FreePoint: Unsupervised Point Cloud Instance Segmentation

Zhikai Zhang[1], Jian Ding[1,2†], Li Jiang[3], Dengxin Dai[4], Guisong Xia[1†]
[1]Wuhan University [2]KAUST [3]CUHK-Shenzhen [4]Huawei Zurich Research Center

Abstract

Instance segmentation of point clouds is a crucial task in 3D field with numerous applications that involve localizing and segmenting objects in a scene. However, achieving satisfactory results requires a large number of manual annotations, which is a time-consuming and expensive process. To alleviate dependency on annotations, we propose a novel framework, FreePoint, for underexplored unsupervised class-agnostic instance segmentation on point clouds. In detail, we represent the point features by combining coordinates, colors, and self-supervised deep features. Based on the point features, we perform a bottom-up multicut algorithm to segment point clouds into coarse instance masks as pseudo labels, which are used to train a point cloud instance segmentation model. We propose an id-as-feature strategy at this stage to alleviate the randomness of the multicut algorithm and improve the pseudo labels' quality. During training, we propose a weakly-supervised two-step training strategy and corresponding losses to overcome the inaccuracy of coarse masks. FreePoint has achieved breakthroughs in unsupervised class-agnostic instance segmentation on point clouds and outperformed previous traditional methods by over 18.2% and a competitive concurrent work UnScene3D by 5.5% in AP. Additionally, when used as a pretext task and fine-tuned on S3DIS, FreePoint performs significantly better than existing self-supervised pre-training methods with limited annotations and surpasses CSC by 6.0% in AP with 10% annotation masks. Code will be released at https://github.com/zzk273/FreePoint.

1. Introduction

Instance segmentation on point clouds aims to segment and recognize objects in a 3D scene, serving as the foundation for a wide range of applications such as autonomous driving, virtual reality, and robot navigation. This task has received increasing attention [6, 13–16, 19, 22, 23, 37, 43, 52] for the availability of large-scale point cloud datasets [4, 12, 28, 40]. Most of the previous works focus on fully-supervised point cloud segmentation, which requires a large number of bounding boxes and per-point annotations to achieve satisfactory results. However, the annotations of point clouds are labor-intensive. For example, labeling an average scene in ScanNet takes about 22.3 minutes [12].

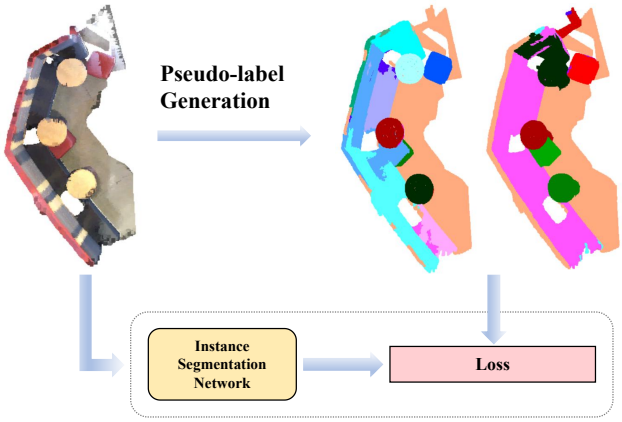

Figure 1. We propose a novel framework for unsupervised point cloud instance segmentation. In detail, we cluster points based on coordinates, colors, and self-supervised deep features. Then we use the clustered pseudo masks to perform a step-training and improve the unsupervised segmentation quality further.

To relieve the annotation requirements, some weakly-supervised 3D segmentation methods [8, 24, 51, 55, 56] and semi-supervised 3D segmentation methods [7, 20] have been proposed. Besides, some works explore unsupervised pre-training methods for 3D point clouds [18, 50, 57], mainly focusing on data-efficient scene understanding and achieving satisfactory results when fine-tuning on downstream tasks with limited annotations. These works, however, still rely on considerable box, point annotations, or a certain proportion of mask annotations to achieve competitive results. A concurrent work Unscene3D [34] explores unsupervised 3D class-agnostic instance segmentation for indoor scenes. It shows promising results while still having

†Corresponding author

large room for improvement in accuracy.

In this work, we propose a novel framework FreePoint for unsupervised point cloud instance segmentation, which can be split into three parts: (1) preprocessing and point feature extraction; (2) pseudo mask label generation by point feature based graph partitioning; (3) step-training using the pseudo labels. We first adopt plane segmentation algorithm repeatedly to split a point cloud scene into foreground points and background points. Then, for foreground points, we use a self-supervised pre-trained backbone to generate deep-learning feature embeddings for each point. To enhance our feature representation, we add coordinates and colors as extra point features. Our main motivation is that the geometry and color features are helpful for point cloud segmentation. These information has been widely adopted by some traditional point-clustering methods [2, 31, 35, 36]. To generate pseudo mask labels, we solve a bottom-up multicut [11] problem based on the affinities of point features and constructed point graphs. We propose an *id-as-feature strategy* at this stage to alleviate the randomness of the multicut algorithm and improve the pseudo labels' quality. This strategy is, in essence, an ensemble of multiple runnings of RAMA. We also adopt down-sampling and up-sampling here to make the computation affordable. These pseudo masks are used to train an existing instance segmentation model. In our work, we choose Mask3D [37] for its efficiency and good performance. Since the pseudo masks are inaccurate and the training can be unstable, we propose a weakly-supervised two-step training strategy and corresponding losses to alleviate this problem. The overview of FreePoint is shown in Figure 1.

We evaluate our method on *unsupervised class-agnostic instance segmentation*. In this setting, our method shows surprising results without any annotations, surpassing previous SOTA by a large margin. Apart from directly acquiring the class-agnostic instance masks, our method can also be used for unsupervised pre-training on 3D point clouds. The learned parameters of the backbone can be used to initialize a supervised instance segmentation model and improve final results with limited annotations.

Our contributions in this paper are three-fold:

- We propose a novel framework, FreePoint, for unsupervised point cloud instance segmentation with deep networks. Freepoint generates pseudo labels based on solving a graph partitioning problem and then uses these pseudo labels to train a 3D instance segmentation model. Our work opens up possibilities for advancing the field.
- We make great efforts to overcome many difficulties brought by the lack of manual annotations. To generate pseudo labels of higher quality, we first propose a *hybrid feature representation* for point affinity computation. Then we design an *id-as-feature strategy* to alleviate the randomness of the graph partitioning method. For better use of the noisy pseudo labels, we further propose a carefully designed *two-step training strategy and corresponding losses* to overcome pseudo labels' noise.
- We evaluate FreePoint's performance on unsupervised class-agnostic point cloud instance segmentation. It surpasses traditional unsupervised segmentation methods by over 18.2%, and even outperforms the competitive concurrent work UnScene3D [34] by 5.5% in AP. We also evaluate FreePoint's performance as a pretext task. For example, when fine-tuning on S3DIS dataset with 10% labeled masks, FreePoint outperforms training from scratch by +8.2% AP and CSC by 5.8% AP.

2. Related work

Point cloud instance segmentation Early works on point cloud instance segmentation focus on grouping points based on their affinities [13, 44, 45]. They use dense labels to train point feature encoders and segment point clouds by measuring the point affinities. 3D-SIS [17] and 3D-BoNet [52] extract bounding box proposals and classify them. Recent works prefer to group points based on predicted semantics and object centers [6, 14, 15, 19, 23]. Mask3D [37] is the first Transformer-based approach to challenge this task. We choose it as our step-training model for its high efficiency. The above works highly rely on per-point labels to achieve good results. However, acquiring such labels is labor-intensive. Some 3D instance segmentation works have been proposed these years to alleviate dependency on costly manual annotations. [7, 18, 20, 24, 51, 51, 55, 56] assume a sparse number of points is annotated and [8] use only bounding box labels. However, they still rely on considerable annotations to achieve competitive results.

Unsupervised segmentation and detection In 2D images, several works explore unsupervised object detection [10, 26, 38, 39, 48], instance segmentation [46, 47], and semantic segmentation [9, 21, 42]. In object detection area, some works [26, 38, 48] use spectral methods to discover and segment main objects in a scene. They first construct an adjacency matrix using spatial features, color features, or features from pre-trained backbones. Then the matrix's eigenvectors and eigenvalues are computed to decompose the image. Recently, a few works [46, 47] have explored unsupervised instance segmentation for 2D images and achieved satisfactory results. UnScene3D [34] has explored unsupervised 3D instance segmentation for indoor scenes. It operates on a basis of geometric oversegmentation to generate pseudo labels and refines them through self-training as many 2D works. UnScene3D shows promising results while still having large room for improvement in accuracy. The main difference between our method and this work lies in: (1) utilizing only 3D color and geometric features instead of multimodal features from 2D and 3D pre-

training backbones; (2) designing a two-step training strategy instead of a multi-round self-training strategy which is very time-costing.

3D feature representation Traditional methods [2, 31] use features like *coordinates*, *colors* and *normals* to describe each point in a scene. Following the tendency of unsupervised pre-training in 2D field, various works [3, 18, 27, 30, 50, 53, 54, 57] have been proposed recently to represent 3D features, but mostly focusing on single-object classification tasks on ShapeNet [5] or ModelNet [49]. Only a few works [18, 50, 57] focus on large-scale indoor point cloud datasets, which are important for multi-object segmentation tasks and contain far more than only one object. [18] mainly explores how to address downstream tasks in a data-efficient semi-supervised way rather than using full annotations. As a result, many works on instance segmentation and semantic segmentation train their model from scratch and can not benefit from 3D pre-training.

3. Method

Our pipeline, as shown in Figure 2, can be split into three parts: (1) preprocessing and point feature extraction; (2) pseudo mask label generation by point feature based graph partitioning; (3) step-training using the pseudo labels. Concretely, we first apply plane segmentation to separate the foreground points and background points. Then, for foreground points, we combine both traditional features (*i.e.,* coordinates and colors) and self-supervised deep-learning embeddings to represent their features. Based on it, we construct an undirected graph $\mathbf{G} = (\mathbf{V}, \mathbf{E}, \mathbf{A})$ viewing the points as vertices $\mathbf{V}$ and their connections as edges $\mathbf{E}$. $\mathbf{A}$ is an affinity cost vector measured by the affinities between point features. After this, a multicut algorithm is adopted to decompose $\mathbf{G}$ into coarse instance masks. Finally, we use the coarse masks to perform step-training with our proposed weakly-supervised loss and step-training strategy.

3.1. Preprocessing and point feature extraction

Preprocessing It is difficult to directly cluster the point clouds into instance masks and backgrounds in the unsupervised setting, since numerous inconspicuous objects are integrated into nearby backgrounds. However, we find that for indoor point cloud datasets, backgrounds include floors, walls, and ceilings, which are usually large and flat surfaces and thus can be easily removed. So we apply plane segmentation [58] to filter out major surfaces in a scene and consider them as backgrounds. In detail, we run a non-deep learning plane segmentation algorithm several times for a scene. Each fitted plane will be projected and compared with its corresponding surface of the whole indoor scene's bounding box and we will compute the IOU. If the IOU is larger than a threshold, it will be seen as part of the background and removed from the scene. After this step, the original input point cloud $\mathbf{V}_{full} \in \mathbb{R}^{N\times 6}$, which contains coordinate and color information, is divided into two subsets: foreground point cloud $\mathbf{V}_{fg} \in \mathbb{R}^{N_{fg}\times 6}$ and background point cloud $\mathbf{V}_{bg} \in \mathbb{R}^{N_{bg}\times 6}$. Since segmenting backgrounds is not the goal of instance segmentation, we only use $\mathbf{V}_{fg}$ for the next feature extracting and point cloud segmenting step.

We then perform farthest point sampling [32] to down sample $\mathbf{V}_{fg}$ into $\mathbf{V}_{sampled} \in \mathbb{R}^{N_{sampled}\times 6}$. This step is important for that: (1) it can reduce the computation cost of the following point cloud segmenting process and make the pseudo label generation on raw points affordable; (2) this down-sampling can make the point distribution more sparse. Because of the sparsity, the sampled points are farther from each other in feature space, which is beneficial for our point cloud segmenting method described in Section 3.2.

Feature extraction Since our segmenting method is based on the affinity between the feature representation of each point, we should find a way to make points closer in feature embedding space if they belong to the same object and farther otherwise. We first use self-supervised pre-trained backbones to encode points. However, we find it difficult to encode points discriminatively using deep-learning features alone, which means even points belonging to different instances can be close to each other in the feature embedding space.

Before the era of deep learning, some methods [2, 31] use traditional features to cluster points. For example, Supervoxel [31] uses features like *coordinates* and *colors* to measure the affinities between points and cluster them accordingly. Inspired by it, we use both traditional features and deep-learning features to represent each sampled point and measure their affinities in our work.

3.2. Point cloud segmenting

Preliminary Minimum-cost multicut [11] problem aims to decompose an undirected graph $\mathbf{G} = (\mathbf{V}, \mathbf{E}, \mathbf{A})$ into a set of point subsets $\{\mathbf{V}_1, \ldots, \mathbf{V}_k\}$ where $\mathbf{V}_1 \cup \ldots \cup \mathbf{V}_k = \mathbf{V}$ and $\mathbf{V}_i \cap \mathbf{V}_j = \varnothing \; \forall i \neq j$. Edges that straddle distinct clusters which decomposes $\mathbf{G}$ form the *cut* $\delta(\mathbf{V}_1, \ldots, \mathbf{V}_k)$. $\mathbf{A} \in \mathbb{R}^{\mathbf{E}}$ is an affinity cost vector. Each edge $(u, v) \in \mathbf{E}$ has a cost $\mathbf{A}_{(u,v)}$. We need to find a decomposition *cut* of the undirected graph $\mathbf{G}$ that agrees as much as possible with the affinity cost vector, minimizing the whole cost of *cut*. So if more edge cost values are negative, $\mathbf{G}$ will be decomposed into more clusters generally.

Segmenting In our work, we select RAMA [1], a rapid bottom-up multicut algorithm on GPU, to segment point

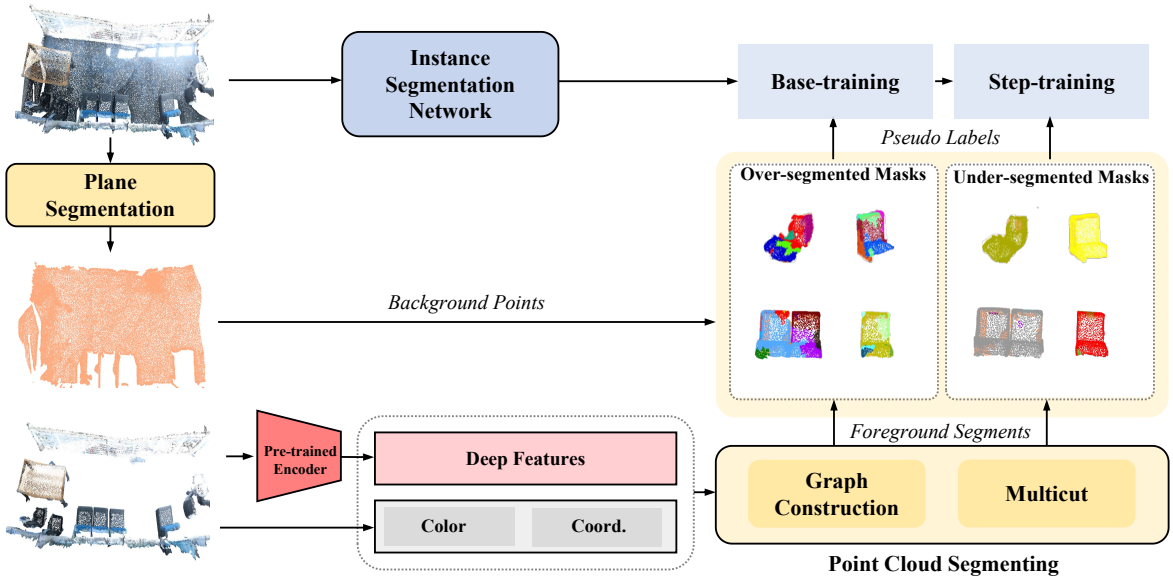

Figure 2. **Overview.** For inputted point clouds, we first use plane segmentation to filter out backgrounds. Then we represent the features for points by combining self-supervised deep features and traditional features. After that, we construct a graph and compute the edge affinity costs between points. Based on the graph, we apply a multicut algorithm to segment point clouds into coarse instance masks. These masks are adopted as pseudo labels to train a 3D instance segmentation model with our proposed weakly-supervised loss and step-training strategy.

clouds. Each $v_i \in \mathbf{V}$ is connected to the closest k_1 points $\{u_{i_1}, ..., u_{i_{k_1}}\} \in \mathbf{V}$ by edges $(v_i, u_{i_j}) \in \mathbf{E}$, where $j \in \{1, \ldots, k_1\}$. Affinity cost vector $\mathbf{A}$ is the affinities of both deep features and traditional features. For deep-learning feature embeddings $\mathbf{F} \in \mathbb{R}^{N_{sampled} \times dim}$, we calculate their cosine similarities:

$$\mathbf{A}_{(i,j),emb} = \texttt{Cos}(\mathbf{F}_i, \mathbf{F}_j). \tag{1}$$

We split $\mathbf{V}_{sampled}$ into point coordinates $\mathbf{P} \in \mathbb{R}^{N_{sampled} \times 3}$ and point colors $\mathbf{C} \in \mathbb{R}^{N_{sampled} \times 3}$. Then we compute L2 distance respectively in XYZ space and RGB space:

$$\mathbf{A}_{(i,j),xyz} = - \parallel \mathbf{P}_i, \mathbf{P}_j \parallel_2, \mathbf{A}_{(i,j),rgb} = - \parallel \mathbf{C}_i, \mathbf{C}_j \parallel_2 . \tag{2}$$

These three affinities are all normalized to have a mean value of 0 and variance of 1. The total affinity can be written as:

$$\mathbf{A} = \alpha_1 \mathbf{A}_{emb} + \alpha_2 \mathbf{A}_{xyz} + \alpha_3 \mathbf{A}_{rgb}, \tag{3}$$

where α_1, α_2, α_3 are the weights to balance the importance of different affinities. $\mathbf{G}$ will be sent to RAMA [1] based on $\mathbf{A}$ and the output is pseudo instance labels.

However, due to the characteristics of this bottom-up segmenting method, the generated coarse masks have randomness. We design an id-as-feature strategy to solve this problem and improve the pseudo labels' quality. This strategy is, in essence, an ensemble of multiple generation results. Concretely, each time t we run RAMA, every point in $\mathbf{V}_{sampled}$ will have an assigned pseudo instance label id_t. We run RAMA multiple times $\mathbf{T}$ and concatenate every id_t to form a new feature for each point in $\mathbf{V}_{sampled}$. For these id-generated features $\mathbf{IDF} \in \mathbb{R}^{N_{sampled} \times T}$, their similarities will be computed as:

$$\mathbf{A}_{(i,j),id} = \frac{1}{T} \sum_{t=1}^{T} \boldsymbol{I} \left[\mathbf{IDF}_i \left[t \right] = \mathbf{IDF}_j \left[t \right] \right] \tag{4}$$

We run RAMA again based on $\mathbf{A}_{id}$ and then preliminary pseudo instance labels $\mathbf{L}_{sampled} \in \mathbb{R}^{N_{sampled}}$ are formed. To recover to original size, we use knn to find the closest k_2 points and corresponding labels in $\mathbf{V}_{sampled}$ for each point in $\mathbf{V}_{fg}$. By majority voting of the k_2 points, we obtain $\mathbf{L}_{fg} \in \mathbb{R}^{N_{fg}}$. Then we annotate points in $\mathbf{V}_{bg}$ as background and concatenate it with $\mathbf{L}_{fg}$ to obtain final pseudo labels $\mathbf{L} \in \mathbb{R}^{N}$. The pipeline of pseudo-label generation is shown in Figure 3.

As mentioned before, RAMA generally segments the scene into more objects if more edge values are negative. When running RAMA based on $\mathbf{A}$, we will add different hyper-parameters $\sigma_{low}, \sigma_{high}$ to affinity:

$$\mathbf{A}_{final} = \mathbf{A} + \{\sigma_{low}, \sigma_{high}\}. \tag{5}$$

By changing σ, we generate coarse masks of two different segmenting levels. One is able to localize and identify most objects in the scene but fails to generate complete masks for instances. We denote these masks as base

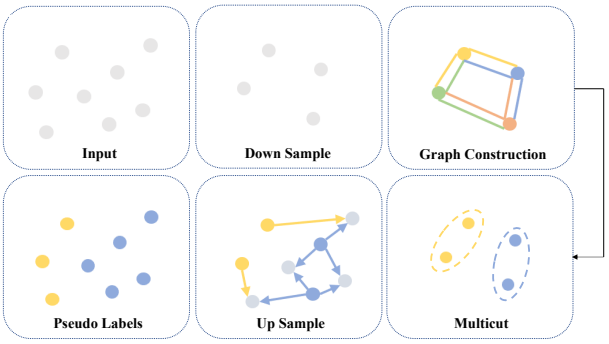

Figure 3. **Pseudo-label Generation.** In this figure, we show the complete pipeline of pseudo-label generation. For simplicity, we set $k_1 = k_2 = 2$.

masks. To overcome base masks' defects, we generate under-segmented masks with a relatively higher σ. They will be in good use for the next step following our weakly-supervised two-step training design. It is worth mentioning that when running RAMA based on $\mathbf{A}_{id}$, σ is automatically chosen to keep the number of generated instances approximately the same as the average instance number of $\mathbf{T}$ runnings of RAMA based on $\mathbf{A}_{final}$.

3.3. Training with coarse masks

To further refine the coarse masks, we aim to train a point cloud instance segmenter using these masks as pseudo labels. In our work, we choose Mask3D [37], a Transformer-based model for semantic instance segmentation, for its good performance and efficiency. Coarse masks are often inaccurate, so directly using them to train an instance segmenter in a fully-supervised way will cause unsatisfactory results. Therefore we propose two designs to solve this problem, including a new weakly-supervised loss and a step-training strategy.

Loss for weakly-supervised training In the original implementation of Mask3D [37], they use both dice loss $\mathcal{L}_{dice}$ and binary cross entropy loss $\mathcal{L}_{BCE}$ as mask loss to train. However, our pseudo labels are inaccurate, so using such per-point loss directly may lead to sub-optimal results. We propose to use these coarse masks as a kind of weak annotation and design a weakly-supervised loss.

Inspired by [8, 41, 46], we believe mask centers and bounding boxes are important for weakly-supervised training. Mask centers can help to localize instances. We compute the mean value of normalized coordinates in a predicted mask $\mathbf{m}$ and target mask $\mathbf{m}^*$ along each axis to get prediction center $c_{mean} \in (x_c, y_c, z_c)$ and target center $t_{mean} \in (x_t, y_t, z_t)$. Our model is trained to minimize the Euclidean distance between c_{mean} and t_{mean}:

$$\mathcal{L}_{mean} = \texttt{Euclidean}(avg(\mathbf{m}), avg(\mathbf{m}^*)). \tag{6}$$

We further propose a bounding box loss. Bounding box supervision enforces predictions with the correct sizes and locations. This design can further improve our work's performance. For implementation, we pick the maximum and minimum value along each axis for a predicted mask and a target mask to get two boundary point pairs (c_{max}, t_{max}) and (c_{min}, t_{min}). The Euclidean distance of each pair is summed to be our bounding-box loss. The loss can be written as:

$$\begin{aligned}\mathcal{L}_{box} = sum(&\texttt{Euclidean}(max(\mathbf{m}), max(\mathbf{m}^*)),\\ &\texttt{Euclidean}(min(\mathbf{m}), min(\mathbf{m}^*))).\end{aligned} \tag{7}$$

We compute the above losses directly on points without voxelization. Then the weighted sum of each term in weakly-supervised loss and fully-supervised loss will be our final loss, which can be written as:

$$\begin{aligned}\mathcal{L} = &\lambda_{dice}\mathcal{L}_{dice} + \lambda_{BCE}\mathcal{L}_{BCE}\\ &+ \lambda_{mean}\mathcal{L}_{mean} + \lambda_{box}\mathcal{L}_{box},\end{aligned} \tag{8}$$

where λ_{dice}, λ_{BCE}, λ_{mean}, and λ_{box} are the weights to balance the importance of different loss terms.

Step training strategy In section 3.2, we observe that our segmenting method can generate masks of different segmenting levels. For base masks, the scene will be generally split into object parts. More instances can be identified and localized in this situation, but they lack complete masks. For the under-segmented setting, the scene has fewer instance proposals, which means we will have more masks covering a whole object. However, instances in this setting are always mistakenly connected with nearby instances especially when they share similar features.

We wonder which kind of masks we should use to achieve better results. Both coarse masks have insurmountable defects if adopted as pseudo labels alone. Therefore, we explore a novel training strategy so that over-segmented and under-segmented masks can compensate for each other's shortcomings and significantly improve final results. Concretely, we use base masks as pseudo labels for the first training step. At this stage, the model is trained to segment points of similar features, regardless of whether they belong to object parts or whole objects. For the second training step, we use under-segmented masks instead. With only a few epochs, the model learns to connect mistakenly segmented object parts into a whole object. This step can improve the results of the first step by a large margin with little time cost.

However, the under-segmented masks which contain multiple objects may harm the model's performance. At

this stage, we propose an undersegmentation-ignore design to relieve this problem. Concretely, during the bipartite matching stage of the training of Mask3D, we ignore the match if the matched pseudo mask contains more than certain times the points of the predicted mask. This design is based on the insight that the model can already predict approximately correct masks and doesn't need much refinement. It ensures the model completes instance masks within a reasonable range.

The improvement in accuracy matches our intuition. The model is first trained to encode points and segment point clouds at a low level. Even though the pseudo labels we use in this step are over-segmented, the model can learn relatively good point feature representations and predict object parts. Then we use under-segmented masks to teach the model how to connect objects and predict complete instance masks.

4. Experiments

Implementation detablackils For point downsampling in preprocessing, we downsample the whole point cloud to the half of the number of original points and set $k_1 = k_2 = 4$.

Datasets We evaluate our work on two publicly available indoor 3D instance segmentation datasets ScanNet [12] and S3DIS [4]. The ScanNet dataset altogether contains 1613 scans, divided into training, validation and testing sets of 1201, 312, 100 scans respectively. We use the 20-class benchmark provided by the dataset. The S3DIS dataset contains 3D scans of 6 areas with 271 scenes in total. The dataset consists of 13 classes for instance segmentation evaluation. For unsupervised instance segmentation, we train on the training set. We report both evaluation results on the training set following UnScene3D [34] and the validation set.

Evaluation Metrics We use standard average precision as our evaluation metrics. AP_{50} and AP_{25} denote the scores with IoU thresholds of 0.5 and 0.25 respectively. AP denotes the average scores with IoU threshold from 0.5 to 0.95 with a step size of 0.05. We evaluate only instance mask AP values without considering any semantic labels.

4.1. Main Results

Unsupervised instance segmentation We mainly compare our work with a concurrent work [34], which is a recently proposed unsupervised instance segmentation method for indoor 3D scenes. It operates on a basis of geometric oversegmentation to generate pseudo labels and refines them through multi-round self-training as many works. We also compare FreePoint with some traditional clustering methods including DBSCAN [33], HDBSCAN [25] and a method originally proposed for outdoor autonomous vehicles [29]. The visualization results are shown in Figure 4.

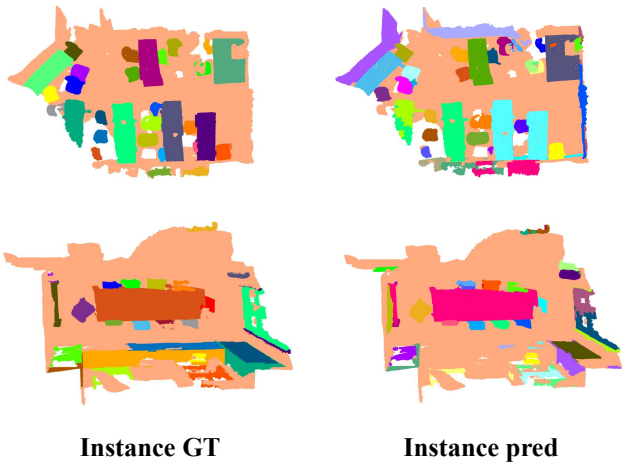

Figure 4. **Qualitative results on ScanNet.** FreePoint shows surprisingly good performance without any annotations.

Method	Train set		Val set	
	AP	AP_{50}	AP	AP_{50}
DBSCAN [33]	3.2	4.1	3.3	3.6
HDBSCAN [25]	1.6	5.5	1.9	5.4
Nunes et al. [29]	2.3	7.3	2.1	6.9
UnScene3D [34]	13.3	-	-	-
UnScene3D* [34]	15.9	32.2	-	-
FreePoint (Ours)	**21.4**	**38.7**	**18.9**	**36.4**

Table 1. **Unsupervised class-agnostic instance segmentation** on ScanNet train split and validation split. We report average precision (AP) with different IoU thresholds. We mainly compare our method with some traditional clustering methods for point clouds and some recently proposed deep-learning-based methods. '*' means the method utilizes both 2D features and 3D features. '-' means the result is not provided by the original paper and we don't have access to the code to evaluate it by ourselves. Our method improves significantly over baselines.

We report the result in Table 1. It is worth noting that UnScene3D utilizes both 3D pretraining deep features and 2D pretraining deep features, while we only use the former. For a more fair comparison, we also show the result of UnScene3D which only uses 3D features from the same pretraining method CSC [18] as FreePoint. Our method surpasses previous methods by a significant margin.

	Pre-train	AP	AP_{50}	AP_{25}
10% masks	Train from scratch	34.7	47.6	56.3
	Supervised	36.9	50.1	55.5
	PointContrast [50]	36.1 (-0.8)	49.4 (-0.7)	56.8 (+1.3)
	DepthContrast [57]	36.8 (-0.1)	49.0 (-1.1)	57.3 (+1.8)
	CSC [18]	37.1 (+0.2)	50.7 (+0.6)	57.1 (+1.6)
	FreePoint (Ours)	**42.9** (+6.0)	**54.6** (+4.5)	**61.1** (+5.6)
20% masks	Train from scratch	44.1	54.3	61.1
	Supervised	45.7	55.2	61.4
	PointContrast [50]	44.4 (-1.3)	54.8 (-0.4)	61.7 (+0.3)
	DepthContrast [57]	45.2 (-0.5)	54.9 (-0.3)	62.4 (+1.0)
	CSC [18]	46.3 (+0.6)	56.4 (+1.2)	61.5 (+0.1)
	FreePoint (Ours)	**47.4** (+1.7)	**60.2** (+5.0)	**65.9** (+4.5)

Table 2. **Supervised semantic instance segmentation** with limited instance masks. "Supervised" denotes the process of *fully-supervised pre-training* on ScanNet, succeeded by fine-tuning on S3DIS. In contrast, other methods employ *unsupervised pre-training*. The numerical values in brackets indicate the relative performance changes of unsupervised pre-training compared to their supervised counterparts.

	Pre-train	AP	AP_{50}	AP_{25}
10% scenes	Train from scratch	30.1	41.2	52.2
	Supervised	32.4	41.8	52.3
	PointContrast [50]	31.0 (-1.4)	42.2 (+0.4)	53.5 (+1.2)
	DepthContrast [57]	32.2 (-0.2)	41.5 (-0.3)	53.7 (+1.4)
	CSC [18]	32.7 (+0.3)	42.7 (+0.9)	54.4 (+2.1)
	FreePoint (Ours)	**37.2** (+4.8)	**48.1** (+6.3)	**59.3** (+7.0)
20% scenes	Train from scratch	42.1	49.5	58.3
	Supervised	44.8	51.7	59.6
	PointContrast [50]	43.7 (-1.1)	50.8 (-0.9)	60.5 (+0.9)
	DepthContrast [57]	44.0 (-0.8)	51.6 (-0.1)	62.1 (+2.5)
	CSC [18]	44.4 (-0.4)	52.9 (+1.2)	61.0 (+1.4)
	FreePoint (Ours)	**48.1** (+3.3)	**56.6** (+4.9)	**64.3** (+4.7)

Table 3. **Supervised semantic instance segmentation** with limited fully annotated point clouds. "Supervised" denotes the process of *fully-supervised pre-training* on ScanNet, succeeded by fine-tuning on S3DIS. In contrast, other methods employ *unsupervised pre-training*. The numerical values in brackets indicate the relative performance changes of unsupervised pre-training compared to their supervised counterparts.

Fine-tuning on semantic instance segmentation Since our work is unsupervised, it can also be seen as a pre-training pretext task. Apart from unsupervised class-agnostic instance segmentation, we further evaluate our work's performance as an unsupervised pre-training model. As shown in Table 2, FreePoint pre-training significantly outperforms other unsupervised pre-training methods [18, 50, 57] by a large margin, and even suppress the supervised pre-training by 6.0% AP and 1.7% AP and when using 10% and 20% training masks respectively.

We also compare the pre-training methods with different amounts of full-scene annotations. As shown in Table 3, we conduct fine-tuning experiments with only limited scenes available. Our work can still achieve satisfactory results. FreePoint pre-training outperforms other unsupervised pre-training methods, and even the supervised pre-training by 4.8% AP and 3.3% AP, when using 10% and 20% full-scene annotations respectively.

Method	AP	AP_{50}	AP	AP_{50}
Traditional	6.3	10.4	10.3	21.6
PointContrast [50]	7.6	13.3	15.7	27.9
CSC [18]	7.9	13.4	16.5	30.8
FreePoint (Ours)	8.5	15.3	18.9	36.4

Table 4. **Different feature representation methods** for generating base masks. We report the accuracy of both base masks (left block) and final results (right block). Our strategy has the best performance.

4.2. Ablation Study

In this part, we conduct ablation experiments to show the effectiveness of each designed component.

Different feature representations We explore results on different kinds of point feature representations. For features generated by various self-supervised pre-training encoders, we compare their performance in generating coarse masks and final instance segmentation results. Then we combine the best performer with traditional features and find the accuracy can be further improved. The comparison between different feature representations is shown in Table 4.

Segmenting methods Owning relatively good feature representation, there are many existing ways to segment point clouds and generate coarse masks accordingly. We compare some methods including Supervoxel [31], FreeMasks, a method proposed by [46], and spectral [26] methods. For each method, we adapt them to the ScanNet dataset and tune parameters to achieve good results as far as we can.

We observe that FreeMasks and spectral methods, which have proven successful in unsupervised object detection or segmentation tasks in the 2D field, fail to transfer to point clouds as shown in Figure 5. These two methods have two main defects due to their shared top-down mechanism. Firstly, they can only identify and localize partial objects in a crowded and cluttered 3D scene. Secondly, it is hard for these non-distance-based segmenting methods to distinguish different objects of the same semantic information even if they are far away from each other. The above two defects do not have much impact on some 2D images since they generally contain only one or a few dominant objects.

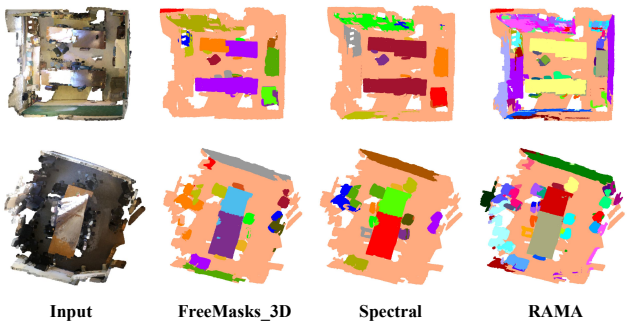

Figure 5. **Comparison with segmenting methods originally for 2D unsupervised instance segmentation.** Recent methods [26, 46] for 2D unsupervised instance segmentation fail to deal with crowded and cluttered point cloud scenes due to their top-down mechanism.

Method	AP	AP_{50}	AP	AP_{50}
Supervoxel [31]	2.4	3.5	3.8	6.9
FreeMasks_3D [46]	2.9	3.2	-	-
Spectral [26]	2.3	4.8	-	-
RAMA	5.4	10.6	13.8	24.7
RAMA-5	8.3	14.7	17.6	35.0
RAMA-10	8.5	15.3	18.9	36.4

Table 5. **Segmenting methods.** We report the accuracy of both base masks (left block) and final results (right block). '-' means failing to converge. The number after RAMA is running times $\mathbf{T}$ for evaluation of our id-as-feature strategy.

But point cloud scenes are not this case. Point clouds usually have many similar objects in each scene, leading to unsatisfactory results. RAMA's bottom-up mechanism relieves the above problems in essence. We also explore the effectiveness of our id-as-feature strategy and the impact of the running times $\mathbf{T}$ of RAMA when adopting this strategy. For each method, we report the accuracy of coarse masks and final predictions. Results are shown in Table 5.

Weakly-supervised learning design. To validate the effectiveness of our weakly-supervised design including different loss terms and undersegmentation-ignore method, we first evaluate the result of using fully-supervised loss(*i.e.,* Dice loss and BCE loss) alone, discovering that directly adopting such loss leads to unsatisfactory results. We also find that only using our proposed weakly-supervised loss terms is even much worse than only using fully-supervised loss terms. This may be attributed to that terms for weak supervision contain too little information, unable to match low-quality predictions with ground truth at the early training stage. Each loss term and undersegmentation-ignore design are validated in Table 6.

Method	AP	AP_{50}
combination(default)	18.9	36.4
- w/o $\mathcal{L}_{mean}$	14.3	30.6
- w/o $\mathcal{L}_{box}$	15.2	31.4
- w/o $\mathcal{L}_{mean}$ and $\mathcal{L}_{box}$	14.0	28.5
- w/o $\mathcal{L}_{dice}$ and $\mathcal{L}_{BCE}$	7.8	15.7
- w/o undersegmentation-ignore	16.8	36.3

Table 6. **Weakly-supervised learning design.** Each design contributes to the final results.

Method	AP	AP_{50}
base masks	8.5	15.3
under-segmented masks	9.1	12.5
train with base masks	14.2	30.5
train with under-segmented masks	6.4	13.8
Ours	18.9	36.4

Table 7. **Training strategy.** Our two-step training strategy significantly improves the accuracy.

Training strategy As mentioned in section 3.2, we can generate coarse masks of different segmenting levels by changing parameters when running RAMA. Base masks are generally over-segmented while can identify and localize most objects in the scene. Therefore after training with the base masks, we further train the model with under-segmented masks with only a few epochs. In Table 7 we report AP and AP_{50} to evaluate our design's effectiveness.

5. Discussion and Conclusion

In this work, we propose an effective framework FreePoint for unsupervised class-agnostic point cloud instance segmentation. FreePoint achieves satisfactory results compared with previous methods in this underexplored field, which proves this task is worthy of further exploration. In our experiment, we also find that top-down segmenting methods proposed in previous 2D unsupervised instance segmentation works fail to be directly adopted by point clouds as shown in Figure 5. Developing a novel unsupervised segmenting method for cluttered 3D indoor scenes may be promising. We hope our work can provide insights for future unsupervised point cloud learning works.

Acknowledgement This work is supported by National Natural Science Foundation of China grants under contracts NO.62325111 and No.U22B2011. We would like to thank Ahmed Abbas for engaging in a discussion about the usage of the RAMA algorithm.

References

[1] Ahmed Abbas and Paul Swoboda. Rama: A rapid multicut algorithm on gpu. In *CVPR*, pages 8193–8202, 2022. 3, 4

[2] Radhakrishna Achanta, Appu Shaji, Kevin Smith, Aurelien Lucchi, Pascal Fua, and Sabine Süsstrunk. Slic superpixels compared to state-of-the-art superpixel methods. *IEEE TPAMI*, 34(11):2274–2282, 2012. 2, 3

[3] Mohamed Afham, Isuru Dissanayake, Dinithi Dissanayake, Amaya Dharmasiri, Kanchana Thilakarathna, and Ranga Rodrigo. Crosspoint: Self-supervised cross-modal contrastive learning for 3d point cloud understanding. In *CVPR*, pages 9902–9912, 2022. 3

[4] Iro Armeni, Ozan Sener, Amir R Zamir, Helen Jiang, Ioannis Brilakis, Martin Fischer, and Silvio Savarese. 3d semantic parsing of large-scale indoor spaces. In *CVPR*, pages 1534–1543, 2016. 1, 6

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

[6] Shaoyu Chen, Jiemin Fang, Qian Zhang, Wenyu Liu, and Xinggang Wang. Hierarchical aggregation for 3d instance segmentation. In *ICCV*, pages 15467–15476, 2021. 1, 2

[7] Mingmei Cheng, Le Hui, Jin Xie, and Jian Yang. Sspc-net: Semi-supervised semantic 3d point cloud segmentation network. In *AAAI*, pages 1140–1147, 2021. 1, 2

[8] Julian Chibane, Francis Engelmann, Tuan Anh Tran, and Gerard Pons-Moll. Box2mask: Weakly supervised 3d semantic instance segmentation using bounding boxes. In *ECCV*, pages 681–699. Springer, 2022. 1, 2, 5

[9] Jang Hyun Cho, Utkarsh Mall, Kavita Bala, and Bharath Hariharan. Picie: Unsupervised semantic segmentation using invariance and equivariance in clustering. In *CVPR*, pages 16794–16804, 2021. 2

[10] Minsu Cho, Suha Kwak, Cordelia Schmid, and Jean Ponce. Unsupervised object discovery and localization in the wild: Part-based matching with bottom-up region proposals. In *CVPR*, pages 1201–1210, 2015. 2

[11] Sunil Chopra and Mendu R Rao. The partition problem. *Mathematical programming*, 59(1-3):87–115, 1993. 2, 3

[12] Angela Dai, Angel X Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. Scannet: Richly-annotated 3d reconstructions of indoor scenes. In *CVPR*, pages 5828–5839, 2017. 1, 6

[13] Cathrin Elich, Francis Engelmann, Theodora Kontogianni, and Bastian Leibe. 3d bird's-eye-view instance segmentation. In *GCPR*, pages 48–61. Springer, 2019. 1, 2

[14] Francis Engelmann, Martin Bokeloh, Alireza Fathi, Bastian Leibe, and Matthias Nießner. 3d-mpa: Multi-proposal aggregation for 3d semantic instance segmentation. In *CVPR*, pages 9031–9040, 2020. 2

[15] Lei Han, Tian Zheng, Lan Xu, and Lu Fang. Occuseg: Occupancy-aware 3d instance segmentation. In *CVPR*, pages 2940–2949, 2020. 2

[16] Tong He, Chunhua Shen, and Anton van den Hengel. Dyco3d: Robust instance segmentation of 3d point clouds through dynamic convolution. In *CVPR*, pages 354–363, 2021. 1

[17] Ji Hou, Angela Dai, and Matthias Nießner. 3d-sis: 3d semantic instance segmentation of rgb-d scans. In *CVPR*, pages 4421–4430, 2019. 2

[18] Ji Hou, Benjamin Graham, Matthias Nießner, and Saining Xie. Exploring data-efficient 3d scene understanding with contrastive scene contexts. In *CVPR*, pages 15587–15597, 2021. 1, 2, 3, 6, 7

[19] Li Jiang, Hengshuang Zhao, Shaoshuai Shi, Shu Liu, Chi-Wing Fu, and Jiaya Jia. Pointgroup: Dual-set point grouping for 3d instance segmentation. In *CVPR*, pages 4867–4876, 2020. 1, 2

[20] Li Jiang, Shaoshuai Shi, Zhuotao Tian, Xin Lai, Shu Liu, Chi-Wing Fu, and Jiaya Jia. Guided point contrastive learning for semi-supervised point cloud semantic segmentation. In *ICCV*, pages 6423–6432, 2021. 1, 2

[21] Tsung-Wei Ke, Jyh-Jing Hwang, Yunhui Guo, Xudong Wang, and Stella X. Yu. Unsupervised hierarchical semantic segmentation with multiview cosegmentation and clustering transformers. In *CVPR*, pages 2571–2581, 2022. 2

[22] Jean Lahoud, Bernard Ghanem, Marc Pollefeys, and Martin R Oswald. 3d instance segmentation via multi-task metric learning. In *ICCV*, pages 9256–9266, 2019. 1

[23] Zhihao Liang, Zhihao Li, Songcen Xu, Mingkui Tan, and Kui Jia. Instance segmentation in 3d scenes using semantic superpoint tree networks. In *ICCV*, pages 2783–2792, 2021. 1, 2

[24] Zhengzhe Liu, Xiaojuan Qi, and Chi-Wing Fu. One thing one click: A self-training approach for weakly supervised 3d semantic segmentation. In *CVPR*, pages 1726–1736, 2021. 1, 2

[25] Leland McInnes and John Healy. Accelerated hierarchical density based clustering. In *2017 IEEE International Conference on Data Mining Workshops (ICDMW)*, pages 33–42. IEEE, 2017. 6

[26] Luke Melas-Kyriazi, Christian Rupprecht, Iro Laina, and Andrea Vedaldi. Deep spectral methods: A surprisingly strong baseline for unsupervised semantic segmentation and localization. In *CVPR*, pages 8364–8375, 2022. 2, 7, 8

[27] Chen Min, Dawei Zhao, Liang Xiao, Yiming Nie, and Bin Dai. Voxel-mae: Masked autoencoders for pre-training large-scale point clouds. *arXiv:2206.09900*, 2022. 3

[28] Kaichun Mo, Shilin Zhu, Angel X Chang, Li Yi, Subarna Tripathi, Leonidas J Guibas, and Hao Su. Partnet: A large-scale benchmark for fine-grained and hierarchical part-level 3d object understanding. In *CVPR*, pages 909–918, 2019. 1

[29] Lucas Nunes, Xieyuanli Chen, Rodrigo Marcuzzi, Aljosa Osep, Laura Leal-Taixé, Cyrill Stachniss, and Jens Behley. Unsupervised class-agnostic instance segmentation of 3d lidar data for autonomous vehicles. *IEEE Robotics and Automation Letters*, 7(4):8713–8720, 2022. 6

[30] Yatian Pang, Wenxiao Wang, Francis EH Tay, Wei Liu, Yonghong Tian, and Li Yuan. Masked autoencoders for point cloud self-supervised learning. In *ECCV*, pages 604–621. Springer, 2022. 3

[31] Jeremie Papon, Alexey Abramov, Markus Schoeler, and Florentin Worgotter. Voxel cloud connectivity segmentation-supervoxels for point clouds. In *CVPR*, pages 2027–2034, 2013. 2, 3, 7, 8

[32] Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. *NeurIPS*, 30, 2017. 3

[33] Anant Ram, Sunita Jalal, Anand S Jalal, and Manoj Kumar. A density based algorithm for discovering density varied clusters in large spatial databases. *International Journal of Computer Applications*, 3(6):1–4, 2010. 6

[34] David Rozenberszki, Or Litany, and Angela Dai. Unscene3d: Unsupervised 3d instance segmentation for indoor scenes. *arXiv preprint arXiv:2303.14541*, 2023. 1, 2, 6

[35] Radu Bogdan Rusu. Semantic 3d object maps for everyday manipulation in human living environments. *KI-Künstliche Intelligenz*, pages 345–348, 2010. 2

[36] Radu Bogdan Rusu and Steve Cousins. 3d is here: Point cloud library (pcl). In *ICRA*, pages 1–4. IEEE, 2011. 2

[37] Jonas Schult, Francis Engelmann, Alexander Hermans, Or Litany, Siyu Tang, and Bastian Leibe. Mask3d for 3d semantic instance segmentation. *arXiv:2210.03105*, 2022. 1, 2, 5

[38] Gyungin Shin, Samuel Albanie, and Weidi Xie. Unsupervised salient object detection with spectral cluster voting. In *CVPR*, pages 3971–3980, 2022. 2

[39] Oriane Siméoni, Gilles Puy, Huy V Vo, Simon Roburin, Spyros Gidaris, Andrei Bursuc, Patrick Pérez, Renaud Marlet, and Jean Ponce. Localizing objects with self-supervised transformers and no labels. *arXiv:2109.14279*, 2021. 2

[40] Shuran Song, Fisher Yu, Andy Zeng, Angel X Chang, Manolis Savva, and Thomas Funkhouser. Semantic scene completion from a single depth image. In *CVPR*, pages 1746–1754, 2017. 1

[41] Zhi Tian, Chunhua Shen, Xinlong Wang, and Hao Chen. Boxinst: High-performance instance segmentation with box annotations. In *CVPR*, pages 5443–5452, 2021. 5

[42] Wouter Van Gansbeke, Simon Vandenhende, Stamatios Georgoulis, and Luc Van Gool. Unsupervised semantic segmentation by contrasting object mask proposals. In *ICCV*, pages 10052–10062, 2021. 2

[43] Thang Vu, Kookhoi Kim, Tung M Luu, Thanh Nguyen, and Chang D Yoo. Softgroup for 3d instance segmentation on point clouds. In *CVPR*, pages 2708–2717, 2022. 1

[44] Weiyue Wang, Ronald Yu, Qiangui Huang, and Ulrich Neumann. Sgpn: Similarity group proposal network for 3d point cloud instance segmentation. In *CVPR*, pages 2569–2578, 2018. 2

[45] Xinlong Wang, Shu Liu, Xiaoyong Shen, Chunhua Shen, and Jiaya Jia. Associatively segmenting instances and semantics in point clouds. In *CVPR*, pages 4096–4105, 2019. 2

[46] Xinlong Wang, Zhiding Yu, Shalini De Mello, Jan Kautz, Anima Anandkumar, Chunhua Shen, and Jose M Alvarez. Freesolo: Learning to segment objects without annotations. In *CVPR*, pages 14176–14186, 2022. 2, 5, 7, 8

[47] Xudong Wang, Rohit Girdhar, Stella X Yu, and Ishan Misra. Cut and learn for unsupervised object detection and instance segmentation. *arXiv:2301.11320*, 2023. 2

[48] Yangtao Wang, Xi Shen, Yuan Yuan, Yuming Du, Maomao Li, Shell Xu Hu, James L Crowley, and Dominique Vaufreydaz. Tokencut: Segmenting objects in images and videos with self-supervised transformer and normalized cut. *arXiv:2209.00383*, 2022. 2

[49] Zhirong Wu, Shuran Song, Aditya Khosla, Fisher Yu, Linguang Zhang, Xiaoou Tang, and Jianxiong Xiao. 3d shapenets: A deep representation for volumetric shapes. In *CVPR*, pages 1912–1920, 2015. 3

[50] Saining Xie, Jiatao Gu, Demi Guo, Charles R Qi, Leonidas Guibas, and Or Litany. Pointcontrast: Unsupervised pre-training for 3d point cloud understanding. In *ECCV*, pages 574–591. Springer, 2020. 1, 3, 7

[51] Xun Xu and Gim Hee Lee. Weakly supervised semantic point cloud segmentation: Towards 10x fewer labels. In *CVPR*, pages 13706–13715, 2020. 1, 2

[52] Bo Yang, Jianan Wang, Ronald Clark, Qingyong Hu, Sen Wang, Andrew Markham, and Niki Trigoni. Learning object bounding boxes for 3d instance segmentation on point clouds. *NeurIPS*, 32, 2019. 1, 2

[53] Xumin Yu, Lulu Tang, Yongming Rao, Tiejun Huang, Jie Zhou, and Jiwen Lu. Point-bert: Pre-training 3d point cloud transformers with masked point modeling. In *CVPR*, pages 19313–19322, 2022. 3

[54] Renrui Zhang, Ziyu Guo, Peng Gao, Rongyao Fang, Bin Zhao, Dong Wang, Yu Qiao, and Hongsheng Li. Point-m2ae: multi-scale masked autoencoders for hierarchical point cloud pre-training. *arXiv:2205.14401*, 2022. 3

[55] Yachao Zhang, Zonghao Li, Yuan Xie, Yanyun Qu, Cuihua Li, and Tao Mei. Weakly supervised semantic segmentation for large-scale point cloud. In *AAAI*, pages 3421–3429, 2021. 1, 2

[56] Yachao Zhang, Yanyun Qu, Yuan Xie, Zonghao Li, Shanshan Zheng, and Cuihua Li. Perturbed self-distillation: Weakly supervised large-scale point cloud semantic segmentation. In *ICCV*, pages 15520–15528, 2021. 1, 2

[57] Zaiwei Zhang, Rohit Girdhar, Armand Joulin, and Ishan Misra. Self-supervised pretraining of 3d features on any point-cloud. In *ICCV*, pages 10252–10263, 2021. 1, 3, 7

[58] Qian-Yi Zhou, Jaesik Park, and Vladlen Koltun. Open3d: A modern library for 3d data processing. *arXiv:1801.09847*, 2018. 3