QUERY-AWARE HUB PROTOTYPE LEARNING FOR FEW-SHOT 3D POINT CLOUD SEGMENTATION

Anonymous authors

000

001

002 003 004

010 011

012

013

014

016

018

021

025

026

027

028029030

031

033

034

037

040

041

042

043

044

046

047

048

051

052

Paper under double-blind review

ABSTRACT

Few-shot 3D point cloud semantic segmentation (FS-3DSeg) aims to segment novel classes with only a few labeled samples. However, existing metric-based prototype learning methods generate prototypes solely from the support set, without considering their relevance to query data. This often results in prototype bias, where prototypes overfit support-specific characteristics and fail to generalize to the query distribution, especially in the presence of distribution shifts, which leads to degraded segmentation performance. To address this issue, we propose a novel Query-aware Hub Prototype (QHP) learning method that explicitly models semantic correlations between support and query sets. Specifically, we propose a Hub Prototype Generation (HPG) module that constructs a bipartite graph connecting query and support points, identifies frequently linked support hubs, and generates query-relevant prototypes that better capture cross-set semantics. To further mitigate the influence of bad hubs and ambiguous prototypes near class boundaries, we introduce a Prototype Distribution Optimization (PDO) module, which employs a purity-reweighted contrastive loss to refine prototype representations by pulling bad hubs and outlier prototypes closer to their corresponding class centers. Extensive experiments on S3DIS and ScanNet demonstrate that QHP achieves substantial performance gains over state-of-the-art methods, effectively narrowing the semantic gap between prototypes and query sets in FS-3DSeg.

1 Introduction

Point cloud semantic segmentation assigns semantic labels to each point in a 3D point cloud and is essential for applications like autonomous driving and robotics. Although fully supervised methods (Qi et al., 2017b; Lin et al., 2020; Qian et al., 2022) have made significant progress, they rely heavily on costly manual annotations and struggle to generalize to novel classes. To address these challenges, few-shot 3D point cloud segmentation (FS-3DSeg) has gained increasing attention, aiming to learn generalizable models from abundant base class data and adapt the model to novel classes with only a few labeled point clouds.

Recent FS-3DSeg methods typically adopt metric-based prototype learning frameworks, where prototypes are derived from a few labeled support point clouds, and the unlabeled query set is segmented by measuring similarity between query points and these prototypes. As illustrated in Figure 1 (a)(b), these methods can be broadly categorized into two groups: single-prototype approaches (Mao et al., 2022; He et al., 2023; Liu et al., 2024), which generate global class prototypes via masked average pooling, and multi-prototype methods (Zhao et al., 2021b; An et al., 2024), which enhance prototype diversity through strategies like Farthest Point Sampling (FPS) and local clustering. Ideally, prototypes should serve as semantic bridges between support and query sets, requiring strong alignment with query semantics. However, existing methods generate prototypes solely from the support set, emphasizing internal representativeness or diversity while ignoring semantic relevance to the query. This often causes prototype bias, especially under distribution shifts between support and query sets. For example, intra-class variations (e.g., square vs. round tables) may share limited similarity, causing prototypes to overfit support-specific traits and poorly represent diverse queries. Moreover, uniform sampling strategies like FPS often introduce redundant or query-irrelevant prototypes, further compromising segmentation accuracy. To address these challenges, it is essential to develop a query-aware prototype generation mechanism to narrow the semantic gap between prototypes and the query set and improve segmentation performance.

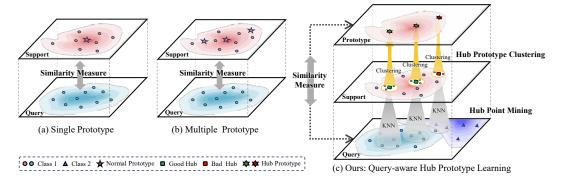


Figure 1: Few-shot 3D point cloud semantic segmentation approaches. (a)(b) Previous prototype learning methods generate prototypes solely based on support points. (c) We propose a Query-aware Hub Prototype Learning method that generates prototypes more closely related to query points.

Realizing the above issues, we propose a Query-aware Hub Prototype (QHP) learning method, as depicted in Figure 1 (c). Hubs (Radovanović et al., 2009) refer to data points that frequently appear among the nearest neighbors of many other points. Therefore, hubs naturally reflect support-query semantic correlation and are well-suited as prototypes. In recent few-shot image classification studies, some methods (Fei et al., 2021; Trosten et al., 2023; Tang et al., 2025) regard hubs as a nightmare and seek to avoid them, worrying that when a support point is a hub, many query points may be retrieved regardless of their true classes. However, we argue that: (1) Not all hubs are harmful. Hubs that emerge within the same class (i.e., good hubs) can capture accurate support-query relationships and serve as effective query-aware prototypes, which helps mitigate prototype bias. (2) The harmful impact of bad hubs is limited in FS-3DSeg. Since each support sample contains numerous points and provides richer point-level supervision, each query point's segmentation can be determined by multiple support-query matches, reducing the risk of being misled by any single bad hub. Therefore, instead of suppressing hubs, we leverage them to bridge the semantic gap between support and query, and propose to learn query-aware hub prototypes. Notably, to further mitigate the influence of bad hubs, we optimize their distributions by pulling those near class boundaries closer to corresponding class centers.

The proposed QHP approach introduces two key components: the Hub Prototype Generation (HPG) module and the Prototype Distribution Optimization (PDO) module. Specifically, HPG explicitly models semantic correlations between support and query sets to learn query-relevant hub prototypes. It constructs a bipartite graph connecting query and support points, identifies support hubs with high linking frequency, and performs local clustering around each hub to generate query-relevant prototypes that better capture cross-set semantics. Query segmentation can be conducted by measuring similarities between query points and these hub prototypes. To further mitigate the influence of bad hubs and ambiguous prototypes near class boundaries, we propose a PDO module during training. PDO constructs a global association graph to identify bad hubs, and adopts a purity-reweighted contrastive loss to pull bad hubs and outlier prototypes toward their corresponding class centers. By jointly leveraging the HPG and PDO modules, our QHP facilitates more query-relevant and discriminative prototype learning, effectively narrowing the semantic gap between prototypes and query sets and yielding improved performance in the FS-3DSeg task.

Our main contributions can be summarized as follows:

- We propose a novel Query-aware Hub Prototype (QHP) Learning method for FS-3DSeg, which explicitly models semantic correlations between support and query sets to generate query-relevant prototypes, addressing prototype bias and narrowing the semantic gap.
- We propose a Hub Prototype Generation (HPG) module to identify support hubs and generate query-relevant hub prototypes that better capture cross-set semantics.
- We design a Prototype Distribution Optimization (PDO) module, optimizing the distributions of bad hubs and outlier prototypes via a purity-reweighted contrastive loss.
- Extensive experiments on S3DIS and ScanNet demonstrate that QHP achieves state-of-theart performance.

2 RELATED WORK

2.1 3D POINT CLOUD SEMANTIC SEGMENTATION

Recent 3D point cloud segmentation methods can be broadly categorized into MLP-based (e.g., PointNet (Qi et al., 2017a) and RandLA-Net (Hu et al., 2020)), convolution-based (e.g., PointCNN (Li et al., 2018), KPConv (Thomas et al., 2019), and RandLA-Net (Hu et al., 2020)), and Transformer-based approaches such as Point Transformer (Zhao et al., 2021a) and Point Transformer V2 (Wu et al., 2022). Although these methods demonstrate strong performance through local feature aggregation or self-attention mechanisms, they typically require expensive, large-scale annotations and struggle to generalize to novel classes unseen during training.

2.2 FEW-SHOT 3D POINT CLOUD SEGMENTATION

Recent FS-3DSeg methods primarily adopt the prototype-based paradigm built upon metric learning. These methods can be broadly categorized into single-prototype and multi-prototype approaches. Single-prototype methods summarize each class using a single representative prototype from the support set. For instance, ProtoNet defines the prototype as the class-wise mean of support features. To mitigate distribution shifts between support and query sets, 2CBR (Zhu et al., 2023) explicitly models such biases, and DPA (Liu et al., 2024) employs query-guided attention to generate task-adaptive prototypes. Seg-NN/PN (Zhu et al., 2024) designs a lightweight module to optimize support-query interaction for prototype generation. However, these methods lack prototype diversity and are unsuitable for handling complex data. To capture intra-class variations, multi-prototype approaches generate multiple prototypes per class. AttMPTI (Zhao et al., 2021b) employs farthest point sampling (FPS) to extract diverse local prototypes. Stratified Transformer (Lai et al., 2022) combines hierarchical sampling with cross-window self-attention. COSeg (An et al., 2024) maintains a momentum-updated pool of base class prototypes. Despite these advances, most methods rely solely on support data to generate prototypes, yielding prototypes biased toward support distribution and poorly aligned with queries, thus limiting generalization to novel classes.

2.3 THE HUBNESS PHENOMENON AND HUBS

Hubness (Radovanovic et al., 2010; Radovanović et al., 2009) describes the tendency of certain points, called *hubs*, to appear frequently in nearest-neighbor lists. It has been studied in areas like multi-view clustering (Xu et al., 2025) and cross-modal retrieval (Bogolin et al., 2022; Wang et al., 2023). In few-/zero-shot classification tasks, prior works (Dinu & Baroni, 2015; Xiao et al., 2021; Cheraghian et al., 2019; Trosten et al., 2023) mostly view hubs as harmful, as query points may be misclassified when dominated by support hubs from different classes. In contrast, we argue that good hubs are beneficial and are primary in our scenario. We thus exploit hubs via query-aware hub prototype learning and mitigate bad hub distance optimization to narrow query-support gaps.

3 METHOD

3.1 PROBLEM FORMULATION AND OVERVIEW

Problem Formulation. FS-3DSeg aims to predict per-point semantic labels for query point clouds using a few labeled support samples. Episodic learning (Zhao et al., 2021b) is typically employed to simulate the few-shot learning process, where each C-way K-shot episode includes a support set $S = \{(I_s^{c,k}, M_s^{c,k})_{k=1}^K\}_{c=1}^C$ and a query set $Q = \{(I_q^l, M_q^l)\}_{l=1}^L$. Each point cloud $I_s^{c,k}, I_q^l \in \mathbb{R}^{T \times (3+f_0)}$ contains T points, each represented by 3D coordinates and auxiliary features (e.g., RGB). $M_s^{c,k} \in \{0,1\}^T$ denotes the binary ground truth (GT) mask indicating whether each point in $I_s^{c,k}$ belongs to class c, while M_q^l denotes the GT labels for the query point cloud I_q^l . During inference, the goal is to predict the query labels \hat{M}_q for the query points in I_q under the guidance of the support set S.

Overview. Figure 2 illustrates the architecture of the proposed QHP framework, comprising two key components: a Hub Prototype Generation (HPG) module and a Prototype Distribution Optimization (PDO) module. We first use a shared backbone Φ to extract point-wise features from the support

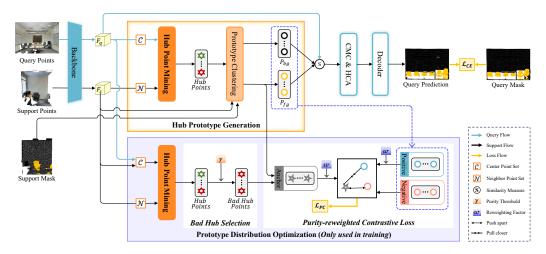


Figure 2: The framework of our Query-aware Hub Prototype Learning method. Initially, we design an HPG module to select support hubs and generate query-relevant hub prototypes. Moreover, during training, a PDO module is integrated to optimize the distribution of bad hubs and outlier prototypes. For clarity, we present the model under the 1-way 1-shot setting.

and query point clouds: $F_s = \Phi\left(I_s\right) \in \mathbb{R}^{C \times K \times T \times D}$ and $F_q = \Phi\left(I_q\right) \in \mathbb{R}^{L \times T \times D}$, where D is the channel dimension. In the HPG module, a Hub Point Mining (HPM) module identifies hub points from S, which are used to generate hub prototypes P via local clustering. These prototypes are matched with query features F_q through similarity measures, and further refined via the CMC and HCA modules (An et al., 2024) to yield query predictions \hat{M}_q . To mitigate the influence of bad hubs and ambiguous prototypes, our PDO module identifies bad hubs by thresholding their purity and applies a Purity-reweighted Contrastive (PC) loss to promote intra-class compactness. During training, our model is jointly optimized by a cross-entropy (CE) loss and the proposed PC loss.

Subsequently, we provide a detailed description of the HPG module, PDO module and each loss.

3.2 Hub Prototype Generation

To mitigate prototype bias, we propose an HPG module. It first identifies frequently occurring support hubs via a Hub Point Mining (HPM) module, then generates query-relevant hub prototypes through Hub Prototype Clustering.

Hub Point Mining. HPM identifies hub points through three sequential steps, as illustrated in Figure 3(a)–(c).

Step 1: k-Nearest Neighbor Mining. Given a center point set \mathcal{C} and a neighbor point set \mathcal{N} , a bipartite graph is constructed by connecting each center point $c \in \mathcal{C}$ to its k-nearest (kNN) neighbors in \mathcal{N} via cosine similarity measure. The k-nearest neighbors of c are formulated as kNN(c, \mathcal{N}).

Step 2: Hubness Score Statistic. The hubness score s(n) quantifies how frequently a point $n \in \mathcal{N}$ is selected as neighbors by all center points in \mathcal{C} , defined as:

$$s(n) = \sum_{c \in \mathcal{C}} \mathbb{1} \left(n \in kNN(c, \mathcal{N}) \right) + \varepsilon, \tag{1}$$

where $\mathbb{1}(\cdot)$ denotes the Iverson bracket indicator function, returning 1 if the condition holds and 0 otherwise. A small positive constant ε is added to avoid zero scores caused by potential outliers, ensuring s(n) > 0 for all n. The collective hubness scores for all points in \mathcal{N} are denoted as $s(\mathcal{N})$.

Step 3: Hub Point Selection. To identify nodes most frequently regarded as neighbors by center points, we select a subset $\mathcal{H} \subseteq \mathcal{N}$ consisting of the top- η neighbor points with the highest hubness scores, defined as:

$$\mathcal{H} = \{ n \in \mathcal{N} \mid s(n) \in \text{Top-}\eta(s(\mathcal{N})) \}. \tag{2}$$

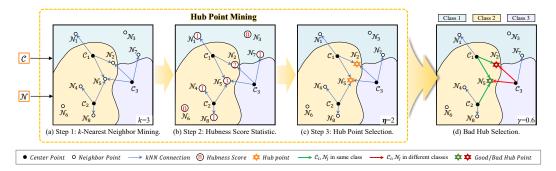


Figure 3: Illustration of Hub Point Mining and Bad Hub Selection modules. We give an example under hyperparameters k=3, $\eta=2$, and $\gamma=0.6$. (a)–(c) Hub Point Mining: using center points $\{C_1, C_2, C_3\}$ and neighbor points $\{N_1, \ldots, N_8\}$ as input, a kNN graph is constructed with k=3. After calculating hubness scores, the top $\eta=2$ points with the highest hubness scores are selected as hubs. (d) Bad Hub Selection: Hubs with purity below the threshold $\gamma=0.6$ are selected as bad hubs.

Hub Prototype Clustering. Before prototype clustering, we treat points in the query set Q and support set S as input node sets C and N of the HPM module, and select Top- η hub points for each class to construct hub point set H.

A prototype set $P = P_{fg} \cup P_{bg}$, where P_{fg} and P_{bg} denote foreground/background prototypes, is generated by conducting point-to-seed clustering (Zhao et al., 2021b) on support features localized around each hub point, defined as:

$$P_{fg} = \mathcal{F}_{clus} \left(F_s \odot M_s, \mathcal{H}_{fg} \right), \mathcal{H}_{fg} = \mathcal{H} \odot M_s, P_{bg} = \mathcal{F}_{clus} \left(F_s \odot \neg M_s, \mathcal{H}_{bg} \right), \mathcal{H}_{bg} = \mathcal{H} \odot \neg M_s,$$
(3)

where \odot denotes the Hadamard product; M_s and $\neg M_s$ are the GT mask and its inverse for support set; \mathcal{H}_{fg} and \mathcal{H}_{bg} are foreground/background hub point subsets; and \mathcal{F}_{clus} denotes the clustering operation. After that, we obtain η prototypes per class, yielding a total of $(C+1) \cdot \eta$ prototypes.

Notably, although support hubs \mathcal{H} originate from S, they are geometrically closer to points in Q as they retain only those support points that best match the query distribution. Consequently, the derived hub prototypes are more aligned with Q in the metric space, facilitating improved prototypequery matching and enhanced segmentation performance.

3.3 PROTOTYPE DISTRIBUTION OPTIMIZATION

In the PDO module, we select potential bad hubs, and then adopt a Purity-reweighted Contrastive (PC) loss to suppress these bad hubs and optimize the prototype distribution.

Bad Hub Selection. To select bad hubs, we construct a global association graph via kNN algorithm, where we merge points from Q and S to form the center point set, treat S as neighbor point set \mathcal{N} , and then utilize the HPM module to identify hub points \mathcal{H} from support set S.

After that, we identify all potential bad hubs within \mathcal{H} , which with stronger connections to center points belonging to different classes, as shown in Figure 3(d). This process involves three steps:

First, we compute the number of times each hub h is connected to center points of the same class, denoted as t(h):

$$t(h) = \sum_{c \in \mathcal{C}} \mathbb{1} \left(h \in kNN(c, \mathcal{N}) \right) \cdot \mathbb{1} \left(M_h = M_c \right), \tag{4}$$

where M_c , M_h are class labels of c and h, respectively.

Next, a purity $\mathscr{P}(h)$ is defined to represent the proportion of connections to center points of the same class, given by:

$$\mathscr{P}(h) = t(h)/s(h). \tag{5}$$

Finally, the bad hub point set \mathcal{BH} is filtered out using a purity threshold $\gamma \in (0,1)$, formulated as:

$$\mathcal{BH} = \{ h \in \mathcal{H} \mid \mathscr{P}(h) < \gamma \}. \tag{6}$$

Purity-reweighted Contrastive Loss. To further eliminate the influence of bad hubs and outlier prototypes, we aim to pull them back to their cluster centers. A typical solution is contrastive learning (Wang & Liu, 2021), which has been widely adopted in various areas to pull positive pairs closer and push negative pairs apart. Despite these successes, contrastive loss has not been explored in FS-3DSeg. However, directly applying standard contrastive loss to bad hub anchors is suboptimal: low-purity anchors, which have high similarity to samples from other classes, tend to lie far from their true class centers and are more likely to be confused, thus requiring stronger guidance to be correctly aligned. To tackle this issue, we propose a Purity-reweighted Contrastive (PC) loss, which dynamically adjusts the attraction strength based on the purity of each anchor sample.

We first introduce a purity reweighting factor w(a) to quantify the strength with which anchor $a \in \mathcal{A} = \{P_{fq} \cup \mathcal{BH}\}$ is pulled toward positive prototypes, formulated as:

$$w(a) = \begin{cases} 1 - \mathscr{P}(a) & \text{if } a \in \mathcal{BH} \\ 1 & \text{otherwise } a \in P_{fg} \end{cases}, \tag{7}$$

where the first line assigns a higher weight $(w(a) \to 1)$ inversely proportional to purity for bad hub anchors $(a \in \mathcal{BH})$ with low purity $(\mathscr{P}(a) \to 0)$, strongly pulling them toward class centers; the second line assigns a fixed weight (w(a) = 1) for all foreground prototype anchors $(a \in P_{fg})$, ensuring intra-class compactness and inter-class discriminability.

Based on the defined purity reweighting factor w(a), we formulate our purity-reweighted contrastive loss for all anchors \mathcal{A} as follows,

$$\mathcal{L}_{PC} = -\frac{1}{|\mathcal{A}|} \sum_{a \in \mathcal{A}} \log \frac{w(a) \cdot \sum_{p \in P^{+}(a)} \exp(\sin(a, p)/\tau)}{w(a) \cdot \sum_{p \in P^{+}(a)} \exp(\sin(a, p)/\tau) + \sum_{p \in P^{-}(a)} \exp(\sin(a, p)/\tau)}, \quad (8)$$

where $P^+(a) = \{p \in P \mid M_p = M_a\}$ and $P^-(a) = \{p \in P \mid M_p \neq M_a\}$ are positive set and negative set, respectively; τ is a temperature parameter that controls smoothing; sim(a, p) is the similarity between a and p.

3.4 Total Loss

During training, the proposed model is supervised by two loss functions, *i.e.*, a standard cross-entropy loss \mathcal{L}_{CE} that serves to optimize segmentation results, and the proposed PC loss \mathcal{L}_{PC} to optimize the prototype distributions. Specifically, the cross-entropy loss \mathcal{L}_{CE} is defined as:

$$\mathcal{L}_{CE} = -\frac{1}{L} \sum_{l=1}^{L} M_q^l \log(\hat{M}_q^l), \tag{9}$$

where M_q^l and \hat{M}_q^l represent GT mask and the predicted mask for query sample.

Overall loss is a weighted combination of \mathcal{L}_{CE} and \mathcal{L}_{PC} with a balancing weight λ , represented as:

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{CE}} + \lambda \cdot \mathcal{L}_{\text{PC}}.$$
 (10)

4 EXPERIMENTS

4.1 IMPLEMENTATION DETAILS

Training. Experiments were conducted on the S3DIS (Armeni et al., 2016) and ScanNet (Dai et al., 2017) datasets. Data preprocessing follows An et al. (2024): rooms are divided into 1m × 1m blocks, input points grid-sampled at 0.02m intervals, and 20,480 points are selected after voxelization to standardize input size. Data augmentation and backbone pre-training were applied as in An et al. (2024), with each fold pre-trained for 100 epochs. Meta-training was performed over 40,000 episodes using AdamW with a learning rate of 5×10^{-5} and weight decay of 0.01. For testing, 1,000 episodes per class were sampled in 1-way settings, and 100 episodes per combination in 2-way settings. We used 100 prototypes per class ($\eta = 100$); in k-shot settings (k > 1), η/k prototypes were selected from each shot and concatenated to form the final prototypes.

Parameters. In both the HPG and PDO modules, the number of neighbors for hub point mining is set to k=5. In the HPG module, the number of hub points/prototypes per class is set to $\eta=100$. In the PDO module, the bad hub purity threshold is set to $\gamma=0.6$ in (Eq. 6). In the total loss function (Eq. 10), the balance weight for \mathcal{L}_{PC} is set to $\lambda=0.1$.

Methods	1-way 1-shot		1-way 5-shot		2-way 1-shot		2-way 5-shot					
	S^0	S^1	mean	S^0	S^1	mean	S^0	S^1	mean	S^0	S^1	mean
AttMPTI (Zhao et al., 2021b)	36.32	38.36	37.34	46.71	42.70	44.71	31.09	29.62	30.36	39.53	32.62	36.08
QGE (Ning et al., 2023)	41.69	39.09	40.39	50.59	46.41	48.50	33.45	30.95	32.20	40.53	36.13	38.33
QGPA (He et al., 2023)	35.50	35.83	35.67	38.07	39.70	38.89	25.52	26.26	25.89	30.22	32.41	31.31
$Seg\text{-}PN^{\dagger}$ (Zhu et al., 2024)	37.01	40.43	38.72	39.72	43.02	41.37	33.21	37.02	35.12	39.08	39.16	39.12
COSeg (An et al., 2024)	46.31	48.10	47.21	51.40	48.68	50.04	37.44	36.45	36.95	42.27	38.45	40.36
COSeg [†]	45.93	47.48	46.71	48.47	48.72	48.60	37.17	37.03	37.10	41.65	38.38	40.02
QHP (ours)	50.33	48.73	49.53	52.27	49.64	50.96	38.86	37.84	38.35	43.90	40.04	41.97

Table 1: Comparison of mIoU (%) performance between our method and previous FS-3DSeg approaches on the **S3DIS** dataset. Methods with † are re-implementation using their official code. The best results are highlighted in **bold**, and the second-best results are marked in blue.

Methods	1-way 1-shot		1-way 5-shot		2-way 1-shot		2-way 5-shot					
	S^0	S^1	mean	S^0	S^1	mean	S^0	S^1	mean	S^0	S^1	mean
AttMPTI (Zhao et al., 2021b)	34.03	30.97	32.50	39.09	37.15	38.12	25.99	23.88	24.94	30.41	27.35	28.88
QGE (Ning et al., 2023)	37.38	33.02	35.20	45.08	41.89	43.49	26.85	25.17	26.01	28.35	31.49	29.92
QGPA (He et al., 2023)	34.57	33.37	33.97	41.22	38.65	39.94	21.86	21.47	21.67	30.67	27.69	29.18
$Seg\text{-}PN^{\dagger}$ (Zhu et al., 2024)	33.98	29.45	31.72	37.24	31.78	34.51	28.20	26.72	27.46	35.52	30.40	32.96
COSeg (An et al., 2024)	41.73	41.82	41.78	48.31	44.11	46.21	28.72	28.83	28.78	35.97	33.39	34.68
COSeg [†]	40.57	41.94	41.26	49.43	43.57	46.50	28.06	28.92	28.49	35.49	34.03	35.06
QHP (ours)	40.70	42.92	41.81	50.10	44.80	47.45	28.45	29.07	28.76	36.11	34.30	35.21

Table 2: Comparison of mIoU (%) performance between our method and previous FS-3DSeg approaches on the **ScanNet** dataset. Methods with [†] are re-implementation using their official code. The best results are highlighted in **bold**, and the second-best results are marked in **blue**.

4.2 COMPARISON RESULTS

Comparison with Previous Methods. We compare our QHP with prior works including AttMPTI (Zhao et al., 2021b), QGE (Ning et al., 2023), QGPA (He et al., 2023), Seg-PN (Zhu et al., 2024) and COSeg (An et al., 2024) on S3DIS (Armeni et al., 2016) and ScanNet (Dai et al., 2017) datasets. Additionally, we exclude MM-FSS (An et al., 2025) method as it uses multimodal information, yielding an unfair comparison. The Seg-PN results presented in Table 1 and Table 2 are reproduced based on the corrected few-shot setting proposed by COSeg (An et al., 2024), and the Seg-NN results in Table 4 of Section 4.3 are obtained using the same setting.

- S3DIS. Table 1 shows that our QHP consistently outperforms prior approaches across all settings. Specifically, compared to the baseline method $COSeg^{\dagger}$, QHP achieves performance gains of 2.82% and 2.36% in the 1-way settings, and enhancements of 1.25% and 1.95% in the 2-way settings. These gains can be attributed to the hub prototypes generated by our method. Unlike $COSeg^{\dagger}$, which relies on FPS-based prototype generation that may produce redundant or irrelevant prototypes, HPG module effectively identifies hub points to generate query-relevant prototypes. Additionally, optimizing prototype distributions further contributes to the superior performance of our model. When compared to query-guided methods such as QGE and QGPA, QHP demonstrates more significant advantages in the 1-way tasks, with improvements of 9.14% and 2.46%, respectively. This highlights the superiority of our method in enhancing the discriminability of prototypes.
- ScanNet. Table 2 shows that QHP outperforms all prior methods across all settings, further validating the effectiveness and applicability of our approach. Notably, in the 1-way 5-shot task, QHP achieves a mIoU of 47.45% and outperforms COSeg[†] by 0.95%, which highlights QHP's adaptability to the complex ScanNet dataset. We note that our performance gains are more pronounced in 5-shot than 1-shot settings: with more support samples available, QHP can mine important hub points from a larger pool of support points to generate query-relevant prototypes. However, improvements on ScanNet are less substantial than S3DIS, as the dataset's higher complexity and inter-class overlap pose greater challenges for distinguishing similar categories.

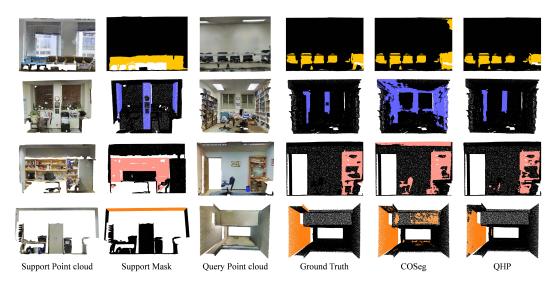


Figure 4: Qualitative comparisons between our proposed model QHP and COSeg. Each row, from top to bottom, represents the 1-way 1-shot task with the target category as chair (yellow), column (blue), bookcase (pink), and ceiling (orange), respectively.

Baseline	HPG	PDO	mIoU (%)
\checkmark			45.93
\checkmark	\checkmark		47.37
\checkmark	\checkmark	\checkmark	50.33

Methods	S^0	S^1	mean
COSeg [†]	45.93	47.48	46.71
Ours	47.37	48.12	47.73
Seg-NN [†]	25.86	30.54	28.20
Ours	28.28	30.99	29.64

Table 3: Ablation study of key components in QHP method.

Table 4: Performance evaluation under different baselines (1-way 1-shot setting).

Qualitative Results. In Figure 4, we compare the results from our QHP (6th column) with COSeg (5th column). QHP improves object boundaries and category shapes, especially for column contours (blue, 2nd row), capturing finer details and reducing redundancy. PDO module excels in chair class (yellow, 1st row), resolving boundary ambiguities for more precise segmentation. Overall, QHP delivers cleaner, more accurate results with improved boundary delineation and reduced redundancy.

4.3 ABLATION STUDY

We present an ablation study on the S3DIS dataset under 1-way 1-shot S^0 setting to validate the effectiveness of HPG and PDO modules, as well as hyperparameter settings.

Effects of Core Components. Using COSeg as the baseline, we conduct experiments to evaluate the effectiveness of the two core components, *i.e.*, the HPG and PDO modules. As shown in Table 3, incorporating the HPG module improves the mIoU from 45.93% to 47.37% (+1.44%), while the addition of the PDO module further increases the mIoU to 50.33% (+2.96%), showing that the joint use of HPG and PDO significantly enhances the query relevance and discriminability of the prototypes, thereby leading to substantial performance gains.

Evaluation on Additional Baselines. To demonstrate the generalization capability of our method, we apply it to additional baselines, including SegNN Zhu et al. (2024), as shown in Table 4. We extend SegNN's single-prototype approach to a multi-prototype variant by replacing it with a method that employs FPS and local clustering. Since SegNN is a non-parametric baseline that does not use loss functions for training, we only apply our HPG module. The results show that our method outperforms Seg-NN, demonstrating the generalization capability of our approach.

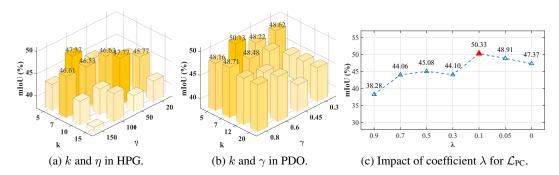


Figure 5: Sensitivity analysis of parameters in HPG, PDO and the coefficient λ for \mathcal{L}_{PC} .

Hub Ratio	0%	30%	50%	80%	100%
mIoU (%)	45.93	42.93	42.32	45.47	47.37

Diff. Losses	Contrastive Loss	Our PC Loss
mIoU (%)	49.82	50.33

Table 5: Analyze the ratio of hub prototypes to total prototypes in HPG.

Table 6: Comparison between PC loss and contrastive loss.

Effects of Hub Prototypes from HPG. Without using the PDO module, we fix the number of prototypes to 100 per class and mix FPS-based prototypes with our hub prototypes in varying ratios. As the hub prototype ratio increases from 0% (*i.e.*, baseline COSeg) to 50%, performance slightly drops, likely because prototype class diversity is reduced while the query relevance of prototypes remains limited, causing suboptimal performance. Beyond 50%, performance steadily improves, peaking at 100%, demonstrating that hub prototypes better capture support-query semantic correlation and provide more discriminative representations, significantly enhancing segmentation.

Impact of Parameters k and η in HPG. We analyze the impact of the number of neighborhoods k in Eq. 1 and the number of hub points η in Eq. 2, as shown in Figure 5a. The best performance is achieved when k=12 and $\eta=50$, followed by k=5 and $\eta=100$. For fair comparison with other multi-prototype methods with 100 prototypes, we select the setting k=5 and $\eta=100$.

Impact of Parameters k and γ in PDO. We analyze the impact of the number of neighborhood k and purity threshold γ in PDO, as shown in Figure 5b. The best performance is achieved when k=5 and $\gamma=0.6$; thus, these parameters are selected in our setup.

Effects of Different Contrastive Losses in PDO. To verify the superiority of our proposed PC loss, we compared it with the standard contrastive loss in the PDO module, as shown in Table 6. The PC loss yields a 0.51% performance improvement, demonstrating that the reweighting factor in our PC loss more effectively pulls outlier prototypes and bad hubs near cluster boundaries toward their respective class centers, reducing boundary ambiguity and enhancing overall performance.

Impact of Coefficient λ in Total Loss. Figure 5c illustrates the impact of the weight λ of the PC loss in Eq. 10. Value of $\lambda=0.1$ yields the best results, indicating that the model achieves a balance between class boundary distribution and query segmentation. However, continued increases in λ could pull prototypes and hub points too tightly into clusters, thereby harming model performance.

5 CONCLUSION

We propose a Query-aware Hub Prototype (QHP) framework for few-shot 3D point cloud semantic segmentation, addressing limitations of prior methods relying solely on support prototypes. QHP models semantic correlations between support and query sets to enhance prototype relevance, with two key modules: Hub Prototype Generation (HPG) module, identifying high-frequency hub points from support to generate query-relevant prototypes; and Prototype Distribution Optimization (PDO) module, reducing the impact of bad hubs and ambiguous prototypes via purity-reweighted contrastive loss. Experiments on S3DIS and ScanNet show the superiority of the proposed model.

REFERENCES

- Zhaochong An, Guolei Sun, Yun Liu, Fayao Liu, Zongwei Wu, Dan Wang, Luc Van Gool, and Serge Belongie. Rethinking few-shot 3d point cloud semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3996–4006, 2024.
- Zhaochong An, Guolei Sun, Yun Liu, Runjia Li, Min Wu, Ming-Ming Cheng, Ender Konukoglu, and Serge J. Belongie. Multimodality helps few-shot 3d point cloud semantic segmentation. In *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025.* OpenReview.net, 2025. URL https://openreview.net/forum?id=jXvwJ51vcK.
- 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 (CVPR)*, pp. 1534–1543, 2016.
- Simion-Vlad Bogolin, Ioana Croitoru, Hailin Jin, Yang Liu, and Samuel Albanie. Cross modal retrieval with querybank normalisation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5184–5195, 2022.
- Ali Cheraghian, Shafin Rahman, Dylan Campbell, and Lars Petersson. Mitigating the hubness problem for zero-shot learning of 3d objects. In *Proceedings of the British Machine Vision Conference* (*BMVC*), pp. 41, 2019.
- 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 (CVPR)*, pp. 5828–5839, 2017.
- Georgiana Dinu and Marco Baroni. Improving zero-shot learning by mitigating the hubness problem. In *International Conference on Learning Representations (ICLR)*, 2015.
- Nanyi Fei, Yizhao Gao, Zhiwu Lu, and Tao Xiang. Z-score normalization, hubness, and few-shot learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 142–151, October 2021.
- Shuting He, Xudong Jiang, Wei Jiang, and Henghui Ding. Prototype adaption and projection for fewand zero-shot 3d point cloud semantic segmentation. *IEEE Transactions on Image Processing* (*TIP*), 32:3199–3211, 2023. doi: 10.1109/TIP.2023.3279660.
- 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 (CVPR)*, pp. 11105–11114, June 2020.
- Xin Lai, Jianhui Liu, Li Jiang, Liwei Wang, Hengshuang Zhao, Shu Liu, Xiaojuan Qi, and Jiaya Jia. Stratified transformer for 3d point cloud segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 8500–8509, 2022.
- 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 (NeurIPS)*, pp. 828–838, 2018.
- Zhi-Hao Lin, Sheng-Yu Huang, and Yu-Chiang Frank Wang. Convolution in the cloud: Learning deformable kernels in 3d graph convolution networks for point cloud analysis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1797–1806, June 2020.
 - Jie Liu, Wenzhe Yin, Haochen Wang, Yunlu CHen, Jan-Jakob Sonke, and Efstratios Gavves. Dynamic prototype adaptation with distillation for few-shot point cloud segmentation. In *International Conference on 3D Vision (3DV)*, pp. 810–819, 2024.

- Yongqiang Mao, Zonghao Guo, Lu Xiaonan, Zhiqiang Yuan, and Haowen Guo. Bidirectional feature globalization for few-shot semantic segmentation of 3d point cloud scenes. In *International Conference on 3D Vision (3DV)*, pp. 505–514, 2022.
 - Zhenhua Ning, Zhuotao Tian, Guangming Lu, and Wenjie Pei. Boosting few-shot 3d point cloud segmentation via query-guided enhancement. In *Proceedings of the ACM International Conference on Multimedia (ACMMM)*, MM '23, pp. 1895–1904, 2023.
 - 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 (CVPR)*, pp. 652–660, 2017a.
 - Charles Ruizhongtai 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 (NeurIPS)*, pp. 5099–5108, 2017b.
 - Guocheng Qian, Yuchen Li, Houwen Peng, Jinjie Mai, Hasan Hammoud, Mohamed Elhoseiny, and Bernard Ghanem. Pointnext: Revisiting pointnet++ with improved training and scaling strategies. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022.
 - Miloš Radovanović, Alexandros Nanopoulos, and Mirjana Ivanović. Nearest neighbors in high-dimensional data: The emergence and influence of hubs. In *International conference on machine learning (ICML)*, pp. 865–872, 2009.
 - Milos Radovanovic, Alexandros Nanopoulos, and Mirjana Ivanovic. Hubs in space: Popular nearest neighbors in high-dimensional data. *Journal of Machine Learning Research (JMLR)*, 11(sept): 2487–2531, 2010.
 - Wenyi Tang, Haocheng Pei, Xin Wang, Zaobo He, Lei Yu, and Xinsong Yang. Reducing hubness to improve inductive few-shot learning. *Neurocomputing*, 651:130879, 2025. ISSN 0925-2312. doi: https://doi.org/10.1016/j.neucom.2025.130879. URL https://www.sciencedirect.com/science/article/pii/S0925231225015516.
 - Hugues Thomas, Charles R. Qi, Jean-Emmanuel Deschaud, Beatriz Marcotegui, Francois Goulette, and Leonidas J. Guibas. Kpconv: Flexible and deformable convolution for point clouds. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 6410–6419, October 2019.
 - Daniel J. Trosten, Rwiddhi Chakraborty, Sigurd Løkse, Kristoffer Knutsen Wickstrøm, Robert Jenssen, and Michael C. Kampffmeyer. Hubs and hyperspheres: Reducing hubness and improving transductive few-shot learning with hyperspherical embeddings. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 7527–7536, June 2023.
 - Feng Wang and Huaping Liu. Understanding the behaviour of contrastive loss. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2495–2504, June 2021.
 - Yimu Wang, Xiangru Jian, and Bo Xue. Balance act: Mitigating hubness in cross-modal retrieval with query and gallery banks. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 10542–10567, 2023.
 - Xiaoyang Wu, Yixing Lao, Li Jiang, Xihui Liu, and Hengshuang Zhao. Point transformer v2: Grouped vector attention and partition-based pooling. In *Advances in Neural Information Processing Systems (NeurIPS)*, pp. 33330–33342, 2022.
- Chenxi Xiao, Naveen Madapana, and Juan Wachs. One-shot image recognition using prototypical encoders with reduced hubness. In *2019 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pp. 2251–2260, 2021.
 - Zheming Xu, He Liu, Congyan Lang, Tao Wang, Yidong Li, and Michael C. Kampffmeyer. A hubness perspective on representation learning for graph-based multi-view clustering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 15528–15537, June 2025.

- 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 (ICCV)*, pp. 16259–16268, October 2021a.
- 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 (CVPR)*, pp. 8873–8882, 2021b.
- Guanyu Zhu, Yong Zhou, Rui Yao, and Hancheng Zhu. Cross-class bias rectification for point cloud few-shot segmentation. *IEEE Transactions on Multimedia (TMM)*, 25:9175–9188, 2023. doi: 10.1109/TMM.2023.3248150.
- Xiangyang Zhu, Renrui Zhang, Bowei He, Ziyu Guo, Jiaming Liu, Han Xiao, Chaoyou Fu, Hao Dong, and Peng Gao. No time to train: Empowering non-parametric networks for few-shot 3d scene segmentation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2024, Seattle, WA, USA, June 16-22, 2024*, pp. 3838–3847. IEEE, 2024. doi: 10.1109/CVPR52733.2024.00368. URL https://doi.org/10.1109/CVPR52733.2024.00368.

A APPENDIX

In the appendix, we present more experiments focused on the effects of hub points, further validating the effectiveness of our method. Besides, we also compare the computational overhead between our proposed method and the baseline. Furthermore, we give experimental details and analysis, including model sizes, inference time, frames per second (FPS) and floating point operations (FLOPs). We also provide extended qualitative comparison results, which demonstrate the superiority of our approach in resolving boundary ambiguity and achieving more precise segmentation.

Notably, we used large language models (LLMs) solely to aid in drafting and polishing the writing of this paper. All scientific content, experimental design, and results are original to this work.

A.1 MORE DETAILS ABOUT HUB POINTS

Trend in the number of bad hub points. Figure 6 depicts the variation trend of bad hub points during training, with the total number of hubs fixed at 100. The curve clearly shows that the number of bad hubs drops rapidly in the early training stages and later gradually stabilizes. This experimental observation is consistent with the design goal of the Prototype Distribution Optimization (PDO) module, validating its efficacy in alleviating the adverse effects caused by bad hubs. Specifically, early in training, the PDO module rectifies misclassified bad hubs via purity-reweighted contrastive loss, gradually aligning them with correct class centers. As training proceeds, the HPG module strengthens semantic association modeling, reducing bad hubs and mitigating prototype bias. In later stages, the stabilization of bad hub counts indicates that the model has captured core features of each class, thereby boosting the performance of FS-3DSeg.

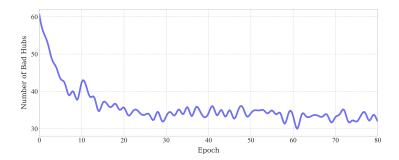


Figure 6: Trends in the number of bad hubs during training.

Impact of Different Hubs in HPG. In the HPG module, to verify the impact of good/bad hubs on hub prototype quality, we use only good hubs versus all hubs (good + bad) in prototype clustering, as in Table 7. Results show that performance with prototypes from only good hubs is suboptimal compared to all hubs. We analyze that prototypes derived solely from good hubs, while effective in reducing noise interference during testing, limit training by capturing only intra-class semantics, weakening class discrimination. In contrast, incorporating bad hubs covers overlooked inter-class discriminative information from class boundaries and complex scenarios, boosting training by refining the model's discriminative ability and yielding a 3.04% improvement. Notably, when combined with the PDO module, bad hub distribution is further optimized, avoiding interference from excessive outliers during testing.

Types of Hubs Good	l hubs All hubs	Types of Anchors	P_{fg}	P_{fg} & Bad hubs
mIoU (%) 44	.33 47.37	mIoU (%)	48.58	50.33

Table 7: Impact of different hubs in HPG.

Table 8: Impact of different anchors in PC Loss.

Impact of Different Anchors in PC Loss. In PC loss, using both foreground prototypes and bad hubs as anchors improves performance by 1.75% compared to using only foreground prototypes (Table 8). Our analysis reveals that relying solely on foreground prototypes as anchors prioritizes intra-class compactness but underemphasizes semantic associations in boundary regions. In contrast,

introducing bad hub anchors enables PC loss to further emphasize boundaries/outliers, pull bad hubs to class centers, accommodate complex scenario ambiguities, break isolation between outlier prototypes and bad hubs, and boost cross-set alignment.

A.2 COMPUTATIONAL COMPLEXITY

In Table 9, we present the number of parameters and computational complexity of our proposed method. Compared to the baseline COSeg, our approach does not introduce additional parameters but achieves higher performance. Notably, our method effectively reduces inference time and improves the FPS value. This is because the farthest point sampling (FPS) strategy adopted in COSeg requires substantial time to search for the farthest points from the support set, whereas our model can quickly sample prototype centers through the hub point mining (HPM) method, thus achieving faster speed and optimal performance.

Methods	#Params	Inference Time (s)	FPS	FLOPs (G)	mIoU (%)
COSeg [†]	6.11M	0.27	3.66	14.20	45.93
QHP	6.11M	0.26	3.83	14.64	50.33

Table 9: Analysis of computational cost and experimental results on the S3DIS dataset under 1-way 1-shot S^0 setting.

A.3 MORE EXPERIMENT DETAILS

We conducted experiments on the S3DIS and ScanNet datasets. The S3DIS dataset Armeni et al. (2016) comprises five large-scale indoor areas across three buildings, annotated with 12 semantic classes for segmentation tasks. The ScanNet dataset Dai et al. (2017) contains 1,513 point cloud scans from 707 indoor scenes, covering 20 semantic categories. Compared to S3DIS, ScanNet features more irregular point clouds, rendering segmentation more challenging. Following An et al. (2024), we also adopt the first three blocks of the Stratified Transformer (Lai et al., 2022) as the backbone of our model. For the S3DIS and ScanNet datasets, the input features consist of both XYZ coordinates and RGB colors. All settings are implemented in PyTorch. The 1-way and 2-way 1-shot settings are trained on 4 RTX A4000 GPUs, while the 2-way 5-shot setting is trained on 4 RTX 3090 GPUs.

In Section 4.3, we conduct extensive ablation studies on the parameters. The detailed results of the parameter analysis are presented in Table 10 and Table 11, with specific sections illustrated in Figure 5a and Figure 5b.

mIoU (%)	0.3	0.45	0.6	0.8
5	41.71	44.52	47.37	43.24
7	44.41	46.63	46.53	46.61
10	45.77	47.77		
15	41.98	40.98	43.67	38.34

mIoU (%)	0.3	0.45	0.6	0.8
5	48.62	48.22	50.13 48.48 47.05 47.62	48.16
7	44.93	47.97	48.48	48.71
12	45.13	42.52	47.05	47.95
20	45.24	43.69	47.62	48.07

Table 10: Experimental Results on Parameter Sensitivity of HPG (column: k, row: η).

Table 11: Experimental Results on Parameter Sensitivity of PDO (column: k, row: γ).

A.4 MORE QUALITATIVE RESULTS

As shown in Figure 7, we present more results of QHP (6th column) compared to COSeg (5th column). QHP consistently demonstrates superior segmentation performance across various scenarios. For instance, in the cases of chairs and bookcases, QHP can more accurately delineate object boundaries, while COSeg tends to produce fragmented or incomplete masks. In cluttered environments, such as walls and floors, QHP is more effective at preserving the overall structure, reducing redundancy, and avoiding ambiguity near class boundaries. This ensures that QHP produces a cleaner and more precise representation compared to COSeg, especially in terms of object completeness.

 These results underscore QHP's substantial improvements in object boundary delineation and category shape representation. Its ability to capture finer details and reduce redundancy allows for more precise segmentation.

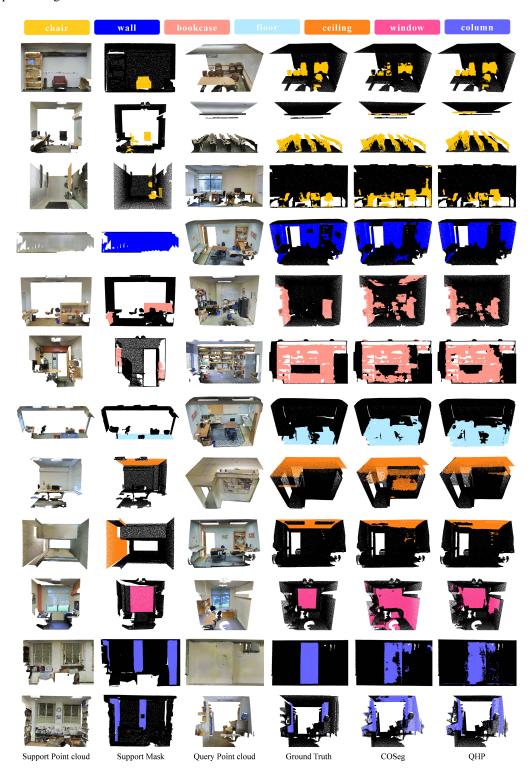


Figure 7: More qualitative comparisons between our proposed model QHP and COSeg.