Rethinking Few-shot 3D Point Cloud Semantic Segmentation

Zhaochong An[1,2], Guolei Sun[1*], Yun Liu[3*], Fayao Liu[3], Zongwei Wu[4],
Dan Wang[2], Luc Van Gool[1], Serge Belongie[2]
[1] Computer Vision Laboratory, ETH Zurich
[2] Pioneer Centre for Artificial Intelligence, University of Copenhagen
[3] Institute for Infocomm Research, A*STAR
[4] Computer Vision Lab, CAIDAS & IFI, University of Wurzburg

Abstract

This paper revisits few-shot 3D point cloud semantic segmentation (FS-PCS), with a focus on two significant issues in the state-of-the-art: foreground leakage and sparse point distribution. The former arises from non-uniform point sampling, allowing models to distinguish the density disparities between foreground and background for easier segmentation. The latter results from sampling only 2,048 points, limiting semantic information and deviating from the real-world practice. To address these issues, we introduce a standardized FS-PCS setting, upon which a new benchmark is built. Moreover, we propose a novel FS-PCS model. While previous methods are based on feature optimization by mainly refining support features to enhance prototypes, our method is based on correlation optimization, referred to as Correlation Optimization Segmentation (COSeg). Specifically, we compute Class-specific Multi-prototypical Correlation (CMC) for each query point, representing its correlations to category prototypes. Then, we propose the Hyper Correlation Augmentation (HCA) module to enhance CMC. Furthermore, tackling the inherent property of few-shot training to incur base susceptibility for models, we propose to learn non-parametric prototypes for the base classes during training. The learned base prototypes are used to calibrate correlations for the background class through a Base Prototypes Calibration (BPC) module. Experiments on popular datasets demonstrate the superiority of COSeg over existing methods. The code is available at github.com/ZhaochongAn/COSeg.

1. Introduction

Rapid advancements in deep neural networks have propelled the exploration of 3D point cloud understanding in various applications [5, 8, 26, 31]. Unlike images, point clouds inherently capture intricate object structures, enabling fine-grained analyses. However, collecting and annotating point cloud data is significantly more labor-intensive than its 2D counterpart, limiting the scale and semantic diversity of existing 3D datasets [1, 4, 7]. To reduce the substantial human effort required for dataset creation, *few-shot point cloud semantic segmentation* (FS-PCS) emerges as a crucial task, which empowers 3D segmentation models to generalize to novel classes with few annotated samples.

In the realm of FS-PCS, attMPTI [56] stands as a pioneering model, introducing a multi-prototype transductive approach that leverages label propagation for predicting segmentation in novel classes. Subsequent works [11,

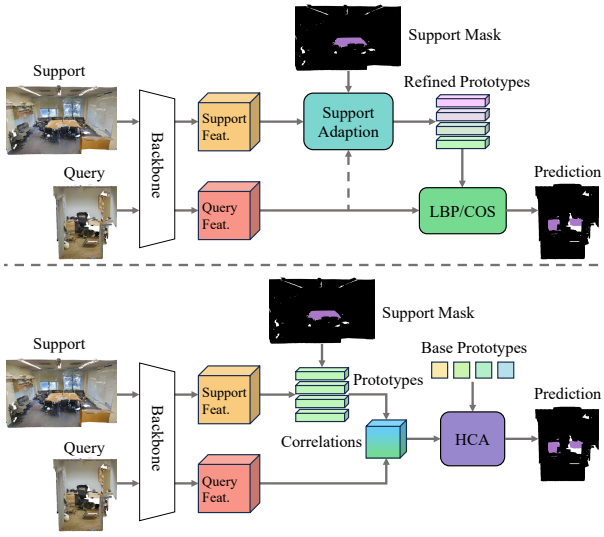


Figure 1. **Previous *feature optimization* vs. our *correlation optimization*.** *Top:* Most prior work [11, 25, 29, 45, 56, 58] on FS-PCS focuses on feature optimization by designing support adaption modules for enhanced prototypes and then making predictions through non-parametric label propagation (LBP) or cosine similarity (COS), implicitly modeling correlations. *Bottom:* Instead of optimizing features, we propose to directly uses correlations as input to learnable modules, explicitly refining correlations.

*Corresponding authors: Guolei Sun and Yun Liu

25, 29, 45, 51, 58] have consistently built upon the attMPTI framework, progressively improving overall performance.

However, we identify two significant issues in the current FS-PCS setting: (1) The first issue is the *foreground leakage*. The common 3D segmentation practice [18, 55] feeds models with randomly sampled points from the scene, but the sampling process in FS-PCS is non-uniform, favoring more points in the foreground than in the background. This leads to foreground leakage, a noticeable density bias toward foreground classes. This leakage allows previous models to exploit density disparities for easier segmentation, sidestepping the need to learn essential knowledge adaptation patterns for novel classes. Consequently, this issue renders the current benchmark unable to reflect the true performance of previous models. (2) The second issue is the *sparse point distribution*. The current setting samples only 2,048 points during both training and inference due to the huge computational burden in the label propagation module adopted by many FS-PCS methods [29, 45, 56]. However, this sparse input distribution limits the semantic information available to models, hindering effective advances to improve their recognition ability. In addition, this input deviation from real-world scenes diminishes the overall value of research progress in this domain.

To steer the research in the right direction, we standardize the FS-PCS task by proposing a more rigorous setting. Specifically, we correct the foreground leakage and improve the framework by enabling the models to process a large number of points, aligning it more closely with real-world scenes. In this well-justified setting, we systematically reevaluate existing methods, establishing a new valid benchmark for future research.

We further introduce a novel FS-PCS model, named **Correlation Optimization Segmentation (COSeg)**. As shown in Fig. 1, existing FS-PCS models are based on *feature optimization*, which means that they optimize support features to enhance prototypes [11, 25, 29, 45, 58] or optimize query features through fine-grained interaction with support features [51]. Instead of operating on features, we propose to optimize the **Class-specific Multi-prototypical Correlation (CMC)** computed for each query point, representing its correlations to all category prototypes. This new *correlation optimization* paradigm allows direct shaping of relationships between query points and category prototypes, leading to better generalization for FS-PCS than feature optimization. Building on CMC, we introduce the **Hyper Correlation Augmentation (HCA)** module. This module refines correlations in the hyperspace by actively interacting them across points and category prototypes.

Moreover, within the meta-learning framework [36, 38, 44] employed by FS-PCS, models undergo training on seen/base classes and are evaluated on unseen/novel classes, revealing an inherent susceptibility. Specifically, these models tend to be susceptible to the familiar base classes within the test scenes, thereby hindering the accurate segmentation of novel classes [19]. To alleviate this susceptibility, we propose a novel approach: learning prototypes for the base classes in a non-parametric and momentum-driven manner during the training phase. Our introduced **Base Prototypes Calibration (BPC)** module utilizes these learned base prototypes to calibrate correlations for the background within HCA. This calibration effectively mitigates the *base susceptibility* problem, enhancing the model's accuracy.

We systematically benchmark existing methods in our well-justified setting and compare COSeg against others on the S3DIS [1] and ScanNet [7] datasets (Sec. 5.2). Our experiments not only reveal the adverse impact of the previous task setting but also highlight the impressive performance of our method. With extensive ablation studies in Sec. 5.3, we offer further insights into the efficacy of our designs and showcase the superior capabilities of the CMC paradigm for FS-PCS, shedding light on future research.

In summary, our contributions include:

- We identify two significant issues in the current FS-PCS setting: the *foreground leakage* and *sparse point distribution*, which are standardized by our introduction of a rigorous setting and a new benchmark.
- We propose a novel *correlation optimization* paradigm operating on Class-specific Multi-prototypical Correlation (CMC), enabling the direct shaping of categorical relationships for query points using the Hyper Correlation Augmentation (HCA) module.
- We tackle the *base susceptibility* issue inherent in meta-learning by introducing non-parametric base prototypes, along with the Base Prototypes Calibration (BPC) module, to calibrate correlations for the background class.

2. Related Work

2.1. 3D Point Cloud Semantic Segmentation

Currently, numerous approaches have emerged for performing point cloud semantic segmentation in a fully supervised manner, categorized into three main groups. The first group, known as MLP-based methods [9, 10, 12, 14, 32, 33, 49], adopts a shared multi-layer perceptron (MLP) as the core building block, complemented by symmetric functions for aggregating features. In the second group, point convolution-based models [2, 13, 17, 21, 23, 24, 39, 40, 48, 54] adapt convolution kernels to the underlying local geometries due to the unordered nature of point clouds, resulting in variations in kernel adaptation techniques. A subset of them [20, 22, 27, 34, 37, 43, 46, 47, 52, 57] embraces graph-based representations to mirror the structure of point clouds. They employ graph convolutions [16] to propagate and aggregate features across the graph. The third group in-

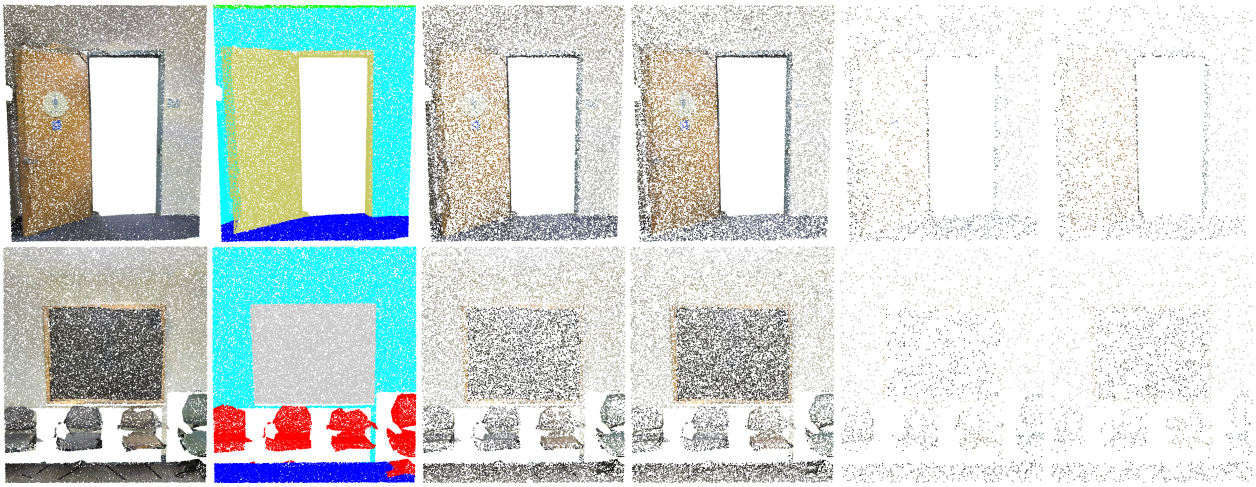


Figure 2. **Visualization of two scenes from the S3DIS dataset [1], with the foreground class as *door* and *board* for 1-way segmentation, respectively.** Each scene includes six types of point clouds, arranged *from left to right:* (1) The original point cloud; (2) Ground truth of all categories; (3) Our corrected input with 20,480 points in a uniform distribution; (4) Input with 20,480 points in a biased distribution; (5) Input with 2,048 points in a uniform distribution; (6) Input with 2,048 points in a biased distribution, as adopted by previous works.

corporates attention mechanisms [42] to model long-range dependencies, suitable for handling point cloud irregularities. As a result, various efforts [18, 28, 30, 35, 50, 55] have been dedicated to leveraging attention mechanisms for feature learning in point cloud segmentation. Notably, Stratified Transformer [18] proposes a stratified sampling strategy within the self-attention module to enlarge the receptive field without incurring significant computational costs.

2.2. Few-shot 3D Point Cloud Segmentation

Given the challenging and labor-intensive nature of point cloud data collection, the importance of FS-PCS becomes increasingly apparent. The pioneering work, attMPTI [56], employs label propagation to exploit relationships among prototypes and query points. Subsequent works further expand on this foundation. PAP [11] addresses large intra-class feature variations by directly adapting prototypes into the query feature space. QGE [29] adapts background prototypes to match the query context, followed by the holistic rectification of prototypes under the guidance of query features. 2CBR [58] leverages co-occurrence features of support and query to calculate bias terms and rectify differences between them. BFG [25] introduces bidirectional feature globalization, activating global perception of prototypes and point features to better aggregate context information. CSSMRA [45] develops a multi-resolution attention module using both the nearest and farthest points to enhance context aggregation. SCAT [51] proposes a stratified class-specific attention-based transformer, constructing fine-grained relationships between support and query features. Notably, these methods all pivot on *feature optimization*, refining either support or query features. In contrast, our approach introduces *correlation optimization* by refining the multi-prototypical support-query correlations, which exhibits superior generalization capabilities to novel classes compared to previous feature optimization.

3. FS-PCS Overview

3.1. Task Description

FS-PCS involves segmenting the foreground semantic categories in a query point cloud as specified by densely annotated support point clouds. Formally, following the episodic paradigm [44], each episode for an N-way K-shot segmentation task contains a support set $\mathcal{S} = \big\{\{\mathbf{X}_{\mathrm{s}}^{n,k}, \mathbf{Y}_{\mathrm{s}}^{n,k}\}_{k=1}^{K}\big\}_{n=1}^{N}$ and a query set $\mathcal{Q} = \{\mathbf{X}_{\mathrm{q}}^{n}, \mathbf{Y}_{\mathrm{q}}^{n}\}_{n=1}^{N}$, where $\mathbf{X}_{\mathrm{s/q}}^{*}$ and $\mathbf{Y}_{\mathrm{s/q}}^{*}$ represent a point cloud and its corresponding segmentation mask. Within $\mathcal{S}$, each K-shot group $\{\mathbf{X}_{\mathrm{s}}^{n,k}, \mathbf{Y}_{\mathrm{s}}^{n,k}\}_{k=1}^{K}$ exclusively describes the n-th semantic class of the total N foreground classes.

The objective, given $\mathcal{S}$ and $\mathbf{X}_{\mathrm{q}}^{n}$, is to predict query masks that closely match $\mathbf{Y}_{\mathrm{q}}^{n}$ by leveraging the knowledge of the N novel categories provided by $N \times K$ support pairs in $\mathcal{S}$. Two semantic category subsets C_{train} and C_{test}, with $C_{\mathrm{train}} \cap C_{\mathrm{test}} = \phi$, are employed for training and testing, respectively. Thus, training exclusively utilizes foreground classes from C_{train}, while testing employs previously unseen classes from C_{test}. The training in FS-PCS usually encompasses two stages, *i.e.*, pre-training and meta-training. The former primarily trains the backbone to learn meaningful semantic features in a fully-supervised manner, while the following meta-training focuses on training the model to transfer knowledge from the support $\mathcal{S}$ to the query $\mathcal{Q}$. Both meta-training and testing adhere to the episodic paradigm.

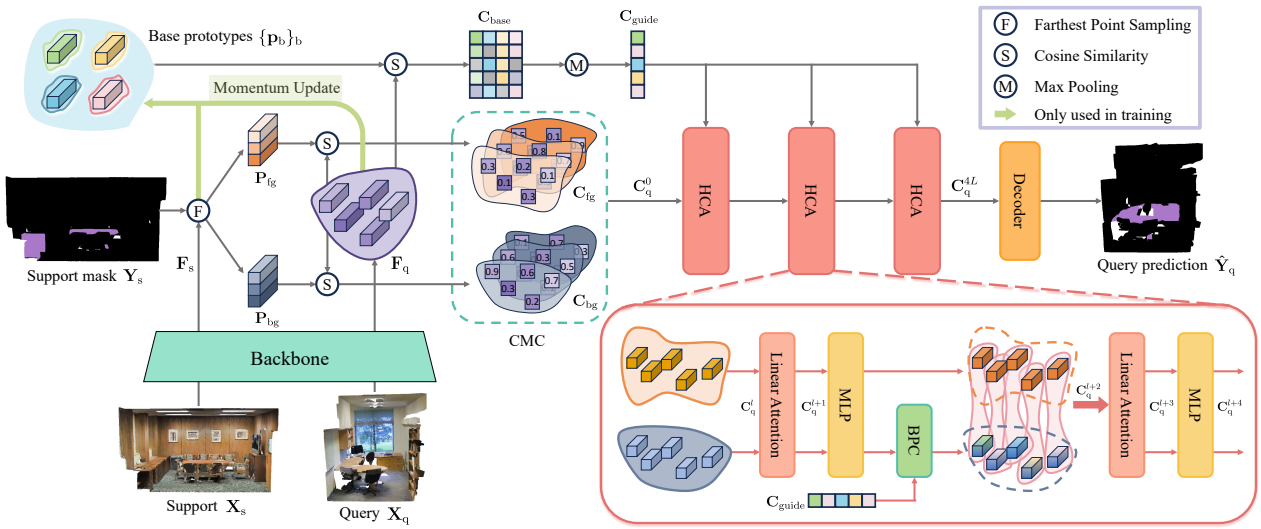


Figure 3. Overall architecture of the proposed COSeg. Initially, we compute CMC for each query point using the backbone features. These correlations are then forwarded to the subsequent HCA module, which actively mines hyper-relations among correlations across points and classes. Additionally, we dynamically learn non-parametric base prototypes on the fly and introduce the BPC module to effectively alleviate the *base susceptibility* problem. For clarity, we present the model under the 1-way 1-shot setting.

3.2. Issues in the Current Setting

The prevailing FS-PCS task setting, initially introduced by [56], has been consistently employed in subsequent works [11, 25, 29, 45, 51, 58]. Despite the previous progress, we identify two crucial issues within this setting.

Foreground Leakage. The prevailing 3D segmentation methodology [18, 55] feeds models with randomly sampled points from the scene. However, in the current FS-PCS setting, the sampling process introduces a bias toward foreground classes. Specifically, this non-uniform sampling favors foreground classes by sampling more points for them compared to the background, leading to a noticeable point density disparity between foreground and background, thereby leaking the foreground classes to the models. More details on this biased sampling are available in the supplementary material. As depicted in Fig. 2, the inputs (3), (5) from the corrected uniform sampling show balanced point distributions, while the inputs (4), (6) using biased sampling exhibit denser distributions in the foreground (*door* or *board*) than in the background. This foreground leakage induces models to segment foreground classes by identifying denser regions, instead of learning semantic knowledge transfer from support to query. This issue, occurring in both training and testing, undermines the benchmark's validity. Addressing this issue, as shown in Sec. 5.2, unveils a significant performance drop in existing methods, emphasizing the imperative need for correction.

Sparse Point Distribution. Besides, the current FS-PCS input is constrained to only 2,048 points due to the high computational cost of constructing a k-nearest neighbor graph in the label propagation module adopted by many FS-PCS methods [29, 45, 56]. However, this sparse point distribution severely limits semantic clarity, making it difficult to distinguish objects. For instance, in Fig. 2, it is even challenging for humans to distinguish the *door* and surrounding *wall* in the 2048-point input (5^{th} column) in the 1^{st} row. The same applies to the 2^{nd} row to discern *board* from other classes like *window*. These sparsely populated, semantically limited inputs introduce significant ambiguities, hindering the model's capacity to exploit semantics in the scenes. Furthermore, this deviation from real-world scenes limits the scope of current research progress.

To address these issues, we introduce a more rigorous setting for FS-PCS. In this standardized setting, we increase the number of input points tenfold to 20,480 and eliminate foreground leakage through uniform sampling. As depicted in Fig. 2, the input (3) from this rigorous setting provides clearer scene representations and uniform distributions, aligning the task setting more closely with real-world scenarios. The new benchmark results under this setting are presented in Sec. 5.2.

4. Methodology

Instead of employing the traditional *feature optimization* [11, 25, 29, 45, 51, 58], our proposed COSeg is built upon the *correlation optimization* paradigm with CMC, allowing for direct refinement of relationships between each query point and category prototypes. Fig. 3 illustrates the pipeline of COSeg. Without loss of generality, we present our model under the 1-way 1-shot setting in the following sections.

4.1. Class-specific Multi-prototypical Correlation

Given the backbone Φ, we extract support features $\mathbf{F}_s = \Phi(\mathbf{X}_s) \in \mathbb{R}^{N_S \times D}$ and query features $\mathbf{F}_q = \Phi(\mathbf{X}_q) \in$

$\mathbb{R}^{N_Q \times D}$, where D is the channel dimension, N_S and N_Q are the number of points in $\mathbf{X}_\mathrm{s}$ and $\mathbf{X}_\mathrm{q}$, respectively. Foreground prototypes $\mathbf{P}_\mathrm{fg}$ and background prototypes $\mathbf{P}_\mathrm{bg}$ are obtained through two steps: sample N_O seeds in the coordinate space based on farthest point sampling, and then conduct point-to-seed clustering [56] as follows:

$$\begin{aligned} \mathbf{P}_\mathrm{fg} &= \mathcal{F}_\mathrm{clus}(\mathbf{F}_\mathrm{s} \odot \mathbf{Y}_\mathrm{s}, \mathbf{S}_\mathrm{fg}),\ \mathbf{S}_\mathrm{fg} = \mathcal{F}_\mathrm{fps}(\mathbf{L}_\mathrm{s} \odot \mathbf{Y}_\mathrm{s}), \\ \mathbf{P}_\mathrm{bg} &= \mathcal{F}_\mathrm{clus}(\mathbf{F}_\mathrm{s} \odot \tilde{\mathbf{Y}}_\mathrm{s}, \mathbf{S}_\mathrm{bg}),\ \mathbf{S}_\mathrm{bg} = \mathcal{F}_\mathrm{fps}(\mathbf{L}_\mathrm{s} \odot \tilde{\mathbf{Y}}_\mathrm{s}), \end{aligned} \tag{1}$$

where $\odot$ is the Hadamard product, $\mathbf{L}_\mathrm{s}$ denotes the xyz coordinates of support points from $\mathbf{X}_\mathrm{s}$, $\tilde{\mathbf{Y}}_\mathrm{s}$ is the inverse mask of $\mathbf{Y}_\mathrm{s}$, and $\mathcal{F}_\mathrm{fps}$ represents the farthest point sampling operation. Continuously, $\mathbf{S}_\mathrm{fg/bg}$ is the set of indices corresponding to the seeds sampled by $\mathcal{F}_\mathrm{fps}$, and $\mathcal{F}_\mathrm{clus}$ stands for the clustering operation. After this, we have $\mathbf{P}_\mathrm{fg}, \mathbf{P}_\mathrm{bg} \in \mathbb{R}^{N_O \times D}$ with N_O prototypes per category.

Next, we compute the cosine similarities of query points with respect to $\mathbf{P}_\mathrm{fg}$ and $\mathbf{P}_\mathrm{bg}$, and obtain the correlations $\mathbf{C}_\mathrm{fg} \in \mathbb{R}^{N_Q \times N_O}$ and $\mathbf{C}_\mathrm{bg} \in \mathbb{R}^{N_Q \times N_O}$, given by:

$$\mathbf{C}_\mathrm{fg} = \frac{\mathbf{F}_\mathrm{q} \cdot \mathbf{P}_\mathrm{fg}^\intercal}{\|\mathbf{F}_\mathrm{q}\| \left\|\mathbf{P}_\mathrm{fg}^\intercal\right\|},\ \mathbf{C}_\mathrm{bg} = \frac{\mathbf{F}_\mathrm{q} \cdot \mathbf{P}_\mathrm{bg}^\intercal}{\|\mathbf{F}_\mathrm{q}\| \left\|\mathbf{P}_\mathrm{bg}^\intercal\right\|}. \tag{2}$$

Finally, the correlations $\mathbf{C}_\mathrm{fg}$ and $\mathbf{C}_\mathrm{bg}$ are both expanded to the size of $\mathbb{R}^{N_Q \times 1 \times N_O}$. We concatenate them along the second dimension and project the last dimension back to D using an MLP $\mathcal{F}_\mathrm{mlp}$, as follows:

$$\mathbf{C}_\mathrm{q}^0 = \mathcal{F}_\mathrm{mlp}(\mathbf{C}_\mathrm{fg} \oplus \mathbf{C}_\mathrm{bg}) \in \mathbb{R}^{N_Q \times N_C \times D}, \tag{3}$$

where $\oplus$ is the concatenation operation. Eq. (3) yields the initial CMC. Notably, the second dimension N_C of $\mathbf{C}_\mathrm{q}^0$ is the number of classes, which is 2 under this 1-way example.

The initial correlations $\mathbf{C}_\mathrm{q}^0$ comprise the correlations of each query point with a number of prototypes for all classes, which allows the subsequent modules to directly shape the relations between the query and support. Such *correlation optimization* leads to enhanced generalization for FS-PCS compared to the traditional *feature optimization* [11, 25, 29, 45, 51, 58], as demonstrated in Sec. 5.3.

4.2. Hyper Correlation Augmentation

Our proposed CMC denotes the correlations of each query point to all category prototypes. To enhance the correlations, we introduce the Hyper Correlation Augmentation (HCA) module, leveraging two underlying relationships. First, the query points are all related and dependent on each other. Their correlations to all prototypes are also connected, leading to point-point relations. Second, classifying a single point into foreground or background depends on its *relative* correlations to foreground or background prototypes, forming foreground-background relations. For an N-way setting, this extends to foregrounds-background relations, considering the *relative* correlations among all classes. The proposed HCA refines correlations by exploiting both point-point and foreground-background relations.

Linear Attention. Due to the irregular nature of 3D point clouds, the attention mechanism with the permutation-invariant property is well-suited for point cloud processing. Here, we adopt linear attention [15] for its global receptive field and superior linear computation efficiency.

Given an input sequence $\mathbf{C} \in \mathbb{R}^{N \times D}$, applying linear transformations to $\mathbf{C}$ results in $\mathbf{Q}, \mathbf{K}, \mathbf{V} \in \mathbb{R}^{N \times D}$. Using $\mathbf{q}_i$, $\mathbf{k}_i$, and $\mathbf{v}_i \in \mathbb{R}^{1 \times D}$ to denote the i-th token vector from $\mathbf{Q}$, $\mathbf{K}$, and $\mathbf{V}$, respectively, the standard attention [42] is:

$$\hat{\mathbf{v}}_i = \frac{\sum_{j=1}^{N} \langle \mathbf{q}_i, \mathbf{k}_j \rangle \mathbf{v}_j}{\sum_{j=1}^{N} \langle \mathbf{q}_i, \mathbf{k}_j \rangle},\ \langle \mathbf{q}_i, \mathbf{k}_j \rangle = \exp(\frac{\mathbf{q}_i \mathbf{k}_j^\intercal}{\sqrt{D}}), \tag{4}$$

where $\langle \cdot, \cdot \rangle$ represents the similarity measure function.

Through the lens of kernels, linear attention defines $\langle \mathbf{q}_i, \mathbf{k}_j \rangle = \varphi(\mathbf{q}_i)\varphi(\mathbf{k}_j)^\intercal$ in Eq. (4) and utilizes the associative property of matrix multiplication to obtain:

$$\hat{\mathbf{v}}_i = \frac{\varphi(\mathbf{q}_i) \sum_{j=1}^{N} \varphi(\mathbf{k}_j)^\intercal \mathbf{v}_j}{\varphi(\mathbf{q}_i) \sum_{j=1}^{N} \varphi(\mathbf{k}_j)^\intercal}, \tag{5}$$

where $\varphi(x) = \mathrm{elu}(x) + 1$ with $\mathrm{elu}(\cdot)$ as the exponential linear unit [6]. Consequently, the computation cost of linear attention is $\mathcal{O}(ND^2)$, significantly more favorable than the standard $\mathcal{O}(N^2 D)$ complexity.

Hyper Correlation Augmentation. Based on linear attention, we introduce the HCA module to enhance the correlations through active interactions across points and classes. Since we stack the HCA module L times as in Fig. 3, the module input is denoted as $\mathbf{C}_\mathrm{q}^l$. For each point, we first attend its correlations with those of all other points. We permute $\mathbf{C}_\mathrm{q}^l$ with the class dimension as its first dimension and then compute linear attention across points:

$$\mathbf{C}_\mathrm{q}^{l+1} = \mathcal{F}_\mathrm{lnatt}(\mathcal{T}(\mathbf{C}_\mathrm{q}^l)) \in \mathbb{R}^{N_C \times N_Q \times D}, \tag{6}$$

where $\mathcal{T}$ transposes the first two dimensions, and $\mathcal{F}_\mathrm{lnatt}$ represents the linear attention layer to process features independently along the first dimension. Following the attention layer, an MLP is applied to each point separately and identically to further enhance the correlations:

$$\mathbf{C}_\mathrm{q}^{l+2} = \mathcal{F}_\mathrm{mlp}(\mathbf{C}_\mathrm{q}^{l+1}) \in \mathbb{R}^{N_C \times N_Q \times D}. \tag{7}$$

Note that multi-head attention [42], layer normalization [3], and residual connections are omitted here for simplicity.

After that, we leverage the foreground-background relations to facilitate learning categorical relationships and determining the best-fit class for each point's semantics. We rearrange the dimensions such that $\mathbf{C}_\mathrm{q}^{l+2} \in \mathbb{R}^{N_C \times N_Q \times D} \rightarrow \mathbb{R}^{N_Q \times N_C \times D}$, and apply linear attention, given by:

$$\mathbf{C}_\mathrm{q}^{l+3} = \mathcal{F}_\mathrm{lnatt}(\mathcal{T}(\mathbf{C}_\mathrm{q}^{l+2})) \in \mathbb{R}^{N_Q \times N_C \times D}. \tag{8}$$

The next MLP transforms $\mathbf{C}_{\mathrm{q}}^{l+3}$ to $\mathbf{C}_{\mathrm{q}}^{l+4}$ as in Eq. (7).

Through this module, CMC can interact not only across the spatial dimension but also across the categorical space. This results in comprehensive contextual dependencies, significantly enhancing meta-learning performance.

4.3. Base Prototypes Calibration

Since the training concentrates on classes in C_{train}, models are inherently biased towards these base classes, hindering the segmentation of novel classes [19]. To address it, we propose employing non-parametric prototypes for base classes through the BPC module to alleviate the base bias.

Let $\{\mathbf{p}_{\mathrm{b}}|\mathbf{p}_{\mathrm{b}} \in \mathbb{R}^{1\times D}\}_{\mathrm{b}=1}^{N_b}$ be a set of non-parametric prototypes corresponding to $N_b = |C_{\mathrm{train}}|$ base classes. During meta-training, these prototypes are zero-initialized and evolve continuously. Specifically, given $\mathbf{F}_{\mathrm{s}}$, $\mathbf{F}_{\mathrm{q}}$, and their binary annotations $\mathbf{Y}_{\mathrm{s}}^{\mathrm{b}}$, $\mathbf{Y}_{\mathrm{q}}^{\mathrm{b}}$ of the b-th base class, we calculate the Masked Average Pooling (MAP) [53] for each base class present in the current point clouds:

$$\mathbf{p}_{\mathrm{b}}' = \mathcal{F}_{\mathrm{pool}}(\mathbf{F}_{\mathrm{s/q}} \odot \mathbf{Y}_{\mathrm{s/q}}^{\mathrm{b}}) \in \mathbb{R}^{1\times D}, \tag{9}$$

where $\mathcal{F}_{\mathrm{pool}}$ represents the MAP operation. Then, the base prototypes can be updated at each training episode as:

$$\mathbf{p}_{\mathrm{b}} \leftarrow \mu\mathbf{p}_{\mathrm{b}} + (1-\mu)\mathbf{p}_{\mathrm{b}}', \tag{10}$$

where $\mu \in [0, 1]$ is a momentum coefficient.

When segmenting the novel classes, the query point corresponding to the base classes should be considered as background. Therefore, leveraging base prototypes, we introduce the BPC module to calibrate correlations to the background class, mitigating potential interference from base susceptibility. Specifically, we calculate the base correlations $\mathbf{C}_{\mathrm{base}}$ between the query and base prototypes:

$$\mathbf{C}_{\mathrm{base}} = \frac{\mathbf{F}_{\mathrm{q}} \cdot \mathcal{I}(\{\mathbf{p}_{\mathrm{b}}\}_{\mathrm{b}=1}^{N_b})^{\intercal}}{\|\mathbf{F}_{\mathrm{q}}\| \left\|\mathcal{I}(\{\mathbf{p}_{\mathrm{b}}\}_{\mathrm{b}=1}^{N_b})^{\intercal}\right\|} \in \mathbb{R}^{N_Q\times N_b}, \tag{11}$$

where $\mathcal{I}$ concatenates all the vectors in the set, such that $\mathcal{I}(\{\mathbf{p}_{\mathrm{b}}\}_{\mathrm{b}=1}^{N_b}) \in \mathbb{R}^{N_b\times D}$. Afterward, we obtain the base guidance for each query point $\mathbf{C}_{\mathrm{guide}} = \mathcal{F}_{\max}(\mathbf{C}_{\mathrm{base}}) \in \mathbb{R}^{N_Q}$, where $\mathcal{F}_{\max}$ is max pooling on each row in $\mathbf{C}_{\mathrm{base}}$. Then, the background correlations are calibrated by $\mathbf{C}_{\mathrm{guide}}$ before interacting with foreground correlations, as in Fig. 3:

$$\mathbf{C}_{\mathrm{q}}^{l+2}[1,\cdot,\cdot] = \mathcal{F}_{\mathrm{fc}}(\mathbf{C}_{\mathrm{q}}^{l+2}[1,\cdot,\cdot] \oplus \mathcal{D}(\mathbf{C}_{\mathrm{guide}})), \tag{12}$$

where $\mathbf{C}_{\mathrm{q}}^{l+2}[1,\cdot,\cdot] \in \mathbb{R}^{N_Q\times D}$ selects the background correlations (the last at the N_C-dim) from CMC, $\mathcal{D}$ expands $\mathbf{C}_{\mathrm{guide}}$ as $\mathbb{R}^{N_Q} \rightarrow \mathbb{R}^{N_Q\times D}$, and $\mathcal{F}_{\mathrm{fc}}$ is a fully connected layer. During meta-training, we exclude the base prototypes of the current target classes in Eq. (11). For evaluation, the base prototypes are frozen and utilized without exclusion.

Finally, $\mathbf{C}_{\mathrm{q}}^{4L}$ is decoded to the final segmentation result $\hat{\mathbf{Y}}_{\mathrm{q}}$ using the decoder. Another MLP is employed to generate base class predictions using query features $\mathbf{F}_{\mathrm{q}}$. The entire model is optimized using cross-entropy (CE) loss:

$$\mathcal{L} = \mathrm{CE}(\mathcal{F}_{\mathrm{mlp}}(\mathbf{F}_{\mathrm{q}}), \{\mathbf{Y}_{\mathrm{q}}^{\mathrm{b}}\}_{\mathrm{b}}) + \mathrm{CE}(\hat{\mathbf{Y}}_{\mathrm{q}}, \mathbf{Y}_{\mathrm{q}}). \tag{13}$$

5. Experiments

5.1. Experimental Setting

Network Architecture. We use the first three blocks from the Stratified Transformer [18] as our backbone. The last two blocks produce features with resolutions 1/4 and 1/16 of the original point cloud. We perform interpolation [33] to 4× upsample the 1/16 feature map and concat it to the 1/4 features, followed by an MLP to obtain final features with a channel dimension of 192. For the S3DIS dataset, we employ a 2-layer HCA module. Due to the richer semantics of ScanNet [7], we use 4 layers of HCA. The final decoder consists of one KPConv [40] layer followed by an MLP.

Implementation Detail. We employ the data processing strategy from [18, 56]. The 3D scene is divided into 1m × 1m blocks to increase data samples, and raw input points are grid-sampled with a grid size of 0.02m. After voxelization, if the input point count exceeds 20,480, we randomly sample 20,480 points to control the input size. Data augmentation and pre-training follow [18] where our backbone is pre-trained on each fold for 100 epochs. Meta-training involves 40,000 episodes, using the AdamW optimizer with a learning rate of 0.00005 and weight decay of 0.01. During testing, we sample 1,000 episodes per class in the 1-way setting and 100 episodes for each combination in the 2-way setting for more stable evaluations. We use 100 prototypes for each class ($N_O = 100$). In the k-shot setting, when $k > 1$, we sample N_O/k prototypes from each shot and concatenate them to obtain N_O prototypes. For benchmarking previous models, we select methods with publicly available code, namely AttMPTI [56], QGE [29], and QGPA [11].

5.2. Main Results

Effects of Foreground Leakage. As discussed in Sec. 3.2, the foreground leakage problem can significantly distort model training and evaluation. Tab. 1 compares the performance of previous methods [11, 29, 56] in two settings: one with foreground leakage (*w/ FG*) and the other without foreground leakage (*w/o FG*) across two datasets. In each setting, we retrain the models and evaluate them on the corresponding test set. The results reveal a substantial mIoU drop after correcting foreground leakage consistently across all splits for 1-way 1/5-shot tasks. On S3DIS, the highest mIoU of 81.80% (*w/ FG*) for the 5-shot task drops dramatically to 45.52% after removing foreground leakage, marking a significant 36.28% drop. Similarly, on ScanNet,

	Methods	1-shot (S3DIS)			5-shot (S3DIS)			1-shot (ScanNet)			5-shot (ScanNet)		
		S^0	S^1	mean	S^0	S^1	mean	S^0	S^1	mean	S^0	S^1	mean
w/ FG	AttMPTI [56]	64.89	66.15	65.52	76.56	83.08	79.82	62.14	58.65	60.39	68.79	68.66	68.73
	QGE [29]	74.05	73.61	73.83	74.65	83.21	78.93	63.50	57.61	60.56	70.72	65.68	68.20
	QGPA [11]	62.72	61.95	62.33	76.30	87.29	81.80	56.47	51.72	54.10	81.57	72.75	77.16
w/o FG	AttMPTI [56]	41.56	41.27	41.41	50.55	46.13	48.34	33.36	31.81	32.58	37.95	36.30	37.12
	QGE [29]	46.27	47.76	47.02	47.74	59.77	53.76	37.72	34.64	36.18	48.73	39.95	44.34
	QGPA [11]	35.62	41.13	38.38	43.54	47.50	45.52	40.03	35.54	37.78	46.17	42.24	44.20

Table 1. **Comparisons in the mIoU metric between *with* foreground leakage (*w/ FG*) and *without* foreground leakage (*w/o FG*) for existing methods.** The results are for 1-way segmentation setting. S^0/S^1 refers to the i-th split for inference. Here, we adopt the previous FS-PCS setting [56], *i.e.*, using DGCNN [47] as the backbone and sampling 2,048 points for each scene.

	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
S3DIS [1]	AttMPTI [56]	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 [29]	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 [11]	35.50	35.83	35.67	38.07	39.70	38.89	25.52	26.26	25.89	30.22	32.41	31.32
	COSeg (ours)	**46.31**	**48.10**	**47.21**	**51.40**	**48.68**	**50.04**	**37.44**	**36.45**	**36.95**	**42.27**	**38.45**	**40.36**
ScanNet [7]	AttMPTI [56]	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 [29]	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 [11]	34.57	33.37	33.97	41.22	38.65	39.94	21.86	21.47	21.67	30.67	27.69	29.18
	COSeg (ours)	**41.73**	**41.82**	**41.78**	**48.31**	**44.11**	**46.21**	**28.72**	**28.83**	**28.78**	**35.97**	**33.39**	**34.68**

Table 2. **Comparisons in the mIoU metric between our method and baselines in the new FS-PCS setting.** The best-performing results are highlighted in **bold**. Previous methods apply the same backbone as ours for fair comparisons.

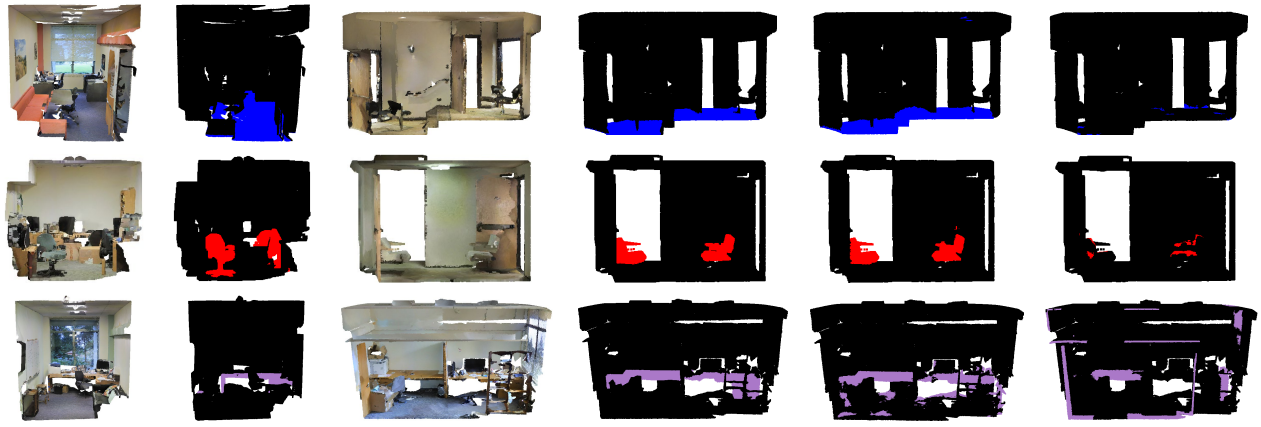


Figure 4. Qualitative comparisons between our proposed model COSeg and QGE [29]. Each row, from top to bottom, represents the 1-way 1-shot task with the target category as floor (blue), chair (red), and table (purple), respectively.

a substantial 32.96% mIoU drop is observed from 77.16% to 44.20% for the 5-shot task. On average, across the three methods, the mIoU drop from removing foreground leakage is 27.97% on S3DIS and 26.16% on ScanNet. This notable performance gap underscores that previous methods largely rely on the density differences exposed by foreground leakage to achieve seemingly superior performance. This underscores the immediate need for our new corrected setting for facilitating the research in FS-PCS.

Comparison with Previous Methods. In Tab. 2, we present the results for $1/2$-way $1/5$-shot experiments on two datasets. Our model demonstrates a significant performance advantage, establishing new state-of-the-art records across all experiments. For instance, in the 1-way 1-shot scenario, we achieve notable mIoU improvements of 6.82% and 6.58% over the second-best model, QGE, on S3DIS and ScanNet, respectively. Extending to the 2-way 5-shot task, our model outperforms the previous best performance by 2.03% (S3DIS) and 4.76% (ScanNet) in mIoU. Similar substantial improvements are observed in all other settings. These consistent enhancements underscore the efficacy of our model within our proposed rigorous FS-PCS setting.

Qualitative Results. In Fig. 4, we visualize predictions from our method (5th column) and the previous best method, QGE (6th column). Our method clearly achieves better segmentation results than the previous best method.

Optimization	HCA	BPC	1-shot	5-shot
feature			30.67	32.58
correlation			39.93	42.33
correlation	✓		43.77	47.98
correlation	✓	✓	47.21	50.04

Table 3. Ablation study of different design choices in COSeg.

N_O	1-shot	5-shot
50	43.50	46.43
100	47.21	50.04
150	47.41	52.33
200	48.27	47.90

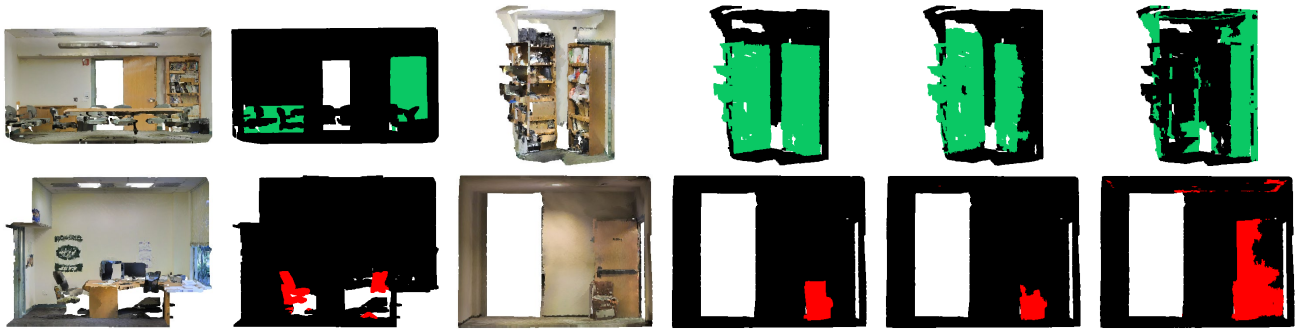
Table 4. Effects of the number of prototypes per class.

L	1-shot	5-shot
1	44.19	45.93
2	47.21	50.04
3	46.12	46.89

Table 5. Impact of the number of HCA layers.

μ	1-shot	5-shot
0.99	46.91	50.26
0.995	47.21	50.04
0.999	47.40	49.85

Table 6. Ablation on the momentum coefficient.



Support | Support Mask | Query | Ground Truth | w/ BPC | w/o BPC

Figure 5. Visual comparisons between our models with BPC (w/ BPC) and without BPC (w/o BPC). Each row corresponds to the 1-way 1-shot task targeting bookcase (green) and chair (red), respectively, arranged from top to bottom.

5.3. Ablation Study

In this section, we report the mIoU results as **the mean of all splits** of S3DIS under the 1-way $1/5$-shot settings.

Different Design Choices. We first assess our proposed correlation optimization paradigm. The baseline model consists of only the backbone and decoder from our approach. We compare the performance of forwarding correlations (correlation optimization) and forwarding features (feature optimization) to the decoder after the backbone. Feature optimization uses the support prototype to directly segment the target object as in [41]. As shown in Tab. 3, transitioning solely from forwarding features to forwarding correlations results in a significant 9.26% and 9.75% increase in mIoU under $1/5$-shot settings, respectively. These results affirm the superiority of our proposed correlation optimization paradigm in enhancing generalization for FS-PCS compared to the traditional feature optimization.

Furthermore, we explore the impact of HCA and BPC in Tab. 3. Adding HCA to the baseline with correlation optimization leads to a 3.84%/5.65% mIoU improvement for the 1-shot/5-shot setting, demonstrating the efficacy of HCA in enriching contextual information for correlations. Incorporating BPC with HCA results in an additional 3.44%/2.06% growth in mIoU for the 1-shot/5-shot setting, highlighting the significance of BPC in calibrating background correlations. Fig. 5 contrasts visual segmentation results between our models with BPC and without BPC. The absence of BPC exhibits base susceptibility issues, with false activations of base classes (*wall* or *door*) in the scenes. Conversely, the inclusion of our BPC design enables models to effectively mitigate susceptibility, ensuring accurate segmentation of novel classes.

Number of Prototypes. Tab. 4 shows increasing the number of prototypes to 150 improves performance. For a fair comparison with others, we set $N_O = 100$ by default.

Number of HCA Layers. We vary the number of HCA layers from 1 to 3 and report the results in Tab. 5. It shows that using two layers achieves the best performance on S3DIS.

Momentum Coefficient. The momentum coefficient μ controls the evolving rate of our base prototypes. We explore its effects on performance in Tab. 6. The results show that varying μ causes minimal differences in mIoU, demonstrating the robustness of our proposed BPC module.

6. Conclusion

In this paper, we identify two critical issues in FS-PCS: foreground leakage and sparse point distribution, which have undermined the validity of previous progress and hindered further advancements. To rectify these issues, we standardize FS-PCS by introducing a rigorous setting along with a new benchmark. Moreover, we propose a novel correlation optimization paradigm that operates on CMC, diverging from the traditional feature optimization approach used by all previous FS-PCS models. Building on this paradigm, our model COSeg incorporates HCA for effective contextual learning and BPC for background correlation adjustment, achieving state-of-the-art results across all FS-PCS settings. We hope that our work could serve as the foundation for FS-PCS and shed light on future research.

Acknowledgments. This work is supported in part by the Agency for Science, Technology and Research (A*STAR) under its MTC Young Individual Research Grant (Grant No. M21K3c0130), the Alexander von Humboldt Foundation, and the Pioneer Centre for AI, DNRF grant number P1.

References

[1] Iro Armeni, Ozan Sener, Amir R Zamir, Helen Jiang, Ioannis Brilakis, Martin Fischer, and Silvio Savarese. 3d semantic parsing of large-scale indoor spaces. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1534–1543, 2016. 1, 2, 3, 7, 11, 12

[2] Matan Atzmon, Haggai Maron, and Yaron Lipman. Point convolutional neural networks by extension operators. *arXiv preprint arXiv:1803.10091*, 2018. 2

[3] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. Layer normalization. *arXiv preprint arXiv:1607.06450*, 2016. 5

[4] Jens Behley, Martin Garbade, Andres Milioto, Jan Quenzel, Sven Behnke, Cyrill Stachniss, and Jurgen Gall. Semantickitti: A dataset for semantic scene understanding of lidar sequences. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 9297–9307, 2019. 1

[5] Xiaozhi Chen, Huimin Ma, Ji Wan, Bo Li, and Tian Xia. Multi-view 3d object detection network for autonomous driving. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 1907–1915, 2017. 1

[6] Djork-Arné Clevert, Thomas Unterthiner, and Sepp Hochreiter. Fast and accurate deep network learning by exponential linear units (elus). *arXiv preprint arXiv:1511.07289*, 2015. 5

[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 conference on computer vision and pattern recognition*, pages 5828–5839, 2017. 1, 2, 6, 7, 11

[8] Martin Engelcke, Dushyant Rao, Dominic Zeng Wang, Chi Hay Tong, and Ingmar Posner. Vote3deep: Fast object detection in 3d point clouds using efficient convolutional neural networks. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 1355–1361. IEEE, 2017. 1

[9] Francis Engelmann, Theodora Kontogianni, Alexander Hermans, and Bastian Leibe. Exploring spatial context for 3d semantic segmentation of point clouds. In *Proceedings of the IEEE international conference on computer vision workshops*, pages 716–724, 2017. 2

[10] Francis Engelmann, Theodora Kontogianni, Jonas Schult, and Bastian Leibe. Know what your neighbors do: 3d semantic segmentation of point clouds. In *Proceedings of the European Conference on Computer Vision (ECCV) Workshops*, pages 0–0, 2018. 2

[11] 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*, 2023. 1, 2, 3, 4, 5, 6, 7, 11

[12] 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*, pages 11108–11117, 2020. 2

[13] Binh-Son Hua, Minh-Khoi Tran, and Sai-Kit Yeung. Pointwise convolutional neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 984–993, 2018. 2

[14] Mingyang Jiang, Yiran Wu, Tianqi Zhao, Zelin Zhao, and Cewu Lu. Pointsift: A sift-like network module for 3d point cloud semantic segmentation. *arXiv preprint arXiv:1807.00652*, 2018. 2

[15] Angelos Katharopoulos, Apoorv Vyas, Nikolaos Pappas, and François Fleuret. Transformers are rnns: Fast autoregressive transformers with linear attention. In *International conference on machine learning*, pages 5156–5165. PMLR, 2020. 5

[16] Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*, 2016. 2

[17] Artem Komarichev, Zichun Zhong, and Jing Hua. A-cnn: Annularly convolutional neural networks on point clouds. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 7421–7430, 2019. 2

[18] Xin Lai, Jianhui Liu, Li Jiang, Liwei Wang, Hengshuang Zhao, Shu Liu, Xiaojuan Qi, and Jiaya Jia. Stratified transformer for 3d point cloud segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8500–8509, 2022. 2, 3, 4, 6, 11

[19] Chunbo Lang, Gong Cheng, Binfei Tu, and Junwei Han. Learning what not to segment: A new perspective on few-shot segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8057–8067, 2022. 2, 6

[20] Huan Lei, Naveed Akhtar, and Ajmal Mian. Spherical kernel for efficient graph convolution on 3d point clouds. *IEEE transactions on pattern analysis and machine intelligence*, 43(10):3664–3680, 2020. 2

[21] Yangyan Li, Rui Bu, Mingchao Sun, Wei Wu, Xinhan Di, and Baoquan Chen. Pointcnn: Convolution on x-transformed points. *Advances in neural information processing systems*, 31, 2018. 2

[22] 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*, pages 1800–1809, 2020. 2

[23] 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*, pages 8895–8904, 2019. 2

[24] Zhijian Liu, Haotian Tang, Yujun Lin, and Song Han. Point-voxel cnn for efficient 3d deep learning. *Advances in Neural Information Processing Systems*, 32, 2019. 2

[25] 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 *2022 International Conference on 3D Vision (3DV)*, pages 505–514. IEEE, 2022. 1, 2, 3, 4, 5, 11

[26] Andres Milioto, Ignacio Vizzo, Jens Behley, and Cyrill Stachniss. Rangenet++: Fast and accurate lidar semantic

segmentation. In *2019 IEEE/RSJ international conference on intelligent robots and systems (IROS)*, pages 4213–4220. IEEE, 2019. 1

[27] Federico Monti, Davide Boscaini, Jonathan Masci, Emanuele Rodola, Jan Svoboda, and Michael M Bronstein. Geometric deep learning on graphs and manifolds using mixture model cnns. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5115–5124, 2017. 2

[28] Dong Nie, Rui Lan, Ling Wang, and Xiaofeng Ren. Pyramid architecture for multi-scale processing in point cloud segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 17284–17294, 2022. 3

[29] Zhenhua Ning, Zhuotao Tian, Guangming Lu, and Wenjie Pei. Boosting few-shot 3d point cloud segmentation via query-guided enhancement. *arXiv preprint arXiv:2308.03177*, 2023. 1, 2, 3, 4, 5, 6, 7, 11, 13

[30] Chunghyun Park, Yoonwoo Jeong, Minsu Cho, and Jaesik Park. Fast point transformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16949–16958, 2022. 3

[31] Roberto Pierdicca, Marina Paolanti, Francesca Matrone, Massimo Martini, Christian Morbidoni, Eva Savina Malinverni, Emanuele Frontoni, and Andrea Maria Lingua. Point cloud semantic segmentation using a deep learning framework for cultural heritage. *Remote Sensing*, 12(6):1005, 2020. 1

[32] Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 652–660, 2017. 2

[33] Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. *Advances in neural information processing systems*, 30, 2017. 2, 6

[34] Xiaojuan Qi, Renjie Liao, Jiaya Jia, Sanja Fidler, and Raquel Urtasun. 3d graph neural networks for rgbd semantic segmentation. In *Proceedings of the IEEE international conference on computer vision*, pages 5199–5208, 2017. 2

[35] Haoxi Ran, Wei Zhuo, Jun Liu, and Li Lu. Learning inner-group relations on point clouds. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 15477–15487, 2021. 3

[36] Mengye Ren, Eleni Triantafillou, Sachin Ravi, Jake Snell, Kevin Swersky, Joshua B Tenenbaum, Hugo Larochelle, and Richard S Zemel. Meta-learning for semi-supervised few-shot classification. *arXiv preprint arXiv:1803.00676*, 2018. 2

[37] 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 conference on computer vision and pattern recognition*, pages 4548–4557, 2018. 2

[38] Jake Snell, Kevin Swersky, and Richard Zemel. Prototypical networks for few-shot learning. *Advances in neural information processing systems*, 30, 2017. 2

[39] Maxim Tatarchenko, Jaesik Park, Vladlen Koltun, and Qian-Yi Zhou. Tangent convolutions for dense prediction in 3d. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3887–3896, 2018. 2

[40] 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*, pages 6411–6420, 2019. 2, 6

[41] 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. 8

[42] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017. 3, 5

[43] Nitika Verma, Edmond Boyer, and Jakob Verbeek. Feastnet: Feature-steered graph convolutions for 3d shape analysis. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2598–2606, 2018. 2

[44] Oriol Vinyals, Charles Blundell, Timothy Lillicrap, Daan Wierstra, et al. Matching networks for one shot learning. *Advances in neural information processing systems*, 29, 2016. 2, 3

[45] Jiahui Wang, Haiyue Zhu, Haoren Guo, Abdullah Al Mamun, Cheng Xiang, and Tong Heng Lee. Few-shot point cloud semantic segmentation via contrastive self-supervision and multi-resolution attention. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 2811–2817. IEEE, 2023. 1, 2, 3, 4, 5, 11

[46] Lei Wang, Yuchun Huang, Yaolin Hou, Shenman Zhang, and Jie Shan. Graph attention convolution for point cloud semantic segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10296–10305, 2019. 2

[47] Yue Wang, Yongbin Sun, Ziwei Liu, Sanjay E Sarma, Michael M Bronstein, and Justin M Solomon. Dynamic graph cnn for learning on point clouds. *ACM Transactions on Graphics (tog)*, 38(5):1–12, 2019. 2, 7

[48] 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 (ECCV)*, pages 87–102, 2018. 2

[49] Xiaoqing Ye, Jiamao Li, Hexiao Huang, Liang Du, and Xiaolin Zhang. 3d recurrent neural networks with context fusion for point cloud semantic segmentation. In *Proceedings of the European conference on computer vision (ECCV)*, pages 403–417, 2018. 2

[50] Cheng Zhang, Haocheng Wan, Xinyi Shen, and Zizhao Wu. Patchformer: An efficient point transformer with patch attention. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11799–11808, 2022. 3

[51] Canyu Zhang, Zhenyao Wu, Xinyi Wu, Ziyu Zhao, and Song Wang. Few-shot 3d point cloud semantic segmentation via

stratified class-specific attention based transformer network. In *AAAI*, 2023. 2, 3, 4, 5, 11

[52] Nan Zhang, Zhiyi Pan, Thomas H Li, Wei Gao, and Ge Li. Improving graph representation for point cloud segmentation via attentive filtering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1244–1254, 2023. 2

[53] 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. 6

[54] Zhiyuan Zhang, Binh-Son Hua, and Sai-Kit Yeung. Shellnet: Efficient point cloud convolutional neural networks using concentric shells statistics. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 1607–1616, 2019. 2

[55] Hengshuang Zhao, Li Jiang, Jiaya Jia, Philip HS Torr, and Vladlen Koltun. Point transformer. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 16259–16268, 2021. 2, 3, 4

[56] Na Zhao, Tat-Seng Chua, and Gim Hee Lee. Few-shot 3d point cloud semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8873–8882, 2021. 1, 2, 3, 4, 5, 6, 7, 11

[57] Haoran Zhou, Yidan Feng, Mingsheng Fang, Mingqiang Wei, Jing Qin, and Tong Lu. Adaptive graph convolution for point cloud analysis. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 4965–4974, 2021. 2

[58] Guanyu Zhu, Yong Zhou, Rui Yao, and Hancheng Zhu. Cross-class bias rectification for point cloud few-shot segmentation. *IEEE Transactions on Multimedia*, 2023. 1, 2, 3, 4, 5, 11

Appendix

A. More Details about Foreground Leakage

As discussed in Sec. 3.2, the current few-shot 3D point cloud semantic segmentation (FS-PCS) setting [11, 25, 29, 45, 51, 56, 58] employs a non-uniform sampling mechanism with a bias toward foreground classes. This biased sampling algorithm samples more points from foreground objects than from the background, resulting in a noticeable point density disparity between foreground and background.

More precisely, the biased sampling algorithm can be outlined in Alg. 1[1]. In line 1, it firstly obtains the input foreground point set $\mathbf{P}_{\mathrm{FG}}$ that includes all the input points belonging to the foreground class C with respect to the current few-shot task. Then, from lines 2 to 6, it calculates the quantity N_{FG} that will be used for sampling foreground points in the output. N_{FG} maintains a proportional relationship to the presence of foreground points in the input data when $n \geq m$. Next, in line 7, it selects N_{FG} points exclusively from the input foreground point set $\mathbf{P}_{\mathrm{FG}}$. However, in line 8, the remaining $m - N_{\mathrm{FG}}$ points are sampled from the entire input points $\mathbf{X} = \{\mathbf{P}_1, .., \mathbf{P}_{\mathrm{n}}\}$, which still includes the foreground points in $\mathbf{P}_{\mathrm{FG}}$. Consequently, this double-sampling of foreground points in these two steps leads to foreground objects having a denser distribution of points in the final output than their background counterparts.

[1]The corresponding source code can be found at the link.

Algorithm 1: The biased sampling algorithm

Data: input point cloud $\mathbf{X}$ with n points $\{\mathbf{P}_1, .., \mathbf{P}_{\mathrm{n}}\}$, sampling number m, foreground class C with respect to current few-shot task

Result: sampled points $\{\mathbf{P}_{\mathrm{i}_1}, .., \mathbf{P}_{\mathrm{i}_m}\}$ from $\mathbf{X}$

1 $\mathbf{P}_{\mathrm{FG}} \leftarrow \{\mathbf{P}_{\mathrm{i}} \mid label\_of(\mathbf{P}_{\mathrm{i}}) = C\}$;
2 **if** $n < m$ **then**
3 $\quad N_{\mathrm{FG}} \leftarrow |\mathbf{P}_{\mathrm{FG}}|$;
4 **else**
5 $\quad N_{\mathrm{FG}} \leftarrow m\frac{|\mathbf{P}_{\mathrm{FG}}|}{n}$;
6 **end**
7 $\mathbf{Res}_1 \leftarrow$ sample N_{FG} points from $\mathbf{P}_{\mathrm{FG}}$;
8 $\mathbf{Res}_2 \leftarrow$ sample $m - N_{\mathrm{FG}}$ points from $\mathbf{X}$;
9 $\{\mathbf{P}_{\mathrm{i}_1}, .., \mathbf{P}_{\mathrm{i}_m}\} \leftarrow \mathbf{Res}_1 \cup \mathbf{Res}_2$;

We also present additional visualizations in Fig. 6. Both the theoretical analysis and visualizations clearly demonstrate that this biased sampling leaks foreground class information to models through density disparity. Consequently, the models no longer need to excel at learning essential knowledge adaptation patterns for few-shot tasks; instead, they can simply segment the target by detecting denser regions. This foreground leakage undermines the validity of existing benchmarks of previous models.

B. More Implementation Details

We employ the first three blocks from the Stratified Transformer [18] as our backbone. Our backbone architecture aligns with the one used for the S3DIS dataset [1] in [18], indicating that we maintain consistency in backbone architectures for both S3DIS and ScanNet [7]. Unlike [18], we do not employ different Stratified Transformer architectures for these two datasets. The momentum coefficient μ within the BPC module is set to 0.995. For both datasets, our input features include both the XYZ coordinates and RGB colors. The training and testing are using 4 RTX 3090 GPUs.

C. More Qualitative Results

We present additional qualitative results in Fig. 7, comparing our method (5th column) with the previous best-performing method, QGE (6th column). Besides, Fig. 8 showcases more visual comparisons between our models

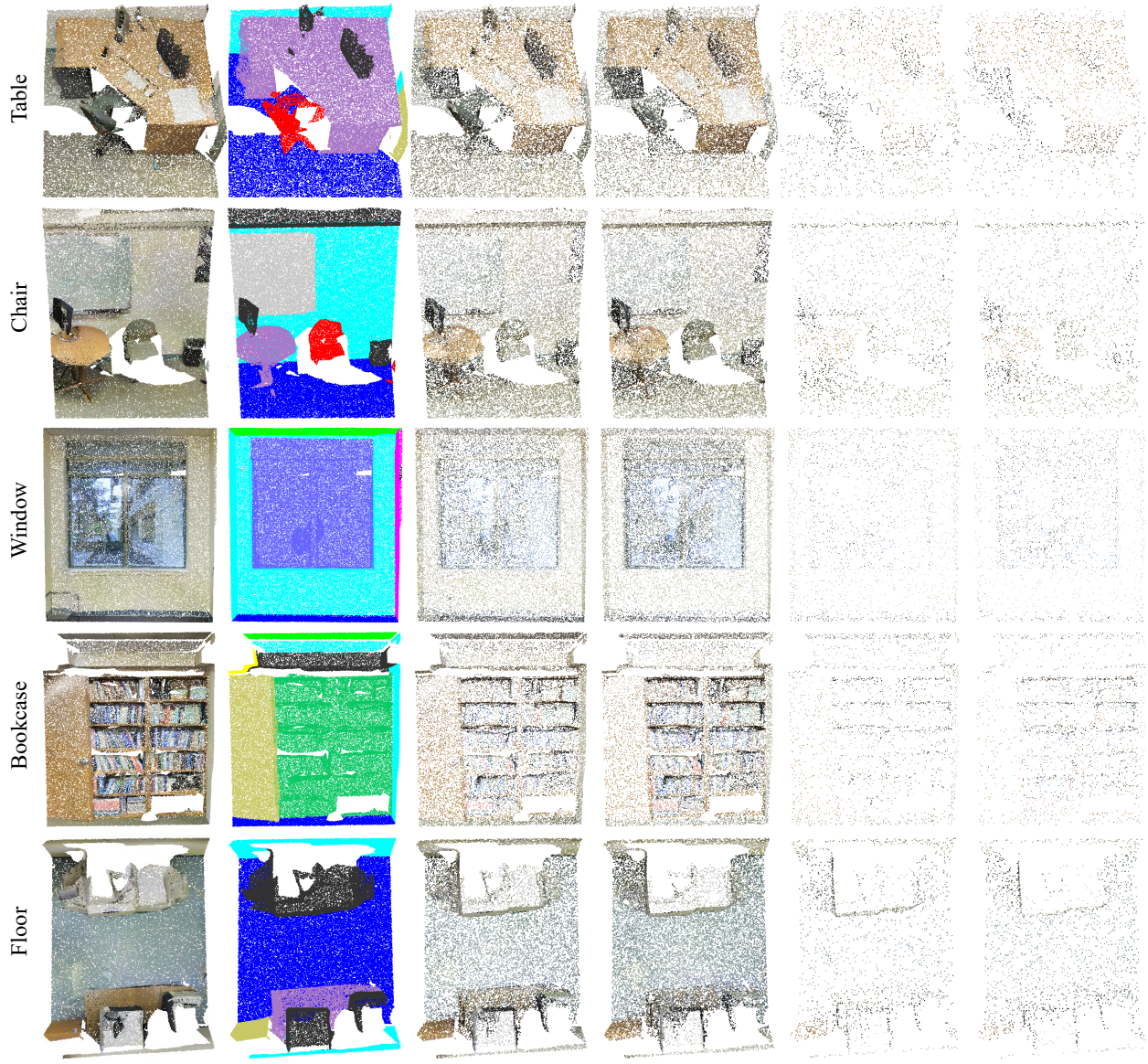


Figure 6. **Visualization of various scenes from the S3DIS dataset [1], with the target class for the 1-way few-shot task labeled at the leftmost of each scene.** Each scene includes six types of point clouds, arranged *from left to right:* (1) The original point cloud; (2) Ground truth of all categories; (3) Our corrected input with 20,480 points in a uniform distribution; (4) Input with 20,480 points in a biased distribution; (5) Input with 2,048 points in a uniform distribution; (6) Input with 2,048 points in a biased distribution, as adopted by previous works.

with BPC (w/ BPC, 5th column) and without BPC (w/o BPC, 6th column).

We have the following observations from the visual comparisons: (1) Our method yields visually better results than the previous best-performing method, highlighting the superiority of our proposed correlation optimization paradigm in enhancing the generalization ability for few-shot tasks. (2) The lightweight BPC module, equipped with non-parametric base prototypes, effectively mitigates the base susceptibility issue inherent in models. This ensures accurate segmentation of novel classes, further validating the efficacy of our approach.

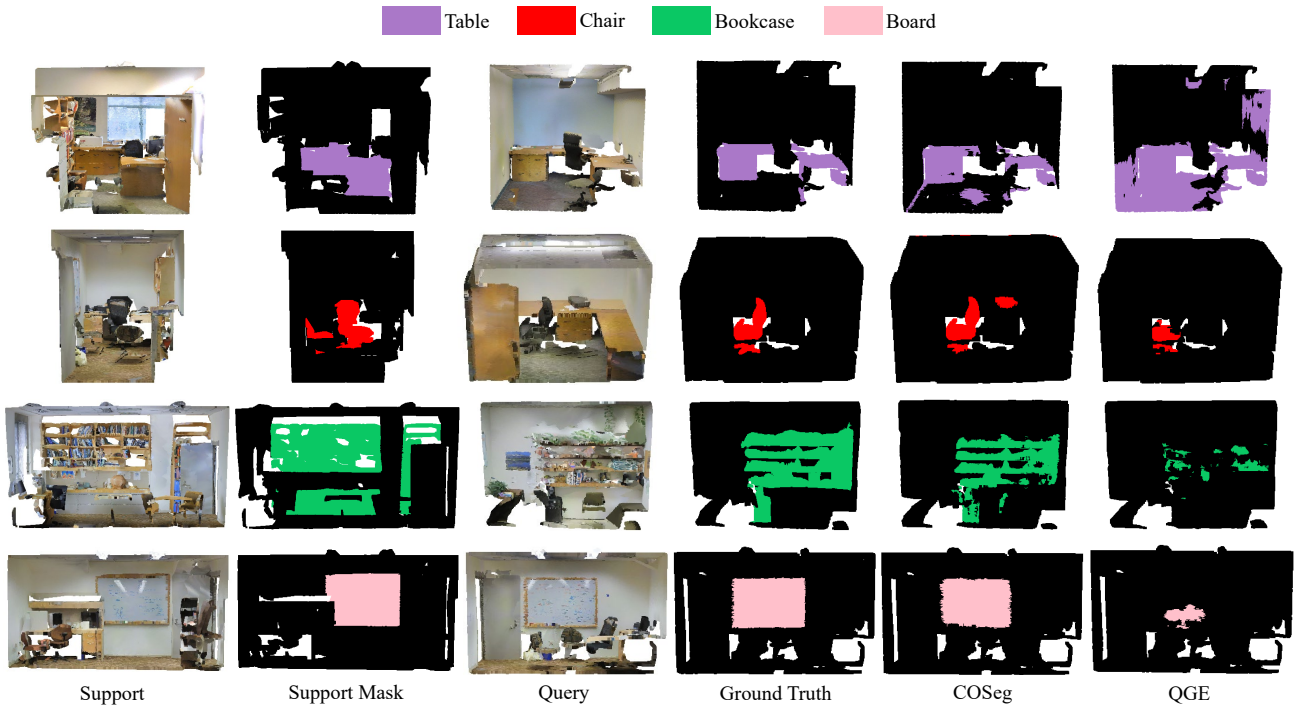


Figure 7. Qualitative comparisons between our proposed model COSeg and QGE [29]. Each row, from top to bottom, represents the 1-way 1-shot task with the target category as table (purple), chair (red), bookcase (green) and board (pink), respectively.

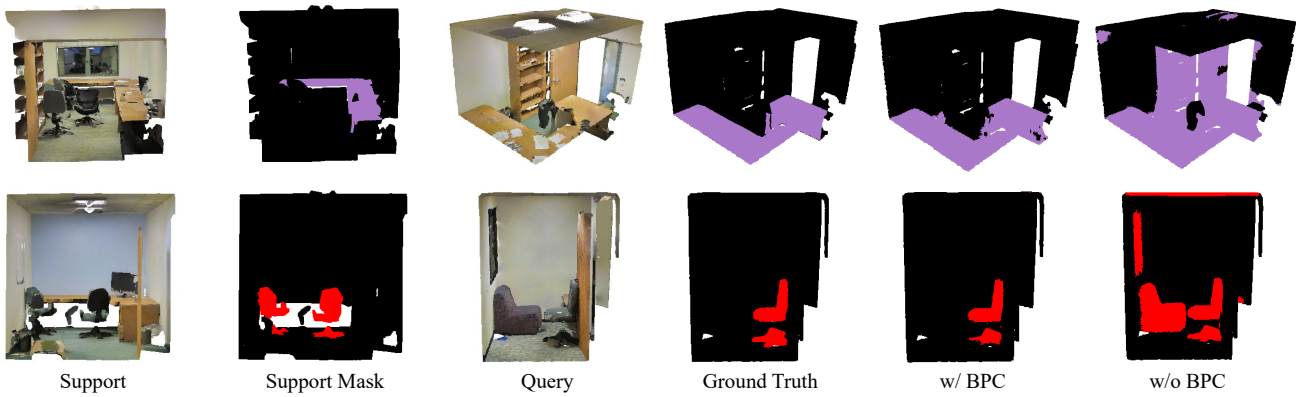


Figure 8. Qualitative comparisons between our models with BPC (w/ BPC) and without BPC (w/o BPC). Each row has the target class under the 1-way 1-shot task as table (purple) and chair (red), respectively, arranged from top to bottom.