Generalized Few-Shot Point Cloud Segmentation Via Geometric Words

Yating Xu[1] Conghui Hu[1] Na Zhao[2*] Gim Hee Lee[1]
[1]Department of Computer Science, National University of Singapore
[2]Singapore University of Technology and Design
xu.yating@u.nus.edu conghui@nus.edu.sg na_zhao@sutd.edu.sg gimhee.lee@nus.edu.sg

Abstract

Existing fully-supervised point cloud segmentation methods suffer in the dynamic testing environment with emerging new classes. Few-shot point cloud segmentation algorithms address this problem by learning to adapt to new classes at the sacrifice of segmentation accuracy for the base classes, which severely impedes its practicality. This largely motivates us to present the first attempt at a more practical paradigm of generalized few-shot point cloud segmentation, which requires the model to generalize to new categories with only a few support point clouds and simultaneously retain the capability to segment base classes. We propose the geometric words to represent geometric components shared between the base and novel classes, and incorporate them into a novel geometric-aware semantic representation to facilitate better generalization to the new classes without forgetting the old ones. Moreover, we introduce geometric prototypes to guide the segmentation with geometric prior knowledge. Extensive experiments on S3DIS and ScanNet consistently illustrate the superior performance of our method over baseline methods. Our code is available at: https://github.com/Pixie8888/GFS-3DSeg_GWs.

1. Introduction

Point cloud segmentation aims at predicting the category of each point in the 3D scenes represented by the point cloud, and has wide applications in autonomous driving, robotics, *etc*. Although fully-supervised point cloud segmentation methods (Full-3DSeg) [19, 20, 28, 11, 9, 29] have achieved impressive performance, they heavily require large-scale annotated training data and rely on the closed set assumption that the class distribution of the testing point cloud remains the same as the training dataset. However, the closed set assumption is unrealistic in the open world where new classes arise continuously. The Full-3DSeg in this challenging open-world setting thus requires large amounts of annotated data for new classes, which are time-consuming and expensive to collect.

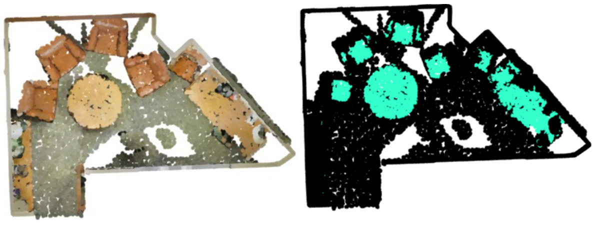


Figure 1. **Visualization of the geometric words (GWs) on S3DIS**. The left figure shows the original point cloud and the right figure shows the activated point cloud to a geometric word. The activated points are colored green. It shows the horizontal planes of the sofa, table and chair are all activated to the same GW due to similar geometric structure.

The few-shot point cloud segmentation (FS-3DSeg) [35, 14, 8] algorithms are designed to ameliorate the lack of data for novel class adaptation. Generally, FS-3DSeg first trains a model with abundant training samples of the base classes, and then targets at segmenting the new classes by learning from only a small number of samples from the corresponding new classes. By adopting episodic training [26] to mimic the testing environment and specific designs for feature extraction [35], FS-3DSeg achieves promising novel class segmentation results for the query point clouds. However, in the task of point cloud segmentation, base and novel classes often appear together in one scene (see Figure 5 as examples). An ideal segmentor in practice is expected to give each point in the scene a semantic label. As a result, the FS-3DSeg setting that only segments points of novel classes while ignoring the base classes suffers limited practicality.

In view of the impracticality of FS-3DSeg, we introduce the generalized few-shot point cloud segmentation (GFS-3DSeg) task. As shown in Tab. 1, given the model originally trained on base classes, the objective of GFS-3DSeg is to segment both base and novel classes using merely a limited

[*] Na Zhao was concurrently a visiting professor at the National University of Singapore when this work was done.

number of labeled samples for the new classes during testing. Furthermore, there is no access to the base training data during testing in GFS-3DSeg considering the issues of data privacy and storage memory limitations. In this demanding but practical context, we expect a good generalized few-shot point cloud segmentor to effectively learn to segment novel classes with few samples and also maintain the knowledge of the base classes. A potential solution to GFS-3DSeg can be using prototype learning [25, 21] to generate class prototypes of the base and novel classes as the classifier weights, which can quickly adapt to new classes and avoid forgetting past knowledge caused by fine-tuning. However, effective learning of the representation and classifier for the novel classes is not trivial and remains as a major challenge.

In this paper, we propose **geometric words**[1] (GWs) as the transferable knowledge obtained from the base classes to enhance the learning of the new classes without forgetting the old ones. Although different (old and new) classes contain distinct semantic representations, they usually share similar local geometric structures as shown in Fig. 1. Based on this observation, we first mine the representation for the local geometric structures from the pretrained features of the base classes and store them as the geometric words to facilitate learning of the new classes with few examples. We then learn a geometric-aware semantic representation based on the geometric words. Specifically, the geometric-aware semantic representation is a fusion of two features: 1) *Class-agnostic geometric feature* obtained by the assignment of low-level features to their most similar geometric words. 2) *Class-specific semantic feature* which is the output of a feature extractor. Intuitively, our geometric-aware semantic representation allows the encoding of the transferable geometric information across classes while preserving the semantic information for effective segmentation.

We further introduce **geometric prototype** (GP) to supplement the original semantic prototype in the prototype learning. Particularly, the geometric prototype refers to the frequency histogram of GWs that can uniquely represent each class from the global geometric perspective despite GWs are class agnostic. We thus leverage GP to propose a geometric-guided classifier re-weighting module for the rectification of biased predictions originating from semantic prototypes. Specifically, we first perform minor frequency pruning on the geometric prototypes to suppress noisy responses of the geometric words for each class. Subsequently, we measure the geometric matching score between each query point and the pruned geometric prototypes, and employ these scores as geometric-guided weights. By re-weighting the semantic logits with geometric-guided weights, our final prediction is enriched with geometric information that is transferable across classes. Such transferable information helps to facilitate the segmentation of new classes that only have a few samples while preserving the knowledge of base classes. Our main contributions can be summarized as:

Setting	Training Stage	Testing Stage		
	Learn D^b_{train}	Learn D^n_{train}	Access D^b_{train}	Test Classes
Full-3DSeg	✓	✗	✗	C^b
FS-3DSeg	✓	✓	✗	C^n
GFS-3DSeg	✓	✓	✗	$C^b \cup C^n$

Table 1. **The comparison of different settings**. D^b_{train} and D^n_{train} denotes the training dataset of base and novel classes, respectively. D^b_{train} has sufficient training data, while D^n_{train} only has few shots for each new class. C^b and C^n denotes the label set for base classes and novel classes, respectively.

- We are the first to study the important generalized few-shot point cloud segmentation task, which is more practical than its counterparts of fully-supervised and few-shot setting in the dynamic testing environment.
- We propose geometric words to represent diverse basic geometric structures that are shared across different classes, and a geometric-aware semantic representation that allows for generalizable knowledge encoding.
- We introduce geometric prototype to supplement the semantic prototype. We design a geometric-guided classifier re-weighting module comprising minor frequency pruning to dynamically guide the segmentation of novel classes with geometric similarity.
- We conduct extensive experiments on S3DIS and ScanNet to verify the effectiveness of our method. Specifically, our method improves over the state-of-the-art FS-3DSeg method by 6 and 8 times on the 5-shot and 1-shot settings of ScanNet, respectively.

2. Related Work

Few-shot Learning. Few-shot learning aims at classification of novel classes by learning with only a few labeled samples during testing stage. There are mainly three types of approaches: meta-learning [5, 15, 16, 10], metric-learning [23, 26, 32, 18] and transfer learning based methods [24, 13]. Meta-learning based methods aim to predict a set of parameters that can be quickly adapted to new tasks. Metric-learning methods learn a mapping from images to an embedding space where images of the same class are clustered together. Transfer learning methods aim to learn a general feature representation during base class training, and then train a classifier for each new task.

Generalized few-shot learning is first proposed in [7], which aims to classify both base and novel classes after learning from the few-shot training samples of novel classes during testing stage. Gidaris *et al*. [6] and Qi *et al*. [21] introduce cosine similarity between image feature and clas-

[1] Analogous to visual words in bag-of-words image retrieval systems [17].

sifier weight as the classifier, which unifies the recognition of both novel and base categories. To reduce the biased learning of base classes, Gidaris *et al.* [6] and Ye *et al.* [31] improve the classifier of the novel classes by utilizing the knowledge of base classes. Gidaris *et al.* [6] propose few-shot classification weight generator to perform attention over the classification weight vectors of the base classes. Ye *et al.* [31] synthesize calibrated few-shot classifiers with a shared neural dictionary learned in the base class training stage. In this work, we study generalized few-shot point cloud segmentation, a more challenging task as it targets at dense 3D point-level classification. Instead of fusion classifier weights of novel classes with past knowledge [6, 31], we use geometric prototype as an additional classifier weight to help calibrate the biased prediction. Moreover, we propose geometric-aware semantic representation to facilitate the representation learning for novel classes.

Few-shot Semantic Segmentation. Few-shot semantic segmentation performs segmentation of novel classes for images [12, 34, 33, 22] or point clouds [35, 8, 14] by only learning from a few support samples. The methods for few-shot image semantic segmentation can be categorized into metric-based [27, 12] and relation-based [34, 33] methods. Metric-based methods aggregate prototypes from the support set as the classifier, and perform cosine similarity with the query features. Relation-based methods concatenate the support features with query features for dense feature comparison via deep convolutional network. Few-shot point cloud segmentation [35] also adopts metric-based technique by performing label propagation among the query points and the multi-prototypes of each class to infer the query point label.

Generalized few-shot image semantic segmentation (GFS-2DSeg) [25, 2] aims to segment both base and novel classes in the query images during testing stage, which is more practical than the few-shot setting. CAPL [25] leverages the contextual cues from the support set and query images to enhance the classifier weights of the base classes. PIFS [2] finetunes on the support set of novel classes during testing stage and proposes prototype-based distillation loss to combat catastrophic forgetting of base classes. In this paper, we study generalized few-shot point cloud semantic segmentation (GFS-3DSeg), a practical yet unexplored task. Different from GFS-2DSeg which assumes the annotation of base classes is available in the novel training samples, we strictly follow the definition of GFSL that only the annotation of novel classes are provided in the support set. Consequently, both CAPL and PIFS fail to work properly under the GFS-3DSeg since both rely on the cooccurence of base classes in the support sets of novel classes to calibrate the imbalanced learning between base and novel classes. To solve this challenging problem, we propose to mine the representation for the basic geometric structures as the transferable knowledge to improve the representation and classifier of the novel classes.

Geometric Primitives Geometric primitives are the fundamental components for the 3D objects, and has been initially studied in the transfer learning in 3D [3, 36]. Chowdhury *et al.* [3] use microshapes as the basic geometric components to describe any 3D objects in the 3D object recognition task. However, object-level annotation is not available for the query point cloud in the point cloud semantic segmentation or object detection. Therefore, Zhao *et al.* [36] only utilize the geometric information at point-level by enhancing the local geometric representation of each query point with a geometric memory bank. Although we also performs point-level enhancement in our geometric-aware semantic representation, different from Zhao *et al.* [36], we inject geometric information to the high-level semantic representation to help model understand the geometric structures through learning the class semantics. Moreover, we propose to use the geometric prototype as the global geometric enhancement to each class to calibrate the biased prediction during testing stage.

3. Problem Formulation

In generalized few-shot point cloud semantic segmentation, a base class training dataset D^b_{train}, and a novel class training dataset D^n_{train} with non-overlapping label space $C^b \cap C^n = \varnothing$ are provided. The testing dataset D_{test} has a label space $C_{\text{test}} = C^b \cup C^n$. During the training stage, the model learns the base classes C^b from abundance of labeled point cloud data $D^b_{\text{train}} = \left\{ \left(P^b_k, M^b_k\right)^{\left|D^b_{\text{train}}\right|}_{k=1} \right\}$, where $\left|D^b_{\text{train}}\right|$ denotes the size of $D^{\text{b}}_{\text{train}}$. Each point cloud $P^b_k \in \mathbb{R}^{m \times d_0}$ contains m points with feature dimension d_0 and M^b_k denotes the annotation of C^b in P^b_k.

During the testing stage, the model first learns C^n from limited labeled data $D^n_{\text{train}} = \left\{ \left(P^{n,i}_k, M^{n,i}_k\right)^K_{k=1} \right\}^{|C^n|}_{i=1}$, with K support point clouds per novel class $C^{n,i} \in C^n$, and $|C^n|$ is the number of novel classes. $P^{n,i}_k \in \mathbb{R}^{m \times d_0}$ and $M^{n,i}_k$ is the binary mask indicating the presence of $C^{n,i}$. Note that during testing stage, the model does not have access to D^b_{train}. The testing dataset $D_{\text{test}} = \left\{ (P^q_k, M^q_k)^{|D_{\text{test}}|}_{k=1} \right\}$, with each query point cloud $P^q_k \in \mathbb{R}^{m \times d_0}$. M^q_k represents the ground-truth annotation of C_{test} in P^q_k. The goal of GFS-3DSeg is to correctly segment both C^b and C^n in the testing query point cloud.

4. Our Method

Background. We adopt prototype learning to segment both base and novel classes during testing stage. Class prototypes are learned as the classifier weight for each class.

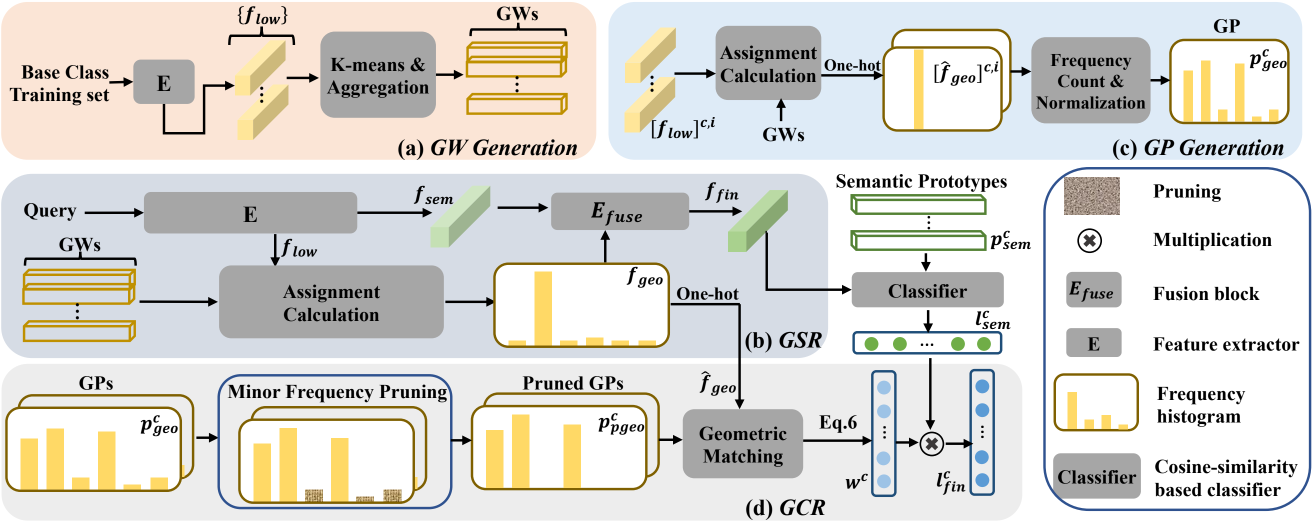


Figure 2. **The overview of our proposed framework**. **(a) GW Generation**: shows the generation of geometric words from base class training data. **(b) Geometric-aware Semantic representation (GSR)**: semantic feature f_{sem} is fused with geometric feature f_{geo} as the final representation f_{fin} for each point. **(c) GP Generation**: shows the generation of the geometric prototype p^c_{geo} for class c. **(d) Geometric-guided Classifier Re-weighting (GCR)**: we compute geometric matching between geometric feature $\hat{f}_{\text{geo}}$ and pruned GP p^c_{pgeo} to find potential target classes and derive weight w^c to supplement semantic prediction l^c_{sem}. l^c_{fin} is the final prediction logit of class c.

Base class prototypes are learned in the training stage via gradient descent, and novel class prototypes are learned by aggregating the foreground features of the support set during the testing stage. We name the prototype "semantic prototype" since it captures the semantic information of each class. The prediction of each query point is assigned by the class label of the most similar semantic prototypes. However, naively adopting prototype learning is insufficient to learn well on new classes due to the small support set. We thus propose geometric-aware semantic representation and geometric-guided classifier re-weighting to help segmenting new classes.

Framework Overview. Fig. 2 shows the overview our framework, which consists of four main parts: a) The **geometric words (GWs)** to enhance the representation and classifier of the new classes. b) The **geometric-aware semantic representation (GSR)** based on the GWs to learn a transferable representation during the base class training stage. we first get the geometric feature of a point as the assignment of the GWs to the point. We then fuse the geometric feature with its corresponding semantic feature to get the final GSR. c) The **geometric prototype (GP)** to supplement the semantic prototype learned from the insufficient training samples. Specifically, a GP is the frequency histogram of the GWs assigned to the points in each class that can uniquely describe the class from the global geometric perspective. d) The **geometric-guided classifier re-weighting (GCR)** based on GPs to provide prior knowledge of each query point belonging to the potential target classes based on the geometric similarity.

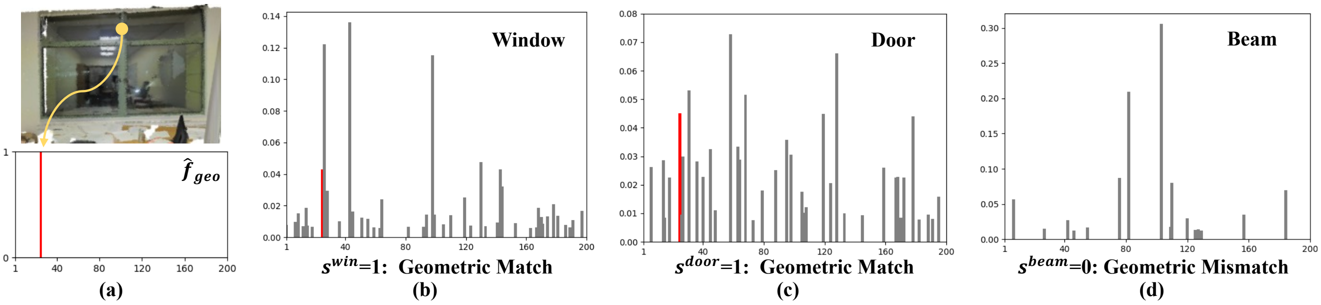
4.1. Geometric Words

Unlike its 2D images counterpart, 3D point clouds contain complete geometric information with shared basic geometric components. Understanding these basic geometric components helps learning across old and new classes due to the shared similar local geometric structures. We thus propose geometric words as the representation of these basic geometric components, and utilize it during training and testing stage to facilitate learning of new classes from few shots training point clouds.

To obtain the geometric words, we pretrain the feature extractor E of attMPTI [35] on D^b_{train} and collect the features $\{f_{\text{low}}\}$ of all the points belonging to C^b. We concatenate the output feature of the first three EdgeConv layers and denote it as $f_{\text{low}} \in \mathbb{R}^{d_1}$ since lower level features contains more geometric cues. We then obtain H geometric words $\mathcal{G} = \{g_h\}^H_{h=1} \in \mathbb{R}^{H \times d_1}$ by applying K-means on $\{f_{\text{low}}\}$ to calculate the H centroids. Each g_h is a local aggregation of the points with similar f_{low}, *i.e.* similar geometric characteristic.

Fig. 1 visualizes the point-to-GWs assignments by searching points with f_{low} that are most similar to a given geometric word g_h. As shown in Fig. 1, the horizontal plane of the chair, tables and sofas are all activated to the same geometric word. It suggests that our GWs are able to represent shared geometric components among different classes.

Geometric-aware Semantic Representation. Based on the GWs, we propose the geometric-aware semantic representation (GSR) to enhance the feature representation of the new classes. The GSR is a fusion of a class-agnostic



s^{win}=1: Geometric Match

(a) (b)

s^{door}=1: Geometric Match

(c)

s^{beam}=0: Geometric Mismatch

(d)

Figure 3. Motivation for geometric-guided classifier re-weighting. For each histogram, the horizontal axis represents the index of the GWs and the vertical axis represents the normalized frequency ratio within the class or point. (a) Visualize the geometric feature $\hat{f}_{\text{geo}}$ of the yellow query point on the window frame. (b),(c) and (d) shows the geometric prototypes of window, door and beam, respectively. The red bar denotes the GW with same index.

geometric feature $f_{\text{geo}} \in \mathbb{R}^H$ and a class-specific semantic feature $f_{\text{sem}} \in \mathbb{R}^{d_2}$. Specifically, the geometric feature f_{geo} that represents the geometric information of each point is computed by the soft-assignment of its feature f_{low} to its most similar GW as follow:

$$f_{\text{geo}} = \text{Softmax}\left([f_{\text{low}} \cdot g_1, \ldots, f_{\text{low}} \cdot g_H; \tau]\right), \quad (1)$$

where $\cdot$ denotes the cosine similarity between the feature and a geometric word. τ is the temperature to sharpen the probability vector, and we empirically set it to 10. We adopt soft assignment to make it differentiable with respect to the feature extractor. We use the final output of E as the semantic feature f_{sem}. Subsequently, f_{geo} and f_{sem} are concatenated and sent into a small convolution block E_{fuse} to obtain the final representation $f_{\text{final}} \in \mathbb{R}^{d_3}$ for each point as follow:

$$f_{\text{fin}} = E_{\text{fuse}}\left(f_{\text{geo}} \parallel f_{\text{sem}}\right), \quad (2)$$

where $\parallel$ represents the concatenation of two vectors.

During base class training, we simulate query and fake novel class support set in each batch following [25] to enhance the model's adaptability to unseen environments. The optimization objective is to minimize cross-entropy loss computed by the prototypes generated through the assembling of $\left\{p_{\text{sem}}^c | c \in C^b\right\}$ and the fake novel prototypes from the simulated support set. We refer readers to [25] for a more comprehensive understanding of the training strategy.

4.2. Geometric prototype

Although GWs are class-agnostic, their combinations are able to represent different classes in a geometric way. We visualize the frequency of GWs assigned to the points in different classes in Fig. 3(b), (c) and (d). The horizontal axis represents the index of the GWs and the vertical axis represents the normalized frequency ratio. The histogram conveys the global structure of each class via the frequency ratios of the GWs, and different classes have different histograms. The histogram thus uniquely represents its corresponding class and we refer to it as geometric prototype $p_{\text{geo}}^c \in \mathbb{R}^H$:

$$p_{\text{geo}}^c = \frac{\sum_{i=1}^{N^c} \left[\hat{f}_{\text{geo}}\right]^{c,i}}{N^c}, \quad (3)$$

where N^c denotes the number of points belonging to class c in the training dataset D_{train}^b or D_{train}^n. $\hat{f}_{\text{geo}} \in \mathbb{R}^H$ is the hard assignment in the form of one-hot vector. We augment the semantic prototype p_{sem}^c with the geometric prototype p_{geo}^c, as the semantic prototype primarily encodes semantic information and, as a result, becomes insufficient in representing the new classes due to limited training samples.

Geometric-guided Classifier Re-weighting. Based on the GP, we propose the geometric-guided classifier re-weighting module to help the prediction of the novel classes. As shown in Fig. 3, the corresponding geometric word for a point on the window frame is activated in the GP of window and the geometrically similar class door, but suppressed in beam which does not have the frame structure. This implies that comparing the geometric feature of the query point with GP can be employed as a hint for segmentation. Therefore, we compute a geometric matching score s^c based on the cosine similarity between a query point's geometric feature and GP as:

$$s^c = \mathbb{1}\left[p_{\text{geo}}^c \cdot \hat{f}_{\text{geo}}\right] = \begin{cases} 1 & p_{\text{geo}}^c \cdot \hat{f}_{\text{geo}} > 0 \\ 0 & \text{otherwise} \end{cases}, \quad (4)$$

where $\mathbb{1}[\cdot]$ is an indicator fucntion, and $s^c = 1$ indicates the query point has the same geometric structure as class c. c is the class name.

However, the geometric matching may be negatively influenced by the noisy GWs in the p_{geo}^c due to the scene context. As shown in Fig. 4, although points on the wall (in red) are on the vertical plane, they are still activated to the same GW as the points on the horizontal table plane due to adjacency. To suppress these noisy GWs in p_{geo}^c and improve the accuracy of geometric matching, we propose minor frequency pruning as shown in Algorithm 1. Our motivation is that the GWs representing the typical geometric structure of the class usually contributes a large frequency on the

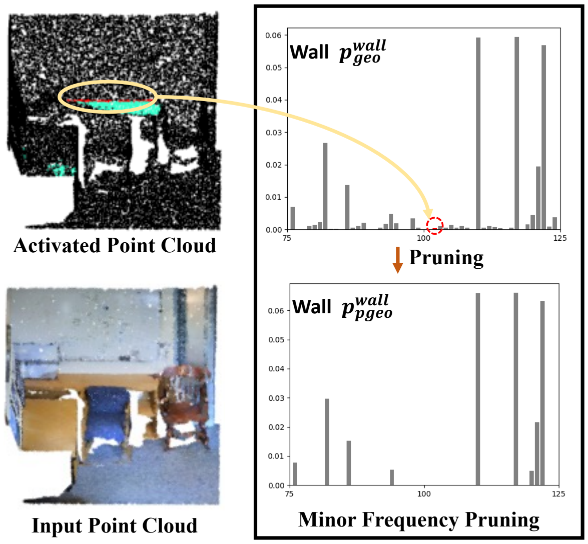


Figure 4. Illustration of minor frequency pruning. The top left figure shows the green points on the table are activated to the GW representing the horizontal plane, while the red points on the vertical plane of the wall are wrongly activated to the same GW due to scene context. The right figure shows the proposed minor frequency pruning to suppress the activation to the wrong GWs introduced by scene context.

histogram, while the GWs that are introduced by the scene context have relatively low frequencies. Consequently, we remove GWs corresponding to lower frequencies to only preserve the representative geometric structures. We denote the pruned geometric prototype as p^c_{pgeo}. The frequency limit α in Algorithm 1 denotes the amount of the frequencies to keep in the original p^c_{geo}, and the $p^{c,j}_{\text{geo}}$ denotes the j-th entry of p^c_{geo}. The computation of the matching score is then updated as:

$$s^c = \mathbb{1}\left[p^c_{\text{pgeo}} \cdot \hat{f}_{\text{geo}}\right], \tag{5}$$

To highlight the geometrically matched class in the prediction, we set a weight w^c to the prediction of each class according to the matching score as follow:

$$w^c := \begin{cases} \beta & s^c = 1 \\ 1 & s^c = 0 \end{cases}. \tag{6}$$

We set $\beta > 1$ to highlight the potential target classes of the query point. We then re-weight the semantic classification logit l^c_{sem} with w^c to compute the final prediction logit l^c_{fin} as follows:

$$l^c_{\text{fin}} = w^c \times l^c_{\text{sem}}, \qquad l^c_{\text{sem}} = p^c_{\text{sem}} \cdot f_{\text{fin}}. \tag{7}$$

The l^c_{fin} considers both semantic and geometric similarity when segmenting new classes, which is more reliable than using semantic prediction logit l^c_{sem} alone as shown in

Algorithm 1 Minor Frequency Pruning

```
Input: Geometric prototype $p^c_{\text{geo}}$, frequency limit $\alpha \in [0,1]$
Output: Pruned geometric prototype $p^c_{\text{pgeo}}$
$\gamma \leftarrow 0$   ▷ $\gamma$ records the accumulated frequencies and is initialized to 0
$p^c_{\text{pgeo}} \leftarrow [0, ..., 0]^H$   ▷ Initialize all entries of $p^c_{\text{pgeo}}$ to 0
$\{\text{idx}_1, ..., \text{idx}_H\} \leftarrow \text{Sort\_Descending}\left\{p^{c,j}_{\text{geo}}\right\}_{j=1}^{H}$   ▷ Get indices of $\left\{p^{c,j}_{\text{geo}}\right\}_{j=1}^{H}$ in descending order
$\text{idx}_i \leftarrow \text{idx}_1$
while $\gamma < \alpha$ do
    $p^{c,\text{idx}_i}_{\text{pgeo}} \leftarrow p^{c,\text{idx}_i}_{\text{geo}}$
    $\gamma \leftarrow \gamma + p^{c,\text{idx}_i}_{\text{geo}}$
    $\text{idx}_i \leftarrow \text{idx}_{i+1}$
end while
$p^c_{\text{pgeo}} \leftarrow p^c_{\text{pgeo}} / \text{Sum}\left(p^c_{\text{pgeo}}\right)$   ▷ Normalization
```

Tab. 4. Finally, we predict the label y for each query point as follow:

$$y = \text{argmax}\left(\text{Softmax}\left(\left[l^1_{\text{fin}}, ..., l^{|C_{\text{test}}|}_{\text{fin}}\right]; \tau\right)\right). \tag{8}$$

5. Experiments

5.1. Datasets and Setup

Datasets. We evaluate on two datasets: 1) S3DIS [1] consists 272 point clouds from six areas with annotation corresponding to 13 semantic classes. We use area 6 as the testing dataset D_{test}, and leverage the other five areas to construct the training dataset for base and novel classes. 2) ScanNet [4] consists of 1,513 point clouds with annotation corresponding to 20 semantic classes. We use 1,201 point clouds to construct training dataset of D^b_{train} and D^n_{train} and the rest 312 point clouds to construct D_{test}.

For both datasets, we choose the last 6 classes with least labeled points in the corresponding dataset as the novel classes C^n and the rest classes as base classes C^b. The motivation is to simulate the scenario in real world, where the frequency of novel class occurring is low and it is hard to collect sufficient training data. Consequently, the novel classes for S3DIS are table, window, column, beam, board and sofa. The novel classes for ScanNet are sink, toilet, bathtub, shower curtain, picture and counter. We follow the data pre-processing of [35] to divide each point cloud into blocks with size of 1 meter × 1 meter on the xy plane. From each block, we sample $m = 2,048$ points as input. The dimension d_0 for the input feature is 9 with XYZ, RGB and normalized XYZ to the block.

Evaluation Metrics. We evaluate the performance of model using mean intersection-over-union (mIoU). We use

Method	5-shot				1-shot			
	mIoU-B	mIoU-N	mIoU-A	HM	mIoU-B	mIoU-N	mIoU-A	HM
Fully Supervised	76.51	58.69	68.29	66.42	76.51	58.69	68.29	66.42
attMPTI [35]	34.90	16.08	26.21	21.99	21.89	11.39	17.05	14.95
PIFS [2]	56.99	19.66	39.76	29.23	57.85	14.59	37.88	23.31
CAPL [25]	73.56	35.18	55.85	47.51	72.80	23.87	50.22	35.67
Ours	**73.61**	**43.26**	**59.60**	**54.42**	**74.10**	**29.66**	**53.58**	**41.92**

Table 2. Results on **S3DIS** under 5-shot and 1-shot settings.

Method	5-shot				1-shot			
	mIoU-B	mIoU-N	mIoU-A	HM	mIoU-B	mIoU-N	mIoU-A	HM
Fully Supervised	43.12	37.04	41.34	39.85	43.12	37.04	41.34	39.85
attMPTI [35]	16.31	3.12	12.35	5.21	12.97	1.62	9.57	2.88
PIFS [2]	35.14	3.21	25.56	5.88	35.80	2.54	25.82	4.75
CAPL [25]	38.22	14.39	31.07	20.88	38.70	10.59	30.27	16.53
Ours	**40.18**	**18.58**	**33.70**	**25.39**	**40.06**	**14.78**	**32.47**	**21.55**

Table 3. Results on **ScanNet** under 5-shot and 1-shot settings.

mIoU-B, mIoU-N and mIoU-A to denote the mIoU on the base classes, novel classes, and all the classes, respectively. In addition, we use harmonic mean of mIoU-B and mIoU-N to better describe the overall performance on base and novel classes, *i.e.*, HM $= \frac{2\times\text{mIoU-B}\times\text{mIoU-N}}{\text{mIoU-B}+\text{mIoU-N}}$. In comparison to mIoU-A, HM is not biased towards the base classes [31].

5.2. Implementation details

We adopt the feature extractor of [35] as E and pretrain it on the D^b_{train} for 100 epochs. We then perform K-means on the collection of the base class features $\{f_{\text{low}}\}$. We set H as 200 for S3DIS and 180 for ScanNet, respectively. During base class training, we perform geometric-aware semantic representation learning and learn semantic prototypes for base classes $\left\{p^c_{\text{sem}} \mid c \in C^b\right\}$. d_1 and d_2 are both 192 following attMPTI [35], and d_3 is set to 128. We set batch size to 32 and train for 150 epochs. We use Adam optimizer with initial learning rate of 0.01 and decayed by 0.5 every 50 epochs. We load the pretrained weight of the first three EdgeConv layers and set their learning rate to be 0.001. We compute the geometric prototypes for base classes $\left\{p^c_{\text{pgeo}} \mid c \in C^b\right\}$ after training completes. During testing stage, we first obtain the semantic prototypes $\{p^c_{\text{sem}} \mid c \in C^n\}$ and geometric prototypes $\left\{p^c_{\text{pgeo}} \mid c \in C^n\right\}$ for novel classes by averaging f_{fin} and $\hat{f}_{\text{geo}}$ (followed by minor frequency pruning) of the foreground points in the support set, respectively. We then predict the class labels for each query point via proposed GCR. α is set to 0.9 and 0.95 for S3DIS and ScanNet, respectively. β is set to 1.2.

5.3. Baselines

We design three baselines for comparison with our method. 1) **attMPTI** [35] is the state-of-the-art FS-3DSeg method. We follow the original implementation in [35] and episodically train attMPTI on base class dataset. Upon finishing training, we collect multi-prototypes for base classes. During testing stage, we first generate multi-prototypes for novel classes from D^n_{train}. We then estimate the query label by performing label propagation among query points and prototypes of base and novel classes. 2) **PIFS** [2] is the state-of-the-art method for GFS-2DSeg that finetunes on D^n_{train} to learn novel classes. We apply their proposed prototype-based distillation loss to only the scores of novel classes since we do not provide the annotation of base classes in D^n_{train}. 3) **CAPL** [25] is the state-of-the-art method in GFS-2DSeg that performs prototype learning to learn novel classes. We remove the SCE module of CAPL since the annotations of base classes in the D^n_{train} are not available. All the baselines use the same feature extractor with us for fair comparison.

In addition, we design an oracle setting, **Fully Supervised**, where the model is trained on the fully annotated dataset of base and novel classes using the same feature extractor with us and a small segmentation head.

5.4. Comparison with Baselines

Tab. 2 and Tab. 3 show the results of GFS-3DSeg on S3DIS and ScanNet, respectively. We conduct experiments in two settings with the number of support point clouds $K = \{1, 5\}$ on each dataset. We randomly generate 5 sets of D^n_{train} using different seeds for each setting and calculate the averaged results over all 5 sets to obtain a more reliable results. It is clear to see that the segmentation accuracy of novel classes increases with more number of shots. Compared with all the baselines, our method is able to utilize the limited number of training samples from D^n_{train} in a more effective way and achieves much better performance in terms

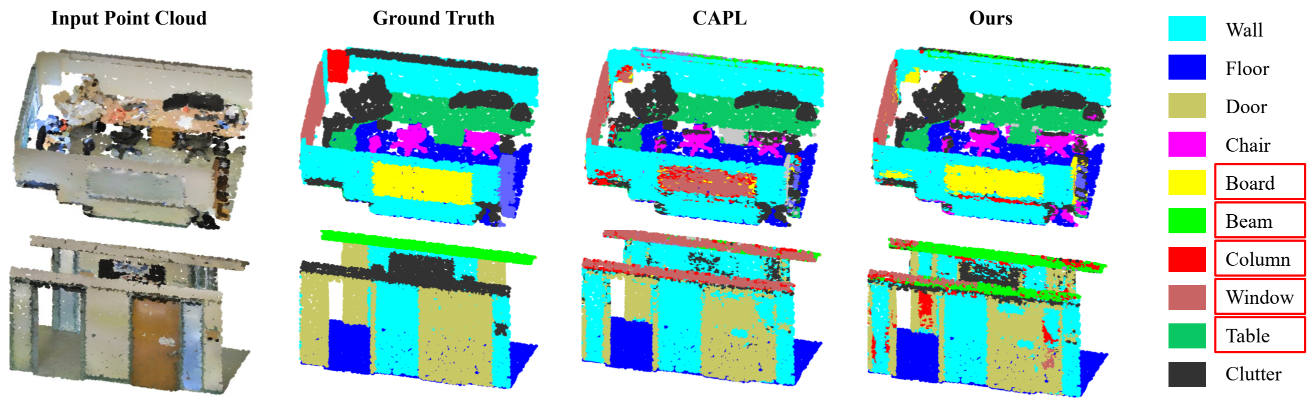


Figure 5. Qualitative comparison on 5-shot setting of S3DIS dataset. Novel classes are marked with red rectangle. The target novel classes in the first row are **board**, **table**, **column** and **window**. The target novel classes in the second row is **beam**.

GSR	GCR	mIoU-B	mIoU-N	mIoU-A	HM
✗	✗	73.56	35.18	55.85	47.51
✓	✗	73.48	40.10	58.07	51.81
✓	✓	**73.61**	**43.26**	**59.60**	**54.42**

Table 4. Effectiveness of geometric-aware semantic representation (GSR) and geometric-guided classifier re-weighting (GCR) on S3DIS.

Number of GWs	mIoU-B	mIoU-N	mIoU-A	HM
$H = 100$	72.45	**43.61**	59.14	54.38
$H = 150$	**73.88**	42.27	59.29	53.71
$H = 200$	73.61	43.26	**59.60**	**54.42**
$H = 250$	72.96	41.29	58.32	52.63
$H = 400$	73.19	40.84	58.26	52.36

Table 5. Ablation study of number of geometric words in S3DIS.

of the HM and mIoU-N.

attMPTI [35] fails to perform well on the GFS-3DSeg since it only focuses on establishing the decision boundary for those classes appearing in each episode. When including all the classes in the evaluation, the original decision boundary collapses. PIFS [2] also performs poorly on the GFS-3DSeg. The large intra-class variances of the 3D objects make novel class adaptation difficult for the 2D fine-tuning method. Moreover, their fine-tuning method leads to severe catastrophic forgetting of the base classes due to the absence of base class training data. Our method is based on CAPL [25]. Compared to CAPL which mainly utilizes the context information to enhance the semantic prototypes of the base classes, we focus on enhancing the learning of the new classes by leveraging transferable knowledge GWs. As a result, our method shows great superiority over CAPL in the performance of the novel classes. This can also be verified in Fig. 5, where our model is able to segment target new classes (board and table in the first row, and beam in the second row) more precisely.

Frequency Limit	mIoU-B	mIoU-N	mIoU-A	HM
$\alpha = 1$	73.50	41.05	58.50	52.59
$\alpha = 0.95$	73.48	43.19	59.50	54.34
$\alpha = 0.9$	**73.61**	**43.26**	**59.60**	**54.42**
$\alpha = 0.85$	73.59	43.01	59.48	54.22

Table 6. Ablation Study of frequency limit α. $\alpha = 1$ denotes model without minor frequency pruning in the GCR.

logits weight	mIoU-B	mIoU-N	mIoU-A	HM
$\beta = 1$	73.48	40.10	58.07	51.81
$\beta = 1.2$	**73.61**	**43.26**	**59.60**	**54.42**
$\beta = 1.5$	73.20	42.25	58.90	53.51

Table 7. Ablation Study of logits weight β. $\beta = 1$ is equivalent to the model without using GCR.

Method	mIoU-B	mIoU-N	mIoU-A	HM
PTv2+CAPL	86.35	27.58	59.22	41.68
PTv2+Ours	**86.38**	**37.26**	**63.71**	**52.01**

Table 8. Comparison using state-of-the-art feature extractor Point Transformer v2 (PTv2) [30].

5.5. Ablation Study

Effectiveness of GSR and GCR. We verify the effectiveness of the geometric-aware semantic representation (GSR) and geometric-guided classifier re-weighting (GCR) on 5-shot setting using S3DIS (Tab. 4).The model without GSR uses f_{sem} as the feature representation for each point. The model without GCR adopts l^c_{sem} as the final prediction logit of class c. Both GSR and GCR are beneficial to new class segmentation, which suggests the successful enhancement of the representation and classifier of the novel classes by using transferable geometric information.

Analysis of the number of GWs. Tab. 5 studies the influence of GW numbers on the performance of 5-shot setting using S3DIS. Although the performance varies with the different numbers of geometric words, they all largely outperform the baselines without GWs. We choose $H = 200$ in our model regarding its best overall performance.

Method	ceiling	floor	wall	beam	column	window	door	table	chair	sofa	bookcase	board	clutter
CAPL	**93.17**	**97.36**	**72.92**	52.44	28.82	35.12	**77.57**	60.60	61.29	10.38	54.42	23.75	58.19
Ours	92.34	97.16	70.73	**60.37**	**32.16**	**46.30**	76.21	**64.41**	**63.42**	**16.99**	**54.77**	**39.30**	**60.66**

Table 9. Per-class IoU. Red denotes new classes.

Method	mIoU-B	mIoU-N	mIoU-A	HM
CAPL	0.17	3.25	1.46	2.94
Ours	0.35	3.34	1.63	2.75

Table 10. Standard deviation.

Effectiveness of the minor frequency pruning. In Tab. 6, we analyze the effect of the minor frequency pruning in the geometric prototype. GP without pruning ($\alpha = 1$) shows worst performance compared to the pruned GPs. $\alpha = 0.9$ gives the best performance, thus we adopt $\alpha = 0.9$ in our final model.

Analysis of the weight β. We present the ablation study of the weight β in Tab. 7. $\beta = 1$ does not highlight any potential target classes predicted by geometric matching, which is equivalent to the model without GCR. Setting a moderate threshold > 1 is helpful to improve the segmentation performance, so we choose $\beta = 1.2$ for model testing.

Using SOTA feature extractor. Tab. 8 shows the result of replacing DGCNN-based feature extractor [35] with Point Transformer v2 (PTv2) under the 5-shot setting of S3DIS. Our method outperforms the strongest baseline CAPL by a large margin on novel classes and overall performance. It verifies that our method can work successfully with a state-of-the-art point cloud feature extractor. We notice that the mIoU-N in Tab. 8 is lower than that of Tab. 2. One possible reason is that the feature extractor in [35] is specially designed to be able to quickly learn new classes from a small support set.

Standard deviation and per-class IoU. Tab. 10 shows the standard deviation results on the 5 testing sets of the 5-shot setting of S3DIS. Our model shows similar variation with CAPL. Tab. 9 shows per-class IoU of the 5-shot setting of S3DIS. Our method largely outperforms CAPL for all new classes while maintaining on-par performance on base classes.

6. Conclusion

In this paper, we present the unexplored yet important generalized few-shot point cloud segmentation. We address the challenge of facilitating new class segmentation with limited training samples by utilizing transferable knowledge geometric words (GWs) mined from the base classes. We propose geometric-aware semantic representation to learn generalizable representation where geometric features described through GWs are fused with semantic representation. We further propose the geometric prototype (GP) to supplement the semantic prototype in the testing stage. Extensive experiments on two benchmark datasets demonstrate the superiority of our method.

Acknowledgement. This research work is fully done at the National University of Singapore and is supported by the Agency for Science, Technology and Research (A*STAR) under its MTC Programmatic Funds (Grant No. M23L7b0021).

References

[1] Iro Armeni, Ozan Sener, Amir R Zamir, Helen Jiang, Ioannis Brilakis, Martin Fischer, and Silvio Savarese. 3d semantic parsing of large-scale indoor spaces. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1534–1543, 2016.

[2] Fabio Cermelli, Massimiliano Mancini, Yongqin Xian, Zeynep Akata, and Barbara Caputo. Prototype-based incremental few-shot semantic segmentation. *arXiv preprint arXiv:2012.01415*, 2020.

[3] Townim Chowdhury, Ali Cheraghian, Sameera Ramasinghe, Sahar Ahmadi, Morteza Saberi, and Shafin Rahman. Few-shot class-incremental learning for 3d point cloud objects. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XX*, pages 204–220. Springer, 2022.

[4] Angela Dai, Angel X Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. Scannet: Richly-annotated 3d reconstructions of indoor scenes. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5828–5839, 2017.

[5] Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *International conference on machine learning*, pages 1126–1135. PMLR, 2017.

[6] Spyros Gidaris and Nikos Komodakis. Dynamic few-shot visual learning without forgetting. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4367–4375, 2018.

[7] Bharath Hariharan and Ross Girshick. Low-shot visual recognition by shrinking and hallucinating features. In *Proceedings of the IEEE international conference on computer vision*, pages 3018–3027, 2017.

[8] Lvlong Lai, Jian Chen, Chi Zhang, Zehong Zhang, Guosheng Lin, and Qingyao Wu. Tackling background ambiguities in multi-class few-shot point cloud semantic segmentation. *Knowledge-Based Systems*, 253:109508, 2022.

[9] Xin Lai, Jianhui Liu, Li Jiang, Liwei Wang, Hengshuang Zhao, Shu Liu, Xiaojuan Qi, and Jiaya Jia. Stratified transformer for 3d point cloud segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8500–8509, 2022.

[10] Kwonjoon Lee, Subhransu Maji, Avinash Ravichandran, and Stefano Soatto. Meta-learning with differentiable convex optimization. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10657–10665, 2019.

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

[12] Yongfei Liu, Xiangyi Zhang, Songyang Zhang, and Xuming He. Part-aware prototype network for few-shot semantic segmentation. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part IX 16*, pages 142–158. Springer, 2020.

[13] Puneet Mangla, Nupur Kumari, Abhishek Sinha, Mayank Singh, Balaji Krishnamurthy, and Vineeth N Balasubramanian. Charting the right manifold: Manifold mixup for few-shot learning. In *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, pages 2218–2227, 2020.

[14] Yongqiang Mao, Zonghao Guo, Xiaonan Lu, Zhiqiang Yuan, and Haowen Guo. Bidirectional feature globalization for few-shot semantic segmentation of 3d point cloud scenes. *arXiv preprint arXiv:2208.06671*, 2022.

[15] Tsendsuren Munkhdalai and Hong Yu. Meta networks. In *International conference on machine learning*, pages 2554–2563. PMLR, 2017.

[16] Alex Nichol, Joshua Achiam, and John Schulman. On first-order meta-learning algorithms. *arXiv preprint arXiv:1803.02999*, 2018.

[17] David Nister and Henrik Stewenius. Scalable recognition with a vocabulary tree. In *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, volume 2, pages 2161–2168. Ieee, 2006.

[18] Boris Oreshkin, Pau Rodríguez López, and Alexandre Lacoste. Tadam: Task dependent adaptive metric for improved few-shot learning. *Advances in neural information processing systems*, 31, 2018.

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

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

[21] Hang Qi, Matthew Brown, and David G Lowe. Low-shot learning with imprinted weights. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5822–5830, 2018.

[22] Amirreza Shaban, Shray Bansal, Zhen Liu, Irfan Essa, and Byron Boots. One-shot learning for semantic segmentation. *arXiv preprint arXiv:1709.03410*, 2017.

[23] Jake Snell, Kevin Swersky, and Richard Zemel. Prototypical networks for few-shot learning. *Advances in neural information processing systems*, 30, 2017.

[24] Yonglong Tian, Yue Wang, Dilip Krishnan, Joshua B Tenenbaum, and Phillip Isola. Rethinking few-shot image classification: a good embedding is all you need? In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XIV 16*, pages 266–282. Springer, 2020.

[25] Zhuotao Tian, Xin Lai, Li Jiang, Shu Liu, Michelle Shu, Hengshuang Zhao, and Jiaya Jia. Generalized few-shot semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11563–11572, 2022.

[26] Oriol Vinyals, Charles Blundell, Timothy Lillicrap, Daan Wierstra, et al. Matching networks for one shot learning. *Advances in neural information processing systems*, 29, 2016.

[27] Kaixin Wang, Jun Hao Liew, Yingtian Zou, Daquan Zhou, and Jiashi Feng. Panet: Few-shot image semantic segmentation with prototype alignment. In *proceedings of the IEEE/CVF international conference on computer vision*, pages 9197–9206, 2019.

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

[29] Ziyi Wang, Yongming Rao, Xumin Yu, Jie Zhou, and Jiwen Lu. Semaffinet: Semantic-affine transformation for point cloud segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11819–11829, 2022.

[30] Xiaoyang Wu, Yixing Lao, Li Jiang, Xihui Liu, and Hengshuang Zhao. Point transformer v2: Grouped vector attention and partition-based pooling. *Advances in Neural Information Processing Systems*, 35:33330–33342, 2022.

[31] Han-Jia Ye, Hexiang Hu, and De-Chuan Zhan. Learning adaptive classifiers synthesis for generalized few-shot learning. *International Journal of Computer Vision*, 129:1930–1953, 2021.

[32] Han-Jia Ye, Hexiang Hu, De-Chuan Zhan, and Fei Sha. Few-shot learning via embedding adaptation with set-to-set functions. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8808–8817, 2020.

[33] Chi Zhang, Guosheng Lin, Fayao Liu, Jiushuang Guo, Qingyao Wu, and Rui Yao. Pyramid graph networks with connection attentions for region-based one-shot semantic segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9587–9595, 2019.

[34] Chi Zhang, Guosheng Lin, Fayao Liu, Rui Yao, and Chunhua Shen. Canet: Class-agnostic segmentation networks with iterative refinement and attentive few-shot learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5217–5226, 2019.

[35] Na Zhao, Tat-Seng Chua, and Gim Hee Lee. Few-shot 3d point cloud semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8873–8882, 2021.

[36] Shizhen Zhao and Xiaojuan Qi. Prototypical votenet for few-shot 3d point cloud object detection. *arXiv preprint arXiv:2210.05593*, 2022.