

Foundation Model-Powered 3D Few-Shot Class Incremental Learning via Training-free Adaptor

Sahar Ahmadi¹, Ali Cheraghian^{2,3}, Morteza Saberi¹, Md.Towsif Abir⁴,
Hamidreza Dastmalchi⁵, Farookh Hussain¹, and Shafin Rahman⁴

¹ University of Technology Sydney, Australia ² Data61, CSIRO, Australia
³ Australian National University, Australia ⁴ North South University, Bangladesh
⁵ York University, Canada

{sahar.ahmadi, morteza.saberi, farookh.hussain}@uts.edu.au,
{shafin.rahman, towsif.abir}@northsouth.edu,
ali.cheraghian@data61.csiro.au, hrd@yorku.ca

Abstract. Recent advances in deep learning for processing point clouds hold increased interest in Few-Shot Class Incremental Learning (FSCIL) for 3D computer vision. This paper introduces a new method to tackle the Few-Shot Continual Incremental Learning (FSCIL) problem in 3D point cloud environments. We leverage a foundational 3D model trained extensively on point cloud data. Drawing from recent improvements in foundation models, known for their ability to work well across different tasks, we propose a novel strategy that does not require additional training to adapt to new tasks. Our approach uses a dual cache system: first, it uses previous test samples based on how confident the model was in its predictions to prevent forgetting, and second, it includes a small number of new task samples to prevent overfitting. This dynamic adaptation ensures strong performance across different learning tasks without needing lots of fine-tuning. We tested our approach on datasets like ModelNet, ShapeNet, ScanObjectNN, and CO3D, showing that it outperforms other FSCIL methods and demonstrating its effectiveness and versatility. The code is available at https://github.com/ahmadisahar/ACCV_FCIL3D.

Keywords: Incremental learning · Few-shot learning · 3D point cloud

1 Introduction

In recent years, point cloud processing based on deep learning models has become a crucial research direction in computer vision due to its wide range of potential applications in real-world scenarios. Despite significant progress in this field [19, 24, 26, 47], much of the research has been carried out in controlled environments. When designing a point cloud classification model, it is practical to consider scenarios where data for all classes cannot be collected simultaneously. Typically, we start with numerous training samples for some classes, termed the base task, to develop a baseline model, and then gradually collect data for the remaining classes, termed novel tasks. Additionally, due to hardware limitations or privacy

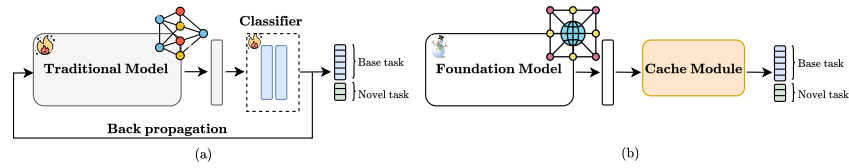


Fig. 1: (a) Existing methods [9, 11, 13] for FSCIL typically employ a traditional vision model trained from scratch on the base task, followed by a classifier. Adding novel classes requires fine-tuning with a few novel training samples, often overfitting the novel classes and forgetting the base classes. (b) In contrast, our proposed FSCIL strategy leverages a foundation model pre-trained on a large dataset, which offers strong generalization with minimal effort compared to traditional vision models. Specifically, to incorporate novel classes into the base classes, we introduce a novel strategy that eliminates the need for fine-tuning, thereby reducing both forgetting and overfitting issues. Instead, we use a novel training-free adaptation module to seamlessly integrate novel classes with existing base classes with minimal effort.

concerns, retraining the model with base task data may not be feasible when adapting the baseline model to novel task samples. This situation leads to the problem of catastrophic forgetting, where the model tends to forget the old classes while learning the new ones. Furthermore, data collection for new classes is often limited and we may not obtain more than a few samples for new classes, leading to overfitting issues in novel classes. The combination of these two issues is studied in the literature under the umbrella of Few-Shot Class Incremental Learning (FSCIL). Specifically for 3D point cloud data, the base task usually consists of synthetic data. In contrast, the novel tasks consist of real scan data, leading to domain gap issues that add more complexity to the FSCIL problem for 3D point cloud data than the 2D image domain.

Existing methods [13, 33, 35] (see Fig. 1 (a)) to address the forgetting problem in FSCIL often rely on rehearsal strategies to mitigate the forgetting issue. This approach entails replaying samples from old classes, usually stored in memory, while learning the novel classes to address the forgetting problem. Additionally, to train the few-shot novel classes, they [11, 12, 21] usually fine-tune a base model through back-propagation, previously trained on the base task, to learn the few-shot novel classes. Unfortunately, this strategy leads to the overfitting issue of novel classes. In contrast, this paper (see Fig. 1 (b)) introduces a novel training-free adaptation strategy applied on top of a foundational model, eliminating the need to fine-tune the base model and thus avoiding common issues in FSCIL such as forgetting and overfitting. Specifically, our approach ensures that the base model remains intact while efficiently learning new classes without compromising the knowledge of previously learned ones. This preserves the integrity of the base model and enhances its ability to generalize on few-shot novel classes.

In this paper, we leverage a 3D foundation model [49], trained on an extensive number of 3D point cloud samples, to tackle the FSCIL problem setting. Foundation models in the literature have demonstrated powerful generalization across

incremental tasks with minimal effort [6, 16]. Moreover, to address the forgetting and overfitting issues in FSCIL, we propose a novel training-free adaptation strategy that eliminates the need to fine-tune the foundation model for novel tasks. Specifically, to mitigate forgetting the base task, we introduce a novel mechanism to leverage test samples from the base task during inference. These test samples are selected on the basis of the confidence score of the foundation model and stored in a cache for later use during inference. Additionally, to control overfitting, we maintain few-shot samples from novel tasks in the adaptor. This dual cache approach ensures the model remains robust against forgetting, while efficiently learning new classes without overfitting.

Overall, the main contributions of our proposed method are:

- We leverage the power of a 3D foundation model, applying it for the first time to the 3D FSCIL task.
- We introduce a novel training-free adaptation strategy that utilizes the incoming test samples to dynamically adapt the model for future test samples. This approach helps maintain performance in previously learned classes while effectively learning new ones.
- We achieve state-of-the-art results in three cross-data set settings, demonstrating the robustness and generalizability of our method.

2 Related work

Point cloud processing: Previous deep learning approaches for 3D point clouds primarily addressed the learning problem by converting the point cloud data into intermediate representations. These methods included rendering the 3D point cloud into 2D images [25, 30], or constructing meshes [5, 20] for further processing. However, these approaches were constrained by their limited ability to accurately represent and understand complex 3D scenes and non-isometric shapes [24]. PointNet [24] was a pioneering work that explored the direct processing of 3D point clouds without any intermediate representation. However, by design, PointNet overlooked the local structures induced by the distance metric. PointNet++ [26] effectively resolved this issue by processing sets of points sampled in a metric space in a hierarchical fashion, allowing the network to capture local structures more accurately. Building upon this, several studies [17, 19, 23, 28, 39, 41] have suggested convolutional techniques designed to extract local features. PointConv [39] introduced an inverse density scale to re-weight the continuous function learned by MLP, which corresponds to the Monte Carlo approximation of the continuous 3D convolution.

Simultaneously, other research efforts [37, 38, 45] considered each point cloud as a graph vertex to extract features in spatial or spectral domains. For example, DGCNN [38] proposed a method that dynamically computes graphs at each layer of the neural network, improving the representation power of point clouds by capturing local geometric structures and recovering topology. In contrast, PointGCN [45] leveraged localized graph convolutions with two types of graph downsampling operations to effectively explore the local structure of point

clouds. To address the challenges posed by the irregular and unordered nature of point cloud data, Guo *et al.* [14] introduced a framework based on the Transformer architecture, which has achieved success in natural language processing and image processing.

Few-shot class incremental learning: Tao et al. [35] pioneered FSCIL framework for image data. The authors proposed a framework that leveraged a neural gas (NG) network to preserve the topology of the feature manifold formed by different classes, stabilizing the old class knowledge and improving representation learning for few-shot new classes. In a subsequent work [12], a novel approach was proposed that extends inductive zero-shot learning (ZSL) to transductive ZSL and Generalized ZSL (GZSL) for 3D point cloud classification while addressing challenges related to domain adaptation, hubness, and data bias. Cheraghian et al. [11] proposed a novel vision-language approach, integrating class semantic information from language space using distillation to mitigate catastrophic forgetting, along with an attention mechanism to address overfitting on few-shot novel tasks. FSCIL3D [13] introduced the innovative concept of Microshape. By leveraging Microshapes, the model could handle incremental training with few-shot examples more effectively, bridging the gap between synthetic and real data. The work by Tan et al. [33] explored cross-domain FSCIL applied to point-cloud recognition, where their base model discriminates between base samples (treated as in-distribution) and new samples (considered out-of-distribution).

Foundation models: Foundation models represent a significant evolution in the field of computer vision, distinguished by their ability to generalize across a wide range of tasks and modalities. These models are trained using vast datasets that span various domains, which imbue them with unprecedented flexibility and capability to handle diverse applications, from image recognition to multimodal reasoning that combines text, images, and audio data. The core architectural innovations in foundation models, such as dual encoders and sophisticated fusion mechanisms, allow efficient integration and processing of multimodal information, enabling these models to perform tasks with a degree of sophistication that mirrors human cognitive abilities [1]. One of the most notable features of foundation models is their proficiency in ‘zero-shot’ learning, where the model applies the knowledge acquired during training to new tasks it has never explicitly learned. For instance, models like CLIP can accurately classify images or generate descriptions based on textual prompts without direct training on those specific tasks. This capability not only showcases the robust generalization of the models, but also reduces the need for extensive task-specific data, simplifying deployment in various real-world scenarios [27]. The deployment of foundational models presents significant challenges. Their training requires substantial computational resources, which presents sustainability concerns. Moreover, using imbalanced datasets can perpetuate biases, leading to ethical issues. Additionally, the absence of standardized benchmarks makes it difficult to assess the effectiveness of these models in various tasks. Addressing these issues requires ongoing research to develop more efficient training methods, ensure fairness, and create reliable evaluation metrics [1].

3 Method

3.1 Problem formulation

Consider a series of T tasks denoted by $\mathbf{Q} = \{Q^1, Q^2, \dots, Q^T\}$. Here, Q^1 represents the base task, and the subsequent tasks are novel tasks that are incrementally added. \mathcal{C}^t signifies the label space i.e, the classes within task Q^t during training. Note that the training label spaces between different tasks are disjoint, i.e., for any $i, j \in [1, T]$ and $i \neq j$, $\mathcal{C}^i \cap \mathcal{C}^j = \emptyset$. Each task's classes are linked with prompt descriptions, noted as \mathcal{P}^t . Therefore, each task can be depicted as a tuple $Q^t = \{\mathcal{X}_i^t, \mathbf{y}_i^t, \mathbf{p}_i^t\}_{i=1}^{n_t}$, where $\mathcal{X}_i^t = \{\mathbf{x}_{i,j}^t\}_{j=1}^l$ represents a 3D point cloud object with coordinates $\mathbf{x}_{i,j}^t \in \mathbb{R}^3$. Furthermore, $\mathbf{y}_i^t \in \mathcal{C}^t$ and $\mathbf{p}_i^t \in \mathcal{P}^t$ denote the label of the point cloud and its associated class prompt description, respectively. Within the FSCIL framework, for the base task Q^1 , the model undergoes training on a large-scale synthetic 3D dataset. As for $t > 1$, training data are sourced from real-world 3D point clouds with only a few instances. The model is trained sequentially across tasks $t = 1, \dots, T$. However, during the training of the t -th task Q^t , the model encounters \mathcal{X}^t , \mathbf{y}^t , and $\{\mathcal{P}^1, \mathcal{P}^2, \dots, \mathcal{P}^t\}$. During inference, the model trained on the current task Q^t is anticipated to classify test samples from both the current and preceding tasks, namely $\{Q^1, Q^2, \dots, Q^t\}$.

3.2 Model overview

Given the input sample \mathcal{X}_i^t , its feature representation $\mathbf{v}_i^t \in \mathbb{R}^m$ is extracted using the vision encoder V_e . The prompts $\{\mathbf{p}_1, \mathbf{p}_2, \dots, \mathbf{p}_C\}$ of all classes from the current and preceding tasks are then processed through the text encoder T_e , resulting in the feature representations $\{\mathbf{e}_1, \mathbf{e}_2, \dots, \mathbf{e}_C\}$, with $\mathbf{e}_j \in \mathbb{R}^m$. Next, the vision and text features are concatenated and fed into an alignment module A . This module connects features from two different modalities: vision and language. Specifically, the alignment module A generates a scalar value a_{ij}^t , ranging from 0 to 1, serving as a measure of the similarity between the visual and prompt feature embeddings. We then construct a similarity vector between the point cloud feature \mathbf{v}_i^t and all class candidate features $\{\mathbf{e}_1, \mathbf{e}_2, \dots, \mathbf{e}_C\}$ as follows: $\mathbf{a}_i^t = \{a_{i1}^t, a_{i2}^t, \dots, a_{iC}^t\}$. This similarity vector \mathbf{a}_i^t is then fed into an adaptor module comprising two caches: a base task cache B and a novel task cache N . The base task cache contains test samples of the base task, selected based on a policy during inference to control forgetting of the base task. The novel task cache comprises few-shot training samples of novel classes to help the model learn new classes without fine-tuning. Finally, the updated similarity vector \mathbf{b}_i^t is extracted from the adaptor module and merged with the original \mathbf{a}_i^t to construct the final score \mathbf{z}_i^t .

3.3 Foundation model

In recent years, vision-language foundation models have gained significant attention in computer vision tasks [27, 43, 52]. This paper uses the Uni3D vision-language 3D foundation model [42], exclusively trained on a substantial corpus

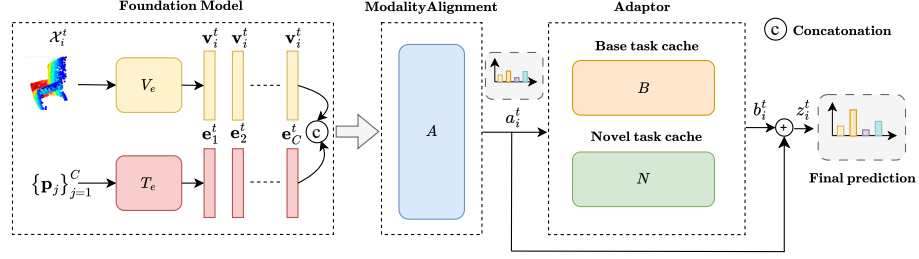


Fig. 2: Feature $\mathbf{v}_i^t \in \mathbb{R}^m$ is extracted from input \mathcal{X}_i^t using vision encoder V_e . Prompts $\{\mathbf{p}_1, \mathbf{p}_2, \dots, \mathbf{p}_C\}$ are processed via text encoder T_e to obtain features $\{\mathbf{e}_1, \mathbf{e}_2, \dots, \mathbf{e}_C\}$. These features are concatenated and aligned by module A , producing similarity vector \mathbf{a}_i^t . This vector is refined by an adaptor module with base task cache B and novel task cache N , resulting in the final score \mathbf{z}_i^t .

of point cloud-text pairs, as our backbone. This model includes a vision encoder V_e , responsible for extracting features from input point cloud data \mathcal{X}_i^t , and a text encoder T_e , which generates embeddings for the input class prompt description \mathbf{p}_i . The outputs of the vision encoder V_e and the text encoder T_e are denoted as $\mathbf{v}_i^t \in \mathbb{R}^m$ and $\mathbf{e}_j \in \mathbb{R}^m$, respectively. Furthermore, these outputs are aligned in the same embedding space. However, additional alignment between vision and text modalities is required for the FSCIL task, using training samples from the base task \mathcal{C}^1 .

Modality alignment: To further align the vision and language branches for the downstream task in the FSCIL setting, we train an alignment module A using a training sample of the base task \mathcal{C}^1 . This alignment module, also referred to as a relation module [32] in the literature, provides a similarity score between $[0, 1]$. For the alignment module, we use three fully connected layers with 2048, 1024, and 1 hidden units. For each training sample, we generate a score for each class of the base task as $a_{ij}^t = \gamma \circ A \circ (\mathbf{v}_i^t \oplus \mathbf{e}_j)$, $j \in \mathcal{C}^t$, $t = 1$, where \oplus is the concatenation operator, A is the alignment module, and γ is the sigmoid function. For each feature a_{ij} and the corresponding ground truth \mathbf{y}_i , we train the A module for the base task using the binary cross-entropy cost function as follows:

$$L_r = -\frac{1}{|\mathcal{S}|} \sum_{\mathbf{y}_i \in \mathcal{S}} \left(\mathbf{1}(y_i^t == k) \log(a_{ik}^t) + (1 - \mathbf{1}(y_i^t == k)) \log(1 - a_{ik}^t) \right), \quad (1)$$

where \mathcal{S} denotes the set of true labels in the base task. The trained alignment module A will be frozen for few-shot novel tasks.

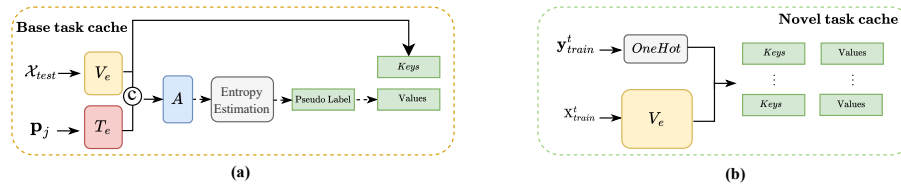


Fig. 3: (a) Base task cache: This cache stores test samples from the base task to address forgetting issues, selecting samples based on their entropy values. The cache updates when a new test sample has a lower entropy than those currently stored. (b) Novel task cache: This cache contains training samples from few-shot novel classes.

3.4 Training-free adaptor and Inference

We propose a novel adaptor module to address the issue of forgetting base task classes while accommodating novel classes without encountering overfitting. This module comprises two caches containing key-value pairs representing the base and novel classes. Specifically, one cache stores features related to base task samples along with their pseudo-labels. At the same time, the other accumulates few-shot training samples from task 1 to the current task with their labels. During inference, test sample features act as queries to retrieve the most relevant information stored in the caches. Leveraging this retrieved information, the output for each test sample is adapted to optimize performance. Importantly, this caching approach does not require additional parameters or training.

Base task cache: The base task cache B consists of key-value pairs organized as a dynamic queue for each class. It aims to store features from the base task test samples as keys that produce high-quality pseudo-labels. Initially, this cache is empty for each class and then filled with appropriate key-value pairs during the inference of the base task. Given the capacity of samples that can be stored for each class in the cache, this method gradually incorporates test predictions with lower entropy to maintain high-quality pseudo-labels. Consider the text and vision encoder networks denoted as T_e and V_e , respectively. For all classes in the base task, we compute the text features using predefined prompts $\mathbf{p}_j \in \mathcal{P}^1$. Each test sample is also processed by the vision encoder to obtain representations v_i . To construct the base cache, a pseudo-label, which is a one-hot vector from a categorical distribution, is generated for each test data \mathcal{X}_i by applying the softmax function on the output logits derived from the combination of text and vision features obtained through the alignment module A . This pseudo-label, along with its corresponding vision feature, must satisfy two conditions to be placed in the cache: 1) The capacity of the number of samples for that class \hat{L}_B has not been reached. In this case, the pseudo-label l along with the corresponding v_i is added to Q_B and L_B as a key-value pair for that class. 2) If the capacity has been reached, we check whether the new sample has a lower entropy than the existing samples in the cache. If it does, it replaces the sample

with the highest entropy, $\{\mathbf{q}^{\text{ent}}, \hat{\ell}_b^{\text{ent}}\}$.

$$H(\mathbf{a}_i^t) < H(\mathbf{a}_i^{\text{ent}}). \quad (2)$$

Here, H denotes the entropy function, indicating the level of uncertainty. By considering these two conditions, in addition to adhering to the sample capacity for each class, we ensure that the pseudo-labels in the cache are of high quality.

Novel task cache: For the novel task cache N , we employ feature embeddings extracted from few-shot training samples of newly introduced classes. With K -shot training samples available per class in the novel task, we aim to construct a key-value cache module as an adapter. Each sample undergoes feature extraction using the vision encoder. These extracted features from the vision encoder, paired with their respective class labels, are then integrated into Q_n and L_n as key-value pairs for each class.

Inference: During inference, the cache module, which includes key-value pairs obtained from the base cache and the novel cache, utilizes the features obtained from the vision encoder for the input test data sample as a query. It checks which of the stored features in the cache has the highest match with this query. In that case, it uses the information from the corresponding key-value pair to retrieve the results from the prediction output generated by the relation module for this data. The adaptive prediction vector \mathbf{b}_i^t using the cache is obtained as follows:

$$P_{\text{cache}}(\mathbf{v}_i^t) = A(\mathbf{v}_i^t \mathbf{Q}^T) \mathbf{L}, \quad (3)$$

where A is the adaptation function introduced by [44]:

$$A(u) = \exp(-\beta(1-u)), \quad \text{where } u \in [0, 1]. \quad (4)$$

and the one-hot vector \mathbf{L} represents the stored value for each key, derived from the corresponding cache information. Thus, the output \mathbf{a}_i^t obtained for the test sample is updated by the cache, and the final output is computed as follows:

$$\mathbf{z}_i^t = \mathbf{a}_i^t + \alpha \mathbf{b}_i^t. \quad (5)$$

4 Experiments

Datasets. Our paper leverages four distinct 3D datasets, which include synthetic structures (ModelNet [40] and ShapeNet [8]) as well as real-scanned datasets (ScanObjectNN [36] and CO3D [29]). Adhering to the experimental setup proposed by [13], our framework is designed to facilitate cross-dataset incremental learning with focused classifications. These experiments aim to bridge the gap between synthetic and scanned data by establishing base classes in synthetic datasets and gradually introducing classes from scan-derived datasets. Detailed experimental configurations are provided in Table 1.

Table 1: Summary of our experimental setups.

Experiment Setups	# Base Classes	# Novel Classes	# Tasks	# Train in Base	# Test in Base	# Test in Novel
ModelNet40 → ScanObjectNN	26	11	4	4999	1496	475
ShapeNet → ScanObjectNN	44	15	4	22797	5845	581
ShapeNet → CO3D	39	50	11	26287	6604	1732

Implementation details. In our implementation, we employ the pre-trained text encoder component of the ‘EVA02-E-14-plus’ CLIP model [31] to extract feature embedding from class names within our dataset. For processing point cloud data, we utilize the ‘base’ scale configuration of the Uni3D architecture [49], specifically the ‘eva02_base_patch14_448’ model as our point cloud encoder. This model choice aligns with the scalability principles outlined by [42], effectively balancing computational efficiency with the ability to capture detailed spatial features. Equipped with 88 million parameters, the ‘Base’ model optimizes our computational resources while ensuring comprehensive 3D data representation. The point cloud and text encoders are initialized with pre-trained weights, which are frozen during training. Furthermore, we incorporate a trainable alignment module as defined by [32] to integrate the point cloud and text features. This alignment network comprises three fully connected layers with configurations of 2048, 1024, and 1 neurons, respectively. LeakyReLU activations are used in the initial layers, while the output layer employs a Sigmoid activation. The alignment module is specifically trained as a feature extractor for task 0, utilizing basic data across 10 epochs. We employ the Adam optimizer, setting a learning rate of 0.001 and a batch size 25. Additionally, we maintain a cache of five key samples and their corresponding values for each task, incrementally building this dataset. Our experiments use the PyTorch framework on a single NVIDIA A100 GPU.

Evaluation metrics. In each incremental phase, we assess the accuracy by considering both base and novel classes. Following the approach outlined in [34], we then determine the rate of accuracy decline, denoted as $\Delta = \frac{|acc_T - acc_0|}{acc_0} \times 100$. Here, acc_T is the accuracy at the final task, while acc_0 is the accuracy at the outset. The parameter Δ provides a consolidated measure of the method’s efficacy, with a lower Δ indicating superior performance. This evaluation is based on the average accuracy calculated over ten trials, each with a different random seed. The accuracy and Δ metrics cannot precisely evaluate the balance between forgetting old class samples and learning novel classes. This is because a large portion of the dataset consists of base classes, and only a small number of training samples are used for new classes. Therefore, even if the model does not perform well on new classes but achieves good accuracy for the base task, it can still report good numbers for both metrics. To better assess how well our model retains knowledge of base tasks and performs on new classes, we use the metric introduced in paper [22], known as harmonic accuracy. This metric is calculated

Table 2: Summary of FSCIL results.

Method	ShapeNet \rightarrow CO3D										ModelNet \rightarrow ScanObjectNN					ShapeNet \rightarrow ScanObjectNN							
	39	44	49	54	59	64	69	74	79	84	89	$\Delta\downarrow$	26	30	34	37	$\Delta\downarrow$	44	49	54	59	$\Delta\downarrow$	
<i>FT</i>	81.0	20.2	2.3	1.7	0.8	1.0	1.0	1.3	0.9	0.5	1.6	98.0	88.4	6.4	6.0	1.9	97.9	81.4	38.7	4.0	0.9	98.9	
<i>Joint</i>	81.0	79.5	78.3	75.2	75.1	74.8	72.3	71.3	70.0	68.8	67.3	16.9	88.4	79.7	74.0	71.2	19.5	81.4	82.5	79.8	78.7	3.3	
LwF [18]	81.0	57.4	19.3	2.3	1.0	0.9	0.8	1.3	1.1	0.8	1.9	97.7	88.4	35.8	5.8	2.5	97.2	81.4	47.9	14.0	5.9	92.8	
IL2M [2]	81.0	45.6	36.8	35.1	31.8	33.3	34.0	31.5	30.6	32.3	30.0	63.0	88.4	58.2	52.9	52.0	41.2	81.4	53.2	43.9	45.8	43.7	
ScaIL [3]	81.0	50.1	45.7	39.1	39.0	37.9	38.0	36.0	33.7	33.0	35.2	56.5	88.4	56.5	55.9	52.9	40.2	81.4	49.0	46.7	40.0	50.9	
EEIL [7]	81.0	75.2	69.3	63.2	60.5	57.9	53.0	51.9	51.3	47.8	47.6	41.2	88.4	70.2	61.0	56.8	35.7	81.4	74.5	69.8	63.4	22.1	
FACT [48]	81.4	76.0	70.3	68.1	65.8	63.5	63.0	60.1	58.2	57.5	55.9	31.3	89.1	72.5	68.3	63.5	28.7	82.3	74.6	69.9	66.8	18.8	
Sem-aware [11]	80.6	69.5	66.5	62.9	63.2	63.0	61.2	58.3	58.1	57.2	55.2	31.6	88.5	73.9	67.7	64.2	27.5	81.3	70.6	65.2	62.9	22.6	
Microshape [13]	82.6	77.9	73.9	72.7	67.7	66.2	65.4	63.4	60.6	58.1	57.1	30.9	89.3	73.2	68.4	65.1	27.1	82.5	74.8	71.2	67.1	18.7	
C3PR [10]	83.6	80.0	77.8	75.4	72.8	72.3	70.3	67.9	64.9	64.1	63.2	24.4	88.3	75.7	70.6	67.8	23.2	84.5	77.8	75.5	71.9	14.9	
Ours	87.386.284.482.280.779.678.276.876.174.572.616.8												87.7 84.7 81.5 79.2 9.6										90.8 86.5 86.4 85.6 5.75

based on the following formula: $A_h = \frac{2 \times A_b \times A_n}{A_b + A_n}$, 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. The higher the harmonic accuracy, the better the network maintains a balance between the accuracy of old and novel classes.

4.1 Main results

In this section, we compare our method against several state-of-the-art (SOTA) approaches, including FT, Joint, LwF [18], IL2M [2], ScaIL [3], EEIL [7], FACT [48], Sem-aware [11], Microshape [13], cross-domain [33] and C3PR [10]. FT (Fine-Tuning) involves fine-tuning the model on new classes without revisiting old classes, often leading to catastrophic forgetting. Jointly retraining the model on all classes, assuming access to all data is often impractical. State-of-the-art methods such as IL2M [2], ScaIL [3], EEIL [7], LwF [18], FACT [48], and Sem-aware [11] were initially reported on 2D datasets. We adapted their implementations using PointNet features for 3D datasets. Our results are summarized in Table 2. Our observations are as follows: Due to the presence of noise in the 3D real-can dataset, achieving satisfactory performance poses significant challenges. FT shows the lowest accuracy across all datasets, with a Δ of 98.0 for ShapeNet to CO3D and 97.2 for ModelNet to ScanObjectNN. This significant drop is attributed to catastrophic forgetting, as the model is fine-tuned on new classes without revisiting old ones. Conversely, Microshape [13] achieves the highest accuracy due to its innovative use of Microshape descriptions and their alignment with semantic prototypes, effectively minimizing domain gaps and providing superior results in each incremental task. IL2M [2] and ScaIL [3] propose special training mechanisms tailored for 2D image examples. However, their performance drops when applied to 3D datasets due to the inherent complexities and noise in 3D data. Both LwF [18] and EEIL [7] apply knowledge distillation in their loss functions, with EEIL [7] enhancing LwF [18] by additionally using exemplars. Despite these enhancements, they struggle with the challenges posed by 3D data, leading to higher performance degradation. FACT addresses few-shot class incremental learning through feature augmentation and classification tuning, while Sem-aware [11] successfully incorporates class-semantic embedding

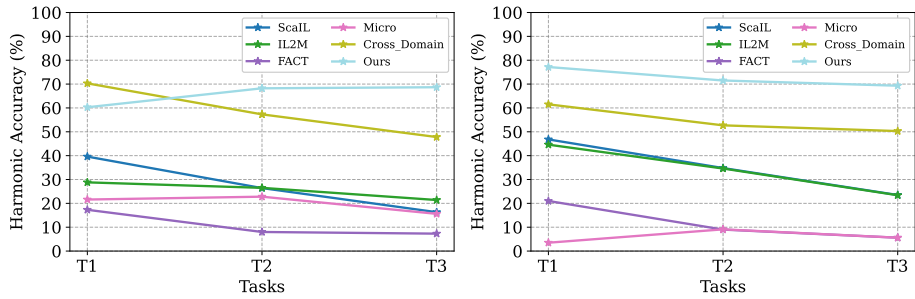


Fig. 4: Comparison of the harmonic accuracy with SOTA methods on ShapeNet to ScanObjectNN and ModelNet40 to ScanObjectNN datasets.

information during training, providing a marginal boost. However, both methods still fall short in 3D scenarios compared to our approach. Overall, these state-of-the-art methods primarily target 2D image data and fail to address the specific challenges of 3D data, such as noise and the need for robust spatial feature extraction. C3PR [10] uses a combination of learned projections, model reprogramming, and prompt engineering to tackle FSCIL for 3D point cloud objects. Overall, our approach significantly outperforms other methods in terms of accuracy.

In FSCIL, achieving high performance on both base and novel classes is essential. To assess this, we compare our proposed method with state-of-the-art approaches using the harmonic mean metric. Higher values in this metric indicate effective performance across both base and novel test samples, while a decrease suggests poorer performance on either base or novel tasks. It is worth noting that this evaluation method was introduced by [33] and is referred to as the cross-domain method. As shown in Fig. 4, our proposed method significantly outperforms all other methods on both ShapeNet-to-ScanObjectNN and ModelNet-to-ScanObjectNN datasets.

4.2 Ablation study

In this section, we conduct ablation studies to evaluate the effectiveness of our designs. All ablation studies are performed on the ShapeNet to ScanObjectNN dataset, where our method achieves an accuracy of 85.6% for the final task under the default settings.

The impact of cache: In Fig. 5(a), we observe a comparison of the model’s performance with and without the use of the cache. When our model relies solely on the predictions obtained from the alignment module A , the model’s accuracy in predicting the base task data does not suffer from the forgetting problem as the number of tasks increases. However, the model’s performance on novel classes significantly drops. For tasks 1 to 3, the values of A_n are 10.08, 5.9, and 4.6, respectively, indicating that the model is incapable of predicting new tasks and

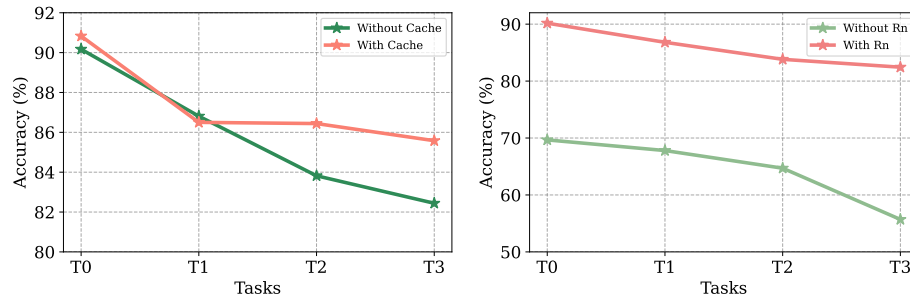


Fig. 5: The influence of caches(a) and the impact use relation module after encoders vs zeroshot(b).

A_{cct} is a result of the model’s good performance on the base task due to training the relation module with the data from this task. However, when we use caches in the adaptor module to adapt the model’s predictions, we observe that for incremental tasks 1 to 3, the values of A_n are 45.8, 55.4, and 56.1, respectively. Consequently, we achieve an average harmonic accuracy of 67.73%, indicating a balance between predicting new class data and not forgetting the previous task data.

The role of alignment module: In Fig. 5(b), we present the results obtained with and without using the alignment module. Suppose that we do not use the Relation Module to classify the results obtained from both encoders. In that case, we are in a zero-shot learning scenario since there are no parameters to train, and we only use the point cloud encoder and text encoder with pre-trained and frozen weights. In this case, the output for each sample is obtained by calculating the maximum cosine similarity between the outputs of the text encoder and the point cloud encoder. The alignment module is also evaluated when trained only for the base task, and no samples are stored in the cache. The use of the Relation Module allows for combining the features obtained from both encoders, resulting in a better-learned feature space.

The impact of the number of samples in the cache: We studied the effect of the shot capacity, which refers to the maximum number of pairs of keys-values per class, both in the basic and the novel caches. The aim is to find the optimal balance between the diversity and accuracy of the key-value pairs. Considering 5 shots per class from the training data for each class except task zero, we examined cache construction from size 1 to 10. As explained in the previous section, the selection of each sample for the test cache is based on entropy, while the training cache is selected randomly. Given that random selection might affect the results, we repeated our experiments three times in this section and reported the average results. The results shown in Fig. 6(a) indicate that increasing the cache size improves accuracy until the entire training data fits into the cache. However, accuracy decreases when the number of test cache data exceeds the training data. This decrease in accuracy results from the base

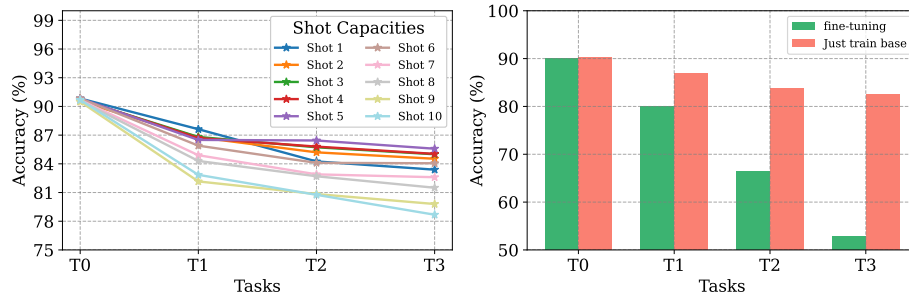


Fig. 6: (a) The influence of the number of samples in the cache. (b) The effect of finetuning the relation module in each task versus training solely for the base task and then freezing.

cache data, including pseudo-labels obtained based on the lowest entropy of the model’s predictions. Consequently, these data are accompanied by noise.

Full fine-tuning vs only base task training: In Fig. 6(b), we examine the effect of fine-tuning the alignment module for novel classes. Making the alignment module trainable for all tasks leads to a decrease in accuracy. This is because training the network with few-shot data during incremental stages enhances overfitting, causing the network weights to shift towards learning the classes of the new tasks, thus forgetting the previous tasks. However, if the alignment module is only trained for the base task and then frozen for subsequent tasks, we observe that the accuracy for the base task is maintained.

The impact of α and β : The hyperparameter α used in Equation 4 controls the extent to which new predictions obtained from the cache module are combined with predictions from the relation module’s output. A larger α indicates greater importance given to the knowledge obtained from the cache data. To select an appropriate α , we evaluated the performance of the model based on the mean harmonic accuracy between tasks, which is a more precise metric than the plain accuracy. With β set to 2, we varied α from 0 to 3. An α of zero means that the cache module’s knowledge is not used at all, effectively resulting in a model without a cache. Next, we examine the hyperparameter β used in Equation 5. This parameter controls the sharpness of similarity. When β is large, only the most similar training samples to the test image in the embedding space significantly affect the prediction and vice versa. With α set to 2, we varied β from 0 to 3. Table 3 shows that the optimal mean harmonic accuracy is achieved when both α and β are set to 2. Therefore, the knowledge obtained from cache data significantly contributes to achieving desirable results in multi-class incremental learning without the need for additional training.

Table 3: Ablation studies of impact α and β on mean Harmonic Accuracy.

Residual Ratio α	0	0.5	1	2	3	Sharpness Ratio β	0	0.5	1	2	3
HM	12.6	35.3	53.2	67.7	65.0	HM	6.3	58.2	64.5	67.7	65.2

5 Discussion

The impact of foundation model: In our approach, we harness the capabilities of a 3D vision-language foundation model [49], which significantly enhances the performance of our method. This observation underscores the broader applicability of 3D foundation models to tackle related downstream tasks under low data conditions, such as zero-shot learning, few-shot learning, and dealing with long-tailed distributions. These models demonstrate their utility by effectively leveraging semantic and structural information embedded in 3D data, thereby improving adaptability and generalization across diverse and challenging learning scenarios. This highlights their potential to advance various applications in 3D computer vision and beyond.

Limitation: Although our method has demonstrated state-of-the-art results in the 3D point cloud domain, it also highlights limitations that warrant discussion. Specifically, we have not fully capitalized on the potential of vision-language foundation models. Future research directions include exploring advanced fine-tuning techniques like prompt tuning strategies [50, 51], LORA [15], and enhanced prompt engineering using large language models (LLMs) such as GPT [4] or in-context learning approaches [46].

6 Conclusion

In conclusion, this paper presents a pioneering approach tailored to address the challenges of Few-Shot Continual Incremental Learning (FSCIL) in 3D computer vision. By leveraging a robust 3D foundation model trained on extensive point cloud data, we design a novel training-free adaptation module to effectively manage forgetting and overfitting issues inherent in FSCIL scenarios. Our method utilizes a dual cache strategy that optimally utilizes previous task test samples based on model confidence scores to maintain performance on base classes while integrating few-shot samples from new tasks to enhance generalization and prevent overfitting. The experimental results across diverse datasets, including ModelNet, ShapeNet, ScanObjectNN, and CO3D, demonstrate our approach’s superior efficacy and versatility compared to existing FSCIL methods. This work contributes significantly to advancing the capabilities of 3D vision-language models in handling continual learning tasks, paving the way for more robust and adaptable solutions in real-world applications of 3D computer vision.

Acknowledgement. This work was supported by the North South University (NSU) Conference Travel and Research Grants (CTRG) 2023–2024 (Grant ID: CTRG-23-SEPS-20).

References

1. Awais, M., Naseer, M., Khan, S., Anwer, R.M., Cholakkal, H., Shah, M., Yang, M.H., Khan, F.S.: Foundational models defining a new era in vision: A survey and outlook. arXiv preprint arXiv:2307.13721 (2023)
2. Belouadah, E., Popescu, A.: Il2m: Class incremental learning with dual memory. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 583–592 (2019)
3. Belouadah, E., Popescu, A.: Scail: Classifier weights scaling for class incremental learning. In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision. pp. 1266–1275 (2020)
4. Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J.D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., Agarwal, S., Herbert-Voss, A., Krueger, G., Henighan, T., Child, R., Ramesh, A., Ziegler, D., Wu, J., Winter, C., Hesse, C., Chen, M., Sigler, E., Litwin, M., Gray, S., Chess, B., Clark, J., Berner, C., McCandlish, S., Radford, A., Sutskever, I., Amodei, D.: Language models are few-shot learners. In: Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M., Lin, H. (eds.) *Advances in Neural Information Processing Systems*. vol. 33, pp. 1877–1901. Curran Associates, Inc. (2020)
5. Bruna, J., Zaremba, W., Szlam, A., LeCun, Y.: Spectral networks and locally connected networks on graphs. arXiv preprint arXiv:1312.6203 (2013)
6. Cao, X., Lu, H., Huang, L., Liu, X., Cheng, M.M.: Generative multi-modal models are good class incremental learners. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 28706–28717 (June 2024)
7. Castro, F.M., Marín-Jiménez, M.J., Guil, N., Schmid, C., Alahari, K.: End-to-end incremental learning. In: Proceedings of the European Conference on Computer Vision (ECCV). pp. 233–248 (2018)
8. Chang, A.X., Funkhouser, T., Guibas, L., Hanrahan, P., Huang, Q., Li, Z., Savarese, S., Savva, M., Song, S., Su, H., et al.: Shapenet: An information-rich 3d model repository. arXiv preprint arXiv:1512.03012 (2015)
9. Chen, K., Lee, C.G.: Incremental few-shot learning via vector quantization in deep embedded space. In: International Conference on Learning Representations (2021)
10. Cheraghian, A., Hayder, Z., Ramasinghe, S., Rahman, S., Jafaryahya, J., Petersson, L., Harandi, M.: Canonical shape projection is all you need for 3d few-shot class incremental learning. In: *Computer Vision – ECCV 2024* (2024)
11. Cheraghian, A., Rahman, S., Fang, P., Roy, S.K., Petersson, L., Harandi, M.: Semantic-aware knowledge distillation for few-shot class-incremental learning. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2021)
12. Cheraghian, A., Rahman, S., Ramasinghe, S., Fang, P., Simon, C., Petersson, L., Harandi, M.: Synthesized feature based few-shot class-incremental learning on a mixture of subspaces. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV) (2021)
13. Chowdhury, T., Cheraghian, A., Ramasinghe, S., Ahmadi, S., Saberi, M., Rahman, S.: Few-shot class-incremental learning for 3d point cloud objects. In: *European Conference on Computer Vision*. pp. 204–220. Springer (2022)
14. Guo, M.H., Cai, J.X., Liu, Z.N., Mu, T.J., Martin, R.R., Hu, S.M.: Pct: Point cloud transformer. *Computational Visual Media* (2021)

15. Hu, E.J., Shen, Y., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., Wang, L., Chen, W.: Lora: Low-rank adaptation of large language models. In: International Conference on Learning Representations (2022), <https://openreview.net/forum?id=nZeVKeeFYf9>
16. Kim, J., Ku, Y., Kim, J., Cha, J., Baek, S.: Vlm-pl: Advanced pseudo labeling approach for class incremental object detection via vision-language model. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops. pp. 4170–4181 (June 2024)
17. Li, Y., Bu, R., Sun, M., Wu, W., Di, X., Chen, B.: Pointcnn: Convolution on x-transformed points. In: Proceedings of the Advances in Neural Information Processing Systems (NeurIPS) (2018)
18. Li, Z., Hoiem, D.: Learning without forgetting. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **40**(12), 2935–2947 (2017)
19. Liu, Y., Fan, B., Xiang, S., Pan, C.: Relation-shape convolutional neural network for point cloud analysis. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2019)
20. Masci, J., Boscaini, D., Bronstein, M., Vandergheynst, P.: Geodesic convolutional neural networks on riemannian manifolds. In: Proceedings of the IEEE International Conference on Computer Vision Workshops. pp. 37–45 (2015)
21. Mazumder, P., Singh, P., Rai, P.: Few-shot lifelong learning. In: AAAI (2021)
22. Peng, C., Zhao, K., Wang, T., Li, M., Lovell, B.C.: Few-shot class-incremental learning from an open-set perspective. In: European Conference on Computer Vision. pp. 382–397. Springer (2022)
23. Poulencard, A., Rakotosaona, M.J., Ponty, Y., Ovsjanikov, M.: Effective rotation-invariant point cnn with spherical harmonics kernels. In: Proceedings of the IEEE International Conference on 3D Vision (3DV) (2019)
24. Qi, C.R., Su, H., Mo, K., Guibas, L.J.: Pointnet: Deep learning on point sets for 3d classification and segmentation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2017)
25. Qi, C.R., Su, H., Nießner, M., Dai, A., Yan, M., Guibas, L.J.: Volumetric and multi-view cnns for object classification on 3d data. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 5648–5656 (2016)
26. Qi, C.R., Yi, L., Su, H., Guibas, L.J.: Pointnet++: Deep hierarchical feature learning on point sets in a metric space. In: Proceedings of the Advances in Neural Information Processing Systems (NeurIPS) (2017)
27. Radford, A., Kim, J.W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J.: Learning transferable visual models from natural language supervision. In: International Conference on Machine Learning. pp. 8748–8763. PMLR (2021)
28. Rao, Y., Lu, J., Zhou, J.: Spherical fractal convolutional neural networks for point cloud recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2019)
29. Reizenstein, J., Shapovalov, R., Henzler, P., Sbordone, L., Labatut, P., Novotny, D.: 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. pp. 10901–10911 (2021)
30. Su, H., Maji, S., Kalogerakis, E., Learned-Miller, E.: Multi-view convolutional neural networks for 3d shape recognition. In: Proceedings of the IEEE International Conference on Computer Vision (ICCV). pp. 945–953 (2015)

31. Sun, Q., Fang, Y., Wu, L., Wang, X., Cao, Y.: Eva-clip: Improved training techniques for clip at scale. arXiv preprint arXiv:2303.15389 (2023)
32. Sung, F., Yang, Y., Zhang, L., Xiang, T., Torr, P.H., Hospedales, T.M.: Learning to compare: Relation network for few-shot learning. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2018)
33. Tan, Y., Xiang, X.: Cross-domain few-shot incremental learning for point-cloud recognition. In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV). pp. 2307–2316 (January 2024)
34. Tan, Z., Ding, K., Guo, R., Liu, H.: Graph few-shot class-incremental learning. In: Proceedings of the Fifteenth ACM International Conference on Web Search and Data Mining (2022)
35. Tao, X., Hong, X., Chang, X., Dong, S., Wei, X., Gong, Y.: Few-shot class-incremental learning. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2020)
36. Uy, M.A., Pham, Q.H., Hua, B.S., Nguyen, T., Yeung, S.K.: 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. pp. 1588–1597 (2019)
37. Wang, C., Samari, B., Siddiqi, K.: Local spectral graph convolution for point set feature learning. In: Proceedings of the European Conference on Computer Vision (ECCV) (2018)
38. Wang, Y., Sun, Y., Liu, Z., Sarma, S.E., Bronstein, M.M., Solomon, J.M.: Dynamic graph cnn for learning on point clouds. ACM Transactions on Graphics (TOG) (2019)
39. Wu, W., Qi, Z., Fuxin, L.: Pointconv: Deep convolutional networks on 3d point clouds. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2019)
40. Wu, Z., Song, S., Khosla, A., Yu, F., Zhang, L., Tang, X., Xiao, J.: 3d shapenets: A deep representation for volumetric shapes. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 1912–1920 (2015)
41. Xu, Y., Fan, T., Xu, M., Zeng, L., Qiao, Y.: Spidercnn: Deep learning on point sets with parameterized convolutional filters. In: Proceedings of the European Conference on Computer Vision (ECCV) (2018)
42. Zhang, B., Yuan, J., Shi, B., Chen, T., Li, Y., Qiao, Y.: Uni3d: A unified baseline for multi-dataset 3d object detection. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 9253–9262 (2023)
43. Zhang, R., Guo, Z., Zhang, W., Li, K., Miao, X., Cui, B., Qiao, Y., Gao, P., Li, H.: Pointclip: Point cloud understanding by clip. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 8552–8562 (2022)
44. Zhang, R., Zhang, W., Fang, R., Gao, P., Li, K., Dai, J., Qiao, Y., Li, H.: Tip-adapter: Training-free adaptation of clip for few-shot classification. In: European Conference on Computer Vision. pp. 493–510. Springer (2022)
45. Zhang, Y., Rabbat, M.: A graph-cnn for 3d point cloud classification. In: Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP) (2018)
46. Zhang, Y., Zhou, K., Liu, Z.: What makes good examples for visual in-context learning? In: Oh, A., Naumann, T., Globerson, A., Saenko, K., Hardt, M., Levine, S. (eds.) Advances in Neural Information Processing Systems. vol. 36, pp. 17773–17794. Curran Associates, Inc. (2023)

47. Zhao, H., Jiang, L., Jia, J., Torr, P.H., Koltun, V.: Point transformer. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV) (2021)
48. Zhou, D.W., Wang, F.Y., Ye, H.J., Ma, L., Pu, S., Zhan, D.C.: Forward compatible few-shot class-incremental learning. In: CVPR (2022)
49. Zhou, J., Wang, J., Ma, B., Liu, Y.S., Huang, T., Wang, X.: Uni3d: Exploring unified 3d representation at scale. In: International Conference on Learning Representations (ICLR) (2024)
50. Zhou, K., Yang, J., Loy, C.C., Liu, Z.: Conditional prompt learning for vision-language models. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2022)
51. Zhou, K., Yang, J., Loy, C.C., Liu, Z.: Learning to prompt for vision-language models. *International Journal of Computer Vision (IJCV)* (2022)
52. Zhu, X., Zhang, R., He, B., Guo, Z., Zeng, Z., Qin, Z., Zhang, S., Gao, P.: Pointclip v2: Prompting clip and gpt for powerful 3d open-world learning. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 2639–2650 (2023)