Unsupervised Point Cloud Object Co-segmentation by Co-contrastive Learning and Mutual Attention Sampling

Cheng-Kun Yang[1] Yung-Yu Chuang[1] Yen-Yu Lin[2,3]
[1]National Taiwan University [2]National Yang Ming Chiao Tung University [3]Academia Sinica

Abstract

This paper presents a new task, point cloud object co-segmentation, aiming to segment the common 3D objects in a set of point clouds. We formulate this task as an object point sampling problem, and develop two techniques, the mutual attention module and co-contrastive learning, to enable it. The proposed method employs two point samplers based on deep neural networks, the object sampler and the background sampler. The former targets at sampling points of common objects while the latter focuses on the rest. The mutual attention module explores point-wise correlation across point clouds. It is embedded in both samplers and can identify points with strong cross-cloud correlation from the rest. After extracting features for points selected by the two samplers, we optimize the networks by developing the co-contrastive loss, which minimizes feature discrepancy of the estimated object points while maximizing feature separation between the estimated object and background points. Our method works on point clouds of an arbitrary object class. It is end-to-end trainable and does not need point-level annotations. It is evaluated on the ScanObjectNN and S3DIS datasets and achieves promising results. The source code will be available at https://github.com/jimmy15923/unsup_point_coseg.

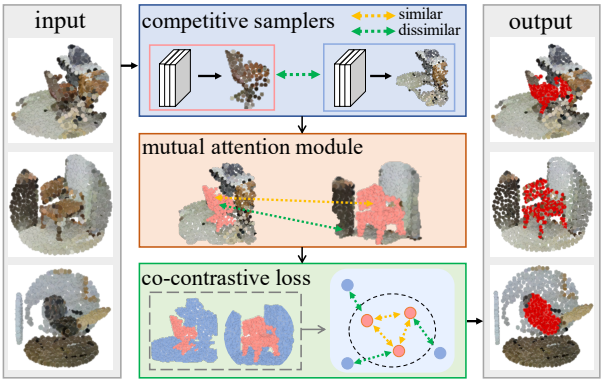


Figure 1: **Overview of our method for unsupervised point cloud co-segmentation**. The *input* to our method is a set of point clouds covering objects of a common category (chairs, in this example). Our method only requires 3D coordinates as the input. Color is added here for visualization. Our method *picks out* the co-segmented points of the common objects (those in red). It formulates co-segmentation as a sampling problem by employing two *competitive samplers*: one for foreground points and the other for the rest. A *mutual attention module* is embedded in each sampler for capturing cross-cloud point correlation. The whole network is end-to-end trained by the proposed *co-contrastive loss*.

1. Introduction

Point clouds retain 3D geometric structures and are adopted in many 3D vision applications, such as remote sensing [24], autonomous driving [9, 14, 30], and robotics [21]. As an essential technique for 3D understanding, point cloud segmentation gains significant progress owing to advanced network architectures [32, 33, 38, 43, 45, 51] and large-scale datasets [3, 5, 6, 12, 14, 26, 30, 34, 40]. Despite effectiveness, deep-learning-based methods for point cloud segmentation rely on lots of training data with point-level annotations. The high annotation cost for training data collection impedes the utility of point cloud segmentation.

2D image object co-segmentation [16,17,20,23,50] aims to segment the common objects in a set of images without additional annotations. It significantly mitigates the problem of high annotation costs in object segmentation. However, it is challenging to apply 2D image co-segmentation techniques to 3D point clouds as it must resolve three major issues. First, most image co-segmentation methods rely on object proposal generators or saliency detectors [16,17,50]. These generators and detectors work on the appearance domain of image pixels, but do not apply to the geometric domain of 3D points. Second, compared to images, point clouds are unordered and unstructured. The extracted point features are typically insufficient for co-segmentation. Third, most 2D co-segmentation methods adopt a network pre-trained on a large dataset, *e.g.* ImageNet [35], to extract high-level semantic features. Although point cloud model pre-training has been studied [7,36,47], training from scratch is still widely used in modern point cloud research.

In this paper, we present an unsupervised method for

point cloud object co-segmentation. As shown in Figure 1, our method comprises three components to tackle the three aforementioned issues, respectively. First, we cast point cloud co-segmentation as an object point sampling problem. A pair of point samplers are employed: The object sampler targets at sampling points belonging to the common objects, while the background sampler grabs on the rest. Through sampling, object proposals or saliency detectors are no longer required. Sampling is non-differentiable. This issue has been resolved by SampleNet [13, 22], which offers differentiable point sampling from a point cloud. In this work, both object and background samplers are developed upon SampleNet. SampleNet is originally designed for supervised applications. To adapt it to unsupervised co-segmentation, we develop novel co-contrastive learning to derive a pair of competitive samplers. After optimization, the two samplers complete co-segmentation.

Second, a mutual attention module is developed to explore point-wise correlation across different point clouds, and is employed by both samplers. Identifying the common 2D pixels or 3D points in the given images or point clouds is a key component for co-segmentation. To this end, this module computes attention maps across clouds and compiles informative features for co-segmentation. Compared with the self-attention module [41, 44] focusing on the correlation of positions within a point cloud, it computes cross-cloud attention to discover plausible foreground. The idea behind this module is that points belonging to common objects typically have strong cross-cloud correspondences. It turns out that samplers equipped with this module result in better foreground-background separation.

Third, a co-contrastive loss is developed to address the lack of data for pre-training and the absence of supervisory signals for co-segmentation. This loss is designed in both object and point levels. It minimizes the feature discrepancy of points sampled by the object sampler while maximizing the feature discrepancy between points selected by different samplers. We use this loss to derive the samplers as well as their associated mutual attention modules.

The main contribution of this work is threefold. First, to the best of our knowledge, our method is the first attempt to develop an end-to-end trainable network for point cloud object co-segmentation. Second, we formulate it as a discriminative sampling task, which is carried out by the proposed mutual attention module and co-contrastive learning. Third, our approach is evaluated on two real datasets [3, 40], and demonstrates promising results.

2. Related Work

Object co-segmentation in 2D images. Methods of this category such as [10, 16, 17, 20, 23, 39, 50] aim to segment the common objects in images without additional supervision and can save the manual annotation cost for object segmentation. Due to the unsupervised nature, many of them leverage *contrastive learning*, namely minimizing inter-image object discrepancy while maximizing intra-image foreground-background separation. For example, Hsu *et al*. [16] propose a co-attention generator to produce co-segment maps with a frozen pre-trained feature extractor. The generator is derived by contrastive learning with additional object proposals. Some methods such as [50] generate the co-occurrence map, which encodes both objectness scores of images and similarity evidence from object proposals across images. To address the unavailability of object saliency detectors and object proposals on point clouds, our method casts co-segmentation on point clouds as a foreground point sampling problem and implements contrastive learning to discriminatively derive the samplers.

Shape co-segmentation. This task parses objects into parts [11, 18, 26, 27, 37, 53]. There are three major differences between *shape* co-segmentation and *object* co-segmentation. First, the tasks differ. Shape co-segmentation decomposes objects into universal parts, while object co-segmentation distinguishes common objects from the background. In the literature, shape co-segmentation is typically applied to clean and complete 3D CAD models [11, 53]. Object co-segmentation need to deal with cluttered background and incomplete objects in real scenes. Second, shape co-segmentation aims to find structural correspondences across shapes of the same category, such as armrests of chairs. Most methods learn proper embedding and assume that structural correspondences bear similarity in the embedding space. Object co-segmentation aims at separating objects from their surroundings. However, object surroundings often pose great variety and do not exhibit similarity in the embedding space. Third, many shape co-segmentation methods rely on online manual annotation [11] or additional datasets [53].

Point cloud sampling. Advanced sensing technologies increase the point densities of point clouds. Thus, sampling crucial points helps reduce computational demands. Farthest point sampling (FPS) [31, 33] is a popular sampling technique, but it is heuristic and does not consider downstream applications. Recent efforts [13, 22, 29, 49] have been made on point cloud sampling based on deep learning. For example, SampleNet [22] is a differentiable module, and can select crucial points to improve underlying applications. Our method establishes a pair of samplers based on SampleNet for foreground and background point selection. Distinguished from SampleNet, the sampler pair in this work is derived in an unsupervised and discriminative manner.

Weakly supervised point cloud segmentation. Two popular types of annotations exist for weakly supervised point cloud segmentation. The first type is point-cloud-level annotations [44]. It is also referred to as inexact supervision, which corresponds to object segmentation with image-level

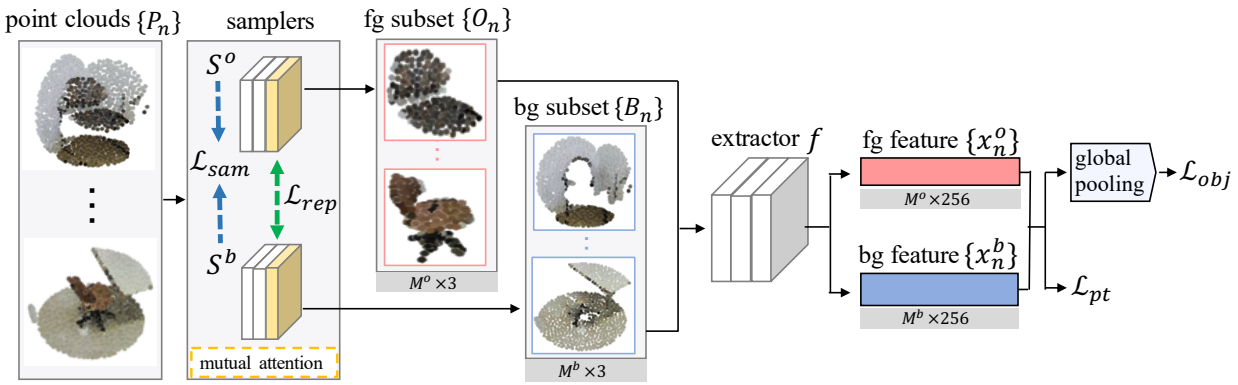


Figure 2: **Approach overview**. The network architecture is composed of two samplers S^o and S^b equipped with the mutual attention modules, and a feature extractor f. The objective function comprises the SampleNet loss $\mathcal{L}_{sam}$ in (1), the repulsion loss $\mathcal{L}_{rep}$ in (2), and the co-contrastive losses, $\mathcal{L}_{pt}$ in (3) and $\mathcal{L}_{obj}$ in (4). Please refer to the text for details.

annotations [2] on images. Wei *et al*. [44] propose multi-path region mining for object segmentation with only cloud-level labels. The second type is to annotate a few points for each point cloud in the training set. It is referred to as incomplete supervision, which is similar to segment objects with coarse pixel annotations, *e.g*. scribble [25], on images. Xu *et al*. [48] utilize multiple instance learning and a spatial constraint loss for object and shape segmentation using training data where a few points are labeled for each cloud.

There are two major differences between weakly supervised segmentation and object co-segmentation. First, weakly supervised segmentation works on *multiple predefined* object categories covered by training data. Co-segmentation works on a *single arbitrary* object class. Second, weakly supervised segmentation requires weakly annotated training data to train the model, and the model remains fixed during testing. It requires re-training on whole dataset when a new category is presented. A co-segmentation model is online optimized by taking a set of point clouds covering the same objects as the input. To sum up, our method requires neither the label of objects nor any point-level annotations, but only a set of point clouds covering objects of the same category for co-segmentation.

3. Proposed Method

Our method is described in this section.

3.1. Problem statement

We are given a set of N point clouds $D = \{P_n\}_{n=1}^N$ covering objects of an unknown category. Without loss of generality, we assume the number of points in each cloud is M, *i.e*., $P_n = \{\mathbf{p}_{nm}\}_{m=1}^M$, where point $\mathbf{p}_{nm} \in \mathbb{R}^3$ is represented by its 3D coordinate. Point cloud object co-segmentation aims to discover the subset $\hat{O}_n \subset P_n$ that contains all points belonging to the common object for each point cloud P_n. Note that neither point-level nor point-cloud-level annotations are provided. And only geometric features are used (without any RGB information).

Figure 2 illustrates the proposed method. Taking $D = \{P_n\}_{n=1}^N$ as the input, the object sampler S^o and background sampler S^b infer a foreground subset $O_n \subset P_n$ and a background subset $B_n \subset P_n$ for each point cloud P_n. The mutual attention module with its details given in Figure 3 is embedded in both samplers S^o and S^b. It estimates cross-cloud, point-wise mutual correlation that is then taken into consideration during sampling. After applying the feature extractor f to all sampled points, we get point-level foreground features $\{\mathbf{x}_n^o\}$ and background features $\{\mathbf{x}_n^b\}$ for each P_n. The object-level foreground features X_n^o and background features X_n^b are obtained via average pooling. Training of the whole network is driven by the proposed co-contrastive losses in both object and point levels.

3.2. Object and background samplers

We formulate point cloud object segmentation as a foreground point sampling problem. Recent studies [13, 22] present differentiable relaxation for point cloud sampling. In this work, we develop the object sampler S^o and the background sampler S^b upon SampleNet [22]. In the following, we will give a brief description of SampleNet and specify our three modifications to it for co-segmentation.

SampleNet. Given a set of point clouds $D = \{P_n\}_{n=1}^N$, SampleNet targets at sampling a subset $R_n \subset P_n$ for each P_n with a pre-defined number of sampled points so that the downstream task working on $R = \{R_n\}$ can be optimized. To keep the whole process differentiable, SampleNet first determines a small point group Q_n for each P_n, where Q_n may not be a subset of P_n. Then, Q_n is projected onto P_n to obtain R_n. The set of point clouds after sampling $R = \{R_n\}$ serve as the input to the downstream task. The

objective function of SampleNet is defined by

$$\mathcal{L}_{sam} = \left[\sum_{n=1}^{N} \alpha \mathcal{L}_s(Q_n, P_n) + \lambda \mathcal{L}_p(Q_n, P_n)\right] + \mathcal{L}_t(R), \tag{1}$$

where α and λ are two coefficients for loss function weighting; $\mathcal{L}_s(Q_n, P_n)$ encourages the *simplified* counterpart Q_n to be similar to P_n by minimizing their Chamfer distance [1]; After *projecting* Q_n onto P_n, $\mathcal{L}_p(Q_n, P_n)$ denotes the approximation loss when using the nearest neighbor sampling operation to construct R_n; Finally, $\mathcal{L}_t(R)$ is the loss function of the downstream *task* working on R.

Modifications. To accomplish co-segmentation on point clouds, we make three modifications to SampleNet, including the downstream task, a pair of competing samplers, and cross-cloud mutual attention as described below.

The downstream task in this work is unsupervised co-segmentation. Due to the lack of data annotations, we develop unsupervised co-contrastive losses, detailed in Section 3.4, to optimize the samplers. Our model has a feature extractor f to generate features for each sampled point. The point-level features can be combined to yield object-level features. The co-contrastive losses are applied in both point and object levels for sampler optimization.

For co-segmentation, we target at separating foreground points from the rest. To this end, we employ an object sampler S^o and a background sampler S^b to respectively infer a foreground subset $O_n \subset P_n$ and a background subset $B_n \subset P_n$ for each point cloud P_n. By using the co-contrastive losses, samplers S^o and S^b tend to collect foreground and background points, respectively. To further prevent the two samplers from selecting the same points, we integrate the repulsion loss [42] into sampler training, *i.e.*,

$$\mathcal{L}_{rep} = \sum_{n=1}^{N} \max(\sigma - d_c(O_n, B_n), 0), \tag{2}$$

where d_c is the Chamfer distance and the hyperparameter $\sigma = 1$ controls the separation margin.

SampleNet infers a point cloud at a time, and cannot explicitly discover common patterns for co-segmentation. As described in the following, we integrate a mutual attention module into SampleNet so that mutual point correlations across point clouds can be leveraged for co-segmentation.

3.3. Mutual attention module

Inspired by the self-attention module [41] where non-local operations help capture long-range dependencies, we develop a mutual attention module to discover cross-cloud point correlation. Compared with the self-attention module exploring position correlation *within* an image, our mutual attention module illustrated in Figure 3 focuses on the mutual point correlation *across* point clouds in a *mini-batch*.

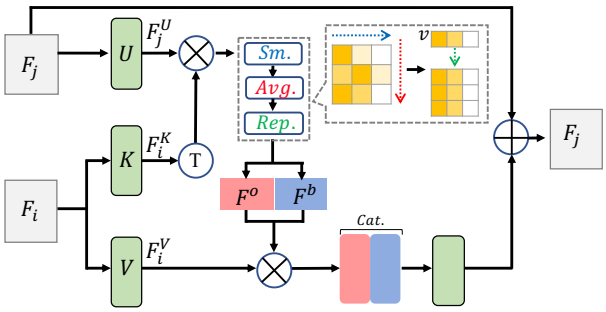


Figure 3: **Architecture of the mutual attention module**. Symbols $\otimes$, $\oplus$, $\top$, $Sm.$, $Avg.$, $Rep.$ and $Cat.$ denote matrix multiplication, element-wise sum, matrix transposition, softmax, averge, repeat, and concatenation respectively. A green box represents a multilayer perceptron.

Let $\{F_n \in \mathbb{R}^{M_n \times C}\}_{n=1}^{B}$ be the feature maps of point clouds in a mini-batch, where B is the mini-batch size, M_n is the number of points in the n-th cloud, and C is the feature dimension. Similar to [41], three different layers, *query* U, *key* K, and *value* V, are applied to all feature maps $\{F_n\}$, to get $\{F_n^U \in \mathbb{R}^{M_n \times C'}, F_n^K \in \mathbb{R}^{M_n \times C'}, F_n^V \in \mathbb{R}^{M_n \times C'}\}$, where C' is the new feature dimension.

To explore cross-cloud mutual attention, we focus on a pair of feature maps F_i and F_j, and treat the former as a *reference* and the latter as the *anchor*. As illustrated in Figure 3, we take the query of the anchor F_j^U and the key of the reference F_i^K, and apply the softmax operation to each row of matrix $F_j^U (F_i^K)^\top \in \mathbb{R}^{M_j \times M_i}$. The matrix $F_j^U (F_i^K)^\top \in \mathbb{R}^{M_j \times M_i}$ stores pair-wise point correlation between the anchor and reference clouds. The row-wise softmax operation softly assigns each point of the anchor cloud to its nearest point of the reference cloud. After row-wise softmax, column-wise average pooling is applied, and yields a row vector $\mathbf{v} \in \mathbb{R}^{1 \times M_i}$, which highlights points of the reference cloud with strong mutual correlation across clouds. We repeat the vector M times to obtain the foreground attention map F^o, and matrix multiplies it by the reference's value F_i^V. This way, we emphasize the features of points with strong mutual correlation. Symmetrically, background points can be emphasized by considering $F^b = 1 - F^o$. As illustrated in Figure 3, residual learning is included for better performance. The mutual attention module is embedded on the last layer of both samplers.

3.4. Co-contrastive loss

Contrastive learning [4,8,15,47] has been studied for unsupervised representation learning. In this work, we implement it in both point and object levels where intra-cloud and inter-cloud contrastive learning are enabled, respectively. A training data pair for contrastive learning is often generated from one data example via augmentation. In the object level, a data pair is created from different point clouds.

Table 1: Segmentation results (mIoU) on the OBJ_BG test set of ScanObjectNN of different methods with diverse supervision levels and settings. 100%, 10%, and 1pt denotes the methods trained with 100%, 10%, and single labeled points per object category, respectively. Cloud indicates the methods trained with cloud-level labels.

Setting	Model	Label	CatAvg	Bag	Bin	Box	Cabinet	Chair	Desk	Display	Door	Shelf	Table	Bed	Pillow	Sink	Sofa	Toilet
Full Sup.	BGA-DGC [43]	100%	0.753	0.76	0.81	0.73	0.73	0.84	0.72	0.76	0.83	0.60	0.76	0.80	0.78	0.64	0.79	0.82
	BGA-PN++ [33]		0.775	0.75	0.83	0.79	0.75	0.84	0.77	0.79	0.83	0.62	0.77	0.81	0.75	0.68	0.81	0.85
Weak Sup.	Xu *et al.* [48]	10%	0.602	0.57	0.67	0.56	0.55	0.76	0.49	0.66	0.80	0.36	0.56	0.61	0.58	0.55	0.68	0.58
	Xu *et al.* [48]	1pt	0.494	0.30	0.62	0.23	0.48	0.67	0.35	0.62	0.70	0.36	0.42	0.54	0.53	0.44	0.57	0.51
	Xu *et al.* [48]	Cloud	0.288	0.34	0.42	0.14	0.15	0.02	0.28	0.63	0.32	0.15	0.38	0.28	0.49	0.41	0.10	0.21
	MPRM [44]	Cloud	0.518	0.47	0.66	0.26	0.59	0.66	0.32	0.60	0.76	0.48	0.41	0.50	0.50	0.51	0.64	0.43
Unsup.	K-means	-	0.389	0.43	0.42	0.41	0.38	0.40	0.38	0.38	0.45	0.34	0.31	0.38	0.37	0.39	0.41	0.40
	AdaCoSeg [53]	-	0.385	0.38	0.31	0.48	0.33	0.55	0.37	0.44	0.35	0.26	0.23	0.37	0.29	0.43	0.54	0.44
	Ours	-	0.605	0.66	0.70	0.68	0.55	0.58	0.46	0.62	0.74	0.48	0.44	0.64	0.60	0.57	0.64	0.68

Besides, it is used for co-segmentation. Thus, we name the resultant objective functions as co-contrastive losses.

As shown in Figure 2, the two samplers, S^o and S^b, infer a foreground subset $O_n = \{\mathbf{p}^o_{nm}\}_{m=1}^{M^o}$ and a background subset $B_n = \{\mathbf{p}^b_{nm}\}_{m=1}^{M^b}$ for each point cloud P_n respectively, where M^o and M^b are the numbers of the sampled foreground and background points. The feature extractor f is applied to each sampled foreground point $\mathbf{p}^o_{nm}$ and gets its 256-dimensional feature vector $\mathbf{x}^o_{nm} = f(\mathbf{p}^o_{nm})$. The object-level foreground feature vector X^o_n for point cloud P_n is obtained by global max pooling over $\{\mathbf{x}^o_{nm}\}_{m=1}^{M^o}$. Similarly, we have the point-level feature vector $\mathbf{x}^b_{nm}$ for each sampled background point $\mathbf{p}^b_{nm}$ and object-level background feature vector X^b_n for each point cloud P_n.

Point co-contrastive loss. It is designed to realize intra-cloud contrastive learning. Namely, for a point cloud, its sampled foreground points should be highly similar to each other while far away from its sampled background points. This point-level co-contrastive loss is defined by

$$\mathcal{L}_{pt} = \sum_{n=1}^{N} \sum_{i,j=1}^{M^o} -\log \frac{\exp(\langle \mathbf{x}^o_{ni}, \mathbf{x}^o_{nj} \rangle)}{\sum_{k=1}^{M^b} \exp(\langle \mathbf{x}^o_{ni}, \mathbf{x}^b_{nk} \rangle)}, \quad (3)$$

where pair-wise similarity is measured by using inner product and N is the number of the given point clouds.

Object co-contrastive loss. It is developed to implement inter-cloud contrastive learning. We maximize the feature similarity of the estimated objects across different point clouds while minimizing the similarity of the estimated object and background, *i.e.*

$$\mathcal{L}_{obj} = \sum_{i,j=1}^{N} -\log \frac{\exp(\langle X^o_i, X^o_j \rangle)}{\sum_{k=1}^{N} \exp(\langle X^o_i, X^b_k \rangle)}. \quad (4)$$

Objective function. After replacing the downstream loss $\mathcal{L}_t$ in (1) by the co-contrastive losses in (3) and (4), the objective function we use for co-segmentation is

$$\mathcal{L} = \mathcal{L}_{sam} + \mathcal{L}_{rep}. \quad (5)$$

3.5. Implementation details

The proposed method is implemented in Pytorch [28]. We use DGCNN [43] as the feature extractor and pre-train it on the ModelNet40 dataset [46] with rotation, jitter, scale and random size of input points for augmentation. Like [22], the feature extractor is fixed during training. We set $M^o = 512$ and $M^b = 512$ for the object and background samplers, respectively. Note that the performance is insensitive to these values since only crucial points are sampled by SampleNet, as illustrated in Figure 6a. The point feature dimension C in the mutual attention module is set to 256. The three layers U, K, and V are applied to generate C'=128 features. During inference, we simply use k nearest neighbor matching between the sampled object points and the input point cloud to spread the predictions. Like [22], PointNet [32] is used as the backbone for both samplers.

4. Experimental Results

This section evaluates the proposed method for point cloud object co-segmentation. We present both quantitative and qualitative results, and conduct ablation studies for analyzing individual components of our method.

4.1. Datasets and evaluation metrics

Since point cloud object co-segmentation is a new task, there is no established benchmark dataset for its evaluation yet. We investigate two datasets for this new co-segmentation task, ScanObjectNN [40] and S3DIS [3], with their details described below.

ScanObjectNN. It is a new real-world point cloud object dataset built on two previous datasets collected from scanned indoor scene data, SceneNN [19] and ScanNet [12]. It has 15 object classes and 2,902 objects. The dataset was collected for object recognition, but it also provides point-level annotation for object segmentation. For offering more practical challenges and different levels of difficulties, ScanObjectNN provides several variants

Table 2: Segmentation results (mIoU) on different variants of the ScanObjectNN dataset.

Setting	Model	Label	T25	T25R	T50R	T50RS
Full Sup.	BGA-DGC [43]	100%	0.75	0.74	0.76	0.75
	BGA-PN++ [33]		0.77	0.77	0.76	0.76
Weak Sup.	Xu *et al*. [48]	10%	0.55	0.53	0.50	0.52
	Xu *et al*. [48]	1pt	0.48	0.50	0.47	0.46
	Xu *et al*. [48]	Cloud	0.21	0.19	0.17	0.13
	MPRM [44]	Cloud	0.43	0.43	0.41	0.40
Unsup.	K-means	-	0.33	0.33	0.32	0.34
	AdaCoSeg [53]	-	0.31	0.35	0.33	0.34
	Ours	-	0.51	0.48	0.46	0.48

Table 3: Segmentation results (mIoU) on the S3DIS dataset.

Setting	Model	Label	CatAvg	bkcase	chair	door	sofa	table
Full Sup.	BGA-DGC [43]	100%	0.716	0.34	0.96	0.69	0.69	0.90
	BGA-PN++ [33]		0.729	0.44	0.92	0.71	0.75	0.83
Weak Sup.	Xu *et al*. [48]	10%	0.706	0.47	0.84	0.87	0.70	0.65
	Xu *et al*. [48]	1pt	0.491	0.16	0.77	0.62	0.44	0.46
	Xu *et al*. [48]	Cloud	0.252	0.11	0.18	0.32	0.19	0.46
	MPRM [44]	Cloud	0.312	0.32	0.31	0.28	0.27	0.38
Unsup.	K-means	-	0.267	0.36	0.18	0.32	0.27	0.20
	AdaCoSeg [53]	-	0.248	0.34	0.24	0.15	0.38	0.28
	Ours	-	0.463	0.36	0.51	0.45	0.50	0.49

by including background points, over-covering, and under-covering objects [40]. There are five variants: OBJ_BG, PB_T25, PB_T25_R, PB_T50_R, and PB_T50_RS. The basic variant is OBJ_BG, where objects are attached with background data cropped with the ground-truth bounding boxes. In reality, the detected bounding boxes could be inaccurate. For modeling the inaccuracy, four variants are derived from OBJ_BG by perturbing the ground-truth bounding boxes and over/under-covering the objects. The prefix PB denotes perturbation while T, R, and S represent translation, rotation and scaling to the bounding boxes, respectively. T25 and T50 respectively denote randomly shifting bounding boxes by up to 25% and 50% of their sizes along axes. The object/background ratios in OBJ_BG, PB_T25, PB_T25_R, PB_T50_R, and PB_T50_RS are 0.64, 0.52, 0.50, 0.48, and 0.54, respectively.

S3DIS. This dataset [3] is proposed for indoor scene understanding and widely used in the point cloud semantic segmentation task. To establish the dataset for object segmentation, we follow a similar process as ScanObjectNN to crop the object point cloud. Instead of the object's bounding box, we use a sphere whose origin is at the object's center for cropping the object and including more background points. The object/background ratio is 0.364 in this dataset. The dataset consists of six areas covering several rooms. Following previous work [48], we use Area 5 for evaluation. We use five object classes (bookcase, chair, door, table, and sofa) with only xyz coordinates in S3DIS for evaluation.

Evaluation metric. Object co-segmentation can be viewed as a two-class (foreground, background) segmentation problem. Hence, we follow the convention of previous work [44, 48] and use mean Intersect over Union (mIoU) as the evaluation metric.

4.2. Competing methods and comparisons

To the best of our knowledge, our method is the first one for point cloud object co-segmentation. Hence, there is no method of the same kind for performance comparison. Thus, we compare our method with point cloud object segmentation methods with three different supervision settings. First, fully supervised methods [40] for point cloud segmentation are compared, and they serve as the references for performance upper bound. Second, two types of state-of-the-art weakly-supervised segmentation methods [3, 12] are compared. They aim to segment 3D objects using either partial point-level labels or cloud-level labels as the weak form of annotations. Third, the state-of-the-art *shape* co-segmentation method, AdaCoSeg [53], is compared. Despite the differences between shape and object co-segmentation discussed in Section 2, the method is relevant and we make a comparison with it by setting the number of parts to 2, *i.e*., foreground and background.

Fully supervised. Ut *et al*. [40] proposed a background-aware classification network (BGA), which can be built on top of PointNet++ [33] (BGA-PN++) or DGCNN [43] (BGA-DGC) by joint learning of classification and segmentation. Their method is trained on ground-truth annotations.

Weakly supervised with partial point-level labels. Xu *et al*. [48] proposed a weakly-supervised method that requires only 10 percent of point-level annotations per object or even a single-point annotation for object segmentation. We apply the method on ScanObjectNN and S3DIS datasets by giving 10 percent labeled points or a single labeled point for each class (including the background class).

Weakly supervised with cloud-level labels. Wei *et al*. [44] proposed NPRM, a 3D semantic segmentation method that only requires cloud-level labels indicating the presence of semantic classes in a cloud. For getting segmentation from classification, the method extends class activation maps [52] to 3D point clouds for locating points of a class. For ScanObjectNN, we train the method on 15 object classes by giving the object-level labels and report the segmentation performance. S3DIS is processed in the same way. Moreover, we extend Xu *et al*. [48] method to cloud-level labels by removing the incomplete branch for comparison.

Unsupervised. Zhu *et al*. [53] proposed an unsupervised shape co-segmentation method. They first conduct offline

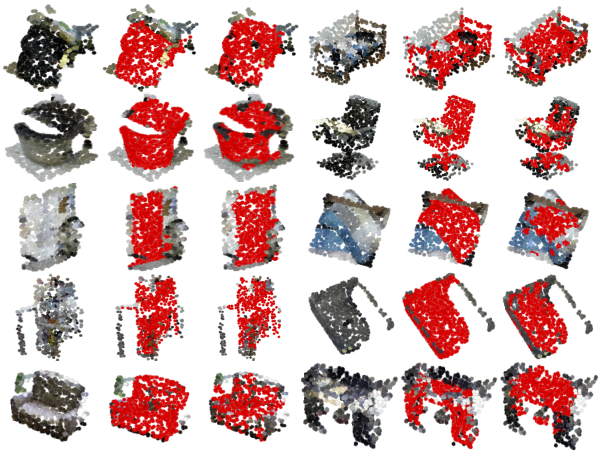


Figure 4: Qualitative results on the ScanObjectNN dataset. We show examples of ten different classes. From left to right and top to down, they are bag, bed, bin, chair, door, pillow, shelf, sink, sofa, and table. For each example, we show the input cloud, the ground-truth label, and our segmentation result.

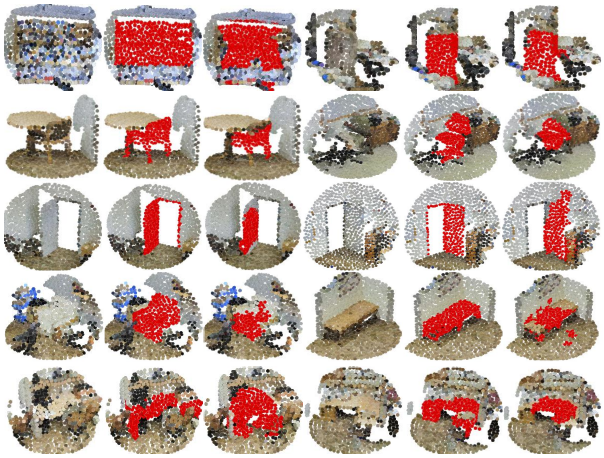


Figure 5: Qualitative results on the S3DIS dataset. We show examples of five object classes: from top to bottom, bookcase, chair, door, sofa, and table, with two examples for each class. For each example, we show the input cloud, the ground-truth label and our segmentation result.

training for the part prior network. Co-segmentation is carried out by minimizing a group consistency loss. We set the number of co-segmented shapes to 2, indicating the foreground and background, in their method for comparison. Moreover, we use K-means to cluster the embeddings ($K = 2$) from the model pre-trained on ModelNet40 as another unsupervised method for comparison.

Table 1 compares the proposed method with competing methods with different supervision settings on the ScanObjectNN dataset. Our method considerably outperforms the shape co-segmentation model AdaCoSeg, since AdaCoSeg is originally designed for synthetic object part decomposition and cannot deal with cluttered backgrounds in the real-world dataset. With similar settings, the proposed method performs favorably against the weakly supervised method with cloud-level labels [44] by a large margin, around 0.09 in mIoU. Their method utilizes CAM for obtaining segmentation from classification. It is well-known that CAM tends to concentrate on the discriminative parts for classification. On the contrary, our method focuses on feature contrastive learning instead of discriminative classification, leading to better results. Our method even outperforms the weakly supervised method with partial point-level labels [48] in most settings, although requiring much less supervision. Given the labels for 10% of points, their method achieves 0.602 in mIoU, while our co-segmentation method achieves 0.605 without requiring any point-level labels. Our method only requires a set of point clouds covering the same class. When working with single-point labels or cloud-level labels, although the annotation effort is much reduced, their performance significantly drops to 0.494 and 0.288, respectively. The supervised BGA methods exhibit superior performance with a much higher annotation cost than ours.

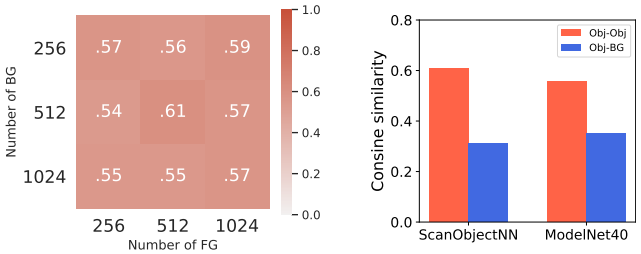
Table 2 reports the performance on different variants of the ScanObjectNN dataset. Our method can keep up with others at a similar pace despite the challenges that its unsupervised nature could suffer due to more clutter and missing parts. Table 3 tabulates the results for the S3DIS dataset. Figure 4 and Figure 5 show several 3D object segmentation results with our method for the ScanObjectNN and S3DIS datasets, respectively. Our method gets promising results across object classes and datasets. In Figure 4, our method can separate the background points from the object well, even with proximity, *e.g.* the chair on the right of the second row. For the more challenging dataset (with more background points included), our methods still achieve the encouraging result. Through contrastive learning, our method can train on an arbitrary class, even with very different scales, *e.g.* bookcase and chair in Figure 5.

4.3. Analysis and ablation study

We report analysis on parameters and ablation studies to evaluate the components of the proposed method.

Number of sampled points. The numbers of sampled foreground and background points, M^o and M^b, are predetermined. Figure 6a shows the performance of our method with different configurations of foreground and background sample numbers. Although the numbers of points in objects could be very different, our experiments show that the samplers' performance is not sensitive to the pre-determined numbers. It is because that SampleNet learns to pick up crucial points for contrastive learning.

Feature extractor pre-training. Our method needs a fea-



(a) (b)

Figure 6: (a) Segmentation performance in mIoU on ScanObjectNN OBJ_BG with different configurations of sampled FG/BG points. (b) Feature similarity of points in ScanObjectNN OBJ_BG by using the feature extractors pre-trained on ScanObjectNN and ModelNet40.

ture extractor that can extract useful features for guiding the learning of SampleNet. We experiment with two ways for pre-training. First, we pre-train the feature extractor on clean object data (without background points) of the ScanObjectNN dataset. The setting is too ideal since it requires ground-truth segmentation. We report this setting only for reference. Second, we pre-train the feature extractor using a synthetic CAD dataset, ModelNet40 [6], for classification. Although there is a significant domain gap between real-world scans and CAD models [40, 47], we find that the feature extractor pre-trained on ModelNet40 can yield good performance. Figure 6b reports the average object-object and object-background similarity values of points on the ScanObjectNN OBJ_BG dataset for pre-training on both datasets. It shows that the feature extractor pre-trained on CAD models offers useful features for real-world scans despite the semantic gap. By focusing on learning the similarity, contrastive learning can further reduce the gap. In all experiments, we pre-train the feature extractor using ModelNet40 for our method. Although our feature extractor is pre-trained on ModelNet40, we consider our model unsupervised because it can be applied to object categories that are not covered by ModelNet40.

Ablation studies. We conduct ablation studies to evaluate the contributions of the developed components, including the two co-contrastive losses, $\mathcal{L}_{obj}$ in (4) and $\mathcal{L}_{pt}$ in (3), the repulsion loss $\mathcal{L}_{rep}$ in (2), and the mutual attention module. As a reference, the self-attention module [41] is included for comparison. Table 4 shows the performance by using different combinations of these losses and the attention modules. The results validate the effectiveness of each of these developed components for co-segmentation.

4.4. Application to improve classification

Uy *et al*. [40] find that joint classification and segmentation improve real-world point cloud classification. They propose a background-aware network (BGA) to handle the presence of clutter in real-scan point clouds and improve classification results by adding a segmentation-guided branch. For segmentation, training BGA requires point-level annotations that are much more expensive than the cloud-level labels required for classification alone. Our method can serve to obtain the pseudo labels for training the BGA network with nearly no extra effort. Table 5 shows that training with pseudo labels can improve classification accuracy. Taking PointNet++ as an example, its classification accuracy is 78.5%. By joint segmentation and classification, BGA improves the accuracy to 80.2% at the expense of high annotation cost. By using our method for generating pseudo labels, the accuracy is 79.4% with minimal cost.

Table 4: Segmentation performance in mIoU with different losses, attention modules, and their combinations.

Loss			Attention		Dataset	
$\mathcal{L}_{obj}$	$\mathcal{L}_{pt}$	$\mathcal{L}_{rep}$	Self [41]	Mutual	ScanObject	S3DIS
√					0.54	0.41
	√				0.49	0.36
√		√			0.54	0.43
	√	√			0.50	0.37
√	√				0.55	0.42
√	√	√			0.57	0.43
√	√	√	√		0.59	0.45
√	√	√		√	**0.61**	**0.46**

Table 5: Accuracy (%) on ScanObjectNN PB_T50_RS.

	PointNet++	DGCNN
classification alone	78.5	78.1
BGA (pseudo label)	79.4	79.1
BGA (ground truth)	80.2	79.7

5. Conclusions

This paper presents a new problem, point cloud object co-segmentation, and proposes a method to solve it without using expensive annotations. The method comprises three key components, a pair of samplers, a mutual attention module, and a contrastive learning task to accomplish the point cloud object co-segmentation task. Our method is extensively evaluated on two challenging real-world datasets featuring incomplete data and clutter. Results show that our method performs favorably against state-of-the-art weakly-supervised object segmentation methods. Furthermore, we demonstrate that our method can provide pseudo labels for improving object classification in a real-world dataset.

Acknowledgments. This work was supported in part by the Ministry of Science and Technology under grants 109-2221-E-009-113-MY3, 110-2628-E-A49-008, 110-2634-F-007-015, 110-2221-E-002-124-MY3, and 110-2634-F-002-026. It was also funded in part by Qualcomm through a Taiwan University Research Collaboration Project.

References

[1] Panos Achlioptas, Olga Diamanti, Ioannis Mitliagkas, and Leonidas Guibas. Learning representations and generative models for 3D point clouds. In *ICML*, 2018. 4

[2] Jiwoon Ahn and Suha Kwak. Learning pixel-level semantic affinity with image-level supervision for weakly supervised semantic segmentation. In *CVPR*, 2018. 3

[3] 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 *CVPR*, 2016. 1, 2, 5, 6

[4] Philip Bachman, R Devon Hjelm, and William Buchwalter. Learning representations by maximizing mutual information across views. In *NIPS*, 2019. 4

[5] 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 *ICCV*, 2019. 1

[6] Angel X Chang, Thomas Funkhouser, Leonidas Guibas, Pat Hanrahan, Qixing Huang, Zimo Li, Silvio Savarese, Manolis Savva, Shuran Song, Hao Su, et al. ShapeNet: An information-rich 3D model repository. *arXiv preprint arXiv:1512.03012*, 2015. 1, 8

[7] Nenglun Chen, Lingjie Liu, Zhiming Cui, Runnan Chen, Duygu Ceylan, Changhe Tu, and Wenping Wang. Unsupervised learning of intrinsic structural representation points. In *CVPR*, 2020. 1

[8] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *ICML*, 2020. 4

[9] Xiaozhi Chen, Huimin Ma, Ji Wan, Bo Li, and Tian Xia. Multi-view 3D object detection network for autonomous driving. In *CVPR*, 2017. 1

[10] Yun-Chun Chen, Yen-Yu Lin, Ming-Hsuan Yang, and Jia-Bin Huang. Show, match and segment: joint weakly supervised learning of semantic matching and object co-segmentation. *TPAMI*, 2020. 2

[11] Zhiqin Chen, Kangxue Yin, Matthew Fisher, Siddhartha Chaudhuri, and Hao Zhang. BAE-NET: Branched autoencoder for shape co-segmentation. In *CVPR*, 2019. 2

[12] Angela Dai, Angel X Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. ScanNet: Richly-annotated 3d reconstructions of indoor scenes. In *ICCV*, 2017. 1, 5, 6

[13] Oren Dovrat, Itai Lang, and Shai Avidan. Learning to sample. In *CVPR*, 2019. 2, 3

[14] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the KITTI vision benchmark suite. In *CVPR*, 2012. 1

[15] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *CVPR*, 2020. 4

[16] Kuang-Jui Hsu, Yen-Yu Lin, and Yung-Yu Chuang. Co-attention CNNs for unsupervised object co-segmentation. In *IJCAI*, 2018. 1, 2

[17] Kuang-Jui Hsu, Yen-Yu Lin, and Yung-Yu Chuang. DeepCo3: Deep instance co-segmentation by co-peak search and co-saliency detection. In *CVPR*, 2019. 1, 2

[18] Ruizhen Hu, Lubin Fan, and Ligang Liu. Co-segmentation of 3d shapes via subspace clustering. In *Computer graphics forum*, 2012. 2

[19] Binh-Son Hua, Quang-Hieu Pham, Duc Thanh Nguyen, Minh-Khoi Tran, Lap-Fai Yu, and Sai-Kit Yeung. SceneNN: A scene meshes dataset with annotations. In *3DV*, 2016. 5

[20] Dingwen Zhang Junwei Han, Rong Quan and Feiping Nie. Robust object co-segmentation using background prior. *TIP*, 2018. 1, 2

[21] Loic Landrieu and Martin Simonovsky. Large-scale point cloud semantic segmentation with superpoint graphs. In *CVPR*, 2018. 1

[22] Itai Lang, Asaf Manor, and Shai Avidan. SampleNet: Differentiable point cloud sampling. In *CVPR*, 2020. 2, 3, 5

[23] Weihao Li, Omid Hosseini Jafari, and Carsten Rother. Deep object co-segmentation. In *ACCV*, 2018. 1, 2

[24] Zhengfa Liang, Yulan Guo, Yiliu Feng, Wei Chen, Linbo Qiao, Li Zhou, Jianfeng Zhang, and Hengzhu Liu. Stereo matching using multi-level cost volume and multi-scale feature constancy. *TPAMI*, 2019. 1

[25] Di Lin, Jifeng Dai, Jiaya Jia, Kaiming He, and Jian Sun. ScribbleSup: Scribble-supervised convolutional networks for semantic segmentation. In *CVPR*, 2016. 3

[26] Kaichun Mo, Shilin Zhu, Angel X Chang, Li Yi, Subarna Tripathi, Leonidas J Guibas, and Hao Su. PartNet: A large-scale benchmark for fine-grained and hierarchical part-level 3d object understanding. In *CVPR*, 2019. 1, 2

[27] Sanjeev Muralikrishnan, Vladimir G Kim, and Siddhartha Chaudhuri. Tags2Parts: Discovering semantic regions from shape tags. In *CVPR*, 2018. 2

[28] Kevin Musgrave, Serge Belongie, and Ser-Nam Lim. Pytorch metric learning. *arXiv preprint arXiv:2008.09164*, 2020. 5

[29] Ehsan Nezhadarya, Ehsan Taghavi, Ryan Razani, Bingbing Liu, and Jun Luo. Adaptive hierarchical down-sampling for point cloud classification. In *CVPR*, 2020. 2

[30] Quang-Hieu Pham, Pierre Sevestre, Ramanpreet Singh Pahwa, Huijing Zhan, Chun Ho Pang, Yuda Chen, Armin Mustafa, Vijay Chandrasekhar, and Jie Lin. A* 3D dataset: Towards autonomous driving in challenging environments. In *ICRA*, 2020. 1

[31] Charles R Qi, Or Litany, Kaiming He, and Leonidas J Guibas. Deep hough voting for 3D object detection in point clouds. In *ICCV*, 2019. 2

[32] Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. PointNet: Deep learning on point sets for 3D classification and segmentation. In *CVPR*, 2017. 1, 5

[33] Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J Guibas. PointNet++: Deep hierarchical feature learning on point sets in a metric space. In *NIPS*, 2017. 1, 2, 5, 6

[34] Xavier Roynard, Jean-Emmanuel Deschaud, and François Goulette. Paris-Lille-3D: A large and high-quality ground-truth urban point cloud dataset for automatic segmentation and classification. *IJRR*, 2018. 1

[35] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. ImageNet large scale visual recognition challenge. *IJCV*, 2015. 1

[36] Jonathan Sauder and Bjarne Sievers. Self-supervised deep learning on point clouds by reconstructing space. In *NIPS*, 2019. 1

[37] Zhenyu Shu, Chengwu Qi, Shiqing Xin, Chao Hu, Li Wang, Yu Zhang, and Ligang Liu. Unsupervised 3D shape segmentation and co-segmentation via deep learning. *CAGD*, 2016. 2

[38] 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 *CVPR*, 2019. 1

[39] Chung-Chi Tsai, Weizhi Li, Kuang-Jui Hsu, Xiaoning Qian, and Yen-Yu Lin. Image co-saliency detection and co-segmentation via progressive joint optimization. *TIP*, 2018. 2

[40] Mikaela Angelina Uy, Quang-Hieu Pham, Binh-Son Hua, Thanh Nguyen, and Sai-Kit Yeung. Revisiting point cloud classification: A new benchmark dataset and classification model on real-world data. In *ICCV*, 2019. 1, 2, 5, 6, 8

[41] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NIPS*, 2017. 2, 4, 8

[42] Xinlong Wang, Tete Xiao, Yuning Jiang, Shuai Shao, Jian Sun, and Chunhua Shen. Repulsion loss: Detecting pedestrians in a crowd. In *CVPR*, 2018. 4

[43] Yue Wang, Yongbin Sun, Ziwei Liu, Sanjay E Sarma, Michael M Bronstein, and Justin M Solomon. Dynamic graph cnn for learning on point clouds. *TOG*, 2019. 1, 5, 6

[44] Jiacheng Wei, Guosheng Lin, Kim-Hui Yap, Tzu-Yi Hung, and Lihua Xie. Multi-path region mining for weakly supervised 3D semantic segmentation on point clouds. In *CVPR*, 2020. 2, 3, 5, 6, 7

[45] Wenxuan Wu, Zhongang Qi, and Li Fuxin. PointConv: Deep convolutional networks on 3D point clouds. In *CVPR*, 2019. 1

[46] Zhirong Wu, Shuran Song, Aditya Khosla, Fisher Yu, Linguang Zhang, Xiaoou Tang, and Jianxiong Xiao. 3D ShapeNets: A deep representation for volumetric shapes. In *CVPR*, 2015. 5

[47] Saining Xie, Jiatao Gu, Demi Guo, Charles R Qi, Leonidas Guibas, and Or Litany. PointContrast: Unsupervised pre-training for 3D point cloud understanding. In *ECCV*, 2020. 1, 4, 8

[48] Xun Xu and Gim Hee Lee. Weakly supervised semantic point cloud segmentation: Towards 10x fewer labels. In *CVPR*, 2020. 3, 5, 6, 7

[49] Jiancheng Yang, Qiang Zhang, Bingbing Ni, Linguo Li, Jinxian Liu, Mengdie Zhou, and Qi Tian. Modeling point clouds with self-attention and gumbel subset sampling. In *CVPR*, 2019. 2

[50] Ze-Huan Yuan, Tong Lu, and Yirui Wu. Deep-dense conditional random fields for object co-segmentation. In *IJCAI*, 2017. 1, 2

[51] Hengshuang Zhao, Li Jiang, Chi-Wing Fu, and Jiaya Jia. PointWeb: Enhancing local neighborhood features for point cloud processing. In *CVPR*, 2019. 1

[52] Bolei Zhou, Aditya Khosla, Agata Lapedriza, Aude Oliva, and Antonio Torralba. Learning deep features for discriminative localization. In *CVPR*, 2016. 6

[53] Chenyang Zhu, Kai Xu, Siddhartha Chaudhuri, Li Yi, Leonidas J Guibas, and Hao Zhang. AdaCoSeg: Adaptive shape co-segmentation with group consistency loss. In *CVPR*, 2020. 2, 5, 6