuracy with scarce samples, and adapting to new categories in dynamic environments, underscoring FSCIL's importance and potential.

Privacy and Safety: Privacy and Security: Privacy protection is a key issue in the application of FSCIL. To address catastrophic forgetting, some FSCIL studies store old category samples for replay, which could lead to privacy breaches when dealing with tasks involving private data. Currently, research on privacy protection in FSCIL is relatively limited, especially in the context of the increasing prevalence of deep learning technologies. Despite improvements in FSCIL's accuracy, AI systems based on deep learning are susceptible to security threats like adversarial and data poisoning attacks. Therefore, in-depth research into the security and privacy protection aspects of FSCIL is essential for its widespread application across various scenarios.

REFERENCES

[1] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *NeurIPS*, vol. 25, 2012.

[2] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *CVPR*, 2016, pp. 770–778.

[3] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," *arXiv preprint arXiv:1810.04805*, 2018.

[4] T. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell *et al.*, "Language models are few-shot learners," *NeurIPS*, vol. 33, pp. 1877–1901, 2020.

[5] M. De Lange, R. Aljundi, M. Masana, S. Parisot, X. Jia, A. Leonardis, G. Slabaugh, and T. Tuytelaars, "A continual learning survey: Defying forgetting in classification tasks," *IEEE TPAMI*, vol. 44, no. 7, pp. 3366–3385, 2022.

[6] G. I. Parisi, R. Kemker, J. L. Part, C. Kanan, and S. Wermter, "Continual lifelong learning with neural networks: A review," *Neural Networks*, vol. 113, pp. 54–71, 2019.

[7] M. Masana, X. Liu, B. Twardowski, M. Menta, A. D. Bagdanov, and J. van de Weijer, "Class-incremental learning: Survey and performance evaluation on image classification," *IEEE TPAMI*, pp. 1–20, 2022.

[8] D. Li and Z. Zeng, "Crnet: A fast continual learning framework with random theory," *IEEE TPAMI*, 2023.

[9] Y. Wu, Y. Chen, L. Wang, Y. Ye, Z. Liu, Y. Guo, and Y. Fu, "Large scale incremental learning," in *CVPR*, 2019, pp. 374–382.

[10] Q. Wang, R. Wang, Y. Wu, X. Jia, and D. Meng, "Cba: Improving online continual learning via continual bias adaptor," in *ICCV*, 2023, pp. 19 082–19 092.

[11] G. M. van de Ven, T. Tuytelaars, and A. S. Tolias, "Three types of incremental learning," *Nature Machine Intelligence*, pp. 1–13, 2022.

[12] M. Ren, R. Liao, E. Fetaya, and R. Zemel, "Incremental few-shot learning with attention attractor networks," *NeurIPS*, vol. 32, 2019.

[13] X. Tao, X. Hong, X. Chang, S. Dong, X. Wei, and Y. Gong, "Few-shot class-incremental learning," in *CVPR*, 2020, pp. 12 183–12 192.

[14] C. Zhang, N. Song, G. Lin, Y. Zheng, P. Pan, and Y. Xu, "Few-shot incremental learning with continually evolved classifiers," in *CVPR*, 2021, pp. 12 455–12 464.

[15] S. Gidaris and N. Komodakis, "Dynamic few-shot visual learning without forgetting," in *CVPR*, 2018, pp. 4367–4375.

[16] D.-W. Zhou, H.-J. Ye, L. Ma, D. Xie, S. Pu, and D.-C. Zhan, "Few-shot class-incremental learning by sampling multi-phase tasks," *IEEE TPAMI*, 2022.

[17] A. Cheraghian, S. Rahman, P. Fang, S. K. Roy, L. Petersson, and M. Harandi, "Semantic-aware knowledge distillation for few-shot class-incremental learning," in *CVPR*, 2021, pp. 2534–2543.

[18] A. Kukleva, H. Kuehne, and B. Schiele, "Generalized and incremental few-shot learning by explicit learning and calibration without forgetting," in *ICCV*, 2021, pp. 9020–9029.

[19] H. Liu, L. Gu, Z. Chi, Y. Wang, Y. Yu, J. Chen, and J. Tang, "Few-shot class-incremental learning via entropy-regularized data-free replay," in *ECCV*, 2022, pp. 146–162.

[20] G. Shi, J. Chen, W. Zhang, L.-M. Zhan, and X.-M. Wu, "Overcoming catastrophic forgetting in incremental few-shot learning by finding flat minima," *NeurIPS*, vol. 34, pp. 6747–6761, 2021.

[21] P. Mazumder, P. Singh, and P. Rai, "Few-shot lifelong learning," in *AAAI*, vol. 35, no. 3, 2021, pp. 2337–2345.

[22] H. Zhao, Y. Fu, M. Kang, Q. Tian, F. Wu, and X. Li, "Mgsvf: Multi-grained slow vs. fast framework for few-shot class-incremental learning," *IEEE TPAMI*, 2021.

[23] D.-W. Zhou, F.-Y. Wang, H.-J. Ye, L. Ma, S. Pu, and D.-C. Zhan, "Forward compatible few-shot class-incremental learning," in *CVPR*, 2022, pp. 9046–9056.

[24] M. Hersche, G. Karunaratne, G. Cherubini, L. Benini, A. Sebastian, and A. Rahimi, "Constrained few-shot class-incremental learning," in *CVPR*, 2022, pp. 9057–9067.

[25] B. Yang, M. Lin, Y. Zhang, B. Liu, X. Liang, R. Ji, and Q. Ye, "Dynamic support network for few-shot class incremental learning," *IEEE TPAMI*, 2022.

[26] Y. Zou, S. Zhang, Y. Li, and R. Li, "Margin-based few-shot class-incremental learning with class-level overfitting mitigation," *NeurIPS*, 2022.

[27] Y. Wang, Q. Yao, J. T. Kwok, and L. M. Ni, "Generalizing from a few examples: A survey on few-shot learning," *ACM Computing Surveys*, vol. 53, no. 3, pp. 1–34, 2020.

[28] J. Lu, P. Gong, J. Ye, J. Zhang, and C. Zhang, "A survey on machine learning from few samples," *Pattern Recognition*, vol. 139, p. 109480, 2023.

[29] S. Antonelli, D. Avola, L. Cinque, D. Crisostomi, G. L. Foresti, F. Galasso, M. R. Marini, A. Mecca, and D. Pannone, "Few-shot object detection: A survey," *ACM Computing Surveys*, vol. 54, no. 11s, pp. 1–37, 2022.

[30] G. Huang, I. Laradji, D. Vazquez, S. Lacoste-Julien, and P. Rodriguez, "A survey of self-supervised and few-shot object detection," *IEEE TPAMI*, 2022.

[31] T. Lesort, V. Lomonaco, A. Stoian, D. Maltoni, D. Filliat, and N. Díaz-Rodríguez, "Continual learning for robotics: Definition, framework, learning strategies, opportunities and challenges," *Information Fusion*, vol. 58, pp. 52–68, 2020.

[32] E. Belouadah, A. Popescu, and I. Kanellos, "A comprehensive study of class incremental learning algorithms for visual tasks," *Neural Networks*, vol. 135, pp. 38–54, 2021.

[33] Z. Mai, R. Li, J. Jeong, D. Quispe, H. Kim, and S. Sanner, "Online continual learning in image classification: An empirical survey," *Neurocomputing*, vol. 469, pp. 28–51, 2022.

[34] D.-W. Zhou, Q.-W. Wang, Z.-H. Qi, H.-J. Ye, D.-C. Zhan, and Z. Liu, "Class-incremental learning: A survey," *IEEE TPAMI*, 2024.

[35] L. Wang, X. Zhang, H. Su, and J. Zhu, "A comprehensive survey of continual learning: theory, method and application," *IEEE TPAMI*, 2024.

[36] S. Tian, L. Li, W. Li, H. Ran, X. Ning, and P. Tiwari, "A survey on few-shot class-incremental learning," *Neural Networks*, vol. 169, pp. 307–324, 2024.

[37] Z. Ji, Z. Hou, X. Liu, Y. Pang, and X. Li, "Memorizing complementation network for few-shot class-incremental learning," *IEEE TIP*, vol. 32, pp. 937–948, 2023.

[38] L. Wang, X. Yang, H. Tan, X. Bai, and F. Zhou, "Few-shot class-incremental sar target recognition based on hierarchical embedding and incremental evolutionary network," *IEEE TGRS*, vol. 61, pp. 1–11, 2023.

[39] Y. Song, T. Wang, P. Cai, S. K. Mondal, and J. P. Sahoo, "A comprehensive survey of few-shot learning: Evolution, applications, challenges, and opportunities," *ACM Computing Surveys*, 2023.

[40] L. Bottou and O. Bousquet, "The tradeoffs of large scale learning," *NeurIPS*, vol. 20, 2007.

[41] L. Bottou, F. E. Curtis, and J. Nocedal, "Optimization methods for large-scale machine learning," *SIAM Review*, vol. 60, no. 2, pp. 223–311, 2018.

[42] L. Yu, B. Twardowski, X. Liu, L. Herranz, K. Wang, Y. Cheng, S. Jui, and J. v. d. Weijer, "Semantic drift compensation for class-incremental learning," in *CVPR*, 2020, pp. 6982–6991.

[43] S.-A. Rebuffi, A. Kolesnikov, G. Sperl, and C. H. Lampert, "icarl: Incremental classifier and representation learning," in *CVPR*, 2017, pp. 2001–2010.

[44] J. Zhu, G. Yao, W. Zhou, G. Zhang, W. Ping, and W. Zhang, "Feature distribution distillation-based few shot class incremental learning," in *PRAI*, 2022, pp. 108–113.

[45] Y. Cui, W. Deng, X. Xu, Z. Liu, Z. Liu, M. Pietikäinen, and L. Liu, "Uncertainty-guided semi-supervised few-shot class-incremental learning with knowledge distillation," *IEEE TMM*, 2022.

[46] Y. Cui, W. Deng, H. Chen, and L. Liu, "Uncertainty-aware distillation for semi-supervised few-shot class-incremental learning," *IEEE TNNLS*, 2023.

[47] C. Peng, K. Zhao, T. Wang, M. Li, and B. C. Lovell, "Few-shot class-incremental learning from an open-set perspective," in *ECCV*, 2022, pp. 382–397.

[48] A. Mallya and S. Lazebnik, "Packnet: Adding multiple tasks to a single network by iterative pruning," in *CVPR*, 2018, pp. 7765–7773.

[49] D. Maltoni and V. Lomonaco, "Continuous learning in single-incremental-task scenarios," *Neural Networks*, vol. 116, pp. 56–73, 2019.

[50] Y. Hu, A. Chapman, G. Wen, and D. W. Hall, "What can knowledge bring to machine learning?—a survey of low-shot learning for structured data," *ACM TIST*, vol. 13, no. 3, pp. 1–45, 2022.

[51] W.-Y. Chen, Y.-C. Liu, Z. Kira, Y.-C. F. Wang, and J.-B. Huang, "A closer look at few-shot classification," in *ICLR*, 2019.

[52] H.-J. Ye, D.-C. Zhan, Y. Jiang, and Z.-H. Zhou, "Heterogeneous few-shot model rectification with semantic mapping," *IEEE TPAMI*, vol. 43, no. 11, pp. 3878–3891, 2020.

[53] I. J. Goodfellow, M. Mirza, D. Xiao, A. Courville, and Y. Bengio, "An empirical investigation of catastrophic forgetting in gradient-based neural networks," *arXiv preprint arXiv:1312.6211*, 2013.

[54] W.-L. Chao, S. Changpinyo, B. Gong, and F. Sha, "An empirical study and analysis of generalized zero-shot learning for object recognition in the wild," in *ECCV*, 2016, pp. 52–68.

[55] H. Qi, M. Brown, and D. G. Lowe, "Low-shot learning with imprinted weights," in *CVPR*, 2018, pp. 5822–5830.

[56] S. W. Yoon, D.-Y. Kim, J. Seo, and J. Moon, "Xtarnet: Learning to extract task-adaptive representation for incremental few-shot learning," in *ICML*, 2020, pp. 10 852–10 860.

[57] O. Vinyals, C. Blundell, T. Lillicrap, D. Wierstra *et al.*, "Matching networks for one shot learning," *NeurIPS*, vol. 29, 2016.

[58] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein *et al.*, "Imagenet large scale visual recognition challenge," *IJCV*, vol. 115, pp. 211–252, 2015.

[59] A. Krizhevsky, G. Hinton *et al.*, "Learning multiple layers of features from tiny images," 2009.

[60] C. Wah, S. Branson, P. Welinder, P. Perona, and S. Belongie, "The caltech-ucsd birds-200-2011 dataset," 2011.

[61] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft coco: Common objects in context," in *ECCV*, 2014, pp. 740–755.

[62] J.-M. Perez-Rua, X. Zhu, T. M. Hospedales, and T. Xiang, "Incremental few-shot object detection," in *CVPR*, 2020, pp. 13 846–13 855.

[63] M. Everingham, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman, "The pascal visual object classes (voc) challenge," *IJCV*, vol. 88, pp. 303–308, 2009.

[64] Z. Pan, X. Yu, M. Zhang, and Y. Gao, "Ssfe-net: Self-supervised feature enhancement for ultra-fine-grained few-shot class incremental learning," in *WACV*, 2023, pp. 6275–6284.

[65] A. R. Shankarampeta and K. Yamauchi, "Few-shot class incremental learning with generative feature replay." in *ICPRAM*, 2021, pp. 259–267.

[66] M. Arjovsky, S. Chintala, and L. Bottou, "Wasserstein generative adversarial networks," in *ICML*, 2017, pp. 214–223.

[67] C. Finn, P. Abbeel, and S. Levine, "Model-agnostic meta-learning for fast adaptation of deep networks," in *ICML*, 2017, pp. 1126–1135.

[68] A. Agarwal, B. Banerjee, F. Cuzzolin, and S. Chaudhuri, "Semantics-driven generative replay for few-shot class incremental learning," in *ACM MM*, 2022, pp. 5246–5254.

[69] K. Zhu, Y. Cao, W. Zhai, J. Cheng, and Z.-J. Zha, "Self-promoted prototype refinement for few-shot class-incremental learning," in *CVPR*, 2021, pp. 6801–6810.

[70] T. Martinetz, "Competitive hebbian learning rule forms perfectly topology preserving maps," in *ICANN*, 1993, pp. 427–434.

[71] B. Yang, M. Lin, B. Liu, M. Fu, C. Liu, R. Ji, and Q. Ye, "Learnable expansion-and-compression network for few-shot class-incremental learning," *arXiv preprint arXiv:2104.02281*, 2021.

[72] T. Ahmad, A. R. Dhamija, S. Cruz, R. Rabinowitz, C. Li, M. Jafarzadeh, and T. E. Boult, "Few-shot class incremental learning leveraging self-supervised features," in *CVPR*, 2022, pp. 3900–3910.

[73] L. Zhao, J. Lu, Y. Xu, Z. Cheng, D. Guo, Y. Niu, and X. Fang, "Few-shot class-incremental learning via class-aware bilateral distillation," in *CVPR*, 2023, pp. 11 838–11 847.

[74] Z. Chi, L. Gu, H. Liu, Y. Wang, Y. Yu, and J. Tang, "Metafscil: a meta-learning approach for few-shot class incremental learning," in *CVPR*, 2022, pp. 14 166–14 175.

[75] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *NeurIPS*, vol. 30, 2017.

[76] Y. Bengio, A. Courville, and P. Vincent, "Representation learning: A review and new perspectives," *IEEE TPAMI*, vol. 35, no. 8, pp. 1798–1828, 2013.

[77] M. Kaya and H. Ş. Bilge, "Deep metric learning: A survey," *Symmetry*, vol. 11, no. 9, p. 1066, 2019.

[78] X. Li, X. Yang, Z. Ma, and J.-H. Xue, "Deep metric learning for few-shot image classification: A review of recent developments," *Pattern Recognition*, p. 109381, 2023.

[79] A. Cheraghian, S. Rahman, S. Ramasinghe, P. Fang, C. Simon, L. Petersson, and M. Harandi, "Synthesized feature based few-shot class-incremental learning on a mixture of subspaces," in *ICCV*, 2021, pp. 8661–8670.

[80] D.-Y. Kim, D.-J. Han, J. Seo, and J. Moon, "Warping the space: Weight space rotation for class-incremental few-shot learning," in *ICLR*, 2023.

[81] P. Khorramshahi, N. Peri, J.-c. Chen, and R. Chellappa, "The devil is in the details: Self-supervised attention for vehicle re-identification," in *ECCV*, 2020, pp. 369–386.

[82] Y. Yang, H. Yuan, X. Li, Z. Lin, P. Torr, and D. Tao, "Neural collapse inspired feature-classifier alignment for few-shot class-incremental learning," in *ICLR*, 2023.

[83] V. Papyan, X. Han, and D. L. Donoho, "Prevalence of neural collapse during the terminal phase of deep learning training," *PNAS*, vol. 117, no. 40, pp. 24 652–24 663, 2020.

[84] N. Sankaran, "Feature fusion for deep representations," Ph.D. dissertation, State University of New York at Buffalo, 2021.

[85] T. Ahmad, A. R. Dhamija, M. Jafarzadeh, S. Cruz, R. Rabinowitz, C. Li, and T. E. Boult, "Variable few shot class incremental and open world learning," in *CVPR Workshops*, 2022, pp. 3688–3699.
[86] J. Kalla and S. Biswas, "S3c: Self-supervised stochastic classifiers for few-shot class-incremental learning," in *ECCV*, 2022, pp. 432–448.
[87] A. Kuznetsova, H. Rom, N. Alldrin, J. Uijlings, I. Krasin, J. Pont-Tuset, S. Kamali, S. Popov, M. Malloci, A. Kolesnikov *et al.*, "The open images dataset v4: Unified image classification, object detection, and visual relationship detection at scale," *IJCV*, vol. 128, no. 7, pp. 1956–1981, 2020.
[88] R. M. Neal, *Bayesian learning for neural networks*. Springer Science & Business Media, 2012, vol. 118.
[89] H. Lee, S. J. Hwang, and J. Shin, "Self-supervised label augmentation via input transformations," in *ICML*, 2020, pp. 5714–5724.
[90] G. Yao, J. Zhu, W. Zhou, and J. Li, "Few-shot class-incremental learning based on representation enhancement," *Journal of Electronic Imaging*, vol. 31, no. 4, p. 043027, 2022.
[91] J. Gou, B. Yu, S. J. Maybank, and D. Tao, "Knowledge distillation: A survey," *IJCV*, vol. 129, pp. 1789–1819, 2021.
[92] S. Dong, X. Hong, X. Tao, X. Chang, X. Wei, and Y. Gong, "Few-shot class-incremental learning via relation knowledge distillation," in *AAAI*, vol. 35, no. 2, 2021, pp. 1255–1263.
[93] Y. Cui, W. Xiong, M. Tavakolian, and L. Liu, "Semi-supervised few-shot class-incremental learning," in *ICIP*, 2021, pp. 1239–1243.
[94] T. Hospedales, A. Antoniou, P. Micaelli, and A. Storkey, "Meta-learning in neural networks: A survey," *IEEE TPAMI*, vol. 44, no. 9, pp. 5149–5169, 2021.
[95] G. Zheng and A. Zhang, "Few-shot class-incremental learning with meta-learned class structures," in *ICDM Workshops*, 2021, pp. 421–430.
[96] Z. Song, Y. Zhao, Y. Shi, P. Peng, L. Yuan, and Y. Tian, "Learning with fantasy: Semantic-aware virtual contrastive constraint for few-shot class-incremental learning," in *CVPR*, 2023, pp. 24 183–24 192.
[97] Q.-W. Wang, D.-W. Zhou, Y.-K. Zhang, D.-C. Zhan, and H.-J. Ye, "Few-shot class-incremental learning via training-free prototype calibration," *NeurIPS*, vol. 36, 2023.
[98] D. A. Ganea, B. Boom, and R. Poppe, "Incremental few-shot instance segmentation," in *CVPR*, 2021, pp. 1185–1194.
[99] L. Yin, J. M. Perez-Rua, and K. J. Liang, "Sylph: A hypernetwork framework for incremental few-shot object detection," in *CVPR*, 2022, pp. 9035–9045.
[100] N. Dong, Y. Zhang, M. Ding, and G. H. Lee, "Incremental-detr: Incremental few-shot object detection via self-supervised learning," in *AAAI*, 2023.
[101] M. Cheng, H. Wang, and Y. Long, "Meta-learning-based incremental few-shot object detection," *IEEE TCSVT*, vol. 32, no. 4, pp. 2158–2169, 2021.
[102] H. Feng, L. Zhang, X. Yang, and Z. Liu, "Incremental few-shot object detection via knowledge transfer," *Pattern Recognition Letters*, vol. 156, pp. 67–73, 2022.
[103] X. Zhou, D. Wang, and P. Krähenbühl, "Objects as points," *arXiv preprint arXiv:1904.07850*, 2019.
[104] Z. Tian, C. Shen, H. Chen, and T. He, "Fcos: Fully convolutional one-stage object detection," in *ICCV*, 2019, pp. 9627–9636.
[105] S. Farquhar and Y. Gal, "Towards robust evaluations of continual learning," *arXiv preprint arXiv:1805.09733*, 2018.
[106] S. Zhang, C. Chi, Y. Yao, Z. Lei, and S. Z. Li, "Bridging the gap between anchor-based and anchor-free detection via adaptive training sample selection," in *CVPR*, 2020, pp. 9759–9768.
[107] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko, "End-to-end object detection with transformers," in *ECCV*, 2020, pp. 213–229.
[108] J. R. Uijlings, K. E. Van De Sande, T. Gevers, and A. W. Smeulders, "Selective search for object recognition," *IJCV*, vol. 104, pp. 154–171, 2013.
[109] K. Nguyen and S. Todorovic, "ifs-rcnn: An incremental few-shot instance segmenter," in *CVPR*, 2022, pp. 7010–7019.
[110] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask r-cnn," in *ICCV*, 2017, pp. 2961–2969.
[111] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," *NeurIPS*, vol. 28, 2015.
[112] X. Wang, T. E. Huang, T. Darrell, J. E. Gonzalez, and F. Yu, "Frustratingly simple few-shot object detection," in *ICML*, 2020, pp. 9919–9928.
[113] D. J. Spiegelhalter and S. L. Lauritzen, "Sequential updating of conditional probabilities on directed graphical structures," *Networks*, vol. 20, no. 5, pp. 579–605, 1990.
[114] S. B. Klein, "What memory is," *Wiley Interdisciplinary Reviews: Cognitive Science*, vol. 6, no. 1, pp. 1–38, 2015.
[115] D. Kudithipudi, M. Aguilar-Simon, J. Babb, M. Bazhenov, D. Blackiston, J. Bongard, A. P. Brna, S. Chakravarthi Raja, N. Cheney, J. Clune *et al.*, "Biological underpinnings for lifelong learning machines," *Nature Machine Intelligence*, vol. 4, no. 3, pp. 196–210, 2022.
[116] M. D'Alessandro, A. Alonso, E. Calabrés, and M. Galar, "Multimodal parameter-efficient few-shot class incremental learning," in *ICCV Workshops*, 2023, pp. 3393–3403.
[117] J. Zhang, P. Zhao, Y. Zhao, C. Li, and D. Hu, "Few-shot class-incremental learning for retinal disease recognition," *IEEE JBHI*, 2024.

Jinghua Zhang received the B.E. degree from the Hefei University, China, in 2018, and the M.E. degree from the Northeastern University, China, in 2021. He is currently pursuing the Ph.D. degree in control science and engineering with the National University of Defense Technology, and is also with the Center for Machine Vision and Signal Analysis, University of Oulu. His research interests include computer vision and deep learning.

Li Liu received her Ph.D. from National University of Defense Technology, China, in 2012 and is now a Full Professor there. She has visited the University of Waterloo, Chinese University of Hong Kong, and University of Oulu. She has co-chaired workshops for CVPR and ICCV, served as lead guest editor for IEEE TPAMI and IJCV, and is an Associate Editor for IEEE TCSVT and Pattern Recognition. Her research in computer vision, pattern recognition, and machine learning has garnered over 16,000 citations.

Olli Silvén received the M.Sc. and Ph.D. degrees in electrical and computer engineering from the University of Oulu, Finland, in 1982 and 1988, respectively. Since 1996, he has been a professor of signal processing engineering with the University of Oulu. He has contributed to the development of numerous solutions from real-time 3-D imaging in reverse vending machines to IP blocks for mobile video coding. His research focuses on ultra-energy-efficient-embedded signal processing and machine vision system design.

Matti Pietikäinen , who earned his Ph.D. degree from the University of Oulu and serves as an emeritus professor at its Center for Machine Vision and Signal Analysis, is notable for his contributions to Local Binary Pattern (LBP). His work has attracted about 91,600 citations on Google Scholar. He has been honored with the Koenderink Prize in 2014 and the IAPR King-Sun Fu Prize in 2018 for his machine vision contributions. He is also an IEEE fellow, recognized for his work in machine vision.

Dewen Hu received his B.S. and M.S. degrees from Xi'an Jiaotong University, China, in 1983 and 1986, and his Ph.D. from the National University of Defense Technology, China, in 1999. He is currently a Professor at the same university. He has visited the University of Sheffield, U.K., in 1995-1996. He has over 400 publications in journals and conferences like PNAS, IEEE TPAMI, and IJCV, focusing on pattern recognition and cognitive science, and serves as an associate editor for Neural Networks and IEEE TSMCS.