

# Few-Shot Point Cloud Semantic Segmentation via Contrastive Self-Supervision and Multi-Resolution Attention

Jiahui Wang, Haiyue Zhu<sup>†</sup>, Haoren Guo, Abdullah Al Mamun, Cheng Xiang and Tong Heng Lee

**Abstract**—This paper presents an effective few-shot point cloud semantic segmentation approach for real-world applications. Existing few-shot segmentation methods on point cloud heavily rely on the fully-supervised pretrain with large annotated datasets, which causes the learned feature extraction bias to those pretrained classes. However, as the purpose of few-shot learning is to handle unknown/unseen classes, such class-specific feature extraction in pretrain is not ideal to generalize into new classes for few-shot learning. Moreover, point cloud datasets hardly have a large number of classes due to the annotation difficulty. To address these issues, we propose a contrastive self-supervision framework for few-shot learning pretrain, which aims to eliminate the feature extraction bias through class-agnostic contrastive supervision. Specifically, we implement a novel contrastive learning approach with a learnable augmentor for a 3D point cloud to achieve point-wise differentiation, so that to enhance the pretrain with managed overfitting through the self-supervision. Furthermore, we develop a multi-resolution attention module using both the nearest and farthest points to extract the local and global point information more effectively, and a center-concentrated multi-prototype is adopted to mitigate the intra-class sparsity. Comprehensive experiments are conducted to evaluate the proposed approach, which shows our approach achieves state-of-the-art performance. Moreover, a case study on practical CAM/CAD segmentation is presented to demonstrate the effectiveness of our approach for real-world applications.

## I. INTRODUCTION

Point cloud semantic segmentation is a significant task in computer vision, which benefits many practical applications such as intelligent manufacturing, autonomous driving, robot manipulation, etc. Due to its complexity and unstructured property, point cloud segmentation learning generally requires more resources including annotated datasets, computation hardware, etc. Although existing fully-supervised methods have demonstrated their capability [1]–[6], one problem with these methods is that they heavily rely on large-scale annotated datasets, which are unpractical or costly to obtain for many real-world applications. In addition, models trained under fully-supervised learning has relatively poor generalization capability to novel classes. As many practical applications demand the learning capability in a data-efficient manner, few-shot point cloud semantic segmentation [7]–[9] gains more and more popularity due to its effectiveness and practicability.

<sup>†</sup> Corresponding author: zhu.haiyue@simtech.a-star.edu.sg

J. Wang, H. Guo, A. Mamun, C. Xiang and T. H. Lee are with College of Design and Engineering, Electrical and Computer Engineering, National University of Singapore, 117582, Singapore.

H. Zhu is with Singapore Institute of Manufacturing Technology (SIMTech), Agency for Science, Technology and Research (A\*STAR), 2 Fusionopolis Way, Singapore 138634, Republic of Singapore.

Meta-learning and metric-learning are commonly used branches of few-shot learning, which mostly follow pretrain, train and test paradigm [10], [11]. By embracing few-shot learning, point cloud segmentation models generally have the abilities to learn more effectively from a relatively small annotated dataset with better generalization performance. WPS-Net [12] proposed a learnable coherent point transformer for point cloud segmentation that can predict a smooth geometric transformation field to morph the template 3D shape towards the input shape. The multi-view comparison method is also developed for few-shot point cloud semantic segmentation [13]. Recently, attMPTI [7] proposed a notable multi-prototype transductive method that captures the geometric dependencies and semantic correlations between points. Our previous work [14] adopts T-Net [2] to enhance the model comprehension ability with a center loss. However, its performance is still limited in complex scenarios.

Traditionally, few-shot point cloud segmentation is usually pretrained using fully-supervised learning, where the reinforced feature projection is highly coupled with those pretrained classes leading to class-specific feature bias. As a result, such biased feature extraction might be harmful to the few-shot generalization as its purpose is to handle unseen/unknown classes. Moreover, fully-supervised pretrain generally requires a large-scale annotated dataset in a close domain to the few-shot tasks, which may not be practically feasible for many real-world tasks. To address such weaknesses, we adopt the contrastive self-supervision technique [15] for few-shot point cloud segmentation to eliminate such feature extraction bias, so that the learned encoder projection can be more universal to extract better and more general features for further few-shot learning. Moreover, as our approach is self-supervised, it largely removes the burden to prepare an annotated dataset for the pretrain purpose.

In this work, we propose learnable-augmentor-based contrastive learning to achieve the effective self-supervised pretrain, where the learning objective is to differentiate the individual points in two alternative views. Unlike 2D image augmentation, point cloud augmentation process data in an unstructured and out-of-order format. Consider these characteristics, we integrate a trainable augmentor from PointAugment [16] in our contrastive learning which can automatically optimize the augmentation to exploit better unstructured variation and deformation in both shape and point levels. As a result, the self-supervised feature extraction is reinforced in the differentiation capability on the point level without class-specific guidance, which benefits the following few-shot adaptation.

Moreover, we propose a Multi-Resolution Attention (MRA) module for the few-shot learning to better capture both the local and global information. Although the attention mechanisms [17], [18] are widely used in point cloud tasks, most approaches directly use attention to all points, which includes excessive distractive information from the redundant background and affects the performance. This problem is especially more serious when segmenting small objects, where the vast background points can easily overwhelm the foreground feature. Based on this finding, we design a MRA block that only conducts attention to local nearest neighbours and global farthest points. Due to the selective point attention, our approach also saves the computation memory and achieves better performance.

Finally, the contribution of this work can be summarized as,

- We propose a few-shot point cloud semantic segmentation framework with contrastive self-supervised pretrain, which can not only mitigate the influence of class-specific bias but also significantly reduce the requirement for annotated large-scale pretrain datasets.
- A computation-efficient MRA block is proposed, which is able to conduct attention more effectively and reduce the influence of distractive background information.
- We propose a centering-based multi-prototype generation process that can decrease the intra-class distance and enlarge the inter-class distance, which ensures the dense distribution within the class and discrimination between classes.

## II. RELATED WORK

### A. Point Cloud Semantic Segmentation

Point cloud semantic segmentation can be treated as a point-wise classification problem, which needs to consider point-to-point difference and topology distribution. PointNet [1] firstly expresses an end-to-end deep neural network can directly deal with raw point clouds and obtain reasonable performance rather than transforming point clouds to voxel grids or multi-view images. However, PointNet lacks the ability to extract local geometric features. PointMLP [19] extracts local features using residual multi-layer perceptron (MLP) blocks and a geometric affine module. While DGCNN [4] proposed Edgeconv block to enhance the ability of capturing local features. With the increasing popularity of the attention mechanism, Stratified Transformer [6] addresses this problem with an attention-based block. In our work, we propose an MRA block that can not only extract local nearest-neighbour features but also reserve the global key-shape features.

### B. Few-Shot Learning

Few-shot learning is a research area to deal with sample scarcity. It aims to train a model which is capable of generalizing to new classes without a large number of labeled data [20]. Several approaches have been proposed to address the few-shot issue, for example, transfer learning [21] transfers experience from the source task with abundant

labeled data to the target tasks, where available data is scarce. Meta-learning [22], [23] trains a meta-learner that can extract meta-knowledge across tasks. Among these methods, metric-learning is noticeable because of its simplicity and effectiveness. Metric-learning-based approaches usually have the paradigm that it computes the representation of each support sample to propagate the information to the query feature and assigns each query instance to the support class with the largest similarity to itself. Few-shot learning is also capable to collaborate with other learning algorithm such as incremental learning [24] to achieve better generalization ability. In the point cloud few-shot domain, attMPTI [7] represents each class with multiple prototypes in the feature space and apply a non-parametric method to predict the classes for query instance based on the support feature. In our work, we propose a contrastive self-supervision to enhance the feature extraction ability for the few-shot learning through the pretrain phase, and a MRA module with a metric-learning-based prototype and label propagation is proposed to facilitate the few-shot adaptation.

## III. METHODOLOGY

### A. Problem Formulation

According to the previous works [7], the few-shot point cloud semantic segmentation to address in this work is a  $N$ -way  $K$ -shot segmentation learning problem. Given a support set  $\mathcal{S} = \{\{\mathbf{P}_s^{n,k}, \mathbf{Y}_s^{n,k}\}_{k=1}^K\}_{n=1}^N$  with  $N$  classes in total to segment, each  $n$ -th class contains a number of  $K$  support point clouds  $\mathbf{P}_s$  with their corresponding labels  $\mathbf{Y}_s$ , our learning task is to obtain an optimal network  $F_{\theta^*}(\cdot)$  through  $\mathcal{S}$  that can perform the segmentation task on the query set  $\mathcal{Q} = \{\mathbf{P}_q, \mathbf{Y}_q\}_i$ , i.e.,

$$\begin{aligned} \mathbf{Y}_q' &= F_{\theta}(\mathbf{P}_q|\mathcal{S}), \mathbf{P}_q \in \mathcal{Q}, \\ \theta^* &= \underset{\theta}{\operatorname{argmin}} \sum_{\mathbf{P}_q \in \mathcal{Q}} \mathcal{L}(\mathbf{Y}_q, \mathbf{Y}_q'), \end{aligned} \quad (1)$$

where  $\mathbf{P}_s \in \mathbb{R}^{M \times f}$  contains  $M$  points with  $f$  channels of features including  $xyz$  coordinates, RGB channels, etc.,  $\mathbf{Y}_q$  and  $\mathbf{Y}_q' \in \mathbb{R}^{M \times 1}$ ,  $\mathcal{L}$  is the loss function,

Following the common practices, the learning procedures for few-shot point cloud segmentation is generally divided into two phases, i.e., pretrain phase and few-shot learning phase. The point cloud dataset  $\mathcal{D} = \mathcal{D}_{train} \cup \mathcal{D}_{test}$  contains two non-overlapping subsets  $\mathcal{D}_{train}$  and  $\mathcal{D}_{test}$  with class sets  $\mathcal{Z}_{train}$  and  $\mathcal{Z}_{test}$ , respectively, where  $\mathcal{Z}_{train} \cap \mathcal{Z}_{test} = \emptyset$ . The pretrain phase aims to obtain a better base point cloud feature extractor so that it can facilitate the following few-shot learning. Usually, fully-supervised learning is adopted in pretrain to obtain a better initialization of the feature extractor. The few-shot learning phase employs the standard episode learning for the training and testing, where each episode consists of  $N \times K$  support point clouds and  $N$  query point clouds to form a  $N$ -way  $K$ -shot episode. In both the pretrain phase and training part of the few-shot learning phase, the testing dataset  $\mathcal{D}_{test}$  with class set  $\mathcal{Z}_{test}$  is not accessible for the training supervision.

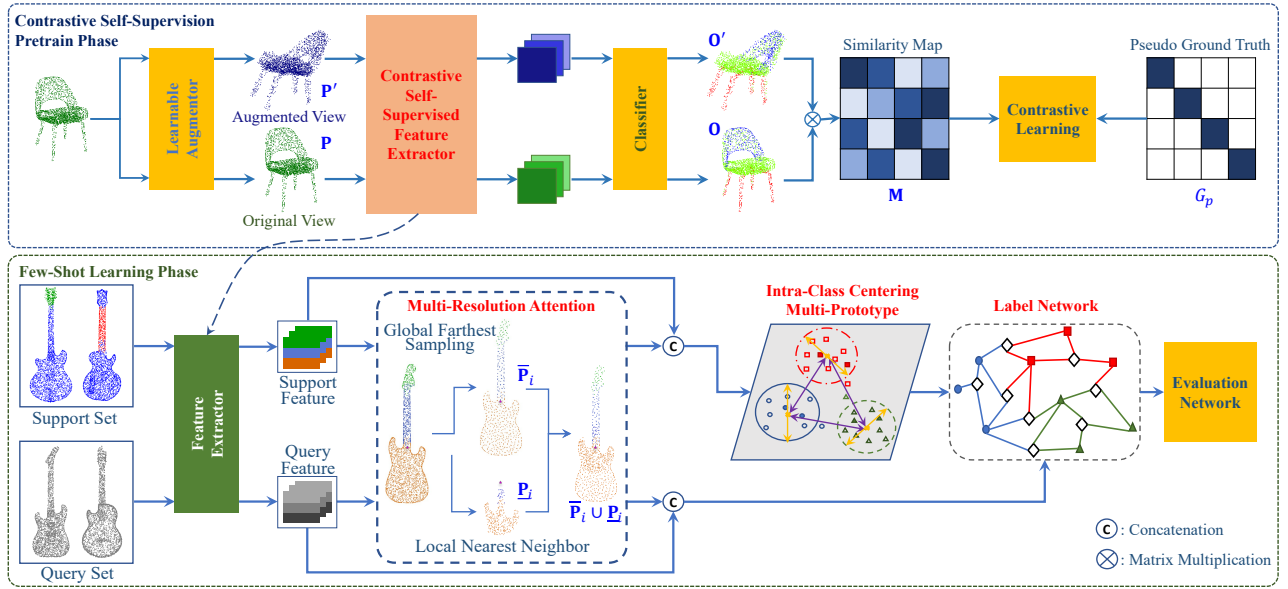


Fig. 1. Our framework includes pretrain and few-shot learning phases. In the pretrain phase, a trainable augmentor is employed to generate the augmented view of input point cloud, where a point-wise contrastive loss is designed to self-supervise the encoder. After the pretrain phase, only the encoder will be used. In the few-shot learning phase, the encoder maps input point clouds to feature space, and a Multi-Resolution Attention (MRA) block extracts the attention feature and concatenates it with the input. With the support feature, the centering-based multi-prototype generation block extracts prototypes and ensures the dense distribution in feature space. The label network then takes prototypes and query features as input and makes graph-based label propagation. Lastly, the loss function is calculated in the evaluation block with the ground-truth label of the query set.

## B. Framework Overview

Fig. 1 illustrates our proposed framework for few-shot point cloud semantic segmentation. In this work, our key idea is to improve the feature extraction capability through the pretrain phase to obtain a better extractor initialization for few-shot learning, where a contrastive self-supervision learning approach is proposed to serve this purpose. The self-supervised encoder projection  $E_\alpha(\cdot)$  is supposed to be less biased to pretrained classes, and thus extract better and more general features for further few-shot learning.

In the few-shot learning phase, our pipeline contains five parts as illustrated in Fig. 1. Firstly, the pretrained DGCNN-based encoder  $E_\alpha$  projects both the query and support point clouds into the feature space, denoted as

$$\mathbf{X}_{s/q} = E_\alpha(\mathbf{P}_{s/q}). \quad (2)$$

In this work, we propose an MRA block to better extract both the local and global information with computational efficiency, denoted as,

$$\mathbf{X}'_{s/q} = MRA_\alpha(\mathbf{X}_{s/q}) \oplus \mathbf{X}_{s/q}, \quad (3)$$

which takes the residual structure that concatenates the original feature with the output of MRA, and  $\oplus$  denotes the concatenation operation. Next, we sample multi-prototype for each class in the MRA feature space to better represent their complex data distribution. Since the multiple prototypes representing the same class are unregularized, which may have large intra-class distances, we use a centering block to ensure the prototypes have dense distribution in feature space. With prototypes of support samples learned, the prediction of query samples is finished in the label network,

which takes  $\mathbf{X}'_q$  and multi-prototype of  $\mathbf{X}'_s$  as input. With label propagation, the network delivers the prediction of each point. Finally, the evaluation network calculates the whole loss function  $\mathcal{L}$  of our network and updates the network parameters by gradient descent.

## C. Contrastive Self-Supervision with Learnable Augmentor

We propose a learnable-augmentor-based contrastive self-supervision technique in the pretrain phase to eliminate the feature bias due to the traditional direct class-specific fully-supervised pretraining. Uniquely, our self-supervision employs a trainable point cloud augmentor to obtain two alternative views for more effective contrastive learning. Inspired by [16], the augmentor first projects the original point cloud  $(xyz)$   $\mathbf{P} \in \mathbb{R}^{M \times 3}$  into a higher-dimension feature space and obtain  $\tilde{\mathbf{P}} \in \mathbb{R}^{M \times C_a}$ , where  $C_a \gg 3$  denotes the feature dimension of the augmentor. Both point-level and shape-level augmentations are considered. The shape-level augmentation trains an MLP network to produce a transformation matrix  $\mathbf{T}_s$ . The point-level augmentation is a trainable MLP producing a displacement noise  $\mathbf{N}_p \in \mathbb{R}^{M \times 3}$ . For shape-level augmentation, we firstly use max pooling on  $\tilde{\mathbf{P}}$  to obtain a feature vector  $F_s \in \mathbb{R}^{1 \times C_a}$ , merge it with a Gaussian distribution noise and fed to the MLP to get  $\mathbf{T}_s$ . For point-level augmentation, we directly fed our network  $N$  copies of  $F_s$  together with  $\tilde{\mathbf{P}}$  to obtain  $\mathbf{N}_p$ . Lastly, our augmented point cloud for contrastive learning can be denoted as,

$$\mathbf{P}' = \mathbf{P} \otimes \mathbf{T}_s + \mathbf{N}_p, \quad (4)$$

where  $\otimes$  refers to matrix multiplication. The augmentor also have multiple non-trainable random augmentation layer such as translation and jittering to enhance the augmentation.

Different from the standard contrastive learning in the instance level, our proposed contrastive learning for point cloud is to differentiate in the point level. Specifically, the two alternative views  $\{\mathbf{P}, \mathbf{P}'\} \in \mathbb{R}^{M \times f}$  are fed into the encoder  $E_\alpha$  with the classification head to get two segmentation outputs  $\{\mathbf{O}, \mathbf{O}'\} \in \mathbb{R}^{M \times N_c}$ , where  $N_c$  is the number of total classes in the pretrain. The contrastive loss is formulated to differentiate the same points from  $\mathbf{O}$  to  $\mathbf{O}'$  as,

$$\mathcal{L}_c = \mathbb{E}_{\mathbf{o}_i \in \mathbf{O}} \left[ -\log \left( \frac{e^{\mathbf{o}_i(\mathbf{o}'_i)^\top}}{\sum_{j=1}^M e^{\mathbf{o}_i(\mathbf{o}'_j)^\top}} \right) \middle| \mathbf{o}'_j \in \mathbf{O}' \right], \quad (5)$$

where  $\mathbf{o}_i$  and  $\mathbf{o}'_i$  are the  $i$ -th point features in  $\mathbf{O}$  and  $\mathbf{O}'$ , respectively.

#### D. Multi-Resolution Attention & Multi-Prototype Module

The multi-resolution attention module is proposed to better enhance both the local and global feature extraction, which with the advantages of computational efficiency and effectiveness. We argue that for the local information, the nearest points are of most critical interest as they are most likely to share the same class. For the global information, key points should be sampled globally to represent the overall scene/body instead of with restrictions in a certain level ‘‘bigger’’ window. Therefore, our key idea is to make use of both the nearest and farthest points as the core concept of multi-resolution attention.

Specifically, consider one query point  $P_i \in \mathbb{R}^{1 \times f}$  in a query point cloud  $\mathbf{P} \in \mathbb{R}^{M \times f}$ , our approach finds its  $K$ -Nearest Neighbours (KNN) set, denoted as

$$\underline{\mathbf{P}}_i = \mathcal{N}(P_i) \in \mathbb{R}^{N_K \times f}, \quad (6)$$

where  $\mathcal{N}(\cdot)$  denotes the KNN operation, and  $N_K$  is the number of nearest neighbours. In contrast, the Farthest Point Sampling (FPS) algorithm is employed to obtain the global key points as

$$\overline{\mathbf{P}}_i = \mathcal{F}(P_i) \in \mathbb{R}^{N_F \times f}, \quad (7)$$

where  $\mathcal{F}(\cdot)$  denotes the FPS operation with respective to  $P_i$ , and  $N_F$  is the number of farthest points. Therefore, we can represent the features of  $\underline{\mathbf{P}}_i$  and  $\overline{\mathbf{P}}_i$  as

$$\begin{aligned} \underline{\mathbf{X}}_i &= \mathbf{E}_\alpha(\underline{\mathbf{P}}_i) \in \mathbb{R}^{N_K \times C}, \\ \overline{\mathbf{X}}_i &= \mathbf{E}_\alpha(\overline{\mathbf{P}}_i) \in \mathbb{R}^{N_F \times C}, \end{aligned} \quad (8)$$

where  $C$  denotes the feature dimension.

In the subsequent multi-resolution attention mechanism, the query, key and value maps  $\mathbf{Q}$ ,  $\mathbf{K}$ , and  $\mathbf{V}$  of a point cloud is generated for  $i$ -th point as,

$$\begin{aligned} \mathbf{Q}_i &= \text{Net}_q(\mathbf{X}_q^i) \in \mathbb{R}^{1 \times C}, \\ \mathbf{K}_i &= \text{Net}_k(\overline{\mathbf{X}}_i \oplus \underline{\mathbf{X}}_i) \in \mathbb{R}^{(N_F + N_K) \times C}, \\ \mathbf{V}_i &= \text{Net}_v(\overline{\mathbf{X}}_i \oplus \underline{\mathbf{X}}_i) \in \mathbb{R}^{(N_F + N_K) \times C}, \end{aligned} \quad (9)$$

where  $\text{Net}_q(\cdot)$ ,  $\text{Net}_k(\cdot)$ , and  $\text{Net}_v(\cdot)$  denotes the corresponding extractor mapping respectively,  $\mathbf{X}_q^i$  is the  $i$ -th feature of the encoder output  $\mathbf{X}_q$ . Take single head as an example, the multi-resolution attention calculation process can be written as,

$$\begin{aligned} \mathbf{A}_i &= \mathbf{Q}_i \otimes \mathbf{K}_i^\top \in \mathbb{R}^{1 \times (N_F + N_K)}, \\ \hat{\mathbf{Q}}_i &= \text{softmax}(\mathbf{A}_i) \otimes \mathbf{V}_i^\top \in \mathbb{R}^{1 \times C}, \end{aligned} \quad (10)$$

where  $\mathbf{A}_i$  is the  $i$ -th point attention map and  $\hat{\mathbf{Q}}_i$  is the multi-resolution attention block output feature,  $\hat{\mathbf{Q}} \in \mathbb{R}^{M \times C}$ . The final feature of query point cloud can be obtained through the concatenation of  $\hat{\mathbf{Q}}$  and  $\mathbf{X}_q$ ,

$$\mathbf{X}'_q = \hat{\mathbf{Q}} \oplus \mathbf{X}_q \in \mathbb{R}^{M \times 2C}. \quad (11)$$

Similarly, support feature  $\mathbf{X}'_s$  can be obtained in the corresponding way.

The support features might be dispersive due to a not well-trained encoder which would lead to a larger misclassification probability. Moreover, we would like to make support features far away from centers of other different classes. Hence, we adopt the center loss [25] to guarantee the close distribution of the same class and sparse distribution of different classes in feature space. The loss function is:

$$\mathcal{L}_r = \frac{1}{2} \sum_{n=0}^N \sum_{i=1}^{M \times K} \frac{\|\mathbf{X}'_{s,i} - C_n\|_2^2}{\left( \sum_{j=0, j \neq n}^N \|\mathbf{X}'_{s,i} - C_j\|_2^2 \right) + \eta} \quad (12)$$

where  $\mathbf{X}'_{s,i}$  is the feature of the  $i$ -th support points of the  $n$ -th class,  $C_n$  is the center of class  $n$  in feature space and  $\eta$  is a hyper-parameter to avoid the denominator becoming 0, it is set to 1 as default. Finally, the compact center-regularized multi-prototype  $\{\boldsymbol{\mu}^n\}_{n=0}^N$  can be generated using the standard method [26].

#### E. Label Network

The label network serves to predict the class for each point in query point clouds based on the feature  $\mathbf{X}'_q$  and prototypes of support set  $\{\boldsymbol{\mu}^n\}_{n=0}^N$ . The basic idea of label propagation algorithm employed is to predict the label of unknown nodes from the information of labeled nodes using graph relationship between samples. For a  $N$ -way  $K$ -shot problem, the graph size is  $Z = N_p \times (N + 1) + |\mathcal{Q}| \times M$ .  $|\mathcal{Q}|$  represent the number of point clouds in the query set,  $\mathbf{X}' \in \mathbb{R}^{Z \times Z}$  denotes the feature matrix of all nodes. We additionally define feature matrix  $\mathbf{W} \in \mathbb{R}^{Z \times Z}$ , where its each value is calculated as

$$\mathbf{W}_{ij} = \exp\left(-\frac{\|\mathbf{X}'_i - \mathbf{X}'_j\|_2^2}{2\sigma^2}\right), \quad (13)$$

where  $\mathbf{X}'_i$  is the  $i$ -th feature in  $\mathbf{X}$ , and  $\sigma$  is the standard deviation of the distance between two nodes. We follow [27] to obtain the optimal prediction matrix  $\mathbf{F}$  and the prediction output map  $\mathbf{H}$ ,

$$\mathbf{F} = (\mathbf{I} - \gamma(\mathbf{D}^{-1/2}(\mathbf{W} + \mathbf{W}^T)\mathbf{D}^{-1/2}))^{-1}\mathbf{L}, \quad (14)$$

$$\mathbf{H}_i^n = \frac{\exp(\mathbf{F}_i^n)}{\sum_{j=0}^N \exp(\mathbf{F}_i^j)}, \quad (15)$$

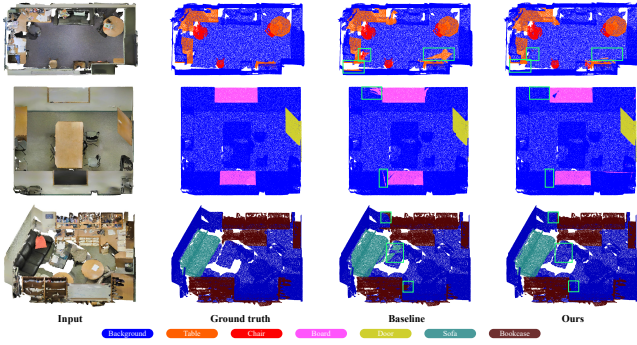


Fig. 2. Result visualization and comparison on s3dis segmentation task.

where  $\mathbf{D}$  is a diagonal matrix whose value is the sum of corresponding rows of  $\mathbf{W}$ .  $\gamma \in (0, 1)$  is probability coefficient.  $\mathbf{L} \in \mathbb{R}^{Z \times (N+1)}$  is a reference matrix, rows of  $\mathbf{L}$  represent prototypes are one-hot labels and other represent unlabeled points(query points) are initial zero. The loss function of this block is a cross-entropy loss,

$$\mathcal{L}_l = -\frac{1}{|\mathcal{Q}|} \frac{1}{M} \sum_{i=1}^{|\mathcal{Q}|} \sum_{j=1}^M \sum_{n=0}^N \delta(\mathbf{Y}_j^i, n) \log(\mathbf{H}_{i,j}^n), \quad (16)$$

where the ground truth label  $\mathbf{Y}_j^i$  represents the  $j$ -th point in the  $i$ -th point cloud of query set and function  $\delta(\cdot)$  is defined as:

$$\delta(x, y) = \begin{cases} 1 & x = y \\ 0 & x \neq y \end{cases} \quad (17)$$

Therefore, the complete loss function in the training phase of our framework is

$$\mathcal{L} = \mathcal{L}_l + \lambda \mathcal{L}_r, \quad (18)$$

where  $\lambda$  is the coefficient to adjust the intensity of centering and concentration.

## IV. EXPERIMENTS

### A. Implementation Details

The network structure is shown in Fig. 1. In the pretrain phase, S3DIS [28] and ShapeNet [29] are adopted to figure out the influence of pretraining on different datasets, we simply add an MLP network as our classifier for prediction. Our batch size is set to 16, the initial learning rate is set to 0.001 and the pretrain epochs are set to 120. The augmentor is trained in advance on ShapeNet dataset. The  $k$ -NN number for the DGCNN-based encoder is set to 200. In the few-shot learning phase, the depth of the MRA block is set to 1 for computing efficiency, we additionally set the centering intensity coefficient  $\lambda$  as 0.1 the label propagation parameter  $\gamma$  is set to 0.9 to ensure the performance. Moreover, the  $k$ -NN number and fps ratio in the MRA block is set to 250 and 0.4, respectively. After initializing the feature extractor weights with the pretrained encoder, we set the Adam optimizer with an initial learning rate of 0.0005 for the feature extractor module, for other modules the learning rate is set to 0.001. Two subsets  $S^0$  and  $S^1$  are split for test in standard way [7],

TABLE I  
COMPARISON OF METHODS ON S3DIS DATASET (%)

Method	2-Way 1-Shot			2-Way 5-Shot		
	$S^0$	$S^1$	mean	$S^0$	$S^1$	mean
Fine-Tuning	36.04	36.69	37.37	55.19	55.40	55.30
attMPTI	52.29	53.84	53.07	61.33	62.42	61.88
Ours+VWA	54.48	54.02	54.26	62.16	61.89	62.03
Ours	<b>54.80</b>	<b>54.13</b>	<b>54.47</b>	<b>63.26</b>	<b>62.61</b>	<b>62.84</b>

we train our model with 100 episodes and 40,000 iterations on both subsets.

### B. Baselines

For baselines, we choose attMPTI [7] and typical fine-tuning network [30] as our comparative models. AttMPTI is a simple but effective network, it extracts prototypes from support feature and uses vanilla label propagation to predict query samples. For fine-tuning network, we freeze the weights of the pretrained feature extractor for the fine-tuning network and only update the parameters in the MLP-based segmentor of it to avoid overfitting. In this work, we only use xyzXYZ without RGB in all datasets for all the experiments conducted in this work.

### C. Comparison Result

Table I shows the result comparison between our framework and baselines. It shows that our model outperforms the baselines in all different few-shot settings, the cost is that our framework trains a little bit slower than the fine-tuning network and attMPTI due to feature concatenation. We also test vanilla window-attention from Stratified Transformer in Table I, denoted as ours+VWA, which shows it is lower than our approach. This agrees with our assumption that the window-attention may lose some global information. The fine-tuning network has an unsatisfied performance especially on 2 way 1-shot task, this is because the embedding features from the feature extractor is insufficient to train the segmentor of the fine-tuning network, and with the scarcity of data, the segmentor hardly can learn the features well. This result also indexes the importance of the attention mechanism which helps in learning a more representative embedding space. The vanilla attention block might projects redundant attention features between query and background points to the feature space.

### D. Ablation Study

We conduct an ablation studies to verify the effectiveness of each component in our method, table II illustrated the result of our experiments. We make the evaluation on the S3DIS  $S^0$  dataset with the 2-way 1-shot few-shot setting. The verification was made by removing each component from the final model, *i.e.*, Exp. VIII. Compared to Exp. I, Exp. II shows the improvement of contrastive pretrain. Obviously, it enhances the segmentation performance. The result in Exp. III and Exp. IV illustrates that the performance of using the centering-based multi-prototype or the MRA block

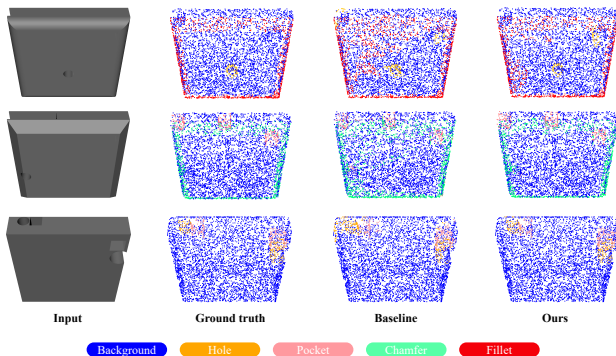


Fig. 3. Result visualization and comparison on 3D CAD/CAM segmentation task

individually is lower than employing the contrastive pretrain. This comparison shows the significance of a well-pretrained encoder in few-shot semantic segmentation.

Table III shows the effectiveness of our contrastive self-supervision pretrain. Without any other additional operations, we only adopt our contrastive self-supervised pretrain to attMPTI in the 2-way 1-shot few-shot semantic segmentation, the result shows that it significantly improves the few-shot segmentation performance. Particularly, Table III indicates that it greatly helps to surmount the overfitting issue in small-dataset pretrain situation. As a result, our contrastive self-supervised pretrain is a general plug-and-play module, which is compatible to many other point cloud few-shot semantic segmentation frameworks.

#### E. Case Study on Practical CAD/CAM Segmentation

To demonstrate the practical effectiveness of our approach on real-world small-data tasks, we prepare a few-shot dataset from EIF 3D CAD [31] and evaluate our algorithm on it, which is an important step in advanced automation workflow especially in CAD/CAM process. The dataset, named as phcf, is a CAD/CAM 3D semantic segmentation task, which includes four types of features, *i.e.* pocket, hole, chamfer, and fillet. However, due to the similar shape between point clouds, this dataset is easily overfitted in the training due to the lack of complexity. Table IV illustrated the best and mean results of different methods, repeated 5 times, and the visualization is shown in Fig. 3. From the results, it can see that our approach is more effective in dealing with real-world data-scarce tasks, where those dataset are often small-scale and will be easily-overfitted in the supervision. Notably, Table III also shows that without an additional large-scale dataset, our approach improves significantly for only contrastive pretrain using the same-domain dataset (phcf). Since our self-supervision does not require annotated dataset, it greatly facilitates the real-world few-shot tasks to conduct the effective pretrain on the close-domain dataset, which finally improves the segmentation accuracy.

## V. CONCLUSIONS

In this work, we propose and develop a noteworthy few-shot point cloud semantic segmentation framework. Firstly,

TABLE II  
2-WAY 1-SHOT ABLATION STUDY ON S3DIS DATASET ( $S^0$ )

ID	Multi-Resolution Attention	Centering-Based Multi-Prototype	Contrastive Pretrain	Accuracy (mIoU)
I				52.29
II			✓	53.78
III		✓		52.56
IV	✓			52.92
V	✓	✓		53.46
VI	✓		✓	54.23
VII		✓	✓	54.07
VIII	✓	✓	✓	<b>54.80</b>

TABLE III  
COMPARISON OF W/O CONTRASTIVE PRETRAIN

Pretrain Dataset	Train Dataset	Supervised Pretrain Result	Contrastive Pretrain Result	$\Delta$
S3DIS	S3DIS	52.29	53.78	+1.49
ShapeNet	S3DIS	52.87	54.21	+1.34
phcf	S3DIS	45.15	50.19	+5.04
S3DIS	phcf	49.40	55.38	+5.98
ShapeNet	phcf	52.34	56.80	+4.46
phcf	phcf	29.97	47.11	<b>+17.14</b>

TABLE IV  
COMPARISON METHODS ON PHCF DATASET (%)

Method	2-Way 1-Shot		2-Way 3-Shot		2-Way 5-Shot	
	Best	Mean	Best	Mean	Best	Mean
Fine-Tuning	41.54	39.01	46.39	46.32	50.41	49.71
attMPTI	49.23	48.64	51.81	51.59	53.02	52.49
Ours	<b>52.71</b>	<b>51.92</b>	<b>53.70</b>	<b>52.83</b>	<b>54.24</b>	<b>54.06</b>

by using contrastive self-supervised pretrain, our approach is capable to reduce the feature bias and overfitting due to the class-specific fully-supervision. Moreover, it enables to use unlabeled datasets for close-domain pretrain in real-world tasks. Secondly, the MRA mechanism helps to mitigate the influence of redundant background information without overlooking local or global features. Thirdly, an intra-class centering multi-prototype generation is adopted to improve the accuracy of prediction. Comprehensive experiments shows that our approach achieves better accuracy comparing to the existing methods. Finally, a practical case study is conducted to shows the effectiveness of our framework in a real-world application.

## ACKNOWLEDGMENT

This research is supported by the National University of Singapore under the NUS College of Design and Engineering Industry-focused Ring-Fenced PhD Scholarship programme. This research is also supported by A\*STAR under its “RIE2025 IAF-PP Advanced ROS2-native Platform Technologies for Cross sectorial Robotics Adoption (M21K1a0104)” programme. The authors would like to acknowledge useful discussions with Dr.Arno Zinke from Hexagon, Manufacturing Intelligence Division, Simufact Engineering GmbH.

## REFERENCES

- [1] C. R. Qi, H. Su, K. Mo, and L. 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 (CVPR)*, 2017, pp. 652–660.
- [2] C. R. Qi, L. Yi, H. Su, and L. J. Guibas, "Pointnet++: Deep hierarchical feature learning on point sets in a metric space," *Advances in neural information processing systems*, vol. 30, 2017.
- [3] H. Thomas, C. R. Qi, J.-E. Deschaud, B. Marcotegui, F. Goulette, and L. J. Guibas, "Kpconv: Flexible and deformable convolution for point clouds," in *Proceedings of the IEEE/CVF international conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 6411–6420.
- [4] Y. Wang, Y. Sun, Z. Liu, S. E. Sarma, M. M. Bronstein, and J. M. Solomon, "Dynamic graph cnn for learning on point clouds," *Acm Transactions On Graphics (tog)*, vol. 38, no. 5, pp. 1–12, 2019.
- [5] H. Zhao, L. Jiang, J. Jia, P. H. Torr, and V. Koltun, "Point transformer," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2021, pp. 16259–16268.
- [6] X. Lai, J. Liu, L. Jiang, L. Wang, H. Zhao, S. Liu, X. Qi, and J. Jia, "Stratified transformer for 3d point cloud segmentation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022, pp. 8500–8509.
- [7] N. Zhao, T.-S. Chua, and G. H. Lee, "Few-shot 3d point cloud semantic segmentation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021, pp. 8873–8882.
- [8] X. Li, L. Feng, L. Li, and C. Wang, "Few-Shot Meta-Learning on Point Cloud for Semantic Segmentation," Apr. 2021, arXiv:2104.02979 [cs].
- [9] Y. Ren, S. Zhao, and L. Bingbing, "Object insertion based data augmentation for semantic segmentation," in *2022 International Conference on Robotics and Automation (ICRA)*, 2022, pp. 359–365.
- [10] Y. Wang, Q. Yao, J. T. Kwok, and L. M. Ni, "Generalizing from a few examples: A survey on few-shot learning," *ACM Comput. Surv.*, vol. 53, no. 3, jun 2020.
- [11] Z. Tian, X. Lai, L. Jiang, S. Liu, M. Shu, H. Zhao, and J. Jia, "Generalized few-shot semantic segmentation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022, pp. 11563–11572.
- [12] L. Wang, X. Li, and Y. Fang, "Few-shot learning of part-specific probability space for 3d shape segmentation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020, pp. 4504–4513.
- [13] X. Chen, C. Zhang, G. Lin, and J. Han, "Compositional prototype network with multi-view comparison for few-shot point cloud semantic segmentation," 2020.
- [14] J. Wang, H. Zhu, H. Guo, A. Al Mamun, P. Vadakkepat, and T. H. Lee, "Cam/cad point cloud part segmentation via few-shot learning," in *2022 IEEE 20th International Conference on Industrial Informatics (INDIN)*. IEEE, 2022, pp. 359–365.
- [15] H. Guo, H. Zhu, J. Wang, P. Vadakkcpat, W. K. Ho, and T. H. Lee, "Masked self-supervision for remaining useful lifetime prediction in machine tools," in *2022 IEEE 20th International Conference on Industrial Informatics (INDIN)*. IEEE, 2022, pp. 353–358.
- [16] R. Li, X. Li, P.-A. Heng, and C.-W. Fu, "Pointaugmt: an auto-augmentation framework for point cloud classification," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020, pp. 6378–6387.
- [17] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. u. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in Neural Information Processing Systems*, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, Eds., vol. 30. Curran Associates, Inc., 2017.
- [18] M.-H. Guo, J.-X. Cai, Z.-N. Liu, T.-J. Mu, R. R. Martin, and S.-M. Hu, "Pct: Point cloud transformer," *Computational Visual Media*, vol. 7, no. 2, pp. 187–199, 2021.
- [19] X. Ma, C. Qin, H. You, H. Ran, and Y. Fu, "Rethinking network design and local geometry in point cloud: A simple residual MLP framework," in *International Conference on Learning Representations*, 2022.
- [20] Y. Li, H. Zhu, Y. Cheng, W. Wang, C. S. Teo, C. Xiang, P. Vadakkepat, and T. H. Lee, "Few-shot object detection via classification refinement and distractor retreatment," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021, pp. 15395–15403.
- [21] K. Weiss, T. M. Khoshgoftaar, and D. Wang, "A survey of transfer learning," *Journal of Big data*, vol. 3, no. 1, pp. 1–40, 2016.
- [22] X. Li, X. Yang, Z. Ma, and J.-H. Xue, "Deep Metric Learning for Few-Shot Image Classification: A Selective Review," arXiv, Tech. Rep. arXiv:2105.08149, May 2021, arXiv:2105.08149 [cs] type: article.
- [23] D. Wang, Y. Cheng, M. Yu, X. Guo, and T. Zhang, "A hybrid approach with optimization-based and metric-based meta-learner for few-shot learning," *Neurocomputing*, vol. 349, pp. 202–211, 2019.
- [24] Y. Li, H. Zhu, S. Tian, F. Feng, J. Ma, C. S. Teo, C. Xiang, P. Vadakkepat, and T. H. Lee, "Incremental few-shot object detection for robotics," in *2022 International Conference on Robotics and Automation (ICRA)*. IEEE, 2022, pp. 8447–8453.
- [25] C. Qi and F. Su, "Contrastive-center loss for deep neural networks," in *2017 IEEE international conference on image processing (ICIP)*. IEEE, 2017, pp. 2851–2855.
- [26] J. Snell, K. Swersky, and R. Zemel, "Prototypical networks for few-shot learning," in *Advances in Neural Information Processing Systems*, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, Eds., vol. 30. Curran Associates, Inc., 2017.
- [27] A. Iscen, G. Tolia, Y. Avrithis, and O. Chum, "Label propagation for deep semi-supervised learning," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.
- [28] I. Armeni, A. Sax, A. R. Zamir, and S. Savarese, "Joint 2D-3D-Semantic Data for Indoor Scene Understanding," *ArXiv e-prints*, Feb. 2017.
- [29] A. X. Chang, T. Funkhouser, L. Guibas, P. Hanrahan, Q. Huang, Z. Li, S. Savarese, M. Savva, S. Song, H. Su, J. Xiao, L. Yi, and F. Yu, "Shapenet: An information-rich 3d model repository," 2015.
- [30] S. Amirreza, B. Shray, L. Zhen, E. Irfan, and B. Byron, "One-shot learning for semantic segmentation," in *Proceedings of the British Machine Vision Conference (BMVC)*, G. B. Tae-Kyun Kim, Stefanos Zafeiriou and K. Mikolajczyk, Eds. BMVA Press, September 2017, pp. 167.1–167.13.
- [31] H. Lee, J. Lee, H. Kim, and D. Mun, "Dataset and method for deep learning-based reconstruction of 3d cad models containing machining features for mechanical parts," *Journal of Computational Design and Engineering*, vol. 9, no. 1, pp. 114–127, 2022.