2024 IEEE International Conference on Robotics and Automation (ICRA 2024)
May 13-17, 2024. Yokohama, Japan

Noisy Few-shot 3D Point Cloud Scene Segmentation

Hao Huang†, Shuaihang Yuan†, CongCong Wen, Yu Hao, and Yi Fang

***Abstract*— 3D scene semantic segmentation plays a crucial role in robotics by enabling robots to understand and interpret their environment in a detailed and context-aware manner, facilitating tasks such as navigation, object manipulation, and interaction within complex spaces. A preponderance of methodology predominantly adopts a fully supervised framework for 3D point cloud scene semantic segmentation. Such paradigms exhibit an intrinsic dependency on extensive labeled datasets, presenting challenges in acquisition and exhibiting incapacity to segment novel classes, especially when the training data are contaminated by noisy samples. To address these limitations, this study introduces a novel few-shot segmentation approach to robustly segment 3D point cloud scenes with noisy labels using a meta-learning scheme. Specifically, we first build a multi-prototype graph and then suppress samples with noisy labels based on the graph structure. A subgraph bagging scheme is then proposed to conduct semi-supervised transductive learning to propagate labels. To optimize the graph structure to learn discriminative prototype features, we design a triplet contrastive loss to increase the compactness of these subgraphs. We evaluated our method on two widely used 3D point cloud scene segmentation benchmarks within few-shot (*i.e.*, 2/3-way 5-shot) segmentation settings with noisy samples. Experimental results demonstrate the improvement of our method over the compared baselines, illustrating the robustness of our method in few-shot 3D scene segmentation against noisy samples. The code is available at: `https://github.com/hhuang-code/Noisy_Fewshot_Segmentation`.**

I. Introduction

3D scene point cloud semantic segmentation stands as a cornerstone challenge not only in computer vision but also in robotics. The objective of 3D scene point cloud segmentation is to categorize each point in a scene into a corresponding semantic class, which can significantly enhance robotics' perception, enabling them to accurately identify and classify various elements within their operational environment. The significance of this task is underscored by a wide range of applications: from improving autonomous vehicle navigation [1], [2] to enabling more sophisticated interactions in robot-assisted tasks [3], [4], [5], despite the complexities introduced by the non-sequential and non-structured nature of point clouds. However, irregular point clouds present unique challenges. Their inherent non-sequential and non-structured nature adds layers of complexity to the segmentation process. The academic community has developed a variety of fully supervised 3D segmentation methods [6], [7], [8], [9], [10], [11]. Impressively, many of these approaches have shown promising results in multiple benchmark datasets [12], [13]. These methods, while often effective, come with a significant caveat: a heavy dependence on extensive labeled datasets. This data-intensive requirement has posed a palpable challenge, prompting researchers to seek alternative learning paradigms that could alleviate the burden of data acquisition.

Recently, the research landscape has witnessed a burgeoning interest in self-supervised [14], [15] and weakly supervised [16], [17], [18] 3D scene segmentation. These approaches, in essence, are geared towards mitigating the heavy reliance on large, labeled datasets, attempting to strike a balance between data dependency and performance efficiency. However, it is worth noting that while they offer certain advantages in terms of data economy, there are instances where their ability to generalize over previously unseen or novel categories is found to be lacking. In other words, these methods operate on a foundational assumption: training and evaluation datasets share an identical data distribution. Nevertheless, this premise often falls short in real-world scenarios. After training, unseen categories can appear, which throws a wrench into the existing segmentation approaches, particularly when against novel classes with only a sparse set of samples during training.

It is against this backdrop that *few-shot learning* [19], [20] has emerged as a compelling alternative. Rooted in the premise of making the most out of minimal data, few-shot learning allows models to accurately extrapolate from a limited set of examples, granting them the ability to classify previously unseen categories. When this paradigm is applied to the realm of 3D point cloud segmentation, the objective is to train a model to segment points from novel classes, all while operating on a paucity of labeled 3D scenes [21], [22], [23]. Few-shot 3D point cloud scene segmentation enables resource-efficient robots to rapidly recognize and understand novel objects or environments with minimal examples, thus significantly enhancing their adaptability and efficiency in completing manipulation and navigation tasks in dynamic settings [24], [25].

In this work, we adopt a meta-learning framework with an episodic training [26] scheme to tackle the few-shot 3D point cloud scene segmentation task. Specifically, our model is trained on a set of few-shot tasks each of which consists of a limited set of labeled samples, dubbed as *support set*, paired with the unlabeled counterparts, denoted as *query set*. Using the knowledge obtained from the support set, the model aims to accurately segment the query point clouds. However, as indicated in [27], the existing few-shot learning approaches predominantly operate under the assumption that the examples in the limited support set are meticulously annotated with the correct class labels. However, such idealized

† Hao Huang and Shuaihang Yuan equally contribute to this paper.

All Authors are with Center for Artificial Intelligence and Robotics (CAIR), New York University Abu Dhabi, UAE, and New York University, USA. hh1811@nyu.edu, sy2366@nyu.edu, cw3437@nyu.edu, yh3252@nyu.edu, yf23@nyu.edu

conditions are rarely met within real-world contexts, *e.g.*, even the datasets that have undergone careful annotation processes are not immune to the inclusion of erroneously labeled samples [28], [29]. Due to noise at the instance level during annotation, objects of other classes are incorrectly labeled as target classes and gathered in the support set [30] and such instances are regarded as *noisy* samples. Although methods designed for few-shot learning with noise in the 2D image domain [31], [27], 3D point cloud few-shot learning with noise has seldom been explored. This work aims to segment 3D scene point clouds against the interference introduced by these noisy samples similar to [31], [30], [27] in few-shot settings with an episodic training scheme.

II. Related Work

3D Semantic Segmentation. The existing 3D semantic segmetation approaches generally fit into one of three categories: projection-based, voxel-based, and point-based. Projection-based approaches [32], [33], [34] transform 3D point clouds into multi-view or spherical representations, making them suitable for 2D CNN feature extraction. Such transformations, however, may inadvertently distort the 3D topology and geometric relations. Voxel-based approaches [35], [36], [37], convert point clouds into dense grids, and then 3D CNN can effectively extract input features, ensuring that the intrinsic geometry of point clouds remains intact. Point-based methods, exemplified by the seminal PointNet [38], opt to work directly with raw unaltered point clouds. Successive works [6], [39], [40], [41], [9] focus on extracting detailed and discriminative local geometric features. Recent efforts [42], [11] adopt the prevalent Transformers and self-attention [43], [44], [45], to point cloud segmentation. With the rapid evolution in the field of 3D point cloud processing, Denoising Diffusion Probabilistic Model (DDPM) [46] has been extensively used for point cloud generation [47], [48], [49], [50]. Although DDPM is used mainly for shape generation, it can also be applied to segment point clouds through joint generation and segmentation training [51], [50].

Few-shot Learning. The goal of few-shot learning lies in learning a classifier capable of extrapolating to novel classes, even when presented with only a handful of training samples. To achieve this objective, a variety of meta-learning approaches [52], [26], [53], [54], [55] have been proposed. A noteworthy paradigm in this field is the metric-based approach [55], [26], [53], At the heart of which is a meticulously designed metric function crafted to generate a similarity-embedded space. This space, in turn, encapsulates the intricate ties that connect labeled samples to their unlabeled counterparts, creating a bridge between the known and novel data samples. The aforementioned approaches only focus on the 2D image domain, and there is also some literature aiming to learn 3D shape or scene representation under the few-shot settings for shape classification [56], [57], shape segmentation [58], [59], object detection [60], [61], scene classification [62], [63]

Few-shot Semantic Segmentation. Few-shot semantic segmentation has been extensively studied for 2D images [64], [65], [66], [67], [68], [69], and these works majorly pivot around metric-based approach [53], [26], addressing the one-to-many correspondence challenge between the support and query sets, with each class represented by a single global vector. Subsequent work [58], [22], [57], [23], [70] extends the few-shot segmentation from 2D images to a more complex arena of 3D point clouds with only a few annotated support samples. Specifically, in [22], the sampling of the farthest points is used to generate multiple prototypes, in contrast to a single prototype per class as in [53], allowing for a more comprehensive representation of the intricate data distribution inherent in point clouds. Then, a transductive inference [71], [72] is undertaken to bridge the gap between multi-prototypes and query points, and to deduce the most possible labels for each point in the query set. Our work is built upon [22] but instead considers a more difficult setting with different types of noisy samples as in [31], [30], [27]. The most recent concurrent work [73] considers a subset of all noisy sample cases as in [27].

III. Methods

In this section, we first formalize our noisy few-shot segmentation problem in Section III-A, followed by a brief review of Prototypical Network [53] in Section III-B which provides some fundamental concepts for metric-based few-shot learning as adopted in our work. Then, we introduce our noise removal scheme to filter out noisy samples in Section III-C. Next, we detail how to construct a graph for label propagation using weighted subgraph bagging in Section III-D. Finally, we end with a formulation of the loss function that contains a triple contrastive comparison to boost performance in Section III-E.

A. Problem Formulation

The few-shot learning semantic segmentation trains a model on a diverse collection of tasks, each of which is referred as an *episode* [26]. These episodes are sampled from a training set of different classes, denoted by $\mathcal{C}_{\texttt{train}}$. Subsequent to the training phase, the performance of the model is evaluated on a separate set of tasks $\mathcal{C}_{\texttt{test}}$, bound to a set of novel classes that are never seen during training. A crucial design ensures that there is no overlap or intersection between the classes in $\mathcal{C}_{\texttt{train}}$ and $\mathcal{C}_{\texttt{test}}$. Each episode embodies a N-way K-shot point cloud semantic segmentation task. In this framework, for every episode, a support set $\mathcal{S}$ encapsulates K labeled samples for each of the N distinct classes. Specifically, the set $\mathcal{S}$ comprises a collection of point clouds represented by $\{\boldsymbol{P}_n^k\}_{n=1,\cdots,N}^{k=1,\cdots,K}$, with their corresponding labels given by $\{\boldsymbol{L}_n^k\}_{n=1,\cdots,N}^{k=1,\cdots,K}$. Each point cloud $\boldsymbol{P}_n^k$ contains M points that encompass raw coordinates and additional C dimensional features, such as color and/or normal. Therefore, we denote $\boldsymbol{P}_n^k \in \mathbb{R}^{(3+C)\times M}$ and $\boldsymbol{L}_n^k \in \mathbb{R}^M$. In addition, an integral component of each episode is the query set $\mathcal{Q}$. This set consists of query point clouds, represented as $\{\boldsymbol{P}_n^q\}_{n=1,\cdots,N}^{q=1,\cdots,Q}$, which are sampled from identical N classes from which the support set is constructed. During training, the true class labels for these

query point clouds remain inaccessible. The objective of few-shot semantic segmentation is to learn a model that, when given the support set $\mathcal{S}$, can accurately predict the labels for each point cloud $\boldsymbol{P}_n^q$ in the query set $\mathcal{Q}$.

However, in our noisy few-shot segmentation setting, we consider a scenario where the label $\boldsymbol{L}_n^k$ contains instance-level mis-labeling, *i.e.*, some points in $\boldsymbol{P}_n^k$ belonging to one class are mistakenly labeled as another class in $\boldsymbol{L}_n^k$ [30], [27]. We denote the mislabeled pair $(\hat{\boldsymbol{P}}_n^k, \hat{\boldsymbol{L}}_n^k)$ as *noisy* samples in the support set $\mathcal{S}$. Consequently, the support set $\mathcal{S}$ now contains both clean samples, *i.e.*, correctly labeled point clouds, and noisy samples. Such a case is common in the real world due to imperfection of point cloud scanning and annotation. The goal of our noisy few-shot 3D scene point cloud semantic segmentation is to effectively infer point labels from the query set $\mathcal{Q}$, even when a model is trained on the support set $\mathcal{S}$ contaminated with noisy samples.

B. Prototypical Network

Prototypical Network [53] utilizes an episodic scheme to train a meta-learner classifier for object classification. For a given episode comprising a support set $\mathcal{S}$ and a query set $\mathcal{Q}$, the representations of objects within $\mathcal{S}$ are computed through a meta-learner feature extractor f_ϕ parameterized by ϕ, and these representations are then aggregated to derive the prototypes p_c for every class $c \in \mathcal{C} \subset \mathcal{C}_{\texttt{train}}$ contained within the support set $\mathcal{S}$ as follows:

$$p_c = \frac{1}{K} \sum_{(x_i, y_i) \in \mathcal{S}_c} f_\phi(x_i) \ , \tag{1}$$

where $\mathcal{S}_c$ is the subset of the support set $\mathcal{S}$ that contains all samples of class c, *i.e.*, $y_i = c$. To optimize the feature extractor f_ϕ, [53] minimizes the loss:

$$\mathcal{L} = -\frac{1}{N \times K} \sum_{c \in \mathcal{C}} \sum_{(x_j, y_j) \in \mathcal{Q}_c} \log p_\phi(y_j = c \mid x_j) \ . \tag{2}$$

where $\mathcal{Q}_c$ is the subset of the query set $\mathcal{Q}$ that contains samples of class c, $p_\phi(y_j = c \mid x_j)$ is the probability of a query sample $(x_j, y_j) \in \mathcal{Q}$ as class c calculated as:

$$p_\phi(y_j = c \mid x_j) = \frac{\exp(-d(f_\phi(x_j), p_c))}{\sum_{c' \in \mathcal{C}} \exp(-d(f_\phi(x_j), p_{c'}))} \ , \tag{3}$$

where $d(\cdot, \cdot)$ represents a certain distance metric, such as Euclidean distance or Cosine distance.

C. Noise Removal

In Prototypical Network [53], each class is represented by a single prototype, as shown in Eq. 1. Instead, within the support set $\mathcal{S}$ comprising $N + 1$[1] classes, we produce $m (m > 1)$ prototypes for each class, with the aim of capturing the intricate point distribution in an episode. Following [22], we select seed points and assign points to these seeds based on distance in a learned embedding space. Specifically, we first sample m seed points from all points of a single class in the support set using farthest point sampling [6]. Then, we compute point-to-seed distance and take the index of the closest seed as the assignment of a given point. Finally, multi-prototypes for class c is denoted as $\boldsymbol{\mu}_c = \{\mu_c^1, \cdots, \mu_c^m\}$ where μ_c^m is the mean of all features of the points assigned to the m-th seed.

However, since the support set contains noisy samples, the prototypes are not completely reliable. To mitigate adverse effects introduced by noisy samples, we propose a noise removal mechanism to eliminate potential noisy prototypes as accurately as possible. First, we build a fully connected graph in which each vertex i denotes a prototype μ_i where $i \in \{1, 2, \cdots, N \times m\}$. The edge weight between μ_i and μ_j is designed as:

$$W_{ij} = W_{ji} = \sum_{n=1}^{N \times m} \frac{w_{nj} \times w_{in}}{1 + |w_{nj} - w_{in}|} \ , \tag{4}$$

where w_{ij} is the Euclidean or Cosine distance between μ_i and μ_j. It can be regarded as a two-hop walk between vertex i and vertex j. If μ_i and μ_j belong to the same class, W_{ij} (or W_{ji}) results in a relatively large value regardless of whether μ_n belongs to the same class as μ_i (or μ_j) or not. Then, we calculate the degree $D_i = \sum_j W_{ij}$ for each vertex i, indicative of its connectivity within the graph. We reasonably speculate that *noisy* prototypes μ_k which largely contain noisy samples typically exhibit lower degrees due to their dispersion in the feature space, *i.e.*, different from μ_i and μ_j, leading to a lower value of D_k, while other *clean* prototypes cluster together. Note that it could also be possible that some vertices with low degree do not represent noisy prototypes, considering that the data distribution for a given class is multimodal, and one of the modes may be very different from all other prototypes, but still representative of the given class. To reduce this annoying possibility, we pretrain our model on a dataset without noisy samples using the scheme described in Section III-B as in [27] to reduce the modes in the embedding space. Afterward, we remove prototypes with the top-k small degree values, where the value k depends on the ratio of noisy samples per episode, *i.e.*, $k < m$ to ensure that we do not remove all prototypes for a given class.

D. Label Propagation

Given the cleaned $\tilde{N}$ $(\tilde{N} < N \times m)$ prototypes after noise removal, we construct a k-Nearest Neighbor (k-NN) graph $\mathcal{G}$ in which the vertices consist of all the cleaned prototypes and the point features from the query set, such that the total number of vertices in $\mathcal{G}$ is $V = \tilde{N} + (N \times Q \times M)$. A sparse affinity matrix $\boldsymbol{A} \in \mathbb{R}^{V \times V}$, is computed based on the Euclidean or Cosine similarity between each vertex and its k nearest neighbors in the embedding space. Following [74], [22], we set $\boldsymbol{W} = \boldsymbol{A} + \boldsymbol{A}^\top$ to ensure that the adjacency matrix is non-negative and symmetric. Then, $\boldsymbol{W}$ is normalized by $\boldsymbol{S} = \boldsymbol{D}^{-\frac{1}{2}} \boldsymbol{W} \boldsymbol{D}^{-\frac{1}{2}}$, where $\boldsymbol{D}$ denotes the diagonal degree matrix. In addition, we introduce a label matrix $\boldsymbol{Y} \in \mathbb{R}^{\tilde{N} + (N \times Q \times M)}$ in which the rows corresponding to the clean labeled prototypes are set as one-hot ground-truth label vectors, while all other rows are initialized to

[1]We add an additional class to represent "background" as in [22].

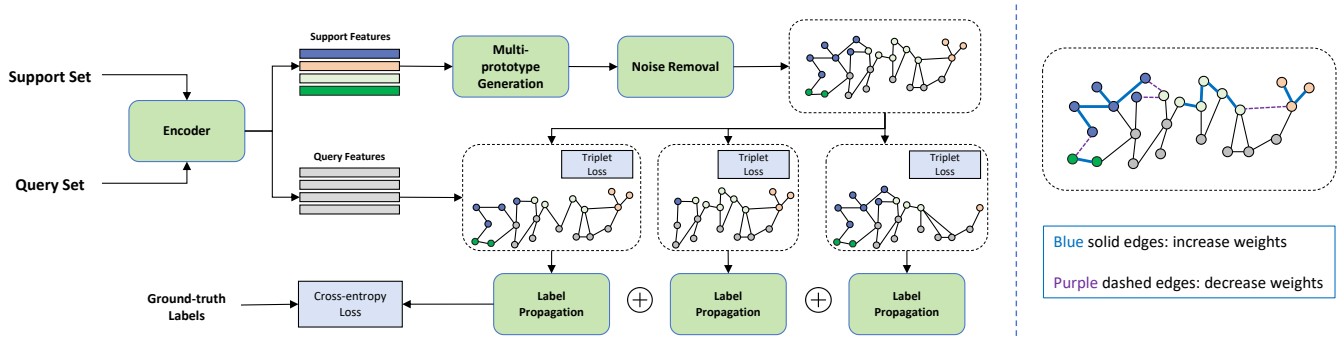

Fig. 1. Left: Overview of the proposed method. Right: Graph concentration by adjusting edges. The vertices with the same label are in the same color.

zero. Given $\boldsymbol{S}$ and $\boldsymbol{Y}$, label propagation iteratively diffuses labels through the k-NN graph [71], [72], [22] at each step t:

$$\boldsymbol{Z}_{t+1} = \alpha \boldsymbol{S} \boldsymbol{Z}_t + (1-\alpha)\boldsymbol{Y} \ , \tag{5}$$

where $\boldsymbol{Z}_t$ denotes the predicted labels at step t. The parameter $\alpha \in (0,1)$ modulates the probability of accepting the label information of the adjacency vertices or keeping the initial labels. The work [71] proves that the sequence $\{\boldsymbol{Z}_t\}$ converges to a closed-form solution:

$$\boldsymbol{Z}^* = (\boldsymbol{I} - \alpha \boldsymbol{S})^{-1} \boldsymbol{Y} \ . \tag{6}$$

Similar to [22], [72], we also utilize the closed-form solution to directly compute point labels in the query set.

However, although we adopt noise removal in Section III-C, it cannot still be guaranteed that all noisy prototypes are removed. To further make label propagation more robust against noisy samples, we design a weighted subgraph bagging scheme to improve the influence of cleaned prototypes during the diffusion process, while suppressing the negative influence of potentially remaining noisy prototypes. Specifically, given a sample ratio $r \in (0,1)$, we randomly select $r \times \tilde{N}$ vertices (with replacement) exclusively from the labeled prototypes along with their connection weights in graph $\mathcal{G}$ and denote it as subgraph $\mathcal{G}_i^s$ where i denotes the i-th selection. Following Eq. 6, we compute the predicted label $\boldsymbol{Z}_i^*$ using prototypes in $\mathcal{G}_i^s$, and the final prediction is averaged by all $\boldsymbol{Z}_i^*$ weighted by w_i^s, $i \in \{1, \cdots, I\}$ where I is the total number of selection of prototype vertices. Ideally, our goal is to increase the weight w_i^s of $\boldsymbol{Z}_i^*$ during the final label prediction, only if $\mathcal{G}_i^s$ is more reliable. To realize this, we divide the set of edges of $\mathcal{G}_i^s$ into two subsets, with the set one $\mathcal{E}_{i,+}^s$ containing edges connecting vertices of the same class and the other $\mathcal{E}_{i,-}^s$ containing edges connecting vertices from different classes. The more concentrated the subgraph $\mathcal{G}_i^s$ as shown in the right panel of Figure 1, the more reliable the propagation of the label from the set of vertices in this subgraph. Therefore, we define w_i^s as *compactness* of $\mathcal{G}_i^s$ as:

$$w_i^s = \frac{\sum_j e_{ij} \times \mathbb{I}[e_{ij} \in \mathcal{E}_{i,+}^s]}{\sum_j e_{ij} \times \mathbb{I}[e_{ij} \in \mathcal{E}_{i,-}^s]} \ , \tag{7}$$

where $\mathbb{I}[\cdot]$ represents an indicator function, and the final label prediction is defined as $\boldsymbol{Z}^* = \sum_i w_i^s \boldsymbol{Z}_i^*$. From the definition of w_i^s in Eq. 7, if the prototypes of the same class cluster together, *i.e.*, edges in $\mathcal{E}_{i,+}^s$ connecting them have larger weights, w_i^s will result in a larger value, which in turn helps in a more accurate and confident of $\boldsymbol{Z}_i^*$. A brief derivation is provided in Appendix[2]. Note that the operation of averaging predictions from models trained on selected vertices with replacement shares similarity with the standard Bagging strategy, but instead we incorporate weights w_i^s for averaging to get the final prediction.

E. Loss Function

Once $\boldsymbol{Z}^*$ is obtained, we compute the cross-entropy loss $\mathcal{L}_{\texttt{ce}}$[3] between the predictions of the points in the query set $\mathcal{Q}$ and the corresponding ground-truth labels. Furthermore, we propose a triplet loss to increase the *compactness* of each subgraph $\mathcal{G}^s$ as discussed in Section III-D. Specifically, a conventional triplet loss is defined as:

$$\mathcal{L}_{\texttt{triplet}} = \sum_k^K \max(0, \|f(x_k^a) - f(x_k^p)\|_2^2 - \|f(x_k^a) - f(x_k^n)\|_2^2 + \beta) \ , \tag{8}$$

where x_k^a, x_k^p, and x_k^n denote an anchor, a positive sample and a negative sample, respectively. The margin β is a predefined threshold that specifies the minimum distance that should separate the positive and negative pairs from the anchor. Inspired by Eq. 8, we regard each prototype as an anchor, the prototypes with the same label as positive samples and the ones with different labels as negative samples. Thus, we define our triplet loss as follows:

$$\tilde{\mathcal{L}}_{\texttt{triplet}} = \sum_{i,j,k}^{\tilde{N}} \max(0, (w_{ij} - w_{ik} + \beta) \times \mathbb{I}[w_{ij} \in \mathcal{E}_+ \cap w_{ik} \in \mathcal{E}_-]) \ , \tag{9}$$

where w_{ij} represents the edge weight between vertex i and vertex j in subgraph $\mathcal{G}^s$ and the total loss is therefore defined as: $\mathcal{L} = \mathcal{L}_{\texttt{ce}} + \tilde{\mathcal{L}}_{\texttt{triplet}}$. Intuitively, Eq. 9 increases the difference between the edge weight of the vertices with the same label and the weight of the vertices with different labels. Note that both Eq. 7 and Eq. 9 can be efficiently implemented using a matrix slicing operation without loops.

[2]Appendix: https://github.com/hhuang-code/Noisy_Fewshot_Segmentation/blob/main/Appendix.pdf.

[3]For conciseness, we omit the expression of the cross-entropy loss here.

TABLE I

QUANTITATIVE RESULTS FOR THE SYMMETRIC SWAP LABEL NOISE CASE FOR S3DIS AND SCANNET DATASETS.

Methods	S3DIS				ScanNet			
	20%		*40%*		*20%*		*40%*	
	2-way	**3-way**	**2-way**	**3-way**	**2-way**	**3-way**	**2-way**	**3-way**
ProtoNet [53]	40.32	39.40	21.20	24.16	22.50	23.60	18.23	14.35
MPTI-SAN [22]	45.91	41.38	23.36	26.82	24.13	25.41	20.20	15.32
TraNFS [27]	41.20	38.15	24.34	26.30	26.28	25.36	21.35	**16.17**
Ours	**47.38**	**44.20**	**26.26**	**30.37**	**27.40**	**27.09**	**23.84**	12.71

TABLE II

QUANTITATIVE RESULTS FOR THE PAIRED SWAP LABEL NOISE CASE FOR S3DIS AND SCANNET DATASETS.

Methods	S3DIS				ScanNet			
	20%		*40%*		*20%*		*40%*	
	2-way	**3-way**	**2-way**	**3-way**	**2-way**	**3-way**	**2-way**	**3-way**
ProtoNet [53]	41.30	41.81	18.10	22.73	18.32	17.46	16.38	8.27
MPTI-SAN [22]	45.14	42.32	23.18	28.27	25.20	22.18	20.06	10.12
TraNFS [27]	46.25	**45.50**	23.37	28.90	26.30	23.37	22.10	**11.39**
Ours	**48.60**	44.46	**24.68**	**30.18**	**28.75**	**26.51**	**24.45**	10.40

TABLE III

QUANTITATIVE RESULTS FOR THE OUTLIER LABEL NOISE CASE.

Methods	S3DIS			
	20%		*40%*	
	2-way	**3-way**	**2-way**	**3-way**
ProtoNet [53]	34.26	38.10	25.18	23.13
MPTI-SAN [22]	36.12	41.35	28.40	26.16
TraNFS [27]	37.34	42.90	27.64	25.28
Ours	**40.88**	**46.93**	**32.33**	**30.43**

TABLE IV

CLASS SPLITS. WE FOLLOW THE SAME CLASS SPLITS AS IN [22] AND USE SPLIT-0 FOR TRAINING AND SPLIT-1 FOR TESTING.

	$\mathcal{C}_{\texttt{train}}$	$\mathcal{C}_{\texttt{test}}$
S3DIS	beam, board, bookcase, ceiling, chair, column	door, floor, sofa, table, wall, window
ScanNet	bathtub, bed, bookshelf, cabinet, chair, counter, curtain, desk, door, floor	otherfurniture, picture, refrigerator, shower curtain, sink, sofa, table, toilet, wall, window

IV. EXPERIMENTS

A. Datasets

We carried out experiments on two public 3D semantic segmentation benchmark datasets: S3DIS [12] and ScanNet-v2 [13]. The ScanNet dataset comprises point cloud data from 1,513 scans, representing 707 distinct indoor environments, annotated across 20 semantic categories. The S3DIS dataset features point clouds from 272 rooms that span six varied indoor locations, annotated within 12 semantic categories. Scenes from both the S3DIS and ScanNet datasets were partitioned into non-overlapping $1m \times 1m$ blocks on the xy-plane [22]. We sampled $M = 2048$ points from each block with the input feature as a concatenation of XYZ coordinates, RGB values, and normalized XYZ coordinates. During training, we randomly generated episodes using the `learn2learn` library [75]. This involves sampling N classes from $\mathcal{C}_{\texttt{train}}$ and drawing K point clouds from each of the sampled classes to construct the support set $\mathcal{S}$, and selecting Q point clouds from each class to form the query set $\mathcal{Q}$. Note that the original class labels are remapped in each episode to avoid the model memorizing these labels. The testing episodes were constructed from $\mathcal{C}_{\texttt{test}}$ similarly. For both datasets, the class splits for $\mathcal{C}_{\texttt{train}}$ and $\mathcal{C}_{\texttt{test}}$ are shown in Table IV.

B. Label Noise Types

We consider three types of label noise used in [27] and a formal probability formulation can be found in [31]:

- **Symmetric label swap noise** selects mislabeled samples through a uniform random distribution from the remaining $N-1$ classes within a given episode, adhering to a constraint whereby the number of samples of any given noisy class does not equate to or exceed the number of the clean class samples.
- **Paired label swap noise** draws mislabeled samples from the same class and assigns each class intentionally with a wrong class counterpart. These assignments are generated through random derangement.
- **Outlier noise** draws noisy samples from classes that are not contained in the N-way episodes. For this type of noise, we only evaluate on S3DIS and use ScanNet as the source of noisy classes. Specifically, we remove the common classes[4] and draw noisy samples from the remaining classes in ScanNet.

C. Implementation Details

The feature encoder is DGCNN as a backbone as in [22] and pretrain it on datasets without noise using the ProtoNet objective as described in Section III-B. We adopt AdamW [76] optimizer with a learning rate of 0.001. As we transition to episodic training, we freeze the pretrained DGCNN backbone, and train other newly introduced layers including an attention module. Following [22], we set the few-shot settings to 2-way 5-shot and 3-way 5-shot. To ensure that clean samples outnumber noisy ones in each episode, we set the noise ratio to 20% and 40%.

[4]The common classes contained in both S3DIS and ScanNet are: chair, door, floor, sofa, table, wall, window, bookshelf.

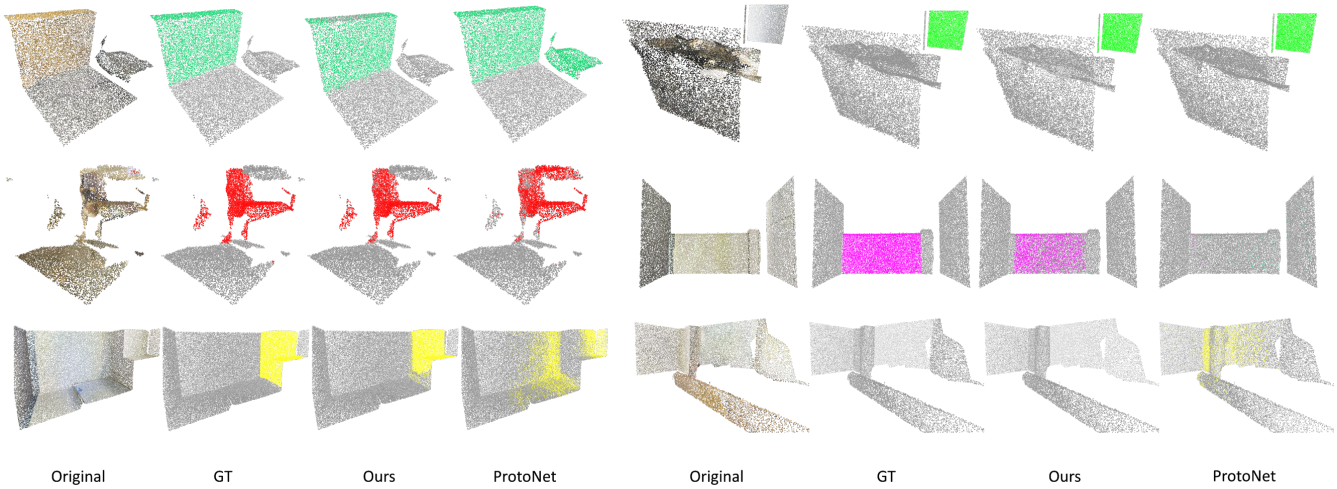

Fig. 2. Visual comparisons of our method, ProtoNet [53], and the ground-truth labels on six blocks of scenes from S3DIS dataset.

D. Baselines

We select Prototypical Network (ProtoNet) [53], MPTI-SAN [22] and TraNFS [27] as baselines against which we compare our method, as our method is mainly based on these works. Note that for ProtoNet and TraNFS which are initially designed for 2D image classification, we replace their 2D backbone networks and classifiers with the 3D networks.

TABLE V

QUANTITATIVE RESULTS OF ABLATION STUDY ON S3DIS DATASET.

Methods	Symmetric Swap				Paired Swap			
	20%		*40%*		*20%*		40%	
	2-way	3-way	2-way	3-way	2-way	3-way	2-way	3-way
W/o noise removal	38.21	39.13	21.10	22.35	34.12	32.46	17.57	22.19
W/o bagging	43.30	38.24	20.42	25.31	35.27	33.72	20.96	25.40
W/o triplet loss	46.22	43.38	25.10	28.74	44.50	41.96	22.50	28.71
Ours	47.38	44.20	26.26	30.37	48.60	44.46	24.68	30.18

E. Results

We adopt textitmean Intersection over Union (mIoU, %) as the evaluation metric, which is commonly used in segmentation tasks. We report our results with different noise ratio in the case of symmetric swap label noise on the S3DIS and ScanNet-v2 datasets in Table I, and the results under paired swap label noise in Table II. For the outlier label noise case, we only evaluated on the S3DIS dataset and the results are reported in Table III. We notice that our method achieves superior segmentation results across different noise ratios in most cases, indicating the robustness of our model against noisy labels. As expected, when the noise ratio increases, the performance drops. We also note that the performance of ScanNet with the 40% noise ratio is the worst, and we speculate that the reason probably could be that ScanNet contains more classes than S3DIS, and thus a higher noise ratio introduces more noise patterns or modes from distracting classes, which is difficult to discriminate.

In Figure 2, we visualize a qualitative comparison between our method against ProtoNet [53] and the ground-truth annotation. It is obvious that ProtoNet fails for some instance segmentation, which may be attributed to a single prototype per class and the lack of noisy sample removal.

V. ABLATION STUDY

In Table V, we evaluate the effectiveness of each component of our model: 1). noise removal, 2). bagging scheme, and 3). triplet loss on the S3DIS dataset. We conducted ablative studies using the S3DIS dataset. Independently, ablating one of the three components leads to a performance drop, and the combination of all the components results in optimal performance. In particular, the robustness of our model is enhanced by simply incorporating our noise removal component. This supports our claim that our model has the capacity to robustly segment scene point clouds against noise.

VI. CONCLUSION

The majority of 3D point cloud scene semantic segmentation approaches depend on extensive fully labeled datasets and are susceptible to inaccuracies introduced by noisy data. In this paper, we introduce a novel few-shot segmentation method crafted to enhance the robustness of segmenting 3D point cloud scenes against noise. By constructing a multi-prototype graph and employing noise removal, coupled with weighted subgraph bagging for effective label propagation, we facilitate the learning of discriminative point features for segmentation. This approach is evaluated across two 3D point cloud scene segmentation benchmarks in few-shot scenarios, exhibits superior resilience to noisy data and outperforms established baselines.

LIMITAIONS AND BROADER IMPACT

The limitations of this work are two aspects: 1. Although incorrect labels are allowed during training, it still needs to annotate amounts of points, which is time-consuming and labor-extensive; and 2. We only consider indoor scenes, while many robot applications focus on outdoor scenes. Despite these limitations, 3D few-shot scene segmentation could enhance the adaptability and efficiency of robots in understanding and interacting with complex outdoor environments.

REFERENCES

[1] D. Zermas, I. Izzat, and N. Papanikolopoulos, "Fast segmentation of 3d point clouds: A paradigm on lidar data for autonomous vehicle applications," in *IEEE International Conference on Robotics and Automation*. IEEE, 2017, pp. 5067–5073.

[2] T. Guan, D. Kothandaraman, R. Chandra, A. J. Sathyamoorthy, K. Weerakoon, and D. Manocha, "Ga-nav: Efficient terrain segmentation for robot navigation in unstructured outdoor environments," *IEEE Robotics and Automation Letters*, vol. 7, no. 3, pp. 8138–8145, 2022.

[3] S. Jain and B. Argall, "Automated perception of safe docking locations with alignment information for assistive wheelchairs," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2014, pp. 4997–5002.

[4] A. C. Dometios, X. S. Papageorgiou, A. Arvanitakis, C. S. Tzafestas, and P. Maragos, "Real-time end-effector motion behavior planning approach using on-line point-cloud data towards a user adaptive assistive bath robot," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2017, pp. 5031–5036.

[5] D. Wang, C. Kohler, A. ten Pas, A. Wilkinson, M. Liu, H. Yanco, and R. Platt, "Towards assistive robotic pick and place in open world environments," in *International Symposium of Robotics Research*. Springer, 2019, pp. 360–375.

[6] 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.

[7] Q. Huang, W. Wang, and U. Neumann, "Recurrent slice networks for 3d segmentation of point clouds," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 2626–2635.

[8] L. Landrieu and M. Simonovsky, "Large-scale point cloud semantic segmentation with superpoint graphs," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 4558–4567.

[9] Q. Hu, B. Yang, L. Xie, S. Rosa, Y. Guo, Z. Wang, N. Trigoni, and A. Markham, "Randla-net: Efficient semantic segmentation of large-scale point clouds," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2020, pp. 11 108–11 117.

[10] W. Hu, H. Zhao, L. Jiang, J. Jia, and T.-T. Wong, "Bidirectional projection network for cross dimension scene understanding," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2021, pp. 14 373–14 382.

[11] 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 Conference on Computer Vision and Pattern Recognition*, 2022, pp. 8500–8509.

[12] I. Armeni, O. Sener, A. R. Zamir, H. Jiang, I. Brilakis, M. Fischer, and S. Savarese, "3d semantic parsing of large-scale indoor spaces," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 1534–1543.

[13] A. Dai, A. X. Chang, M. Savva, M. Halber, T. Funkhouser, and M. Nießner, "Scannet: Richly-annotated 3d reconstructions of indoor scenes," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 5828–5839.

[14] Z. Zhang, R. Girdhar, A. Joulin, and I. Misra, "Self-supervised pretraining of 3d features on any point-cloud," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 10 252–10 263.

[15] L. Nunes, R. Marcuzzi, X. Chen, J. Behley, and C. Stachniss, "Segcontrast: 3d point cloud feature representation learning through self-supervised segment discrimination," *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 2116–2123, 2022.

[16] X. Xu and G. H. Lee, "Weakly supervised semantic point cloud segmentation: Towards 10x fewer labels," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2020, pp. 13 706–13 715.

[17] Q. Hu, B. Yang, G. Fang, Y. Guo, A. Leonardis, N. Trigoni, and A. Markham, "Sqn: Weakly-supervised semantic segmentation of large-scale 3d point clouds," in *European Conference on Computer Vision*. Springer, 2022, pp. 600–619.

[18] K. Liu, Y. Zhao, Z. Gao, and B. M. Chen, "Weaklabel3d-net: A complete framework for real-scene lidar point clouds weakly supervised multi-tasks understanding," in *International Conference on Robotics and Automation*. IEEE, 2022, pp. 5108–5115.

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

[20] S. Jadon and A. Jadon, "An overview of deep learning architectures in few-shot learning domain," *arXiv preprint arXiv:2008.06365*, 2020.

[21] C. Sharma and M. Kaul, "Self-supervised few-shot learning on point clouds," *Advances in Neural Information Processing Systems*, vol. 33, pp. 7212–7221, 2020.

[22] 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*, 2021, pp. 8873–8882.

[23] Y. Mao, Z. Guo, L. Xiaonan, Z. Yuan, and H. Guo, "Bidirectional feature globalization for few-shot semantic segmentation of 3d point cloud scenes," in *International Conference on 3D Vision*. IEEE, 2022, pp. 505–514.

[24] K. Koreitem, F. Shkurti, T. Manderson, W.-D. Chang, J. C. G. Higuera, and G. Dudek, "One-shot informed robotic visual search in the wild," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2020, pp. 5800–5807.

[25] X. Liu, Y. Zhang, and D. Shan, "Unseen object few-shot semantic segmentation for robotic grasping," *IEEE Robotics and Automation Letters*, vol. 8, no. 1, pp. 320–327, 2022.

[26] O. Vinyals, C. Blundell, T. Lillicrap, D. Wierstra *et al.*, "Matching networks for one shot learning," *Advances in Neural Information Processing Systems*, vol. 29, 2016.

[27] K. J. Liang, S. B. Rangrej, V. Petrovic, and T. Hassner, "Few-shot learning with noisy labels," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 9089–9098.

[28] D. Tsipras, S. Santurkar, L. Engstrom, A. Ilyas, and A. Madry, "From imagenet to image classification: Contextualizing progress on benchmarks," in *International Conference on Machine Learning*. PMLR, 2020, pp. 9625–9635.

[29] C. G. Northcutt, A. Athalye, and J. Mueller, "Pervasive label errors in test sets destabilize machine learning benchmarks," in *Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2021.

[30] S. Ye, D. Chen, S. Han, and J. Liao, "Learning with noisy labels for robust point cloud segmentation," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 6443–6452.

[31] B. Han, Q. Yao, X. Yu, G. Niu, M. Xu, W. Hu, I. Tsang, and M. Sugiyama, "Co-teaching: Robust training of deep neural networks with extremely noisy labels," *Advances in Neural Information Processing Systems*, vol. 31, 2018.

[32] M. Tatarchenko, J. Park, V. Koltun, and Q.-Y. Zhou, "Tangent convolutions for dense prediction in 3d," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 3887–3896.

[33] B. Wu, A. Wan, X. Yue, and K. Keutzer, "Squeezeseg: Convolutional neural nets with recurrent crf for real-time road-object segmentation from 3d lidar point cloud," in *IEEE International Conference on Robotics and Automation*. IEEE, 2018, pp. 1887–1893.

[34] B. Wu, X. Zhou, S. Zhao, X. Yue, and K. Keutzer, "Squeezesegv2: Improved model structure and unsupervised domain adaptation for road-object segmentation from a lidar point cloud," in *International Conference on Robotics and Automation*. IEEE, 2019, pp. 4376–4382.

[35] B. Graham, M. Engelcke, and L. Van Der Maaten, "3d semantic segmentation with submanifold sparse convolutional networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 9224–9232.

[36] H. Su, V. Jampani, D. Sun, S. Maji, E. Kalogerakis, M.-H. Yang, and J. Kautz, "Splatnet: Sparse lattice networks for point cloud processing," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 2530–2539.

[37] X. Zhu, H. Zhou, T. Wang, F. Hong, Y. Ma, W. Li, H. Li, and D. Lin, "Cylindrical and asymmetrical 3d convolution networks for lidar segmentation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern recognition*, 2021, pp. 9939–9948.

[38] 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*, 2017, pp. 652–660.

[39] 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*, vol. 38, no. 5, pp. 1–12, 2019.

[40] 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 International Conference on Computer Vision*, 2019, pp. 6411–6420.

[41] F. Engelmann, T. Kontogianni, and B. Leibe, "Dilated point convolutions: On the receptive field size of point convolutions on 3d point clouds," in *IEEE International Conference on Robotics and Automation*. IEEE, 2020, pp. 9463–9469.

[42] H. Zhao, L. Jiang, J. Jia, P. H. Torr, and V. Koltun, "Point transformer," in *Proceedings of the IEEE International Conference on Computer Vision*, 2021, pp. 16 259–16 268.

[43] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in Neural Information Processing Systems*, vol. 30, 2017.

[44] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly *et al.*, "An image is worth 16x16 words: Transformers for image recognition at scale," in *International Conference on Learning Representations*, 2020.

[45] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko, "End-to-end object detection with transformers," in *European Conference on Computer Vision*. Springer, 2020, pp. 213–229.

[46] J. Ho, A. Jain, and P. Abbeel, "Denoising diffusion probabilistic models," *Advances in Neural Information Processing Systems*, vol. 33, pp. 6840–6851, 2020.

[47] L. Zhou, Y. Du, and J. Wu, "3d shape generation and completion through point-voxel diffusion," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 5826–5835.

[48] S. Luo and W. Hu, "Diffusion probabilistic models for 3d point cloud generation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 2837–2845.

[49] A. Nichol, H. Jun, P. Dhariwal, P. Mishkin, and M. Chen, "Point-e: A system for generating 3d point clouds from complex prompts," *arXiv preprint arXiv:2212.08751*, 2022.

[50] Z. Wu, Y. Wang, M. Feng, H. Xie, and A. Mian, "Sketch and text guided diffusion model for colored point cloud generation," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, pp. 8929–8939.

[51] T. Li, Y. Fu, X. Han, H. Liang, J. J. Zhang, and J. Chang, "Diffusion-pointlabel: Annotated point cloud generation with diffusion model," in *Computer Graphics Forum*, vol. 41, no. 7. Wiley Online Library, 2022, pp. 131–139.

[52] S. Ravi and H. Larochelle, "Optimization as a model for few-shot learning," in *International Conference on Learning Representations*, 2016.

[53] J. Snell, K. Swersky, and R. Zemel, "Prototypical networks for few-shot learning," *Advances in Neural Information Processing Systems*, vol. 30, 2017.

[54] C. Finn, P. Abbeel, and S. Levine, "Model-agnostic meta-learning for fast adaptation of deep networks," in *International Conference on Machine Learning*. PMLR, 2017, pp. 1126–1135.

[55] V. Garcia and J. Bruna, "Few-shot learning with graph neural networks," in *International Conference on Learning Representations*, 2018.

[56] J. Nie, N. Xu, M. Zhou, G. Yan, and Z. Wei, "3d model classification based on few-shot learning," *Neurocomputing*, vol. 398, pp. 539–546, 2020.

[57] H. Huang, X. Li, L. Wang, and Y. Fang, "3d-metaconnet: meta-learning for 3d shape classification and segmentation," in *International Conference on 3D Vision*. IEEE, 2021, pp. 982–991.

[58] P. Tian, Z. Wu, L. Qi, L. Wang, Y. Shi, and Y. Gao, "Differentiable meta-learning model for few-shot semantic segmentation," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, no. 07, 2020, pp. 12 087–12 094.

[59] X. Li, L. Feng, L. Li, and C. Wang, "Few-shot meta-learning on point cloud for semantic segmentation," *arXiv preprint arXiv:2104.02979*, 2021.

[60] S. Zhao and X. Qi, "Prototypical votenet for few-shot 3d point cloud object detection," *Advances in Neural Information Processing Systems*, vol. 35, pp. 13 838–13 851, 2022.

[61] S. Yuan, X. Li, H. Huang, and Y. Fang, "Meta-det3d: Learn to learn few-shot 3d object detection," in *Proceedings of the Asian Conference on Computer Vision*, 2022, pp. 1761–1776.

[62] C. Ma, X. Mu, P. Zhao, and X. Yan, "Meta-learning based on parameter transfer for few-shot classification of remote sensing scenes," *Remote Sensing Letters*, vol. 12, no. 6, pp. 531–541, 2021.

[63] S. Dong, S. Wang, Y. Zhuang, J. Kannala, M. Pollefeys, and B. Chen, "Visual localization via few-shot scene region classification," in *International Conference on 3D Vision*. IEEE, 2022, pp. 393–402.

[64] N. Dong and E. Xing, "Few-shot semantic segmentation with prototype learning." in *British Machine Vision Conference*, 2017.

[65] K. Nguyen and S. Todorovic, "Feature weighting and boosting for few-shot segmentation," in *Proceedings of the IEEE International Conference on Computer Vision*, 2019, pp. 622–631.

[66] K. Wang, J. H. Liew, Y. Zou, D. Zhou, and J. Feng, "Panet: Few-shot image semantic segmentation with prototype alignment," in *proceedings of the IEEE International Conference on Computer Vision*, 2019, pp. 9197–9206.

[67] C. Zhang, G. Lin, F. Liu, J. Guo, Q. Wu, and R. Yao, "Pyramid graph networks with connection attentions for region-based one-shot semantic segmentation," in *Proceedings of the IEEE International Conference on Computer Vision*, 2019, pp. 9587–9595.

[68] W. Liu, C. Zhang, G. Lin, and F. Liu, "Crnet: Cross-reference networks for few-shot segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2020, pp. 4165–4173.

[69] C. Zhang, G. Lin, F. Liu, R. Yao, and C. Shen, "Canet: Class-agnostic segmentation networks with iterative refinement and attentive few-shot learning," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 5217–5226.

[70] C. Zhang, Z. Wu, X. Wu, Z. Zhao, and S. Wang, "Few-shot 3d point cloud semantic segmentation via stratified class-specific attention based transformer network," *arXiv preprint arXiv:2303.15654*, 2023.

[71] D. Zhou, O. Bousquet, T. Lal, J. Weston, and B. Schölkopf, "Learning with local and global consistency," *Advances in Neural Information Processing Systems*, vol. 16, 2003.

[72] M. Lazarou, T. Stathaki, and Y. Avrithis, "Iterative label cleaning for transductive and semi-supervised few-shot learning," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 8751–8760.

[73] Y. Xu, N. Zhao, and G. H. Lee, "Towards robust few-shot point cloud semantic segmentation," in *British Machine Vision Conference*, 2023.

[74] A. Iscen, G. Tolias, 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*, 2019, pp. 5070–5079.

[75] S. M. Arnold, P. Mahajan, D. Datta, I. Bunner, and K. S. Zarkias, "learn2learn: A library for meta-learning research," *arXiv preprint arXiv:2008.12284*, 2020.

[76] I. Loshchilov and F. Hutter, "Decoupled weight decay regularization," in *International Conference on Learning Representations*, 2018.