Few-shot 3D Point Cloud Semantic Segmentation – Supplementary Material

This appendix contains the splitting details of the S3DIS and ScanNet datasets (Section A), more framework details including the architecture of EdgeConv (Section B.1) and the settings of three hyper-parameters (Section B.2).

A. Dataset Split

Table 1 lists the class names in each split of the S3DIS and ScanNet datasets.

	split=0	split=1
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, show curtain, sink, sofa, table, toilet, wall, window

Table 1: Test class names for each split of S3DIS and ScanNet.

B. More Framework Details

B.1. EdgeConv architecture details

Figure 1 illustrates the architecture and configuration of EdgeConv, which is a basic block of the feature extractor. To perform graph CNN, a k-NN graph is dynamically computed from the input point-wise features to EdgeConv. Note that this k-NN graph is different from the k-NN graph in Section 3.2.3. We set $k = 20$ in our experiments. Each point $\mathbf{x}_i$ in the point cloud is concatenated with its translated neighbor point $(\mathbf{x}_j - \mathbf{x}_i)$, which is yielded by translating $\mathbf{x}_j$ to the local system with $\mathbf{x}_i$ as the center. Consequently, a $N \times k \times 2f_{in}$ feature tensor is produced from the input tensor $N \times f_{in}$ and further passed to two MLP layers. Finally, EdgeConv aggregates the resultant feature tensor over the k neighboring features using a max-pooling operator to generate the output point-wise features.

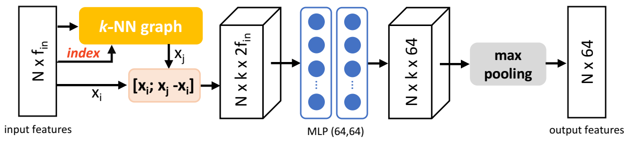

Figure 1. The architecture of EdgeConv component in the embedding network.

B.2. Hyper-parameter settings

As mentioned in Section 4.4, we empirically find that the optimal value of σ varies in different datasets. Additionally, we also observe varying optimal number of prototypes per class n under different few-shot settings. Table 2 shows the optimal value of n in different few-shot settings. It can be seen that n becomes larger when the number of shots increases. This is reasonable since more shots result in larger number of observed support points for each class, which requires larger n to model the larger variety. From Table 2, we also observe that n becomes larger when the number of "ways" increases. This is probably due to the more difficult 3-way segmentation requires fine-grained multi-prototypes for each class. We set $k = 200$ for the k-NN graph mentioned in Section 3.2.3 on all few-shot settings in both datasets.

	2-way 1-shot	2-way 5-shot	3-way 1-shot	3-way 5-shot
$n =$	100	150	150	300

Table 2: The value of n in different few-shot settings.