Unsupervised Point Cloud Co-part Segmentation via Co-attended Superpoint Generation and Aggregation

Ardian Umam, Cheng-Kun Yang, Jen-Hui Chuang *Senior Member, IEEE*,
and Yen-Yu Lin *Senior Member, IEEE*

***Abstract*—We propose a co-part segmentation method that takes a set of point clouds of the same category as input where neither a ground truth label nor a prior network is required. With difficulties caused by the label absence, we formulate the co-part segmentation task into two subtasks, including superpoint generation and part aggregation. In the first subtask, our superpoint generation network divides each point cloud into homogeneous partitions, each called superpoint, while in the second subtask, these superpoints are further aggregated into a few semantic parts via our part aggregation network. We introduce the coupled attention blocks in the part aggregation network to explicitly enforce semantic consistency in the segmentation by exploiting intra-, inter-, and paired-cloud geometrical information by minimizing the devised intra-, inter-, and paired-cloud losses, respectively. The intra-cloud loss triggers a semantic segmentation in each point cloud, while the inter-cloud loss considers all clouds to enforce their semantic consistency. The paired-cloud loss is designed to ensure that each part of one point cloud can be discriminatively reconstructed from the superpoints of another point cloud. We perform experiments on two benchmark datasets, ShapeNet part and COSEG, and provide quantitative and qualitative results to demonstrate the superiority of our method over existing methods. We also show that the proposed method can help several downstream tasks, including semi-supervised part segmentation and data augmentation for shape classification. The code for this work is publicly available[1].**

***Index Terms*—Point cloud segmentation, co-part segmentation, co-segmentation, unsupervised learning.**

I. Introduction

Point cloud segmentation is essential to various 3D multimedia analyses and applications, such as navigation [1]–[3], medical imaging [4]–[6], industrial inspection [7], [8], and computer-aided design [9]. Existing state-of-the-art methods for point cloud segmentation [10]–[13] are developed based on deep learning techniques and rely on large-scale, annotated datasets [14]–[18], which entail high costs in terms of point labeling and reduce the applicability.

Co-segmentation is one of the ways to mitigate the high cost of annotating training data for segmentation [19]–[21]. It was first introduced in [22], with the objective of segmenting the common object in a set of images without relying on pixel-level annotations. The underlying assumption is that common objects share similar appearances across images and can thus be jointly segmented from the background.

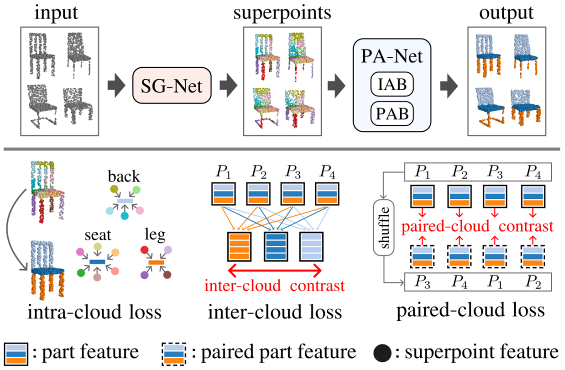

Fig. 1. Given a set of point clouds of the same category, our method first divides each cloud into several homogeneous partitions, called superpoints, via SG-Net. These superpoints are then aggregated into semantically consistent parts via PA-Net. The upper figure visualizes four point clouds with three discovered parts shown in different colors. To this end, the proposed method exploits intra-, inter-, and paired-cloud geometrical information posed in the superpoints with our designated coupled attentions (IAB and PAB) within the PA-Net by optimizing the corresponding intra-, inter-, and paired-cloud losses, respectively, as illustrated in the bottom figure.

Research efforts have been made to extend co-segmentation to 3D scenes such as object co-segmentation [23]–[25] and co-part segmentation [26]–[28]. While object co-segmentation is targeted at segmenting the common objects from the background, co-part segmentation, which is considered in this paper, aims to jointly decompose the common objects into semantically consistent parts across these point clouds. Earlier methods formulate 3D co-part segmentation as clustering with handcrafted features [29], [30], leading to sub-optimal results. To resolve this issue, recent approaches [26]–[28], [31] adopt deep learning, where the features are learned via designated loss functions.

Due to the unavailability of point-level annotations in co-part segmentation, getting accurate segmentation results is challenging. As a result, several methods [26], [28], [31] use weak supervision to enhance segmentation accuracy. Sung *et al.* [26] utilize subsets of part labels to build a shape-dependent dictionary, while AdaCoSeg [28] uses binary part labels from the external ComplementMe dataset [32] to train the prior network for regularizing the shapes of segmented parts. Nonetheless, weak supervision in these methods introduces costly annotations since point-level labeling is required. BAE-Net [27] formulates co-part segmentation as a branched autoencoder that reconstructs the original input; hence no label is needed.

A. Umam, J.-H. Chuang, and Y.-Y. Lin are with the Department of Computer Science, National Yang Ming Chiao Tung University (e-mail: ardianumam.ee09@nycu.edu.tw; jchuang@cs.nctu.edu.tw; lin@cs.nycu.edu.tw).

C.-K. Yang is with the Department of Computer Science and Information Engineering, National Taiwan University (e-mail: d08922002@csie.ntu.edu.tw).

This paper has supplementary downloadable material available at http://ieeexplore.ieee.org., provided by the authors.

[1] https://github.com/ardianumam/pointcloud-copart-segmentation

Each branch learns an implicit function for one universal part. Since there is no explicit loss to encourage semantic consistency, it is prone to mixing different semantics into one part and vice versa, reducing the semantic consensus of the discovered parts. Furthermore, in addition to taking point clouds as input, BAE-Net also requires the mesh representation of the shapes to estimate the implicit functions.

This paper presents a co-part segmentation framework for 3D point clouds that explicitly encourages semantic consistency without using any point or part labels. The resultant segmentation can be further utilized in some downstream tasks, such as shape editing [33], [34], stylization [35], and augmentation [36], as demonstrated in Sec. IV-D. Due to the difficulty caused by the absence of labels, the proposed method formulates the co-part segmentation task into two subtasks, *superpoint generation* and *part aggregation*, which are performed by the designated superpoint generation network (SG-Net) and part aggregation network (PA-Net), respectively, as shown in the first row of Figure 1. Given a set of point clouds, the SG-Net decomposes every cloud into several partitions called superpoints, each of which is derived to be homogeneous, *i.e.*, containing points that belong to the same semantic part of the object. The PA-Net then aggregates these superpoints into a few parts by exploiting intra-, inter-, and paired-cloud geometrical features posed in the superpoints with our designated coupled attentions that consist of intra-inter attention block (IAB) and paired attention block (PAB).

PA-Net identifies R parts of each point cloud by learning R *part tokens*, which can retrieve part features from superpoints by exploiting intra-, inter-, and paired-cloud features with the corresponding losses, *i.e.*, intra-, inter-, and paired-cloud losses, as illustrated in the second row of Figure 1.

First, the intra-cloud loss considers a single cloud and achieves semantic segmentation by enforcing part tokens to attend to similar superpoints. For example, in the category of chairs, one of the part tokens will attend to those superpoints from leg parts where the superpoints are similar to each other. Second, the inter-cloud loss considers part features across all clouds and performs contrastive learning, which enforces similarity among part features associated with the same semantic part while enhancing dissimilarity among part features of different semantic parts. This inter-cloud loss is intended to trigger semantic consistency across clouds.

Third, the paired-cloud loss is designed to ensure that each part of one point cloud can be discriminatively reconstructed from superpoints of another point cloud. For a point cloud pair, the part features of one cloud are used as the queries to aggregate the new part features from another cloud via the attention mechanism. Contrastive learning is performed between the original and new part features to preserve the consensus between the two clouds. In mini-batch optimization, cloud pairing is carried out by random shuffling. With the aforementioned three main losses, the proposed method explicitly enforces semantic consistency on the segmentation result, and such consistency is observed in Figure 1.

The main contributions of this work are summarized as follows. First, we introduce the coupled attention blocks that exploit intra-, inter-, and paired-cloud geometrical information to enable unsupervised point cloud co-part segmentation. Second, we develop the superpoint generation network that learns point features without any point-level labels and can decompose a point cloud into homogeneous partitions, called superpoints. Third, the proposed method outperforms the state-of-the-arts on two part datasets, the ShapeNet part [37] and COSEG [38] datasets. Fourth, the proposed method facilitates several downstream tasks, including semi-supervised part segmentation and data augmentation for shape classification.

II. Related work

Image co-segmentation. Co-segmentation on 2D images has been widely studied, *e.g.*, object co-segmentation [22], [39]–[42] and co-part segmentation [43]–[47]. Image co-segmentation is often formulated as a clustering problem guided by color and pixel positions that characterize the local appearance of objects of interest [22], [40]–[42], [46]–[48]. It can also be accomplished in various ways such as energy minimization [40], region correspondence cost function optimization [41], [42], region histogram modeling [22], visual modeling [39], [43]–[45], discriminative clustering [48], spectral clustering [40], and graph cuts [22], [41]. Unlike these methods, our method is developed for unordered and unstructured 3D point clouds without any additional supervision.

Point cloud co-segmentation. Several tasks explore co-segmentation for 3D point clouds, such as *object* co-segmentation [23]–[25] and co-*part* segmentation [26]–[28]. The two tasks have three major differences, including the task goal, the number of segmentation classes, and the model input. First, object co-segmentation aims to separate common objects from the backgrounds, and co-part segmentation is targeted to decompose objects into matchable and semantically meaningful parts. Second, while object co-segmentation typically segments point clouds into two classes [23], [25], *i.e.*, object (foreground) and background points, co-part segmentation decomposes an object into several semantic parts, and the number of parts varies from category to category, such as two (cup and handle) for a mug and three (body, wing, and tail) for an airplane. Third, object co-segmentation takes point clouds of objects with background points as input, while co-part segmentation accepts clean object-only clouds [26]–[28]. Due to these underlying differences, object co-segmentation methods are typically not applicable to co-part segmentation

To carry out co-part segmentation, research works in [26], [28] employ weak supervision: While Sung *et al.* [26] rely on subsets of parts labels to build a shape-dependent dictionary, AdaCoSeg [28] utilizes prior network trained on binary part labels from the ComplementMe dataset [32]. In addition, AdaCoSeg optimizes the network only for the given data, and the network cannot be applied to new data. BAE-Net [27] introduces a branched-autoencoder to reconstruct the original input; Hence, no label is needed. It tasks each branch to estimate universal part representations. However, its derived loss only considers each point cloud individually and does not explicitly enforce semantic consistency, which may result in incoherent parts across clouds and segmenting several semantics into one part. Unlike these related methods, our method enforces semantic consistency by exploring intra-, inter-, and paired-cloud geometric evidence. Furthermore, our method does not rely on prior data or point/part annotations, and the learned model can be applied to new point clouds.

Superpoints and primitive-based shape abstraction. Superpoints are formed from 3D point clouds by grouping points that share similar characteristics, which can be considered the 3D equivalent of superpixels in 2D data. To divide a point cloud into superpoints, Hui *et al.* [49] devise a contrastive loss that enforces embedding dissimilarity of adjacent points with different superpoint labels and encourages similarity of those within the same labels. Landrieu *et al.* [50] design a label consistency loss such that each superpoint is forced to contain points with the same labels. While the methods in [49]–[51] utilize entire pointwise labels, the work in [52] uses a subset of labels to form the superpoints. Nonetheless, such an approach still relies on the high annotation cost of pointwise labeling. SPG shares the same motivation with our SG-Net on avoiding the need for annotation. Nevertheless, there are two major differences. First, while SPG utilizes handcrafted features defined in [53], our SG-Net employs learnable point features optimized with our designated losses, which leads to consistently better performance, as demonstrated in Table IV. Second, SPG utilizes the l_0-cut pursuit algorithm [54] to recursively split the point cloud into superpoints, which requires a substantial amount of time. Meanwhile, our SG-Net only needs one forward pass to derive the superpoints, with three orders of magnitude faster than SPG.

Several works [55]–[59] perform shape abstraction by decomposing a point cloud into several shape primitives, such as superquadrics [55]–[57], convexes [58] and implicit functions [59]. Since these works more focus on shape decomposition, each semantic part of the shape could be decomposed into several primitives. The works in [58], [59] further utilize part labels to group the decomposed primitives into the corresponding semantic parts. While shape abstraction mainly aims to minimize the cumulative discrepancy between primitives and the original shape where the decomposed primitives may cover several semantics, our superpoint generation network is developed to decompose a point cloud into semantically homogeneous partitions, which are also coherent across clouds.

Attention Mechanism and Contrastive Learning. In this work, the attention mechanism and contrastive learning are utilized for co-part segmentation. The attention mechanism can be viewed as a technique that dynamically weights the input data based on their importance for feature extraction [60], while contrastive learning can be regarded as learning by comparing similar and dissimilar data [61]. Both techniques have been proven essential to many vision applications, which can be jointly [23], [62] or individually adopted [28], [63], [64]. Several works [63]–[65] employ the attention mechanism by adopting the attention blocks introduced in [66] to improve the feature representations for point cloud segmentation. The attention blocks are optimized via the cross-entropy loss which requires pointwise labels. While the works in [67]–[69] adopt contrastive learning [70], Sun *et al.* [62] jointly utilize contrastive learning and attention mechanism to learn a rich feature representation for pre-training, and improve the performance on several downstream tasks, including point cloud segmentation, by fine-tuning the pre-trained model with the corresponding labeled data. Yang *et al.* [23] also jointly employ contrastive learning and attention mechanism for *object* co-segmentation where no pointwise label is required. Their method learns point-to-foreground and point-to-background attention maps, where the latter is derived by subtracting the former from a matrix of ones. As a result, the method is limited to segmenting point clouds into two fixed classes, which is not applicable to co-*part* segmentation where the number of parts varies from category to category. AdaCoSeg [28] proposes to utilize contrastive learning for co-*part* segmentation by learning a part classifier. It exploits inter-cloud consistency by contrasting the aggregated part features across point clouds.

Compared to the above methods, our proposed PA-Net is developed with three major differences. First, our method jointly employs attention mechanism and contrastive learning without any labels and can segment semantic parts with varying numbers across shape categories. Second, instead of working at the point level as those in [23], [28], our attention mechanism operates at the superpoint level, which is more robust in capturing the part semantics, as demonstrated in Table V. Third, unlike AdaCoSeg [28] which solely enforces inter-cloud consistency, our proposed coupled attention blocks in PA-Net enable our method to enforce intra-, inter-, and paired-cloud segmentation consistency, leading to more satisfactory results, as revealed in Table V.

III. Proposed method

In this section, we first give an overview of the proposed method. Then, the SG-Net for superpoint generation and the PA-Net for part aggregation are elaborated. Finally, the implementation details are provided.

A. Method Overview

We are given a set of C point clouds $\{P_c\}_{c=1}^{C}$ of the same category, where we assume that each cloud has N points, *i.e.*, $P_c = \{\boldsymbol{p}_{cn}\}_{n=1}^{N}$, and each point $\mathbf{p}_{cn} \in \mathbb{R}^3$ is represented by its 3D coordinate. Co-part segmentation aims to segment each point cloud into R parts, with each matched part sharing similar semantic meaning across different point clouds. The value of R depends on the number of semantic parts of a particular object category and is typically pre-defined by a user or in a benchmark dataset. This is a common practice in the literature of unsupervised co-part segmentation, such as [27], [29], [30], [71], as many object categories can be naturally defined by small numbers of parts. Take categories "table," "mug," and "airplane" as examples: These categories can usually be defined into 2, 2, and 3 parts, respectively, in the benchmark dataset. To address the lack of point- and part-level annotations and the large variations among point clouds, we decompose this co-part segmentation task into two subtasks, *superpoint generation* and *part aggregation*, as depicted in Figure 2.

In the superpoint generation subtask, each point cloud P_c is segmented into $M > R$ superpoints, using the superpoint generation network (SG-Net). A point cloud feature extractor $\mathcal{E}^{\alpha}$, *e.g.*, PointNet [72], is employed to compile the point

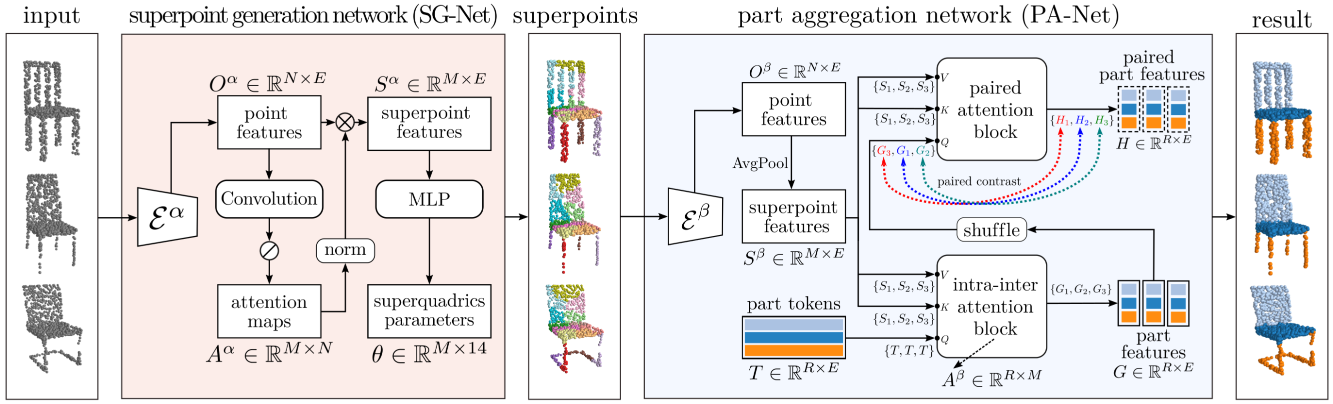
input
superpoint generation network (SG-Net)
superpoints
part aggregation network (PA-Net)
result
$O^\alpha \in \mathbb{R}^{N\times E}$
point features
$S^\alpha \in \mathbb{R}^{M\times E}$
superpoint features
Convolution
MLP
norm
attention maps
superquadrics parameters
$A^\alpha \in \mathbb{R}^{M\times N}$
$\theta \in \mathbb{R}^{M\times 14}$
$\mathcal{E}^\alpha$
$\mathcal{E}^\beta$
$O^\beta \in \mathbb{R}^{N\times E}$
point features
AvgPool
superpoint features
$S^\beta \in \mathbb{R}^{M\times E}$
part tokens
$T \in \mathbb{R}^{R\times E}$
$\{S_1, S_2, S_3\}$
V
K
Q
paired attention block
paired part features
$\{H_1, H_2, H_3\}$
$H \in \mathbb{R}^{R\times E}$
$\{G_3, G_1, G_2\}$
paired contrast
shuffle
intra-inter attention block
$\{T, T, T\}$
$\{G_1, G_2, G_3\}$
part features
$G \in \mathbb{R}^{R\times E}$
$A^\beta \in \mathbb{R}^{R\times M}$

Fig. 2. The proposed method carries out point cloud co-part segmentation with two subtasks: superpoint generation and part construction. In the first subtask, the proposed superpoint generation network (SG-Net) divides each of the input point clouds into M partitions, each called superpoint. In the second subtask, our part aggregation network (PA-Net) aggregates these superpoints to produce R semantic parts with the designated objective functions to enforce semantic consistency. In this figure, $M = 15$ superpoints and $R = 3$ parts are depicted with different colors. ⊘ denotes the transpose operation.

features $O^\alpha \in \mathbb{R}^{N\times E}$ to capture both local and global geometrical information, where E is the embedding dimension. A convolution layer with M filters is then used to map the extracted point features from embedding dimension E to M, followed by transpose operation, establishing the attention maps $A^\alpha \in \mathbb{R}^{M\times N}$ which encode the superpoint-to-point relationships. Matrix multiplication between the normalized attention maps and the point features is performed to produce superpoint features $S^\alpha \in \mathbb{R}^{M\times E}$. Through the devised losses which include superquadric fitting loss, each generated superpoint is expected to be homogeneous, namely covering points belonging to the same semantic part of the object.

In the part aggregation subtask, we merge the M superpoints into R parts. To that end, we introduce R learnable part tokens $T \in \mathbb{R}^{R\times E}$, which act as queries to aggregate part features $G \in \mathbb{R}^{R\times E}$ from superpoint feature S^β across clouds. The proposed part aggregation network (PA-Net) exploits intra-, inter-, and paired-cloud information posed in the superpoint features $S^\beta \in \mathbb{R}^{M\times E}$ and part features G via the coupled attention layer, consisting of the intra-inter attention block and the paired attention block. Superpoint generation simplifies cross-cloud mapping of R parts, since a large set of points with limited information is converted into a few superpoints with enriched features. Finally, co-part segmentation is carried out by referring to the attention maps $A^\beta \in \mathbb{R}^{R\times M}$, where each superpoint is assigned to the part with the highest attention value. Note that we learn different point features, O^α and O^β, and different superpoint features, S^α and S^β, in the two subtasks, as optimal features for superpoint generation and part aggregation are different.

B. Superpoint Generation Network

The proposed superpoint generation network divides each cloud into several disjoint superpoints by learning superpoint-to-point attention maps A^α. It is developed with two expected properties regarding the resulting superpoints: ($\mathcal{P}1$) Superpoints are homogeneous: Each of them only covers one semantic part label; ($\mathcal{P}2$) The numbers of superpoints are minimized but remains consistent for different point clouds. While keeping $\mathcal{P}1$ offers a better upper bound of co-part segmentation performance, enforcing $\mathcal{P}2$ enriches superpoint features, leading to reliable cross-cloud mapping and efficient optimization. There is a trade-off between $\mathcal{P}1$ and $\mathcal{P}2$, in which more superpoints enhance the degree of their homogeneity and vice versa.

To jointly enforce properties $\mathcal{P}1$ and $\mathcal{P}2$ on the superpoints without using any pointwise part labels, we develop three losses, including the *superquadric fitting loss* $\mathcal{L}_{fit}$, *superpoint similarity loss* $\mathcal{L}_{ss}$, and *localization loss* $\mathcal{L}_{loc}$. $\mathcal{L}_{fit}$ regularizes each superpoint to have a superquadric-like shape. While superquadrics can cover various shapes enabling each semantic part to be estimated by one or several superquadrics, superquadric fitting helps reduce the curvature in which semantic transition may occur, thus making superpoints homogeneous. $\mathcal{L}_{ss}$ encourages similar superpoints to attend to each other and can further induce homogeneous superpoints. Meanwhile, $\mathcal{L}_{loc}$ regularizes each superpoint to be concentrated in the coordinate space.

As shown in Figure 2, an extractor $\mathcal{E}^\alpha$ is applied to each of the given point clouds to obtain its point features $O^\alpha \in \mathbb{R}^{N\times E}$, where N and E are the number of points and the embedding dimension, respectively. We compute the attention maps for this cloud via

$$A^\alpha = \text{softmax}(\text{Conv1D}((O^\alpha)^T), \quad (1)$$

where the one-dimensional convolution layer with M filters of size 1 is used to fuse the embedding features, resulting in $A^\alpha \in \mathbb{R}^{M\times N}$ where M is set to the pre-defined number of superpoints. The softmax operation is applied along the superpoint dimension so that every point is softly assigned to a certain superpoint. Namely, the n-th point is assigned to superpoint m^* if

$$m^* = \underset{m}{\text{argmax}}(A^\alpha_{m,n}). \quad (2)$$

We model each superquadric with a set of 12 continuous parameters, including three parameters for the size, two for the shape, three for translation, and four for rotation. The details of these parameters are given in the supplementary material. To learn the superquadric parameters θ, we first compute superpoint features $S^\alpha \in \mathbb{R}^{M\times E}$ via

$$S^\alpha = \text{norm}(A^\alpha)O^\alpha, \quad (3)$$

where norm denotes L_1 normalization for each row of A^α, and superpoint features S^α are convex combinations of point features O^α. Then, superquadric parameters θ can be learned by applying multi-layer perceptron (MLP) to the superpoint features, *i.e.*,

$$\theta = \text{MLP}(S^\alpha), \tag{4}$$

where $\theta \in \mathbb{R}^{M\times 12}$. We discuss the three losses developed to derive our superpoint generation network as follows.

Superquadric fitting loss. After getting the point-superpoint assignment in Eq. 2 and superquadrics parameters in Eq. 4, this loss $\mathcal{L}_{fit}$ is computed for the superpoints and their associated superquadrics. We follow the common practice [55], [57]: Point-sampling is applied to the superquadric of the m-th superpoint, resulting in $D_m \in \mathbb{R}^{L\times 3}$, where L is the number of sampled points, and the fitting distance can be computed between these sampled points and the superpoints. We choose Chamfer distance and formulate this fitting loss in two directions, which are the distance from the superpoint to the superquadric and that from the superquadric to the superpoint, denoted by $\mathcal{L}_{fit}^{P\rightarrow D}$ and $\mathcal{L}_{fit}^{D\rightarrow P}$, respectively. While the existing works express the total fitting distance in Gaussian [73] and Bernoulli [55] distributions between superquadrics and the whole points, we formulate it in an attention-based fitting.

Given points sampled from M superquadrics, $D = \{D_1, D_2, ..., D_M\}$, $\mathcal{L}_{fit}^{P\rightarrow D}$ is defined by

$$\mathcal{L}_{fit}^{P\rightarrow D} = \frac{1}{N\times M}\sum_{n=1}^{N}\sum_{m=1}^{M} A_{m,n}^{\alpha}\triangle_{m,n}, \tag{5}$$

where

$$\triangle_{m,n} = \min_{i=1,...,L} ||\mathcal{F}_m(\boldsymbol{p}_n) - D_{m,i}||_2 \tag{6}$$

denotes the minimal distance from the n-th point $\boldsymbol{p}_n$ of cloud P to m-th superquadric, and $\mathcal{F}_m$ is a function that transforms a point to the coordinate system of the m-th superquadric by its rotation and translation parameters. The loss in Eq. 5 is computed for every point $\boldsymbol{p}_n$ in Eq. 6 and all superquadrics by using the memberships of $\boldsymbol{p}_n$ to all superpoints as the weights.

Thus, minimizing this loss encourages each point to attend to the closest superquadric in maps A^α.

In the other direction, the fitting loss $\mathcal{L}_{fit}^{D\rightarrow P}$ is specified as

$$\mathcal{L}_{fit}^{D\rightarrow P} = \frac{1}{M\times L}\sum_{m=1}^{M}\sum_{l=1}^{L}\triangle_{m,l}, \tag{7}$$

where

$$\triangle_{m,l} = \min_{n\in\mathcal{I}} ||D_{m,l} - \mathcal{F}_m(\boldsymbol{p}_n)||_2 \tag{8}$$

denotes the minimal distance from the l-th sampled point of the m-th superquadric to the m-th superpoint. Here, $\mathcal{I}$ is the set of point indices assigned to the m-th superpoint, given in Eq. 2. This loss enforces superquadrics to stay close to their superpoints.

Superpoint similarity loss. While the superquadric fitting loss regularizes the shape of a superpoint, this loss makes the points of a superpoint similar to each other and is defined by

$$\mathcal{L}_{ss} = \sum_{m=1}^{M}\sum_{n=1}^{N} A_{m,n}^{\alpha}||S_m^\alpha - O_n^\alpha||_2. \tag{9}$$

Minimizing Eq. 9 enforces the similarity of each point and its associated superpoint, which then enhances the feature consensus of points belonging to the same superpoint.

Localization loss. We regularize the learned superpoints by minimizing the distance of each point $\boldsymbol{p}_n$ and the representative superpoint centroids $Z \in \mathbb{R}^{M\times 3}$, defined by

$$Z = \text{norm}(A^\alpha)P, \tag{10}$$

and this loss is formulated as follows:

$$\mathcal{L}_{loc} = \sum_{m=1}^{M}\sum_{n=1}^{N} A_{m,n}^{\alpha}||Z_m - \boldsymbol{p}_n||_2. \tag{11}$$

To prevent the attention maps A^α from assigning all points to one or few superpoints, we develop a balancing loss based on attention variance, namely

$$\mathcal{L}_{balance}^{sp} = \frac{1}{M}\sum_{m=1}^{M}(a_m^\alpha - \frac{1}{M}\sum_{m'=1}^{M} a_{m'}^{\alpha})^2, \tag{12}$$

with $a_m^\alpha = \sum_{n=1}^{N}(A_{m,n}^{\alpha})$ denotes the total attention exerted by the m-th superpoint to all points.

C. Part Aggregation Network

The proposed part aggregation network aims to merge M superpoints to yield R semantic parts. As shown in Figure 2, it learns a superpoint-to-part assignment, recorded in the attention maps A^β, to accomplish point cloud co-part segmentation. Here, R is a pre-defined number of parts. The part aggregation network is a stack of our proposed coupled attention blocks, each of which is composed of the intra-inter attention block and the paired attention block. We develop three loss functions to trigger semantic consistency in the segmentation result, including the intra-cloud similarity loss $\mathcal{L}_{intra}$, inter-cloud contrastive loss $\mathcal{L}_{inter}$, and paired-cloud contrastive loss $\mathcal{L}_{paired}$. The losses exploit geometrical features in a single cloud, all clouds, and paired clouds, respectively.

The part aggregation network accepts the same C point clouds $\{P_c\}_{c=1}^{C}$ and their superpoint partitions $\{Y_c\}_{c=1}^{C}$, where $Y_c \in \mathbb{R}^N$, inferred via Eq. 2, gives the superpoint labels of the N points in cloud P_c. Another learnable extractor $\mathcal{E}^\beta$ is applied to each point cloud producing its point features $O^\beta \in \mathbb{R}^{N\times E}$. With the superpoint labels Y, the superpoint features $S^\beta \in \mathbb{R}^{M\times E}$ are computed by applying average pooling for each superpoint. Inspired by DETR [74], which uses *object tokens* to aggregate object elements in a single image, we extend this idea to *multiple*, *3D* point clouds since the desired parts are required to be coherent and matched across different clouds. Specifically, R *part tokens*, $T \in \mathbb{R}^{R\times E}$, are defined as learnable parameters and used to query part features $G \in \mathbb{R}^{R\times E}$ via the intra-inter attention block:

$$\begin{aligned} G(Q,K,V) &= \text{norm}\left(\text{softmax}\left(\tfrac{QK^T}{\sqrt{E}}\right)\right)V \\ &= \text{norm}\left(A^\beta\right)V \end{aligned} \tag{13}$$

where $Q = TQ_w$, $K = S^\beta K_w$, and $V = S^\beta V_w$ are the queries, keys, and values of this attention block, respectively, and Q_w, K_w and V_w are three learnable matrices for linear embedding. The softmax operation is applied along the first

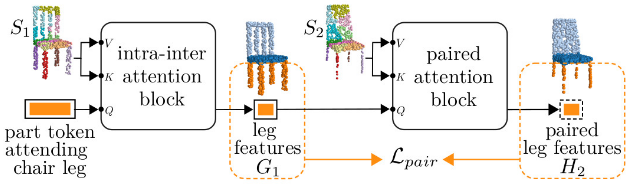
S_1
V
K
Q
intra-inter attention block
part token attending chair leg
leg features G_1
S_2
V
K
Q
paired attention block
paired leg features H_2
$\mathcal{L}_{pair}$

Fig. 3. Illustration of the proposed paired-cloud contrastive loss. See texts for details.

dimension (part dimension) followed by normalization along the second dimension (superpoint dimension). The softmax operation softly makes each superpoint attend to a certain part. The m-th superpoint is assigned to part r^* if

$$r^* = \operatorname*{argmax}_{r}(A^{\beta}_{r,m}). \tag{14}$$

Part features G produced via Eq. 13 are linear combinations of superpoint features, and those associated with the same part token are the matched parts across clouds, depicted with the same color in the rightmost column of Figure 2. We describe the developed loss functions to derive PA-Net as follows.

Intra-cloud similarity loss. This loss enforces the similarity between a part token and all superpoints that this part token attends to, and is formulated as

$$\mathcal{L}_{intra} = \sum_{r=i}^{R}\sum_{m=i}^{M} A^{\beta}_{r,m}||G_r - S_m||_2. \tag{15}$$

It follows that those attended superpoints belonging to a part are expected to be similar to each other. For example, in the category of chair, one of the part tokens will attend to those superpoints from leg parts in which the superpoints are similar.

Inter-cloud contrastive loss. This loss is devised to trigger cross-cloud semantic consistency. Specifically, part features associated with the same semantic part token across clouds are encouraged to be similar, and part features attending to different part tokens can be further differentiated. To achieve this, a pairwise infoNCE contrastive loss [75] is adopted across different clouds:

$$\mathcal{L}_{inter} = \sum_{i,j=1,i\neq j}^{C}\sum_{r=1}^{R} -\log\frac{\exp(\langle G_{i,r}, G_{j,r}\rangle)}{\sum_{k,l\in\mathcal{I}(i,r)}\exp(\langle G_{i,r}, G_{k,l}\rangle)} \tag{16}$$

where $G_{i,r}$ is the part features of cloud P_i that is associated with the r-th part token, serving as the anchor, and $G_{j,r}$ as the positive pair. $\langle\cdot,\cdot\rangle$ and $\mathcal{I}(i,r)$ denote cosine similarity and a function generating positive and negative pairs indices of anchor index (i,r). In this case, negative pairs are all part features associated with the different part token, *i.e.*, $r \neq l$, from both intra- and inter-cloud in the set.

Paired-cloud contrastive loss. To further enhance semantic consistency on the segmentation result, we formulate the paired-cloud contrastive loss that encourages semantic consistency for cloud pairs. This loss is designed to ensure that each part of one point cloud can be well reconstructed from superpoints of another point cloud. To demonstrate the idea, we provide an illustration in Figure 3 of applying this loss to the leg parts of a chair pair. Let chair 1 and chair 2 be that pair. We first extract the leg features of chair 1, G_1, from the superpoint features of chair 1, S_1, using a particular token that attends leg parts. We then use G_1 to query its paired leg features, H_2, from the superpoint features of chair 2, S_2. Based on the attention mechanism, H_2 is a linear combination of S_2, queried using G_1. We apply this loss to these leg features, G_1 and H_2, to promote their similarity. In this way, the leg feature of chair 1 is encouraged to be reconstructed using the leg feature of chair 2.

To define the pairing, we shuffle part features G, before being employed as queries in the paired attention block. Specifically, we use random rolling to shuffle part features G to avoid self-pairings. The loss is expressed as

$$\mathcal{L}_{pair} = \sum_{c=1}^{C}\sum_{r=1}^{R} -\log\frac{\exp(\langle\Gamma(G)_{c,r}, H_{c,r}\rangle)}{\sum_{r'=1}^{R}\exp(\langle\Gamma(G)_{c,r}, H_{c,r'}\rangle)}, \tag{17}$$

where Γ denotes the aforementioned shuffle operation.

We also perform balancing loss to the part aggregation network to prevent it from assigning superpoints only to certain parts, by modifying Eq. 12 to

$$\mathcal{L}^{part}_{balance} = \frac{1}{R}\sum_{r=1}^{R}(a^{\beta}_{r} - \frac{1}{R}\sum_{r'=1}^{R} a^{\beta}_{r'})^2. \tag{18}$$

D. Implementation Details

The proposed method which includes SG-Net and PA-Net are trained end-to-end with 100 epochs via the Adam optimizer with a learning rate and batch size of 0.001 and 32, respectively, where PointNet-like [50], [72], [76] is adopted for extractors $\mathcal{E}^{\alpha}$ and $\mathcal{E}^{\beta}$ due to its simplicity. The embedding dimensions, E, in both SG-Net and PA-Net, including the attention blocks, are all set to 256, wherein two layers of MLP in SG-Net, consisting of a linear layer and an activation function, are employed. We set the weights of all the losses to 1.0, except the balancing losses $\mathcal{L}^{sp}_{balance}$ and $\mathcal{L}^{part}_{balance}$. As suggested in the related work [55], we use a small weight for the balancing losses. That is, we set the weights of $\mathcal{L}^{sp}_{balance}$ and $\mathcal{L}^{part}_{balance}$ to 0.001.

We uniformly sample 1,024 points on the mesh faces according to the face areas and then normalize them into a unit sphere. For each cloud, SG-Net divides the cloud into $M = 15$ superpoints, and in each superpoint, we sample $L = 160$ points of the corresponding superquadric to compute the fitting loss.

IV. Experimental results

In this section, we evaluate our method of co-part segmentation with the ShapeNet part [37] and COSEG [38] datasets. Following the prior works [27], [28], Intersection over Union (IoU) and Rand Index (RI) [77] are adopted as the evaluation metrics for the former and the latter datasets, respectively. We also report the superpoint performance in Overall Accuracy (OA), following SPG [78]. While larger values are preferred in the IoU and OA, smaller ones are better in RI.

TABLE I
UNSUPERVISED SEGMENTATION COMPARISON WITH BENCHMARK METHODS ON THE SHAPENET PART DATASET, REPORTED IN IOU (%) MEASURED AGAINST GROUND TRUTH PARTS. NUMBERS BESIDE EACH CATEGORY NAME INDICATE THE NUMBER OF SEMANTIC PARTS.

method	Airplane (3)	Bag (2)	Cap (2)	Chair (3)	Chair (4)	Guitar (3)	Mug (2)	Table (2)
BAE-Net [27]	61.1	82.5	87.3	65.5	83.7	72.3	93.4	78.7
AdaCoSeg [28]	65.3	78.5	84.2	72.8	79.2	71.6	86.7	77.3
Ours	**70.4**	**84.3**	**88.4**	**77.0**	**85.2**	**75.2**	**93.8**	**80.2**

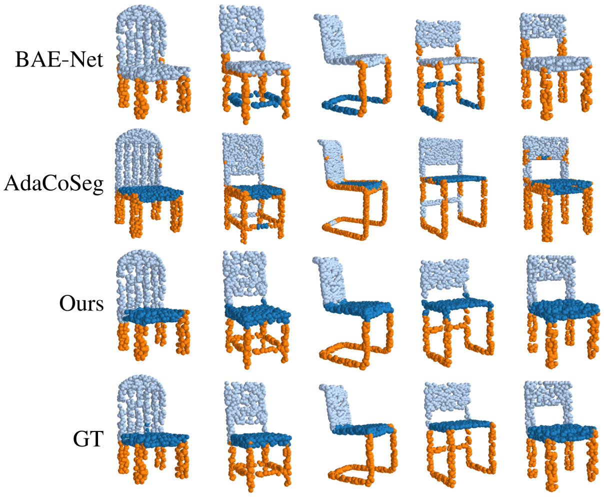

Fig. 4. Visualization of some unsupervised segmentation results along with the ground truth (GT) on the ShapeNet part dataset.

TABLE II
UNSUPERVISED SEGMENTATION RESULTS ON THE COSEG DATASET, REPORTED IN RAND INDEX MEASURED AGAINST GROUND TRUTH PARTS.

method	Chair	Lamp	Vase	Guitar
Shu *et al.* [71]	0.076	0.069	0.198	0.041
Hu *et al.* [29]	0.121	0.103	0.230	0.037
Sidi *et al.* [30]	0.135	0.092	0.102	0.081
BAE-Net [27]	0.124	0.069	0.156	0.072
AdaCoSeg [28]	0.085	0.084	0.211	0.065
Ours	**0.062**	**0.059**	**0.092**	**0.034**

TABLE III
SEGMENTATION COMPARISON ON THE SHAPENET PART DATASET IN A MORE RELAXED SETTING WHERE PART LABELS ARE USED TO GROUP SUPERPOINTS OR PRIMITIVES, REPORTED IN IOU (%)

method	Airplane	Chair	Table
BSP-Net [58]	72.1	66.9	85.9
ProGRIP [59]	75.7	75.2	85.7
Ours	**81.2**	**85.4**	**86.3**

A. Co-part Segmentation

We first compare the proposed method in an unsupervised setting on the ShapeNet part dataset with the existing methods, BAE-Net [27] and AdaCoSeg [28], with the latter being an unsupervised version obtained by turning off the part-prior network that is trained using binary part labels from the ComplementMe dataset [32]. In unsupervised learning, since the label-to-semantic mapping between segmentation result and ground truth may be different, each predicted part will be mapped to the majority label that appears in the corresponding points of the ground truth data, which is also adopted in the prior work [27].

As demonstrated in Table I, our method outperforms the benchmark methods in all categories. Specifically, in the airplane and chair (3 parts) categories, the proposed method achieves significant improvement margins, which are more than 5%, while obtaining notable margins in the remaining categories. Such results demonstrate that the proposed method which exploits intra-, inter-, and paired-cloud geometrical information is effective in attaining the goal of the co-part segmentation task.

We further visualize our segmentation results along with those from the benchmark methods, as shown in Figure 4. While there is no explicit loss to encourage consistency in BAE-Net, *i.e.*, only minimizing reconstruction loss in each individual cloud, AdaCoSeg only enforces inter-cloud consistency. As a result, BAE-Net may assign two different semantics into one segment, *e.g.*, chair back and seat, or produce an incomplete semantic as one segment, and AdaCoSeg may produce unsatisfactory semantic overlap in their segmentation result. In contrast, it is readily observable from Figure 4 (3rd row) that favorable segmentation results are obtained with our method, compared with the ground truth (4th row), as intra-, inter-, and paired-cloud geometrical information are fully exploited to explicitly enforce semantic consistency and is successfully observed in the segmentation.

Segmentation results on different categories are depicted in Figure 5. In each category, the left side shows the superpoints generated by SG-Net, while the right side shows the aggregated co-part segmentation results by PA-Net. As shown in the figure, semantic consistency is successfully achieved by our method. While our method segments the chair category into leg, seat, and back, it segments the airplane category into body, wing, and tail. Appropriate segmentations are also observed in the guitar, table, mug, and cap categories.

To further demonstrate the effectiveness of our method, various segmentation results obtained for the COSEG dataset, reported in Rand Index (RI) measured against ground truth parts, are shown in Table II. In addition to the results obtained by running the official code of BAE-Net and AdaCoSeg, RI values reported in Shu *et al.* [71], Hu *et al.* [29], and Sidi *et al.* [30] are also included. For the COSEG dataset, which has a much smaller sample number per category compared with the ShapeNet part dataset, our method also consistently outperforms other benchmark methods.

Recent researches [58], [59] that mainly work for primitive-based shape abstraction also report their performance on shape part segmentation. Since these works more focus on shape decomposition where each part could be decomposed into several primitives, they require part labels to group the decomposed primitives into the corresponding semantic parts.

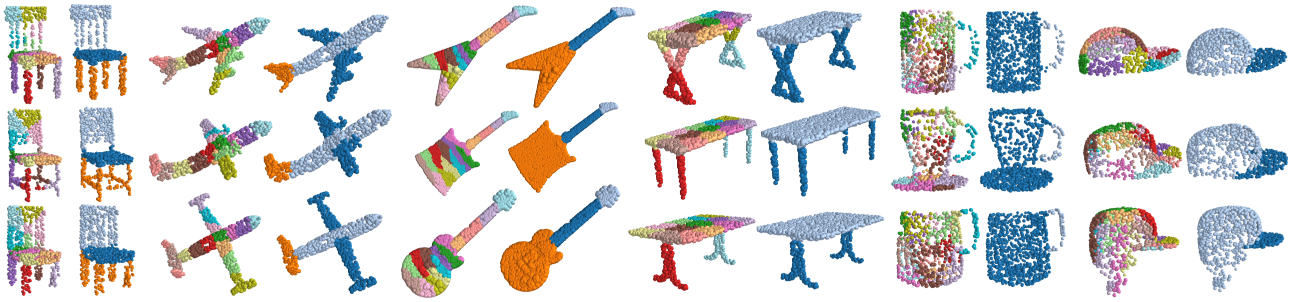

Fig. 5. Unsupervised segmentation results obtained with the proposed method on the ShapeNet part dataset. In each category, the left side shows the superpoints and the right side shows the aggregation of them into a few parts. Each superpoint and part are drawn in different colors.

TABLE IV
SUPERPOINT COMPARISON WITH THE BENCHMARK METHOD ON THE SHAPENET PART DATASET, REPORTED IN OVERALL ACCURACY (OA) MEASURED AGAINST THE GROUND TRUTH.

Methods	Airplane	Bag	Chair	Guitar	Mug	Table
SPG [78]	84.3	86.1	86.5	88.5	92.6	86.2
Ours	**87.4**	**88.9**	**91.4**	**92.1**	**95.2**	**90.3**

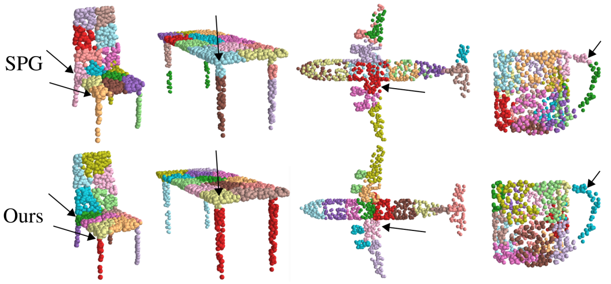

Fig. 6. Superpoint results obtained with the benchmark method (top) and our method (bottom) on the ShapeNet part dataset. The arrows highlight the differences between the two methods.

We then compare our method with theirs by following their more relaxed setting, where part labels are utilized to group their primitives or our superpoints, namely each primitive or superpoint is assigned a part label by *voting*. Their methods report the IoU scores on three categories of the ShapeNet part dataset. We follow them to report these scores, presented in Table III, and our method consistently achieves better IoU scores. Such results suggest that our method which divides each point cloud of a 3D shape into homogeneous partitions can semantically better decompose the shapes than those using primitive-based approaches which focus on reconstructing the original shapes.

B. Superpoint Generation

We report the superpoint performance, as shown in Table IV, compared with the previous work that shares similar motivation on avoiding pointwise labeling, *i.e.*, SPG [78]. While SPG uses handcrafted features and casts the task as a partitioning problem that is solved by l_0-cut pursuit algorithm [54], our SG-Net learns point features and employs the devised losses to divide point cloud into semantically homogeneous partitions.

As summarized in Table IV, it is readily observable that SG-Net achieves notable improvements compared to SPG, which demonstrates that our method can produce overall more semantically homogeneous superpoints. Some superpoint results are depicted in Figure 6, each shown in different colors. The handcrafted point features employed by SPG which mainly include linearity, planarity, and scattering within a spherical radius to capture the 1D, 2D, and 3D characteristics of the neighboring points may be sub-optimal in describing the points with respect to their semantic parts. On the other hand, our SG-Net extracts learnable features that minimize the devised losses and generates more semantically homogeneous superpoints within the boundary of a part, as indicated with (black) arrows in Figure 6.

TABLE V
ABLATION STUDIES OF THE SYSTEM COMPONENTS IN THE PROPOSED METHOD, ON THE SHAPENET PART DATASET, REPORTED IN MIOU.

Setup	Extractors	Superpoints generator	PA-Net			mIoU (%)
			$\mathcal{L}_{intra}$	$\mathcal{L}_{inter}$	$\mathcal{L}_{paired}$	
1	independent	SG-Net	✓			71.7
2	independent	SG-Net		✓	✓	62.4
3	independent	SG-Net	✓	✓		74.6
4	independent	SG-Net	✓		✓	74.8
5	independent	SG-Net	✓	✓	✓	**77.0**
6	independent	SG-Net	✓	✓	modified	74.5
7	shared	SG-Net	✓	✓	✓	74.1
8	independent	SPG [78]	✓	✓	✓	72.7
9	independent	-	✓	✓	✓	68.6

C. Ablation Study and Analysis

The proposed method exploits intra-, inter-, and paired-cloud geometrical information posed in the superpoints via minimizing the devised losses $\mathcal{L}_{intra}$, $\mathcal{L}_{inter}$ and $\mathcal{L}_{paired}$, respectively. While $\mathcal{L}_{intra}$ only considers individual clouds, the other two consider multi-clouds, *i.e.*, all clouds in the batch for $\mathcal{L}_{inter}$ and pairs of clouds in the batch for $\mathcal{L}_{paired}$. We report the impact of using different combinations of the losses, on the mIoU performance, as summarized in Table V.

In Setup 1, with only $\mathcal{L}_{intra}$ considered for individual cloud, an mIoU of 71.7% can be achieved. While $\mathcal{L}_{intra}$ encourages superpoints to attend parts that share similarities to them in each individual cloud, some superpoints may share geometrical similarities to the part of different semantics which can cause unexpected segmentation, *e.g.*, two superpoints of the chair back, shown in the first row of Setup 1 in Figure 7, are segmented as leg part as they are geometrically similar. In Setup 2, we employ $\mathcal{L}_{inter}$ and $\mathcal{L}_{paired}$ with both considering multi-clouds but attain an unfavorable mIoU performance of

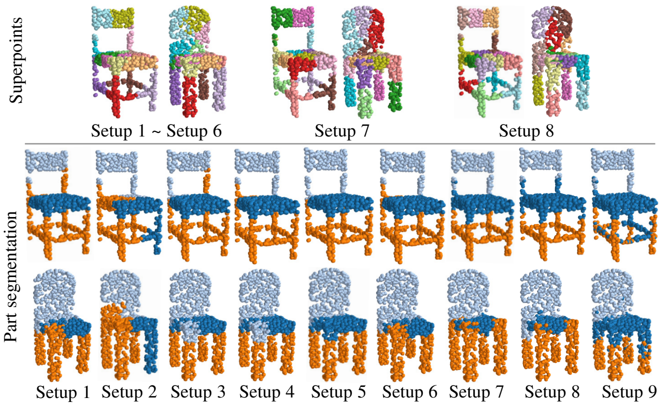

Fig. 7. Visualization of the segmentation results in different setups of ablation studies.

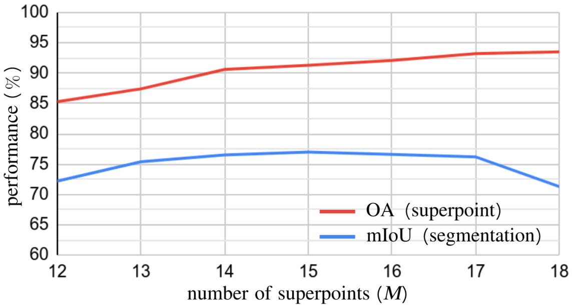

Fig. 8. The impact of different superpoint numbers, M, to OA and mIoU on ShapeNet part dataset.

62.4%. This is because the combination of $\mathcal{L}_{inter}$ and $\mathcal{L}_{paired}$ may produce consistent (but undesirable) segmentation across multiple clouds. For example, one can see that consistent segmentation of seat-leg parts into left-diagonal and right-diagonal in the 3rd column of Figure 7 is definitely not suitable.

In Setup 3 and Setup 4, we combine losses that consider both individual as well as multi-clouds. Specifically, Setup 3 employs $\mathcal{L}_{intra}$ and $\mathcal{L}_{inter}$, while Setup 4 combines $\mathcal{L}_{intra}$ and $\mathcal{L}_{paired}$. Note that both setups actually produce notable mIoUs of more than 74%. In Setup 5, we further combine all three losses and achieve the best mIoU performance of 77.0%. Such results confirm that the three losses are complementary and none of them is redundant. In other words, while $\mathcal{L}_{intra}$ is in charge of triggering semantically meaningful segmentation in each individual cloud, $\mathcal{L}_{inter}$ and $\mathcal{L}_{paired}$ further help to induce the semantic consistency across all clouds in the batch and between each pair of clouds, respectively.

We modify $\mathcal{L}_{paired}$ in Setup 6 by using the part tokens, in place of part features, as queries, which also consider cloud pairs in the batch. A lower mIoU of 74.5% in such a setup indicates that while part tokens are shared across clouds, part features that are unique for every cloud as queries can trigger better semantic consistency in cloud pairs. In addition, we investigate the impact of using two independent extractors or a shared extractor employed in SG-Net and PA-Net, as represented in Setups 5 and 7, respectively. Since the features in SG-Net and PA-Net are optimized with different objectives, using independent extractors facilitates the networks to learn the optimal features for different objectives, resulting in a superior result.

TABLE VI
ACCURACY IMPROVEMENTS OF CLASSIFICATION TRAINED WITH 20%, 50%, AND 100% OF THE MODELNET40 DATASET.

Method	Rate 20%	Rate 50%	Rate 100%
DGCNN [79]	87.0	90.3	91.5
DGCNN + Point MixSwap [36]	90.1	91.3	92.3
DGCNN + Ours	**91.4**	**92.1**	**93.0**
GDANet [12]	89.4	91.7	93.8
GDANet + Point MixSwap [36]	91.4	92.9	94.0
GDANet + Ours	**92.5**	**93.7**	**94.5**

TABLE VII
PERFORMANCE IMPROVEMENTS (MIOU) OF SEMI-SUPERVISED PART SEGMENTATION ON THE SHAPENET PART DATASET.

Method	1% labeled	5% labeled
Gadelha *et al.* [80]	75.7	79.7
Ours	**78.0**	**81.6**

We further investigate the impact of employing superpoints in the proposed method. In Setup 8, we utilize SPG [78] to generate superpoints, while in Setup 9, no superpoint generator is used, namely the attention blocks in PA-Net directly work with point features, instead of superpoint features. Compared to Setup 5, which uses full components in the proposed method, Setups 8 and 9 yield substantially lower mIoU scores of 72.7% and 68.6%, respectively. The reason for this performance gap is that in Setup 8, the generated superpoints by SPG are less semantically homogeneous than those of SG-Net utilized in Setup 5, as can be observed in the upper part of Figure 7. On the other hand, Setup 9 directly works with point features, which may not be sufficiently robust to capture the part semantics, resulting in noisy segmentation results (see the rightmost part of Figure 7).

We then analyze the impact of the superpoint number, M, with respect to OA and mIoU performances of SG-Net and PA-Net, respectively, as shown in Figure 8. Note that a smaller value of M produces more coarse superpoints where each superpoint tends to be less semantically homogeneous resulting in low OA. Since OA can be considered as the upper bound of the segmentation performance, a smaller value of M leads to lower mIoU. As for larger values of M, SG-Net generates more semantically homogeneous superpoints with higher OA. However, a larger value of M gives a more challenging task for PA-Net, which may result in lower mIoU because (i) the resultant superpoints are of smaller sizes on average and thus bring limited geometrical information and (ii) the increasing number of superpoints increases the complexity of assigning them to appropriate semantic parts. With this trade-off, the proposed method achieves favorable mIoU performances in a broad range of superpoint numbers, namely from $M = 13$ to $M = 17$, with its best in $M = 15$, as shown in Figure 8.

D. Downstream Task Applications

We demonstrate the advantage of the proposed method on some downstream tasks, including data augmentation for point

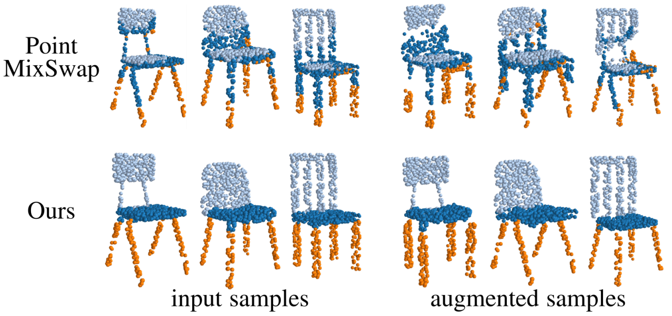

Fig. 9. Three augmented samples on the right are generated by swapping the semantic parts of three input samples on the left. The upper and lower figures show the input and augmented samples of PointMixswap [36] and our method, respectively.

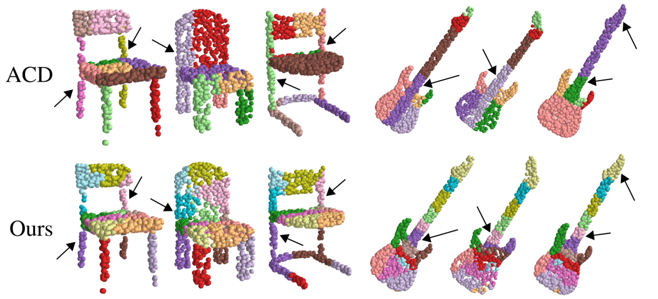

Fig. 10. Decomposed point clouds obtained via ACD [81] (top) and our method (bottom) that are utilized to perform semi-supervised part segmentation reported in Table VII. The arrows indicate major differences between the two methods.

cloud classification (DAPC) and semi-supervised part segmentation (SSPS). In DAPC, we follow Point MixSwap [36], which generates augmented data by swapping the semantic parts based on their attentional part decomposition and replacing its part decomposition with our co-part segmentation results. While in SSPS, we compare with previous work, Gadelha *et al*. [80], which decomposes a point cloud into partitions using ACD [81] and applies contrastive learning based on the partitions. The work reports the performance in mIoU trained on 1% and 5% labeled data. We use the same training setting but replace the ACD with our superpoints. From Tables VI and VII, our method outperforms and improves the performances of both downstream tasks. A justification for such improvements is that better co-part segmentation and superpoints can be provided by our method than those generated by Point MixSwap and ACD, as shown in Figures 9 and 10, respectively. In Figure 10, the arrows highlight the different partition results of ACD and our method. Our method produces more semantically homogeneous partitions than ACD. For example, in the second chair, the arrow indicates that ACD's partition mixes the leg and back parts, while our partition only contains the back part.

E. Computational Comparison

To compare the computational resources needed in each unsupervised method, we report the inference time, GFLOPs (giga floating point operations per second), and the number of parameters of the evaluated methods in Table VIII. Note that based on our design choice, the proposed method runs around 9 and 11 times faster with 13 and 3 times fewer parameters than BAE-Net and AdaCoSeg, respectively, and also consumes significantly fewer FLOPs. In the inference time, our method accepts point clouds of N points with lightweight modules, *e.g.*, PointNet-based extractors and the intra-inter attention block which maps M superpoints to R parts, with $M \ll N$ and $R \ll N$. Meanwhile, BAE-Net and AdaCoSeg utilize fully connected layers and Multi-resolution Grouping (MRG) [76], respectively, that are computationally expensive. In addition, besides accepting point clouds, BAE-Net also requires $64 \times 64 \times 64$ voxelized features which further increases its computational burden.

TABLE VIII
INFERENCE TIME AND NUMBER OF PARAMETERS COMPARISON WITH UNSUPERVISED BENCHMARK METHODS, RUN IN NIVIDA RTX 3090.

Method	Inference time (ms)	GFLOPs	Numb. of params.
BAE-Net [27]	222	4.1	∼5,250K
AdaCoSeg [28]	260	4.9	∼1,099K
Ours	24	0.9	∼393K

V. Conclusion

We present unsupervised co-part segmentation for point clouds by formulating the task into two subtasks, including superpoint generation and part aggregation. The proposed method divides each individual cloud into superpoints in the first subtask and these superpoints can be further aggregated into a few parts in the second subtask by exploiting intra-, inter-, and paired-cloud geometrical information. Evaluated on two common part segmentation datasets, the proposed method consistently outperforms the existing methods. Furthermore, it also demonstrates the ability in facilitating several downstream tasks, including semi-supervised part segmentation and data augmentation for shape classification.

References

[1] P.-J. Duh, Y.-C. Sung, L.-Y. F. Chiang, Y.-J. Chang, and K.-W. Chen, "V-eye: A vision-based navigation system for the visually impaired," *IEEE Transactions on Multimedia*, pp. 1567–1580, 2020.

[2] A. A. Khan, J. Shao, Y. Rao, L. She, and H. T. Shen, "Lrdnet: Lightweight lidar aided cascaded feature pools for free road space detection," *IEEE Transactions on Multimedia*, 2022.

[3] M. Wen, Y. Dai, T. Chen, C. Zhao, J. Zhang, and D. Wang, "A robust sidewalk navigation method for mobile robots based on sparse semantic point cloud," in *Proceedings of International Conference on Intelligent Robots and Systems*, 2022, pp. 7841–7846.

[4] N. N. Kaashki, P. Hu, and A. Munteanu, "Anet: A deep neural network for automatic 3d anthropometric measurement extraction," *IEEE Transactions on Multimedia*, pp. 831–844, 2021.

[5] D. Liu, Y. Tian, Y. Zhang, J. Gelernter, and X. Wang, "Heterogeneous data fusion and loss function design for tooth point cloud segmentation," *Neural Computing and Applications*, pp. 17 371–17 380, 2022.

[6] F. G. Zanjani, D. A. Moin, B. Verheij, F. Claessen, T. Cherici, T. Tan *et al.*, "Deep learning approach to semantic segmentation in 3d point cloud intra-oral scans of teeth," in *Proceedings of Medical Imaging with Deep Learning*, 2019, pp. 557–571.

[7] Z. Fan, H. Liu, J. He, M. Zhang, and X. Du, "Mpdnet: A 3d missing part detection network based on point cloud segmentation," in *Proceedings of International Conference on Acoustics, Speech and Signal Processing*, 2021, pp. 1810–1814.

[8] M. Makuch and P. Gawronek, "3d point cloud analysis for damage detection on hyperboloid cooling tower shells," *Remote Sensing*, p. 1542, 2020.

[9] F. G. Zanjani, D. A. Moin, F. Claessen, T. Cherici, S. Parinussa, A. Pourtaherian, S. Zinger, and P. H. de With, "Mask-mcnet: Instance segmentation in 3d point cloud of intra-oral scans," in *Proceedings of International Conference on Medical Image Computing and Computer Assisted Intervention*, 2019, pp. 128–136.

[10] S. Kim and D. C. Alexander, "Agcn: Adversarial graph convolutional network for 3d point cloud segmentation," in *Proceedings of British Machine Vision Conference*, 2021, pp. 1–10.

[11] A. Nekrasov, J. Schult, O. Litany, B. Leibe, and F. Engelmann, "Mix3d: Out-of-context data augmentation for 3d scenes," in *Proceedings of International Conference on 3D Vision*, 2021, pp. 116–125.

[12] M. Xu, J. Zhang, Z. Zhou, M. Xu, X. Qi, and Y. Qiao, "Learning geometry-disentangled representation for complementary understanding of 3d object point cloud," in *Proceedings of the AAAI Conference on Artificial Intelligence*, 2021, pp. 3056–3064.

[13] H. Zhao, L. Jiang, J. Jia, P. H. Torr, and V. Koltun, "Point transformer," in *Proceedings of International Conference on Computer Vision*, 2021, pp. 16 259–16 268.

[14] I. Armeni, O. Sener, A. R. Zamir, H. Jiang, I. Brilakis, M. Fischer, and S. Savarese, "3d semantic parsing of large-scale indoor spaces," in *Proceedings of Conference on Computer Vision and Pattern Recognition*, 2016, pp. 1534–1543.

[15] J. Behley, M. Garbade, A. Milioto, J. Quenzel, S. Behnke, C. Stachniss, and J. Gall, "Semantickitti: A dataset for semantic scene understanding of lidar sequences," in *Proceedings of International Conference on Computer Vision*, 2019, pp. 9297–9307.

[16] A. Dai, A. X. Chang, M. Savva, M. Halber, T. Funkhouser, and M. Nießner, "Scannet: Richly-annotated 3d reconstructions of indoor scenes," in *Proceedings of Conference on Computer Vision and Pattern Recognition*, 2017, pp. 5828–5839.

[17] M. A. Uy, Q.-H. Pham, B.-S. Hua, T. Nguyen, and S.-K. Yeung, "Revisiting point cloud classification: A new benchmark dataset and classification model on real-world data," in *Proceedings of Conference on Computer Vision and Pattern Recognition*, 2019, pp. 1588–1597.

[18] Z. Wu, S. Song, A. Khosla, F. Yu, L. Zhang, X. Tang, and J. Xiao, "3d shapenets: A deep representation for volumetric shapes," in *Proceedings of Conference on Computer Vision and Pattern Recognition*, 2015, pp. 1912–1920.

[19] Z. Lu, H. Xu, and G. Liu, "A survey of object co-segmentation," *IEEE Access*, pp. 62 875–62 893, 2019.

[20] T. Li, K. Zhang, S. Shen, B. Liu, Q. Liu, and Z. Li, "Image co-saliency detection and instance co-segmentation using attention graph clustering based graph convolutional network," *IEEE Transactions on Multimedia*, pp. 492–505, 2021.

[21] B. Jiang, X. Jiang, J. Tang, and B. Luo, "Co-saliency detection via a general optimization model and adaptive graph learning," *IEEE Transactions on Multimedia*, pp. 3193–3202, 2020.

[22] C. Rother, T. Minka, A. Blake, and V. Kolmogorov, "Cosegmentation of image pairs by histogram matching-incorporating a global constraint into mrfs," in *Proceedings of Conference on Computer Vision and Pattern Recognition*, 2006, pp. 993–1000.

[23] C.-K. Yang, Y.-Y. Chuang, and Y.-Y. Lin, "Unsupervised point cloud object co-segmentation by co-contrastive learning and mutual attention sampling," in *Proceedings of Conference on Computer Vision and Pattern Recognition*, 2021, pp. 7335–7344.

[24] Z. Song and B. Yang, "Ogc: Unsupervised 3d object segmentation from rigid dynamics of point clouds," *Advances in Neural Information Processing Systems*, pp. 30 798–30 812, 2022.

[25] J. Lei, C. Deng, K. Schmeckpeper, L. Guibas, and K. Daniilidis, "Efem: Equivariant neural field expectation maximization for 3d object segmentation without scene supervision," in *Proceedings of Conference on Computer Vision and Pattern Recognition*, 2023, pp. 4902–4912.

[26] M. Sung, H. Su, R. Yu, and L. J. Guibas, "Deep functional dictionaries: Learning consistent semantic structures on 3d models from functions," *Advances in Neural Information Processing Systems*, 2018.

[27] Z. Chen, K. Yin, M. Fisher, S. Chaudhuri, and H. Zhang, "Bae-net: Branched autoencoder for shape co-segmentation," in *Proceedings of International Conference on Computer Vision*, 2019, pp. 8490–8499.

[28] C. Zhu, K. Xu, S. Chaudhuri, L. Yi, L. J. Guibas, and H. Zhang, "AdaCoSeg: Adaptive shape co-segmentation with group consistency loss," in *Proceedings of Conference on Computer Vision and Pattern Recognition*, 2020, pp. 8543–8552.

[29] R. Hu, L. Fan, and L. Liu, "Co-segmentation of 3d shapes via subspace clustering," in *Computer graphics forum*, 2012, pp. 1703–1713.

[30] O. Sidi, O. Van Kaick, Y. Kleiman, H. Zhang, and D. Cohen-Or, "Unsupervised co-segmentation of a set of shapes via descriptor-space spectral clustering," in *Proceedings of SIGGRAPH Asia Conference*, 2011, pp. 1–10.

[31] S. Muralikrishnan, V. G. Kim, and S. Chaudhuri, "Tags2parts: Discovering semantic regions from shape tags," in *Proceedings of Conference on Computer Vision and Pattern Recognition*, 2018, pp. 2926–2935.

[32] M. Sung, H. Su, V. G. Kim, S. Chaudhuri, and L. Guibas, "Complementme: Weakly-supervised component suggestions for 3d modeling," *ACM Transactions on Graphics*, pp. 1–12, 2017.

[33] C. Lin, N. Mitra, G. Wetzstein, L. J. Guibas, and P. Guerrero, "Neuform: Adaptive overfitting for neural shape editing," *Advances in Neural Information Processing Systems*, 2022.

[34] Y.-J. Yuan, Y.-K. Lai, T. Wu, L. Gao, and L. Liu, "A revisit of shape editing techniques: From the geometric to the neural viewpoint," *Journal of Computer Science and Technology*, 2021.

[35] O. Michel, R. Bar-On, R. Liu, S. Benaim, and R. Hanocka, "Text2mesh: Text-driven neural stylization for meshes," in *Proceedings of International Conference on Computer Vision and Pattern Recognition*, 2022.

[36] A. Umam, C.-K. Yang, Y.-Y. Chuang, J.-H. Chuang, and Y.-Y. Lin, "Point mixswap: Attentional point cloud mixing via swapping matched structural divisions," in *Proceedings of European Conference on Computer Vision*, 2022, pp. 596–611.

[37] L. Yi, V. G. Kim, D. Ceylan, I.-C. Shen, M. Yan, H. Su, C. Lu, Q. Huang, A. Sheffer, and L. Guibas, "A scalable active framework for region annotation in 3d shape collections," *ACM Transactions on Graphics*, pp. 1–12, 2016.

[38] Y. Wang, S. Asafi, O. Van Kaick, H. Zhang, D. Cohen-Or, and B. Chen, "Active co-analysis of a set of shapes," *ACM Transactions on Graphics*, pp. 1–10, 2012.

[39] K. R. Jerripothula, J. Cai, and J. Yuan, "Quality-guided fusion-based co-saliency estimation for image co-segmentation and colocalization," *IEEE Transactions on Multimedia*, pp. 2466–2477, 2018.

[40] A. Joulin, F. Bach, and J. Ponce, "Multi-class cosegmentation," in *Proceedings of Conference on Computer Vision and Pattern Recognition*, 2012, pp. 542–549.

[41] M. Rubinstein, A. Joulin, J. Kopf, and C. Liu, "Unsupervised joint object discovery and segmentation in internet images," in *Proceedings of Conference on Computer Vision and Pattern Recognition*, 2013, pp. 1939–1946.

[42] J. C. Rubio, J. Serrat, A. López, and N. Paragios, "Unsupervised co-segmentation through region matching," in *Proceedings of Conference on Computer Vision and Pattern Recognition*, 2012, pp. 749–756.

[43] W.-C. Hung, V. Jampani, S. Liu, P. Molchanov, M.-H. Yang, and J. Kautz, "Scops: Self-supervised co-part segmentation," in *Proceedings of Conference on Computer Vision and Pattern Recognition*, 2019, pp. 14 502–14 511.

[44] S. Choudhury, I. Laina, C. Rupprecht, and A. Vedaldi, "Unsupervised part discovery from contrastive reconstruction," *Advances in Neural Information Processing Systems*, pp. 28 104–28 118, 2021.

[45] S. Liu, L. Zhang, X. Yang, H. Su, and J. Zhu, "Unsupervised part segmentation through disentangling appearance and shape," in *Proceedings of Conference on Computer Vision and Pattern Recognition*, 2021, pp. 8355–8364.

[46] A. Ziegler and Y. M. Asano, "Self-supervised learning of object parts for semantic segmentation," in *Proceedings of Conference on Computer Vision and Pattern Recognition*, 2022, pp. 14 502–14 511.

[47] C. Yu, X. Zhu, X. Zhang, Z. Wang, Z. Zhang, and Z. Lei, "Hp-capsule: Unsupervised face part discovery by hierarchical parsing capsule network," in *Proceedings of Conference on Computer Vision and Pattern Recognition*, 2022, pp. 4032–4041.

[48] A. Joulin, F. Bach, and J. Ponce, "Discriminative clustering for image co-segmentation," in *Proceedings of Conference on Computer Vision and Pattern Recognition*, 2010, pp. 1943–1950.

[49] L. Hui, J. Yuan, M. Cheng, J. Xie, X. Zhang, and J. Yang, "Superpoint network for point cloud oversegmentation," in *Proceedings of International Conference on Computer Vision*, 2021, pp. 5510–5519.

[50] L. Landrieu and M. Boussaha, "Point cloud oversegmentation with graph-structured deep metric learning," in *Proceedings of Conference on Computer Vision and Pattern Recognition*, 2019, pp. 7440–7449.

[51] L. Hui, L. Tang, Y. Shen, J. Xie, and J. Yang, "Learning superpoint graph cut for 3d instance segmentation," in *Advances in Neural Information Processing Systems*, 2022, pp. 36 804–36 817.

[52] S. Deng, Q. Dong, B. Liu, and Z. Hu, "Superpoint-guided semi-supervised semantic segmentation of 3d point clouds," in *Proceedings of Conference on Robotics and Automation*, 2022, pp. 9214–9220.

[53] J. Demantké, C. Mallet, N. David, and B. Vallet, "Dimensionality based scale selection in 3d lidar point clouds," *International Society for Photogrammetry and Remote Sensing*, pp. 97–102, 2012.

[54] L. Landrieu and G. Obozinski, "Cut pursuit: Fast algorithms to learn piecewise constant functions on general weighted graphs," *SIAM Journal on Imaging Sciences*, pp. 1724–1766, 2017.

[55] D. Paschalidou, A. O. Ulusoy, and A. Geiger, "Superquadrics revisited: Learning 3d shape parsing beyond cuboids," in *Proceedings of Conference on Computer Vision and Pattern Recognition*, 2019, pp. 10 344–10 353.

[56] D. Paschalidou, L. V. Gool, and A. Geiger, "Learning unsupervised hierarchical part decomposition of 3d objects from a single rgb image," in *Proceedings of Conference on Computer Vision and Pattern Recognition*, 2020, pp. 1060–1070.

[57] W. Liu, Y. Wu, S. Ruan, and G. S. Chirikjian, "Robust and accurate superquadric recovery: a probabilistic approach," in *Proceedings of Conference on Computer Vision and Pattern Recognition*, 2022, p. 2676.

[58] Z. Chen, A. Tagliasacchi, and H. Zhang, "Bsp-net: Generating compact meshes via binary space partitioning," in *Proceedings of International Conference on Computer Vision and Pattern Recognition*, 2020, pp. 45–54.

[59] B. Deng, S. Kulal, Z. Dong, C. Deng, Y. Tian, and J. Wu, "Unsupervised learning of shape programs with repeatable implicit parts," *Advances in Neural Information Processing Systems*, 2022.

[60] M.-H. Guo, T.-X. Xu, J.-J. Liu, Z.-N. Liu, P.-T. Jiang, T.-J. Mu, S.-H. Zhang, R. R. Martin, M.-M. Cheng, and S.-M. Hu, "Attention mechanisms in computer vision: A survey," *Computational visual media*, pp. 331–368, 2022.

[61] P. H. Le-Khac, G. Healy, and A. F. Smeaton, "Contrastive representation learning: A framework and review," *IEEE Access*, pp. 193 907–193 934, 2020.

[62] C. Sun, Z. Zheng, X. Wang, M. Xu, and Y. Yang, "Self-supervised point cloud representation learning via separating mixed shapes," *IEEE Transactions on Multimedia*, 2022.

[63] J. Li, H. Dai, H. Han, and Y. Ding, "Mseg3d: Multi-modal 3d semantic segmentation for autonomous driving," in *Proceedings of Conference on Computer Vision and Pattern Recognition*, 2023, pp. 21 694–21 704.

[64] B. Guo, L. Deng, R. Wang, W. Guo, A. H.-M. Ng, and W. Bai, "Mctnet: Multiscale cross-attention based transformer network for semantic segmentation of large-scale point cloud," *IEEE Transactions on Geoscience and Remote Sensing*, 2023.

[65] T. Jiang, Y. Wang, S. Liu, Y. Cong, L. Dai, and J. Sun, "Local and global structure for urban als point cloud semantic segmentation with ground-aware attention," *IEEE Transactions on Geoscience and Remote Sensing*, pp. 1–15, 2022.

[66] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in Neural Information Processing Systems*, 2017.

[67] S. Xie, J. Gu, D. Guo, C. R. Qi, L. Guibas, and O. Litany, "Pointcontrast: Unsupervised pre-training for 3d point cloud understanding," in *Proceedings of European Conference on Computer Vision*, 2020, pp. 574–591.

[68] X. Wu, X. Wen, X. Liu, and H. Zhao, "Masked scene contrast: A scalable framework for unsupervised 3d representation learning," in *Proceedings of Conference on Computer Vision and Pattern Recognition*, 2023, pp. 9415–9424.

[69] X. Li, J. Chen, J. Ouyang, H. Deng, S. Velipasalar, and D. Wu, "Tothepoint: Efficient contrastive learning of 3d point clouds via recycling," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 21 781–21 790.

[70] A. v. d. Oord, Y. Li, and O. Vinyals, "Representation learning with contrastive predictive coding," *arXiv preprint*, 2018.

[71] Z. Shu, C. Qi, S. Xin, C. Hu, L. Wang, Y. Zhang, and L. Liu, "Unsupervised 3d shape segmentation and co-segmentation via deep learning," *Computer Aided Geometric Design*, pp. 39–52, 2016.

[72] C. R. Qi, H. Su, K. Mo, and L. J. Guibas, "Pointnet: Deep learning on point sets for 3d classification and segmentation," in *Proceedings of Conference on Computer Vision and Pattern Recognition*, 2017, p. 652.

[73] G. Liu, S. Wei, S. Zhong, S. Huang, and R. Zhong, "Reconstruction of indoor navigation elements for point cloud of buildings with occlusions and openings by wall segment restoration from indoor context labeling," *Remote Sensing*, p. 4275, 2022.

[74] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko, "End-to-end object detection with transformers," in *Proceedings of European Conference on Computer Vision*, 2020, p. 213.

[75] A. v. d. Oord, Y. Li, and O. Vinyals, "Representation learning with contrastive predictive coding," *arXiv preprint*, 2018.

[76] C. R. Qi, L. Yi, H. Su, and L. J. Guibas, "Pointnet++: Deep hierarchical feature learning on point sets in a metric space," *Advances in Neural Information Processing Systems*, 2017.

[77] X. Chen, A. Golovinskiy, and T. Funkhouser, "A benchmark for 3d mesh segmentation," *ACM Transactions on Graphics*, pp. 1–12, 2009.

[78] L. Landrieu and M. Simonovsky, "Large-scale point cloud semantic segmentation with superpoint graphs," in *Proceedings of Conference on Computer Vision and Pattern Recognition*, 2018, pp. 4558–4567.

[79] Y. Wang, Y. Sun, Z. Liu, S. E. Sarma, M. M. Bronstein, and J. M. Solomon, "Dynamic graph cnn for learning on point clouds," *ACM Transactions on Graphics*, pp. 1–12, 2019.

[80] M. Gadelha, A. RoyChowdhury, G. Sharma, E. Kalogerakis, L. Cao, E. Learned-Miller, R. Wang, and S. Maji, "Label-efficient learning on point clouds using approximate convex decompositions," in *Proceedings of European Conference on Computer Vision*, 2020, pp. 473–491.

[81] K. Mamou, E. Lengyel, and A. Peters, "Volumetric hierarchical approximate convex decomposition," in *Game Engine Gems 3*, 2016, ch. 11, pp. 141–158.

Ardian Umam received the B.S. degree in electrical engineering from Gadjah Mada University, Indonesia, in 2014, and the M.S. degree in electrical engineering and computer science from National Chiao Tung University, Taiwan, in 2018. He is currently a PhD student in electrical engineering and computer science at National Yang Ming Chiao Tung University and a faculty member of the School of Electrical Engineering and Informatics, Bandung Institute of Technology, Indonesia. His research interests include computer vision, machine learning, data science, and artificial intelligence.

Cheng-Kun Yang received his Ph.D. degree from National Taiwan University in Computer Science and Information Engineering in 2023. His Ph.D. thesis has received the Honorable Mention in 2023 Technologies and Applications of Artificial Intelligence (TAAI) and 2023 Chinese Image Processing and Pattern Recognition Society (IPPR). He is currently a senior engineer for the multimedia development team at MediaTek, Inc. His research interests include computer vision, medical imaging, and artificial intelligence.

Jen-Hui Chuang (Senior Member, IEEE) received the BS degree in electrical engineering from National Taiwan University in 1980, the MS and Ph.D. degrees, both in electrical and computer engineering, from University of California at Santa Barbara and Urbana-Champaign University of Illinois at Urbana-Champaign in 1883 and 1991, respectively. Since then he has been on the faculty of the Department of Computer Science of National Chiao Tung University (NCTU). From 2004 to 2005, he was the Chairman of the Department of Computer and Information Science of NCTU. From 2017 to 2020 he was the Dean of the College of Computer Science of NCTU. In 2021, he served as Vice President for Academic Affairs of National Yang Ming Chiao Tung University. Dr. Chuang's research interests include signal and image processing, computer vision and pattern recognition, and robotics.

Yen-Yu Lin (Senior Member, IEEE) received the B.B.A. degree in information management, and the M.S. and Ph.D. degrees in computer science and information engineering from National Taiwan University, Taipei, Taiwan, in 2001, 2003, and 2010, respectively. He is currently a Professor with the Department of Computer Science, National Yang Ming Chiao Tung University, Taiwan. His research interests include computer vision, machine learning, and artificial intelligence.