Part2Object: Hierarchical Unsupervised 3D Instance Segmentation

Cheng Shi*, Yulin Zhang*, Bin Yang, Jiajin Tang,
Yuexin Ma, and Sibei Yang✉

School of Information Science and Technology
ShanghaiTech University
{shicheng2022,yangsb}@shanghaitech.edu.cn

Abstract. Unsupervised 3D instance segmentation aims to segment objects from a 3D point cloud without any annotations. Existing methods face the challenge of either too loose or too tight clustering, leading to under-segmentation or over-segmentation. To address this issue, we propose Part2Object, hierarchical clustering with object guidance. Part2Object employs multi-layer clustering from points to object parts and objects, allowing objects to manifest at any layer. Additionally, it extracts and utilizes 3D objectness priors from temporally consecutive 2D RGB frames to guide the clustering process. Moreover, we propose Hi-Mask3D to support hierarchical 3D object part and instance segmentation. By training Hi-Mask3D on the objects and object parts extracted from Part2Object, we achieve consistent and superior performance compared to state-of-the-art models in various settings, including unsupervised instance segmentation, data-efficient fine-tuning, and cross-dataset generalization. Code is release at `https://github.com/ChengShiest/Part2Object`.

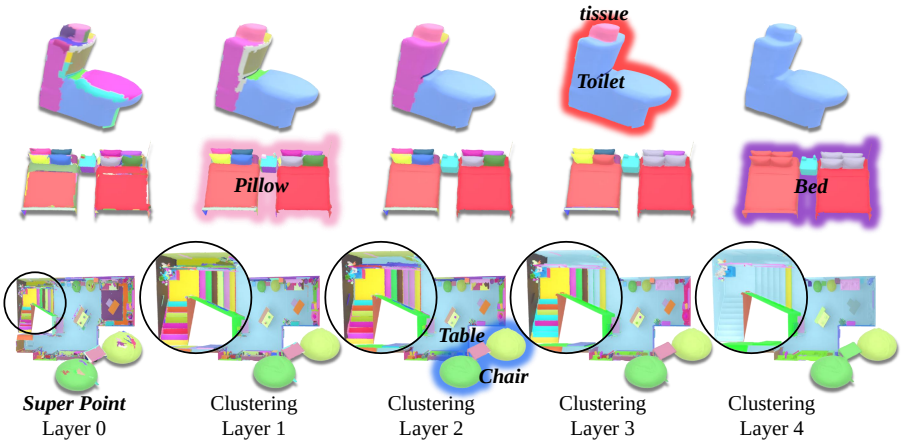
Keywords: 3D Instance Segmentation · Unsupervised Learning

1 Introduction

3D instance segmentation, parsing semantic compositions from complex point cloud scenes into distinct objects, is fundamental and crucial in real-world applications, such as mixed reality [51], autonomous navigation [3, 52], planning [23], and manipulation [17]. In recent years, 3D instance segmentation has seen substantial advancements [19, 37, 38, 42, 55, 62], improving both the accuracy and efficiency of object localization and recognition. Nevertheless, it often demands large-scale human annotations for fully-supervised training, resulting in costly and manpower-intensive efforts. While a few studies [33, 47] have initiated to address 3D instance segmentation without human annotations, their reliance on 3D scene flows from sequential point clouds restricts their applicability to single point cloud scenarios, such as indoor scenes in ScanNet [12].

∗ Equal contribution. ✉ Corresponding author.

Fig. 1: Motivation of our hierarchical clustering. Single-level clustering results in a trade-off between under-segmentation for certain objects and over-segmentation for others. In contrast, our hierarchical clustering allows for gathering and identifying objects at varying levels of clustering granularity.



In this paper, we study unsupervised 3D instance segmentation, aiming to segment class-agnostic instances from indoor 3D scenes without relying on any human labels. To address this problem, two straightforward strategies have been explored recently: 1) to apply traditional clustering [16, 32] or graph-cut methods [66] to group points into objects based on their RGB-D data, such as coordinates, colors, normal vectors, and self-supervised pretraining features [41]. 2) To project 2D instance segmentation results from 2D RGB frames onto the point cloud using the corresponding camera pose. Typically, 2D RGB frames are captured concurrently with indoor point cloud data [12, 39], while 2D instance masks are extracted using 2D unsupervised instance segmentation advancements [59–61]. However, both strategies can only work in straightforward scenes with few salient objects, as they cannot tackle the fundamental challenges inherent in unsupervised 3D instance segmentation, as outlined below.

In complex scenes, ***achieving consistent segmentation granularity for different objects within a single clustering and graph-cut result is challenging***, given the significant variations in geometric shape, color, and size among 3D objects, as well as the diverse compositions of these objects within scenes. This entails a trade-off between under-segmentation for certain objects and over-segmentation for others, making it almost impossible to attain satisfactory segmentation granularity for all of them. As shown in Fig 1, tighter grouping can accurately segment objects with simple geometric shapes and textures, such as the pillow on the bed (see layer one). However, it can also result in the over-segmentation of larger and more complex objects like the toilet in the same layer (layer one), and vice versa. ***In addition, there is a lack of an effective strategy to identify whether a point cluster or group represents a 3D instance.*** For example, the toilet back (see layer two) forms a well-grouped clus-

ter, but it does not constitute a meaningful object. The 3D geometric features are fundamentally semantically insufficient to identify objects. Recently, it has been discovered that self-supervised vision transformer (ViT) features contain information about object segmentations in 2D images [6, 60]. Building on this discovery, 3D unsupervised instance segmentation approaches [41] project the 2D self-supervised ViT features (or their inferred object segmentations) from 2D RGB frames onto 3D point clouds, thereby enhancing the perception of objectness for 3D point features. However, a 3D point typically corresponds to multiple pixels in different 2D RGB frames, and this many-to-one mapping is fragile, easily disrupted by noise features or imprecise masks on any frame, resulting in an inability to distinguish one object from others or the segmentation of objects into fragmented segments, as illustrated in Fig 3(b) and (c). In addition, despite enhanced point features, discerning nearby objects or those sharing similar semantics remains challenging due to the inherently greater complexity of 3D scenes compared to 2D images.

To tackle these challenges, we introduce a simple yet effective principle, "Gather & Aim", for the automatic discovery and segmentation of 3D objects. ***For the gathering***, inspired by the observation that tighter or looser groupings can respectively include certain objects within a scene, we propose allowing the gathering of potential objects at varying levels of grouping granularity, rather than being restricted to a single level. Specifically, we perform hierarchical clustering in the 3D scene, gradually grouping points into larger clusters, as illustrated in Fig 1. Therefore, instead of striving to improve single-layer clustering results, ours is expected that as long as each potential instance appears in one of the layers, we will be able to lock onto the segmentation mask of that instance. This mitigates excessive reliance on any single-level clustering results, thereby enhancing the clustering's adaptability to significant variations among objects and scenes. ***For the aiming***, we propose to target the 3D objectness priors, a set of candidate objects, from temporally consecutive 2D RGB frames. We leverage both the temporal consistency across frames and the prior knowledge of unsupervised object discovery within frames to extract the 3D objectness priors. Note that we jointly estimate 2D objects from multiple frames first and then approximate object-level 3D bounding boxes, departing from the point-level projection used in previous methods [41], to address the fragility issue of the many-to-one mapping from pixels to points. It avoids adhesive or scattered 3D fragments caused by inconsistencies and occlusions over 2D RGB frames. Finally, the 3D bounding boxes serve as indicators to identify the 3D instance masks within the multi-level clusters. Our "Gather & Aim" strategy enables leverage of 2D semantic priors for 3D object identification while harnessing 3D geometric priors for precise instance segmentation.

Furthermore, the taxonomy of how object parts are composed into an object is a strong yet disregarded prior information for unsupervised 3D instance segmentation. For example, the taxonomy based on the parts of a chair, including the back, arm, seat, and leg, can assist in addressing highly challenging scenarios, such as distinguishing two closely positioned and semantically similar

chairs as two separate instances. Thanks to our hierarchical clustering, we can not only identify objects but also trace their constituent parts (see the toilet and its parts in layers three and two, respectively). Therefore, we extract both objects and object parts from our hierarchical clustering and name our clustering algorithm Part2Object. Expanding on this, we enhance the 3D instance segmentation framework, Mask3D [42], to support hierarchical 3D part and instance segmentation. Our improved model, named Hi-Mask3D, takes the object and object parts from Part2Object as pseudo-labels for learning 3D instance segmentation by incorporating explicit interactions between object parts and objects.

To evaluate the effectiveness of our Part2Object clustering and Hi-Mask3D model, we conduct experiments on the challenging and cluttered indoor environments [4, 12, 39] in three settings: 1) direct evaluation on unsupervised instance segmentation, 2) application to data-efficient fine-tuning, and 3) cross-dataset generalization, all showing significant performance improvement. In summary, our contributions are multi-fold:

- We propose two key insights for unsupervised 3D instance segmentation: 1) Employing hierarchical clustering enables the gathering of objects at varying levels of clustering granularity. 2) Leveraging 3D objectness priors from temporally consecutive 2D frames as guidance, while harnessing 3D geometric priors to clustering on the point cloud for precise instance segmentation.
- Based on our insights, we propose an innovative hierarchical clustering approach, Part2Object. It progressively groups points into object parts and objects while extracting and leveraging 3D objectness priors to guide the clustering process. Our Part2Object significantly outperforms the state-of-the-art training-free unsupervised 3D instance segmentation methods by 16.8% mAP@50 on the ScanNet dataset.
- We propose Hi-Mask3D, an extension of 3D instance segmentation to support hierarchical unsupervised 3D part and instance segmentation. Experiments demonstrate that Hi-Mask3D consistently and significantly outperforms state-of-the-art models in all the settings.

2 Related Work

Unsupervised 3D instance segmentation. 3D instance segmentation is an essential task in 3D scene understanding, which aims to locate and recognize different objects in a 3D point cloud. However, the cost and labor intensity of 3D scene-level instance annotation underscore the significance of exploring challenging unsupervised instance segmentation methods. Early works utilize raw geometric information, including coordinates, colors, and normal vectors, to perform traditional clustering [15, 16, 32] for segmentation. Recently, inspired by the new paradigm in 2D unsupervised instance segmentation, some works [41, 66] introduce pseudo labels and self-training to unsupervised 3D instance segmentation. Specifically, these methods use 2D or 3D self-supervised models to extract

point cloud features, applying graph-cut algorithms to generate pseudo-labels for training and prediction. As illustrated earlier, the clustering method encounters a trade-off between under-segmentation and over-segmentation. Relying on a single graph-cut algorithm, as recent works do, falls short of achieving consistent granularity in complex indoor environments.

Transfer 2D foundation models into 3D. The rapid advancements in 2D vision have given rise to powerful foundation models [2,6,13,18,26,28,43–45,53], proving beneficial across a variety of visual tasks. In contrast to the 2D domain, 3D vision encounters challenges with data scarcity and training limitations, hindering the development of foundation models. Consequently, several works in the 3D turn to leveraging 2D foundation models to address 3D problems. In 3D instance segmentation, some works [7,64] utilize 2D vision foundation models to generate pseudo-labels for 2D images, aiding in the training and prediction of 3D models through 2D to 3D projection algorithms. Other works [31,36] employ features rather than outputs obtained from 2D foundation models, enhancing the feature extraction capabilities of 3D models through pixel-point alignment. However, relying solely on 2D features or 2D pseudo-labels poses challenges in addressing cluttered indoor scenes with stacked objects.

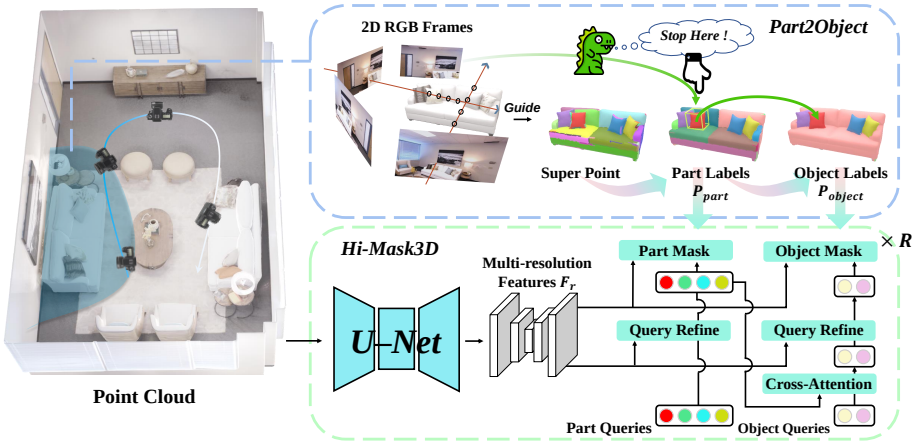
Supervised point cloud segmentation. Supervised point cloud segmentation can be divided into instance-level and part-level segmentation. The former has been significant breakthroughs in recent years [8, 19, 20, 22, 25, 27, 37, 38, 42, 50, 55, 56, 58, 62, 63]. However, these approaches demand a substantial amount of annotated data and focus on learning object-level attributes, neglecting the understanding of parts and their contributions to the composition of objects. In the realm of 3D part segmentation, various setups, including supervised [30,37,65], weakly supervised [9,10,57] instance segmentation and semantic segmentation, as well as open-world semantic segmentation [24, 36], have been extensively explored. While these approaches have demonstrated promising result, their primary emphasis lies in determining how to divide an object into parts. To integrate instance-level and part-level perspectives, we perform hierarchical clustering with the prior knowledge of how parts compose objects, unifying instance-level object identification and part-level constituent tracing.

3 Methodology

Problem definition. Unsupervised 3D instance segmentation [41, 67] is the task of generating class-agnostic masks for all foreground objects in a 3D point cloud scene solely based on raw RGB-D data, without relying on any object detection or instance segmentation annotations. The raw RGB-D data [12, 39] consists of the point cloud $P \in \mathbb{R}^{N\times 3}$ and corresponding 2D RGB frames $\{I_i\}_{i=1}^{M}$, where each image frame $I_m \in \mathbb{R}^{H\times W\times 3}$. With raw RGB-D data, 3D instance segmentation task aims to output all object masks P_{object} in the scene.

Method overview. As shown in Fig 2, our unsupervised framework has two major components: 1) the hierarchical clustering algorithm (Sec 3.1 and Sec 3.2), Part2Object, which progressively group points into larger clusters. During clus-

Fig. 2: Overview of our Part2Object hierarchical clustering and Hi-Mask3D instance segmentation framework. Part2Object extracts 3D objectness priors from consecutive 2D RGB frames and uses them to guide hierarchical clustering from points to object parts and objects. Hi-Mask3D utilizes objects and parts identified by Part2Object as pseudo-labels, learning for improving instance segmentation through the utilization of object parts.



tering, aided by objectness priors from 2D frames, we can effectively obtain both object-level and part-level segmentation masks, which serve as pseudo-labels for learning unsupervised instance segmentation. 2) The end-to-end 3D instance segmentation model Hi-Mask3D (Sec 3.3) extends the original Mask3D model [42] by explicitly predicting segmentation results of object parts and leveraging them to aid instance segmentation. By self-training on the pseudo-labels extracted from the first-stage Part2Object, Hi-Mask3D can effectively predict 3D instance segments.

3.1 Hierarchical Clustering on 3D Point Cloud

In this section, we introduce the hierarchical clustering process of our Part2Object, which starts by grouping initial points into initial clusters and then progressively merging them into larger clusters. We will sequentially cover: 1) the feature representation of points, 2) the initialization of clusters from points, and 3) the hierarchical clustering process for merging clusters. **Point Feature Representation.** Given a 3D surface point cloud P, we denote its norm and color as P^{Norm} and P^{RGB}, respectively. The 3D feature $\boldsymbol{f}_i^{\text{3D}}$ of point $p_i \in P$ comprises p_i, p_i^{Norm}, and p_i^{RGB}, representing the location, color, and geometry information, respectively. Additionally, we define point cloud 2D semantics feature $\boldsymbol{f}_i$ by projecting 2D features [6] of RGB frames into 3D space using the corresponding camera pose. Here, 2D features are extracted using the 2D self-supervised

method DINO [6], without rely on any 2D instance annotations. **Cluster Initialization and Feature Representation.** Given the vast number of points in the point cloud data [1], we initially apply the VCCS algorithm [34] to generate super-points on the point cloud, forming the first-layer clusters $\{c_i^0\}_{i=1}^{N_0}$ as follows:

$$\{c_i^0\}_{i=1}^{N_0} = \text{VCCS}(P, P^{\text{Norm}}, P^{\text{RGB}}), \tag{1}$$

where VCCS algorithm takes points' coordinates P, norms P^{Norm}, and colors P^{RGB} as input and group original N points into N_0 super-points as $\{c_i^0\}_{i=1}^{N_0}$. We consider each super-point c_i^0 as a cluster, consisting of multiple points that are closely positioned and share similar color and norm characteristics. Then, we define the feature representation for the cluster based on the features of the points within the cluster. Instead of directly averaging point features within the cluster, we compute a weight for each point and perform a weighted sum of features. These weights are determined by the similarity between each point's feature and the average feature of points within the cluster. This approach helps mitigate the influence of noisy points in the cluster, leading to more robust representations. Specifically, for the cluster c_i^0, its cluster feature $\boldsymbol{f}_i^0$ is computed as follows,

$$\boldsymbol{f}_i^0 = \sum_{p_j \in c_i^0} \frac{\text{sim}(\boldsymbol{f}_j, \bar{\boldsymbol{f}}_i^0)}{\sum_{p_j \in c_j^0} \text{sim}(\boldsymbol{f}_j, \bar{\boldsymbol{f}}_i^0)} \boldsymbol{f}_j, \;\; \texttt{where} \;\; \bar{\boldsymbol{f}}_i^0 = \sum_{p_j \in c_i^0} \frac{1}{|c_i^0|} \boldsymbol{f}_j. \tag{2}$$

The $p_j \in c_i^0$ denotes each point p_j within the cluster c_i^0, and $\boldsymbol{f}_j$ is the feature of point p_j. The $\bar{\boldsymbol{f}}_i^0$ is the average feature of points within c_i^0, and $\text{sim}(\cdot,\cdot)$ denotes the computation of cosine similarity. As depicted in the example in Fig 3(a), our cluster feature $\boldsymbol{f}_i^0$ demonstrates greater robustness compared to $\bar{\boldsymbol{f}}_i^0$. For simplicity, we abbreviate the operations in Equ 2 as function $\texttt{FU}(\cdot)$, which computes the cluster feature for the inputted cluster based on point features, *e.g.*, $\boldsymbol{f}_i^0 = \texttt{FU}(c_i^0)$.

Hierarchical Clustering with One Stop Criteria. Next, we group and merge the first-layer clusters $\{c_i^0\}_{i=1}^{N_0}$ to the next-layer clusters $\{c_i^1\}_{i=1}^{N_1}$ based on their features $\{\boldsymbol{f}_i^0\}_{i=0}^{N_0}$ and spatial coordinates, progressively and iteratively forming higher-hierarchical clusters $\{c_i^t\}_{i=1}^{N_t}$, where t represents the t-th layer in hierarchical clustering. In each single clustering layer, our clustering principle is that two clusters can only be merged when they are semantically similar and spatially adjacent: 1) the feature similarity between them ranks among the top K similarities between any pair of clusters. 2) The closest points between two clusters are adjacent. Specifically, when considering two clusters in the t-th layer, c_i^t and c_j^t, along with their features $\boldsymbol{f}_i^t$ and $\boldsymbol{f}_j^t$, the merging process to form the next-layer cluster c_k^{t+1} and feature $\boldsymbol{f}_k^{t+1}$ is as follows:

$$\begin{gathered} c_k^{t+1} = c_i^t \cup c_j^t \quad \texttt{if} \;\; \text{rank}(\text{sim}(\boldsymbol{f}_i^t, \boldsymbol{f}_j^t)) \leq K \;\; \texttt{and} \;\; \text{dist}(c_i^t, c_j^t) \leq T, \\ \boldsymbol{f}_k^{t+1} = \texttt{FU}(c_i^t \cup c_j^t), \end{gathered} \tag{3}$$

where rank$(\cdot)$ denotes the ranking of feature similarity among pairwise cluster similarities in this clustering layer, where a higher rank signifies a higher similarity, and dist$(\cdot, \cdot)$ computes the Euclidean distance between closest points of the two clusters. The K and T represent the threshold values for ranking and distance, respectively. However, we observe that solely using the merging metric in Equ 3 can result in the incorrect merging of parts from different objects in the early stages of clustering (*i.e.*, shallow layers of clustering). This occurs because clusters at the shallow layers are typically scattered fragments with highly local features, which hinder them from perceiving the entire object. To track this issue, we first extract 3D objectness priors, represented by a set of 3D bounding boxes of potential objects B^{3D}, detailed in Sec 3.2. Then, we design a stopping criteria based on the 3D objectness priors to prevent clusters belonging to different objects from being merged. ***Criteria***: Stop merging clusters belonging to different objects! Specifically, for any pair of clusters c_i^t and c_j^t that meet the merging metric, we extra assess their spatial relationship relative to each 3D object $b_k^{3D} \in B^{3D}$. If clusters c_i^t and c_j^t are respectively inside and outside the object b_k^{3D}, then reject their merging because they do not belong to the same object.

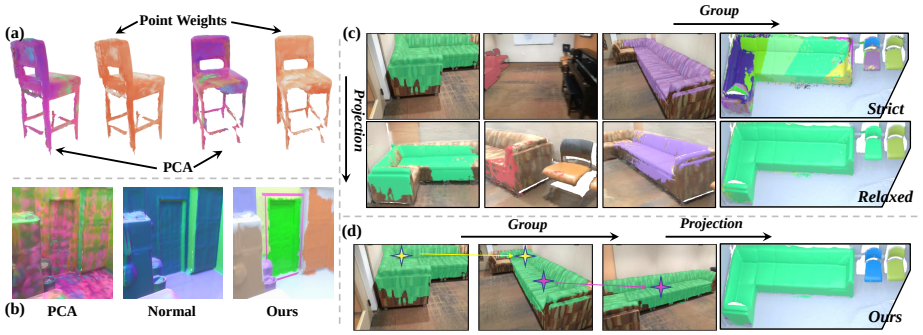
Collect Objects and Parts from Clustering. By employing hierarchical clustering with the stopping criteria, we identify clusters that stop merging with others as 3D objects, denoted as $\hat{P}_{\text{object}}$. Additionally, in hierarchical clustering, objects with complex geometric are typically merged from scattered fragments into meaningful parts, and then into complete objects. As shown in Fig 1, scattered "toilet" fragments are combined into object parts such as the toilet back and seat (see layer two), which are then formed into the object "toilet" (layer three). Therefore, we can trace back from objects $\hat{P}_{\text{object}}$ to identify the clusters from the previous level that form them, considering these clusters as potential object parts, denoted as $\hat{P}_{\text{part}}$. Both objects $\hat{P}_{\text{object}}$ and object parts $\hat{P}_{\text{part}}$ identified in hierarchical clustering serve as pseudo-labels for training our instance segmentation model, Hi-Mask3D (Sec 3.3).

3.2 3D Objectness Priors From 2D Frames

In this section, we introduce our grouping-first-then-projection pipeline to extract 3D objectness priors B^{3D} from 2D RGB frames [12], where B^{3D} is the set of 3D bounding boxes for potential objects in the scene.

Grouping-first-then-projection Pipeline. To extract 3D objectness priors, a simple strategy involves projection-first-then-grouping: projecting 2D unsupervised instance segmentation masks [46, 60] from 2D frames onto the 3D point cloud to acquire 3D masks, then grouping and merging these 3D masks based on their spatial relationships to generate 3D objects, as shown in Fig 3(c). However, due to variations in object sizes and corresponding camera poses of 2D frames across different scenes, it is nearly impossible to establish a unified grouping criterion for different objects. If the grouping criterion is too strict, allowing only masks with high overlap to be grouped, masks from multiple parts of a large object cannot be merged (see Fig 3c-"strict"). Conversely, if the criterion

Fig. 3: Robustness of our cluster features and 3D objectness priors. (a) Visualization of the first 3 PCA components and our computed weights of points in Equ 2. (b) Visualization of 2D DINO features' PCA components, 3D points' normal vectors, and our 3D object priors. The comparison between (c) the projection-first-then-grouping pipeline and (d) our grouping-first-then-projection pipeline.



is relaxed, masks that should belong to multiple objects may be merged into a single object, *e.g.*, the "chair" and "sofa" in Fig 3c-"relaxed". We attribute this limitation to the inability to rely solely on 3D spatial relationships and 3D point features (see Fig 3b) to determine if multiple 3D masks belong to the same object. To tackle this issue, we propose a grouping-first-then-projection pipeline: 1) it co-segments objects across multiple 2D frames by exploiting the temporal consistency of the frames, naturally identifying and grouping multiple 2D masks for each single object. 2) It projects the 2D masks corresponding to each object onto the 3D point cloud and then directly merges them.

Object Co-segmentation from Consecutive 2D RGB Frames. Given that 2D RGB frames are captured consecutively, we leverage their temporal consistency to co-segment multi-frame 2D masks corresponding to the same objects. First, for each RGB frame I_m, we apply the 2D unsupervised instance segmentation method MaskCut [60] to obtain the 2D masks of objects in it, denoted as O_m. As MaskCut uses the image encoder of 2D self-supervised DINO [6] to extract image feature map, we employ the same encoder to extract features for objects O_m by masked average pooling each object mask $o_m^i \in O_m$ over the frame's feature map. Next, for any adjacent pair of frames I_m and I_{m+1}, we compute the pairwise similarity between their object masks O_m and O_{m+1} based on mask features. Then, for each object mask $o_m^i \in O_m$ in frame I_m, we find the object mask $o_{m+1}^{j^*}$ in frame I_{m+1} with the highest similarity as its corresponding candidate mask as follows,

$$\begin{aligned} o_{m+1}^{j^*} &= \underset{o_{m+1}^j \in O_{m+1}}{\arg\max} \ \ \mathrm{sim}(o_{m+1}^j, o_m^i), \\ g_{m \to m+1}^i &= o_{m+1}^{j^*} \ \texttt{if} \ \mathrm{sim}(o_{m+1}^{j^*}, o_m^i) > \tau \ \texttt{else} \ \mathrm{Null}, \end{aligned} \tag{4}$$

where $\text{sim}(o^j_{m+1}, o^i_m)$ is to compute the cosine similarity between mask features of object masks o^j_{m+1} and o^i_m. The $o^{j^*}_{m+1}$ in frame I_{m+1} is the candidate object mask of the object mask o^i_m in frame I_m. Notably, only when the similarity between the object mask o^i_m and its candidate object mask $o^{j^*}_{m+1}$ exceeds the threshold π, they are considered to belong to the same object. After determining the same objects between any adjacent frames, $g^i_{m \to m+1}$, we propagate this "sameness" to all adjacent frames to identify the same objects across all frames. Consequently, we can gather objects and identify all 2D masks in different frames of each object.

3D Objectness Priors. Given that we have grouped 2D masks into different objects, for each object, we can directly project its 2D masks onto the 3D point cloud and seamlessly merge their 3D masks into a unified whole [64], forming the object's 3D mask. Furthermore, instead of directly utilizing the 3D masks of objects, we utilize the 3D bounding boxes $B^{3\text{D}}$ of their masks to guide hierarchical clustering. This decision is based on treating the collected 3D objects from 2D frames as objectness priors and leaving precise segmentation to the clustering process itself.

3.3 Hi-Mask3D and Learning Objective

In this section, we introduce our 3D instance segmentation framework, Hi-Mask3D. It extends and adapts the 3D instance segmentation model, Mask3D [42], to predict and utilize object parts to improve 3D instance segmentation.

Introduction to Mask3D. The 3D instance segmentation model Mask3D [42] mainly consists of three components: 1) a feature backbone, a sparse convolutional U-net [11] that takes a colored point cloud as input and yields multi-resolution feature matrices $\{\boldsymbol{F}_r\}$, where each $\boldsymbol{F}_r \in \mathbb{R}^{M_r \times C}$ of size M_r. 2) A mask module predicts instance masks B corresponding to instance queries $Q \in \mathbb{R}^{K \times C}$ by computing the dot product between queries Q and scene features $\{\boldsymbol{F}_r\}$. 3) A query refinement module, a transformer decoder [54] that cross-attends K instance queries Q with multi-resolution feature matrices $\{\boldsymbol{F}_r\}$. In the r-th decoder layer of query refinement module, the cross-attention between instance queries Q_r and feature matrix $\boldsymbol{F}_r$ is formulated as follows,

$$Q_r = \text{softmax}(Q_{r-1}\boldsymbol{F}_r^T / \sqrt{C})\boldsymbol{F}_r. \tag{5}$$

Note that for simplicity of demonstration, we omit the QKV projection [54] and intermediate instance mask B [42] in the cross-attention formulation.

Hi-Mask3D with Hierarchical-Aware Detector. Our Hi-Mask3D modifies the query refinement module of Mask3D to adapt for both object and object part segmentation. Specifically, we start with two sets of queries Q^o and Q^p representing object queries and object part queries, respectively. Next, in each decoder layer, we no longer directly perform cross-attention between object queries and feature matrix to update object queries. Instead, we perform hierarchical attention: 1) conducting cross-attention between object part queries and feature matrix, and 2) updating object queries using part queries. This enables explicit

interaction between object part queries and object queries, which helps to leverage the information from object parts to improve object segmentation. In the r-th decoder layer, the hierarchical attention is formulated as follows:

$$\begin{aligned} Q_r^p &= \text{softmax}(Q_{r-1}^p \boldsymbol{F}_r^T / \sqrt{C}) \boldsymbol{F}_r, \\ Q_r^o &= \text{softmax}(Q_{r-1}^o (Q_r^p)^T / \sqrt{C}) Q_r^p. \end{aligned} \tag{6}$$

The last-layer part queries and object queries undergo the mask module to obtain object part and part segmentation results, P_{part} and P_{object}, respectively.

Training Objective. Following Mask3D [42], we adopt bipartite graph algorithm [5, 48] to match prediction P_{object} and P_{part} with pseudo-labels $\hat{P}_{\text{object}}$ and $\hat{P}_{\text{part}}$, respectively. To supervise segmentation masks, we combine DICE [14] loss and Binary Cross Entropy (BCE) loss [29]. Similar to [41], we also employ iterative rounds of self-training to enhance the segmentation capabilities of Hi-Mask3D. During self-training, predictions from the previous round are utilized as pseudo-labels for the subsequent round.

4 Experiment

Datasets and Implementations. We evaluate our method on four public indoor point cloud datasets, including ScanNet [12], Scannet200 [40], S3DIS [4] and Replica [49] . Following Mask3D [42], we employ a Sparse CNN, Res16UNet34C [11], implemented in MinkowskiEngine [11], as the point cloud encoder. Our Hi-Mask3D is trained for 600 epochs with a learning rate 1e-4 and a batch size 4. Experiments are conducted using PyTorch [35] and executed on NVIDIA Tesla A40 GPUs. We evaluate the performance of class-agnostic instance segmentation using standard average precision scores. The scores consist of mAP@25, mAP@50, and mAP, representing performance metrics at IoU thresholds of 0.25, 0.5, and mean average precision, respectively. Hyper-parameter $T = 0.05$ in Equ 3 and $\tau = 0.3$ in Equ 4.

4.1 Comparison with State-of-the-Art Unsupervised Methods

Table 1 shows the results on ScanNet dataset [12] under two different settings:

Traning-free Setting. In the training-free setting, all methods rely solely on clustering [6] or graph-cut [46, 60] algorithms for prediction, as no annotations are available for use. Table 1a compares our method, Part2Object, with the previous state-of-the-art method, Unscene3D [41], which also leverages DINO features but applies graph-cut algorithms on 3D point clouds. Compared to Unscene3D, our approach demonstrates significant enhancements across all metrics, with increases of 35.2%, 16.8%, and 6.7% in mAP@25, mAP@50, and mAP, respectively. These improvements are attributed to our utilization of 3D objectness priors, which guide the clustering process.

Data-efficient Fine-tuning Setting. In the data-efficient fine-tuning setting, all methods initially undergo either pretraining or self-training. Subsequently,

Table 1: Comparison of instance segmentation results on ScanNet val.

(a) Training-free Setting. All methods do not require gradient backpropagation to obtain prediction results.

w/o training	ScanNet Val [12]		
Methods	mAP@25	mAP@50	mAP
HDBSCAN [32]	32.1	5.5	1.6
Nunes [33]	30.5	7.3	2.3
Felzenswalb [16]	38.9	12.7	5.0
CutLER [60]	7.0	0.2	0.3
Unscene3D [41]	19.9	10.0	5.9
Ours	55.1	26.8	12.6

(b) Data-efficient Setting with x% available data annotations. A 0% value indicates that the method can provide pseudo-labels, and training is conducted on them.

w/ training	mAP@50 ScanNet Val [12]				
Methods	0%	1%	5%	10%	20%
Scratch [42]	/	14.1	33.3	39.2	43.4
CSC [21]	/	22.1	39.9	43.8	48.9
HDBSCAN [32]	10.5	15.1	36.3	40.0	42.7
Felzenswalb [16]	15.2	25.3	37.2	45.7	50.0
Unscene3D [41]	23.2	28.4	46.8	55.7	60.7
Ours	32.6	44.1	64.2	68.0	72.1

they fine-tune on downstream data with varying percentages (0% represents pseudo-labels solely). Under this setting, the extraction of pseudo-labels is the primary requirement, followed by the necessity for models to exhibit strong learning capabilities. As demonstrated in Table 1b, Hi-Mask3D exhibits remarkable performance even with limited data, showcasing the robust 3D representation capabilities learned on the pseudo-labels from Part2object. Specifically, our method surpasses the previous SOTA by 15.7% and 11.4% when considering the utilization of only 1% and 20% of the available data, respectively.

4.2 Comparison on Cross-dataset Generalization

Table 2 and Table 3 show our comparisons with fully-supervised Mask3D [42] on cross-dataset generalization, including both zero-shot and dataset-efficient settings. In both settings, our Hi-Mask3D is trained on pseudo-labels from our unsupervised Part2Object on the ScanNet dataset [12], while Mask3D is trained fully supervised on ScanNet.

Cross-dataset Zero-shot Generalization Setting. In Table 2, we compare unsupervised Hi-Mask3D with fully-supervised Mask3D on three downstream datasets, including ScanNet200 [39], S3DIS [4] and Replica [49]. Compared with the fully supervised Mask3D trained on ScanNet, our Hi-Mask3D trained on pseudo-labels from Part2Object achieves consistent performance improvement without using any manually annotated data. Hi-Mask3D surpasses fully-supervised Mask3D by 10.7%, 4.8%, and 6.4% in terms of mAP@50 on ScanNet200, S3DIS, and Replica datasets, respectively. Thanks to the learning from pseudo-labels in Part2Object, Hi-Mask3D has acquired more generalizable representations of 3D objects. Unlike fully-supervised methods, which are constrained to annotated classes in the training dataset, Hi-Mask3D demonstrates robust performance in cross-dataset generalization.

Cross-dataset Data-efficient Generalization Setting. In Table 3, we conduct data-efficient training on downstream datasets, such as ScanNet200 [39] and S3DIS [4], after pre-training our Hi-Mask3D on ScanNet's pseudo-labels ex-

Table 2: Comparison on cross-dataset zero-shot generalization setting. We compare the zero-shot generalization ability of unsupervised Hi-Mask3D with fully-supervised Mask3D on three downstream datasets: ScanNet200, S3DIS and Replica.

Zero-shot	ScanNet200 Val [39]			S3DIS 6-fold [4]			Replica [49]		
Methods	m@25	m@50	mAP	m@25	m@50	mAP	m@25	m@50	mAP
Mask3D [42]	30.8	24.2	15.4	17.6	11.7	7.7	20.8	15.6	9.7
Ours	63.2	34.9	16.3	24.5	16.5	8.5	36.5	22.0	11.2

Table 3: Comparison on cross-dataset data-efficient generalization setting. We utilize pseudo-labels from Part2Object to pre-train our Hi-Mask3D and conduct data-efficient finetuning on downstream datasets, comparing them with Mask3D.

Method	Dataset	mAP@25				mAP@50			
		1%	5%	10%	20%	1%	5%	10%	20%
Scratch	ScanNet200 [39]	2.1	22.5	30.6	40.2	1.3	8.2	18.4	22.9
Ours	ScanNet200 [39]	59.4	70.4	72.3	77.8	35.4	52.8	56.2	62.8
Scratch	S3DIS [4]	1.7	10.1	20.8	45.4	1.0	2.9	9.1	25.9
Ours	S3DIS [4]	49.1	55.9	65.1	70.2	25.7	35.0	46.4	49.4

tracted from Part2Object. For ScanNet200 and S3DIS, Hi-Mask3D outperforms Mask3D 47.4% and 24.7% mAP@50 with 20% data, respectively. When limited data is available, the improvement brought by pretraining on pseudo-labels from Part2Object is even greater, with the enhancement increasing from 37.6% to 57.3% as the available data decreases from 20% to 1%.

4.3 Ablation Study

To evaluate the effectiveness of our hierarchical clustering Part2Object and 3D instance segmentation architecture Hi-Mask3D, as well as the effect of hyper-parameters, we conduct comprehensive and detailed ablation experiments, as outlined in Table 4a and Table 4b.

Ablation on Part2Object Clustering. Table 4a illustrates the variants of Part2Object clustering, starting with a baseline utilizing the single-layer clustering algorithm. The performance of this baseline yields only 6.1% in mAP and 13.8% in mAP@50. This limitation arises due to the single-layer clustering being either too loose or too tight, leading to over-clustered or under-clustered objects. Subsequently, we remove the guidance of objectness priors from our proposed clustering algorithm, denoted as "w/o `OG`". Without object priors, while objects may appear at every layer, it becomes challenging to determine which layer they belong to. Furthermore, we replace our cluster feature computation function, "`FU`$(\cdot)$", as simple averaged over features of points within the cluster, "w/o `FU`". The 2.0% decline in mAP highlights the importance of removing noise features for more robust cluster feature representations.

Table 4: Ablation study. We conduct ablation studies on ScanNet val. We validate the effectiveness of our Part2Object clustering, Hi-Mask3D, and self-training procedure.

(a) Ablation study on Part2Object clustering and Hi-Mask3D architecture. `OG`, `FU` denotes objectness guidance and cluster feature function, respectively.

Ablation on	ScanNet Val [12]		
Clustering	m@25	m@50	mAP
Baseline	37.8	13.8	6.1
w/o `OG`	43.2	20.7	10.4
w/o `FU`	44.2	21.5	10.6
Ours	**55.1**	**26.8**	**12.6**
Architecture			
Mask3D	59.0	31.0	14.2
Hi-Mask3D	**64.9**	**36.0**	**16.9**

(b) Ablation study on Self-training and Hyper-parameter.

Ablation on	ScanNet Val [12]		
Self-Training	m@25	m@50	mAP
Pseudo labels	55.1	26.8	12.6
Round 1	60.7	32.6	15.3
Round 2	64.3	35.2	16.5
Round 3	**64.9**	**36.0**	**16.9**
Hyper-parameter			
K = 0.4	55.7	22.6	10.3
K = 0.5	**57.0**	25.3	12.0
K = 0.6	55.1	**26.8**	**12.6**
K = 0.7	44.5	21.1	10.6

Ablation on Hi-Mask3D. We compare our Hi-Mask3D with Mask3D using the same pseudo-labels extracted from Part2Object. Hi-Mask3D improves Mask3D by 2.7% and 5.0% in terms of mAP and mAP@50, respectively. This demonstrates that additional part information aids in object understanding. Furthermore, through reporting the performance of self-training, we observe that after several rounds, the mAP increases from 15.3% to 16.9%.

Ablation on Hyper-parameter. We conduct ablation on the import hyper-parameter K in Equ 3. Since the number of clusters at each layer varies, we use percentages (K=0.6 indicates the top 60%) to measure how many of the top clusters from each layer can be aggregated. Table 4b shows the performance of Part2Object at different K. We use grey to indicate the default setting.

5 Conclusion and Limitations

We introduce Part2Object, an efficient hierarchical clustering algorithm, that progressively groups point clouds into object parts and objects, while leveraging 3D object-ness prior to precisely target objects. Furthermore, we propose the Hierarchical-Aware Mask3D, facilitating self-training with pseudo-object and part labels from Part2Object. Experimental results demonstrate that we consistently surpass all existing methods in both unsupervised settings and data-efficient settings. **Ethics Statement:** Given that our 2D knowledge is derived from the self-supervised models DINO, we acknowledge that biases and controversies inherent in the training data for these models may be introduced into our model.**Acknowledgment:** This work was supported by the National Natural Science Foundation of China (No.62206174) and MoE Key Laboratory of Intelligent Perception and Human-Machine Collaboration (ShanghaiTech University).

References

1. Adams, R., Bischof, L.: Seeded region growing. IEEE Transactions on pattern analysis and machine intelligence **16**(6), 641–647 (1994)
2. Amir, S., Gandelsman, Y., Bagon, S., Dekel, T.: Deep vit features as dense visual descriptors. arXiv preprint arXiv:2112.05814 (2021)
3. An, D., Wang, H., Wang, W., Wang, Z., Huang, Y., He, K., Wang, L.: Etpnav: Evolving topological planning for vision-language navigation in continuous environments. arXiv preprint arXiv:2304.03047 (2023)
4. Armeni, I., Sener, O., Zamir, A.R., Jiang, H., Brilakis, I., Fischer, M., Savarese, S.: 3d semantic parsing of large-scale indoor spaces. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 1534–1543 (2016)
5. Carion, N., Massa, F., Synnaeve, G., Usunier, N., Kirillov, A., Zagoruyko, S.: End-to-end object detection with transformers. In: European conference on computer vision. pp. 213–229. Springer (2020)
6. Caron, M., Touvron, H., Misra, I., Jégou, H., Mairal, J., Bojanowski, P., Joulin, A.: Emerging properties in self-supervised vision transformers. In: Proceedings of the International Conference on Computer Vision (ICCV) (2021)
7. Chen, R., Liu, Y., Kong, L., Chen, N., Xinge, Z., Ma, Y., Liu, T., Wang, W.: Towards label-free scene understanding by vision foundation models. In: Thirty-seventh Conference on Neural Information Processing Systems (2023)
8. Chen, S., Fang, J., Zhang, Q., Liu, W., Wang, X.: Hierarchical aggregation for 3d instance segmentation. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 15467–15476 (2021)
9. Chen, Z., Yin, K., Fisher, M., Chaudhuri, S., Zhang, H.: Bae-net: Branched autoencoder for shape co-segmentation. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 8490–8499 (2019)
10. Chibane, J., Engelmann, F., Anh Tran, T., Pons-Moll, G.: Box2mask: Weakly supervised 3d semantic instance segmentation using bounding boxes. In: European Conference on Computer Vision. pp. 681–699. Springer (2022)
11. Choy, C., Gwak, J., Savarese, S.: 4d spatio-temporal convnets: Minkowski convolutional neural networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 3075–3084 (2019)
12. Dai, A., Chang, A.X., Savva, M., Halber, M., Funkhouser, T., Nießner, M.: Scannet: Richly-annotated 3d reconstructions of indoor scenes. In: Proc. Computer Vision and Pattern Recognition (CVPR), IEEE (2017)
13. Dai, Q., Yang, S.: Curriculum point prompting for weakly-supervised referring segmentation (2024)
14. Deng, R., Shen, C., Liu, S., Wang, H., Liu, X.: Learning to predict crisp boundaries. In: Proceedings of the European conference on computer vision (ECCV). pp. 562–578 (2018)
15. Ester, M., Kriegel, H.P., Sander, J., Xu, X.: A density-based algorithm for discovering clusters in large spatial databases with noise. In: Proceedings of the Second International Conference on Knowledge Discovery and Data Mining. p. 226–231. KDD'96, AAAI Press (1996)
16. Felzenszwalb, P.F., Huttenlocher, D.P.: Efficient graph-based image segmentation. International journal of computer vision **59**, 167–181 (2004)
17. Geng, H., Xu, H., Zhao, C., Xu, C., Yi, L., Huang, S., Wang, H.: Gapartnet: Cross-category domain-generalizable object perception and manipulation via generalizable and actionable parts. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 7081–7091 (2023)

18. Ghiasi, G., Gu, X., Cui, Y., Lin, T.Y.: Open-vocabulary image segmentation. arXiv preprint arXiv:2112.12143 (2021)
19. Han, L., Zheng, T., Xu, L., Fang, L.: Occuseg: Occupancy-aware 3d instance segmentation. In: 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 2937–2946 (2020). https://doi.org/10.1109/CVPR42600.2020.00301
20. Hou, J., Dai, A., Nießner, M.: 3d-sis: 3d semantic instance segmentation of rgb-d scans. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 4421–4430 (2019)
21. Hou, J., Graham, B., Nießner, M., Xie, S.: Exploring data-efficient 3d scene understanding with contrastive scene contexts. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 15587–15597 (2021)
22. Hu, Q., Yang, B., Xie, L., Rosa, S., Guo, Y., Wang, Z., Trigoni, N., Markham, A.: Randla-net: Efficient semantic segmentation of large-scale point clouds. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 11108–11117 (2020)
23. Huang, S., Wang, Z., Li, P., Jia, B., Liu, T., Zhu, Y., Liang, W., Zhu, S.C.: Diffusion-based generation, optimization, and planning in 3d scenes. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 16750–16761 (2023)
24. Huang, Z., Wu, X., Chen, X., Zhao, H., Zhu, L., Lasenby, J.: Openins3d: Snap and lookup for 3d open-vocabulary instance segmentation. arXiv preprint arXiv:2309.00616 (2023)
25. Hui, L., Tang, L., Shen, Y., Xie, J., Yang, J.: Learning superpoint graph cut for 3d instance segmentation. In: NeurIPS (2022)
26. Kirillov, A., Mintun, E., Ravi, N., Mao, H., Rolland, C., Gustafson, L., Xiao, T., Whitehead, S., Berg, A.C., Lo, W.Y., et al.: Segment anything. arXiv preprint arXiv:2304.02643 (2023)
27. Kolodiazhnyi, M., Rukhovich, D., Vorontsova, A., Konushin, A.: Top-down beats bottom-up in 3d instance segmentation (2023). https://doi.org/10.48550/ARXIV.2302.02871, https://arxiv.org/abs/2302.02871
28. Li, B., Weinberger, K.Q., Belongie, S., Koltun, V., Ranftl, R.: Language-driven semantic segmentation. In: International Conference on Learning Representations (2022), https://openreview.net/forum?id=RriDjddCLN
29. Lin, T.Y., Goyal, P., Girshick, R., He, K., Dollár, P.: Focal loss for dense object detection. In: Proceedings of the IEEE international conference on computer vision. pp. 2980–2988 (2017)
30. Liu, J., Yu, M., Ni, B., Chen, Y.: Self-prediction for joint instance and semantic segmentation of point clouds. In: Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXII 16. pp. 187–204. Springer (2020)
31. Liu, Y., Kong, L., Cen, J., Chen, R., Zhang, W., Pan, L., Chen, K., Liu, Z.: Segment any point cloud sequences by distilling vision foundation models. arXiv preprint arXiv:2306.09347 (2023)
32. McInnes, L., Healy, J.: Accelerated hierarchical density based clustering. In: 2017 IEEE International Conference on Data Mining Workshops (ICDMW). pp. 33–42. IEEE (2017)
33. Nunes, L., Chen, X., Marcuzzi, R., Osep, A., Leal-Taixé, L., Stachniss, C., Behley, J.: Unsupervised class-agnostic instance segmentation of 3d lidar data for autonomous vehicles. IEEE Robotics and Automation Letters **7**(4), 8713–8720 (2022)

34. Papon, J., Abramov, A., Schoeler, M., Worgotter, F.: Voxel cloud connectivity segmentation-supervoxels for point clouds. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 2027–2034 (2013)
35. Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., et al.: Pytorch: An imperative style, high-performance deep learning library. Advances in neural information processing systems **32** (2019)
36. Peng, S., Genova, K., Jiang, C., Tagliasacchi, A., Pollefeys, M., Funkhouser, T., et al.: Openscene: 3d scene understanding with open vocabularies. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 815–824 (2023)
37. Qi, C.R., Yi, L., Su, H., Guibas, L.J.: Pointnet++: Deep hierarchical feature learning on point sets in a metric space. Advances in neural information processing systems **30** (2017)
38. Rethage, D., Wald, J., Sturm, J., Navab, N., Tombari, F.: Fully-convolutional point networks for large-scale point clouds. In: Proceedings of the European Conference on Computer Vision (ECCV). pp. 596–611 (2018)
39. Rozenberszki, D., Litany, O., Dai, A.: Language-grounded indoor 3d semantic segmentation in the wild. In: European Conference on Computer Vision. pp. 125–141. Springer (2022)
40. Rozenberszki, D., Litany, O., Dai, A.: Language-grounded indoor 3d semantic segmentation in the wild. In: Proceedings of the European Conference on Computer Vision (ECCV) (2022)
41. Rozenberszki, D., Litany, O., Dai, A.: Unscene3d: Unsupervised 3d instance segmentation for indoor scenes. arXiv preprint arXiv:2303.14541 (2023)
42. Schult, J., Engelmann, F., Hermans, A., Litany, O., Tang, S., Leibe, B.: Mask3D for 3D Semantic Instance Segmentation. In: International Conference on Robotics and Automation (ICRA) (2023)
43. Shi, C., Yang, S.: Edadet: Open-vocabulary object detection using early dense alignment. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 15724–15734 (2023)
44. Shi, C., Yang, S.: Logoprompt: Synthetic text images can be good visual prompts for vision-language models. arXiv preprint arXiv:2309.01155 (2023)
45. Shi, C., Yang, S.: The devil is in the object boundary: Towards annotation-free instance segmentation using foundation models. In: The Twelfth International Conference on Learning Representations (2024), `https://openreview.net/forum?id=4JbrdrHxYy`
46. Shi, J., Malik, J.: Normalized cuts and image segmentation. IEEE Transactions on pattern analysis and machine intelligence **22**(8), 888–905 (2000)
47. Song, Z., Yang, B.: Ogc: Unsupervised 3d object segmentation from rigid dynamics of point clouds. Advances in Neural Information Processing Systems **35**, 30798–30812 (2022)
48. Stewart, R., Andriluka, M., Ng, A.Y.: End-to-end people detection in crowded scenes. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 2325–2333 (2016)
49. Straub, J., Whelan, T., Ma, L., Chen, Y., Wijmans, E., Green, S., Engel, J.J., Mur-Artal, R., Ren, C., Verma, S., et al.: The replica dataset: A digital replica of indoor spaces. arXiv preprint arXiv:1906.05797 (2019)
50. Sun, J., Qing, C., Tan, J., Xu, X.: Superpoint transformer for 3d scene instance segmentation (2022)

51. Suo, S., Wong, K., Xu, J., Tu, J., Cui, A., Casas, S., Urtasun, R.: Mixsim: A hierarchical framework for mixed reality traffic simulation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 9622–9631 (2023)
52. Suomela, L., Kalliola, J., Dag, A., Edelman, H., Kämäräinen, J.K.: Benchmarking visual localization for autonomous navigation. In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision. pp. 2945–2955 (2023)
53. Tang, J., Zheng, G., Shi, C., Yang, S.: Contrastive grouping with transformer for referring image segmentation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 23570–23580 (2023)
54. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I.: Attention is all you need. Advances in neural information processing systems **30** (2017)
55. Vu, T., Kim, K., Luu, T.M., Nguyen, T., Kim, J., Yoo, C.D.: Softgroup++: Scalable 3d instance segmentation with octree pyramid grouping. arXiv preprint arXiv:2209.08263 (2022)
56. Vu, T., Kim, K., Luu, T.M., Nguyen, X.T., Yoo, C.D.: Softgroup for 3d instance segmentation on 3d point clouds. In: CVPR (2022)
57. Wang, R., Zhang, Y., Mao, J., Zhang, R., Cheng, C.Y., Wu, J.: Ikea-manual: Seeing shape assembly step by step. Advances in Neural Information Processing Systems **35**, 28428–28440 (2022)
58. Wang, W., Yu, R., Huang, Q., Neumann, U.: Sgpn: Similarity group proposal network for 3d point cloud instance segmentation. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 2569–2578 (2018)
59. Wang, X., Yu, Z., De Mello, S., Kautz, J., Anandkumar, A., Shen, C., Alvarez, J.M.: FreeSOLO: Learning to segment objects without annotations. arXiv preprint arXiv:2202.12181 (2022)
60. Wang, X., Girdhar, R., Yu, S.X., Misra, I.: Cut and learn for unsupervised object detection and instance segmentation. arXiv preprint arXiv:2301.11320 (2023)
61. Wang, Y., Shen, X., Hu, S.X., Yuan, Y., Crowley, J.L., Vaufreydaz, D.: Self-supervised transformers for unsupervised object discovery using normalized cut. In: Conference on Computer Vision and Pattern Recognition. New Orleans, LA, USA (June 2022)
62. Wang, Y., Sun, Y., Liu, Z., Sarma, S.E., Bronstein, M.M., Solomon, J.M.: Dynamic graph cnn for learning on point clouds. Acm Transactions On Graphics (tog) **38**(5), 1–12 (2019)
63. Wu, W., Qi, Z., Fuxin, L.: Pointconv: Deep convolutional networks on 3d point clouds. In: Proceedings of the IEEE/CVF Conference on computer vision and pattern recognition. pp. 9621–9630 (2019)
64. Yang, Y., Wu, X., He, T., Zhao, H., Liu, X.: Sam3d: Segment anything in 3d scenes. arXiv preprint arXiv:2306.03908 (2023)
65. Zhang, B., Wonka, P.: Point cloud instance segmentation using probabilistic embeddings. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 8883–8892 (2021)
66. Zhang, Z., Ding, J., Jiang, L., Dai, D., Xia, G.S.: Freepoint: Unsupervised point cloud instance segmentation. arXiv preprint arXiv:2305.06973 (2023)
67. Zhang, Z., Yang, B., Wang, B., Li, B.: Growsp: Unsupervised semantic segmentation of 3d point clouds. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 17619–17629 (2023)