Generalized Few-Shot 3D Point Cloud Segmentation

Yang Shuqian

School of Electrical & Electronic Engineering

A thesis submitted to the Nanyang Technological University
in partial fulfillment of the requirements for the degree of
Master of Engineering

2024

Statement of Originality

I hereby certify that the work embodied in this thesis is the result of original research, is free of plagiarised materials, and has not been submitted for a higher degree to any other University or Institution.

18/1/2024

Date

Yang Shuqian

Supervisor Declaration Statement

I have reviewed the content and presentation style of this thesis and declare it is free of plagiarism and of sufficient grammatical clarity to be examined. To the best of my knowledge, the research and writing are those of the candidate except as acknowledged in the Author Attribution Statement. I confirm that the investigations were conducted in accord with the ethics policies and integrity standards of Nanyang Technological University and that the research data are presented honestly and without prejudice.

24 Jan 2024

.....

Date

.....

Prof. Jiang Xudong

Authorship Attribution Statement

Please select one of the following; *delete as appropriate:

This thesis contains material from 1 paper(s) published in the following peer-reviewed journal(s) / from papers accepted at conferences in which I am listed as an author.

Please amend the typical statements below to suit your circumstances if (B) is selected.

Chapter 1-6 are published as Shuqian Yang, Henghui Ding, and Xudong Jiang. Generalized Few-Shot 3D Point Cloud Segmentation. IEEE International Symposium on Circuits and Systems, 2024 (accepted as oral presentation).

The contributions of the co-authors are as follows:

- I designed the model framework, implemented the source code, and conducted all experiments at the Rapid-Rich Object Search (ROSE) Lab, School of Electrical and Electronic Engineering.
- I finished the first version of the manuscript draft. Dr. Henghui Ding and my supervisor Prof. Xudong Jiang reviewed it and provided many useful writing suggestions.
- Dr. Henghui Ding and my supervisor Prof. Xudong Jiang gave a lot of valuable guidance on the idea proposal and model design.

18/1/2024

.....

Date

.....

Yang Shuqian

Acknowledgements

First and foremost, I would like to express my deepest gratitude to my supervisor, Xudong Jiang, for this insightful guidance, kind support, and profound knowledge throughout the course of my M.Eng journey. His expertise has not only been the cornerstone of my academic development but also a constant source of inspiration. I am particularly grateful for the direction he provided while allowing me the freedom to explore my own interests in point cloud processing. Besides, As an international student in a foreign country, I am deeply grateful to my supervisor for his instrumental role in helping me integrate smoothly into the school environment.

I would also like to express my sincere thanks to my collaborator, Dr. Henghui Ding, the Presidential Postdoctoral Fellow from NTU. He is not only smart and insightful but also exceptionally capable. Our weekly meetings during my M.Eng journey was invaluable. His guidance and contributions have been crucial to my research, providing direction that significantly enhanced my work.

I would also like to thank my colleagues and peers at Rose Lab. Their diverse views and engineering capabilities have greatly enriched my experience. Their encouragement and help provided me with great support when I encountered difficulties. Additionally, I must express my heartfelt thanks to my family and friends for their support and encouragement. Their belief in my abilities and constant encouragement has been a source of strength and motivation.

I would like to extend my sincere thanks to all the professors who participated in the review and defense of my thesis. Thanks for their willingness to take time out of their busy schedules to provide valuable feedback and guidance. This opportunity not only allowed me to refine my thesis but also to summarize and reflect on my past research, which was an invaluable part of my learning process.

This thesis is not only a reflection of my work but also a testament to the collective effort and support of all those who have guided and stood by me throughout this journey. Thank them again and wish them a happy life!

Contents

List of Figures

List of Tables

Abstract

Few-Shot 3D Point Cloud Semantic Segmentation (3D-FS) mitigates the issues of insufficient data annotation and emerging novel classes in real-world scenarios, but it totally ignores the performance on base classes.

In this paper, we address a more practical task, Generalized Few-Shot 3D Point Cloud Semantic Segmentation (3D-GFS), which aims to perform segmentation simultaneously on base classes with adequate samples and novel classes with few samples. Based on the prototypical Base Model, we propose Adaptive Support Enrichment module and Query Aware Representation module to utilize the contextual information of semantic segmentation. The former exploits the co-relationship between base and novel classes in support samples while the latter mines semantic information from query samples. Besides, considering the different embedding spaces, we propose a new training strategy to get a better representation of prototypes. Experiments on S3DIS and ScanNet show that our proposed method outperforms our Base Model and the conventional 3D-FS methods.

Chapter 1

Introduction

1.1 Background

3D perception is of great importance in computer vision. Diverse 3D sensors like LiDAR, light field cameras, ultrasonic sensors, and structured light sensors, which record real-world objects with digital representations derived from surface sampling points, help a lot in 3D perception. The 3D data processing methods contribute a lot to various real-world applications, including autonomous driving [8], assistive robots [9], digital urban [10], augmented/virtual reality [11], and so on. Therefore, 3D perception is a challenging and worthwhile research problem. Fig. 1.1 illustrates various real-life application scenarios of 3D point cloud technology.

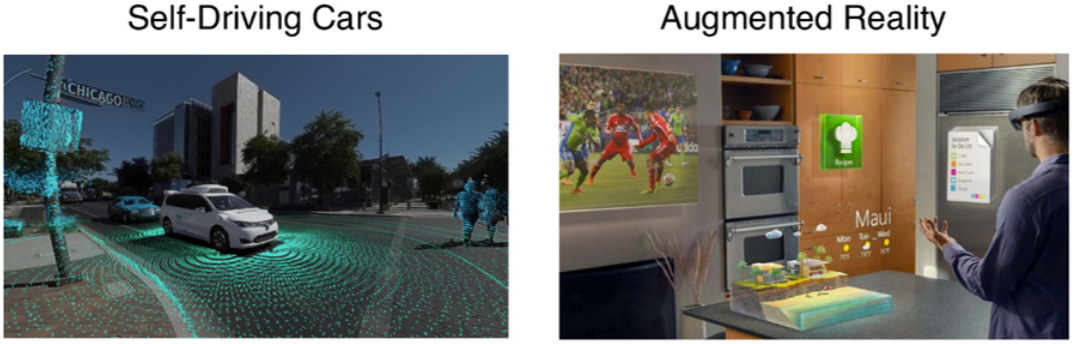

FIGURE 1.1: The application of 3D point cloud.

Different from 2D images, 3D point cloud suffers from inherent properties: unordered, irregular, and invariant under transformations [1], making it impractical

to directly apply established 2D image processing techniques to the 3D point cloud. As a result, investigating specialized approaches designed for 3D point cloud becomes crucial. The thriving development of deep learning brings new improvements. The pioneering work PointNet [1], the enhanced edition PointNet++ [2], and the subsequent work [3, 12–15] employ well-designed neural networks to process point clouds directly, achieving impressive performance.

Semantic segmentation is a key task among different 3D point cloud problems. It involves classifying each point within a scene represented by 3D point clouds. However, the challenge in fully-supervised semantic segmentation is that the promising result heavily relies on training over a huge amount of labeled data, which requires laborious annotations in practice. Besides, these approaches lie in the assumption of a closed set that the training and testing datasets share the same class space. Unfortunately, such an assumption may be too restrictive in real-world scenarios, because novel classes can emerge unpredictably after training. This largely limits the generalizability of the trained models. Therefore, it still remains a challenging problem to obtain a practical solution for 3D point cloud segmentation.

Various studies have tried to address the data constraints in fully-supervised 3D semantic segmentation by employing self- [16], weakly- [17], and semi-supervised [18] learning methods. However, these approaches still largely adhere to the closed-set assumption. This means they overlook the importance of generalizing to novel classes.

Very recently, the concept of few-shot learning has gained great interest within the 3D community. Few-shot 3D Point Cloud Semantic Segmentation [5, 19], denoted as 3D-FS, seeks to overcome such limitations by training a model to segment each point of novel classes in query samples, using support samples as a reference. The dataset of the 3D-FS task consists of two sets, a support set and a query set. In the training/testing phase, the support set provides 3D-FS models with target prior information, which allows the 3D-FS models to recognize target classes in query samples. Subsequently, the 3D-FS model makes predictions on the query samples with the prior knowledge from the support data. 3D-FS imitates the situation with limited labeled data for novel classes.

While the 3D-FS have achieved remarkable segmentation accuracy on the novel classes, the performance of the base classes is completely ignored, because 3D-FS

treats all base classes as background. This is inconsistent with real-world applications. Consequently, 3D-FS models face difficulties in addressing practical evaluation involving both base and novel classes.

1.2 Contribution

With these facts, we consider a practical yet challenging setting **Generalized Few-shot 3D Point Cloud Semantic Segmentation**, denoted as 3D-GFS, which performs segmentation on both well-represented base classes and scarcely-sampled novel classes simultaneously. A typical 3D-GFS approach is structured into three key phases: 1) *base prototype generation phase*, focusing on base classes; 2) *novel prototype registration phase* using support samples of novel classes; and 3) *final evaluation phase*, targeting all classes. The primary distinction between 3D-GFS and 3D-FS lies in their evaluation approach. In contrast to 3D-FS, 3D-GFS avoids complex episodic training [20]. It does not need to process support samples alongside query samples containing identical target classes for prediction. This is because 3D-GFS has acquired essential information of all classes during the *base prototype generation phase* and *novel prototype registration phase* respectively. As a result, the 3D-GFS model achieves remarkable performance on base and novel classes while preserving the prediction accuracy in base classes.

Inspired by 3D-FS task [5], we design a prototypical Base Model for 3D-GFS with decent performance. To utilize the contextual information of semantic segmentation, an Adaptive Support Enrichment (ASE) module is proposed to exploit the essential co-relationship between base and novel classes in support samples in combination with a Query Aware Representation (QAR) module to explore the semantic information from individual query samples. Besides, considering the different embedding spaces, we further design a new training strategy to imitate the scenarios of real novel and base classes and get a better representation of prototypes. Building upon this, we evaluate two benchmarks available for the 3D-GFS task, namely S3DIS [6] and ScanNet [7]. Compared with the Base Model, our method demonstrates notable improvements in the Generalized Few-Shot settings of 5-shot, 10-shot, and 30-shot, achieving approximately 3.66%, 3.82%, and 3.73% enhancement on the S3DIS dataset, and 4.72%, 4.67%, and 3.9% on the ScanNet dataset, respectively. Besides, we demonstrate that applying the 3D-FS model to 3D-GFS

setting performs poorly. Specifically, our method achieves a better performance than the state-of-the-art 3D-FS method distinctly by margins of 26.72/24.17% and 15.05/14.84% under the 5/10-shot on S3DIS and ScanNet datasets, respectively.

Our contributions are as follows:

1. We analyze the shorts of classic Few-shot 3D Point Cloud Semantic Segmentation and we focus on a more pragmatic and applicable task, Generalized Few-shot 3D Point Cloud Semantic Segmentation.
2. We introduce the Base Model for Generalized 3D Few-shot Semantic Segmentation. We further introduce an Adaptive Support Enrichment module and a Query Aware Representation module to utilize the contextual information.
3. We design a new training strategy to imitate the scenarios of real novel and base classes and get a better representation of prototypes.
4. We established two benchmarks using the S3DIS [6] and ScanNet [7] datasets to validate the efficacy of our method.

1.3 Organization of the Report

The report is organized as follows:

1. Chapter 1 outlines the background and motivation of our Generalized Few-shot 3D Point Cloud Semantic Segmentation task and details the significant contributions of our proposed methods.
2. Chapter 2 gives a comprehensive review of the existing literature.
3. Chapter 3 defines the setting of classic Few-shot 3D Point Cloud Semantic Segmentation and our task, Generalized Few-shot 3D Point Cloud Semantic Segmentation.
4. Chapter 4 demonstrates the proposed solution of the task, including the Base Model, the proposed Adaptive Support Enrichment module and the Query Aware Representation module, along with the new training strategy.

5. Chapter 5 shows the datasets, implementation details, and the results of our experiments.

6. Chapter 6 draws the conclusion of our work and lists the possible improved methods for future work.

Chapter 2

Literature Review

2.1 3D Semantic Segmentation

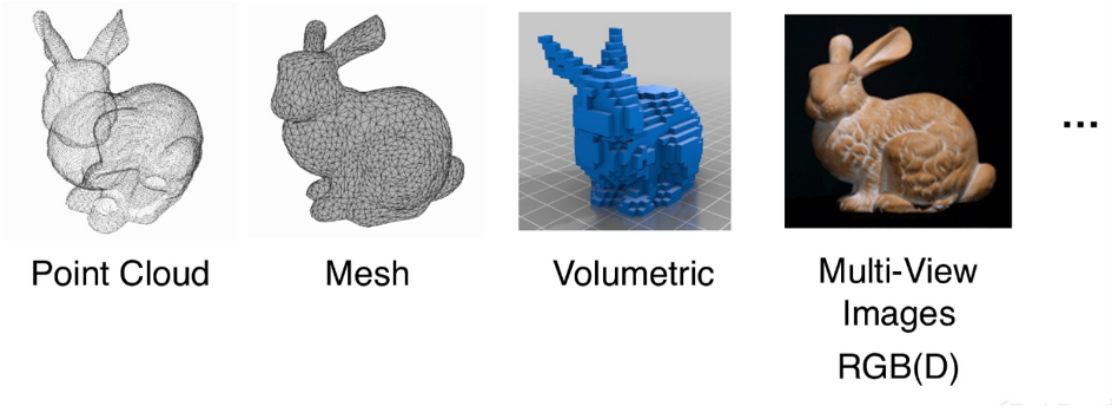

FIGURE 2.1: Several representation methods of 3D data.

The objective of 3D point cloud semantic segmentation is to accurately assign a unique label to each point in a 3D point cloud. These labels are selected from predefined classes, each corresponding to specific semantic meanings. Effective 3D point cloud semantic segmentation demands the comprehension of not only the global geometric structure but also the local aggregation feature of each 3D object.

Different from 2D images where pixels are organized in regular grids and amenable to classical convolution, 3D point clouds are intrinsic in irregular and unordered

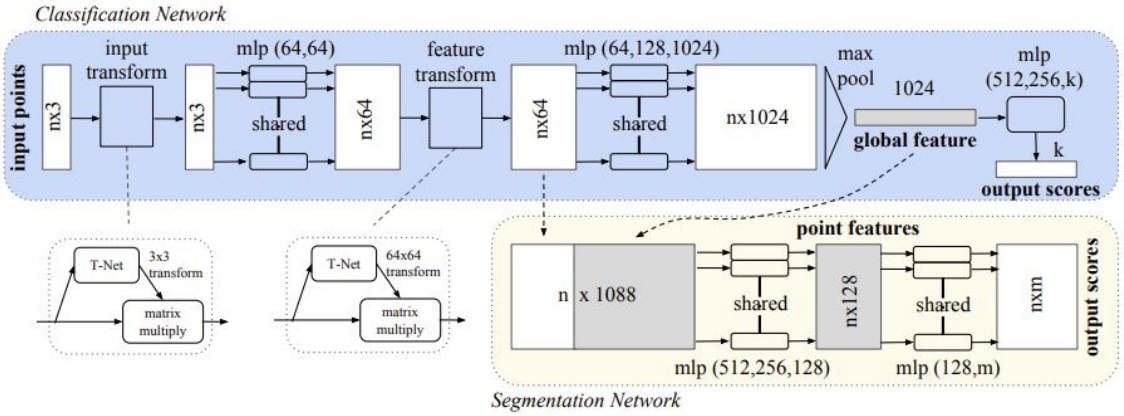

FIGURE 2.2: The architectures of the PointNet model [1]. PointNet employs several MLP layers to learn the feature from each point and utilizes a max pooling method to extract global geometric structure.

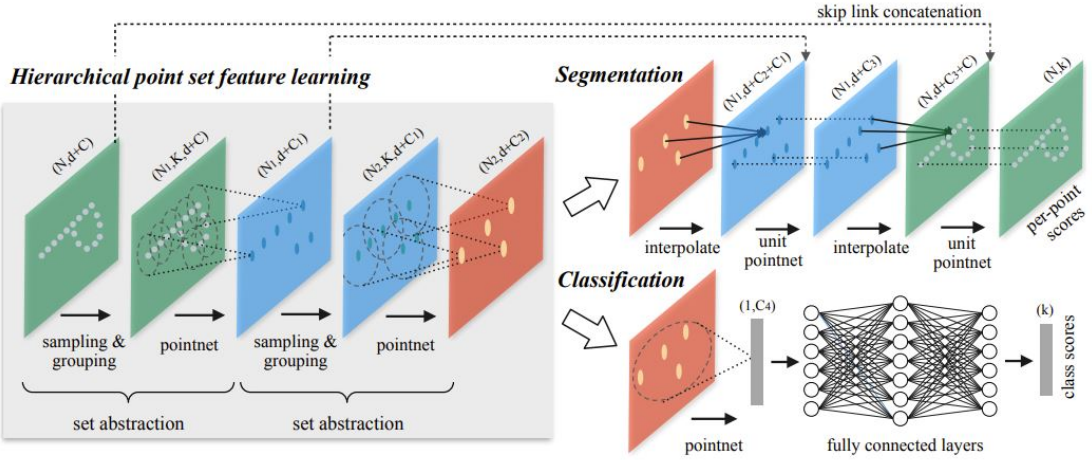

FIGURE 2.3: The architectures of the PointNet++ model [2]. PointNet++ uses a hierarchical spatial structure, which stacks multiple set abstraction levels.

forms. This unstructured nature poses significant challenges in processing and understanding 3D point clouds. To mitigate these challenges, indirect yet regular representations such as octree [21], kdtree [22], multi-view projections [23], voxel [24, 25] and mesh [26] have been proposed. Figure 2.1 illustrates 3D point cloud representation, including the point, mesh, voxel, and multi-view projections. However, with the thriving of advanced deep learning architectures, many contemporary approaches now focus on directly processing raw 3D points using neural networks, preserving the original 3D geometric information. The core issue in point cloud

analysis is how to learn the global geometric structure from the entirety of the 3D object while simultaneously aggregating its local features.

As the pioneer work, PointNet [1] respects the permutation invariance of points and designs a single symmetric function. PointNet employs several Multilayer Perceptron layers to learn the feature from each point and utilizes a max pooling method to extract global geometric structure. Figure 2.2 shows the architectures of the PointNet model. However, PointNet ignores the local structural information between points because features are learned independently from each point. Subsequently, PointNet++ [2] is designed to recognize local fine-grained patterns from the neighborhood of each point using a hierarchical spatial structure. The architecture involves stacking multiple set abstraction levels, with each level comprising three distinct layers: a sampling layer, a grouping layer, and a learning layer based on PointNet. The sampling layer is responsible for selectively choosing a subset of points, the grouping layer clusters these points based on their spatial proximity, and the PointNet-based learning layer extracts and processes the features from each group. The architectures of the PointNet++ model are showcased in Figure 2.3. In the following work, efficient sampling of the point set significantly enhances the performance of such models, and the development of various sampling strategies can be found in [27–30].

Numerous methods represent the point cloud with a graph and pass messages within this graph structure to aggregate global and local features. Edge-Conditioned Convolution (ECC) [31] treats each point as a graph vertex, interlinking vertices through directed edges. Here, dynamic edge-conditioned filters are employed, generating convolution kernels based on intra-cloud edges. Superpoint Graphs (SPG) [32] focuses on representing contextual relationships through a superpoint graph, i.e. superpoints and simple shapes. It forms an attributed directed graph, referred to as a superpoint graph, to capture the structural and contextual representation of the point cloud. KCNet [33] adopts a novel strategy to explore local geometric structures by defining a set of learnable points as kernels. Then, it applies recursive feature aggregation on a nearest-neighbor-graph to explore local high-dimensional features. RGCNN [34] employs a unique approach to graph construction, connecting each point in a 3D object with all others and dynamically updating the graph Laplacian matrix layer by layer. To promote feature similarity between adjacent vertices, the model incorporates a graph-signal smoothness before its loss

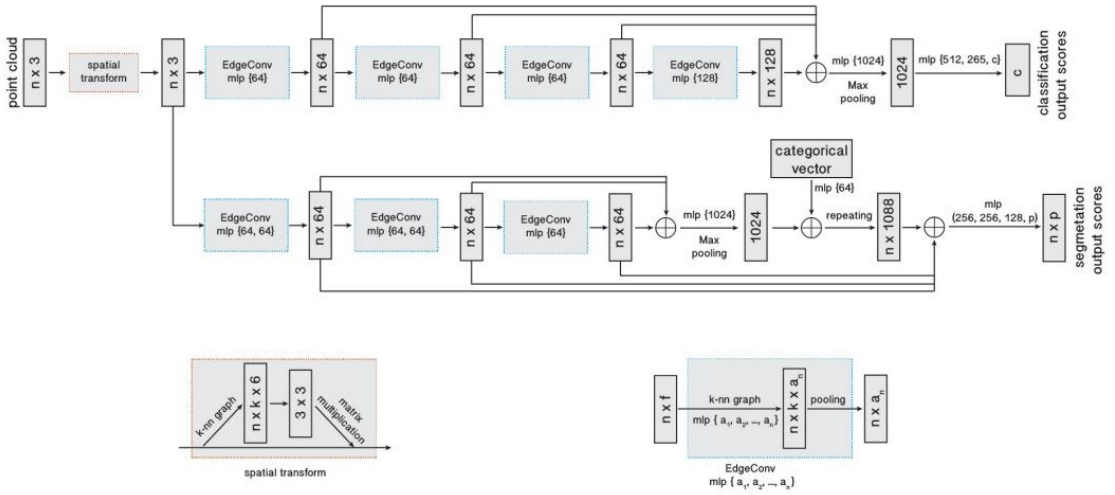

FIGURE 2.4: The overview of the DGCNN [3] model. The top branch of the model is designated for classification, while the bottom branch focuses on segmentation. The input of the model is n points, each being analyzed within an EdgeConv layer to generate an edge feature set of size k. These features within each set are then aggregated to produce EdgeConv responses for the respective points. The symbol $\bigoplus$ represents the concatenation process in this architecture.

function. The research in [35] introduces LocalSpecGCN, an innovative end-to-end spectral convolution network specifically tailored for local graphs, which are generated based on the k-nearest neighbor principle. DeepGCNs [36] focuses on leveraging the potential of increasing depth within graph convolutional networks. It borrows concepts from Convolutional Neural Networks such as residual/dense connections and tailors them to fit the context of GCNs. DGCNN [3] applies graph convolutions on k-nearest neighbors (kNN) graphs. A dynamic approach to graph construction is employed within the feature space, with the graph being updated following each network layer. The EdgeConv module, the core of this architecture, utilizes Multilayer Perceptron layers for edge feature learning and channel-wise symmetric aggregation. Figure 2.4 illustrates the architectures of the DGCNN model.

Additionally, many approaches employ continuous convolutions directly on the 3D point set to aggregate global and local features. RS-CNN [13] features the RS-Conv layer, which processes a local subset of points and applies convolution via a Multilayer Perceptron layer. This layer is designed to learn the transformation from low-level relational features, like Euclidean distance and relative position,

into high-level relational characteristics. SpiderCNN [37] utilizes a series of polynomial functions to define kernel weights for the convolution. Spherical CNN [38] introduces spherical convolution, specifically addressing 3D rotation equivariance challenges. It processes multi-valued spherical functions and parameterizes convolutional filters using anchor points in the spherical harmonic domain. In PointConv [14], convolution is seen as a Monte Carlo estimation of continuous 3D convolution, grounded in importance sampling. The convolutional kernels in this framework are composed of two components: a weighting function learned through Multilayer Perceptron layers, and a density function determined via kernelized density estimation and additional Multilayer Perceptron layers. KPConv [39] develops learnable convolution weights for 3D point clouds that are contingent on input coordinates. PointCNN [12] learns an X-transformation from the input points into a latent order, executed through Multilayer Perceptron layers, and then applies standard convolutional operators.

Although the aforementioned networks have achieved a promising performance in the fully-supervised 3D semantic segmentation task, they lose their effectiveness when encountering new categories without enough labeled samples. Consequently, our study focuses on generalized few-shot learning applied to 3D semantic segmentation, generalizing segmentation across base and novel classes simultaneously. Considering both the accuracy and efficiency, we apply the most common network structure DGCNN [3] as our feature extractor to capture both global and local features.

2.2 Few-shot Learning

Few-shot learning, an important domain in deep learning, aims to classify novel classes when only a limited number of training samples with supervised information are available [40]. To address this challenging task, several meta-learning approaches [20, 41–46] have proposed to learn 'how to learn' to enhance the generalization capabilities of the model. Few-shot Learning models can be roughly categorized into three types: model-based, metric-based, and optimization-based methods. Specifically, *Metric-based* method is of particular interest, which involves learning an effective metric function capable of generating an embedding space of similarity to effectively represent the correlation between labeled and unlabeled

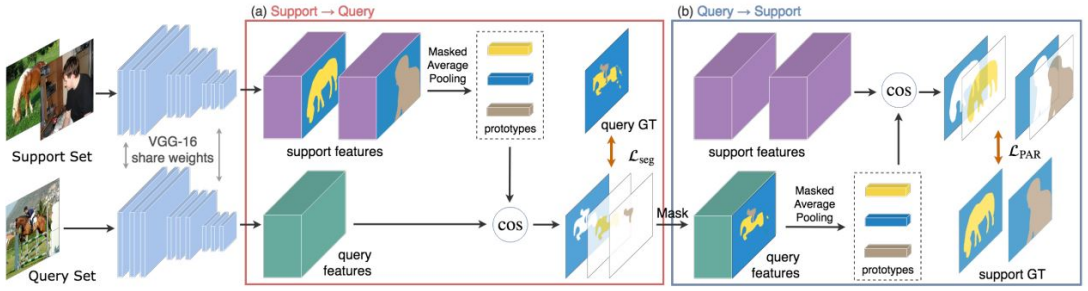

FIGURE 2.5: The architecture of PANet [4] model. Block (a) outlines the support-to-query few-shot segmentation process of PANet. Initially, deep feature representations are extracted from both support and query sets. Subsequently, prototypes are derived through masked average pooling. The segmentation of the query images is conducted by calculating the cosine distances (marked as *cos* in the figure) between each prototype and the spatially located query features. Block (b) introduces the Prototype Alignment Refinement (PAR) mechanism, which facilitates the alignment of support and query prototypes via a query-to-support few-shot segmentation.

data. Key examples include the Matching Network [20] and Prototypical Network [44]. The support and query sets are projected into a space by neural network feature extractors. Subsequently, they use a non-parametric approach for class prediction of the query samples, guided by the information from the support set. The Matching Network builds distinct encoders for support and query sets, and the final results are the weighted sum between these sets. Prototype Networks, on the other hand, are based on the assumption that each class can be represented by a prototype, which is the mean feature vector of the support samples in the embedding space. Thus, the classification and segmentation turn into a nearest-neighbor problem within this space.

2.3 Few-shot Segmentation

Few-shot segmentation presents a complex challenge based upon the concept of few-shot classification: it requires performing detailed pixel classification [47, 48] on entirely new classes, with the model having access to only a few support examples for guidance. OSLSM [49] pioneers few-shot segmentation by developing a classical two-branch method to learn the generation of classifier weights specific to

each class, following the pipeline of Siamese network [50]. Following this, Prototypical Learning (PL) [51] integrates the concept of prototype [44] into segmentation. PL focuses on creating a distinct prototype for each class from the support set and employs cosine similarity to make accurate predictions by comparing these class-specific prototypes with the pixels in the query images. SG-One [52] introduces masked average pooling to create object-related prototypes, serving as the foundational technology in the development of subsequent methods. More recently, PANet [4], introduces prototype alignment regularization to harness the knowledge inherent in support data and foster consistent embedding of prototypes. Figure 2.5 provides a detailed view of the architectures of the PANet model. CANet [53], built on the method with masked average pooling, further extends the prototype to align with the query feature's size and then concatenates them. Besides, an iterative optimization module is integrated within CANet to refine the segmentation results further. PGNet[54] and BriNet[55] have each introduced a notable advancement in the realm of feature connections by establishing dense pixel-to-pixel links between support and query features. PPNet [56] and PMMs [57] share a similar approach to split objects into multiple parts, which extract fine-grained and diverse representative features. This methodology allows for a more nuanced understanding of object structures. PFENet [58], on the other hand, introduces an innovative method of generating training-free prior masks and Feature Enrichment Module. This approach effectively addresses spatial inconsistencies by augmenting query features with these prior masks and support features, enhancing the overall segmentation process.

2.4 Few-shot 3D Point Cloud Semantic Segmentation

Few-shot 3D Point Cloud Semantic Segmentation adapts the semantic segmentation of 3D point clouds to the few-shot learning setting. In this setting, the model leverages the support set to perform segmentation on novel classes with sparse labeling. Zhao *et al.* [5] were the first to establish a baseline for this segmentation task and further introduced the attention-aware multi-prototype transductive approach. This method integrates multiple prototypes with a label propagation technique, as shown in Figure 2.6. However, the complexity of *attMPTI* and the

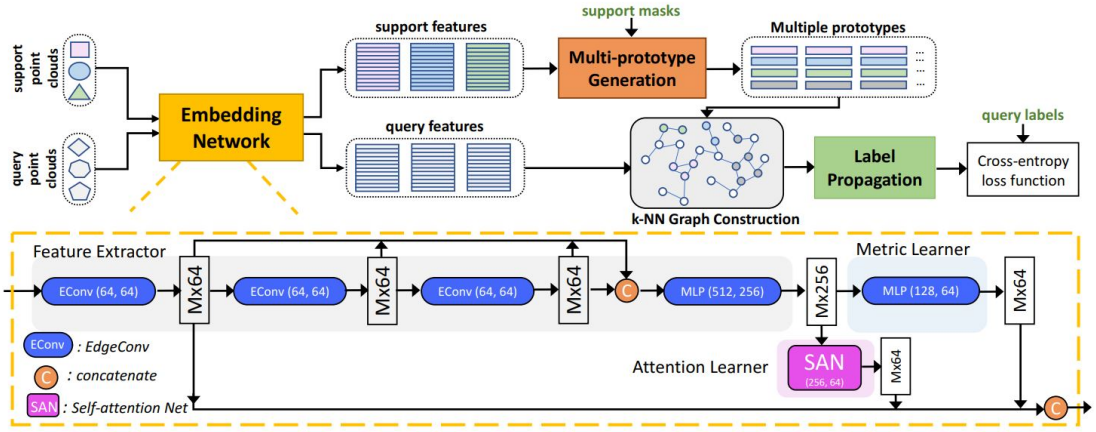

FIGURE 2.6: The architecture of attMPTI [5] method. Multiple-prototypes method is proposed to represent different classes. Then, a transductive label propagation technique is applied to harness the relationships between the labeled multi-prototypes and the unlabeled points. Additionally, an attention-aware multi-level feature learning network is specifically developed to effectively capture both geometric and semantic correlations among the points.

time-intensive nature of its graph construction for few-shot segmentation limits its effectiveness. He *et al.* [19] notices a significant image-gap in 3D objects with identical labels but from different objects, because of the absence of a robustly pretrained 3D model. To address this, they introduced the Query-Guided Prototype Adaption (QGPA) and a self-reconstruction module for enhanced prototype representation for each class.

Despite these advancements, the inherent problem of the 3D-FS task still exists, namely the lack of segmentation on base classes, and limits its applicability in real-world scenarios. Consequently, our research shifts focus to Generalized Few-shot 3D Point Cloud Semantic Segmentation.

Chapter 3

Problem Definition

3.1 Revisit the Classic Few-Shot Setting

In defining our protocols for few-shot point cloud semantic segmentation, we adhere to the methodologies established in prior work [5]. In the established Few-Shot Segmentation setting, the dataset is divided into two disjoint parts: base classes, denoted as C_{b}, and novel classes, denoted as C_{n}, with no overlap between them ($C_{\mathrm{b}} \cap C_{\mathrm{n}} = \varnothing$). The number of base classes and novel classes are represented as N_{b} and N_{n}, respectively. The training dataset D_{train} is comprised of samples from C_{b}, while the testing dataset D_{test} is formed from C_{n}. The segmentation model is trained with D_{train} during the training phase. Subsequently, its performance is evaluated on D_{test} in the testing phase. Each point cloud $P \in \mathbb{R}^{n\times(3+f_o)}$ includes n points of the coordinate information (X, Y, Z) and additional features f_o, such as normalized coordinate and RGB values.

To better adapt to novel classes, the episodes paradigm [20] has been introduced as a training and evaluation method for few-shot learning models. Each episode, whether for training or testing, consists of an annotated support set S and a query set Q from the same classes. Formally, the training set is represented as $D_{\mathrm{train}} = \{(S, Q)\}^{n_{\mathrm{train}}}$, and the testing set as $D_{\mathrm{test}} = \{(S, Q)\}^{n_{\mathrm{test}}}$. Here, n_{train} and n_{test} signify the complete number of episodes allocated respectively for the training and testing phases.

In each N-way K-shot (N base/novel classes with K support samples for each base/novel class) episode, one support set S contains pairs of $S_i \in \{S_1, S_2, ..., S_N\}$ for N unique classes. Each S_i consists of K pairs of support point clouds $P_{\mathrm{s}}^{k,i}$ and its corresponding mask $M_{\mathrm{s}}^{k,i}$ of the class c^i, defined as $S_i = \left\{(P_{\mathrm{s}}^{k,i}, M_{\mathrm{s}}^{k,i})_{k=1}^{K}\right\}$. The query set $Q = \left\{(P_{\mathrm{q}}^{t}, L^t)_{t=1}^{T}\right\}$ encompasses T pairs of query point clouds P_{q}^{t} and the ground-truth label L^t. Note that the support set and the query set in one episode are constructed from the same target classes.

The objective of N-way K-shot point cloud semantic segmentation is to train a model $\mathcal{M}(Q, S)$ that predicts the label $\hat{L}^t$ for any query point cloud P_{q}^{t} from the query set Q, leveraging insights from the support set S. The input data batch to the model $\mathcal{M}$ is the query-support pair $\{Q; S\} = \left\{(P_{\mathrm{q}}^{t})_{t=1}^{T}; ((P_{\mathrm{s}}^{k,i}, M_{\mathrm{s}}^{k,i})_{k=1}^{K})_{i=1}^{N}\right\}$ and the ground-truth label L^t serves as benchmarks for evaluating predictions in each episode.

3.2 The Problems in Few-shot Setting

To summarize, the classic Few-shot Segmentation task revolves around two foundational rules. First, it mandates that samples from testing classes C_{n} remain unseen during the training phase, to impartially assess the model's generalization ability. Second, it necessitates support samples corresponding to the target classes, to provide the model with essential prior information for predicting labels on query samples.

Despite its efficacy in solving the N-way K-shot problem, this approach has its drawbacks:

1. Criterion (1) limits the evaluation of the model's generalization to only the C_{n} classes with restricted samples while overlooking the comprehensive learning from C_{b} classes during training. This narrow evaluation scope might not fully capture the model's overall adaptability.

2. Since users might lack precise knowledge of the exact classes and the corresponding quantity contained in each testing sample, this setting often proves to be impractical in many scenarios according to criterion (2). Consequently,

this assumption poses a significant challenge in providing the model with suitable support samples mirroring the classes in the query samples.

3. Even though users already know the existence of N classes in the test samples, traditional few-shot models [4, 5] often require processing $N * K$ extra manually chosen support point clouds/labels as preliminary information before making predictions of overall potential classes for the testing point clouds. However, this method is not ideal for practical situations. In real-world applications, models are expected to predict across all classes in the test point clouds independently but not rely on the manual input of support samples.

3.3 The Proposed Generalized Few-Shot Setting

In this paper, we introduce the Generalized Few-Shot Semantic Segmentation (GFS) setting, which serves as an expansion of the traditional few-shot segmentation framework. The primary difference in the GFS setting is its aim to concurrently evaluate the generalization capabilities of the model over both base and novel classes after training, which aligns more closely with actual real-world conditions scenarios.

Similar to the conventional setup, our dataset comprises two mutually exclusive sets of classes: base classes C_{b} of number N_{b} and novel classes C_{n} of number N_{n}, ensuring $C_{\mathrm{b}} \cap C_{\mathrm{n}} = \varnothing$. Base classes C_{b} are characterized by an abundance of labeled training data, while each novel class C_{n} is limited to K labeled samples (e.g., $K = 5, 10, 30$). Each point cloud $P \in \mathbb{R}^{n \times (3+f_o)}$ includes n points of the coordinate information (X, Y, Z) and additional features f_o.

The GFS methodology encompasses three phases. To enhance clarity and distinguish between the stages of the training process, we designate the data used in the first phase as the 'base sample', in the second phase as the 'support samples', and in the third as the 'query samples'. Firstly, in the *base prototype generation phase*, the 3D-GFS model is trained solely on the base set $D_{\mathrm{b}}^{\mathrm{train}}$ constructed only from well-labeled base classes C_{b} to develop robust feature representations and derive base class prototypes. Specifically, $D_{\mathrm{b}}^{\mathrm{train}} = \left\{(P_{\mathrm{b}}^{i}, M_{\mathrm{b}}^{i})_{i=1}^{n_{\mathrm{train}}}\right\}$, where n_{train} is the number of samples in $D_{\mathrm{b}}^{\mathrm{train}}$. P_{b}^{i} is a base point cloud, while M_{b}^{i} is the corresponding annotation. Secondly, during the *novel prototype registration phase*, the model is exposed

to N_{n} novel classes from C_{n} and each with K limited support samples to create novel class prototypes. The support set $D_{\mathrm{s}}^{\mathrm{train}} = \left\{ \{(P_{\mathrm{s}}^{k,i}, M_{\mathrm{s}}^{k,i})_{k=1}^{K}\}_{i=1}^{N_{\mathrm{n}}} \right\}$, where $P_{\mathrm{s}}^{k,i}$ is a support point cloud and $M_{\mathrm{s}}^{k,i}$ is its binary mask. In the *final evaluation phase*, 3D-GFS model will predict the labels of the query set $D^{\mathrm{test}} = \left\{ \left(P_{\mathrm{q}}^{i}, M_{\mathrm{q}}^{i}\right)_{i=1}^{n_{\mathrm{test}}} \right\}$, which built from the both base and novel classes $(C_{\mathrm{b}} \cup C_{\mathrm{n}})$, where n_{test} is the number of query samples. For an in-depth exploration of the three distinct phases, refer to Chapter 4, where we present a comprehensive overview of our methodological framework.

Apparently, the adoption of the episodic training paradigm in the 3D-GFS task presents limitations, since prior information from the support set becomes redundant. Additionally, the data format of the query set deviates from the query-support pairs structure of episodic training.

Chapter 4

Methodology

In this part, we introduce the approach we propose. Initially, the definition of the prototype learning network is presented first in Sec. 4.1. Subsequently, Sec. 4.2 provides a detailed description of the 3D-GFS Base Model. We then explore the core components of our model in detail: Sec. 4.3 discusses the Adaptive Support Enrichment (ASE) module, Sec. 4.4 focuses on the Query Aware Representation (QAR) module, and Sec. 4.5 explains the proposed training strategy.

4.1 Prototype Learning Network

Inspired by the classic Few-Shot learning framework, our work adopts the commonly used meta-learning strategy, prototypical network [44, 59]. The foundational idea of the prototypical network is the existence of an embedding space where points from the same category cluster around a central prototype. This prototype represents the average vector of the samples from that category in that space. The segmentation decisions are based on the category of the nearest prototype within the embedding space.

To appreciate the concept of the prototypical network, it is essential to understand the basic process of the N-way K-shot few-shot segmentation task. Given the support set $S = \left\{((P_{\mathrm{s}}^{k,i}, M_{\mathrm{s}}^{k,i})_{k=1}^{K})_{i=1}^{N}\right\}$, let $\mathcal{F}(\cdot)$ be the shared feature extractor processing point $P_{\mathrm{s}}^{k,i}$. The prototypes p^i of the class c^i ($i \in \{1, 2, ..., N\}$) from the

support set is obtained by masked average pooling (MAP) over K shots:

$$p^i = \frac{1}{K} \times \sum_{k=1}^{K} \frac{\sum_n [M_{\mathrm{s}}^{k,i} \times \mathcal{F}(P_{\mathrm{s}}^{k,i})]}{\sum_{\mathrm{n}} M_{\mathrm{s}}^{k,i}}, \ (i \in \{1, 2, ..., N\}) \tag{4.1}$$

where $P_{\mathrm{s}}^{k,i}$ denotes the k-th support samples of class c^i, $M_{\mathrm{s}}^{k,i}$ signifies the binary mask corresponding to class c^i over the feature $\mathcal{F}(P_{\mathrm{s}}^{i,k})$, and n is the count of total points in the k-th support samples of class c^i.

The method of non-parametric metric learning is always employed to learn the optimal prototypes and conduct segmentation. Given that segmentation essentially equates to a point-wise classification task, the few-shot segmentation method measures the distance between the feature vector at each query point and the established prototypes. Following this, a softmax function is utilized to these distances, generating a probability map $\widetilde{M}_{\mathrm{q}}^t$ that covers all semantic classes:

$$\widetilde{M}_{\mathrm{q}}^{t,i} = \frac{exp(-\phi(\mathcal{F}(P_{\mathrm{q}}^t), p^i))}{\sum_{i=1}^{N} exp(-\phi(\mathcal{F}(P_{\mathrm{q}}^t), p^i))}, \ (i \in \{1, 2, ..., N\}) \tag{4.2}$$

where P_{q}^t represents the query sample, $\mathcal{F}(\cdot)$ is the shared feature extractor for both the support and query set, and p^i is the prototype of the class c^i. In this context, cosine similarity [5, 60] is utilized as the distance metric ϕ.

Then, the estimated query mask $\hat{L}^t$ is determined as follows:

$$\hat{L}^t = \arg\max_i \widetilde{M}_{\mathrm{q}}^{t,i}, \ (i \in \{1, 2, ..., N\}), \tag{4.3}$$

Fig. 4.1 provides an overview of the conventional 3D-FS model. This model generates prototypes through mask average pooling based on prior information from the support set, and then predicts the labels of the query set.

4.2 The Base Model of 3D-GFS

3D-FS framework primarily focuses on prototype formation and target identification within novel classes. In contrast, 3D-GFS extends this requirement to both

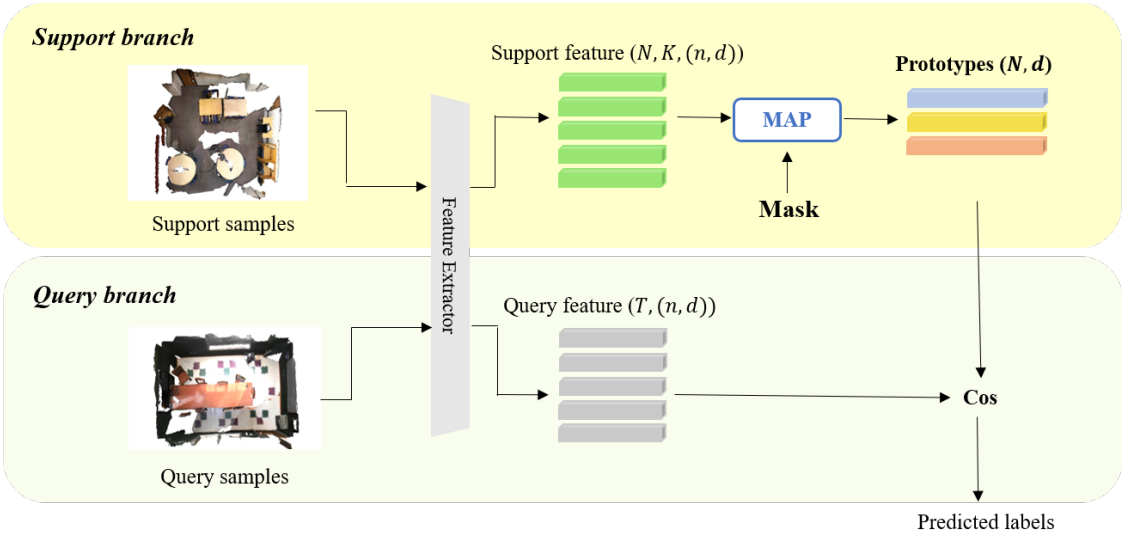

FIGURE 4.1: The overview of the conventional 3D-FS model. Each input to the prototypical network forms an N-way K-shot learning episode. Each episode includes a support set with N distinct classes and K samples per class, along with T query samples. Both support and query point clouds are embedded into deep feature spaces via a feature extractor with shared weights. Prototypes are generated through masked average pooling (MAP) applied to the support features. Here, n denotes the total number of input points, d represents the dimension of features and prototypes, and 'Cos' refers to cosine similarity.

base and novel classes simultaneously. Creating prototypes for base classes as Eq. (4.1) is notably challenging and inefficient for the 3D-GFS model. This inefficiency stems from the need to input all samples from each base class into the feature extractor, and it becomes increasingly demanding with larger training sets. Consequently, we implement distinct methods tailored to the specific phases of the 3D-GFS framework, named **Base Model**, optimizing the model's performance across different class types. The overview of the proposed 3D-GFS Base Model is shown in Figure 4.2.

Base prototype generation phase. In the prototypical network, the classifier weights are learned as class prototypes for each class. The d-dimensional base class prototypical classifiers $P_{\text{b,cls}} = \left\{p^i_{\text{b,cls}}\right\}_{i=1}^{N_\text{b}} \in \mathbb{R}^{N_\text{b} \times d}$ are learned using back-propagation approach via a cross-entropy loss:

$$\widetilde{M}^{j,i}_\text{b} = \frac{exp(-\phi(\mathcal{F}(P^j_\text{b}), p^i_{\text{b,cls}}))}{\sum_{i=1}^{N_\text{b}} exp(-\phi(\mathcal{F}(P^j_\text{b}), p^i_{\text{b,cls}}))}, \ (i \in \{1, 2, ..., N_\text{b}\}) \tag{4.4}$$

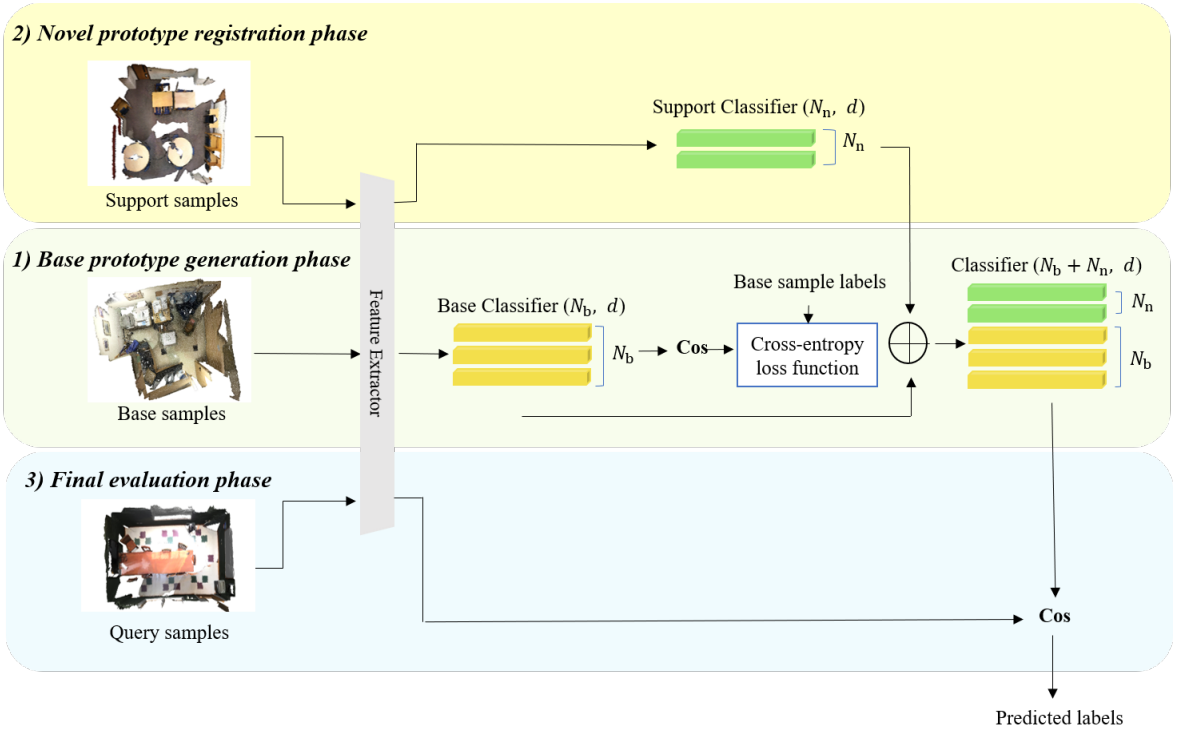

FIGURE 4.2: The overview of the proposed 3D-GFS Base Model. There are three phases in our 3D-GFS model: *base prototype generation phase*, *novel prototype registration phase* and *final evaluation phase*. The d-dimensional base class prototypical classifiers $P_{\text{b,cls}} \in \mathbb{R}^{N_\text{b} \times d}$ are learned using back-propagation approach via a cross-entropy loss. The novel class prototypical classifiers $P_\text{n} \in \mathbb{R}^{N_\text{n} \times d}$ are formed by masked average pooling (MAP) over N_n-way K-shot support samples. The classifiers $P_\text{all} \in \mathbb{R}^{(N_\text{b}+N_\text{n}) \times d}$ of all classes are adopted to evaluate the query samples by concatenating the base class classifiers $P_{\text{b,cls}}$ and the novel class classifiers P_n. d denotes the dimension of classifiers and 'Cos' denotes cosine similarity.

where $\widetilde{M}_\text{b}^{j,i}$ denotes the probability map of the point cloud sample P_b^j and the prototype $p_{\text{b,cls}}^i$, the term $\mathcal{F}(\cdot)$ refers to the shared feature extractor which is applied to the base samples $D_\text{b}^\text{train} = \left\{ \left(P_\text{b}^j, M_\text{b}^j \right)_{j=1}^{n_\text{train}} \right\}$. The function ϕ indicates cosine similarity, a measure adopted in line with the methodologies presented in [5, 60].

Then, the estimated base mask $\hat{M}_\text{b}^j$ is determined as follows:

$$\hat{M}_\text{b}^j = \arg\max_i \widetilde{M}_\text{b}^{j,i}, \ (i \in \{1, 2, ..., N_\text{b}\}), \tag{4.5}$$

Novel prototype registration phase. Following the 3D-FS in 4.1, the novel class prototypical classifiers $P_\text{n} = \{p_\text{n}^i\}_{i=1}^{N_\text{n}} \in \mathbb{R}^{N_\text{n} \times d}$ are formed by masked average

pooling (MAP) over N_{n}-way K-shot support samples:

$$p_{\mathrm{n}}^{i} = \frac{1}{K} \times \sum_{k=1}^{K} \frac{\sum_{n}[M_{\mathrm{s}}^{k,i} \times \mathcal{F}(P_{\mathrm{s}}^{k,i})]}{\sum_{n} M_{\mathrm{s}}^{k,i}}, \ (i \in \{1, 2, ..., N_{\mathrm{n}}\}). \tag{4.6}$$

Note that $\mathcal{F}(\cdot)$ is the fixed feature extractor well-trained in the *base prototype generation phase*. n is the count of total points in the k-th support samples of the novel class c^i

Final evaluation phase. In the last phase, the classifiers $P_{\mathrm{all}} = \{p^i\}_{i=1}^{(N_{\mathrm{b}}+N_{\mathrm{n}})} \in \mathbb{R}^{(N_{\mathrm{b}}+N_{\mathrm{n}})\times d}$ of all classes are adopted to evaluate the query samples by concatenating the base class classifiers $P_{\mathrm{b,cls}}$ and the novel class classifiers P_{n}:

$$P_{\mathrm{all}} = concat(P_{\mathrm{b,cls}}, P_{\mathrm{n}}). \tag{4.7}$$

The technique of non-parametric metric learning is applied to determine the distance between the feature vector at every query point and each established prototype. Then, a probability map $\widetilde{M}_{\mathrm{q}}^{j}$ is generated by softmax function to these distances,

$$\widetilde{M}_{\mathrm{q}}^{j,i} = \frac{exp(-\phi(\mathcal{F}(P_{\mathrm{q}}^{j}), p^i))}{\sum_{i=1}^{(N_{\mathrm{b}}+N_{\mathrm{n}})} exp(-\phi(\mathcal{F}(P_{\mathrm{q}}^{j}), p^i))}, \ (i \in \{1, 2, ..., (N_{\mathrm{b}} + N_{\mathrm{n}})\}) \tag{4.8}$$

Then, the estimated query mask $\hat{M}_{\mathrm{q}}^{j}$ is determined as follows:

$$\hat{M}_{\mathrm{q}}^{j} = \arg\max_{i} \widetilde{M}_{\mathrm{q}}^{j,i}, \ (i \in \{1, 2, ..., (N_{\mathrm{b}} + N_{\mathrm{n}})\}), \tag{4.9}$$

Fig. 4.3 shows an overview of the proposed 3D-GFS training model that includes the proposed Adaptive Support Enrichment (ASE) and Query Aware Representation (QAR) modules in the base framework.

4.3 Adaptive Support Enrichment Module

Motivation. The fundamental distinction between the 3D-GFS and 3D-FS lies in the categories that are required to be predicted during evaluation. This difference explains the reason why the performance of employing a 3D-FS network directly

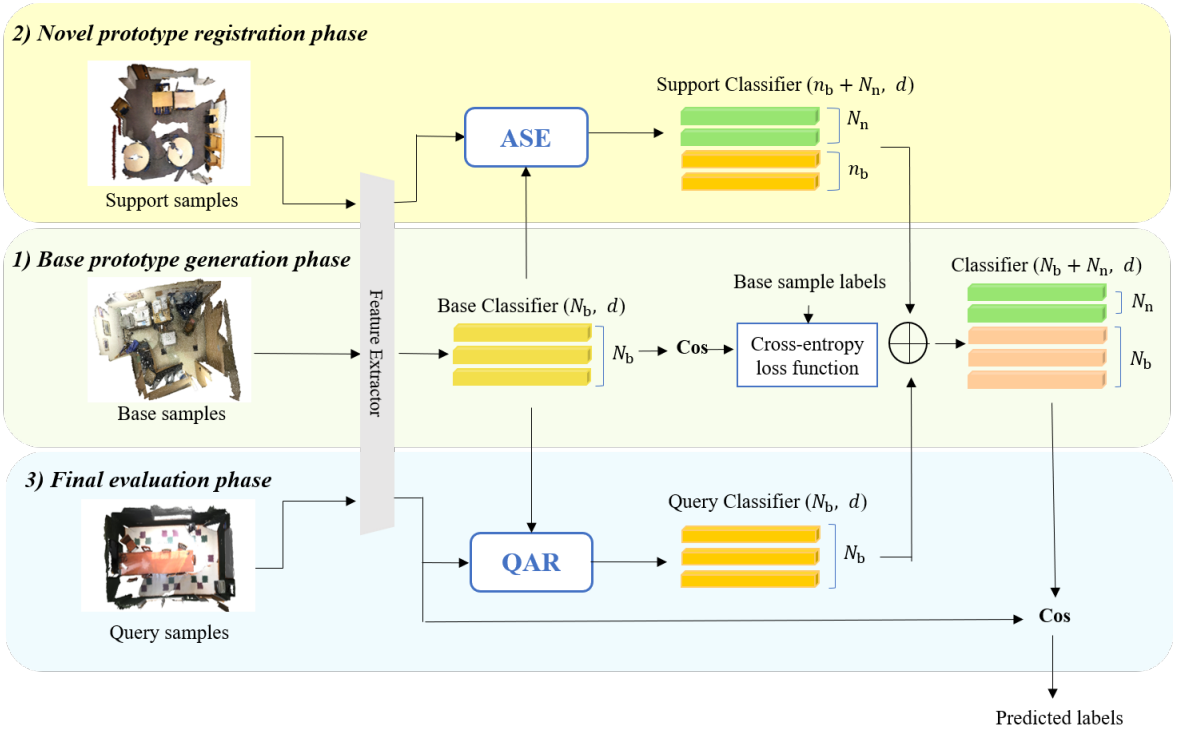

FIGURE 4.3: The overview of the proposed 3D-GFS model. There are three phases in our 3D-GFS model: *base prototype generation phase*, *novel prototype registration phase* and *final evaluation phase*. We utilize a Dynamic Graph CNN for Learning on Point Clouds (DGCNN) with shared weights to embed base, support and query point clouds into high-dimensional deep features. During the second phase, the ASE module updates the base classifiers using base classes with n_b number of classes and captures the prototypes of the novel classes with N_n number of classes in the support samples. The QAR module modifies the whole base classifiers with query samples in the third phase. d denotes the dimension of classifiers and 'Cos' denotes cosine similarity.

in a 3D-GFS setting is not ideal. In the 3D-FS task, the model is trained towards distinguishing N novel classes and the background in the test dataset. These novel classes are randomly selected from unseen class sets, while all base classes are treated as background. This leads to a lack of crucial co-occurrence interaction between novel and base classes. In contrast, the 3D-GFS framework involves predictions for both the base and novel classes within query samples. Thus, understanding the interactions between these two class types is crucial for the model's generalization capabilities.

The contextual information in the semantic segmentation task is very important [61, 62]. However, our proposed Base Model that develops prototypes for base

classes only captures the contextual feature within these classes, but ignores the interplay between base and novel classes. This is primarily because the novel class prototypes, formed through masked average pooling over support samples, may omit crucial base class information. Consequently, incorporating contextual features to enrich base class representation is imperative for enhancing the efficacy of the 3D-GFS model.

Adaptive Support Enrichment Module (ASE). During the *novel prototype registration phase*, ASE module captures the essential co-relationship between base classes and novel classes in support samples, then based on that updates the base prototypical classifier $P_{\mathrm{b,cls}}$ formed in the first phase.

Let n_{b} ($n_{\mathrm{b}} \leq N_{\mathrm{b}}$) be the total number of base classes in a N_{n}-way K-shot support sample. Specifically, support samples are first utilized to form novel classifiers P_{n} via Eq. (4.1). At the same time, the support classifiers $p^i_{\mathrm{b,sup}}$ contained in the support samples are calculated as:

$$p^i_{\mathrm{b,sup}} = \frac{\sum_{j=1}^{N^{\mathrm{n}}} \sum_{k=1}^{K} \sum_n [M_{\mathrm{s}}^{k,i} \times \mathcal{F}(P_{\mathrm{s}}^{k,i})]}{\sum_{j=1}^{N^{\mathrm{n}}} \sum_{k=1}^{K} \sum_n M_{\mathrm{s}}^{k,i}}, \ (i \in \{1, ..., n_{\mathrm{b}}\}) \tag{4.10}$$

where n is the count of total points in the k-th support samples of the base class c^i_{b}. Then, the new classifier $p^i_{\mathrm{b,adp}}$ of c^i_{b} is obtained through weighted sum of the original base classifier $p^i_{\mathrm{b,cls}}$ from the *base class learning phase* and support base classifier $p^i_{\mathrm{b,sup}}$ from the *novel class registration phase*, which can be written as:

$$p^i_{\mathrm{b,adp}} = \lambda^i_{\mathrm{sup}} \times p^i_{\mathrm{b,cls}} + (1 - \lambda^i_{\mathrm{sup}}) \times p^i_{\mathrm{b,sup}}, \ (i \in \{1, ..., n_{\mathrm{b}}\}) \tag{4.11}$$

where λ^i_{sup} is the adaptive parameter to balance the impacts of the original and support base classifier. λ^i_{sup} is conditioned on distinct classes of original and support classifier, which is calculated as:

$$\lambda^i_{\mathrm{sup}} = \xi_{\mathrm{sup}}(p^i_{\mathrm{b,cls}}, p^i_{\mathrm{b,sup}}) \tag{4.12}$$

where ξ_{sup} is implemented by a few MLP layers in our model.

4.4 Query Aware Representation Module

Motivation. ASE module provides hints to the classifiers from support samples during the *novel prototype registration phase*. However, the hints lead to bias towards certain base classes contained in support samples and affect the model's capacity to generalize effectively. Therefore, we design a Query Aware Representation module in the *final evaluation phase*, which dynamically modifies classifier representation by being aware of the prior information derived from query samples.

Query Aware Representation module (QAR). Specifically, the soft mask $Y_{\text{qry}} \in \mathbb{R}^{n\times(N_\text{b}+N_\text{n})}$ for query samples is generated from the original base classifiers $P_{\text{b,cls}}$ and the novel classifier P_n, where n represents the number of points in a single sample. The categorical representation $P_{\text{b,qry}} = \left\{p^i_{\text{b,qry}}\right\}_{i=1}^{N_\text{b}} \in \mathbb{R}^{N_\text{b}\times d}$ of the query base classifier is computed by the combination of the query feature $\mathcal{F}(P_\text{q})$ and the Softmax function along the second dimension of Y_{qry}:

$$p^i_{\text{b,qry}} = \left[\text{Softmax}(Y_{\text{qry}}) \times \mathcal{F}(P_\text{q})\right]^i,\ (i \in \{1, ..., N_\text{b}\}). \tag{4.13}$$

Similar to Eq. (4.11), the query classifiers $p^i_{\text{b,awr}}$ are updated by a weighted sum of the original base classifiers $p^i_{\text{b,cls}}$ and the query base classifiers $p^i_{\text{b,qry}}$:

$$p^i_{\text{b,awr}} = \lambda^i_{\text{qry}} \times p^i_{\text{b,cls}} + (1 - \lambda^i_{\text{qry}}) \times p^i_{\text{b,qry}},\ (i \in \{1, ..., N_\text{b}\}), \tag{4.14}$$

where λ^i_{qry} serves as a coefficient to maintain equilibrium between the original and query base classifiers. λ^i_{qry} is calculated by:

$$\lambda^i_{\text{qry}} = \xi_{\text{qry}}(p^i_{\text{b,cls}}, p^i_{\text{b,qry}}) \tag{4.15}$$

where ξ_{qry} is implemented by a few MLP layers in our model. Accordingly, the base classifiers are:

$$p^i_\text{b} = p^i_{\text{b,adp}} + p^i_{\text{b,awr}},\ (i \in \{1, ..., N_\text{b}\}), \tag{4.16}$$

which are used for prediction in the *final evaluation phase*.

4.5 Training Strategy

The direct application of Eq. (4.10-4.12) is challenging due to the distinct embedding spaces of the averaged features yielded by feature extractor $\mathcal{F}(\cdot)$ and the original base classifier $P_{\text{b,cls}}$ and the novel class classifiers P_{n}. Accordingly, we propose a new training strategy in the *base class learning phase*, trying to imitate scenarios of actual novel and base classes within support samples.

Note that all the selected samples below are in base classes. Firstly, the training samples are randomly divided into two equal halves: one set is designated as 'Pseudo Support' and the rest as 'Pseudo Query'. 'Pseudo Support' is utilized to form support prototypes, aiming to provide prior information for 'Pseudo Query' segmentation. Secondly, let N_{b}^{PS} donate the count of base classes in 'Pseudo Support' samples. Consequently, a random half of these are chosen as 'Pseudo Novel' classes C_{b}^{PN} while the remaining are designated as the 'Pseudo Base' classes C_{b}^{PB}. The 'Pseudo Novel' classes C_{b}^{PN} and the 'Pseudo Base' classes C_{b}^{PB} in 'Pseudo Support' samples are exploited to imitate the roles of novel classes and base classes in the *novel prototype registration phase* respectively. The updated base classifier can be written as:

$$p_{\text{b}}^{i} = \begin{cases} \lambda_{\text{sup}}^{i} \times p_{\text{b,cls}}^{i} + (1 - \lambda_{\text{sup}}^{i}) \times p_{\text{b,sup}}^{i} & c^{i} \in C^{PB} \\ p_{\text{b,sup}}^{i} & c^{i} \in C^{PN} \\ p_{\text{b,cls}}^{i} & \textit{others if any.} \end{cases} \tag{4.17}$$

Specifically, the 'Pseudo Base' classifiers are updated by the original base classifier $p_{\text{b,cls}}^{i}$ with the parameter λ_{sup}^{i} while the 'Pseudo Novel' classifier is directly used to replace the original one.

Chapter 5

Experiments and Results

This chapter begins with the evaluation of our model on two extensive datasets, namely S3DIS and ScanNet in Sec. 5.1. Subsequently, in Sec. 5.2 we give a comprehensive description of our experimental setup. Additionally, Sec. 5.3 undertake detailed ablation studies to thoroughly validate the efficacy of each component of our approach. Concluding this chapter, in Sec. 5.4, we analyze the performance of the 3D Few-Shot (3D-FS) method when adapted to our 3D Generalized Few-Shot (3D-GFS) setting, thereby examining the scalability and ability to generalize our proposed method.

5.1 Datasets and Setup

5.1.1 Datasets

Following 3D-FS[5], our evaluation of the proposed approaches is conducted on two public 3D point clouds datasets:

S3DIS. Stanford Large-Scale 3D Indoor Spaces Dataset (S3DIS) [6] is a comprehensive collection of 3D point clouds of 6 indoor areas with total 272 rooms in various environments such as lobbies, hallways, and offices. Each point in these scans carries a semantic label from one of 13 categories, encompassing 12 semantic classes and an additional 'clutter' category.

ScanNet. ScanNet[7] contains 1,513 scanned point clouds of 707 unique indoor scenes. Following the experimental protocol in [7], our training set comprises 1,201 scenes, with the remaining 312 scenes reserved for testing. Each point in these scans carries a semantic label from one of 21 categories, encompassing 20 semantic classes and an additional unannotated space.

5.1.2 Setup

For efficient processing, we divide the large rooms in the datasets into non-overlapping $1m \times 1m$ blocks along the xy plane, following the pre-processing strategy in [1, 3]. Consequently, there are 31187 blocks for S3DIS and 36350 blocks for ScanNet. Each block comprises a random sample of $N = 2048$ points, where each point is represented as a 9D vector, including XYZ, RGB, and normalized spatial coordinates.

To customize the dataset to the Generalized Few-shot learning framework, we construct a unique 3D-GFS split of base and novel classes for each dataset. We carefully consider the limitations of point cloud datasets, such as the scarcity of samples in certain classes and their distribution, which may lead to insufficient training data. These limitations restrict the available options, as a sufficient number of samples from base classes are required for effective training. Consequently, we consider three novel classes in S3DIS (*window, sofa, board*) and five in ScanNet (*sink, bathtub, toilet, shower curtain, refrigerator*), based on the number of categories available in each dataset. Notably, semantically related pairs like *sofa* (novel) and *chair* (base) in S3DIS, and *refrigerator* (novel) and *cabinet* (base) in ScanNet, are considered.

For the training set, we divide it into two distinct parts: the first includes samples containing only the base classes C_{b}, while the second part comprises samples with the novel classes C_{n}. We utilize the samples in the first part to construct the base set $D_{\mathrm{b}}^{\mathrm{train}}$, which is then used to train the model in the *base prototype generation phase*. During the subsequent novel prototype registration phase, we traverse N_{n} classes out of novel classes, as opposed to making a random selection. For each chosen novel class c, K samples ($K = 5, 10, 30$) that contain only the category c and exclude other novel classes are selected to form the support set $D_{\mathrm{s}}^{\mathrm{train}}$, in line with the conventional 3D-FS setting [5]. In the *final evaluation phase*, we

assess our model to the entire testing set, referred to as the query set, and achieve segmentation results simultaneously on all categories.

5.1.3 Evaluation Metric

In this work, we evaluate our proposed method with *mean Intersection-over-Union (mean-IoU)*, the most widely used metric in point cloud semantic segmentation. In the semantic segmentation task, Intersection-over-union (IoU) per class is the overlapping area between the predicted label and ground truth divided by the joint area between the predicted label and ground truth (intersection of both/union of both). IoU values range from zero to one, where zero means no overlap and one means perfect overlap.

$$IoU_i = \frac{TP_i}{TP_i \ + \ FP_i \ + \ FN_i}, \ (i \in \{1, 2, ..., (N_\mathrm{b} + N_\mathrm{n})\}) \tag{5.1}$$

where TP_i, FP_i, FN_i denote the number of true positive, false positive, and false negative and False Negative predictions for the class i. mIoU is the average of the IoU for all categories.

$$mIoU = \frac{1}{N_\mathrm{b} + N_\mathrm{n}} \sum_{i=0}^{N_\mathrm{b}+N_\mathrm{n}-1} IoU_i \tag{5.2}$$

5.2 Training Details

In our approach, we adopt the DGCNN framework[3] as the feature extractor. During the *base class learning phase*, we configure the training with a batch size of 16 and an epoch number of 150. We use Adam optimizer with a learning rate of 0.001 and weight decay of 0.0001. In the *final evaluation phase*, we evaluate the model on five distinct sets generated from different seeds, with the ultimate result being the average of these five evaluations. Our proposed method is developed using the PyTorch framework and is trained on a single NVIDIA Tesla P100 GPU.

5.3 Results and Analyses

5.3.1 Comparison with Base Model

Baseline. We define two baselines to scale our results.

1) *Full Supervision.* A fully-supervised DGCNN model, referred to as *Full Supervision*, is trained using both the base classes with adequate samples $D_{\mathrm{b}}^{\mathrm{train}}$ and the novel classes with few samples $D_{\mathrm{s}}^{\mathrm{train}}$ simultaneously, and then evaluated on D^{test}.

2) *Base Model.* To verify the effectiveness of our proposed modules and training strategy, we compare it with the Base Model, which is the prototypical network integrating a DGCNN feature extractor and a prototype classifier (as detailed in Sec 4.2). The classifiers for the base and novel classes are obtained through back-propagation and masked average pooling during *base class learning phase* and *novel class registration phase* respectively, and then concatenated for the final classification.

Result. Our approach tests the performance on $K = 5, 10, 30$ shots, with results presented in Tables 5.1 and 5.2 for the S3DIS and ScanNet datasets.

Table 5.1 and Table 5.2 indicate that the Full Supervision method behaves comparably to our proposed method for base classes but is significantly less effective for novel classes. Notably, the performance of the Full Supervision method improves with an increased number of shots for novel classes. Our method outperforms the supervised models in *5-shot* and *10-shot* settings but is less effective in the *30-shot* scenario. This suggests that when trained simultaneously with abundant base class samples and inadequate novel class samples, the novel class classifiers learned under the Full Supervision method are of inferior quality, with their performance for novel classes approaching zero when the samples are insufficient. Therefore, when the number of samples of different categories is very imbalanced, directly using fully supervised methods cannot segment effectively the categories with small samples.

Compared with the Base Model, our method demonstrates notable improvements in the Generalized Few-Shot settings of 5-shot, 10-shot, and 30-shot, achieving approximately 3.66%, 3.82%, and 3.73% enhancement on the S3DIS dataset, and 4.72%, 4.67%, and 3.9% on the ScanNet dataset, respectively. These gains show

TABLE 5.1: 3D-GFS comparison results (in %) on S3DIS [6]. B represents the mIoU results across base classes; N details the mIoU results for novel classes; and *All* encompasses the mIoU results for the combined set of both base and novel classes.

Method	Tr	Te	5-shot			10-shot			30-shot		
	Class	Class	*B*	*N*	*All*	*B*	*N*	*All*	*B*	*N*	*All*
Full Supervision	$C_b \cup C_n$	$C_b \cup C_n$	76.07	2.95	59.19	75.94	9.29	60.56	75.62	29.03	64.87
Base Model	C_b	$C_b \cup C_n$	75.54	10.33	60.49	75.76	9.96	60.57	75.97	10.18	60.78
Ours	C_b	$C_b \cup C_n$	**79.33**	**13.56**	**64.15**	**79.52**	**13.98**	**64.39**	**79.58**	**14.29**	**64.51**

that our proposed Adaptive Support Enrichment (ASE) and Query Aware Representation (QAR) modules, alongside our innovative training strategy, can capture critical contextual information between base and novel classes, and refine prototype representations. Notably, the superior performance on the ScanNet dataset suggests the enhanced adaptability of our model to scenarios with a larger number of classes.

Our method shows significant improvement in scenarios with limited samples. Actually, in the context of the boom of large models, a more realistic situation is that we need to fine-tune some classes with only a small number of samples, rather than training from scratch with a large amount of base category data. Therefore, this aspect of our approach is particularly relevant in the era of large-scale models, offering practical advantages in terms of both results and real-world applicability, compared to fully-supervised models.

Quantitative Results. Figure 5.1 and Figure 5.2 provide qualitative results of our proposed method on the two datasets.

5.3.2 Ablation Study

Design options of ASE. The impact of the ASE module is analyzed in Table 5.3 and Table 5.4. ASE module exploits essential co-relationships between base classes and novel classes in support samples. The ξ_{sup} function used to produce λ^i_{sup} plays

TABLE 5.2: 3D-GFS comparison results (in %) on ScanNet [7]. *B* represents the mIoU results across base classes; *N* details the mIoU results for novel classes; and *All* encompasses the mIoU results for the combined set of both base and novel classes.

Method	Tr	Te	5-shot			10-shot			30-shot		
	Class	Class	*B*	*N*	*All*	*B*	*N*	*All*	*B*	*N*	*All*
Full Supervision	$C_b \cup C_n$	$C_b \cup C_n$	40.76	0.00	30.57	39.60	3.22	30.50	39.48	13.82	33.07
Base Model	C_b	$C_b \cup C_n$	37.14	2.43	28.46	37.02	2.49	28.39	37.82	2.57	29.00
Ours	C_b	$C_b \cup C_n$	**42.89**	**4.04**	**33.18**	**42.81**	**3.81**	**33.06**	**42.71**	**3.52**	**32.91**

an important role. We explore two variations of the ξ_{sup} function, two-layer MLPs and cosine similarity. As shown in Table 5.3 and Table 5.4, our 'MLP' method outperforms the 'Cos' method in the 5-shot, 10-shot, and 30-shot GFS settings by approximately 0.67%, 0.37%, and 0.49% on S3DIS, and 1.27%, 1.43%, and 1.9% on ScanNet, respectively. This suggests that ASE is more conducive to the Base Model when implemented with two-layer MLPs. The possible reason behind this might be that $p^i_{\text{b,sup}}$, derived from ground-truth masks, provides more accurate categorical representations with reduced noise. Consequently, the 'MLP' method enables $p^i_{\text{b,cls}}$ to adaptively compute ratios and integrate contextual information from $p^i_{\text{b,sup}}$.

Design options of QAR. The effectiveness of the QAR module is assessed in Table 5.3 and Table 5.4. The QAR module is designed to dynamically adjust the classifier representation by incorporating prior information obtained from query samples. Central to its efficacy is the ξ_{qry} function, responsible for generating λ^i_{qry}. We investigate two variations of the ξ_{qry} function, two-layer MLPs and cosine similarity. According to the results shown in Table 5.3 and Table 5.4, our 'MLP' method demonstrates superior performance over the 'Cos' method in the 5-shot, 10-shot, and 30-shot GFS settings by approximately 0.15%, 0.10%, and 0.28% on S3DIS, and 0.33%, 0.27%, and 0.26% on ScanNet, respectively. This indicates that the two-layer MLPs configuration of QAR significantly enhances the effectiveness of the Base Model.

TABLE 5.3: Ablation study of ASE and QAR modules (in %) on S3DIS [6]. 'MLP' and 'Cos' refer to two-layer MLPs and cosine similarity to generate weighing factors λ_{sup} and λ_{qry}.

Method	Cos	MLP	5-shot			10-shot			30-shot		
			B	***N***	***All***	***B***	***N***	***All***	***B***	***N***	***All***
Base Model	-	-	75.54	10.33	60.49	75.76	9.96	60.57	75.97	10.18	60.78
ASE	✓	-	75.93	14.00	61.64	76.35	14.55	62.09	76.40	14.41	62.10
ASE	-	✓	76.59	14.91	62.36	76.85	15.84	62.78	77.10	15.98	62.99
QAR	✓	-	78.79	13.37	63.70	78.82	14.00	63.86	78.80	13.82	63.81
QAR	-	✓	79.11	13.00	63.85	79.13	13.39	63.96	79.15	13.90	64.09
Ours	-	-	**79.33**	**13.56**	**64.15**	**79.52**	**13.98**	**64.39**	**79.58**	**14.29**	**64.51**

Training strategy. The effectiveness of our proposed training strategy is analyzed in Table 5.5 and Table 5.6. In this approach, 'Pseudo Support' samples and 'Pseudo Query' samples are selected to imitate the behaviors of actual novel and base classes in the support samples. The 'Base Model +' is a baseline of the training strategy, in which only the prototypes of 'Pseudo Novel' are formed. 'Ours - Tr' indicates that ASE and QAR are exclusively implemented during the *base prototype generation phase*, maintaining 'Base Model+' for the subsequent *novel prototype registration phase* and *final evaluation phase*. 'Ours - Te' is the complete opposite. Note that in 'Ours - Te', the variable λ^{i}_{sup}, derived from two-layer trainable MLPs, is set as the average ratio for ASE and QAR. As shown in Table 5.5 and Table 5.6, the findings provided strong evidence for the remarkable effectiveness of the training strategy.

5.4 Comparison with 3D-FS Models in 3D-GFS

In order to demonstrate the limitations of 3D-FS models when it comes to both base and novel classes, we assess the performance of the 3D-FS frameworks within

TABLE 5.4: Ablation study of ASE and QAR modules (in %) on ScanNet [7]. 'MLP' and 'Cos' refer to two-layer MLPs and cosine similarity to generate weighing factors λ_{sup} and λ_{qry}.

Method	Cos	Mlp	5-shot			10-shot			30-shot		
			B	***N***	***All***	***B***	***N***	***All***	***B***	***N***	***All***
Base Model	-	-	37.14	2.43	28.46	37.02	2.49	28.39	37.82	2.57	29.00
ASE	✓	-	38.15	4.00	29.61	38.16	4.27	29.68	38.41	3.94	29.79
ASE	-	✓	38.61	4.32	30.04	38.28	4.82	29.92	39.22	4.11	30.45
QAR	✓	-	41.30	3.99	31.98	41.15	4.10	31.88	40.95	3.78	31.66
QAR	-	✓	41.78	3.92	32.31	41.66	3.61	32.15	41.49	3.21	31.92
Ours	-	-	**42.89**	**4.04**	**33.18**	**42.81**	**3.81**	**33.06**	**42.71**	**3.52**	**32.91**

the setting of 3D-GFS.

Baseline. Due to constraints in computing resources, we compare the two 3D-FS models under 5-way and 10-way configurations.

1) *ProtoNet* [5] is the baseline method of 3D-FS task. Zhao *et al.* [5] was the first to establish a baseline for this segmentation task. We follow the original implementation and use the episodic strategy. During the training processing, base prototypes are formed by MAP over the base classes of $D_{\mathrm{b}}^{\mathrm{train}}$. We modify the evaluation code to generate novel class prototypes from $D_{\mathrm{s}}^{\mathrm{train}}$ and feed D^{test} of all classes into the model.

2) *PAP-FZS3D* [19] is the state-of-the-art 3D-FS method. *PAP-FZS3D* identified a significant image-gap in 3D objects due to the lack of a robust 3D pre-trained model, so introduced the Query-Guided Prototype Adaption (QGPA) and a self-reconstruction module to improve prototype representation for each class. The implementation of PAP-FZS3D in the 3D-GFS setting is similar to 1).

We employed the publicly available codes and adhered to the default training configurations for this comparison. In our testing approach, all prototypes – both

TABLE 5.5: Ablation study of training strategy (in %) on S3DIS [6].

Method	5-shot			10-shot			30-shot		
	B	***N***	***All***	***B***	***N***	***All***	***B***	***N***	***All***
Base Model	75.54	10.33	60.49	75.76	9.96	60.57	75.97	10.18	60.78
Base Model+	76.41	14.14	62.04	76.83	14.54	62.45	76.76	15.78	62.69
Ours - Tr	67.12	9.67	53.87	67.94	9.35	54.42	67.61	10.51	54.43
Ours - Te	75.98	13.70	61.61	75.01	13.98	60.93	74.99	14.12	60.94
Ours	**79.33**	**13.56**	**64.15**	**79.52**	**13.98**	**64.39**	**79.58**	**14.29**	**64.51**

TABLE 5.6: Ablation study of training strategy (in %) on ScanNet [7].

Method	5-shot			10-shot			30-shot		
	B	***N***	***All***	***B***	***N***	***All***	***B***	***N***	***All***
Base Model	37.14	2.43	28.46	37.02	2.49	28.39	37.82	2.57	29.00
Base Model+	38.99	4.04	30.26	38.84	4.28	29.75	38.86	3.78	30.09
Ours - Tr	31.16	2.39	23.97	30.90	2.55	23.81	30.76	2.30	23.65
Ours - Te	37.13	3.24	28.65	37.64	3.57	29.12	37.61	3.21	29.01
Ours	**42.89**	**4.04**	**33.18**	**42.81**	**3.81**	**33.06**	**42.71**	**3.52**	**32.91**

base and novel – were made accessible to the test set. For the creation of novel prototypes, support samples $D_{\text{s}}^{\text{train}}$ were used, whereas the base prototypes were generated through the average pooling of base class-related samples $D_{\text{b}}^{\text{train}}$ derived from the training set.

Results. The results of 3D-GFS on S3DIS and ScanNet datasets are shown in Table 5.7 and 5.8. Results in the bracket are from approaches of [5, 19] without

TABLE 5.7: Comparative results of 3D-FS models in 3D-GFS (in %) on S3DIS [6]

Method	5-shot			10-shot		
	B	***N***	***All***	***B***	***N***	***All***
ProtoNet [5]	56.50	11.49(59.09)	46.12	56.56	12.08(62.80)	46.29
PAP-FZS3D [19]	48.62	0.12(71.37)	37.43	52.17	0.42(75.05)	40.22
Ours	**79.33**	**13.56**	**64.15**	**79.52**	**13.98**	**64.39**

TABLE 5.8: Comparative results of 3D-FS models in 3D-GFS (in %) on ScanNet [7]

Method	5-shot			10-shot		
	B	***N***	***All***	***B***	***N***	***All***
ProtoNet [5]	30.25	2.37(46.53)	23.28	30.99	2.50(48.42)	23.86
PAP-FZS3D [19]	24.16	0.05(64.15)	18.13	24.27	0.07(69.60)	18.22
Ours	**42.89**	**4.04**	**33.18**	**42.81**	**3.81**	**33.06**

any modification, which can only handle the novel classes. Table 5.7 and 5.8 show that ProtoNet and PAP-FZS3D fall short under the 3D-GFS setting because 3D-FS models are trained to differentiate only between appearing novel target classes and backgrounds. All the base classes are treated as background and the decision boundary only lies between N specific target novel classes and the background in an episode. The model becomes less effective when facing the task of segmenting all the classes simultaneously because the number of classes that need to be distinguished at this time has greatly increased during training. Thus, even though they work well on the novel classes alone, they perform poorly when directly extending them to all classes. When required to divide all classes, the original segmentation decision boundary between novel classes breaks down. Our proposed method significantly achieves better performance than the state-of-the-art 3D-FS method consistently on the both S3DIS and ScanNet datasets.

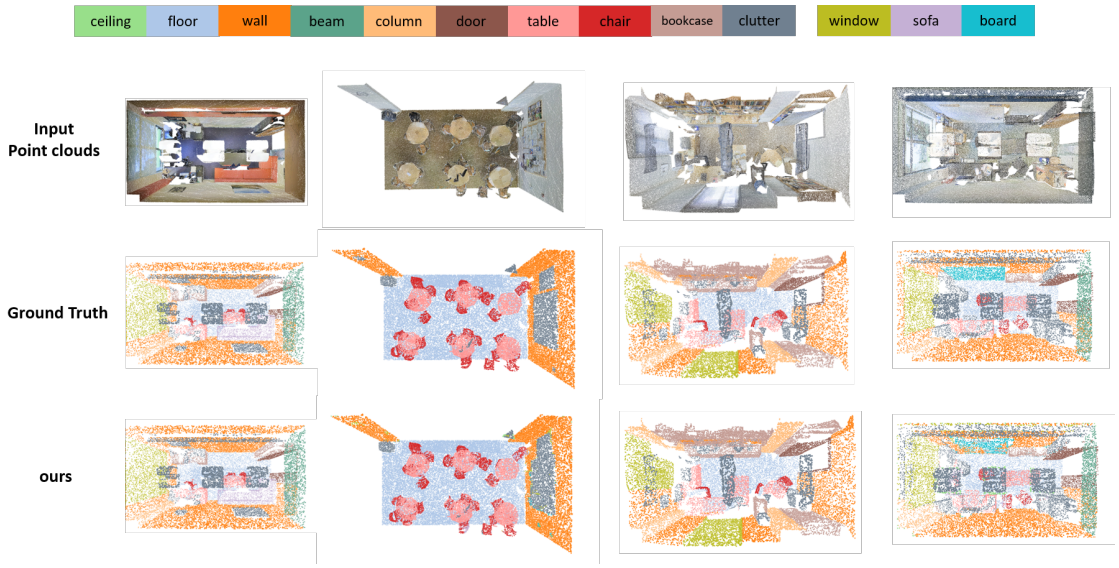

FIGURE 5.1: The qualitative results of our proposed method in *5-shot* setting on S3DIS [6]. The novel classes are *window*, *sofa* and *board*.

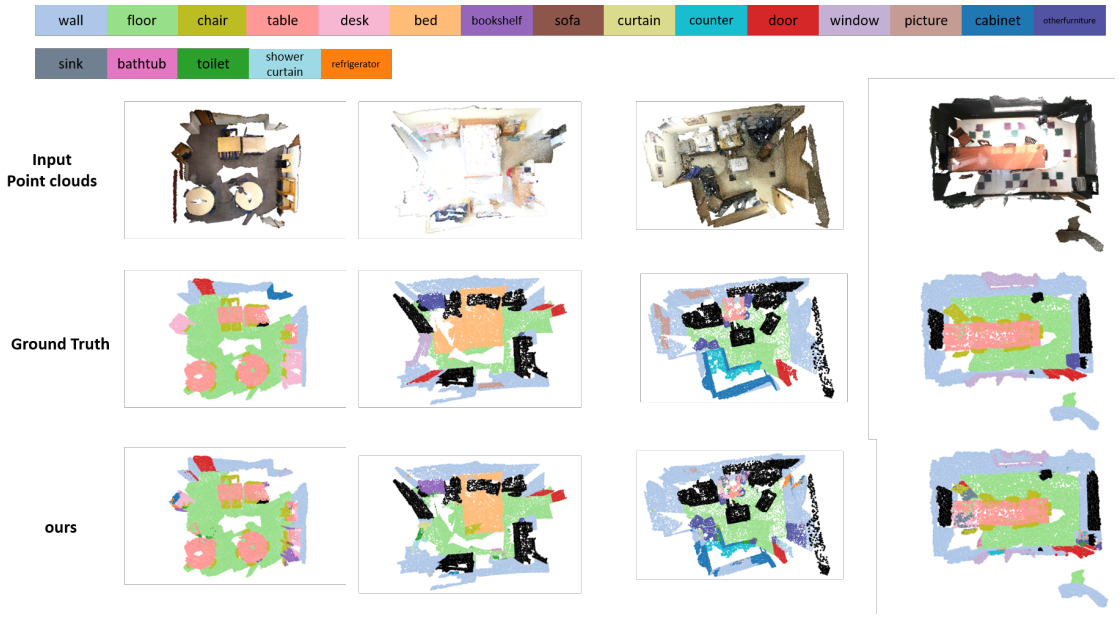

FIGURE 5.2: The qualitative results of our proposed method in *5-shot* setting on ScanNet [7]. The novel classes are *sink*, *bathtub*, *toilet*, *shower curtain* and *refrigerator*.

Chapter 6

Conclusion and Future Work

6.1 Conclusion

In conclusion, we focus on a novel task Generalized Few-Shot 3D Point Cloud Semantic Segmentation (3D-GFS). Unlike traditional Few-Shot 3D Point Cloud Semantic Segmentation (3D-FS), 3D-GFS addresses the challenges of segmenting both the base classes with adequate samples and the novel classes with few samples simultaneously. Specifically, Our 3D-GFS method consists of three essential phases: base prototype generation phase, novel prototype registration phase, and final evaluation phase.

Inspired by the classic Few-Shot learning framework, we design a prototypical Base Model for 3D-GFS, showing promising performance. To adopt the contextual information, we design the Adaptive Support Enrichment Module (ASE) module, effectively leveraging the co-relationship between support and base samples. Then, we further design the Query Aware Representation (QAR) module to utilize prior information derived from query samples. Considering the better representation of both base and novel prototypes, we propose a new training strategy to make the embedding space consistent.

Our method achieves state-of-the-art results on two benchmarks, S3DIS and ScanNet datasets. These findings pave the way for more efficient and accurate few-shot

segmentation techniques in real-world scenarios and have great potential for applications in various domains, such as augmented reality, robotics, and autonomous vehicles.

6.2 Future Work

Several potential techniques are in consideration. Primarily, achieving accurate and generalized prototype representations for all samples within a category is crucial for segmentation tasks. Therefore, a potential future direction is to obtain better prototype representation for each category.

Besides, our current approach only uses the contextual information within the samples. However, it is valuable and helpful to explore other relationships between base and novel classes, such as structural similarity or semantic relevance. This helps to improve the segmentation results of novel classes by leveraging the information from base classes.

We find common ground with other tasks such as incremental few-shot learning and few-shot detection, which exhibit resemblances to our own objectives. By probing into the methods in these domains, we might have valuable insights that can potentially enhance the performance of novel classes in our proposed Generalized Few-Shot 3D Point Cloud Semantic Segmentation task.

List of Author's Awards, Patents, and Publications

Conference Proceedings

- **Shuqian Yang**, Henghui Ding and Xudong Jiang, "Generalized Few-Shot Point Cloud Segmentation", in *IEEE International Symposium on Circuits and Systems (ISCAS), 2024* (accepted as oral presentation).

Bibliography

[1] 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/CVF Conference on Computer Vision and Pattern Recognition*, 2017. xiii, 1, 2, 8, 9, 30

[2] Charles R. Qi, Li Yi, Hao Su, and Leonidas J Guibas. PointNet++: Deep hierarchical feature learning on point sets in a metric space. In *Advances in Neural Information Processing Systems*, 2017. xiii, 2, 8, 9

[3] 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*, 38(5), 2019. xiii, 2, 10, 11, 30, 31

[4] 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*, 2019. xiii, 12, 13, 17

[5] 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, 2021. xiii, 2, 3, 13, 14, 15, 17, 20, 22, 29, 30, 36, 37, 38

[6] 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/CVF Conference on Computer Vision and Pattern Recognition*, 2016. xiv, xv, 3, 4, 29, 33, 35, 37, 38, 39

[7] 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/CVF Conference on Computer Vision and Pattern Recognition*, 2017. xiv, xv, 3, 4, 30, 34, 36, 37, 38, 39

[8] Xinge Zhu, Hui Zhou, Tai Wang, Fangzhou Hong, Yuexin Ma, Wei Li, Hongsheng Li, and Dahua Lin. Cylindrical and asymmetrical 3d convolution networks for lidar segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021. 1

[9] Oleg Yu Sergiyenko and Vera V. Tyrsa. 3d optical machine vision sensors with intelligent data management for robotic swarm navigation improvement. *IEEE Sensors Journal*, 21(10):11262–11274, 2021. 1

[10] Christian Hürzeler Gerhard Schrotter. The digital twin of the city of zurich for urban planning. *Journal of Photogrammetry, Remote Sensing and Geoinformation Science*, pages 99–120, 2020. 1

[11] Philipp A. Rauschnabel, Barry J. Babin, M. Claudia tom Dieck, Nina Krey, and Timothy Jung. What is augmented reality marketing? its definition, complexity, and future. *Journal of Business Research*, 142:1140–1150, 2022. 1

[12] Yangyan Li, Rui Bu, Mingchao Sun, Wei Wu, Xinhan Di, and Baoquan Chen. PointCNN: Convolution on x-transformed points. In *Advances in Neural Information Processing Systems*, 2018. 2, 11

[13] Yongcheng Liu, Bin Fan, Shiming Xiang, and Chunhong Pan. Relation-shape convolutional neural network for point cloud analysis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019. 10

[14] Wenxuan Wu, Zhongang Qi, and Li Fuxin. PointConv: Deep convolutional networks on 3D point clouds. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019. 11

[15] Hengshuang Zhao, Li Jiang, Jiaya Jia, Philip H.S. Torr, and Vladlen Koltun. Point Transformer. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021. 2

[16] Saining Xie, Jiatao Gu, Demi Guo, Charles R. Qi, Leonidas Guibas, and Or Litany. PointContrast: Unsupervised pre-training for 3D point cloud understanding. In *Proceedings of the European Conference on Computer Vision Workshops*, 2020. 2

[17] Xun Xu and Gim Hee Lee. Weakly supervised semantic point cloud segmentation: Towards 10x fewer labels. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020. 2

[18] Jilin Mei, Biao Gao, Donghao Xu, Wen Yao, Xijun Zhao, and Huijing Zhao. Semantic segmentation of 3D lidar data in dynamic scene using semi-supervised learning. *IEEE Transactions on Intelligent Transportation Systems*, 21(6):2496–2509, 2020. 2

[19] Shuting He, Xudong Jiang, Wei Jiang, and Henghui Ding. Prototype adaption and projection for few-and zero-shot 3d point cloud semantic segmentation. *IEEE Transactions on Image Processing*, 32:3199–3211, 2023. 2, 14, 36, 37, 38

[20] Oriol Vinyals, Charles Blundell, Timothy Lillicrap, koray kavukcuoglu, and Daan Wierstra. Matching networks for one shot learning. In *Advances in Neural Information Processing Systems*, 2016. 3, 11, 12, 15

[21] Anh-Vu Vo, Linh Truong-Hong, Debra F. Laefer, and Michela Bertolotto. Octree-based region growing for point cloud segmentation. *ISPRS Journal of Photogrammetry and Remote Sensing*, 104:88–100, 2015. 8

[22] Wei Zeng and Theo Gevers. 3DContextNet: K-d tree guided hierarchical learning of point clouds using local and global contextual cues. In *Proceedings of the European Conference on Computer Vision Workshops*, 2018. 8

[23] Hang Su, Subhransu Maji, Evangelos Kalogerakis, and Erik Learned-Miller. Multi-view convolutional neural networks for 3D shape recognition. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2015. 8

[24] Daniel Maturana and Sebastian Scherer. VoxNet: A 3D convolutional neural network for real-time object recognition. In *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2015. 8

[25] Gernot Riegler, Ali Osman Ulusoy, and Andreas Geiger. OctNet: Learning deep 3D representations at high resolutions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2017. 8

[26] Hong-Wei Lin, Chiew-Lan Tai, and Guo-Jin Wang. A mesh reconstruction algorithm driven by an intrinsic property of a point cloud. *Computer-Aided Design*, 36:1–9, 2004. 8

[27] Oren Dovrat, Itai Lang, and Shai Avidan. Learning to sample. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019. 9

[28] Wenxuan Wu, Zhongang Qi, and Li Fuxin. Pointconv: Deep convolutional networks on 3d point clouds. In *Proceedings of the IEEE/CVF Conference on computer vision and pattern recognition*, 2019.

[29] Jiancheng Yang, Qiang Zhang, Bingbing Ni, Linguo Li, Jinxian Liu, Mengdie Zhou, and Qi Tian. Modeling point clouds with self-attention and gumbel

subset sampling. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019.

[30] Qingyong Hu, Bo Yang, Linhai Xie, Stefano Rosa, Yulan Guo, Zhihua Wang, Niki Trigoni, and Andrew Markham. Randla-net: Efficient semantic segmentation of large-scale point clouds. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020. 9

[31] Martin Simonovsky and Nikos Komodakis. Dynamic edge-conditioned filters in convolutional neural networks on graphs. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2017. 9

[32] Loic Landrieu and Martin Simonovsky. Large-scale point cloud semantic segmentation with superpoint graphs. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2018. 9

[33] Yiru Shen, Chen Feng, Yaoqing Yang, and Dong Tian. Mining point cloud local structures by kernel correlation and graph pooling. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2018. 9

[34] Gusi Te, Wei Hu, Amin Zheng, and Zongming Guo. RGCNN: Regularized graph cnn for point cloud segmentation. In *Proceedings of the 26th ACM international conference on Multimedia*, 2018. 9

[35] Chu Wang, Babak Samari, and Kaleem Siddiqi. Local spectral graph convolution for point set feature learning. In *Proceedings of the European Conference on Computer Vision*, 2018. 10

[36] Guohao Li, Matthias Muller, Ali Thabet, and Bernard Ghanem. Deepgcns: Can gcns go as deep as cnns? In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 9267–9276, 2019. 10

[37] Yifan Xu, Tianqi Fan, Mingye Xu, Long Zeng, and Yu Qiao. Spidercnn: Deep learning on point sets with parameterized convolutional filters. In *Proceedings of the European Conference on Computer Vision*, 2018. 11

[38] Carlos Esteves, Christine Allen-Blanchette, Ameesh Makadia, and Kostas Daniilidis. Learning so (3) equivariant representations with spherical cnns. In *Proceedings of the European Conference on Computer Vision*, 2018. 11

[39] Hugues Thomas, Charles R Qi, Jean-Emmanuel Deschaud, Beatriz Marcotegui, François Goulette, and Leonidas J Guibas. Kpconv: Flexible and deformable convolution for point clouds. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019. 11

[40] Yaqing Wang, Quanming Yao, James T Kwok, and Lionel M Ni. Generalizing from a few examples: A survey on few-shot learning. *ACM computing surveys (csur)*, 53:1–34, 2020. 11

[41] Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *Proceedings of the 34th International Conference on Machine Learning*, Proceedings of Machine Learning Research, 2017. 11

[42] Qi Cai, Yingwei Pan, Ting Yao, Chenggang Yan, and Tao Mei. Memory matching networks for one-shot image recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018.

[43] Andrei A. Rusu, Dushyant Rao, Jakub Sygnowski, Oriol Vinyals, Razvan Pascanu, Simon Osindero, and Raia Hadsell. Meta-learning with latent embedding optimization, 2019.

[44] Jake Snell, Kevin Swersky, and Richard Zemel. Prototypical networks for few-shot learning. *Advances in Neural Information Processing Systems*, 2017. 12, 13, 19

[45] Flood Sung, Yongxin Yang, Li Zhang, Tao Xiang, Philip H.S. Torr, and Timothy M. Hospedales. Learning to Compare: Relation network for few-shot learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018.

[46] Chi Zhang, Yujun Cai, Guosheng Lin, and Chunhua Shen. DeepEMD: Few-shot image classification with differentiable earth mover's distance and structured classifiers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020. 11

[47] Henghui Ding, Xudong Jiang, Bing Shuai, Ai Qun Liu, and Gang Wang. Context contrasted feature and gated multi-scale aggregation for scene segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018. 12

[48] Bing Shuai, Henghui Ding, Ting Liu, Gang Wang, and Xudong Jiang. Toward achieving robust low-level and high-level scene parsing. *IEEE Transactions on Image Processing*, 28(3):1378–1390, 2018. 12

[49] Amirreza Shaban, Shray Bansal, Zhen Liu, Irfan Essa, and Byron Boots. One-shot learning for semantic segmentation. In *British Machine Vision Conference*, 2017. 12

[50] Gregory Koch, Richard Zemel, Ruslan Salakhutdinov, et al. Siamese neural networks for one-shot image recognition. In *ICML deep learning workshop*, volume 2, 2015. 13

[51] Nanqing Dong and Eric P Xing. Few-shot semantic segmentation with prototype learning. In *British Machine Vision Conference*, 2018. 13

[52] Xiaolin Zhang, Yunchao Wei, Yi Yang, and Thomas S Huang. Sg-one: Similarity guidance network for one-shot semantic segmentation. *IEEE transactions on cybernetics*, 50(9):3855–3865, 2020. 13

[53] 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*, 2019. 13

[54] 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*, 2019. 13

[55] Xianghui Yang, Bairun Wang, Kaige Chen, Xinchi Zhou, Shuai Yi, Wanli Ouyang, and Luping Zhou. Brinet: Towards bridging the intra-class and inter-class gaps in one-shot segmentation. *arXiv preprint arXiv:2008.06226*, 2020. 13

[56] Yongfei Liu, Xiangyi Zhang, Songyang Zhang, and Xuming He. Part-aware prototype network for few-shot semantic segmentation. In *Proceedings of the European Conference on Computer Vision*, 2020. 13

[57] Boyu Yang, Chang Liu, Bohao Li, Jianbin Jiao, and Qixiang Ye. Prototype mixture models for few-shot semantic segmentation. In *Proceedings of the European Conference on Computer Vision*, 2020. 13

[58] Zhuotao Tian, Hengshuang Zhao, Michelle Shu, Zhicheng Yang, Ruiyu Li, and Jiaya Jia. Prior guided feature enrichment network for few-shot segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(2):1050–1065, 2020. 13

[59] Hang Qi, Matthew Brown, and David G Lowe. Low-shot learning with imprinted weights. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018. 19

[60] Boris Oreshkin, Pau Rodríguez López, and Alexandre Lacoste. Tadam: Task dependent adaptive metric for improved few-shot learning. *Advances in Neural Information Processing Systems*, 2018. 20, 22

[61] Liang-Chieh Chen, Yukun Zhu, George Papandreou, Florian Schroff, and Hartwig Adam. Encoder-decoder with atrous separable convolution for semantic image segmentation. In *Proceedings of the European Conference on Computer Vision*, 2018. 24

[62] Hengshuang Zhao, Jianping Shi, Xiaojuan Qi, Xiaogang Wang, and Jiaya Jia. Pyramid scene parsing network. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2017. 24