Cross-Domain Few-Shot Incremental Learning for Point-Cloud Recognition

Yuwen Tan and Xiang Xiang*
Key Lab of Image Processing and Intelligent Control, Ministry of Education
School of Artificial Intelligence and Automation
Huazhong University of Science and Technology, Wuhan 430074, China

Abstract

Sensing 3D objects is critical when 2D object recognition is not accessible. A robot pre-trained on a large point-cloud dataset will encounter unseen classes of 3D objects after deploying it. Therefore, the robot should be able to learn continuously in real-world scenarios. Few-shot class-incremental learning (FSCIL) requires the model to learn from few-shot new examples continually and not forget past classes. However, there is an implicit but strong assumption in the FSCIL that the distribution of the base and incremental classes is the same. In this paper, we focus on cross-domain FSCIL for point-cloud recognition. We decompose the catastrophic forgetting into base class forgetting and incremental class forgetting and alleviate them separately. We utilize the base model to discriminate base samples and new samples by treating base samples as in-distribution samples, and new objects as out-of-distribution samples. We retain the base model to avoid catastrophic forgetting of base classes and train an extra domain-specific module for all new samples to adapt to new classes. At inference, we first discriminate whether the sample belongs to the base class or the new class. Once classified at the model level, test samples are then passed to the corresponding model for class-level classification. To better mitigate the forgetting of new classes, we adopt the soft label and hard label replay together. Extensive experiments on synthetic-to-real incremental 3D datasets show that our proposed method can balance the performance between the base and new objects and outperforms the previous state-of-the-art methods.

1. Introduction

3D object recognition has made significant progress in various applications including robotics, shape analysis, and autonomous driving. There exists a lot of work for point-cloud recognition especially point-based [16, 17, 27, 29, 35, 39] methods have been extensively researched. However, most of the work focuses on the static classification ability of the model where the number of categories is pre-defined and fixed. This static setup impedes the model application in the real world, where classes of 3D objects arrive in continuous streams. As shown in Fig. 1, a home-assistance robot pre-trained with abundant instances (*e.g.*, synthetic 3D objects) can only grasp the pre-defined objects in a clean environment, it needs to continually recognize new categories of objects which may have occlusions, cluttered backgrounds, and poor-quality point-cloud. Such a need also exists in vision-driven autonomous driving systems equipped with LiDARs. The pre-trained recognition model may encounter new classes in a more complex environment after deploying it. A self-driving car needs to continually recognize new classes as it runs like humans. However, new classes often consist of few-shot samples and there is a domain shift with the base classes. Such a realistic and challenging setting has been proposed in the paper [8]. The incremental sessions contain few-shot samples from the real world, while the base session contains many synthetic 3D objects for training. The proposed setting mimics the few-shot and domain shifts that commonly exist in the continuous learning process for real-world applications.

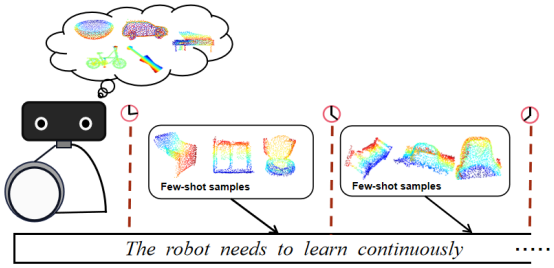

Figure 1. A simple diagram of FSCIL for cross-domain point-cloud recognition. Base classes contain rich synthetic 3D objects, while incremental sessions only have few-shot real-world scanned 3D objects. The model needs to learn new classes continuously and preserve existing knowledge.

Most of the FSCIL methods proposed in the 2D domain [24, 41, 44, 50] propose to freeze the feature extractor after the base session to alleviate the forgetting. These

*Corresponding author (E-mail: xex@hust.edu.cn).

prototype-based methods in FSCIL are effective at mitigating forgetting and achieving good performance of new classes in the 2D domain. However, due to the low accuracy of new classes in our setting, these methods fail to show their superiority in the cross-domain FSCIL for 3D object recognition. Unlike 2D image datasets, 3D point cloud datasets [43] are generally smaller and do not have strong pre-trained models trained on large-scale datasets. The low-quality embedding of point clouds which have high intra-class discrepancy and low inter-class discriminability would reduce the effectiveness of prototype-based methods. Besides that, we propose that the poor performance of new classes also comes from the fact that these methods only aim to learn an expandable and compact feature space in the base session for better generalization. However, since the distribution of samples differs between the base and incremental phases, expanding the feature space only using the base samples is not sufficient for the new classes. Numerous methods have been proposed to tackle the domain shift problem in point-cloud tasks including 3D object recognition and semantic segmentation [1, 28, 42], but they need a large number of unlabeled target samples during the training process to align the feature space. Due to the inaccessibility of new class samples in the base session, it is challenging to align the feature space. Additionally, few-shot new samples that appear asynchronously make it quite difficult to achieve alignment during incremental learning stages.

As the prototype-based FSCIL method has its limitations, we propose a new perspective to address the challenges in cross-domain FSCIL for 3D object recognition. We do not focus on training a well-aligned and expandable feature space in the base session and then freezing it. Instead, we fine-tune the network to better adapt to the new samples in incremental learning sessions. Since the base model is trained with abundant samples, it possesses higher feature generalizability and discriminability towards unknown classes compared to the incremental learning model. As shallow layers are inclined to learn generalized representations, we only fine-tune deep layers to encode the new class information and fix the shallow layers to keep the generalized representation ability from the base model. The modification of the deep layers inevitably causes the model to classify base samples into new classes and thus lead to severe forgetting. To further avoid the confusion between the base and new classes, we adopt a two-branch structure to first classify the sample to base or new classes and then pass it to the corresponding model for class-level classification. We store the task-specific layers of the base model instead of retaining abundant base samples and this operation can highly address the forgetting of base samples.

In this paper, we predict the task ID from the out-of-distribution detection (OOD) view and only divide the tasks into the base and new sessions. We regard real samples from the incremental sessions as OOD data compared to base samples. The learning-based method [38] has been proposed to discriminate between the old and new samples. However, such a learning-based method performs poorly in FSCIL due to the limited number of new samples in the training stage. Instead, we use the maximum logit of the base model as the score to detect base and new samples. Since cosine similarity is bounded and represents more discrimination, we use the cosine classifier [21] in the base model training. Decoupling the base and incremental training stages can alleviate base class forgetting. However, new classes with few-shot samples also face severe forgetting as they share one domain-specific module. In this paper, we adopt a dual replay that not only retains the samples but also the output logits of the past model. Through the hard label replay optimized by cross-entropy loss and soft label replay optimized by logits matching, the performance of past new classes can be better retained.

The contributions of this paper are as follows:

1) Due to the significant differences between the base training stage and few-shot incremental learning stages in FSCIL, we propose to decompose the learning sessions into the base and new stages and predict the task ID from the out-of-distribution detection perspective. By predicting the task ID, we can perform fine-tuning on the task-specific layers to better adapt to new classes with a different distribution without sacrificing the performance of the base stage;

2) To alleviate the forgetting of new incremental classes, we propose a dual replay method that not only retains samples with one-hot labels but also the output logits of the past model. Through the hard label and soft label replay, the forgetting of incremental new classes can be highly alleviated;

3) Significant performance improvements on three 3D cross-domain few-shot incremental benchmarks have demonstrated our simple but effective method can balance the base classes and new classes.

2. Related Work

2.1. Point-cloud object recognition

Many deep learning-based methods have been proposed to recognize point cloud objects. PointNet [26] was the first work to process raw points which combined multi-layer perceptron and symmetric function to learn and aggregate point features. However, PointNet [26] ignored the local spatial relationships between the points. Several methods have been proposed to extract local and global features simultaneously. PointNet++ [27] extracted local features via the hierarchical structures. Some work [17, 39] proposed new convolution operations on 3D points. Additionally, several networks regarded 3D points as the vertex of the graph and encoded local information through neighbor points [16, 35]. DGCNN [35] proposed an EdgeConv module computed on

the feature space and contacted the features of the center point and its neighbor points. Further, transformer-based methods have [48] been proposed. Zhang *et al.* [46] applied CLIP to point cloud recognition, which migrates 2D pre-trained knowledge to the 3D domain.

2.2. Few-shot class-incremental learning

Tao *et al.* [31] first proposed few-shot class-incremental learning, and they proposed a neural gas network. Further, vector quantization [6] in the embedding space, word vectors distillation [7], and parameter selecting [23] methods have been proposed. Most of the FSCIL methods [25, 41, 44, 50] froze the feature extractor and only trained the linear classifier or used prototypes for classification. CEC [44] focused on classifier adaptation and designed an extra graph model. FACT [50] and ALICE [25] were both concentrated on learning an extendable and compact feature space in the base session. FACT [50] pushed the samples in the same class to be closer and used virtual prototypes in the base training. ALICE [25] used cosine similarity and margin to learn a better feature space for lateral learning. Constrained FSCIL [14] contained a trainable fully connected layer, a rewritable memory, and provided three update modes. Liu *et al.* [19] proposed a data-free replay scheme that synthesized data through the generator without access to past samples.

2.3. Class-incremental learning on point cloud

Dong *et al.* [10] first proposed the incremental setting for 3D object recognition (I3DOL). They first constructed geometric centroids, used an attention mechanism, and designed a score fairness compensation to avoid forgetting. Liu *et al.* [20] proposed a new model named L3DOC, which used a layer-wise point-knowledge factorization module to capture the point knowledge, thus reducing catastrophic forgetting. A realistic and challenging setting was proposed in [8] for 3D point-cloud recognition. They used Singular Value Decomposition to choose a set of basis vectors and enhanced the ability of the model to adapt to real-world data. Though it can maintain the base performance well, the accuracy of new classes cannot meet the practical need. Cen *et al.* [4] extended the open-world problem to the semantic segmentation for LIDAR point clouds. Zhao *et al.* [49] proposed an effective static-dynamic co-teaching method which can incrementally detect novel classes without revisiting any previous training samples.

2.4. Out-of-distribution detection

Out-of-distribution (OOD) detection aims to detect test samples drawn from a different distribution from training samples and maintain the classification performance of in-distribution (ID) data. For FSCIL in 3D object recognition, new samples during the incremental sessions can be regarded as OOD data compared to base class examples. Several methods [9, 11, 12, 33, 36] have been proposed for OOD detection, and those methods can be divided into three categories: discriminative methods [12, 32, 33, 36, 47], generative methods [11, 45], and classifier-based [9, 13] methods. Classifier-based methods need extra OOD data to train a binary classifier, and generative-based methods generate pseudo-OOD data. All the discriminative methods are applied to the classifier after training and it does not change the original training objective.

3. Methodology

3.1. Problem definition

Let us formalize the definition of cross-domain FSCIL for point-cloud recognition. The base classes set and new classes set are represented by $\mathcal{C}_{base}$ and $\mathcal{C}_{new}$, respectively. Base classes have sufficient instances N_0 for training while new classes come in data streams denoted as $L_t = \{x_i^t, y_i^t\}_{i=1}^{N_t}$, $t \in \{1, 2, ..., T\}$ with $N_t (N_t \ll N_0)$ samples. We denote each data stream as a session and the class set in t-th session is denoted as $\mathcal{C}(t)$. Note that for all $t_1 \neq t_2$, $\mathcal{C}(t_1) \cap \mathcal{C}(t_2) = \varnothing$. The base classes set is denoted as $\mathcal{C}(0)$ and the new classes set in the t -th session is denoted as $\mathcal{C}_{new} = \sum_{i=1}^{t} \mathcal{C}(i)(t \geq 1)$. In the t-th training session, the model can only access the classes set $\mathcal{C}(t)$ and a limited buffer $\mathcal{B}(t)$. In the t-th testing session, the model needs to evaluate all the seen classes $\{\mathcal{C}(0), \mathcal{C}(1), ...\mathcal{C}(t)\}$. We decompose the model into three modules: feature extractor $f(\cdot;\theta)$ with the parameter θ, projection layers $g(\cdot;\varphi)$ with parameter set φ, and the linear classification layer with parameter set ϕ. The feature extractor defines the high-dimension feature space $\mathcal{F} \subseteq \mathbb{R}^h$. The projection layer maps the high dimension feature $\mathcal{F} \subseteq \mathbb{R}^h$ into the lower dimension feature $\mathcal{F} \subseteq \mathbb{R}^d$. The classification layer with the parameter ϕ outputs the probability of all classes. The whole parameters of the model in the t-th session can be denoted as $\mathcal{M}_t = \{\theta_t, \varphi_t, \phi_t\}$. We regard the parameter set $\{\varphi_t, \phi_t\}$ as the task-specific parameters and θ_t as the generalizable parameters that are fixed during training.

3.2. Overview

The whole pipeline of our proposed method is shown in Fig. 2. We decouple the whole learning process into the base classes training stage and incremental classes (new classes) training stage and train their classifier respectively. Thus, the goal of FSCIL is to learn the classification probability $p(C_{k,i_0}|x)$ and the probability can be decoupled into two probabilities, in-task probability $p(C_{k,i_0}|x \in C_k)$ and task-prediction probability $p(C_k|x)$ the same as [15]. The classification probability can be defined as

$$p(C_{k,i_0}|x) = \sum_{k=0,1} p(C_k|x)p(C_{k,i_0}|x \in C_k) \quad (1)$$

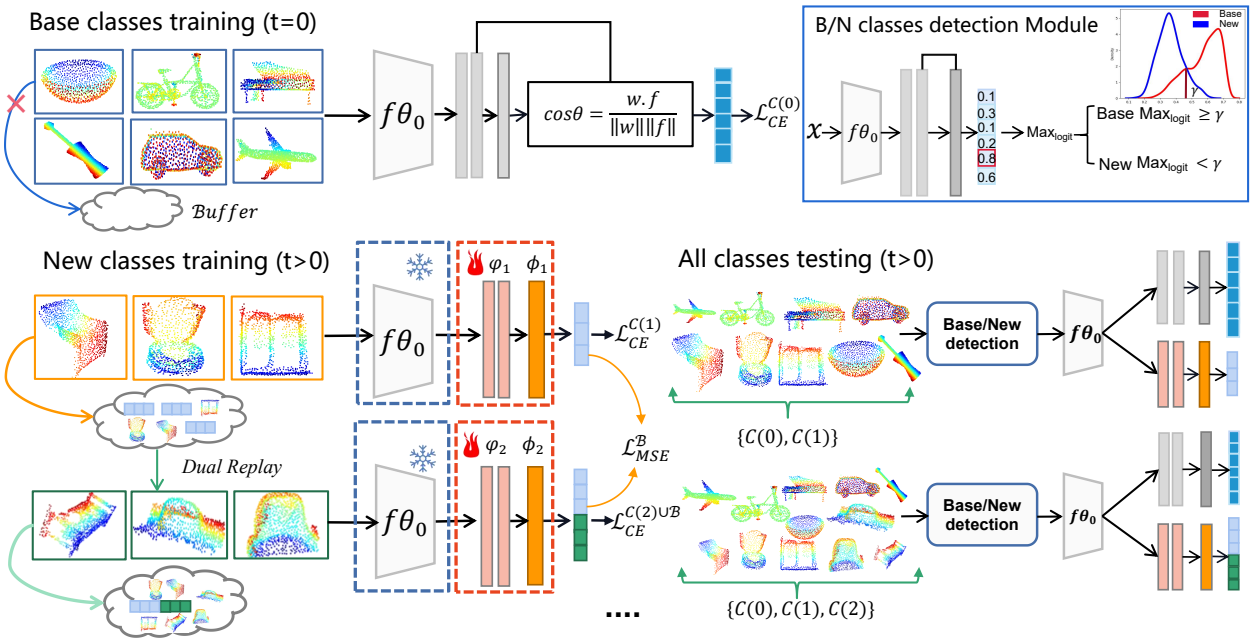

Figure 2. The proposed framework for incremental point-cloud recognition. The model in the base session is trained with a cosine classifier on a large synthetic dataset. In the incremental sessions, part of the parameters of the model is modified with few-shot real-world samples. At inference, samples are first identified as the base or new classes and then passed to different branches of the model for prediction.

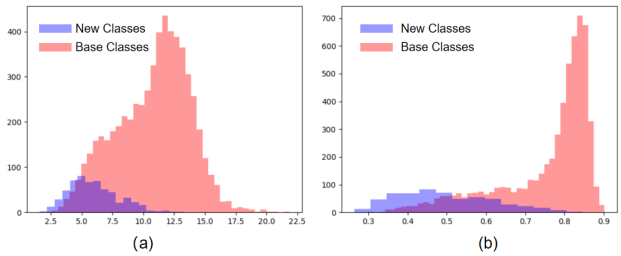

Figure 3. Histograms of the maximum of logits over all classes. The samples of base classes are in red and the samples of new classes are in blue. (a) Standard cross-entropy. (b) Cross-entropy with the scaled cosine classifier.

where $\mathcal{C}_0 = \mathcal{C}_{base}$, $\mathcal{C}_1 = \mathcal{C}_{new}$, i_0 stands for a specific class in the base classes set or new classes set. We use the base model as the OOD detector to predict whether test samples belong to the base classes training stage or the new classes training stage. As deep layers learn task-specific information, we retain the deep layers of the base model and add an extra block for new class training. To further alleviate the new classes forgetting, we adopt soft label and hard label replay. In the testing stage, we first discriminate the base/new classes and then pass them to the corresponding classifier. Effectively discriminating between the base and new samples is essential for obtaining high performance.

3.3. Base-classes training with a cosine classifier

As base classes have abundant training instances, the model trained in the base session is equipped with strong feature representation ability, and the confidence in the predicted results should also be relatively high. Thus, we utilize the base model as an OOD sample detector to discriminate base samples and new samples during the testing stages. The base model needs to maintain the classification ability of base class samples (ID data) while being able to detect new samples (OOD data) without access to new samples. Since the softmax function may smooth the confidence of the model predictions, we use the maximum logit instead of Maximum Softmax Probability (MSP) as the score to discriminate the ID/OOD samples.

Most methods use a linear classifier in the base model and optimize the standard cross-entropy loss, which is formulated as

$$\mathcal{L}^{ce} = -log\frac{e^{w_c^T f + b_c}}{\sum_{i=1}^{C(0)} e^{w_i^T f + b_i}}, \tag{2}$$

where $W = [w_1, w_2, .., w_{C(0)}]$ is the weight of the classifier layer and $b = [b_1, b_2, .., b_{C(0)}]$ stands for the bias of the classifier layer. We suppose new classes have a low logit response, and base classes should have a high logit response. However, as shown in Fig. 3, when training the base model with standard cross-entropy loss, the maximum logit exhibits less distinguishable information between base classes (ID data) and new classes (OOD data). There is a large overlapping area between maximum logits, and it is hard to classify the base and new classes through the logits. Additionally, since the logit outputted by the linear classi-

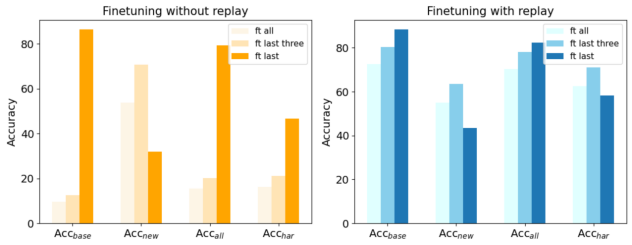

Figure 4. Results of different fine-tuning methods w and w/o replay in the first incremental session. Acc_{all} stands for the accuracy for all classes, and Acc_{har} stands for the harmonic accuracy of base and new classes.

fier is unbounded, it is hard to define a logit threshold which is crucial for classifying base samples and new samples.

As shown in Fig. 3, when we replace the standard classifier with a cosine classifier, the maximum logits are bounded in [-1,1] and are more discriminative between the base and new classes. Thus, we train the base model using the softmax of scaled cosine similarities instead of the ordinary softmax of logits. The loss is formulated as

$$\mathcal{L}^b = -log\frac{e^{\tau s_c}}{\sum_{i=1}^{C(0)} e^{\tau s_i}}, \tag{3}$$

where $s_c = \frac{w_c^T f}{||w_c||.||f||}$ stands for the cosine similarity of the feature f and the weight w of the classifier layer, and τ is a scale factor.

3.4. Incremental-classes training with dual replay

Some methods freeze the whole backbone and only train the new classifier layer to avoid forgetting base classes. However, when base classes and incremental classes come from different domains, only fine-tuning the linear classifier layer cannot guarantee the good performance of the new classes. As shown in Fig. 4, when only training the classifier layer, the accuracy of new classes is much lower. We initialize the $f(\cdot;\theta_t)$ using the base model $f(\cdot;\theta_0)$ and then freeze it to avoid over-fitting in the incremental training session. As shallow layers tend to learn generalized representations, we only fine-tune deep layers.

Moreover, we also adopt the cosine classifier in the incremental session as it can solve the norm and bias [41] problem usually encountered in FSCIL. To further alleviate the forgetting of past new samples, we use a small buffer that stores only one instance per past class. As soft labels contain more information than one-hot labels, we also store the logits of the old model for past samples. Although soft labels contain more information than hard labels, we also use hard labels as auxiliary information to avoid the misclassification of the soft labels produced by the model. It is an effective way to avoid the incremental model mimicking incorrect information from the previous model. Through the hard label and soft label replay, the model can preserve more information about past classes.

We train the classification layer and projection layer with the normalized cross-entropy loss and logit match loss. We used mean squared error (MSE) instead of KL divergence the same as [19] because logit matching has better generalization ability. The total loss is defined as

$$\mathcal{L}^t = -\sum_{n=1}^{N_t+N_{\mathcal{B}}} log\frac{e^{\tau s_c^n}}{\sum_{i=1}^{C_{new}} e^{\tau s_i^n}} + \sum_{n=1}^{N_{\mathcal{B}}} ||o^t - o^{(t-1)}||_2, \tag{4}$$

where N_t is the number of new training samples in the session t , $N_{\mathcal{B}}$ stands for number of the samples in the buffer $\mathcal{B}$, o stands for the output logits and C_{new} stands for the all incremental classes till session t. For each session, we randomly select *one example of each class* and put them into the buffer $\mathcal{B}_t$ which is defined as

$$\mathcal{B}_t = \sum_{j}^{\mathcal{C}(t)} \{x_j^i, y_j^i, o_j^i\}(i=1) \cup \mathcal{B}_{t-1}, t > 1. \tag{5}$$

The buffer $\mathcal{B}_t$ is empty in the first session as we retain no base sample and only update in the incremental sessions.

3.5. Testing pipelines with base/new detection

A robust base model should have a high response for base class testing samples (i.i.d. with the training samples) and a low response for new class testing samples out of the distribution. It is supposed that the output of the base model has a lower response to new classes and a much higher response to base classes. Thus, we use the output of the base model to detect base classes and new classes. For the input x, we compute the cosine similarity $cos\theta_i = \frac{w_i^T f}{||w_i||.||f||}$ through the base model. We choose the maximum cosine similarity as the binary classification score

$$S(x) = \max(cos\theta_i)_{i=1}^{C(0)}, \tag{6}$$

where x is the 3D point object input and $C(0)$ is the class set of base classes and we set a threshold γ to classify the base and new classes. The detection of base classes and new classes can be regarded as a binary classification problem

$$G(x) = \begin{cases} 0 & S(x) \geq \gamma \\ 1 & S(x) < \gamma \end{cases} \tag{7}$$

We set the label of all base class examples as 0 and all incremental samples as 1. By convention, samples with higher scores $S(x)$ are classified as base classes and vice versa. As we get the predicted task ID, we can decouple the base classes and new classes testing. The classifier output can be formulated as

$$o(x) = \begin{cases} \phi_0^T g(f(x,\theta_0),\varphi_0) & G(x) = 0 \\ \phi_t^T g(f(x,\theta_0),\varphi_t) & G(x) = 1 \end{cases} \tag{8}$$

where $G(x)$ is the binary label (base or new) of the input data x and $o(x)$ is the output logit.

4. Experiments

4.1. Datasets and evaluation

Datasets. We use four 3D object classification datasets including two synthetic datasets (ModelNet [40], ShapeNet [5]) and two real-scanned datasets (CO3D [30] and ScanObjectNN [33]). We construct three datasets the same as paper [8] which contains three cross-domain FSCIL tasks, *e.g.,* ModelNet $\rightarrow$ ScanObjectNN, ShapeNet $\rightarrow$ CO3D and ShapeNet $\rightarrow$ ScanObjectNN. For ModelNet $\rightarrow$ ScanObjectNN (M2S), we choose 26 classes from ModelNet as the base classes and the incremental sessions contain 11 classes from ScanObjectNN. And for the ShapeNet $\rightarrow$ CO3D (S2C), 39 classes from the ShapeNet are chosen as base classes, and 50 classes from CO3D are used as incremental classes. For the ShapeNet $\rightarrow$ ScanObjectNN (S2S), we choose 44 base classes from ShapeNet and use all of the classes of ScanObjectNN as the incremental classes. We use all the training samples from base classes in the base session training and randomly select 5-shot samples from each new class for incremental sessions.

Evaluation Metric. The SOTA [8] method only reports the total accuracy. However, this evaluation cannot reveal the balance between forgetting the old class samples and learning the new class. The majority ($70\% \sim 80\%$) of the test samples come from the base class, and only a tiny fraction of the test samples are new classes. The total accuracy would be high if the model does not learn new classes and only retains the base class classification ability. Thus, we use a more reasonable evaluation metric proposed in the paper [24] to represent the model's ability to balance base classes and new classes. We formulate the harmonic accuracy as

$$\mathcal{A}_h = \frac{2 \times A_b \times A_n}{A_b + A_n}, \tag{9}$$

where A_b is the accuracy of the base classes and A_n stands for the accuracy of new classes. Additionally, we report the performance of the base classes and the new classes in each learning session.

4.2. Implementation details

We use DGCNN [35] as the backbone for all the compared methods. We adopt the farthest 1024 points as the model input, the same as [8] for a fair comparison. We train the base model for 50 epochs using the Adam optimizer and set the batch size as 32. The learning rate is initialized to 0.0001 with decay by a factor of 0.5 in epoch 30. For the training in incremental sessions, the learning rate is 0.001 for all the compared methods, and in our proposed method, the initial learning rate is 0.0005. The epochs in incremental sessions are set as 60. To explore different backbones, we also use PointCLIP [46] as the backbone which shows significant performance improvement in the new classes. We fine-tune the visual encoder and text encoder in the base session and only train the adapter in the incremental session to avoid over-fitting.

4.3. Comparison with SOTA

We compare our methods with several proposed SOTA FSCIL methods for image classification and one method [8] for 3D object recognition. FT is regarded as lower-bound of the proposed setting which fine-tunes the whole network initialized with the previous model without any past examples. In joint training, incremental classes are jointly trained using all samples of the classes. However, joint training can not be regarded as the upper bound as it tends to classify the new classes into base classes which harms the performance of novel classes. LwF [18] is a no-rehearsal-based method that uses distillation loss to regularize the change of parameters. ScaIL [3], IL2M [2], and Micro [8] all store examples of base and new classes. FACT [50] and ALICE [24] store prototypes of past classes. Our method *does not store any base samples and only constructs a small buffer to store past few-shot new samples and logits of the past model*.

ModelNet $\rightarrow$ ScanObjectNN. We evaluate the performance of our proposed method on the M2S dataset to validate its effectiveness. Tab. 1 shows all the compared results, where our method outperforms other methods by a large margin. FACT is the prototype-based method that greatly alleviates the forgetting of base classes. However, the performance of new classes of our method in the last session surpasses FACT by a substantial margin. ALICE also uses the cosine classifier and data augmentation for base session training, but it tends to maintain the performance of base classes and have a lower accuracy of new classes. Micro has the strongest ability to maintain the performance of old classes but has the lowest performance in new classes. The best average harmonic accuracy confirms that our method can better learn new classes while preserving the performance of old ones.

ShapeNet $\rightarrow$ ScanObjectNN. As shown in Tab. 2, our approach outperforms other proposed methods by a large margin. LwF which retains no old examples performs worst due to its severe forgetting of past samples. Since ScaIL and IL2M store limited old samples, the performance drop of base classes is much lower than LwF. Our method outperforms ScaIL and IL2M both in the learning of new classes and the ability to maintain the classification ability of base classes. While the performance drop of base classes is the lowest in FACT, the performance of new classes is the worst. Our method strikes a better balance between learning new classes and maintaining base class performance. Furthermore, our approach outperforms Micro, recent work on FSCIL for 3D object recognition, with a higher average harmonic accuracy of 55.0% vs. 20.0%. When we replace DGCNN with PointCLIP as the backbone, the performance

Method	Session-0	Session-1			Session-2			Session-3			Incre. Avg↑		
	Base	Base	New	Har	Base	New	Har.	Base	New	Har.	Base	New	Har.
Joint	90.8	78.1	18.7	30.2	71.6	22.4	34.1	72.0	24.8	36.9	73.9	22.0	33.7
FT	90.8	20.5	44.9	28.1	0.0	17.3	0.0	0.0	7.8	0.0	6.8	23.3	9.4
LwF [18]	90.8	17.6	14.7	16.0	8.0	11.5	9.4	6.7	3.8	4.8	10.8	10.0	10.1
ScaIL [3]	90.8	51.1	43.1	46.8	47.1	27.6	34.8	41.3	16.4	23.5	46.5	29.0	35.0
IL2M [2]	90.8	50.3	40.0	44.6	49.7	26.5	34.6	48.5	15.4	23.4	49.5	27.3	34.2
FACT [50]	90.5	83.8	12.0	21.0	74.5	4.8	9.0	74.2	2.9	5.6	77.5	6.6	11.9
Micro [8]	87.1	85.0	1.8	3.5	80.2	4.8	9.1	75.0	2.9	5.6	80.1	3.2	6.1
ALICE [24]	88.7	85.0	10.7	19.0	84.2	9.9	17.7	83.2	6.1	11.4	84.1	8.9	16.0
Ours (DGCNN)	90.7	72.8	61.8	66.8	72.8	43.4	54.4	72.8	29.3	41.7	72.8	44.8	54.3
Ours*(PointCLIP)	92.2	71.1	54.2	61.5	71.1	41.8	52.7	71.1	38.9	50.3	71.1	45.0	54.8

Table 1. Comparison with SOTA methods on M2S dataset. 'Base' stands for the accuracy of base classes, and 'New' represents the accuracy of all the incremental classes till t-th the session.

Method	Session-0	Session-1			Session-2			Session-3			Incre. Avg↑		
	Base	Base	New	Har	Base	New	Har.	Base	New	Har.	Base	New	Har.
Joint	87.9	52.6	9.7	16.4	49.1	11.2	18.2	45.0	15.5	23.1	48.9	12.1	19.2
FT	87.9	4.1	23.1	7.0	0.0	16.0	0.0	0.0	5.0	0.0	1.4	14.7	2.3
LwF [18]	87.9	20.3	25.6	22.6	13.3	8.8	10.6	1.1	8.4	1.9	11.6	14.3	11.7
ScaIL [3]	87.9	64.3	28.6	39.6	60.3	16.9	26.4	57.4	9.5	16.3	60.7	18.3	27.4
IL2M [2]	87.9	70.2	18.1	28.8	63.5	16.7	26.5	60.8	13.0	21.4	64.8	15.9	25.6
FACT [50]	82.3	81.6	9.7	17.3	80.0	4.2	8.0	78.2	3.8	7.3	79.9	5.9	10.9
Micro [8]	84.2	75.1	12.6	21.6	70.5	13.6	22.8	68.5	8.8	15.6	71.4	11.7	20.0
ALICE [24]	77.8	71.2	8.4	15.0	69.8	11.4	19.6	69.0	9.8	17.2	70.0	9.9	17.3
Ours (DGCNN)	87.1	76.6	60.1	67.4	76.6	38.2	51.0	76.6	33.4	46.5	76.6	43.9	55.0
Ours*(PointCLIP)	89.6	80.8	62.2	70.3	80.8	44.4	57.3	80.8	33.9	47.8	80.8	46.8	58.5

Table 2. Comparison with SOTA methods on S2S dataset. 'Base' stands for the accuracy of base classes, and 'New' represents the accuracy of all the incremental classes till t-th the session.

of the new class increases due to its excellent performance for few-shot point-cloud recognition.

ShapeNet → CO3D. The ShapeNet→ CO3D dataset has the longest number of tasks and it is the most complicated of all datasets. We report the class-wise average accuracy and harmonic accuracy of each session in Fig. 5. The prototype-based methods have high class-wise average accuracy which reveals it can better maintain the performance of base classes. However, the harmonic accuracy of FACT is much lower than ours in the last session. When freezing the whole network and saving prototypes of old classes can maintain the performance of base classes but performs worse for the new classes. Our method outperforms Micro in both class-wise average accuracy and harmonic accuracy.

4.4. Ablation study

Different base model training strategies. In this section, we compare different training methods on the base session and evaluate their ability to detect base classes and incremental classes. We regard base classes as ID data and all the incremental classes as OOD data. As base classes/new classes detection is a binary classification problem, we use one standard metric: Area Under the ROC Curve (AUROC)

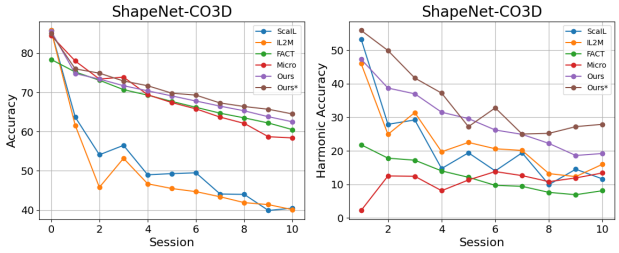

Figure 5. Comparison with SOTA methods on S2C dataset. We report the class-wise average accuracy and harmonic accuracy.

which is widely used in evaluating binary classification performance and also commonly used in OOD detection tasks. For the OOD score, we adopt two simple but effective score functions MSP [12], and MLS [34]. As shown in Tab. 3, using the cosine classifier achieves the best OOD detection performance (the highest AUROC) of all the compared methods, and the incremental performance is the best (the highest average harmonic accuracy).

Different base/new detection methods. We also compare our method with other detection-based methods, including confidence-based [38] and learning-based [38]. The confidence-based method compares the maximum probability of the base model output and the incremental model

Method	MSP [12]	MLS [34]	Task Per.	
	AUROC ↑		Avg(Har.)	In Acc
Cross Entropy	87.4	89.9	42.3	87.6
LogitNorm [36]	86.5	85.9	42.4	87.4
CenterLoss [37]	90.7	92.5	50.0	87.3
Cosine(τ=30)	86.4	**92.6**	**55.0**	87.1

Table 3. Different base model training methods in S2S dataset. AUROC stands for the performance of base/new class detection (higher is better). In Acc is the accuracy of base classes. Average harmonic accuracy (Avg) is used to evaluate the performance of the incremental tasks.

Method	Session-1	Session-2	Session-3
Upper Bound	73.1	57.8	52.3
Confidence-Based [38]	57.2	42.0	36.4
Learning-Based [38]	8.7	13.4	9.4
Ours	67.4	51.0	46.5

Table 4. The harmonic accuracy of different detection methods in the S2S dataset.

output. As shown in Tab. 4, the harmonic accuracy of the confidence-based method in each session is nearly 10 points (36.4% vs. 46.5%) lower than our method. The harmonic accuracy of the learning-based method is extremely lower than our method due to the binary classifier tends to classify all the examples to the base classes.

Different incremental models training methods. We compare three fine-tuning ways: 1) only fine-tuning the classifier layer; 2) fine-tuning the projection layers with the classifier layer, and 3) fine-tuning the whole network. Besides that, we also compare our simple fine-tuning methods with one cross-domain few-shot learning method [22]. As shown in Tab. 5, the harmonic accuracy is the highest when fine-tuning the last three layers. Fine-tuning the last three layers outperforms the second-best results, fine-tuning the whole network by up to 4.9% in the last session as it can somewhat avoid over-fitting. When fine-tuning the last layer, the channel importance method [22] outperforms the fine-tuning ones, but the whole performance is much lower than fine-tuning the previous three layers. As shown in Tab. 6, only storing one example of past classes can lead to 3.5% performance improvement in the average harmonic accuracy. Compared to directly using the one-hot label of the past samples, matching the soft label can better retain the performance of past classes. When retaining both the hard labels and soft labels for replay, the harmonic accuracy achieves the best.

Discussion. As there exists the label and domain shift between base samples and new samples, the OOD detection performance is better when the difference between ID data and OOD data is significant. We are curious whether our approach still performs well when the new class samples and base class samples come from the same dataset. As shown in Tab. 7, the high last and average accuracy have confirmed that our method can generalize to the standard FSCIL learning where the detection of base and new classes can be regarded as the open-set problem.

Method	Session-1	Session-2	Session-3
Harmonic Accuracy			
channel import.(0.5) [22]	51.0	35.5	22.3
channel import.(1.2) [22]	51.0	33.9	23.1
finetune last one	42.5	34.4	19.5
fine-tune last three	**67.4**	**50.0**	**41.4**
fine-tune all	65.2	46.3	36.5

Table 5. Results of different training methods in S2S dataset.

Method	Session-1	Session-2	Session-3	Avg
Harmonic Accuracy				
No Label	67.4	47.0	33.8	49.4
Hard Label	67.4	50.0	41.4	52.9
Soft Label	67.4	50.4	45.6	54.5
Hard+Soft Label	**67.4**	**51.0**	**46.5**	**55.0**

Table 6. The harmonic accuracy in each incremental session of different replay methods in the S2S dataset.

Method	Sess.1	Sess.2	Sess.3	Sess.4	Sess.5	Avg ↑
FACT [50]	90.4	81.3	77.1	73.5	65.0	77.5
Micro [8]	93.6	83.1	78.2	75.8	67.1	79.6
Ours	93.9	78.9	77.6	74.8	**73.2**	**79.7**

Table 7. Results in ModelNet dataset. We report the class-wise average accuracy in each session.

5. Conclusion

This paper focuses on how to balance the performance of base classes and new classes during the FSCIL process for point-cloud recognition, where the base session may contain many synthetic objects, and the coming data is from the real world. We discriminate between the base and new classes from an OOD detection perspective and use the maximum cosine logit of the base model as the score. Then, we adopt the two-branch structure to avoid catastrophic forgetting of the base classes. By discriminating base and new samples to achieve parameter isolation, the model can better adapt to new classes without sacrificing the performance of base classes. To better mitigate the forgetting of incremental classes, we adopt the soft label and hard label replay together to retain the performance of past new samples. We conduct extensive evaluations of the proposed method on three synthetic-to-real point-cloud datasets and the results show the superiority of our proposed method.

Acknowledgement. This research was supported by the Natural Science Fund of Hubei Province under Grant 2022CFB823, the HUST Independent Innovation Research Fund under Grant 2021XXJS096, the Alibaba Innovation Research program under Grant CRAQ7WHZ11220001-20978282, and grants from the Key Lab of Image Processing and Intelligent Control, Ministry of Education, China.

References

[1] Idan Achituve, Haggai Maron, and Gal Chechik. Self-supervised learning for domain adaptation on point clouds. In *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, pages 123–133, 2021. 2

[2] Eden Belouadah and Adrian Popescu. Il2m: Class incremental learning with dual memory. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 583–592, 2019. 6, 7

[3] Eden Belouadah and Adrian Popescu. Scail: Classifier weights scaling for class incremental learning. In *2020 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 1255–1264, 2020. 6, 7

[4] Jun Cen, Peng Yun, Shiwei Zhang, Junhao Cai, Di Luan, Michael Yu Wang, Ming Liu, and Mingqian Tang. Open-world semantic segmentation for lidar point clouds. *arXiv preprint arXiv:2207.01452*, 2022. 3

[5] Angel X Chang, Thomas Funkhouser, Leonidas Guibas, Pat Hanrahan, Qixing Huang, Zimo Li, Silvio Savarese, Manolis Savva, Shuran Song, Hao Su, et al. Shapenet: An information-rich 3d model repository. *arXiv preprint arXiv:1512.03012*, 2015. 6

[6] Kuilin Chen and Chi-Guhn Lee. Incremental few-shot learning via vector quantization in deep embedded space. In *ICLR*, 2021. 3

[7] Ali Cheraghian, Shafin Rahman, Pengfei Fang, Soumava Kumar Roy, Lars Petersson, and Mehrtash Harandi. Semantic-aware knowledge distillation for few-shot class-incremental learning. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2534–2543, 2021. 3

[8] Townim Faisal Chowdhury, Ali Cheraghian, Sameera Ramasinghe, Sahar Ahmadi, Morteza Saberi, and Shafin Rahman. Few-shot class-incremental learning for 3d point cloud objects. In *ECCV*, 2022. 1, 3, 6, 7, 8

[9] Terrance DeVries and Graham W Taylor. Learning confidence for out-of-distribution detection in neural networks. *arXiv preprint arXiv:1802.04865*, 2018. 3

[10] Jiahua Dong, Yang Cong, Gan Sun, Bingtao Ma, and Lichen Wang. I3dol: Incremental 3d object learning without catastrophic forgetting. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 6066–6074, 2021. 3

[11] Xuefeng Du, Zhaoning Wang, Mu Cai, and Yixuan Li. Vos: Learning what you don't know by virtual outlier synthesis. *arXiv preprint arXiv:2202.01197*, 2022. 3

[12] Dan Hendrycks and Kevin Gimpel. A baseline for detecting misclassified and out-of-distribution examples in neural networks. *arXiv preprint arXiv:1610.02136*, 2016. 3, 7, 8

[13] Dan Hendrycks, Mantas Mazeika, and Thomas Dietterich. Deep anomaly detection with outlier exposure. *arXiv preprint arXiv:1812.04606*, 2018. 3

[14] Michael Hersche, Geethan Karunaratne, Giovanni Cherubini, Luca Benini, Abu Sebastian, and Abbas Rahimi. Constrained few-shot class-incremental learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9057–9067, 2022. 3

[15] Gyuhak Kim, Changnan Xiao, Tatsuya Konishi, Zixuan Ke, and Bing Liu. A theoretical, study on solving continual learning. *arXiv preprint arXiv:2211.02633*, 2022. 3

[16] Guohao Li, Matthias Muller, Ali Thabet, and Bernard Ghanem. Deepgcns: Can gcns go as deep as cnns? In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 9267–9276, 2019. 1, 2

[17] Yangyan Li, Rui Bu, Mingchao Sun, Wei Wu, Xinhan Di, and Baoquan Chen. Pointcnn: Convolution on x-transformed points. *Advances in neural information processing systems*, 31, 2018. 1, 2

[18] Zhizhong Li and Derek Hoiem. Learning without forgetting. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(12):2935–2947, 2018. 6, 7

[19] Huan Liu, Li Gu, Zhixiang Chi, Yang Wang, Yuanhao Yu, Jun Chen, and Jin Tang. Few-shot class-incremental learning via entropy-regularized data-free replay. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXIV*, pages 146–162. Springer, 2022. 3, 5

[20] Yuyang Liu, Yang Cong, Gan Sun, Tao Zhang, Jiahua Dong, and Hongsen Liu. L3doc: Lifelong 3d object classification. *IEEE Transactions on Image Processing*, 30:7486–7498, 2021. 3

[21] Chunjie Luo, Jianfeng Zhan, Xiaohe Xue, Lei Wang, Rui Ren, and Qiang Yang. Cosine normalization: Using cosine similarity instead of dot product in neural networks. In *Artificial Neural Networks and Machine Learning–ICANN 2018: 27th International Conference on Artificial Neural Networks, Rhodes, Greece, October 4-7, 2018, Proceedings, Part I 27*, pages 382–391. Springer, 2018. 2

[22] Xu Luo, Jing Xu, and Zenglin Xu. Channel importance matters in few-shot image classification. In *International Conference on Machine Learning*, pages 14542–14559. PMLR, 2022. 8

[23] Pratik Mazumder, Pravendra Singh, and Piyush Rai. Few-shot lifelong learning. In *AAAI*, 2021. 3

[24] Can Peng, Kun Zhao, Tianren Wang, Meng Li, and Brian C. Lovell. Few-shot class-incremental learning from an open-set perspective. In *ECCV*, 2022. 1, 6, 7

[25] Can Peng, Kun Zhao, Tianren Wang, Meng Li, and Brian C Lovell. Few-shot class-incremental learning from an open-set perspective. In *European Conference on Computer Vision*, pages 382–397. Springer, 2022. 3

[26] Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 652–660, 2017. 2

[27] Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. *Advances in neural information processing systems*, 30, 2017. 1, 2

[28] Can Qin, Haoxuan You, Lichen Wang, C-C Jay Kuo, and Yun Fu. Pointdan: A multi-scale 3d domain adaption network for point cloud representation. *Advances in Neural Information Processing Systems*, 32, 2019. 2

[29] Haoxi Ran, Jun Liu, and Chengjie Wang. Surface representation for point clouds. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18942–18952, 2022. 1

[30] Jeremy Reizenstein, Roman Shapovalov, Philipp Henzler, Luca Sbordone, Patrick Labatut, and David Novotny. Common objects in 3d: Large-scale learning and evaluation of real-life 3d category reconstruction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10901–10911, 2021. 6

[31] Xiaoyu Tao, Xiaopeng Hong, Xinyuan Chang, Songlin Dong, Xing Wei, and Yihong Gong. Few-shot class-incremental learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020. 3

[32] Engkarat Techapanurak, Masanori Suganuma, and Takayuki Okatani. Hyperparameter-free out-of-distribution detection using softmax of scaled cosine similarity. *arXiv preprint arXiv:1905.10628*, 2019. 3

[33] Mikaela Angelina Uy, Quang-Hieu Pham, Binh-Son Hua, Thanh Nguyen, and Sai-Kit Yeung. Revisiting point cloud classification: A new benchmark dataset and classification model on real-world data. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 1588–1597, 2019. 3, 6

[34] Sagar Vaze, Kai Han, Andrea Vedaldi, and Andrew Zisserman. Open-set recognition: A good closed-set classifier is all you need. *arXiv preprint arXiv:2110.06207*, 2021. 7, 8

[35] Yue Wang, Yongbin Sun, Ziwei Liu, Sanjay E Sarma, Michael M Bronstein, and Justin M Solomon. Dynamic graph cnn for learning on point clouds. *Acm Transactions On Graphics (tog)*, 38(5):1–12, 2019. 1, 2, 6

[36] Hongxin Wei, Renchunzi Xie, Hao Cheng, Lei Feng, Bo An, and Yixuan Li. Mitigating neural network overconfidence with logit normalization. *arXiv preprint arXiv:2205.09310*, 2022. 3, 8

[37] Yandong Wen, Kaipeng Zhang, Zhifeng Li, and Yu Qiao. A discriminative feature learning approach for deep face recognition. In *European conference on computer vision*, pages 499–515. Springer, 2016. 8

[38] Tz-Ying Wu, Gurumurthy Swaminathan, Zhizhong Li, Avinash Ravichandran, Nuno Vasconcelos, Rahul Bhotika, and Stefano Soatto. Class-incremental learning with strong pre-trained models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9601–9610, 2022. 2, 7, 8

[39] Wenxuan Wu, Zhongang Qi, and Li Fuxin. Pointconv: Deep convolutional networks on 3d point clouds. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9621–9630, 2019. 1, 2

[40] Zhirong Wu, Shuran Song, Aditya Khosla, Fisher Yu, Linguang Zhang, Xiaoou Tang, and Jianxiong Xiao. 3d shapenets: A deep representation for volumetric shapes. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1912–1920, 2015. 6

[41] Xiang Xiang, Yuwen Tan, Qian Wan, and Jing Ma. Coarse-to-fine incremental few-shot learning. *arXiv preprint arXiv:2111.14806*, 2021. 1, 3, 5

[42] Jihan Yang, Shaoshuai Shi, Zhe Wang, Hongsheng Li, and Xiaojuan Qi. St3d: Self-training for unsupervised domain adaptation on 3d object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10368–10378, 2021. 2

[43] Chuangguan Ye, Hongyuan Zhu, Bo Zhang, and Tao Chen. A closer look at few-shot 3d point cloud classification. *International Journal of Computer Vision*, 131(3):772–795, 2023. 2

[44] Chi Zhang, Nan Song, Guosheng Lin, Yun Zheng, Pan Pan, and Yinghui Xu. Few-shot incremental learning with continually evolved classifiers. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 12450–12459, 2021. 1, 3

[45] Jingyang Zhang, Nathan Inkawhich, Yiran Chen, and Hai Li. Fine-grained out-of-distribution detection with mixup outlier exposure. *arXiv preprint arXiv:2106.03917*, 2021. 3

[46] Renrui Zhang, Ziyu Guo, Wei Zhang, Kunchang Li, Xupeng Miao, Bin Cui, Yu Qiao, Peng Gao, and Hongsheng Li. Pointclip: Point cloud understanding by clip. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8552–8562, 2022. 3, 6

[47] Zihan Zhang and Xiang Xiang. Decoupling maxlogit for out-of-distribution detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3388–3397, 2023. 3

[48] Hengshuang Zhao, Li Jiang, Jiaya Jia, Philip HS Torr, and Vladlen Koltun. Point transformer. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 16259–16268, 2021. 3

[49] Na Zhao and Gim Hee Lee. Static-dynamic co-teaching for class-incremental 3d object detection. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 3436–3445, 2022. 3

[50] Da-Wei Zhou, Fu-Yun Wang, Han-Jia Ye, Liang Ma, Shiliang Pu, and De-Chuan Zhan. Forward compatible few-shot class-incremental learning. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9036–9046, 2022. 1, 3, 6, 7, 8